

16
2oj.



Universidad Nacional Autónoma de México

FACULTAD DE INGENIERIA

DISEÑO DE UNA TARJETA DE FUNCIONES MULTIPLES

T E S I S

Que para obtener el Título de:

INGENIERO (Computación)

P R E S E N T - A:

CARLOS CALDERON GUTIERREZ

BENIGNO A. GONZALEZ NUÑEZ

JOSE CAMPOS ROSAS

DIRECTOR: M. I. LAURO SANTIAGO CRUZ

TESIS CON
FALLA DE ORIGEN

1992



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

INDICE

INTRODUCCION	1
--------------	---

CAPITULO 1

CONSIDERACIONES SOBRE EL DISEÑO DE SISTEMAS DE ADQUISICION DE DATOS

1.1 INTRODUCCION	3
1.2 ETAPAS FUNCIONALES DE LA TARJETA	4
1.3 COMPUTADORAS PERSONALES	11

CAPITULO 2

ELEMENTOS QUE INTEGRAN LA TARJETA DE FUNCIONES MULTIPLES

2.1 INTRODUCCION	13
2.2 MULTIPLEXAJE DE LAS SEÑALES ANALOGICAS	14
2.3 ACONDICIONAMIENTO DE SEÑALES	15
2.4 METODOS DE CONVERSION A/D	23
2.5 CIRCUITOS DE MUESTREO/RETEN	33
2.6 PUERTOS ENTRADA/SALIDA DIGITALES	36
2.7 CONVERTIDOR DIGITAL/ANALOGICO	45
2.8 CONVERSION DIGITAL/ANALOGICA EN EL DISEÑO	52
2.9 CIRCUITOS DE SINCRONIA DE LA TARJETA DE FUNCIONES MULTIPLES	59

CAPITULO 3

PROGRAMACION DE LA TARJETA DE FUNCIONES MULTIPLES (TFM)

3.1 ESTRUCTURA GENERAL PARA LA PROGRAMACION DE LA TFM	81
3.2 PROGRAMACION DE LA TFM PARA LA ADQUISICION DE DATOS DE ORIGEN ANALOGICO	83
3.3 PROGRAMACION DE LA TFM PARA LA SALIDA DE SEÑALES EN FORMA ANALOGICA	90
3.4 PROGRAMACION DE LA TFM PARA EL PUERTO DE ENTRADA/SALIDA DIGITAL	91
3.5 SIMULACION DE LA TFM	93

CONCLUSIONES

98

BIBLIOGRAFIA

99

APENDICES

APENDICE 1

APENDICE 2

APENDICE 3

APENDICE 4

APENDICE 5

APENDICE 6

INTRODUCCION

El presente trabajo surge como respuesta a la necesidad que existe en los laboratorios de instrumentación de contar con un equipo que ayude en la recolección de la información proveniente de algún experimento o de un fenómeno físico real, y poder utilizarla con fines estadísticos o de control. Aunado a que este dispositivo deberá ser versátil y de bajo costo, se pensó en la construcción de una tarjeta interfaz, a través de la cual pudiéramos capturar una o varias señales externas en forma eléctrica, ya sea digital o analógica, y almacenarla en una computadora personal. Para que esta interfaz pueda ser utilizada en alguna aplicación de control, contará con salidas eléctricas digitales o analógicas; por ejemplo, para accionar una válvula, cerrar algún interruptor, etc.

Las computadoras personales han incrementado su popularidad en laboratorios, estaciones de prueba, y líneas de producción, como plataforma para la automatización de adquisición de datos. Debido a que ofrecen una gran flexibilidad y características de costo/productividad atractivas, pueden ser usadas para el proceso de datos, administración, ayudas de pruebas y graficación, adicionales a la adquisición de datos y a la facilidad de elaboración de algún algoritmo empleado en su análisis. En forma complementaria, un usuario de computadoras personales puede elegir dentro de una gran variedad de lenguajes de programación y herramientas de paquetería disponibles en el mercado y a bajo costo.

Por todo esto, los sistemas de adquisición basados en computadoras superan por mucho a otro tipo de métodos de captación, como son graficadores sobre papel fotosensible, térmico, etc. o de grabadoras digitales; además de que éstos requieren de mantenimiento y de gastos en provisiones como papel, cintas, etc., dado lo anterior el costo se incrementa sustancialmente.

Por lo que el principal objetivo es el diseño de una tarjeta interfaz, la cual es un dispositivo que permite la comunicación electrónica entre otros dos dispositivos (uno de ellos una computadora personal) cuyas señales a intercambiar no son compatibles. Será diseñada de forma tal que puede aceptar señales externas, y convertirlas para que la computadora las pueda reconocer. Adicionalmente, el usuario podrá, de una manera sencilla, a través de paneles o menus de selección en la computadora, decirle el número y

tipo de señales que tendrá en la entrada, para que se efectúe una lectura correcta.

A continuación se da un resumen del contenido de los capítulos en que está dividida la Tesis.

En el capítulo 1, se presenta un esquema general sobre las partes que componen un sistema de adquisición de datos; en el capítulo 2, se muestra un entorno teórico de todos y cada uno de los bloques que integran la tarjeta de funciones múltiples, así como el desarrollo específico de cada uno de éstos; en el capítulo 3, se explican los puntos necesarios para establecer la comunicación entre la tarjeta y la computadora, señales de control, direcciones de memoria, etc. y la programación necesaria. Además, se describe como se efectuará una simulación de la tarjeta ya funcionando y el *software* necesario (debido a que no se construirá la tarjeta) y al final de este capítulo se presentan las conclusiones del trabajo y recomendaciones del mismo, a continuación la bibliografía y finalmente los apéndices.

CAPITULO 1

CONSIDERACIONES IMPORTANTES SOBRE EL DISEÑO DE SISTEMAS DE ADQUISICION DE DATOS

1.1 INTRODUCCION

Utilizando como base una computadora personal se pueden implementar sistemas de adquisición automática de datos. Con ella se miden señales en tiempo real de diferentes formas, algunas de ellas se muestran en la Fig.1.1.

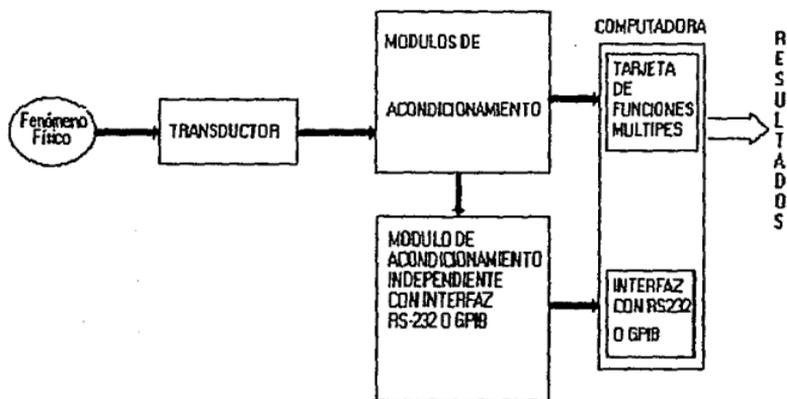


Fig.1.1. Etapas principales para la adquisición de datos

Como se puede ver en la figura, existen dos posibilidades de diseño de un sistema automático de adquisición de datos, una forma es diseñar el sistema utilizando la instrumentación clásica para medir las señales. Los datos adquiridos son concentrados a través de un módulo de adquisición independiente, y posteriormente son enviados a la computadora.

O bien, sin ser concentrados, son enviados conforme son adquiridos a la computadora personal (PC). El envío de los datos se puede efectuar a través de un canal de comunicación estandar tal como el IEE-488 ó el RS-232.

Otra opción de diseño es adquirir las señales a través de una tarjeta de funciones múltiples localizada dentro de la PC.

Dependiendo de la aplicación práctica que se presente, uno de estos métodos puede ser una mejor opción que el otro. Para nuestro caso, se seleccionó el método correspondiente al diseño de un sistema de adquisición de datos basado en una tarjeta de funciones múltiples.

Los equipos de instrumentación son a menudo usados cuando la señal tiene propiedades que le hacen imposible a la tarjeta de adquisición interpretarla adecuadamente. Por ejemplo, existen señales que tienen cientos de Megahertz que sólo un osciloscopio, con un gran ancho de banda, podría digitalizar apropiadamente, mientras que la tarjeta de adquisición de datos no lo podría hacer. Sin embargo, muchas señales no tienen las propiedades mencionadas anteriormente. Para la mayoría de los propósitos una tarjeta de funciones múltiples de adquisición de datos es la mejor solución, y ésta es la opción que se presenta en nuestro trabajo.

Por lo tanto, a la tarjeta de funciones múltiples la podemos definir como una interfaz, que permite el manejo de señales entrada/salida de tipo analógico/digital, que combina la conversión analógico/digital-digital/analógico, que incluye además circuitos temporizadores, y puertos digitales y analógicos en una sola tarjeta.

1.2 ETAPAS FUNCIONALES DE LA TARJETA

La tarjeta de funciones múltiples a desarrollar en el presente trabajo está integrada principalmente por las siguientes etapas.

1.2.1 Transductores

Como es sabido, un transductor sensa un fenómeno físico, y debido a su construcción emite una señal eléctrica, con un valor proporcional a los cambios que ocurren en el fenómeno.

Estos dispositivos tendrán que trabajar en el rango especificado para la tarjeta de funciones múltiples, algunos tipos de transductores más comunes en el mercado son:

TERMICOS

- . Termopares
- . Detectores resistivos de temperatura
- . Termistores
- . Circuitos integrados sensores de temperatura

MECANICOS

- . Transductores de flujo
- . Transductores de presión

ELECTRICOS

- . Electrodo
- . Elementos pasivos
- . Semiconductores

OPTICOS

- . Fotodetectores

ULTRASONIDO

- . Diafragmas
- . Etc.

1.2.2 Captación de señales

En esta etapa se define el tipo de señales de entrada a la tarjeta de funciones múltiples. Se pueden manejar tanto analógicas como digitales. Dentro de las analógicas podemos tener entradas sencillas (si son referidas a una tierra común) o de entrada diferencial (cada una de ellas es referidas a su propia tierra). Mediante el uso de señales

diferenciales se puede eliminar el error debido al ruido, ya que el amplificador utilizado sustrae las señales de modo común presente en las terminales. Cuando se utilizan entradas diferenciales el número de canales se reduce a la mitad.

Además, tendremos entradas/salidas digitales, las cuales se presentan al estar sensando fenómenos físicos que nos den resultados en forma de pulsos eléctricos. Así mismo, en determinados casos, se tendrán salidas que controlen algún proceso dado, a través de una señal digital.

1.2.3 Entradas analógicas

En el diseño de la tarjeta de funciones múltiples, y para el caso de entradas analógicas, se deberán considerar principalmente los siguientes aspectos:

- Tipo de señales a manejar
- Número de canales
- Tasa de muestreo
- Resolución
- Rango
- Ruido
- No linealidad

Las señales analógicas, como hemos visto, pueden ser de dos tipos, diferenciales y sencillas.

Las señales diferenciales se usan para poder eliminar los errores correspondientes al ruido, debido a que sustraen el ruido, adquirido de manera común en las terminales.

Las señales eléctricas indeseables, presentes en el voltaje de salida, se clasifican como ruido. Estas señales pueden ocurrir como voltajes inducidos en la señal de tierra o en los conductores por donde se trasmite la señal o por campos magnéticos parásitos, o variaciones del voltaje de línea. Lo más importante en esta consideración es que estas señales de ruido son señales no deseadas que llegan al amplificador operacional.

Las variaciones y desviaciones pueden considerarse como ruido de frecuencia muy variable. Cuando se observa el voltaje de salida de un amplificador operacional con un

osciloscopio, se ve una señal de voltajes de ruido al azar denominado "estática". Las frecuencias de estos voltajes de ruido varían de 0.01 Hz. hasta MHz.

El ruido se genera por cualquier material que se encuentre arriba del 0 absoluto (-273°C). Al parecer el ruido no se podrá extraer del todo si es que deseamos eliminarlo.

Las señales sencillas, o no diferenciales, se usan cuando las señales de entrada son de un nivel de voltaje alto, mayores de 1 volt, la longitud de las terminales de la fuente a la entrada de la tarjeta serán menores a 5 metros y todas las señales de entrada estarán compartiendo una referencia común de tierra.

1.2.4 Multiplexaje de las señales

Una vez conectadas la señales, se canalizarán una por una a la siguiente etapa, por medio de un circuito multiplexor. Debemos tomar en cuenta que podría hacerse de otra forma, pero resultaría muy problemático, ya que para el uso que se le daría tendrían que hacerse tantas tarjetas como canales se quisieran. Además de no ser práctico, esto haría el diseño más complejo.

El multiplexaje es una técnica común que es usada para incrementar el número de canales y manejarlos por un solo convertidor analógico/digital.

1.2.5 Acondicionamiento de la señal

Las señales eléctricas provenientes de los transductores se generan a un nivel de voltaje o corriente dado. La tarjeta de adquisición de datos deberá trabajar a niveles de corriente y de voltaje estándares, de tal manera que se pueda incorporar una etapa de acondicionamiento, la cual amplificará, atenuará o acoplará impedancias de la señal eléctrica según sea el caso.

Con base en la experiencia, y por lo general, la señal proveniente de la mayoría de los transductores siempre deberá ser amplificada. Esto es porque se ha observado que la mayoría de los transductores emiten señales de voltaje del orden de los milivolts (mV). Cuando éste no sea el caso, se deberán acondicionar los voltajes y corrientes, para evitar daños al operador y a la computadora. Para conseguir la resolución más alta posible, la variación de la señal de entrada deberá ser amplificada o atenuada, teniendo como limitante los valores de voltaje aceptados por el convertidor analógico-digital.

Otra tarea común de acondicionamiento de una señal es la linealización. Muchos módulos de acondicionamiento están hechos para trabajar con diferentes tipos de transductores, de tal forma que amplifican y linealizan las señales.

Generalmente el proceso de linealización se efectúa a través de circuitos (*hardware*), liberando el tiempo consumido en ecuaciones de linealización, que se efectúan a través de programación (*software*).

Una de las aplicaciones donde es necesario el acondicionamiento de señales es el campo médico, donde utilizan transductores para poder registrar los signos vitales de los pacientes, y con un patrón reconocible de señales se determina el estado de los mismos.

Otra función importante en la etapa de acondicionamiento es el aislamiento, ya que si el sistema que se está monitoreando contiene transitorios de alto voltaje, y éstos pasan a través de la computadora, podrían dañar al operador o paciente según sea el caso.

Otro aspecto importante en el acondicionamiento de señales es el uso de etapas de filtrado para las señales no deseadas. Una aplicación típica es el filtro para la frecuencia de ruido de 60 Hz, emitida por sistemas estandar de corriente alterna.

1.2.6 Muestreo y retención

Cuando la velocidad de variación de una señal analógica es muy alta, y esta señal que ya ha sido multiplexada debe ser digitalizada, se tiene aparentemente un problema. Dicho problema consiste en que los componentes que realizarán ese trabajo no pueden muestrear los cambios de la señal, por lo que se requiere del uso de un circuito de retención. La función de estos circuitos es muestrear el dato y retenerlo el tiempo suficiente para que sea muestreado.

Como observamos, la etapa anterior es de gran utilidad para muestrear y digitalizar los datos, para así poderlos canalizar de alguna forma a algún dispositivo de almacenamiento de la PC, y como se mencionó en un principio, procesarla según los requerimientos del usuario.

1.2.7 Conversión A/D

Una vez realizado el muestreo, el convertidor A/D tomará este nivel de la señal y lo convertirá en información digital o binaria, a este proceso se le conoce como digitalización.

La frecuencia de muestreo del convertidor, en conjunto con el número de canales a explorar, definen la tasa de muestreo de cada canal individual. Esto es, la tasa de muestreo varía en razón inversa al número de canales a muestrear. Por ejemplo, las señales de audio convertidas a señales eléctricas, por medio del micrófono, tienen componentes de frecuencia de hasta 20 KHz, y para digitalizar esta señal, la Teoría de Muestreo de Nyquist dice que se debe de muestrear en más del doble del valor máximo de la frecuencia, para este caso se necesitará una tasa de muestreo mayor a los 40 KHz.

Dentro de las características principales del proceso de conversión A/D tenemos:

- Resolución: está definida por el número de bits que el convertidor A/D usa para representar a la señal analógica.
- Rango: se refiere a los niveles de voltaje máximo y mínimo que el convertidor A/D pueda cuantificar.

El rango, la resolución, y la ganancia disponibles en una tarjeta de adquisición de datos determinarán el valor del voltaje mínimo a detectar.

Un error que puede afectar a la señal de entrada analógica es el ruido. Como es sabido, el ruido reduce la resolución del convertidor conforme el nivel de ruido se aproxima a 1 LSB (Bit Menos Significativo); por lo que el convertidor no puede distinguir entre el incremento de la señal de una magnitud de un ancho del código ó el nivel del ruido equivalente.

1.2.8 Entradas / Salidas digitales

Las señales de entrada/salida digitales se usan en los sistemas de adquisición de datos para conformar sistemas de control, de generación de señales o de temporización. En cada caso, los parámetros importantes incluyen el número de líneas digitales disponibles, la tasa con la cual los datos

digitales pueden ser entrada/salida de estas líneas, y la capacidad de manejo de las líneas.

1.2.9 Salidas analógicas

Generalmente este tipo de salidas son requeridas en un sistema de control. Estas salidas provienen de un convertidor D/A, cuyas características determinan la calidad de señal de salida.

Dentro de los parámetros importantes de un convertidor D/A tenemos: el tiempo de establecimiento, la velocidad de respuesta y la resolución.

- El tiempo de establecimiento y la velocidad de respuesta determinan que tan rápido puede adecuar el convertidor D/A los niveles de la señal de salida.
- La velocidad de respuesta, es la máxima tasa de cambio que el convertidor D/A puede producir en la señal de salida. Un tiempo de establecimiento pequeño y una velocidad de respuesta alta, permiten generar señales de alta frecuencia, por lo que es necesario que en corto tiempo se adecúe el cambio en la salida para un nuevo nivel de voltaje en la entrada.
- La resolución en la salida es similar a la resolución a la entrada, esto es, el número de bits en el código digital generarán el valor mínimo del voltaje en la salida analógica. Un gran número de bits reduce la magnitud de cada incremento de voltaje a la salida.

1.2.10 Etapa de control

Una etapa que es de gran importancia es la correspondiente al control de la Tarjeta, dicho control tiene que ser por medio de una programación. Además, podemos mencionar que las aplicaciones de este tipo de tarjeta son amplias, por el simple hecho de poder recibir cualquier señal proveniente de algún transductor.

Por otro lado, la ganancia y el error de *offset* pueden ser calibrados utilizando la combinación de circuitos electrónicos con la programación.

La calibración a través de los circuitos electrónicos se realiza aplicando un voltaje conocido en la entrada, mientras se ajusta con algún potenciómetro en la tarjeta, las características de la señal deseada en la salida.

La calibración mediante programación se realiza considerando el voltaje de error, que se obtiene al conectar las señales de entrada a tierra, y mediante programación compensar esta diferencia.

Los circuitos electrónicos para la adquisición de datos, junto con la programación, los transductores y el acondicionamiento de señales harán un sistema de adquisición de datos completo.

La buena operación de un sistema depende de muchos factores, asociados con cada uno de estos componentes.

1.3 COMPUTADORAS PERSONALES

Según el tipo de computadora será la velocidad de procesamiento de la información. En las aplicaciones donde se requiera de un procesamiento en tiempo real, se tendrán que saber de antemano algunas especificaciones de las computadoras, en particular la velocidad de procesamiento, así como los niveles de voltaje que maneja cada uno de sus puertos, para que se efectúe una conexión adecuada con la tarjeta.

Otros factores a considerar en la elección de una computadora son los siguientes:

1. La capacidad de memoria
2. La capacidad y velocidad del disco duro
3. El tipo de monitor
4. Los puertos de comunicación serial y paralelo
5. Los tipos de aplicación:
 - Laboratorio
 - Portátil
6. La versatilidad

La capacidad de memoria y la velocidad del disco duro nos permiten determinar el volumen de información que puede procesar y/o almacenar la computadora, esto estará determinado por el número de muestras de las señales del fenómeno físico que se pretende almacenar, y por el tipo de respuesta deseada.

El tipo de monitor influye en la representación de los resultados obtenidos en el proceso. Esto es una mayor resolución, y así mostrarlos en forma más respresentativa en el monitor.

Los puertos de comunicación serial y paralelo son algunos de los medios que permitan la interacción entre la computadora y el mundo exterior.

El puerto paralelo, como su nombre lo dice, transmite/recibe varias señales al mismo tiempo por un número definido de líneas. Este tipo de conexión es recomendable en distancias cortas, y cuando se requiere de altas velocidades de transmisión de información.

Para el caso del puerto serial, en una misma línea de comunicación se envía toda la información, para lo cual, necesitamos adicionar a la información que se requiere transmitir/recibir, algunos pulsos de control.

Tipos de aplicación, éstas dependen del lugar fisico donde se efectúen procesos de adquisición de datos.

Respecto a la versátilidad, podemos decir que dependiendo del ~~software~~ que maneje el equipo, éste será más amigable al usuario y podrá tener más alternativas de proceso que algún otro.

CAPITULO 2

ELEMENTOS QUE INTEGRAN LA TARJETA DE FUNCIONES MULTIPLES

2.1 INTRODUCCION

Para tener una visión clara del funcionamiento de la tarjeta de adquisición de datos, se presenta un diagrama a bloques de las partes más importantes que las componen. Para nuestro trabajo se propone el diagrama mostrado en la fig.2.1, y la descripción de los elementos que la integran.

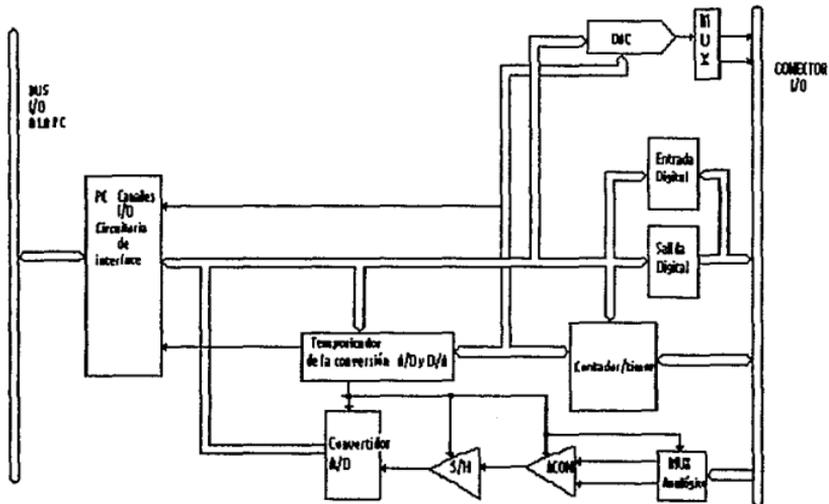


Fig.2.1 Diagrama de bloques de la tarjeta de funciones múltiples.

De acuerdo a la secuencia de eventos del diagrama, a continuación se describen cada uno de sus bloques, comenzando con un entorno teórico y terminando con el circuito final.

2.2 MULTIPLEXAJE DE LAS SEÑALES ANALÓGICAS

El multiplexaje significa que se canalizan por un solo medio n entradas, direccionadas por un número determinado de líneas de control.

Como la tarjeta de adquisiciones de datos deberá tener 16 líneas de entrada, éstas se harán pasar por un circuito, el cual determinará, por la posición de las líneas de control, la forma en que se irán pasando por su salida única.

Al variar las líneas de selección (A, B, C, D), fig.2.2, se efectuará la selección de uno de los 16 canales de entrada al circuito multiplexor a la vez, estas cuatro líneas nos dan 16 combinaciones posibles, por lo que se podrán seleccionar 16 canales analógicos.

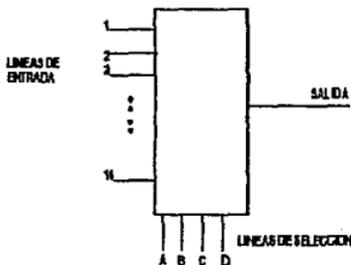


Fig.2.2 Circuito multiplexor

Para efectuar el proceso de multiplexaje con la tarjeta, se utilizan dos circuitos multiplexores, debido a que estaremos manejando 16 canales analógicos sencillos u ocho diferenciales. Los multiplexores estarán activados por el modo de operación (sencillo o diferencial).

Las señales analógicas entran a la circuitería a través de los multiplexores C10 y C11 (fig.2.3), estos dispositivos son dos selectores analógicos, cada uno cuenta con 8 canales, direccionables a través de 3 líneas binarias. Con

estas líneas (A, B, C) se direccionan los B puertos de cada multicanal y con la línea D se seleccionarán los multiplexores CIO ó CII, a través del selector analógico (dos tiros un polo). Como se observa en la fig.2.3, se utiliza el mismo selector para controlar el Modo (circuito CI2), con el cual se podrá seleccionar si son señales sencillas o diferenciales. Las dos primeras salidas del selector analógico (X,Y) definen la entrada al amplificador de instrumentación.

2.3 ACONDICIONAMIENTO DE SEÑALES

El objetivo del acondicionamiento de señales es el entregar una señal al convertidor A/D que tenga la amplitud óptima, para que se aproveche al máximo el rango de conversión disponible, y cada muestra sea lo más representativo de la señal analógica. Es decir, atenuar o amplificar la señal analógica y posteriormente entregarla al convertidor analógico-digital (A/D).

Las señales analógicas están divididas en dos categorías: unipolares y diferenciales. Una señal unipolar ó sencilla aparece en una terminal única y se mide con respecto a la tierra del circuito. Las señales unipolares son usualmente señales de un alto nivel de voltaje. Para distancias cortas, la señal se transmite sobre un cable único y su voltaje esta referido a la tierra del circuito. Para distancias mayores se utiliza un par de cables entrelazados para minimizar la captación de ruidos, uno de los cables del par está puesto a tierra y el otro transporta la señal.

Las señales de bajo nivel de voltaje se manejan generalmente como señales diferenciales, que requieren de dos cables de señal. La señal de información es la diferencia entre las tensiones de los dos cables, medido cada uno de ellos con respecto a tierra. La mayoría de los transductores de bajo nivel de voltaje generan señales diferenciales.

El amplificador más útil para medición, instrumentación o control, es el amplificador de instrumentación. Este está diseñado con varios amplificadores operacionales, y resistencias de precisión, lo cual hace al circuito en extremo estable y útil cuando es importante la exactitud.

Un primer pariente del amplificador de instrumentación, de bajo costo, es el amplificador diferencial básico.

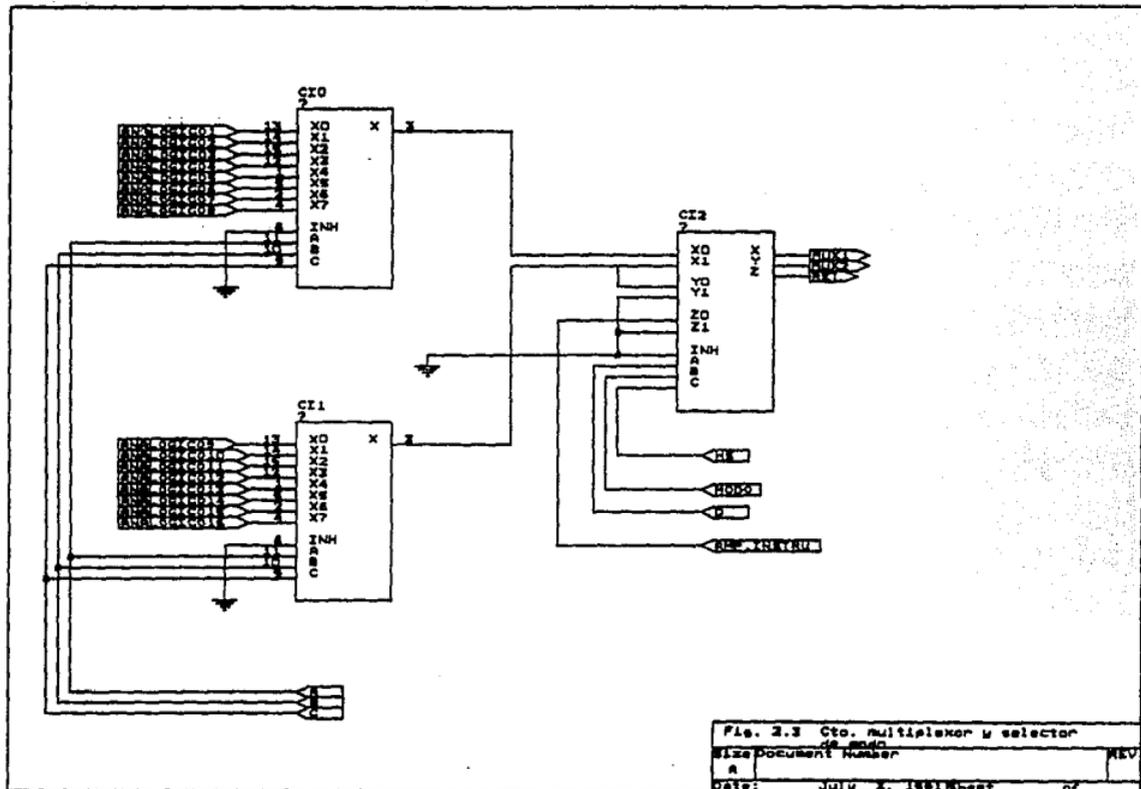


Fig. 2.3 Cto. multiplexor y selector
 de 2 a 1
 Size Document Number REV
 A
 Date: JULY 2, 1974/MSJ 07

2.3.1 Amplificador diferencial básico

El amplificador diferencial básico puede medir, y también amplificar, señales débiles que quedan enterradas en señales mucho más intensas.

Cuatro resistencias de precisión al 1% y un amplificador operacional componen un amplificador diferencial, como se muestra en la fig.2.4. Las señales de entrada se aplican en las terminales positiva y negativa, E_1 y E_2 respectivamente. Si la fuente E_1 se reemplaza por un corto circuito, la fuente E_2 ve un amplificador inversor con ganancia de $-m$.

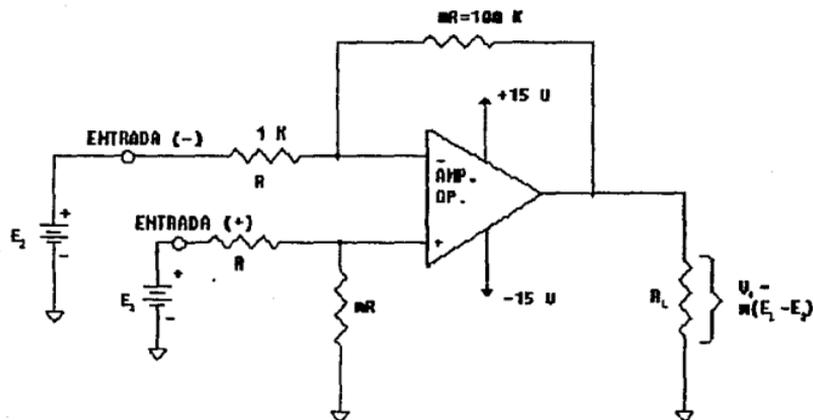


Fig.2.4 Amplificador diferencial básico

Si ponemos ahora a E_2 en corto circuito, E_1 se divide en las resistencias R y mR , por lo que el voltaje aplicado en la terminal positiva ve un amplificador no inversor con una ganancia de $(m+1)$ y considerando el divisor de voltaje se tendrá una ganancia $+m$. Finalmente, sumando los efectos tanto de E_1 como de E_2 , presentes en las entradas $(+)$ y $(-)$ respectivamente, V_0 es igual a:

$$V_0 = mE_1 - mE_2 = m(E_1 - E_2) \quad (2.1)$$

Esta ecuación muestra que el voltaje de salida del amplificador diferencial V_0 es proporcional a la diferencia en voltaje aplicado a las entradas $(+)$ y $(-)$. El

multiplicador m se denomina ganancia diferencial y se establece por las razones de las resistencias.

De la ecuación 2.1 puede esperarse que cuando $E_1=E_2$, $V_0=0$. Es decir, la aplicación de un voltaje de modo común da como resultado un voltaje de salida igual a cero volts.

2.3.2 Voltaje de modo común

El entorno del transductor puede inducir ruido eléctrico sobre la señal proveniente del mismo, que puede ser comparable en magnitud o incluso mayor que ella, y si la señal proveniente del transductor es unipolar, la señal amplificada será la suma del ruido más ella misma. Si la señal proveniente del transductor es comparable o menor en magnitud que el ruido, la relación señal a ruido, SNR, en la entrada del amplificador es muy baja, y el ruido se amplifica juntamente con la señal. Sin embargo, si se obtiene una señal diferencial del transductor, puede utilizarse un amplificador diseñado para amplificar la señal diferencial y rechazar la señal en modo común.

La salida del amplificador diferencial debe ser 0V cuando $E_1 = E_2$. El modo más simple de aplicar voltajes iguales es alambrear ambas entradas juntas y conectarlas al voltaje fuente, tal voltaje de entrada se denomina voltaje de entrada en modo común, E_{cm} . V_0 será 0V si las razones de resistencias son iguales (mR a R para la ganancia del amplificador inversor, es igual a mR del circuito divisor de voltaje). Prácticamente las razones de resistencias se igualan por la instalación de un potenciómetro en serie con una resistencia, como se muestra en la fig.2.5.

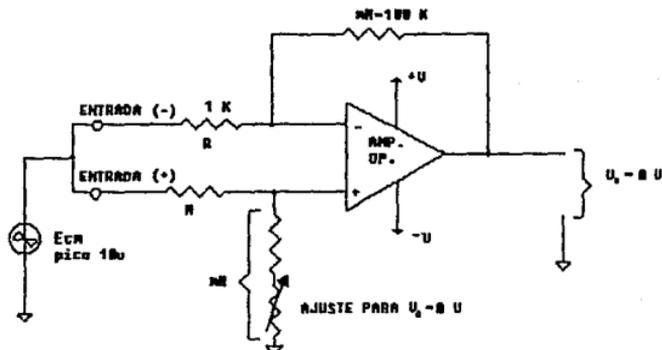


Fig.2.5 Circuito divisor de voltaje

El potenciómetro se afina hasta que V_0 se reduce a un valor despreciable. Esto causa que la ganancia en modo común, V_0/E_{cm} , se aproxime a 0V. Esta es la característica de un amplificador diferencial que permite que una señal débil se capte, sacándola de una señal más intensa. Puede ser posible arreglar el circuito de modo que la señal más intensa (no deseada) sea el voltaje de entrada en modo común y que la señal débil sea el voltaje diferencial de entrada. Entonces el voltaje de salida del amplificador diferencial contendrá una versión amplificada del voltaje diferencial de entrada.

2.3.3 Mejoras al amplificador diferencial básico

Hay dos desventajas del amplificador diferencial básico:

- Tiene baja impedancia de entrada y el cambio de ganancia es difícil, debido a que las razones de resistencias deben igualarse estrechamente. La primera desventaja se elimina al reforzar o aislar las entradas con seguidores de voltaje; y la segunda, agregando tres resistencias más al amplificador ya reforzado. El circuito resultante se muestra en la fig.2.6. La alta resistencia se preserva por los seguidores de voltaje.

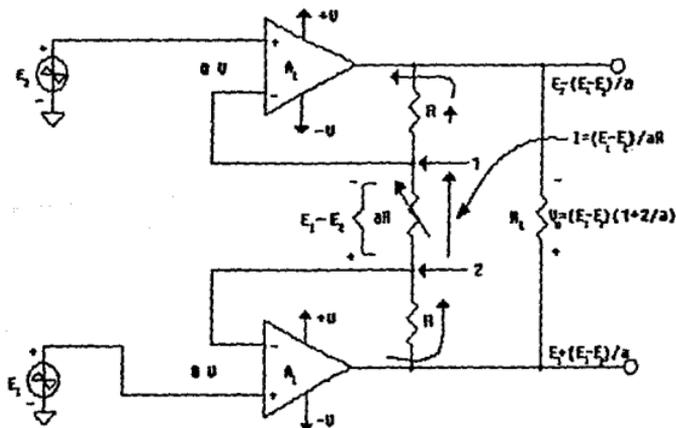


Fig.2.6 Amplificador diferencial

Ya que el voltaje diferencial de entrada de cada amplificador operacional es 0V, los voltajes en los puntos 1 y 2 (con respecto a tierra), son respectivamente iguales a E_1 y E_2 . Por lo tanto, el voltaje a través de la resistencia aR es $E_1 - E_2$, la resistencia aR es un potenciómetro que se utiliza para ajustar la ganancia. La corriente a través de aR es:

$$I = (E_1 - E_2) / aR$$

Cuando E_1 es más grande que E_2 , la dirección de la corriente I es como se muestra en la fig.2.6. La corriente I fluye a través de ambas resistencias designadas por R , y el voltaje a través de los tres resistencias, establecen el valor de V_o . En forma de ecuación:

$$V_o = (E_1 - E_2) (1 + (2/a))$$

2.3.4 Amplificador de instrumentación

El amplificador de instrumentación es uno de los amplificadores más útiles, precisos y versátiles. Está compuesto por tres amplificadores operacionales y siete resistencias como se muestra en la fig.2.7.

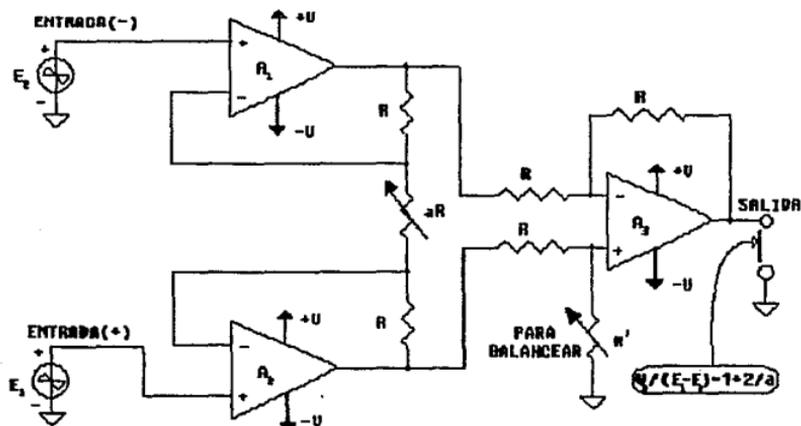


Fig.2.7 Amplificador de instrumentación

El amplificador de instrumentación en realidad se hace conectando un amplificador reforzado a un amplificador diferencial básico. El amplificador operacional A_3 y sus cuatro resistencias iguales (R) forman un amplificador diferencial con una ganancia de 1. Solo las resistencias de A_3 tienen que igualarse. La resistencia marcada con prima, R' , puede hacerse variable para eliminar cualquier voltaje en modo común. Solo una resistencia (aR), se usa para establecer la ganancia de acuerdo con:

$$V_o / (E_1 - E_2) = 1 + (2/a)$$

E_1 se aplica a la entrada (+) y E_2 a la entrada (-). V_o es proporcional a la diferencia entre los voltajes de entrada.

Las características del amplificador de instrumentación se resumen como sigue:

1. La ganancia de voltaje, desde la entrada diferencial ($E_1 - E_2$) a la salida de extremo único, se establece por una resistencia.
2. La resistencia de entrada de ambos amplificadores es muy alta y no cambia conforme varía la ganancia.
3. V_o no depende del voltaje común a ambos E_1 y E_2 (Voltaje en modo común), sólo en su diferencia.

Reuniendo lo que hasta ahora tenemos, la configuración del amplificador de instrumentación que planteamos está configurado por 3 amplificadores operacionales, con alta impedancia de entrada, alta ganancia, bajo ruido y bajo consumo de energía. Para nuestro caso la ganancia del amplificador es unitaria, la señal que se obtiene de él es amplificada por un amplificador de ganancia variable, siendo la ganancia controlada por programa, formado por un amplificador y un multiplexor analógico.

Las ocho ganancias con las que cuenta este amplificador se seleccionan mediante líneas de control (G_1 , G_2 y G_3) codificados binariamente. Las ocho ganancias del amplificador son: 0.5, 1, 2, 4, 8, 10, 100 y 500.

En la configuración del amplificador de instrumentación deben utilizarse resistencias de precisión para tener una buena RRMC (Razón de Rechazo de Modo Común), obteniéndose hasta 72 dB fig.2.8 (Desarrollo matemático apéndice 1).

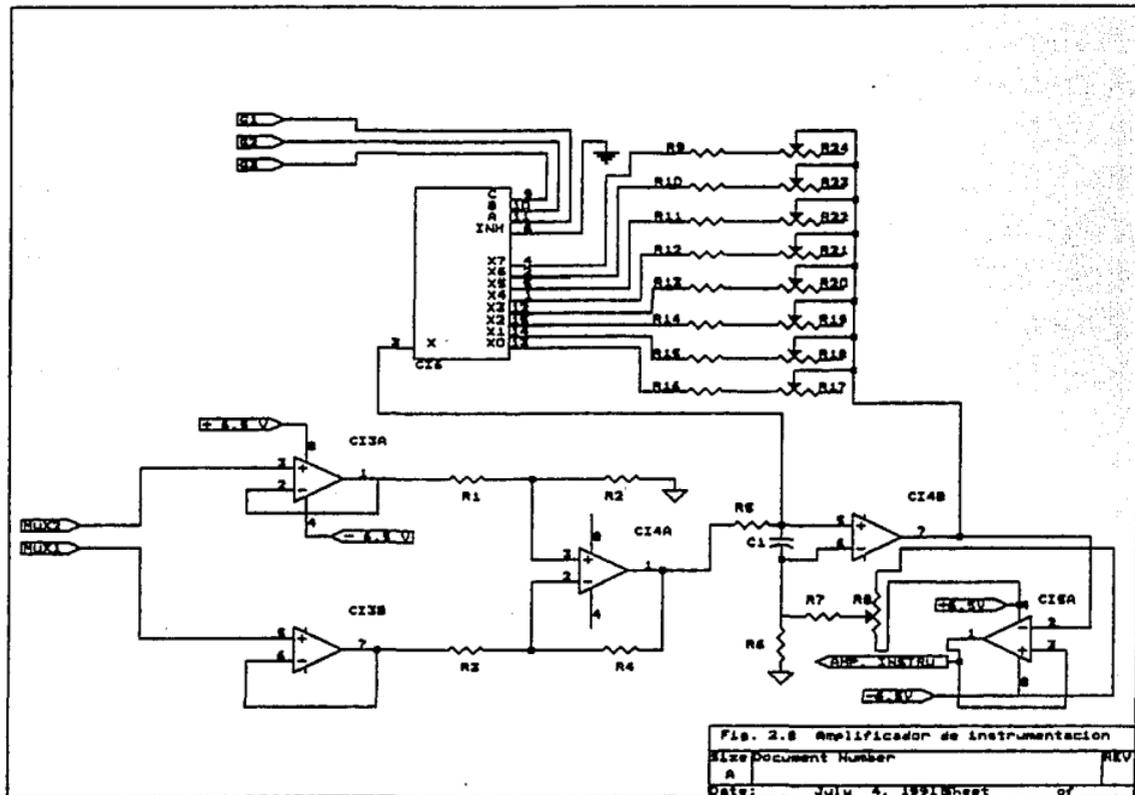


Fig. 2.8 Amplificador de instrumentación
 Size document Number
 A
 Date: July 4, 1991 sheet of

2.4 METODOS DE CONVERSION A/D

Dentro de los diferentes métodos para convertir las señales analógicas a digitales trataremos los siguientes:

1. Método de la RAMPA CONTADORA
2. Método de la RAMPA CONTADORA CONTINUA
3. Método del INTEGRADOR DE PENDIENTE DOBLE
4. Método de CONVERSION EN PARALELO
5. Método de APROXIMACIONES SUCESIVAS

2.4.1 Características importantes de los convertidores A/D

Un convertidor de señales analógicas a digitales tiene la función de traducir, o como su nombre lo dice convertir, cualquier tipo de señal analógica a digital.

Las características importantes de una conversión A/D son las siguientes:

- Tiempo de conversión: es el tiempo requerido por el convertidor para completar el proceso de conversión de una señal analógica a una digital
- Error de Modo Común: esta característica se aplica a convertidores A/D con entradas diferenciales. Este es el cambio en el código de salida que ocurre cuando el voltaje analógico en las dos entradas es de un nivel de voltaje igual, manifestándose el error en el bit menos significativo.
- No linealidad diferencial: es una medición del peor caso de desviación ideal de 1 LSB, y puede ser expresado en bits fraccionarios o como un porcentaje de la escala completa.
- Error de ganancia: es la diferencia entre el voltaje de entrada, que debe idealmente producir

a escala completa el código de salida, y el voltaje de entrada actual que produce este código.

- Dependencia de la ganancia con respecto a la temperatura: este es el error en la ganancia por el cambio de temperatura, generalmente se expresa en partes por millón por grados Celcius (ppm/°C).
- No linealidad integral (Error de linealidad): se mide para el peor caso de desviación de la línea entre los puntos extremos (cero y escala completa) y se puede expresar como un porcentaje de la escala completa o en fracciones de un LSB.
- LSB (Least Significant Bit; Bit Menos Significativo) en un sistema de código binario este es el bit de menor peso, y cuyo valor de voltaje (o corriente) se determina al dividir el voltaje total entre 2^n , donde n es la resolución del convertidor.
- Falla en el código: el incremento o decremento de voltaje a la entrada del convertidor causa que el convertidor incremente o decremente esta salida numérica por más de 1 LSB, se dice entonces que el convertidor exhibe falla en el código.
- Monotonicidad: una función monotónica tiene una pendiente donde no hay cambio de signo. Un convertidor monotónico A/D tiene una salida que cambia en la misma dirección (o se mantiene constante) por cada incremento en el código de entrada.
- Resolución: representado por el cambio analógico más pequeño y corresponde a 1 LSB del código del convertidor, expresado en bits, donde el número de niveles es igual a 2^n .

2.4.2 Rampa contadora

Este tipo de convertidor A/D utiliza como base un convertidor digital/analógico. La parte que constituye la lógica digital es un contador binario que se puede poner en cero; esto sucede cuando se le indica al contador que inicia el proceso de conversión (*START*) y automáticamente se pone en cero.

Para un mejor entendimiento de la forma de operación de este convertidor consideremos el circuito de la fig.2.9.

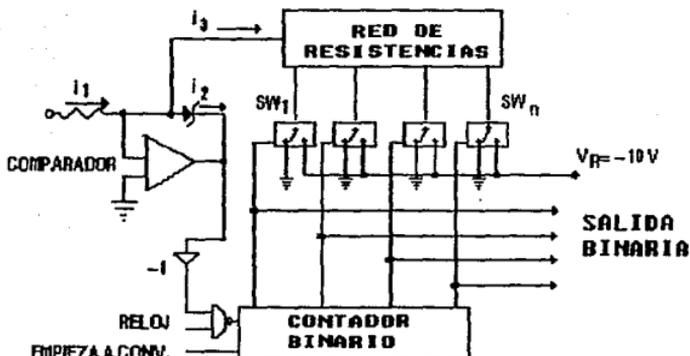


Fig.2.9 Convertidor de rampa contadora

Al principio del periodo de conversión la corriente i_1 será mayor que la corriente i_3 , haciendo positivo el valor de la corriente i_2 ; bajo estas condiciones, la corriente a la salida del comparador será baja y la compuerta *NAND* hará que los pulsos de reloj hagan avanzar el contador. La corriente i_3 aumentará en forma de escalera, hasta que i_3 sea mayor que i_1 , entonces la salida del comparador pasa a un estado alto; ahora la compuerta *NAND* inhibirá los pulsos de reloj parando el contador.

El tiempo de adquisición del convertidor de rampa dependerá básicamente de la entrada analógica y de la frecuencia del reloj.

Como podemos observar, la salida del convertidor digital/análogo cambia por el peso del bit menos significativo. Esto es que cada vez que el reloj hace la transición positiva, la rapidez de cambio del convertidor estará definida por la siguiente ecuación:

$$V_t = \frac{BMS}{1/f_c}$$

DONDE:

f_c : es la frecuencia de reloj en Hertz

BMS: (Bit Menos Significativo) dado en Volts

2.4.3 Método de la rampa contadora continua

Este es otro de los tipos de convertidores que utilizan un convertidor digital/análogo. A diferencia del convertidor de rampa contadora este incluye un contador binario ascendente-descendente.

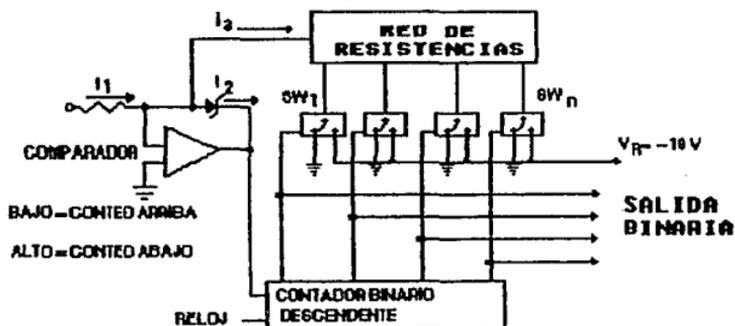


Fig.2.10 Convertidor rampa contadora continua

De la fig.2.10, podemos observar que cuando la salida del comparador se encuentra en estado BAJO (significa que $i_3 < i_1$), entonces el contador cuenta en forma ascendente y la corriente i_3 aumenta.

Cuando la salida del comparador está en estado ALTO ($i_3 > i_1$), el contador cuenta en forma descendente. Una característica que tiene este tipo de convertidor es que siempre está buscando un valor nulo.

El convertidor cambia continuamente la señal de salida siempre y cuando la señal de entrada no exceda la velocidad de cambio máxima del convertidor.

La rapidez de cambio máxima de este tipo de convertidor es la misma que la del contador de rampa.

2.4.4 Integrador de pendiente doble

Un método popular para convertir el voltaje analógico a digital es el método de doble pendiente.

El voltaje analógico que debe convertirse se aplica a través de un conmutador electrónico a un circuito integrador o generador de rampa (esencialmente una fuente de corriente constante que carga un condensador para producir un voltaje de rampa lineal). La salida digital se obtiene de un contador operado durante los intervalos negativos y positivos de la pendiente del integrador.

El método de conversión es el siguiente, fig 2.11: el voltaje de entrada analógico es conectado al integrador; dependiendo de lo que se obtenga a la salida el comparador, cambiará su estado activando la lógica de control. Dependiendo de esta entrada se activará x-tiempo el contador digital.

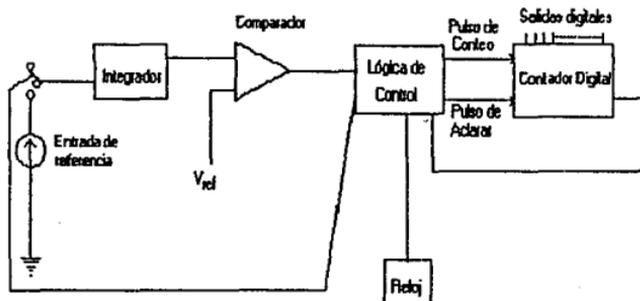


Fig.2.11 Diagrama de bloques del convertidor de doble pendiente

Al final del intervalo de cuenta fijo (fig.2.12), el voltaje del integrador es mayor para voltajes de entrada más grandes, por otro lado la cuenta se ha ajustado a cero y el conmutador electrónico conecta al integrador a una entrada de referencia o entrada fija. La salida del integrador se reduce a una tasa fija, entonces el contador avanza durante este tiempo.

La salida del integrador se reduce hasta que cae por debajo del voltaje de referencia del comparador, en cuyo instante el control lógico recibe una señal (la salida del comparador) para detener la cuenta. El valor digital

almacenado en el contador es entonces la salida digital del convertidor.

Se tiene el voltaje de entrada que dependiendo del nivel se tendrá una señal en el integrador en un intervalo de tiempo fijo. Ahora se conmuta el voltaje de referencia y empieza a integrar en forma descendente, entonces el contador empieza a contar y se detendrá cuando se cruce en cero, siendo éste el valor digital de la señal de entrada. Se conmuta de nuevo el voltaje de entrada de una nueva muestra.

Utilizando el mismo reloj y el integrador, al realizar la conversión durante los intervalos positivos y negativos de pendiente, se trata de compensar la oscilación de la frecuencia del reloj y las limitaciones de exactitud del integrador. Ajustando el valor de la entrada de referencia, la tasa de reloj se puede ajustar y se puede colocar a una escala en la salida del contador como se haya deseado.

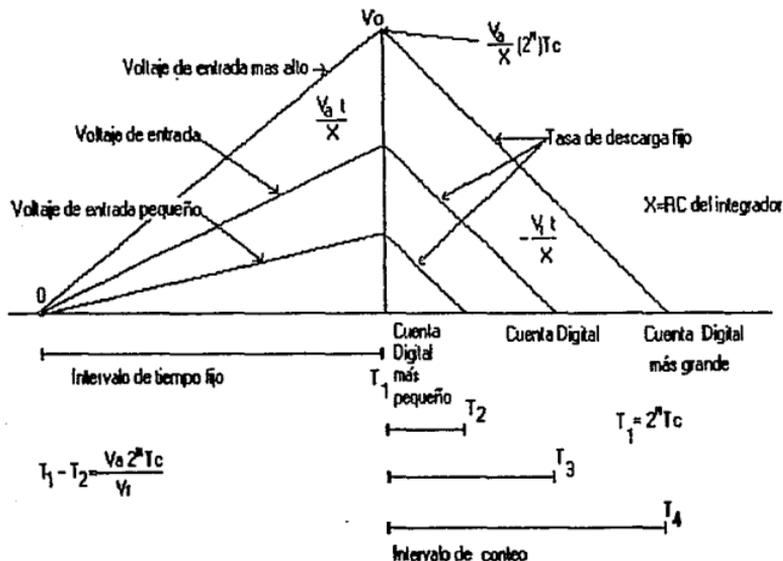


Fig.2.12 Forma de onda del convertidor de doble pendiente

2.4.5 Convertidor en Paralelo

Para poder entender el funcionamiento de este proceso veamos la fig 2.13.

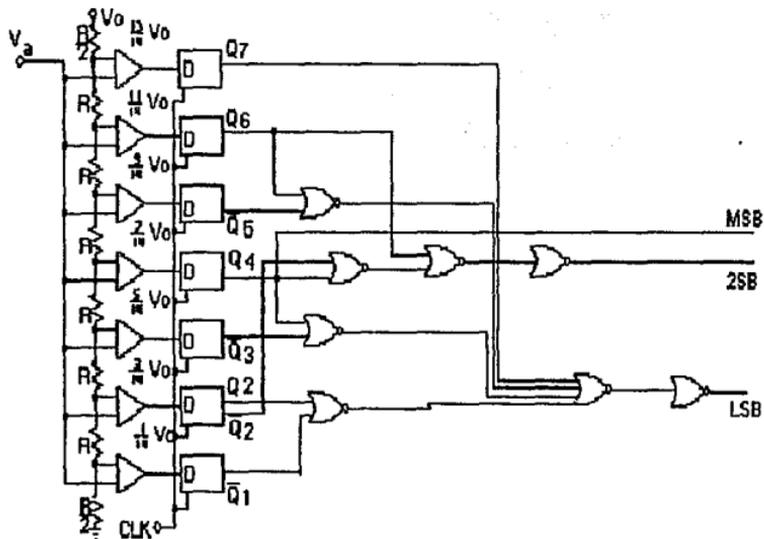


Fig.2.13 Convertidor paralelo

Se puede observar que practicamente la primera etapa son comparadores, y que dependiendo del voltaje de entrada V_a cambiará el estado de cada *flip-flop*. En la salida de cada *flip-flop* se tendrá un estado lógico definido, y por medio de un decodificador se transformarán estas señales o estados en palabras de 3 bits.

La relación entre la salida digital y la entrada analógica se muestra en la fig 2.14.

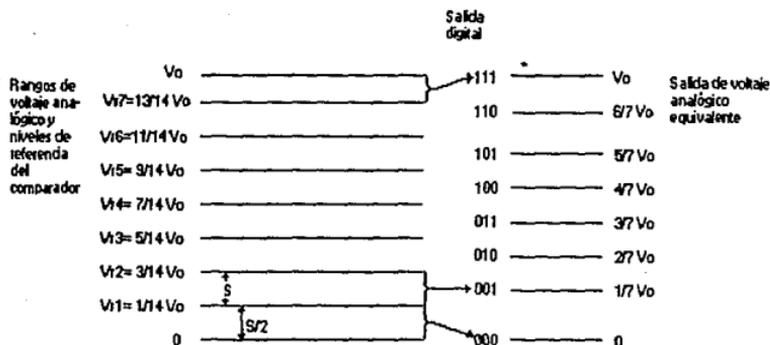


Fig.2.14. Voltaje de entrada analógico y salida digital

La entrada analógica se divide en 8 partes en intervalos de $S = V_0/7$ y los otros dos rangos o sea los extremos, en intervalos de $S/2 = V_0/14$. Si tenemos una entrada analógica en el rango bajo de 0 a $V_0/14$, a la salida del convertidor A/D se tendrá 000. Similarmente cuando la entrada está en el rango S de $V_0/14$ a $3V_0/14$ la salida digital correspondiente será 001.

Un convertidor A/D con este tipo de comparador de N-bits de salida requerirá $2^N - 1$ comparadores. Para el caso del circuito de la fig 2.13, para los 3 bits de salida de este circuito se requieren 7 comparadores.

2.4.6 Aproximaciones sucesivas

Este es otro de los convertidores que utiliza un convertidor digital/análogo. En la fig.2.15 se muestra un diagrama esquemático de dicho convertidor.

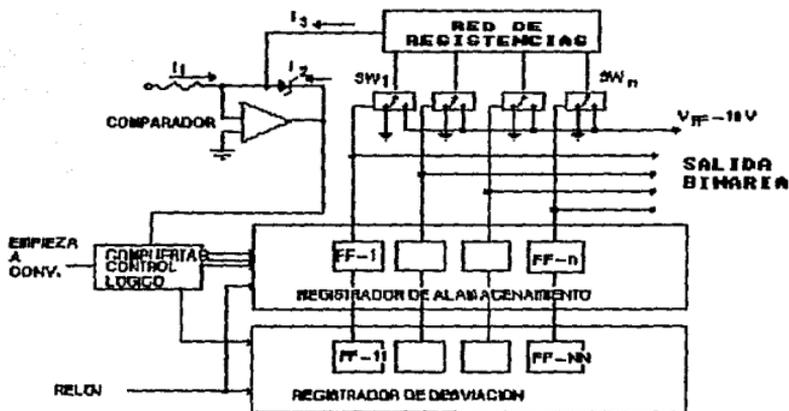


Fig.2.15 Convertidor de aproximaciones sucesivas

El convertidor A/D de aproximaciones sucesivas, de n dígitos binarios, requiere de n periodos de reloj para poder completar una conversión, para cualquier magnitud de la señal de entrada.

Dentro de las aplicaciones de carácter general, el método de aproximaciones sucesivas se considera más rápido que los métodos anteriores, siendo así también muy compleja su lógica digital.

El proceso de conversión no es continuo, como en el convertidor de rampa contadora continua. Este convertidor inicia con el pulso de reloj y termina n periodos de reloj después, cuando la salida digital es válida. Por otro lado, la señal de entrada deberá permanecer constante durante el proceso de conversión, para que la salida digital al final de este proceso pueda ser válida.

Por lo anterior, es una técnica común, hacer que preceda al convertidor A/D un circuito muestreador/retenedor.

En el inicio del proceso de conversión A/D, fig.2.15, se requiere de un pulso de conversión que pone en cero los *flip-flop's* del registrador de desviación, además de todos los *flip-flop's* que activan a los interruptores del D/A, excepto los *flip-flop's* F_1 y F_{11} que se ponen en 1 lógico.

Si el interruptor SW_1 se activa, puede hacer que la salida del comparador vaya a estado bajo si la corriente i_1 es mayor que la mitad de la escala, o en estado alto si la corriente i_1 es menor que la mitad de la escala.

Cuando el reloj se hace negativo suceden 3 cosas:

- El *flip-flop* F_1 quedará en estado alto (1) si la salida del comparador está en estado bajo, o se vuelve a cero si esta última está en estado alto.
- Si la salida del comparador es estado alto, el *flip-flop* F_2 se ajusta en 1 y desvia un 1 en el *flip-flop* F_{22} .
- A la siguiente vez que se hace negativo el reloj F_2 permanecerá en 1 o se volverá a cero. Se observa que estos estados dependen de la salida del convertidor. El proceso continúa hasta que el valor lógico 1 en F_n se quita o permanece, completándose así el proceso de conversión, de lo anterior la conversión y la salida digital tiene una posición de \pm un bit más significativo (BMS).

El método de conversión seleccionado en el presente trabajo es el de aproximaciones sucesivas, en donde se convierte una señal analógica en un número binario de 12 bits de longitud, más uno de signo. El circuito a utilizar hace la conversión con diferentes líneas de control. Una de las señales de entrada es el reloj (CLK), la cual sincroniza internamente el proceso de conversión. Para iniciar la conversión tenemos que las señales $-CS$ y $-WR$ deben ser señales activas bajas, y para leer el resultado de la conversión $-CS$ y $-RD$ deberán ser activas bajas, de esta manera el dato convertido pasa al bus local (fig 2.16).

La polarización de este circuito se obtiene del bloque de regulación de voltaje, como se observa en la fig. 2.16, el voltaje regulado es de +6.5 V y para garantizar la exactitud de las conversiones se requiere de un voltaje de

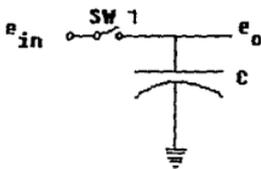
referencia de +5 V dicho circuito se muestra en la fig.2.16 que consta de, dos amplificadores operacionales, CI10A y CI10B. Los cuales proporcionan este voltaje constante que es necesario en el circuito convertidor A/D, CI9 (ADC1205).

2.5 CIRCUITOS DE MUESTREO/RETEN

Estos circuitos son los que siguen una señal de entrada, la muestrean y luego mantienen el valor instantáneo al recibir la orden de una señal de control lógico. Este tipo de circuitos se utilizan principalmente cuando la señal de entrada varía muy rápido, con respecto al tiempo que debe permanecer constante en la entrada del convertidor analógico-digital.

El circuito de muestreo/retén más simple está formado por un arreglo que incluye un interruptor (SW_1) y un capacitor (C), como se muestra en la siguiente figura (circuito básico de muestreo/retén).

CIRCUITO BASICO DE MUESTRO/RETEN



Las dos especificaciones más importantes de este tipo de circuitos son el tiempo de apertura y el tiempo de adquisición.

El tiempo de apertura es el tiempo de reacción del circuito, y está definido por el tiempo en que el control lógico ordena al interruptor (SW_1) que abra y el tiempo en que esto realmente sucede. Cuando puede tolerarse un tiempo de apertura muy largo (milisegundos), el interruptor (SW) puede ser un relevador. Utilizando un Transistor de efecto de campo *FET* como conmutador, son factibles los tiempos de apertura menores de 100 ns, fig.2.17.

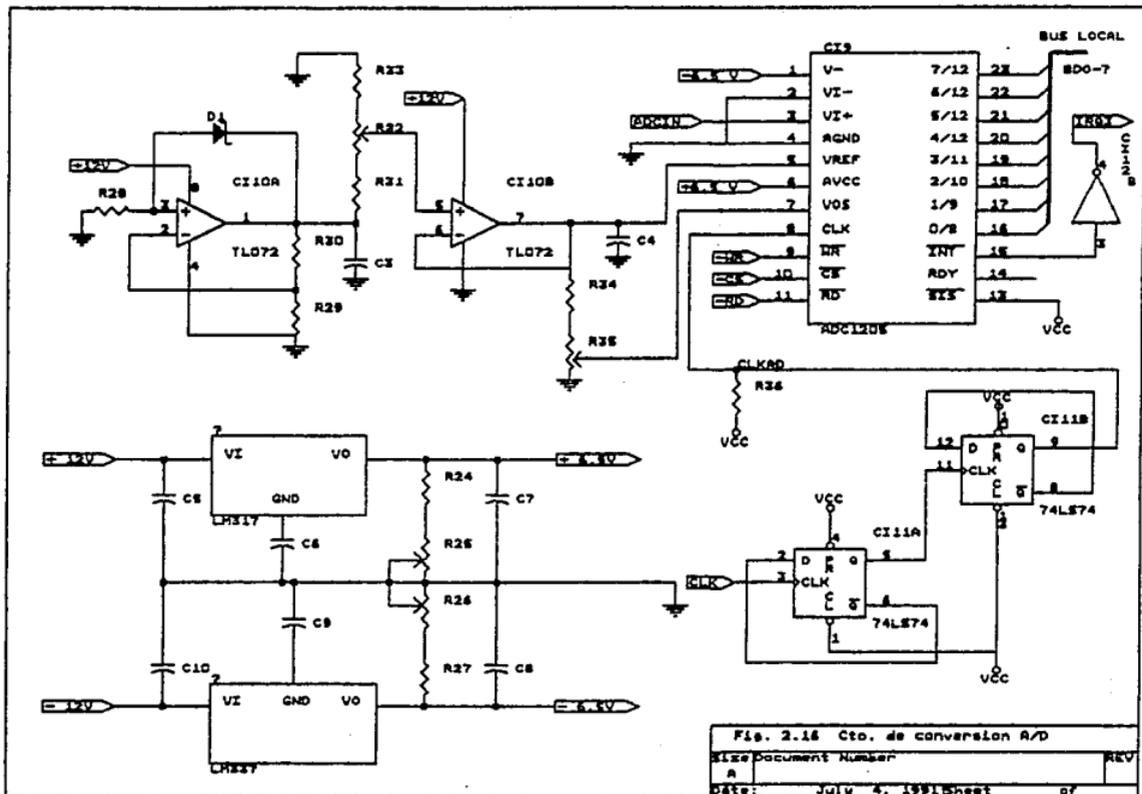


Fig. 2.16 Cto. de conversión A/D

Size	Document	Number	REV
A			
Date:			July 4, 1991 sheet 07

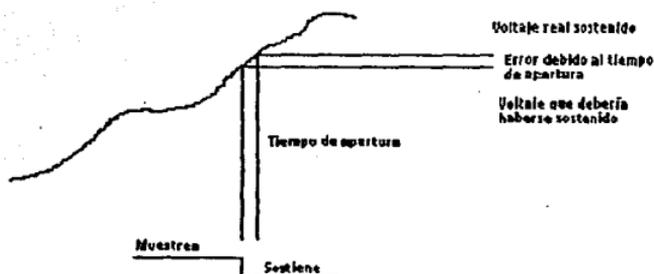


Fig.2.17 Proceso de muestreo

La fig.2.17 ilustra el error de retención producido por el tiempo de apertura.

Para el tiempo de adquisición, en los sistemas en los que la señal de entrada al circuito está variando con respecto al tiempo, el circuito muestreador/retenedor sostiene el tiempo necesario un valor para leerlo, para después adquirir un nuevo valor de la señal de entrada, que estará definida por el tiempo de conmutación MUESTREAR a RETENER.

Para el proceso de muestreo/retén contamos con el circuito de la fig.2.18, el cual está formado por un interruptor analógico y un amplificador operacional. Para el funcionamiento de esta etapa es necesario que el interruptor analógico se active por la señal de control H/S, siendo así la señal obtenida en el retenedor para enviarse al convertidor A/D, denominada ADCIN.

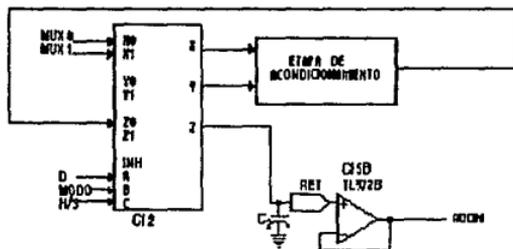


FIG.2.18. Circuito muestreador/retenedor

2.6 PUERTOS DE ENTRADA / SALIDA DIGITALES.

2.6.1 Características generales de los puertos de E/S digitales

Los puertos de entrada/salida digitales de un computador proporcionan un modo eficiente de comunicación entre el sistema central y el ambiente exterior. Los programas y los datos deben introducirse al computador para su procesamiento, y los resultados deben registrarse o exhibirse al usuario. Un computador no tiene ningún uso útil sin la habilidad para recibir información de una fuente externa y de transmitirla en forma significativa.

La interfaz de Entrada/Salida, E/S (I/O), proporciona un método para transferir información binaria entre el almacenamiento interno, tal como la memoria y los registros de la CPU, y los dispositivos de E/S. Los periféricos conectados en línea a un computador necesitan enlaces de comunicación especial para la interfaz entre ellos con el procesador central. El propósito del enlace de comunicación es resolver las diferencias que existen entre el computador central y cada uno de los periféricos.

Las principales diferencias son:

- a) La velocidad de transferencia de datos de los periféricos es mucho más lenta que la velocidad de transferencia del computador central.
- b) La operación de los periféricos debe sincronizarse con la operación de la CPU y la memoria.
- c) Los formatos de datos en los periféricos difieren del formato de palabra en la CPU.
- d) La operación de los periféricos debe controlarse de tal manera que no perturbe la operación de la CPU.

Para resolver estas diferencias, los sistemas de computador incluyen componentes especiales de *hardware*, entre la CPU y los periféricos, para supervisar todas las transferencias de E/S.

La CPU tiene el control de estos dispositivos de E/S, que supervisa el flujo de información en el canal de E/S. El control de E/S del computador recibe instrucciones de la memoria programada y procede a ejecutarlos, comunicándose con los periféricos a través de su interfaz.

Para comunicarse con un dispositivo en particular, la CPU coloca una dirección de dispositivo en un conjunto de líneas en el canal. Cada una de las interfaces asignadas al canal común contiene un decodificador de dirección que monitorea continuamente las líneas de dirección. Cuando la interfaz detecta su código de dirección, activa las rutas entre las líneas del canal de E/S y los periféricos asignados a él.

Al mismo tiempo que la dirección queda disponible en las líneas de dirección, el computador proporciona un código de función en otro conjunto de líneas en el canal. La interfaz seleccionada decodifica el código de función y procede a ejecutarlo. El código de función es denominado a menudo un comando y es en esencia una instrucción que se ejecuta en el módulo de interfaz.

Un ejemplo de como el computador genera los comandos para el periférico, es el siguiente formato de instrucción de E/S (fig.2.19).

CÓDIGO DE OPERACION	DIRECCION DEL DISPOSITIVO	CÓDIGO DE FUNCION
---------------------	---------------------------	-------------------

Fig.2.19 Formato de instrucciones E/S

La instrucción es leída de la memoria durante el ciclo de búsqueda (**FETCH**) y colocada en un registro, la instrucción contiene tres campos, como se indica en la Fig.2.19:

- a) Campo del código de operación, distingue la instrucción como de E/S.
- b) Dirección del dispositivo, nos proporciona en código binario la dirección en la cual se encontrará el dispositivo.
- c) El código de función especifica el comando para la interfaz de E/S.

La conexión del canal E/S al computador es como sigue:

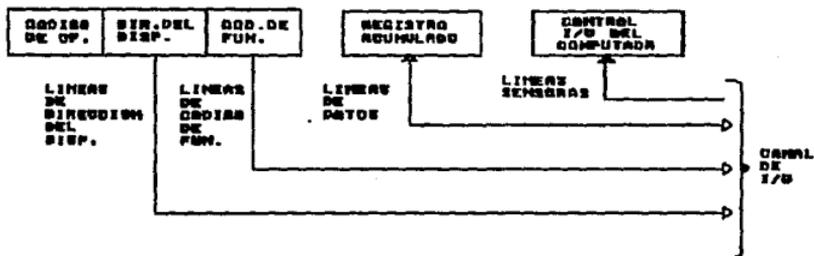


Fig.2.20 Diagrama de conexiones del canal de E/S al computador

Como se muestra en la Fig.2.20, una vez identificada la operación como de E/S (de acuerdo al código de operación), el canal reconocerá la dirección del dispositivo al cual se le requiere para la operación, una vez que el control ha recibido la indicación comenzará el flujo de información.

La conexión del canal de E/S a una interfaz se muestra en la Fig.2.21.

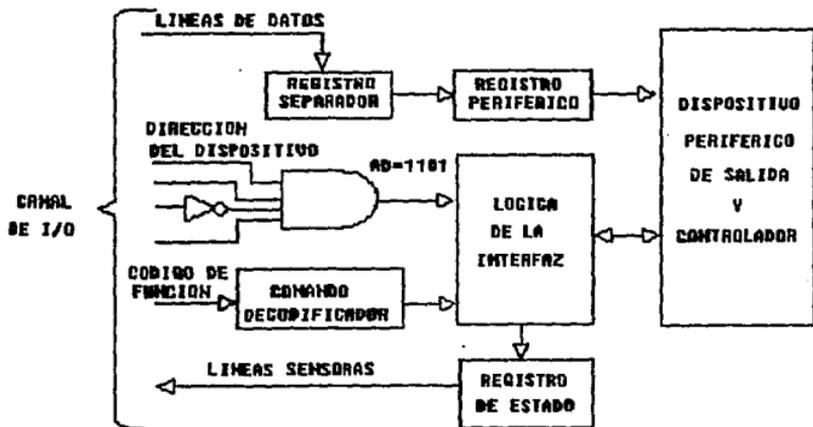


Fig.2.21 Diagrama de codificación de la señal

Los bits de dirección son decodificados por medio de una compuerta AND y la salida de la compuerta habilita la sección de control de la interfaz.

El código de función es entonces aplicado al control de interfaz para activar el comando especificado. La interfaz tiene un registro separador para retener los datos que están siendo transferidos a través del canal de E/S. Un segundo registro se utiliza para transferir datos al dispositivo periférico de salida. El registro de estado almacena las banderas que notifican al computador de las condiciones importantes en la interfaz. Estas condiciones comprenden los datos que han sido recibidos del periférico, la condición de que la interfaz esté lista para recibir datos del computador, así como también indicaciones de error tales como error de paridad, o que la potencia del dispositivo ha fallado.

A una interfaz se le asigna la tarea de sincronizar el flujo de datos entre el computador y los periféricos. Para la sincronización de la entrada de datos, la interfaz recibe un bloque de datos a la velocidad que puede proporcionar el periférico y la transfiere al computador a través del canal de E/S cuando está disponible. Para la sincronización de salida, la interfaz recibe un bloque de datos del canal de E/S y lo retiene en su registro separador. Posteriormente es transferido al periférico a una velocidad que el pueda aceptar.

TIPOS DE COMANDOS :

El computador emite comandos a través de los bits del código de función. La interpretación del comando depende del periférico que se está direccionando. Hay cuatro tipos de comandos que puede recibir la interfaz cuando es direccionada por la CPU, y se clasifican como comandos de control, de verificación, y de entrada/salida de datos.

Un comando de control es emitido para activar el periférico y para informarle que hacer.

Un comando de verificación se utiliza para verificar las diversas condiciones de estado en la interfaz o el periférico.

Un comando de salida de datos hace que la interfaz responda tomando un bloque de datos de las líneas de datos en el canal de E/S.

En un comando de entrada de datos la interfaz recibe un bloque de datos del periférico y lo coloca en su registro separador.

2.6.2 Métodos de transferencia

Las operaciones internas en un sistema digital son sincronizadas por medio de pulsos de reloj suministrados por un generador común de pulsos. Los pulsos de reloj son aplicados a todos los registros dentro de una unidad y todas las transferencias de datos en los registros internos ocurren simultáneamente durante la presencia de un pulso de reloj. Dos unidades, tales como una CPU y una interfaz de E/S, son diseñadas independientemente la una de la otra. Si los registros en la interfaz comparten un reloj común con los registros de la CPU entonces la transferencia entre las dos unidades se dice que es síncrona. En la mayoría de los casos, la sincronización interna en cada unidad es independiente de la otra, cada una utiliza sus pulsos de reloj privados para los registros internos.

En cada caso se dice que las dos unidades son asíncronas la una de la otra. Este enfoque es utilizado en la mayoría de los sistemas.

La transferencia asíncrona de datos requiere que las señales de control sean transmitidas entre las unidades que se están comunicando, para indicar el tiempo en el cual los datos están siendo transmitidos. Una manera de lograr esto es por medio de un pulso suministrado por una de las unidades, para indicar a la otra cuando va a ocurrir la transferencia de datos. Otro método comúnmente utilizado es acompañar cada uno de los bloques de datos que se están transfiriendo con una señal de control, que indica la presencia de datos en el canal. La unidad que recibe el bloque responde con otra señal de control para reconocer el recibo de datos.

La transferencia de datos entre dos unidades se puede hacer en paralelo o en serie. En la transmisión de datos en paralelo cada bit del mensaje tiene su propia ruta y el mensaje total es transmitido al mismo tiempo. Esto significa que un mensaje de n bits debe ser transmitido a través de n rutas conductoras separadas. En la transmisión de datos en serie, cada uno de los bits en el mensaje es enviado en secuencia uno a uno. Este método requiere el uso de un par de conductores o un conductor y una tierra común. La transmisión paralela es más rápida pero requiere de varios cables. Es utilizada para distancias cortas y donde la velocidad es importante. La transmisión en serie es más lenta pero es menos costosa y de mayor distancia de comunicación.

De acuerdo con las características de diseño de la tarjeta de funciones múltiples, es necesario una interfaz paralelo que comunique la tarjeta con la computadora, ya que

necesitamos un medio a través del cual podamos enviar y recibir información de la computadora de la manera más rápida posible, adicionalmente la tarjeta se encontrará en el interior de la computadora, por lo que la distancia es mínima, y cumple con todas las características del puerto paralelo.

2.6.3 Puerto paralelo 8255

El controlador programable paralelo de interfaz 8255 (PPIC) sirve de ayuda en la conexión a la computadora de dispositivos que envían octetos completos de información, cada vez que hay una transferencia de datos (Fig. 2.22).

La transmisión paralelo es útil en todas aquellas aplicaciones que requieran transmisiones a gran velocidad, y utilicen dispositivos no demasiado alejados del computador central. No hay ninguna sincronización especial en las transmisiones paralelo. Los octetos se envían tan rápidamente, o tan lentamente, como el *software* lo permita. Si la transmisión debe ser lenta, se introducen retardos por *software*. La velocidad máxima de transmisión está limitada por la rapidez con que el sistema saca los datos. La mayor velocidad se obtiene utilizando el método de acceso directo a memoria (DMA).

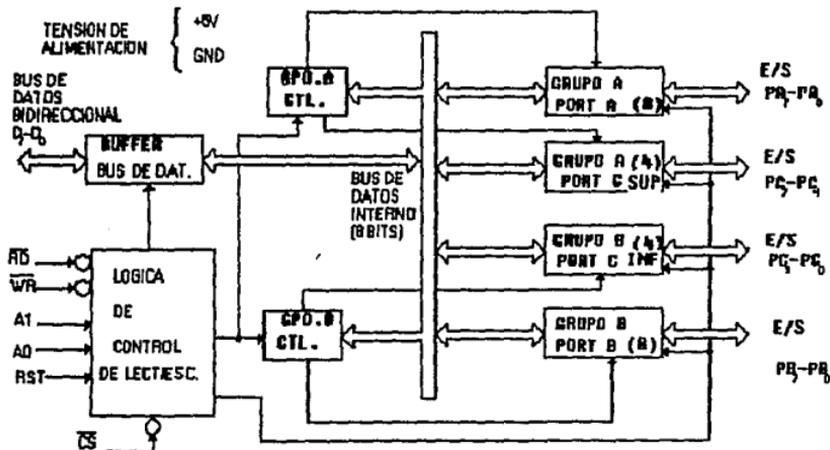


Fig.2.22 Diagrama de bloques del circuito integrado 8255

El circuito integrado (CI) 8255 es de 40 terminales, tiene 24 líneas de entrada o de salida, las cuales el usuario puede configurar o programar. Hay tres modos de transmisión básicos :

- a) El modo 0, entrada / salida básica.
- b) El modo 1, entrada / salida habilitada.
- c) El modo 2, canal bidireccional.

Las 24 líneas se agrupan en dos grupos de 8 bits y un grupo de dos grupos de cuatro. En el modo 0 (cero) hay 16 maneras de programar las líneas de Entrada/Salida de los cuatro grupos, los datos se envían o se reciben por ellas (ver fig.2.22). En el modo 1, el grupo formado por dos grupos de 4 bits sirven de control y estado, y los grupos de 8 bits se utilizan para transferencia de datos. Cada grupo de 8 bits se puede definir como de E/S. En este modo el usuario ve un cierto bit del octeto de estado para saber si el dispositivo externo está preparado. Si lo está, el octeto de datos se puede transmitir.

En el modo 2 se provee un método de comunicación con un dispositivo periférico en un solo bus de 8 bits para transmitir y recibir datos (bus bidireccional de Entrada/Salida).

Las señales de "*Handshaking*" son asignadas para mantener un control apropiado en el flujo de datos, en forma similar al del modo 1.

La generación de interrupciones y funciones de habilitación/deshabilitación están también disponibles. Para la recepción y transmisión de datos se efectúa a través del grupo A, en el grupo C, se manejan señales de control y el grupo B, no es usado (fig.2.22).

Las órdenes se transmiten al 8255 vía un puerto de E/S especial. Estas órdenes afectan a las características tales como el agrupamiento de la dirección de los puertos, y las asignaciones de control y estado (fig.2.23).

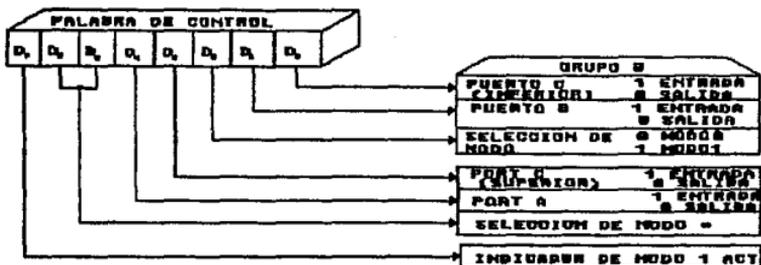


Fig.2.23 Diagrama de codificación de la palabra de control

En la interfaz los puertos A y B se han configurado como salidas, el puerto C como entrada, a fin de proteger el circuito del puerto se deben colocar *buffers* inversores tanto a la salida como a la entrada del mismo.

En la fig.2.24, se muestra el circuito electrónico utilizado para el puerto de entrada/salida digitales, donde se observa el circuito CI26 (de nomenclatura 8255), y el acoplamiento del puerto por medio de *buffers* inversores, CI28, CI29, CI30, CI31, CI32, (todos ellos de nomenclatura 74LS14), también podemos ver al circuito CI27 (de nomenclatura 8253), que es un contador *limer* programable diseñado únicamente para su uso en sistemas microcomputadores. En esta misma figura hemos puesto el diagrama de conexión de las terminales de la ranura de la PC, donde se insertará la tarjeta de funciones múltiples.

SEÑALES DEL CIRCUITO CONTADOR TIMER (8253)

D7-D0	Bus de datos.
CS	Habilitación del Chip.
RD	Lectura al puerto.
WR	Escritura al puerto.
A0,A1	Líneas de control.
CLK0,CLK1,CLK2	Reloj para cada contador.
OUT0,OUT1,OUT2	Salida del contador.

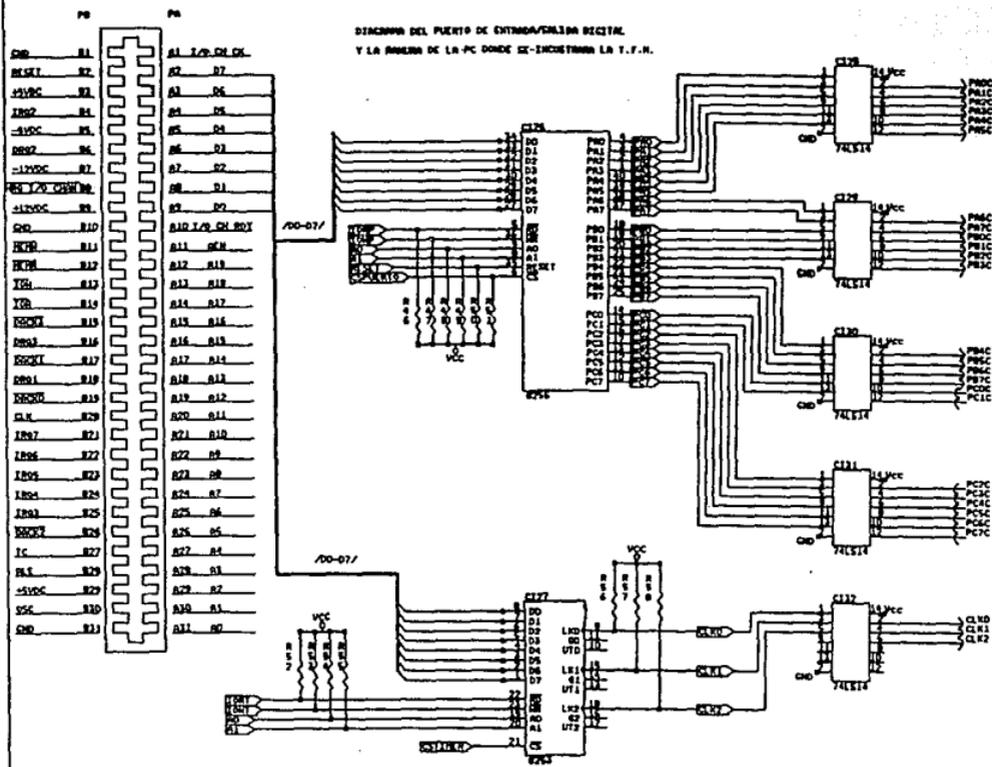


Fig. 2.24 Cto. del puerto paralelo para
microcomputadora digital
Número
8
Evol. September 4, 1981

DESCRIPCION FUNCIONAL DEL CIRCUITO 8253

Este circuito es activado por medio de un programa de aplicación específica para esta interfaz, primero programamos el registro de control que está constituido por las líneas AO, A1 y el *bus* de datos, estas líneas establecen el modo de operación del contador, cada modo está programado independientemente, permitiendo el funcionamiento individual de cada contador. Con la señal \overline{WR} y el *bus* de datos se programa la palabra de control, con RD y el *bus* de datos permite la lectura para cada contador, la cual se puede realizar cuando el usuario lo necesite.

2.7 CONVERTIDOR DIGITAL / ANALOGICO

2.7.1 Características generales

Un convertidor de señales digitales a analógicas (DAC) acepta una palabra digital como entrada y lo convierte a un voltaje analógico. Los convertidores trabajan con códigos digitales monopolares o bipolares. El monopolar incluye el binario puro y el decimal codificado en binario (BCD). El binario desplazado, Código Gray, complemento a uno y complemento a dos, son para la operación bipolar.

El valor de la salida analógica de un convertidor digital-analógico, de n dígitos binarios, se obtiene mediante la expresión:

$$E_0 = V_r (a_1 2^{-1} + a_2 2^{-2} + a_3 2^{-3} + \dots + a_n 2^{-n})$$

En donde V_r es un voltaje de referencia analógico y los coeficientes de a_1 a a_n son iguales un uno ó un cero lógico, dependiendo del estado de los dígitos de entrada.

El peso del dígito binario más significativo es $V_r/2$ y el peso del dígito binario menos significativo es $V_r/2^n$. Cuando todos los dígitos binarios están a 1 lógico, la salida analógica será igual a $V_r(1-2^{-n})$. La adición de todos los bits produce un valor que se aproxima a 1. La diferencia

algebraica, entre el valor binario que se aproxima a 1 y el valor 1, es el error de cuantización del sistema digital.

2.7.2 Métodos de conversión

La operación de conversión de valores digitales monopolares en valores analógicos equivalentes es realizada por dos técnicas de conversión: Resistivo y Escalera R/2R.

CONVERTIDOR RESISTIVO

Es el convertidor digital analógico más simple y consiste de resistencia de valores precisos proporcionales, siguiendo potencias de 2, y un amplificador operacional de muy alta ganancia, alta impedancia de entrada y baja impedancia de salida.

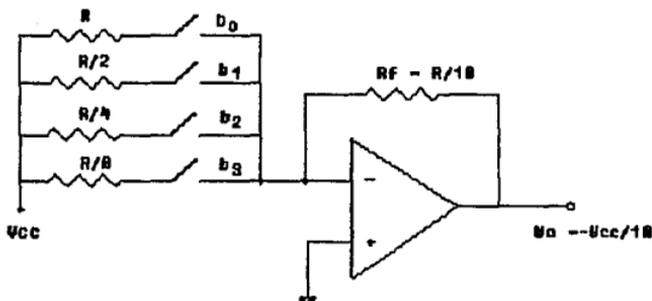


Fig.2.25 Circuito básico

Considerando en el circuito básico de la fig.2.25, el interruptor b_0 cerrado (y la resistencia $R = 10$ ohms), ocurre que por efecto de la tierra virtual que aparece a la entrada negativa del amplificador operacional, se genera una corriente igual a $I = (V_{CC}/R)$, que circula pasando por R y R_f y generando en la salida un voltaje de $V_o = R_f(-I) = R_f(-V_{CC}/R)$ y por lo tanto $V_o = -V_{CC}/10$.

Si consideramos ahora el interruptor b_1 cerrado la corriente que circula por la resistencia $R/2$ será igual a

$2V_{CC}/R$. El interruptor b_2 cerrado contribuye con $4V_{CC}/R$ y el interruptor b_3 cerrado contribuye $8V_{CC}/R$.

Los valores de las resistencias del circuito producen corrientes que tienen los mismos pesos que los dígitos del sistema binario de numeración. Si más de un interruptor se cierra, las corrientes de los interruptores se suman con el punto de tierra virtual y circulan por R_f . Así, el voltaje a la salida del circuito será proporcional a la palabra digital programada en los interruptores b_3, b_2, b_1, b_0 .

La desventaja principal del convertidor anterior radica en el rango de valores de resistencia que se necesita para un convertidor de un número grande de bits.

CONVERTIDOR DE ESCALERA R/2R

Este convertidor opera básicamente igual al convertidor resistivo, pero evita el rango amplio de valores de resistencias, completando solamente 2 valores (fig.2.26).

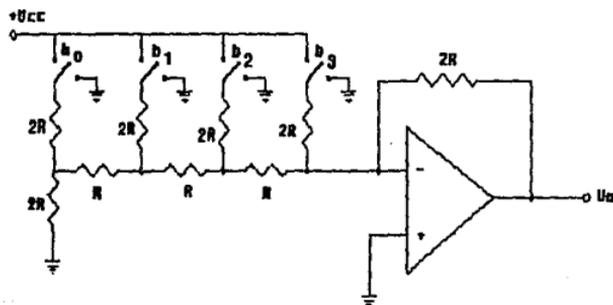


Fig.2.26 Convertidor de escalera R/2R

Considerando el interruptor b_3 conectado a V_{CC} , en tanto que los interruptores b_2, b_1 y b_0 conectados a tierra, resulta el siguiente circuito equivalente (fig.2.27).

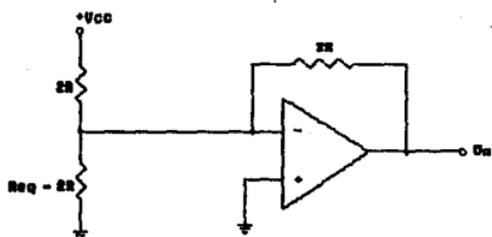


FIG.2.27 Circuito equivalente

Si consideramos la "Tierra Virtual" a la entrada del amplificador podemos eliminar Req. De aquí resulta el valor de la corriente.

$$I = V_{cc}/2R \text{ y } V_o = -V_{cc}$$

Para encontrar la contribución al voltaje de salida para el interruptor b₂ se sigue un procedimiento análogo.

Considerando el interruptor b₂ conectado a V_{cc} y los interruptores b₃, b₁ y b₀ conectados a tierra, resulta el circuito equivalente de la fig. 2.28.

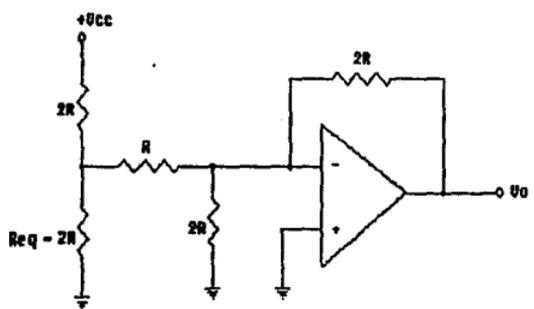


Fig.2.28 Primer circuito equivalente

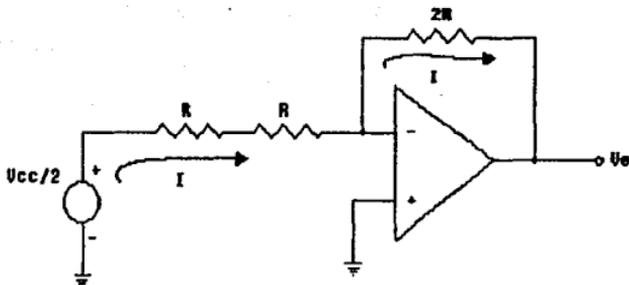


Fig.2.29 Segundo circuito equivalente

Simplificando el circuito anterior se obtiene el circuito de la fig.2.29, el cual resulta de eliminar la resistencia $2R$ conectada entre la terminal negativa del amplificador operacional (tierra virtual) y tierra, y de obtener el equivalente de Thévenin del circuito de la izquierda de la resistencia R . De aquí es inmediato el cálculo $I = V_{CC}/4R$ y $V_O = V_{CC}/2$

Con un análisis similar se puede demostrar que el cierre del interruptor b_1 genera un voltaje de salida $V_O = -V_{CC}/4$. Si más de un interruptor se cierra, sus contribuciones individuales al voltaje de salida se suman. Con este convertidor sólo se necesita controlar dos valores de resistencias. El número de bits puede ser incrementado agregando más secciones a la escalera con los mismos valores de R y $2R$.

OPERACION DE CONVERTIDORES BIPOLARES D/A.

Estos convertidores requieren tanto voltajes de referencia positivos como negativos. Para representar números negativos se elige la polaridad de voltaje de referencia de acuerdo con el dígito binario que representa el signo en la palabra digital. Sin embargo, este procedimiento no es el más adecuado, ya que el interruptor que elige al bit de referencia debe tener una impedancia muy baja.

Un método comunmente usado produce la palabra binaria suministrando una corriente compensadora constante a la entrada del amplificador operativo. Esta corriente se ajusta para que sea igual a la corriente del bit más significativo. El que se conecte o no el bit más significativo determina el signo de la salida. Por lo tanto se llama dígito binario signo.

Como las entradas digitales a un convertidor D/A vienen de un registro de almacenamiento, el dígito binario signo se conecta a la salida inversora.

2.7.3 Parámetros importantes de los convertidores D/A y A/D

MONOTONICIDAD

Un convertidor D/A es monotónico si su salida se incrementa o permanece constante cuando la entrada se incrementa o permanece constante, para todo el rango de escala (fig 2.30).

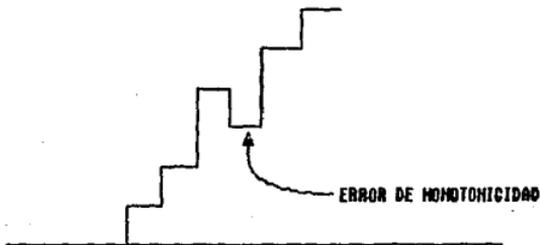
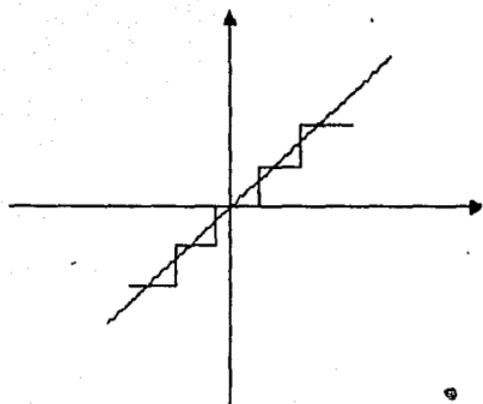


Fig.2.30 Error de monotonicidad

ERROR DE CUANTIZACION (VALOR MAXIMO)

Este error se refiere a la diferencia punto a punto entre el valor de una señal analógica dada, y su correspondiente señal analógica discretizada en amplitud (Cuantificada) (fig.2.31).



EL VALOR MAXIMO DEL ERROR DE CUANTIZACION EQ ESTA DADO POR

$$EQ = (1/2) \left(\frac{\text{RANGO DE LA ESC. COMPLETA}}{2^n} \right)$$

n = NUM. DE BITS DEL CONVERTIDOR

$$EQ = (1/2) \text{ LBS. } \delta E_a = (1/2) \text{ BIT}$$

Fig.2.31 Error de cuantización

PRECISION ABSOLUTA

Para un convertidor D/A se define por:

$$\frac{\text{Rango completo de escala real}}{\text{Rango completo de escala nominal}} \times 100$$

para un convertidor A/D este parámetro no se puede establecer cuantitativamente, depende del error de cuantización y de las imperfecciones de los componentes del circuito.

NO LINEALIDAD DIFERENCIAL

Se refiere a la variación del tamaño del "cuanto" (q) entre dos números digitales consecutivos, sobre el rango completo de valores de la entrada y la salida. La no linealidad diferencial mayor de q, pero positiva, da lugar a la no linealidad monótona.

La no linealidad diferencial especifica este parámetro como:

$$\pm \frac{LSB}{2}$$

LINEALIDAD

Para un convertidor digital/análogo este parámetro significa la desviación del voltaje de salida del DAC con respecto a una línea recta trazada del valor 0 al valor de escala completa de salida (fig. 2.32).

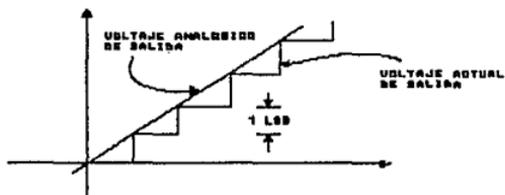


Fig.2.32 Linealidad - Precisión relativa

Para un convertidor A/D la linealidad se conoce con el nombre de precisión relativa, y se define por la desviación del código digital de salida ADC con respecto a una línea recta trazada del valor 0 al valor de escala completa de entrada.

2.6 CONVERSION DIGITAL/ANALOGICA EN EL DISEÑO

En el diseño de conversión D/A contamos con dos salidas analógicas, se constituye esta etapa en base al diagrama de bloques que se representa en la fig.2.33.

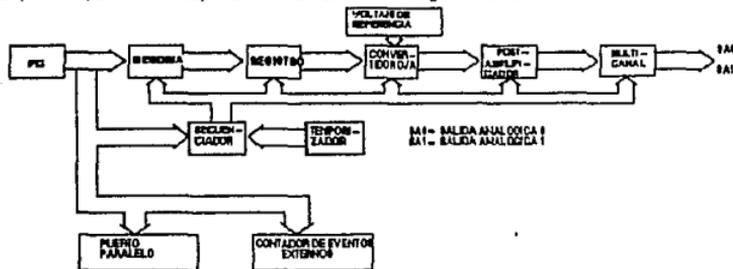


Fig.2.33 Diagrama de bloques etapa de conversión D/A

Las señales analógicas de salida permiten una variación de voltaje entre -5.000 V y $+4.997\text{ V}$ el cual puede variar en incrementos de 25 mV . Los voltajes se actualizan por medio de comandos provenientes del programa que opera la tarjeta, manteniéndose así un intercambio adecuado de señales como son: datos, direcciones y control.

Para facilitar el entendimiento del diagrama funcional de la fig.2.33, abordaremos principalmente lo que son bloques de memoria, registro, convertidor D/A, pos-amplificador y multicanal. Posteriormente nos referiremos a los bloques del secuenciador, temporizador y contador de eventos externos, puesto que determinan el control de la operación de esta sección.

Bloque decodificación de direcciones

Este es una parte común a las secciones de conversión A/D y D/A. El bloque de decodificación de direcciones permite seleccionar a aquel dispositivo en el cual se quiere realizar alguna actividad, por ejemplo, una lectura o una escritura. La selección se hace por medio de una dirección presentada por el programa, la cual corresponde a uno de los dispositivos de la tarjeta. Como podemos ver, este bloque se encuentra también en la parte de conversión A/D.

Los bloques que específicamente integran la conversión D/A son:

Bloque de memoria

Se tiene un registro de memoria para cada canal analógico, cada registro almacena el valor binario del voltaje de salida requerido.

La lectura/escritura en la memoria se realiza por medio de un programa que permite al usuario leer o escribir en cada registro. Tal proceso se lleva a cabo con una instrucción de lectura o escritura a un puerto de entrada/salida.

Después de que el usuario haya indicado los cambios en cada uno de los canales analógicos de salida, las memorias tendrán cierta información, la cual mantendrán hasta que se realice un nuevo cambio.

Bloque de conversión D/A

Se realiza la conversión de un dato digital a un voltaje analógico. La conversión es realizada por medio de un convertidor D/A de 12 bits. Los datos son tomados de la

memoria por un secuenciador y almacenados en el registro del convertidor para su posterior conversión.

Para acoplar y amplificar el voltaje que se encuentra a la salida del convertidor se tiene una etapa de conversión de corriente a voltaje y una etapa de amplificación de voltaje.

Bloque Multicanal

Esta parte realiza la selección del canal analógico de salida que va a ser actualizado. El circuito multicanal se activa por medio de las señales provenientes del secuenciador.

Una vez que se ha seleccionado un canal analógico, la señal de voltaje proveniente del convertidor se presenta a un circuito reten de orden cero. Este retenedor realiza la función de mantener estable el voltaje analógico de salida. Se tiene un circuito de retención para cada uno de los canales analógicos de salida.

Los circuitos correspondientes a estos bloques se describen a continuación:

Circuitos de lectura y escritura en memoria local

En esta parte de la interfaz se realiza el intercambio de información entre la PC y la tarjeta, es decir permite la lectura o escritura en la memoria local. En la fig.2.34 se muestra un diagrama de bloques de la memoria local.

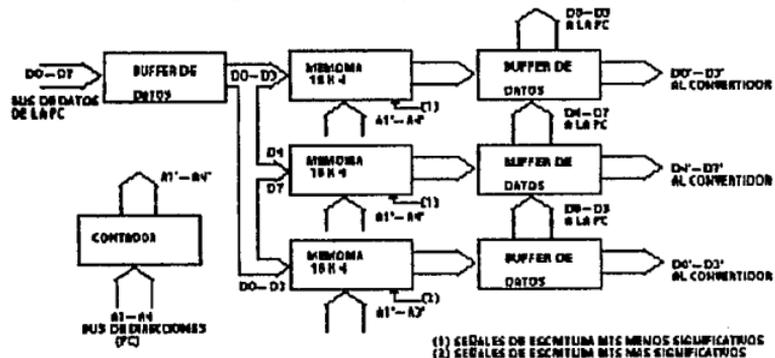


Fig.2.34 Diagrama general para una escritura o lectura en la memoria local

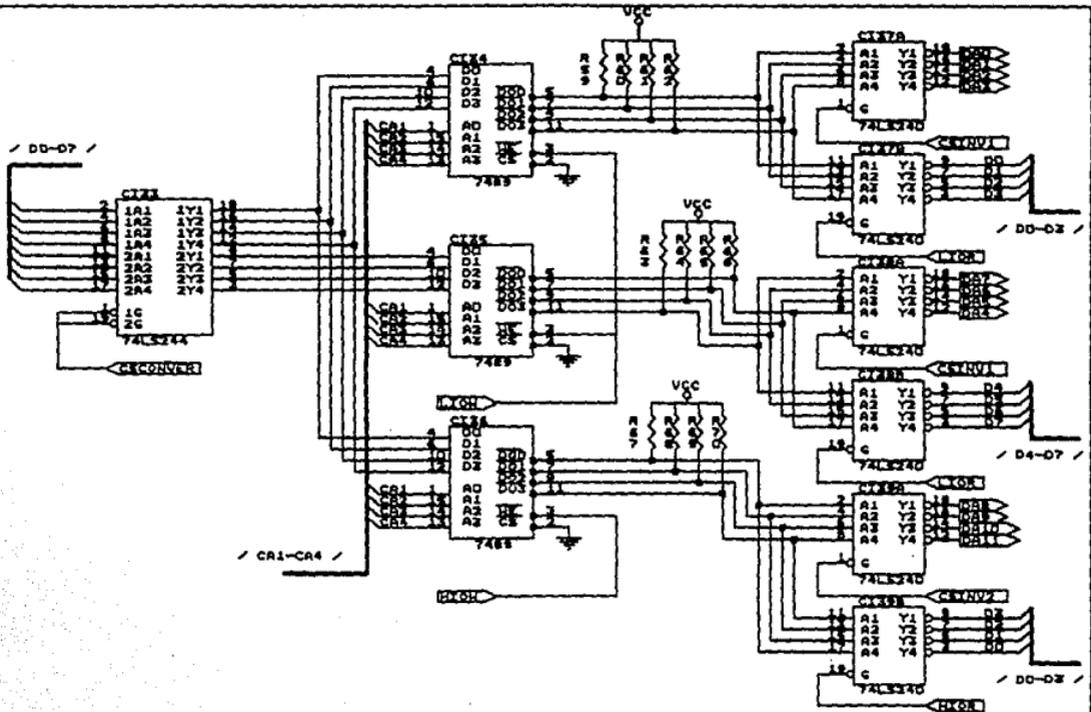


Fig. 2.35 Cto. del arresio de memorias para la conversión D/A
 Size Document Number REV
 A
 Date: JULY 4, 1994 sheet of

El diagrama eléctrico del arreglo de la memoria junto con los *buffer* del *bus* de datos se encuentra en la Fig.2.35. Esta constituido por las memorias RAM CI34, CI35, CI36, cada una de ellas contiene 16 registros de 4 bits. Las memorias son circuitos 7489, RAM, tiene 4 líneas de datos (D0-D3), cuatro líneas de dirección (A0-A3), cuatro líneas de salida de datos (D00-D03).

Las salidas son de lógica negativa con colector abierto, por lo que se colocan resistencias *pull-up* para cada una de las salidas (R59-R70).

Las memorias se controlan con líneas -CS y -WE, las memorias siempre se encuentran en "cero" lógico y la señal -WR se encuentra en estado alto excepto cuando se realiza una lectura en la memoria.

La señal -WR (*write/read*) activa la escritura en la memoria, y es activa baja. La lectura se realiza cuando -WR está en "uno" lógico. Las memorias se conectan en serie para formar palabras de 12 bits de longitud.

Para escribir en la interfaz se emplean las señales LIOW y HLOW, generados en el circuito de sincronía para la conversión A/D. La señal LIOW que permite escribir los bits menos significativos, esta señal activa la señal de escritura -WR en las memorias M3 y M8. Después se presenta la señal HLOW que activa -WR en la memoria -M13 cargando los bits más significativos.

Para realizar una lectura se generan las señales LIOR y HIOR, que permiten obtener información de las memorias; activando la señal de control "G", activa el tercer estado de los circuitos CI37A, CI37B y CI33 (74LS244).

LIOR activa la señal de habilitación "G" en los circuitos CI37B y CI38B, de donde se obtiene los primeros 8 bits menos significativos, y la señal HIOR activa la señal de habilitación "G" del circuito CI39B para obtener los 4 bits más significativos. Las salidas de los circuitos CI37B, CI38B y CI39B (D0-D7) van directamente al bus de la computadora. Cuando la señal de la habilitación "G" no es seleccionada en los circuitos CI37A, CI37B y CI39B sus salidas se encuentran en alta impedancia.

Cuando no se realiza una lectura o escritura en los registros de los canales analógicos, el circuito secuenciador realiza la lectura de cada uno de los registros de la memoria de la interfaz, para realizar la conversión de este dato y obtener un voltaje analógico de salida. De esta forma se leen los 2 registros y se repite la secuencia.

Las figs. 2.45 y 2.46 muestran los diagramas de tiempo especificando en que momento se puede realizar una lectura o escritura en cada uno de los registros. A0 indica cuando se tiene en el bus el octeto más significativo (A0=1), o el menos significativo (A0=0), cuando A0=0 todo el tiempo, la longitud de la palabra es de ocho bits.

Conversión Analógica

Esta sección está formada por un convertidor digital-analógico y dos amplificadores operacionales que realizan la conversión de corriente a un voltaje analógico de salida.

El convertidor digital-analógico CI40 de la fig. 2.36 es el circuito integrado DAC1230. Este convertidor es de 12 bits de resolución, contiene ocho líneas de entrada de datos, cinco líneas de control y cuatro líneas de salida para la construcción del voltaje analógico.

La conversión se realiza bajo la siguiente secuencia:

La señal -CS se activa con CCONVERT; -WR1 del convertidor M2 se activa con la señal WR1 que presenta un pulso activo bajo, mientras que BYTE-XFER se encuentra en estado alto para la realización de la escritura en el primer registro; -WR2 del convertidor se activa con la señal -WR2 y presenta un pulso activo bajo, mientras que BYTE-XFER se encuentra en estado bajo para que se realice la escritura en el segundo registro, al momento de realizarse esta escritura la señal -XFER es activa con un nivel bajo, lo que indica que los datos han sido transferidos al registro interno del convertidor, después de lo cual se realiza la conversión.

Como resultado de la conversión se obtiene una corriente proporcional al número binario, esta corriente se obtiene de la línea IOUTi. Esta señal se encuentra conectada a la entrada inversora del circuito amplificador CI42. La configuración de CI42 realiza la conversión de corriente a voltaje (0V. a V_{ref}). El corrimiento (*Offset*) de la señal se controla por medio del potenciómetro R81. De la segunda parte se obtiene un voltaje bipolar comprendido entre $+V_{ref}$ a $-V_{ref}$ por medio de la configuración sumadora del amplificador CI43 de la fig. 2.36.

Una parte esencial en la obtención del voltaje analógico de salida es el voltaje de referencia (V_{ref}), cuyo circuito se muestra en la fig. 2.36.

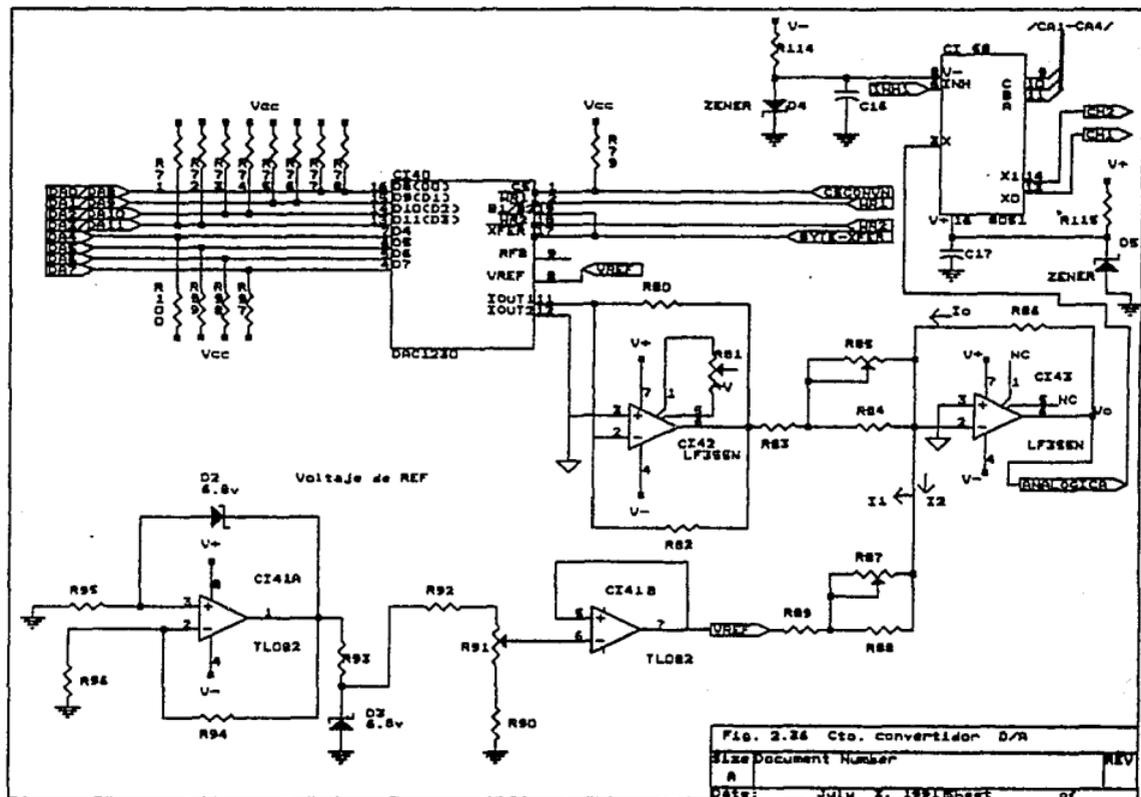


Fig. 2.24 Cto. convertidor D/A
 Size Document Number
 A
 Date: July 2, 1991, Sheet 07

Circuito de Retención

Realiza la retención de los 2 canales analógicos de salida (CH1-CH2), esto se lleva a cabo por una multicanalización y mediante circuitos retenedores de orden cero. El multicanal está constituido por el circuito CI58 mostrado en la fig.2.36. Este circuito es un decodificador de ocho salidas de selección y controlados por las líneas A, B, C.

El circuito contiene los 2 canales de salida con una sola línea de entrada/salida, 2 líneas de dirección para indicar el canal seleccionado y 1 línea de control (la que indica la habilitación del circuito).

2.9 CIRCUITOS DE SINCRONIA DE LA TARJETA DE FUNCIONES MÚLTIPLES

2.9.1 Panorama general

En la fig.2.37 se muestra el diagrama general a bloques que integran los circuitos de sincronía de la tarjeta de funciones múltiples, así como la comunicación con la computadora y el bus local de la tarjeta. Este bloque consta de tres circuitos importantes, los cuales se encargan de la sincronía para la operación de la tarjeta de funciones múltiples. Dichos bloques son: el circuito de señales de sincronía, el circuito generador de interrupciones, el modo de ganancia, el modo de ganancia y el modo de ganancia, el circuito generador de interrupciones, el modo de ganancia y el modo de ganancia.

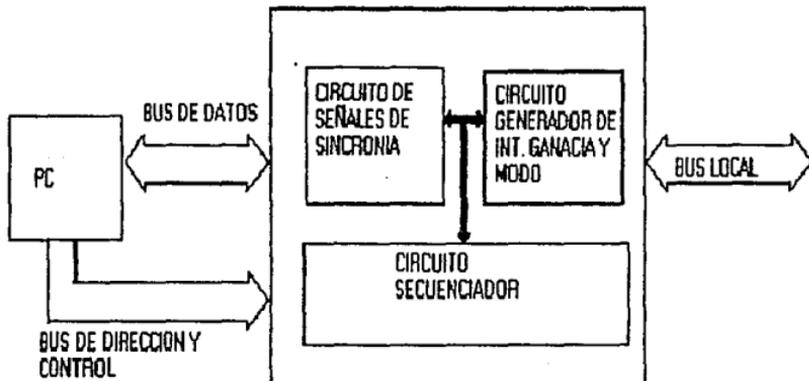


Fig.2.37 Circuitos de sincronía de la tarjeta de funciones múltiples.

Circuito de señales de sincronía

El primer bloque es el circuito de señales de sincronía que se puede observar en la fig.2.38. Este bloque es el encargado de generar las señales que permiten el intercambio de información entre la interfaz y la computadora personal, incluyendo las del control de la conversión A/D, D/A y puertos de E/S Digital. Esta etapa es realizada por medio de una lógica de generación de señales en la interfaz y que como entrada recibe al *bus* de direcciones y algunas líneas del *bus* de control de la PC.

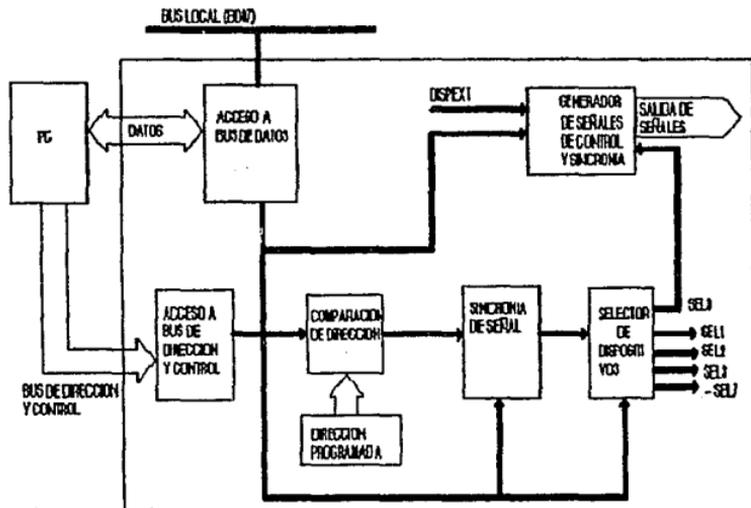


Fig.2.38 Circuito de señales de sincronía

El funcionamiento de este circuito es el siguiente: se efectúa la selección del dispositivo que requiera la tarjeta para su adquisición de información, de esta acción se encarga el selector de dispositivos, el cual es activado mediante los bloques de sincronía de la señal y el de comparación de direcciones válidas. Las direcciones provenientes del microprocesador de la PC junto con las señales de control y sincronía habilitan a los diferentes dispositivos que componen la tarjeta. También este circuito tiene una comunicación de retroalimentación con la PC para que ésta esté informada de todo lo que sucede en el mismo.

Circuito generador de interrupciones, ganancia y modo

El segundo bloque es el circuito generador de interrupciones, de ganancia y modo. Este circuito es el que ordena las interrupciones al procesador de la PC para que efectúe la lectura de los datos entrantes a la computadora, además de ejecutar las instrucciones generadas en las rutinas programadas para el control de la ganancia y del modo de operación de la tarjeta, este circuito se puede observar en la fig.2.39.

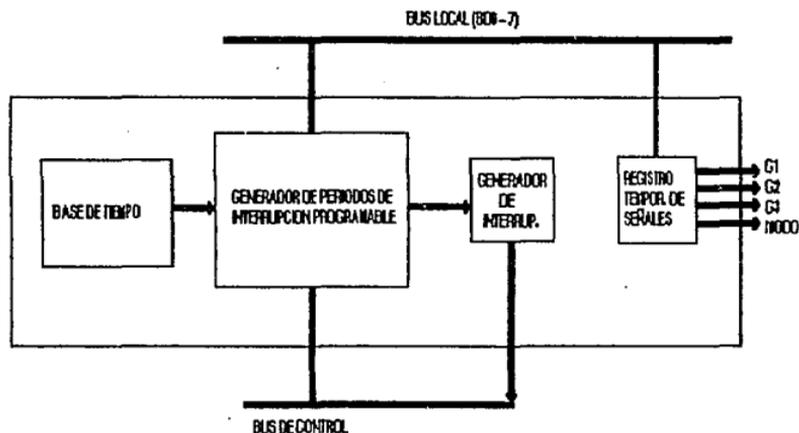


Fig.2.39 Circuito generador de interrupciones, ganancia y modo.

El circuito generador de interrupciones, ganancia y modo, consta principalmente de un generador de periodos de interrupción programable, el cual es sincronizado con una base de tiempo independiente, proporciona las señales adecuadas al circuito generador de interrupciones, que está conectado al bus de control en la PC, para señalar cuando el procesador de ésta debe ser interrumpido. También contiene el registro temporal de señales que se encarga de

seleccionar la ganancia y el modo de operación en el dispositivo retenedor de la tarjeta.

Circuito secuenciador

El tercer bloque es un circuito secuenciador que controla el flujo de señales de control provenientes de la PC, para activar los distintos dispositivos. Este circuito está formado por cuatro secciones:

- a) Temporizador
- b) Decodificador
- c) Contador
- d) Lógica de control

El circuito secuenciador genera ocho pulsos para sincronizar las actividades del convertidor D/A, estos pulsos son en forma secuencial. También genera una serie de señales para ordenar la activación y desactivación de los diversos dispositivos utilizados por el convertidor D/A.

2.9.2 Descripción del circuito generador de señales de sincronía

Descripción del circuito de señales de sincronía del convertidor A/D

Para la conversión A/D, la figura 2.40, muestra el circuito generador de señales de sincronía, el inicio de operación sucede cuando en el bus de dirección de la PC aparece la dirección programada que corresponde a la tarjeta. El acceso al bus de dirección y al de control se hace por medio de los circuitos integrados CI20 y CI21, que son **buffers** activados con nivel bajo. La comparación de la dirección se lleva a cabo mediante el comparador binario de 4 bits (CI23) y los microinterruptores, con estos últimos se programa la dirección de la tarjeta.

Para la programación de la tarjeta seleccionaremos las direcciones disponibles dentro del mapa de memoria de la PC, para asignarlas a los dispositivos de la tarjeta. La señal resultante de la comparación de direcciones junto con -IOR, -IOW y AEN generan la señal DS (bloque de sincronía de señal). Cuando ésta es activa baja esta indicando que el procesador de la PC está accediendo a la tarjeta, ya sea en un ciclo de lectura o en un ciclo de escritura.

Con la señal DS (activa alta) se habilita el circuito de selección de dispositivos, con lo cual una de sus ocho líneas del CI24 es activada, seleccionando al dispositivo correspondiente. El circuito de selección de dispositivo es un decodificador binario de tres por ocho, con control de habilitación.

Mapa de memoria usado para dispositivos de entrada y salida.

Dirección Uso
(HEX)

0200	SIN USO
0201	ADAPTADOR DE CONTROL DE JUEGOS
0202-0277	SIN USO
0278-027F	PUERTO PARA SEGUNDA IMPRESORA
0280-02F7	SIN USO
02F8-02FF	TARJETA DE SEGUNDO PUERTO SERIE
0300-0377	SIN USO
0378-037F	TARJETA PUERTO IMPRESORA
0380-03AF	SIN USO
03B0-03BF	MONOCROMIA Y CONECTOR DE IMPRESORA
03C0-03CF	SIN USO
03D0-03DF	CONECTOR COLOR/GRAFICAS
03E0-03EF	SIN USO
03F0-03F7	MANEJADOR DE DISKETTES (5 1/4)
03F8-03FF	TARJETA PUERTO SERIE

Mapa usado para las direcciones de los dispositivos de la tarjeta de funciones múltiples

Dirección
relativa

Dispositivo

00H-03H	CONVERTIDOR A/D
04H-07H	CRONOMETRO 8253
08H-0BH	RESTAURACION F/F (IRQ2)
0CH-0FH	PROGRAMACION DE GANANCIA Y MODD
10H-1FH	SIN USO

La selección del canal que se va a convertir se realiza con el *Micro-Map* cuádruple (CI22), introduciendo las señales BD0 a BD3. Este es activado por un nivel alto, con lo cual activa el acceso al bus de datos, y grabándose así en la dirección correspondiente, con A, B, C, D (los 4 bits menos significativos del bus de datos), al mismo tiempo, en el bloque generador de señales de sincronía, los circuitos

integrados forman un monoestable activado por flanco, los demás son compuertas lógicas que generan las siguientes señales: S/H que activará el retenedor, -CS y -WR que ordenan el inicio de la conversión al convertidor A/D, también -CS y -RD que juntos permiten la lectura del dato convertido. A continuación mostraremos el diagrama de tiempos de las señales de sincronía fig.2.41.

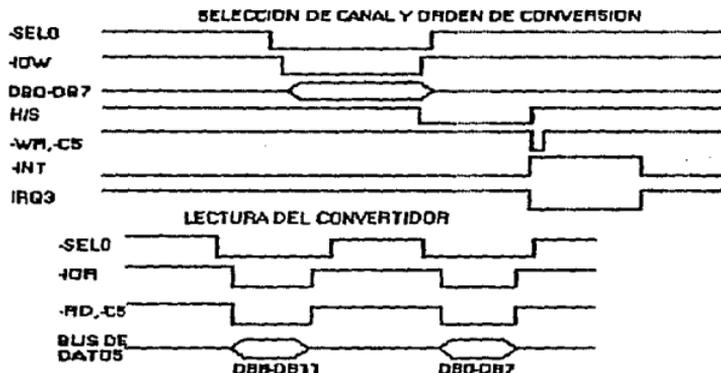


Fig.2.41 Diagrama de tiempos de conversión A/D

La señal DISPEXT es una entrada TTL disponible para el usuario, (ver figura 2.40), con la cual se pueden sincronizar las ordenes de conversión A/D con un generador de señales externo. La señal DISPEXT significa disparo externo, este debe ser un pulso activo alto con un microsegundo como mínimo de ciclo de trabajo.

Descripción del circuito generador de interrupciones, ganancia y modo

En la fig.2.42 se observa el diagrama del circuito generador de interrupciones así como el de ganancia y modo.

El circuito generador de interrupciones consta del circuito integrado 8253, que es un CMOS de alta integración, tiene 3 contadores binarios de 16 bits cada uno, con modo de funcionamiento programable. Con SEL1 este dispositivo es seleccionado y con BAO y BAI se elige cualquiera de sus registros internos. Los contadores del 8253 se encuentran conectados en cascada, por lo tanto los periodos de muestreo a programar pueden ser desde 1 microsegundo hasta 78 horas.

Este dispositivo utiliza una base de tiempo de 1 MHz, la cual es generada por el circuito formado por los inversores D Y E del CI12 en conjunto con el cristal XTAL1 de 1MHz.

Otro circuito que forma parte del generador de interrupciones es el CI15, un *flip-flop* tipo D que es activado por la salida OUT2 del circuito temporizador, su salida IRQ2 interrumpe al procesador de la computadora. Cuando se atiende la solicitud de interrupción, la rutina de servicio deshabilitará IRQ2, generando -SEL2.

El circuito que forma parte de esta sección consiste en la programación del estado de funcionamiento del bloque correspondiente a la ganancia y modo de las señales analógicas de entrada en la tarjeta. Este bloque consiste de un registro temporal, el cual es un *flip-flop* cuádruple CI14, (de nomenclatura 74LS75), activado por nivel, y mediante una compuerta NOR CI16A (de nomenclatura 74LS02), esta es controlada por medio de las señales de control -SEL3 y -IOW. Provenientes del circuito de sincronía de conversión A/D (fig.2.40). Cuando éstas son activas bajas, la compuerta NOR es accionada, activándose a su vez el *flip-flop*, el cual proporciona en su salida las señales de ganancia y modo previamente programadas, y las cuales son tomadas del bus local BDD-3. Al mismo tiempo el circuito de sincronía de conversión A/D (fig.2.40) pone en funcionamiento el circuito retenedor de la tarjeta.

Descripción del circuito de señales de sincronía del convertidor D/A y control del puerto E/S digitales

Este circuito consta de tres partes:

- a) Selección de canales analógicos de salida.
- b) Selección de canales digitales de E/S.
- c) Contador de eventos externos.

La fig.2.43 muestra el circuito correspondiente a las etapas mencionadas. Este circuito consta del decodificador CI34 (de nomenclatura 74LS85) que es un comparador de 4 bits el cual comparará las direcciones A6-A7 con las de un microinterruptor programable (SW2), el resultado de esta comparación genera la señal que indica que alguno de los tres dispositivos es seleccionado, de tal manera que con ella se permite recibir o mandar información a la interfaz. A esta señal se le denomina como DS y su equivalente será la dirección seleccionada ($DS=(A=B)*(-AEN)$).

Con A5 alta se indica la selección de los canales digitales y con A5 baja la selección de los canales analógicos, para la selección de los canales digitales se activa la señal DIS equivalente a $(DIS = -A5 + (-(A=B)) + AEN)$, para los canales analógicos se activa la señal AS y su equivalente será $(AS=A5+(-(A=B))+AEN)$, estos dispositivos son seleccionados siempre y cuando AEN (*access enable*) sea cero lógico.

Cuando se seleccionan los canales analógicos, se indica un proceso de escritura por medio de la línea -IOW (*input/output write*), o un proceso de lectura por medio de -IOR (*input/output read*), éstas se presentan sólo una a la vez. Cuando se trata de escritura -IOW se encuentra activa y se presenta con dos pulsos; el primero significa que se envían los 8 bits menos significativos y el segundo que se envían los 8 bits más significativos, con lo que se logrará tener una palabra de 16 bits. Esto se repetirá de igual forma para el caso de lectura, con la señal -IOR.

La combinación booleana de las señales A0 e -IOW permite tener un pulso de sincronía para los bits menos significativos y otro para los más significativos, esto se hace con las señales LIOW, HLOW, donde sus equivalencias se muestran a continuación:

$$LIOW=A5+(-(A=B))*AEN+(-IOW)+A0$$

$$HLOW=A5+(-(A=B))+AEN+(-IOW)+(-A0)$$

las señales anteriores son para el ciclo de escritura en el convertidor.

La combinación de las señales A0, -IOR permita obtener las señales LIOR, HIOR, que a continuación se describen:

$$LIOR=A5+(-(A=B))+AEN+(-IOR)+A0$$

$$HIOR=A5+(-(A=B))+AEN+(-IOR)+(-A0)$$

estas señales son para el ciclo de lectura en el convertidor.

Si se presenta la selección de canales digitales, la señal DIS indica que hay un ciclo de lectura o escritura en ellos, para la selección de canales de E/S digitales o contadores de eventos externos, se usa la dirección A3, que indica que dispositivo es seleccionado, y con su combinación booleana se obtienen las señales CSPUERTO, CSTIMER las cuales seleccionan el puerto y el timer en el circuito, y su equivalencia es la siguiente:

CSPUERTO=(A3+(-A5)+(-(A=B))+AEN+(-IOW*(-IOR))

CBTIMER=(-A3+(-A5)+(-(A=B))+AEN+(-IOW*(-IOR))

La combinación de -IOR y -DIS realiza la lectura de los canales de E/S digitales, ésta da como resultado las señales IORP, IORT que a continuación se muestran:

IORP=(-IOR)+(-A5)+(-(A=B))+AEN, lectura al puerto.

IORT=(-IOR)+(-A5)+(-(A=B))+AEN, lectura al timer.

De IOW y DIS da como resultado las señales de escritura para los canales de E/S digitales, estas son IOWP, IOWT y que son:

IOWP=(-IOW)+(-A5)+(-(A=B))+AEN, escritura al puerto.

IOWT=(-IOW)+(-A5)+(-(A=B))+AEN, escritura al timer.

Cabe mencionar que todas las líneas de datos, dirección, control, pasan antes por los circuitos integrados CI54, CI55, CI56 (de nomenclatura 74LS244), estos son *buffer* de 3 estados activados por nivel bajo, con los cuales se impide la sobrecarga de las señales eléctricas que proporciona el CPU.

DIRECCIONES ASIGNADAS A LA INTERFAZ

- a) Para canales analógicos:
de 0300H a 031FH, donde las direcciones A0 a la A4 seleccionan los distintos registros.
- b) Para los canales digitales:
de 0320H a 0323H, donde las direcciones A0 y A1 seleccionan los registros del contador.

Las siguientes figuras muestran los diagramas de tiempo para la generación de señales de lectura y escritura, para ambos casos, es decir, del puerto E/S digital, así como del convertidor D/A;

- Fig.2.44 Diagrama de tiempos sobre la selección de los dispositivos timer y puerto (lectura).
- Fig.2.45 Diagrama de tiempos sobre la selección de los dispositivos timer y puerto (escritura).
- Fig.2.46 Diagrama de tiempos sobre la selección del convertidor (escritura).

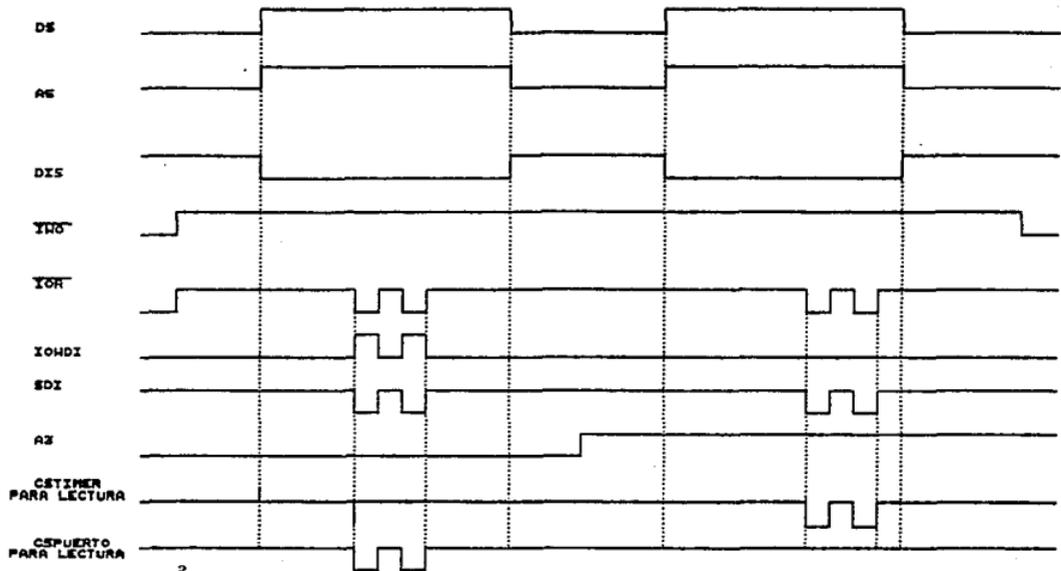


Fig. 2.44 Diagrama de tiempo sobre selecc.
de dispositivos T1600 u. sujeta (lectura)
Size Document Number
A
Date: JUN 4, 1991 Sheet of

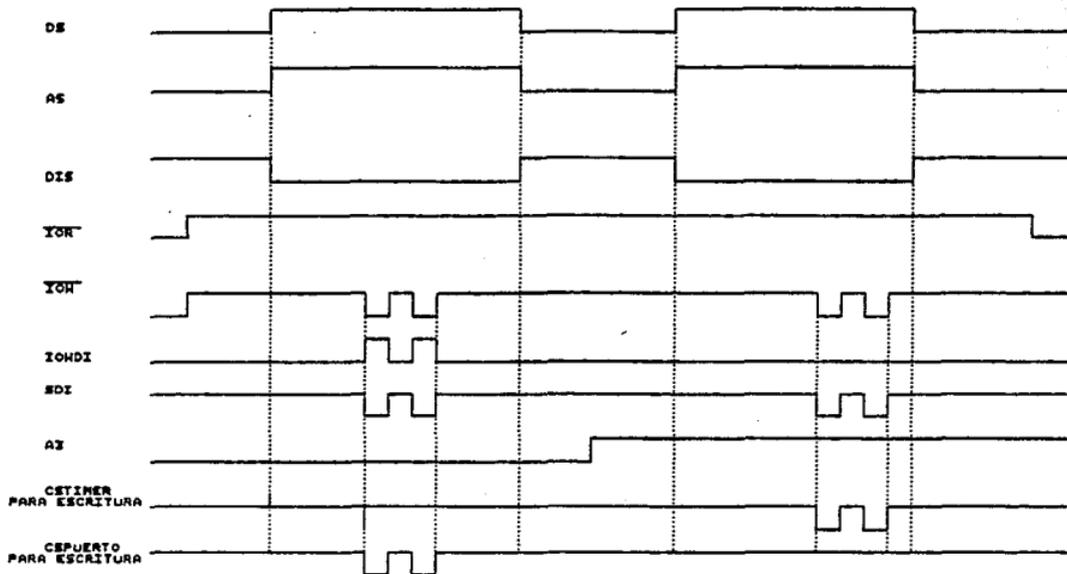


Fig. 2.45 Diagrama de tiempo sobre selecc.
de direcciones. Tercer y cuarto (Escritura)
Size Document Number
A
Date: JULY 4, 1971 Sheet of

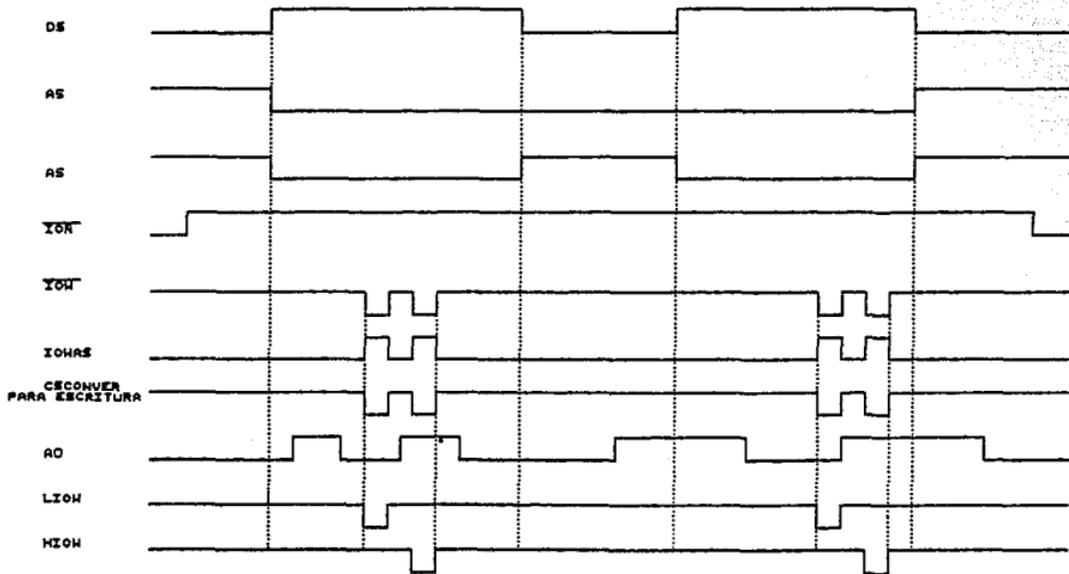


Fig. 2.46 Diagrama de tiempos sobre selección de el conv. D/A (Facilita)

Size Document Number 10

A

DATE: JULY 5, 1951 Sheet of

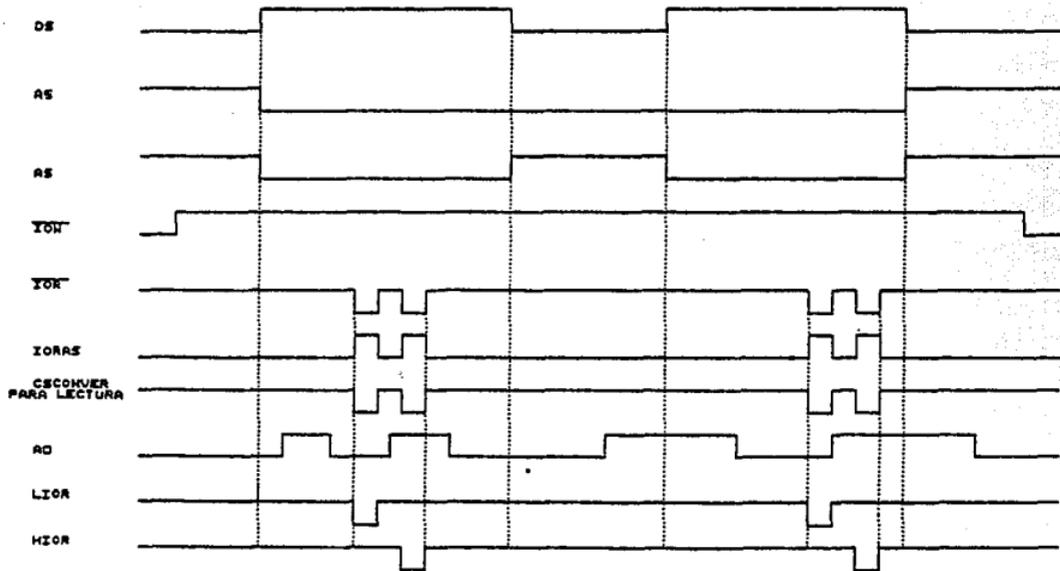


Fig. 2.47 Diagrama de tiempo sobre selección de el modo D/A (lectura)
 Size Document Number: A
 Date: JULY 4, 1998 Sheet 07

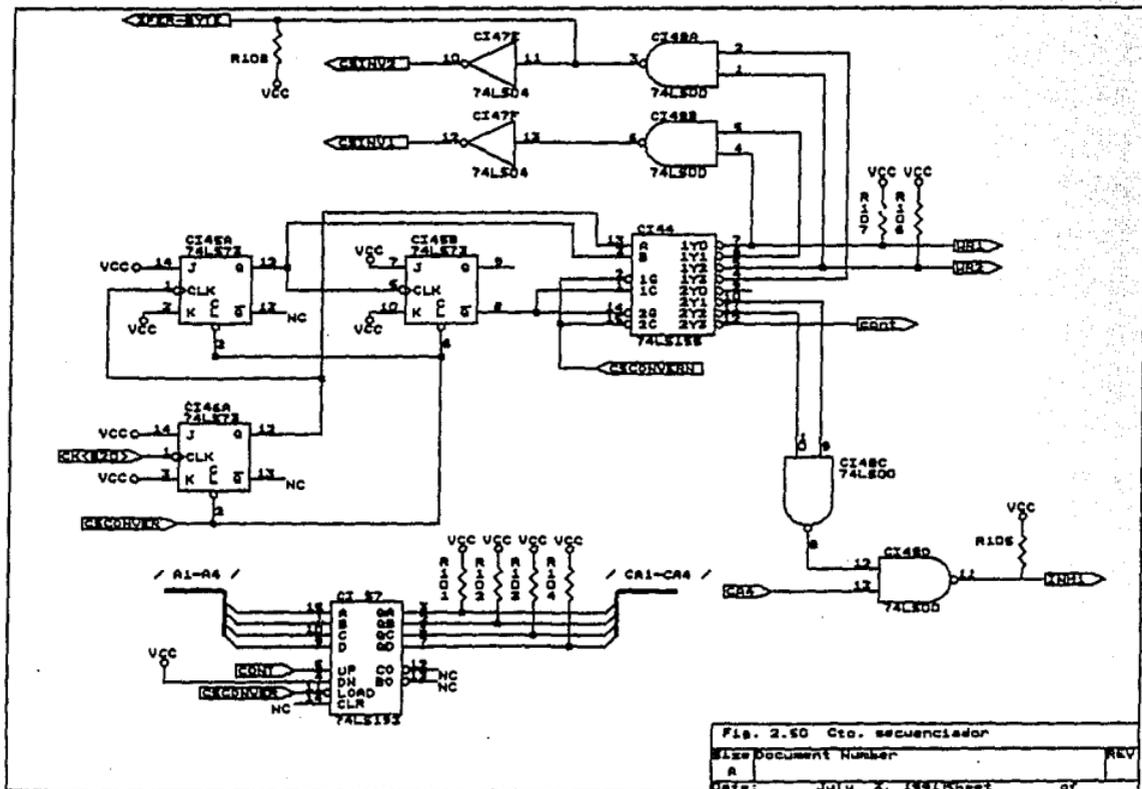


Fig. 2.50 Cto. secuenciador

Sis. Document Number

A

REV

REV: July 3, 1974 Sheet of

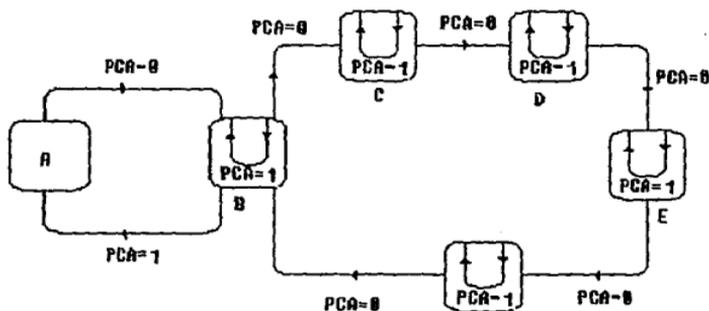
- Fig.2.47 Diagrama de tiempos sobre la selección del convertidor (lectura).

2.9.3 Descripción del circuito secuenciador

Todo elemento de lógica de control tiene un diagrama secuencial de estados.

La secuencia de actividades de conversión y selección del canal analógico es realizada por el diagrama de estados del secuenciador, fig.2.48. En esta figura se muestran los diferentes estados que sigue el secuenciador, y que a continuación se mencionan:

- Lectura de memoria y cargo en el registro del convertidor.
- Inicio de conversión.
- Selección del canal de salida.
- Incremento del apuntador de direcciones.



PCA=1: ACCESO DE LA PC EN LA INTERFAZ
PCA=0: NO ACCESO DE LA PC EN LA INTERFAZ

A: LECTURA O ESCRITURA EN MEMORIA INTERFAZ
B: LECTURA DE MEMORIA INTERFAZ
C: DATO EN REGISTRO
D: CONVERSION D/A
E: MULTICANAL
F: INCREMENTA INDICADOR DE REGISTROS

Fig.2.48 Diagrama de estados del secuenciador

Este circuito inicia su operación con la lectura de un registro en la memoria de la tarjeta; posteriormente, enviará una secuencia de pulsos de control que iniciarán la conversión. Al finalizar la conversión, el secuenciador genera la dirección para el circuito, con lo que activará el canal seleccionado y el valor del voltaje actualizado, después incrementa el apuntador de direcciones y empezará la secuencia nuevamente.

Para que el secuenciador pueda operar necesita que la TFM (Tarjeta de Funciones Múltiples) tenga asignadas un cierto número de direcciones para sus distintas opciones, las cuales son:

- 1- Señales analógicas de salida (4 localidades de memoria)
- 2- Señales digitales de entrada-salida (4 localidades de memoria)
- 3- Conteo de eventos externos (4 localidades de memoria)

0000H	512	DISPONIBLE PARA TARJETA MADRE
01FFFH	512	DISPONIBLE PARA EL USUARIO
0200H	64512	SIN USO
03FFFH		
0400H		
FFFFH		

Fig.2.49 Mapa de memoria de espacio de direcciones de E/S de la PC

Como se observa en la fig.2.49 de la localidad 0000H a la 01FFFH se encuentra el sistema operativo de la PC, y de la 0200H a 03FFFH están disponibles para el usuario. Algunas de éstas se encuentran reservadas para diversos dispositivos de la computadora y sólo algunas son usadas para dispositivos externos, como lo muestra el mapa de memoria.

Localidades asignadas a la TFM:

DIRECCION	USO
0300H-0304H	CANALES ANALOGICOS
0320H-0324H	CANALES DIGITALES
0328H-032BH	CONTEO DE EVENTOS

El circuito secuenciador consta de 4 partes importantes que son:

- a) Temporizador
- b) Decodificador
- c) Contador
- d) Lógica de control

En la fig.2.50 se observa el circuito secuenciador, en la cual el temporizador se encuentra formado por un divisor de frecuencia (tres flip-flop biestables JK en configuración *TOGGLE*), éstos son los circuitos CI45A, CI45B y CI46A. La señal de entrada es el reloj de la computadora (CK<B20>). Este secuenciador genera ocho pulsos para la sincronía de actividades de la TFM. Para ello los decodificadores del circuito CI44 (de nomenclatura 74LS155), se conectan en serie para generar los pulsos.

Las salidas del divisor de frecuencia (Q_1 , Q_2 , Q_3) se conectan a las entradas de CI44 (circuito integrado formado por dos decodificadores de 4 salidas, seleccionadas por dos líneas). La salidas de CI44 son ocho pulsos bajos en forma secuencial (1Y0,1Y1,...,2Y3), debido a esto es llamado secuenciador.

Los circuitos correspondientes al divisor de frecuencia y decodificación son habilitados con la señal CSCONVERN, en activa baja. Esto sucede cuando por parte de la PC no hay un acceso en los registros de los canales analógicos de salida.

Al momento que existe una escritura ó una lectura en uno de los canales analógicos de salida, la terminal LOAD del contador CI57 (de nomenclatura 74LS193), se encuentra habilitado con la señal CSCONVER. Para realizar una carga en paralelo, CSCONVER presenta 2 pulsos activos bajos: el primero habilita la señal de escritura en las memorias; circuitos CI34 y CI35 del arreglo de memorias, y el segundo habilita la señal de lectura de la memoria; CI36 del arreglo anterior. Las salidas del contador CI57 se conectan en forma directa a las líneas de dirección de cada una de las memorias anteriores.

Cabe mencionar que el secuenciador se encuentra en un ciclo continuo de refrescamiento de voltaje en la etapa de salida de la TFM, cuando la PC no accesa las memorias del convertidor.

Secuencia de señales generadas por el secuenciador

Esta parte corresponde a la lógica de control. Primero, se genera la señal CSINV1, fig.2.50, la cual es activa baja y dura 200 nanosegundos, habilita la señal de entrada G de los buffers inversores, circuitos CI37A y CI38A del arreglo de memoria, con lo que activa el paso de la información de las memorias a los registros del convertidor.

Se lee la parte más significativa del dato y al mismo tiempo se genera la señal -WR1 que activa la escritura en el convertidor D/A, DAC1230, (-WR1 y -WR2). Esta escritura se realiza cuando se presenta el flanco de subida. Con el segundo pulso G₁ se genera la señal CSINV2, ésta es activa baja con una duración de 200 nanosegundos, habilita la señal de entrada G del buffer inversor, CI39 del arreglo de memoria. De esta manera lee la parte más significativa del dato a convertir, y al mismo tiempo se genera la señal -WR2, que activa el circuito convertidor D/A (DAC1230), la señal -WR2 activa la escritura cuando se presenta el flanco de subida. La señal XFER-BYTE realiza la transferencia de información al registro de conversión para luego dar inicio a otra adquisición.

Finalmente, se presenta la señal CONT, que es activa baja, la que incrementa el contador; CI58 de la fig.2.50, indicando que se ha terminado la secuencia de conversión de un canal analógico de salida y por lo tanto se direcciona el siguiente canal analógico. El incremento se genera con el flanco de subida y así se inicia la conversión de un nuevo dato.

Con CA1 del secuenciador se habilitará el circuito, CI39 del arreglo de memoria (decodificador de 2 salidas y controlado por una terminal), el cual seleccionará alguno de los dos canales de salida analógicos de la TFM.

Las señales CSCONVER y CSCONVERN provienen del circuito de señales de sincronía, visto en el punto anterior.

A continuación se muestra el diagrama de tiempos del secuenciador fig.2.51.

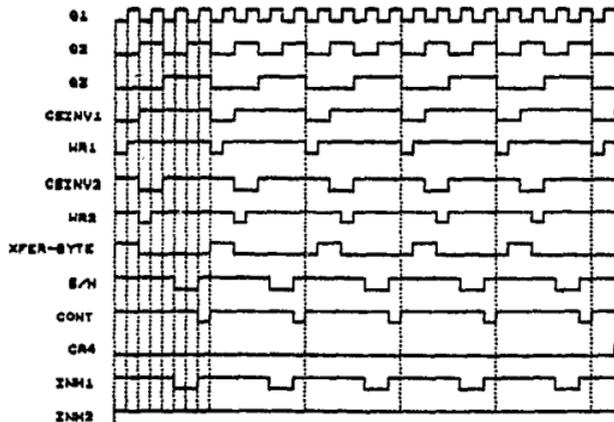


Fig. 2.51 Diagrama de tiempos del registrador		
Size Document Number		REV
A		
DATE: October 8, 1991		Sheet 07

CAPITULO 3

PROGRAMACION DE LA TARJETA DE FUNCIONES MULTIPLES (TFM)

3.1 ESTRUCTURA GENERAL PARA LA PROGRAMACION DE LA TFM

3.1.1 Presentación general

La programación está dividida en tres bloques. El primero sirve para manejar la TFM en la adquisición de las señales analógicas. El segundo se utiliza cuando se necesita que la TFM proporcione datos en forma analógica, y el tercero se usa para adquisición y proporción de datos digitales a través de la TFM, mediante su puerto de entradas/salidas digitales.

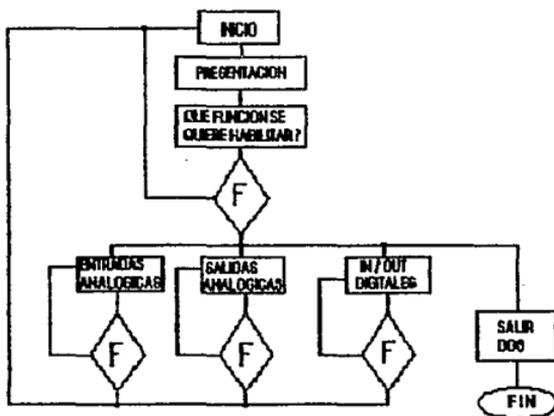


Fig.3.1 Diagrama de bloques del flujo de señales

En cada bloque la programación está compuesta por secciones en lenguaje ensamblador y en TURBO PASCAL. En la parte de TURBO PASCAL se pregunta al usuario la forma en que

quiere que opere la tarjeta, así mismo se habilita y deshabilita las interrupciones de la PC y se almacenan los datos adquiridos en la memoria RAM de la computadora para posteriormente transferirlos a un archivo de memoria magnética (disco flexible). Por otro lado, las secciones en ensamblador se encargan de inicializar los procesos.

3.1.2 Estructura de la programación para entradas analógicas

Las tareas que ha de realizar el programa de control de la TFM, en la adquisición de datos de origen analógico, está ordenado de la siguiente forma:

a) Etapa de inicialización.

- Se establece el modo de operación (diferencial o normal).
- Se establece la ganancia de cada señal analógica antes de entrar al convertidor A/D.
- Se programa el cronómetro.
- Se habilitan las interrupciones del cronómetro y del convertidor A/D.
- Se programa el vector de interrupción correspondiente a cada interrupción.

b) Atención de interrupciones

- Se restablece el **flip-flop** que señala la solicitud de interrupción del cronómetro.
- Se indica el canal a muestrear.
- Se inicia la conversión A/D.
- Se lee el resultado de la conversión A/D.

3.1.3 Tareas del programa para salidas analógicas y puertos de entrada/salida digitales

Tareas del programa:

- Establecer la iniciación de los canales analógicos de salida.

- Programar el contador de eventos externos (palabra de control).
- Programar el puerto paralelo (palabra de control).

Los diferentes componentes de la tarjeta se programan a través de la escritura o lectura en un registro de memoria, cuya dirección ha sido asignada previamente a la TFM.

3.2 PROGRAMACION DE LA TFM PARA LA ADQUISICION DE DATOS DE ORIGEN ANALOGICO

En este apartado se detalla la programación de los diferentes dispositivos que intervienen en la adquisición de datos en forma analógica, como son: el cronómetro, el convertidor A/D, el controlador de interrupciones, el amplificador, etc. Los diferentes componentes de la tarjeta se accesan a través de los puertos (I/O) de la computadora.

A continuación se muestran los registros de la tarjeta y la información que se envía a ésta.

3.2.1 Etapa de iniciación

Como podemos observar en el apéndice 2, los bits del bus de direcciones para accesar dispositivos de entrada/salida del bus de direcciones son del A0 al A9. Una vez que ya se seleccionó el segmento donde se instalará la TFM, los 5 bits menos significativos se utilizarán para direccionar a todos los dispositivos que constituyen la TFM. Es decir, las líneas de A0-A4 se utilizan para realizar las funciones que se muestran en la tabla 3.1.

TABLA 3.1

A4	A3	A2	A1	A0	
0	0	0	X	X	Señalar el canal a muestrear e iniciar conversión A / D
0	0	1	X	X	Accesar los contadores y registros del cronómetro
0	1	0	X	X	Accesar el Ep-Stop del cronómetro
0	1	1	X	X	Señalar el modo de operación y la ganancia

Para el contador, las líneas A0-A1 son usadas para especificar el envío de la información a los contadores o al registro, esto se detallará posteriormente.

3.2.2 Lógica para modo de operación y ganancia del amplificador de instrumentación

Para programar el modo de operación y la ganancia, la información se enviará en un byte, codificado como sigue:

7	6	5	4	3	2	1	0	BITS
X	X	X	X	E ₃	E ₂	G ₁	G ₀	PALABRA PARA MODO DE OPERACIÓN Y GANANCIA

0 = DIFERENCIAL } PARA LOS VALORES DE M
1 = NORMAL

G ₃	G ₂	G ₁	GANANCIA	
0	0	0	0.5	} PARA LOS VALORES DE GANANCIA
0	0	1	1	
0	1	0	2	
0	1	1	4	
1	0	0	8	
1	0	1	16	
1	1	0	32	
1	1	1	100	

3.2.3 Programación del cronómetro

Esta constituido por tres contadores en cascada y un registro de control. Su función es establecer la frecuencia de muestreo del convertidor A/D, ésta es fijada al asignar el valor inicial de la cuenta de cada contador y su modo de operación. La función de cada contador es, cargar el número a decrementar, leer el estado actual de la cuenta y acceder el registro de control, para especificar el modo de operación de cada contador.

La función de control se realiza con las líneas A0-A1 del bus de direcciones de la PC y -RD junto con -WR, como se observa en la tabla 3.2.

TABLA 3.2

-RD	-WR	A1	A0	EJECUCION
0	0	0	0	Cargar contador 0
1	0	0	1	Cargar contador 1
1	0	1	0	Cargar contador 2
1	0	1	1	Escribir el modo de operación al registro de control
0	1	0	0	Leer contador 0
0	1	0	1	Leer contador 1
0	1	1	0	Leer contador 2
0	1	1	1	No operación

En la programación del cronómetro se deberán efectuar dos funciones, establecer el modo de operación y cargar un número base en cada contador, esto se efectúa mediante la palabra de control que se ve en la tabla 3.3.

TABLA 3.3

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	RL1	RL0	M2	M1	M0	BCD

Esta palabra de control es enviada por la computadora al registro de control del cronómetro.

Los bits D7-D6 permiten la selección del contador (SC0-SC1), como se muestra a continuación:

SC1	SC0	FUNCIÓN
0	0	Selecciona contador 0
0	1	Selecciona contador 1
1	0	Selecciona contador 2
1	1	Ilegal

Los bits D5-D4 son usados para la escritura-lectura del valor a decrementar (RL0-RL1), según la relación siguiente:

RL1	RL0	FUNCIÓN
0	0	Lectura al vuelo
0	1	Escribir/leer <i>byte</i> más significativo
1	0	Escribir/leer <i>byte</i> menos significativo
1	1	Escribir/leer <i>byte</i> menos significativo y después el más significativo

El cronómetro tiene cinco modos de operación, éstos son seleccionados por medio de (M0-M2) y de acuerdo con la tabla 3.4. Los modos que se utilizan son sólo el 2 y 3. El primero genera una señal cuadrada cuyo nivel bajo dura un medio de la cuenta a decrementar y el nivel alto la otra mitad, el segundo mantiene en nivel alto la salida del contador y

genera un pulso de nivel bajo, esta duración es de un periodo del reloj de entrada cuando la cuenta llega a cero.

TABLA 3.4

M2	M1	M0	Modo
0	0	0	0
0	0	1	1
X	1	0	2
X	1	1	3
1	0	0	4
1	0	1	5

El último bit es DO, (BCD=0), especifica si el número a decrementar está en código binario, (BCD=1) si el número a decrementar está en BCD.

3.2.4 Habilitar las interrupciones del cronómetro

Las interrupciones son manejadas mediante el circuito integrado (8259A), éste es un controlador de interrupciones de 8 niveles, identificados como (IRQ0/IRQ7), IRQ0 tiene la prioridad más alta e IRQ7 la más baja, este controlador es programado por el sistema operativo de la PC; sin embargo, la forma de operar es modificable, puede modificarse a través de tres palabras de control, de éstas las dos primeras se utilizan para enmascarar y habilitar interrupciones.

El controlador consta de tres registros, uno es el IMR o registro de enmascaramiento de interrupción, cuya función es almacenar los bits que enmascaran las interrupciones. De las líneas de petición de interrupción (IRQ0-IRQ7), la TFM ocupa dos que están libres en el bus de la PC. Con IRQ2 controla el cronómetro y con IRQ3 el convertidor A/D.

Para que el sistema de la PC identifique cada interrupción, ésta asigna un número con el cual se permite, mediante un mapeo, determinar la dirección de la memoria donde se localiza la rutina de atención de la interrupción. Para este diseño a las interrupciones provenientes de IRQ2

se le asignará la dirección OAH y para las provenientes de IRQ3 la asignación será OBH.

Para la programación de las líneas de petición de interrupción (IRQ0-IRQ7) se utiliza una palabra que consta de los bits (M0-M7).

M7	M6	M5	M4	M3	M2	M1	M0
IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0

Para habilitar las interrupciones provenientes de IRQ2 e IRQ3 se modifica la palabra de control de operación (OCW1) y se envía al registro de enmascaramiento de interrupciones, cuyo puerto es 21H. Cada bit de la palabra permite habilitar o deshabilitar a cada una de las interrupciones enmascarables, por lo tanto con M=1 la interrupción es enmascarada (inhibida) y con M=0 la interrupción es habilitada.

3.2.5 Atención de interrupciones

Las tareas a realizar por las rutinas de atención a las interrupciones serán las siguientes:

a) Interrupción del cronómetro

Las tareas principales que ejecutará la rutina de atención para la interrupción del cronómetro serán dos.

- 1.- Restablecer el *flip-flop* del cronómetro, para señalar que la interrupción ya está siendo atendida.
- 2.- Pedir el inicio de la conversión A/D, indicando el canal a muestrear.

b) Interrupción A/D

Cuando el convertidor A/D ha terminado la conversión, interrumpe a la PC para indicarle que ya es posible la lectura de la misma, la cual es leída en dos accesos, ya que el convertidor entrega la conversión en dos *bytes*, primero el *byte* más significativo y después el menos significativo.

c) Habilitación de interrupciones

El controlador de interrupciones, al atender una solicitud de interrupción, deshabilita los demás niveles de interrupción y el microprocesador de la PC asigna un cero a la bandera IF del registro de banderas para deshabilitar las interrupciones. La rutina de atención de la interrupción deberá indicar al controlador 8259A que los otros niveles de

interrupción son habilitados. La señal (EOI) es enviada al puerto 20H y la palabra de control 2 (OCW2) es quien desenmascara a las interrupciones. Sin embargo, las interrupciones que llegan al controlador no son atendidas, por lo que adicionalmente se asignará un 1 lógico a la IF, para que las interrupciones que lleguen al mismo puedan ser atendidas por el microprocesador de la PC.

3.2.6 Diagrama de flujo para la adquisición de datos de origen analógico

El desarrollo de este programa es para realizar el muestreo de señales eléctricas (menos de 100 Hz), primero se realiza un muestreo durante un intervalo de tiempo x , siendo almacenadas inicialmente las muestras en la memoria RAM de la PC, una vez transcurrido este tiempo, se procede a realizar el almacenamiento de la muestra en un archivo de memoria magnética para su posterior análisis.

El programa permite muestrear hasta 16 canales en forma consecutiva, variar la frecuencia de muestreo, seleccionar el modo de operación y la ganancia de la sección de amplificación, variar el tiempo en el cual se realizará el muestreo, dar un nombre al archivo donde se almacenan las muestras. En el programa los canales son muestreados consecutivamente a la misma frecuencia, se pueden aplicar diferentes valores de amplificación y se muestrean con el mismo modo de referencia (diferencial o normal), durante el mismo intervalo de tiempo.

En las figs. 3.2, 3.3 y 3.4 se muestra el diagrama de flujo del programa. En este, la bandera de validación se cambia a 1, una vez que se han muestreado todos los canales, señalando entonces que es posible proceder a almacenar las muestras en la memoria RAM de la PC. El listado de este programa se encuentra en el apéndice 5.

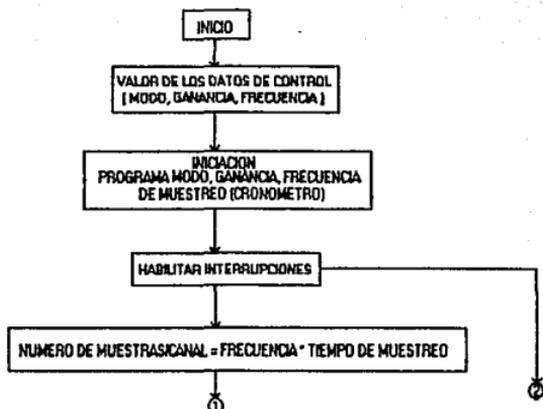


Fig.3.2 Diagrama de flujo del bloque correspondiente a la adquisición de datos analógicos (primera parte).

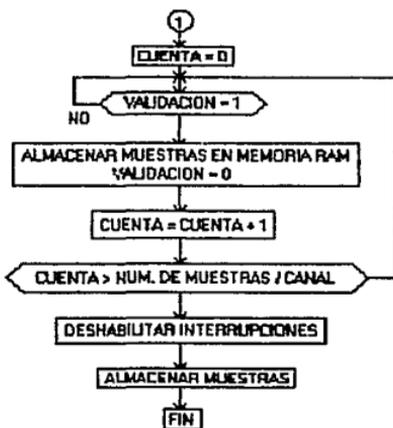


Fig.3.3 Diagrama de flujo del bloque correspondiente a la adquisición de datos analógicos (segunda parte).

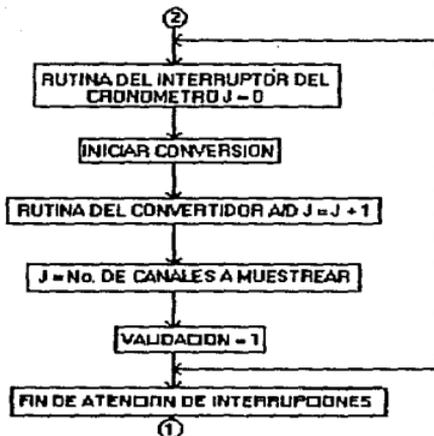


Fig.3.4 Diagrama de flujo del bloque correspondiente a la adquisición de datos analógicos (tercera parte).

3.3 PROGRAMACION DE LA TFM PARA LA SALIDA DE SEÑALES EN FORMA ANALOGICA

En la programación de los diferentes dispositivos de la interfaz para las salidas analógicas, se define la escritura o lectura en un registro de memoria, cuya dirección se ha asignado previamente, de tal modo que no se traslape con las partes de las entradas analógicas. A continuación mencionamos las direcciones de los registros que contiene la interfaz y se especifica la función de cada uno.

0300H - 031FH	Canales analógicos de salida.
0320H - 0324H	Canales digitales entrada/salida.
0328H - 032BH	Contadores de eventos externos.

3.3.1 INICIACION DE LOS CANALES ANALOGICOS DE SALIDA.

Como se vió en el apartado 3.2, en el bus de direcciones A0-A9, sólo se utilizan cinco bits (A0-A4) para direccionar los diferentes canales analógicos de salida con que cuenta la interfaz, que en este caso son 2.

La tabla 3.5 muestra las direcciones (A0-A4) para cada uno de los canales analógicos de salida.

TABLA 3.5

A4	A3	A2	A1	A0	Asignación del canal
0	0	0	0	0	Canal 0 parte menos significativa
0	0	0	0	1	Canal 0 parte más significativa
0	0	0	1	0	Canal 1 parte menos significativa
0	0	0	1	1	Canal 1 parte más significativa

En este aspecto, la programación consiste en cargar un dato con el valor binario al canal analógico seleccionado. Los datos tienen una longitud de 12 bits. Como vemos en la tabla, se emplean dos registros para la programación de cada canal analógico. En el primer registro se programan los bits menos significativos (D0-D7), y en el segundo registro los bits más significativos (D8-D11).

La lectura de datos se inicia al obtener, por uno de los puertos, el dato que viene en complemento a dos, el cual debemos invertir, para posteriormente aplicar la fórmula de conversión a voltaje, teniendo la expresión siguiente:

$$\text{Voltaje a programar} = [(\text{dato} - 2048) / 2048] * 5$$

Para escribir datos se sigue el siguiente procedimiento: se inicia con la conversión del voltaje a programar a un número binario, por medio de la fórmula:

$$\text{Dato previo} = 2048 * (\text{voltaje a programar} / \text{voltaje de referencia})$$

$$\text{Dato a programar} = (\text{2048} + \text{parte entera del dato previo})$$

3.4 PROGRAMACION DE LA TFM PARA EL PUERTO DE ENTRADA/SALIDA DIGITAL

Este puerto consta de 3 puertos independientes y un registro de control. La capacidad de este puerto es para la escritura de 16 canales digitales de salida y la lectura de 8 canales digitales de entrada. Este puerto puede ser

utilizado para la lectura y escritura cuando el usuario de la TFM lo desee.

Actividades básicas que realiza el puerto.

- a) Iniciar la palabra de control.
- b) Leer o escribir el estado de un dato en cada uno de los puertos.

Estas funciones se realizan con las líneas A0-A1 del bus de direcciones, junto con las líneas -RD y -WR, y de acuerdo a la siguiente tabla.

TABLA 3.6

RD	WR	A1	A0	
1	0	0	0	Cargar a puerto A
1	0	0	1	Cargar a puerto B
1	0	1	0	Cargar a puerto C
1	0	1	1	Escribir el modo de operación al registro de control
0	1	0	0	Leer el puerto A
0	1	0	1	Leer el puerto B
0	1	1	0	Leer el puerto C
0	1	1	1	No operación

3.4.1 Formato de la palabra de control enviada al puerto

En la fig.3.5 se muestra la palabra de control enviada al puerto de entrada/salida digital.

D7	D6	D5	D4	D3	D2	D1	D0
F	HQ	M1	PA	PCU	MB	F8	PCL

Fig.3.5 Palabra de control

Las funciones de cada bit se describen a continuación:

- El **bit** 7 indica la activación de la bandera.

- Los **bits** 6 y 5 el modo de selección:

MO	M1	MOD0
0	0	Modo 0 ---- entrada/salida básica
0	1	Modo 1 ---- entrada/salida habilitada
1	X	Modo 2 ---- entrada/salida bidireccional

- El **bit** 4 indica la configuración del puerto A. Como entrada (PA = 1), y como salida (PA = 0)
- El **bit** 3 indica la configuración del puerto C, parte alta. Como entrada (PCU = 1), y como salida (PCU = 0)
- El **bit** 2 indica el modo de operación para el grupo B. Como entrada (MS = 1), y como salida (MS = 0)
- El **bit** 1 indica la configuración del puerto B. Como entrada (PB = 1), y como salida (PB = 0)
- El **bit** 0 indica la configuración del puerto C, parte baja. Como entrada (PCL = 1), y como salida (PCL = 0)

La programación de los puertos será de la siguiente manera: los puertos "A" y "B" se programarán como salidas y el puerto "C" como entrada.

Este programa permite realizar la lectura/escritura en los canales de entrada/salida digitales, obtener el estado de cada uno de los contadores en la tarjeta y la asignación de estados lógicos en los canales digitales de salida, así como, la lectura de los estados lógicos a la entrada de los canales digitales.

3.5 SIMULACION DE LA TFM

Para la simulación de la tarjeta de funciones múltiples o TFM se consideró dar a un futuro usuario una demostración de como opera la TFM o de como el deberá moverse en las diferentes pantallas de presentación, dependiendo de la operación que quiera efectuar en la misma. Esta simulación se efectúa enviando mensajes de la actividad que en ese momento efectuaría, como si realmente estuviera conectada la TFM en la ranura de la PC. No se consideró el efectuar una simulación mas profunda, esto es, generar interrupciones reales al sistema operativo, simular una entrada, pasar los datos a direcciones de memoria, incrementar un reloj, etc. o

sea, como esta diseñado el programa real, debido a que no tiene ningún fin práctico ni educativo y sólo se complicaría el trabajo generando muchas rutinas por SOFTWARE que no reflejan el funcionamiento electrónico de la tarjeta.

A continuación se da una guía práctica para poder manejar el programa real.

Desde el subdirectorío donde se haya instalado el programa, teclear 'TFM', el cual es el nombre del módulo ejecutable, después de esto aparecerá en primer lugar una pantalla con la presentación del trabajo, como se muestra a continuación:

```
*****
*
*           TARJETA DE FUNCIONES MULTIPLES           *
*
*           DIVISION DE EDUCACION CONTINUA DE       *
*           LA FACULTAD DE INGENIERIA DE LA UNAM     *
*
*           PROGRAMA DE APOYO A LA TITULACION       *
*
*
*
*****
```

Para continuar con el programa de simulación se deberá presionar la tecla ENTER, y aparecerá el menú principal, que se presenta como sigue:

```
*****
*
*           MENU PRINCIPAL                           *
*
*           TECLEE LA OPCION DESEADA                *
*           PARA PROGRAMAR LA TFM                   *
*
*           CANALES ANALOGICOS ..... A.            *
*           CANALES DIGITALES ..... B.             *
*           SALIR A DOS ..... C.                   *
*
*
*****
```

3.5.1 Menú Principal

En este panel tenemos las dos opciones principales que puede manejar la tarjeta, o sea si en este momento vamos a trabajar con alguna señal de tipo analógico (ya sea entrada

ó salida), deberemos teclear una 'A' y pasaremos a la siguiente pantalla:

```
*****
*                                     *
*           MENU PARA SEÑALES ANALOGICAS           *
*                                     *
*           TECLEE LA OPCION DESEADA               *
*                                     *
*   PROGRAMAR ENTRADAS ANALOGICAS ... A.          *
*   PROGRAMAR SALIDAS ANALOGICAS .... B.         *
*   REGRESAR AL MENU PRINCIPAL ..... C.         *
*****
```

Como podemos ver las opciones que tenemos aquí son el seleccionar si la función que queremos efectuar será de entrada o de salida. Si queremos recibir señales analógicas el usuario deberá conocer los siguientes datos de la señal a capturar:

- 1.- Tipo de la señal, diferencial o normal.
- 2.- Factor de ganancia, que puede ser del 0 al 7 de acuerdo a la siguiente tabla :

0	= 00.5 V.
1	= 01.0 V.
2	= 02.0 V.
3	= 04.0 V.
4	= 08.0 V.
5	= 10.0 V.
6	= 50.0 V.
7	= 100.0 V.
- 3.- Considerar que en modo diferencial sólo se dispone de 8 canales y que en modo sencillo se dispone de 16.
- 4.- Numero de canales a muestrear.
- 5.- La frecuencia de muestreo, esta no deberá exceder el siguiente algoritmo $1/(0.0005 * \text{número de canales})$.
- 6.- Por último, el tiempo en el cual se efectuarán las muestras dependerá de la memoria disponible que tengamos en nuestra PC.

Si elegimos la opción de salidas analógicas los datos a conocer son los siguientes.

- 1.- El voltaje a programar, sólo podrá estar entre -5.0 V. y 4.9975 V. (de acuerdo a la interfaz).
- 2.- En que puerto queremos la salida (0 ó 1).

Si nuestro requerimiento son entradas o salidas digitales, desde el menú principal se deberá teclear una 'B' , y nos desplegará la siguiente pantalla :

```

*****
*
*           MENU PARA SEÑALES DIGITALES           *
*
*           TECLEE LA OPCION DEBEADA              *
*
* VER EL ESTADO DEL CONTADOR                      *
*   CONTADOR 1 ..... A.                          *
*   CONTADOR 2 ..... B.                          *
*   CONTADOR 3 ..... C.                          *
*   LEER PUERTO C ..... D.                       *
*   ESCRIBIR AL PUERTO A ..... E.                *
*   ESCRIBIR AL PUERTO B ..... F.                *
*   MENU PRINCIPAL ..... G.                      *
*
*****

```

Con las tres primeras opciones 'A', 'B' y 'C', se puede llegar a contabilizar hasta 65,536 eventos externos, por cada uno, estos contadores son independientes y son iniciados en este paso.

Para 'D', entradas digitales, se efectúa una lectura directa del valor en las terminales del puerto C y nos presentará en pantalla el valor de los bits leídos.

Si se requieren salidas digitales, opción 'E' ó 'F', el programa preguntará por cada uno de los canales o bits de salida y se deberá indicar el valor deseado (1/0).

Por último, en cada uno de los paneles de selección aparece una opción para salir del programa e irnos al DOS de la máquina.

CONCLUSIONES

Los análisis que se han hecho para la elaboración de este tipo de tarjeta se realizarán en función de la necesidad de tener mayor flexibilidad y contar así con un número mayor de opciones, como es el tener varias entradas/salidas - analógicas/digitales. Así mismo, en el amplificador de instrumentación se utilizan componentes de bajo nivel de ruido, para obtener una alta fidelidad en la obtención de información de origen analógico.

Por otro lado, encontramos que los componentes más importantes como los convertidores, son difíciles de obtener en el mercado nacional, por lo que estaremos dependiendo del tiempo y forma de adquisición.

En la simulación del funcionamiento de dicha tarjeta, encontramos que no es representativo tener el programa si no se tiene físicamente la tarjeta, por lo que no se consideró este proceso.

Se encontró que el rango de aplicaciones de la interfaz es amplio, ya que es útil en la Industria y en Laboratorios de Instituciones Educativas y de Investigación. El objetivo primordial es por lo tanto, su construcción para Laboratorios de Escuelas de Recursos Moderados, ya que con este diseño se podrá tener una interfaz de datos confiables y a bajo costo, en comparación con su valor comercial en el mercado (ver apéndice 6 de costos).

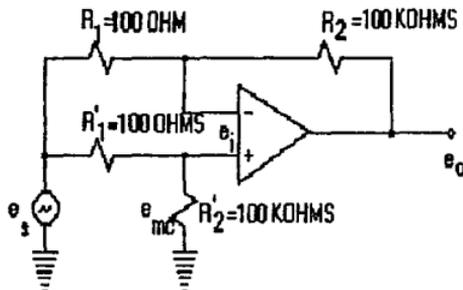
De acuerdo a lo anterior, concluimos que la construcción de esta tarjeta abatirá los costos de las existentes en el mercado, siempre y cuando se tome en cuenta el lapso de tiempo adecuado para su realización.

BIBLIOGRAFIA

- H. TAUB, D.SCHILLING Digital and Integrated Electronics.
Mc-Graw Hill International Editions 1977.
- J.G. GRAEME, G.E. TOBEY Operational Amplifiers.
Mc-Graw Hill Book Company 1978.
- R.F. COUGHLIN, F.F. DRISCOLL Circuitos Integrados Lineales y
Amplificadores Lineales.
Prentice Hall
- J.W. GAULT, R.L. PIMMEL Introduction to Microcomputer-Based
Digital Systems.
Mc-Graw Hill 1ª Edition.
- T.C. BARTEE Digital Computer Fundamentals
Mc-Graw Hill 5ª Edition.
- M.M. MAND Arquitectura de Computadoras
Prentice Hall 1ª Edición
- L.J.AGUILAR Programación en Turbo-Pascal
Mc-Graw Hill
- C.L. MORGAN, M. WAITE Introducción al Microprocesador
8086/8088
Mc-Graw Hill
- G.H. MAC-EWEN Introduction to Computer System
Mc-Graw Hill
- J.M. URUJUELA Microprocesadores, Programación e
Interconexión
Mc-Graw Hill
- R. BOYLESTAD, L. NASHELSKY Electrónica Teoría de Circuitos
Prentice Hall 3ª Edición
- LINEAR DATA BOOK National Semiconductor Corporation 1 & 2

APENDICE 1

DEMOSTRACION PARA LA OBTENCION DE LA RAZON DE RECHAZO DE MODO COMUN.



La relación del rechazo de modo común se define de acuerdo a la siguiente ecuación:

$$RRMC = A_d / A_{mc}$$

La ganancia de modo diferencial es:

$$A_d = e_o / e_i \quad \text{donde: } e_i = (R_1 / (R_1 + R_2)) e_o$$

$e_{mc} = e_s$ siempre y cuando se cumpla que $R_2 \gg R_1$.

La ganancia de modo común es:

$$A_{mc} = e_o / e_{mc}$$

Sustituyendo las ecuaciones y condiciones anteriores tenemos:

$$RRMC = A_d / A_{mc} = (e_o / e_i) / (e_o / e_{mc}) = e_{mc} / (R_1 / (R_2 + R_1)) e_o$$

simplificando:

$$RRMC = A_d / A_{mc} = ((R_2 + R_1) / R_1) (e_{mc} / e_o)$$

dado lo anterior se determina que $e_s / e_o = 1$, por lo tanto tenemos la relación de rechazo de modo común en dB:

$$RRMC = 20 \text{ LOG}((R_2 + R_1) / R_1)$$

APENCICE 2

DESCRIPCION DE LAS SEÑALES DE LA PC, NECESARIAS PARA EL FUNCIONAMIENTO DE LA INTERFAZ

El Bus del Sistema IBM-PC

La mayoría de las componentes funcionales del sistema en la tarjeta principal (tarjeta madre) están unidos al microprocesador 8088 a través del bus del sistema. Este bus está constituido por varios tipos de líneas: bus de datos, bus de dirección, control, sincronía, solicitudes de interrupción, y control de acceso directo a memoria (DMA). Físicamente, el bus empieza en las terminales del microprocesador 8088. Esta interfaz está altamente codificada y multicanalizada y comúnmente se le llama bus local. Conectados al bus local están: (1) un conector auxiliar capaz de aceptar al procesador numérico 8087 de Intel, (2) un controlador de interrupciones de ocho niveles 8259A, (3) el controlador de bus 8288, y (4) circuitos de amplificación de potencia del bus y demulticanalización del bus local, los circuitos de amplificación de potencia manejan las señales básicas que conforman al bus del sistema. Ligados al bus del sistema en la tarjeta del procesador están: (1) Los circuitos de soporte para el procesador (incluyendo los canales de DMA y los contadores del cronómetro), (2) Los puertos de entrada/salida (E/S) y lógica de decodificación de dirección de memoria, (3) las memorias ROM, (4) RAM, (5) Los interruptores de programación, (6) Los adaptadores integrados de entrada/salida, y (7) cinco ranuras para inserción de tarjetas de diferentes usos.

Operaciones en el Bus del Sistema

La mayoría de las aplicaciones de interfaz se conectan a la PC a través de una de las cinco ranuras para tarjeta que se encuentran en la tarjeta madre. En el bus del sistema, los datos se transfieren durante lo que se llama ciclo del bus. Existen dos clasificaciones generales para los ciclos del bus, manejadas por el 8088 y por DMA. Cuando el microprocesador 8088 genera un ciclo de bus, maneja al sistema con una dirección a una localidad de memoria o a un puerto de entrada/salida (E/S), controla la dirección del flujo de datos, y es fuente o receptor de los datos.

Existen cinco tipos diferentes de ciclos de bus. El primero es un ciclo de bus de lectura de memoria. El segundo es un ciclo de bus de escritura en memoria. El tercer tipo es un ciclo de bus de lectura de puerto E/S. El cuarto tipo es un ciclo de bus de escritura en un puerto E/S. El quinto es un ciclo de bus de reconocimiento de interrupción. Este

último ciclo sólo se presenta en el bus local y no en el del sistema.

A continuación se describen los ciclos de escritura y lectura de puertos E/S.

Ciclo de Lectura a Puerto E/S

Cada vez que se ejecuta una instrucción IN, el microprocesador 8088 inicia un ciclo de lectura al puerto de E/S. Su propósito es recoger los datos de una de las direcciones asignadas a los puertos de E/S. En la PC, el ciclo de bus siempre es un mínimo de cinco pulsos de reloj, o aproximadamente 1.05 microsegundos de duración. Un dispositivo específico de E/S puede extender la duración del ciclo de bus desactivando la señal READY del bus. Durante un ciclo de lectura al puerto de E/S, el microprocesador 8088 maneja sobre el bus de direcciones del sistema una dirección de 16 bits. Hay que notar que durante este ciclo de bus, los cuatro bits de mayor orden del bus de dirección nunca son activados. En la figura 1 se ilustra la sincronía básica de un ciclo de bus de lectura de puerto E/S.

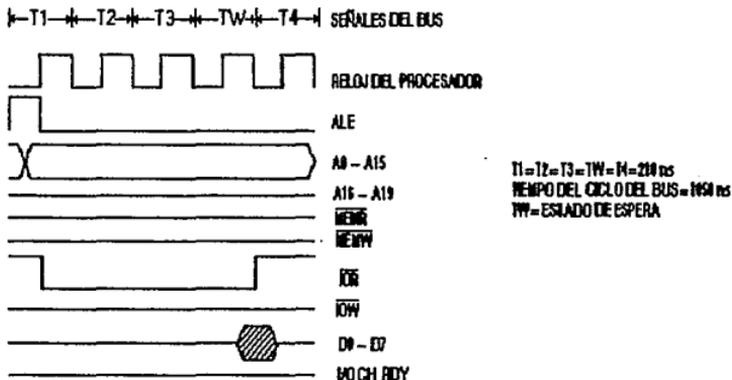


Fig.1 Ciclo de bus para lectura a puerto E/S

también están presentes en las ranuras para tarjetas montadas en la tarjeta madre. El nivel cero se usa en la tarjeta para interrupciones del cronómetro/contador de la computadora. El nivel uno se usa en la tarjeta del sistema para aceptar interrupciones provenientes del teclado de la computadora.

Descripción de las Señales del Bus del Sistema

Todas las ranuras para tarjeta contienen señales idénticas en cada poste. Todas las señales tienen niveles lógicos TTL, excepto para aquellos que fueron provistos para proporcionar energía (*power*) y tierra (*ground*). El bus es el mismo conjunto de señales del bus del microprocesador, sólo que demultiplicadas y amplificadas en potencia. Además de las señales del bus del microprocesador el bus del sistema incluye señales que manejan el acceso directo a la memoria, interrupciones, sincronía, control de lectura y escritura a memoria o dispositivos E/S, generación de estados de espera (*wait state*), y refrescamiento de memoria. A continuación se definen aquellas señales del bus del sistema que se utilizan en la tarjeta de conversión A/D.

Definiciones de Señales

CLK (reloj)

Esta es una señal de salida solamente. Se obtiene de dividir la señal OSC (14.31 MHz de la PC) entre tres, dando una frecuencia de 4.77 MHz. Esta señal no es simétrica y tiene un ciclo de trabajo de un tercio. Esto es, su período es de 210 ns con un pulso alto de 70 ns y uno bajo de 140 ns. Esta señal se encuentra sincronizada con los controles de lectura y escritura en memoria y puede ser usada para generar estados de espera en el bus del sistema. Un ciclo de bus típico es de cuatro períodos de reloj, aproximadamente 840 ns.

A0 - A19

Los bits de dirección A0 hasta A19 son señales de salida solamente, que se usan para direccionar la memoria conectada al bus del sistema y dispositivos E/S, (siendo A0 el bit menos significativo). Esas 20 líneas de señal son manejadas por el microprocesador 8088, durante los ciclos del bus del sistema para lectura o escritura de memoria o dispositivos E/S. También son manejadas por la lógica de acceso directo de memoria durante ciclos DMA. Con 20 líneas de dirección, es posible direccionar un megabyte de memoria, pero no todo el espacio de direcciones está disponible en el bus del sistema. La memoria base del sistema (RAM de 0 a 64K) reside en la tarjeta base y su espacio de direcciones no puede ser

direccionado en el bus del sistema. Similarmente, la tarjeta base del procesador contiene un espacio de direcciones para 48KB de ROM, el cual reside en la parte alta del espacio de direcciones de un megabyte, este tampoco puede ser direccionado por el bus del sistema. El procesador, por medio del uso de las instrucciones IN y OUT, puede direccionar hasta 64K direcciones de puertos E/S. Esas direcciones a puerto se presentan cuando las líneas A0-A15 del bus de direcciones son activadas. Las líneas A16 a A19 se mantienen inactivas durante los ciclos de bus a puerto E/S. Sin embargo, en la computadora, sólo las líneas de dirección A0 a A9 son usadas para direccionar puertos E/S. Además, sólo direcciones de puertos E/S en el rango 0200HEX a 03FFHEX son válidas en el bus del sistema.

D0 a D7

Se trata de ocho líneas bidireccionales usadas para transmitir datos entre el microprocesador 8088, la memoria y los puertos E/S (D0 es el bit menos significativo y D7 el más significativo). Durante los ciclos de escritura iniciados por el 8088 los datos están presentes sobre el bus para enviarse a memoria o puertos E/S. Los flancos de subida de las señales IOW ó MEMW usualmente se usan para sincronizar los datos del bus de datos en la memoria o E/S. Cuando el 8088 inicia el ciclo de lectura la memoria o puerto E/S direccionado, debe poner sus datos sobre el bus de datos antes del flanco de subida de las señales de control IOR o MEMR. Durante los ciclos de acceso directo a memoria el bus de datos se usa para transferir datos directamente entre un puerto E/S y memoria sin la intervención del microprocesador. Durante los ciclos DMA, el procesador se desconecta del bus y del dispositivo controlador del acceso directo a memoria, el bus de datos se usa para transferir datos directamente entre un puerto E/S y memoria sin la intervención del microprocesador. Durante los ciclos DMA, el procesador se desconecta del bus y el dispositivo controlador del acceso directo a memoria, 8237-5, controla la transferencia en el bus.

IRQ2 - IRQ7 (solicitudes de interrupción 2 a 7)

Son seis señales de entrada unicamente, usadas para generar solicitudes de interrupción al microprocesador 8088 desde el bus del sistema. Esas señales van directamente al controlador de interrupciones 8259A en la tarjeta del procesador. Los programas BIOS de la ROM programan al controlador 8259A de tal forma que IRQ2 tiene la prioridad más alta y IRQ7 la menor. Si el nivel no es enmascarado, el flanco de subida de una señal generará una solicitud de interrupción al microprocesador 8088. Una vez que el flanco de subida ocurre, debe permanecer activo hasta que el

procesador 8088 envía una señal INTA (interrupt acknowledge). La señal INTA no está disponible en el bus del sistema, por lo que la solicitud de interrupción es usualmente apagada (reset) con un bit de registro de puerto E/S, usando el comando OUT en la rutina de servicio a la interrupción. Si la solicitud de interrupción no es mantenida activa hasta que aparezca la señal INTA, una interrupción nivel 7 es generada sin importar la prioridad del nivel presentado. Se debe hacer notar que el carácter de esas líneas puede cambiarse reprogramando el parámetro de iniciación del controlador 8259A.

IOR (lectura E/S)

Esta es una señal exclusivamente de salida, proveniente del controlador de bus 8288. Se usa para indicar a los puertos E/S que el ciclo de bus presente, iniciado por el 8088, es un ciclo de lectura de puerto E/S y que la dirección en el bus de direcciones es una dirección a un puerto E/S. El puerto E/S direccionado debe responder colocando el dato a leer sobre el bus de datos del sistema. Esta señal es activa baja y el puerto E/S debe poner su dato en el bus aproximadamente 30 ns antes del flanco de subida de -IOR, para asegurar que el procesador recibe datos válidos. Cuando ocurre un ciclo de acceso directo, la señal IOR se maneja desde el controlador DMA (8237-5) en la tarjeta del procesador. En este caso, el bus de direcciones no contiene una dirección de puerto E/S, en su lugar, contiene la dirección de memoria donde debe escribirse el dato proveniente del puerto. El puerto E/S no se selecciona por una dirección pero sí por la señal activa DACK del controlador DMA.

IOW (escritura E/S)

Esta señal es de salida únicamente, activa con nivel bajo, y se maneja desde el controlador de bus 8288, durante un ciclo de bus iniciado por el 8088, e indica que el bus de direcciones contiene una dirección de puerto E/S y en el bus de datos se encuentra el dato a ser escrito en el puerto E/S. Cuando se activa la señal, el dato en el bus puede ser válido, por lo que, el dato en el puerto debe tomarse con el flanco de subida de esta señal. Cuando ocurre un ciclo DMA, esta señal se maneja desde el controlador DMA (8237-5). La señal -IOW se usa entonces para escribir datos de memoria, los cuales se encuentran en el bus de datos, al puerto E/S seleccionado por DACK. De nuevo, como en el caso de -IOR, el dato puede no ser válido durante el flanco frontal de la

señal activa baja y debe amarrarse en el puerto usando el flanco de subida de esta señal.

AEN (dirección válida)

Esta es una señal activa alta, unidireccional, de salida, enviada por la lógica de control del DMA. Esta señal es usada para inhabilitar los buses de dirección, datos y control del microprocesador 8088, del bus del sistema, y habilitar los buses de dirección y control del controlador DMA. En el bus del sistema, su propósito es inhibir la decodificación de direcciones a puertos E/S durante ciclos DMA para que las direcciones de memoria no se usen como direcciones a puertos E/S durante esos ciclos. Esto es posible ya que -IOW e -IOR pueden estar activas con direcciones de memoria en el bus de direcciones durante ciclos DMA.

Alimentación de energía en el Bus y Tierra.

Además de las señal discutidas previamente, el sistema contiene los siguientes niveles de energía.

* +5 V DC (5 volts de corriente directa).

Está disponible en dos terminales del conector de tarjeta en el bus y está regulado a $\pm 5\%$ (+4.75 a +5.25 volts cd).

* +12 V DC (12 volts de corriente directa).

Esta alimentación de energía se encuentra disponible en una terminal del conector de tarjeta en el bus y está regulado a $\pm 5\%$ (+11.4 a +12.6 volts cd).

* -12 V DC (12 volts de corriente directa).

También se encuentra disponible en una terminal del conector de tarjeta en el bus y está regulado a $\pm 10\%$ (-10.8 a -13.2 volts cd).

* GND (tierra)

La señal de tierra del sistema y de la alimentación se suministra en tres terminales del conector de tarjeta por inserción.

Diagramas de Tiempo del Bus del Sistema

La clave para diseñar cualquier interfaz conectada al bus del sistema es entender su compatibilidad de sincronía

con el bus del sistema. Por ejemplo, es necesario saber qué tan rápido deben estar los datos presentes después de que la memoria o registro es accedido por el microprocesador 8088. La información presentada se refiere sólo a los ciclos utilizados en la tarjeta A/D, y se trata de los tiempos mínimo y máximo para sincronía de señales.

Tablas y diagramas de sincronía

Las figuras 3 y 4 ilustran las relaciones de sincronía de las señales para lectura y escritura en puertos E/S respectivamente. Las tablas B.1 y B.2 contienen los tiempos relativos a las cartas de sincronía para los ciclos de lectura y escritura respectivamente, iniciados por el 8088.

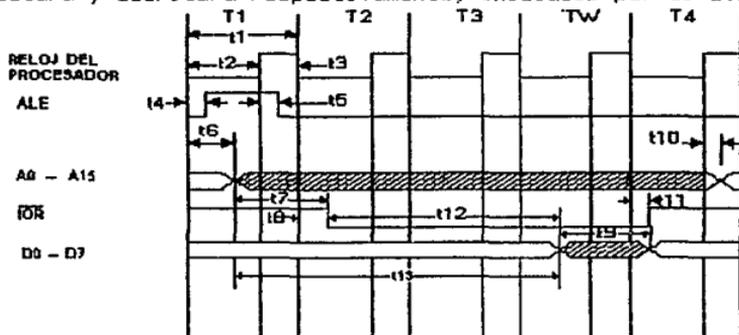


Fig.3 Sincronía del ciclo de lectura en puerto E/S

SÍMBOLO	MAX	MIN	
t1	-	209.5	
t2	-	124.5	
t3	-	71.8	
t4	15	-	
t5	15	-	
t6	128	16	(Tiempo en ns)
t7	-	91.5	
t8	35	10	
t9	-	42	
t10	-	10	
t11	35	10	
t12	-	551.5	
t13	-	668	

Tabla B.1 Tiempos de sincronía en un ciclo de lectura E/S

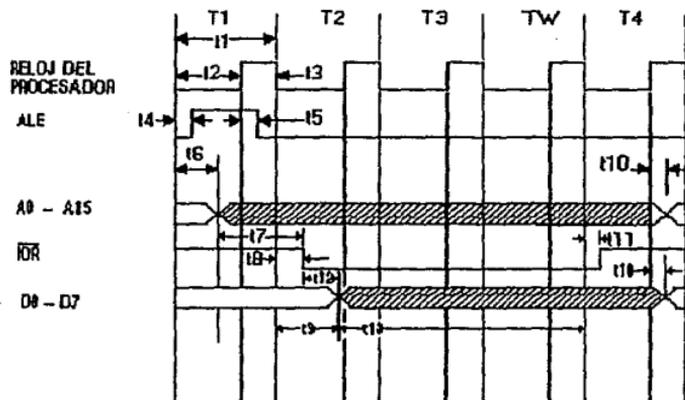


Fig.4 Sincronía del ciclo de escritura E/S

SIMBOLO	MAX	MIN	
t1	-	209.5	
t2	-	124.5	
t3	-	71.8	
t4	15	-	
t5	15	-	
t6	128	16	(Tiempo en ns)
t7	-	91.5	
t8	35	10	
t9	122	14	
t10	-	10	
t11	35	10	
t12	112	-	
t13	-	506.5	

Tabla B.2 Tiempos de sincronía en un ciclo de escritura E/S

Interrupciones en el Sistema

Un total de nueve niveles de solicitud de interrupción están disponibles en la computadora personal (PC). Sin embargo, no todos esos niveles están disponibles para aplicaciones de interfaz. Algunos son usados por adaptadores integrados de E/S del sistema y otros por adaptadores

conectados al bus del sistema. La tabla B.3 presenta un resumen de los niveles de interrupción y su uso actual.

Nivel alto NMI		Verifica el canal I/O, procesador numérico
	IRQ 3	Salida temporizador 8253-5 canal 0
	IRQ 1	Código de interrupción
Disponibles en el Bus	IRQ 2	No se usa
	IRQ 3	No se usa
	IRQ 4	Puerto serial RS-232-C
	IRQ 5	No se usa
	IRQ 6	Estado del diskette DRV
	IRQ 7	Puerto paralelo PRT (no usado en BIOS)

Tabla B.3 Sumario de los niveles de interrupción

Secuencia de Eventos en una Interrupción

Cuando ocurre una solicitud de interrupción en el sistema, una secuencia de eventos dirige la solicitud al programa apropiado, necesario para darle servicio. Antes de que pueda ocurrir esta secuencia de eventos, es necesario una iniciación del sistema, de tal forma que las solicitudes se manejen apropiadamente cuando ocurran. La secuencia de eventos que ocurre cuando una interrupción se activa es la siguiente, (Se asume que la secuencia de habilitación de interrupciones ya ha sido hecha y que la solicitud de interrupción no está enmascarada):

1. La lógica de la interfaz hace una solicitud de interrupción que se presenta en el bus del sistema.
2. El controlador de interrupciones (8259A) recibe la solicitud y compara su prioridad con otras peticiones que pueden llegar o estén pendientes.
3. Si la solicitud es única, o le corresponde el nivel superior siguiente al nivel de la interrupción que se esté atendiendo, al terminar la atención a la rutina de mayor nivel, se hace una solicitud de interrupción que se envía al microprocesador.
4. El MPU 8088 envía dos pulsos INTA de respuesta al controlador de interrupciones 8259A. El primero,

congela la prioridad y carga los niveles en el registro de servicio del 8259A; el segundo, solicita el valor de un apuntador de 8 bits.

5. El procesador 8088 recibe el valor del apuntador de 8 bits. Este se usa para indexar dentro de una tabla de memoria baja, la cual contiene al IP (apuntador de interrupciones) y el valor del corrimiento de la rutina de servicio a la interrupción para el nivel específico que está siendo atendido.
6. El microprocesador 8088 recoge el IP (*Instruction Pointer*) y el valor del segmento de código, pone en el stack el IP actual, el código de segmento, y las banderas y, entonces, salta el IP recién traído junto con su segmento de código. El programa de servicio a la interrupción empieza su ejecución.

La figura 5 presenta un diagrama de bloques de la señal y flujo de datos durante una interrupción al sistema.

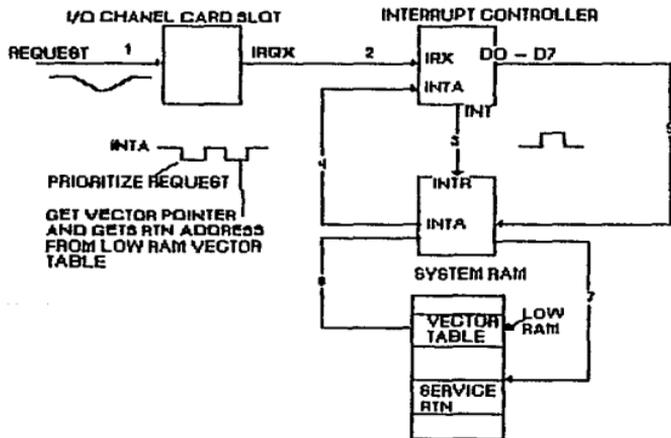


Fig.5 Flujo de señales en una interrupción

APENDICE 3

PROCEDIMIENTO DE AJUSTE

Para realizar los ajustes se requiere que la tarjeta de conversión esté polarizada, ya sea dentro de la PC o fuera de ella.

Voltaje Regulado

Se coloca un voltímetro en la terminal 2 (V+) del CI7 (LM317), con el potenciómetro R25 se ajusta el voltaje a +6.5 V. Para el voltaje negativo se coloca el voltímetro en la terminal 3 (V-) del CI8 LM 337 y con R26 se ajusta el voltaje a -6.5V.

Voltaje de Corriente (*offset*) del amplificador de instrumentación.

En la terminal 1 de CI5A se mide el voltaje de salida del amplificador de instrumentación. Se programa un modo de lectura diferencial (SECC 3.2.2) con ganancia 10, se direcciona el canal 1. El potenciómetro R8 se ajusta para que el voltaje leído sea 0.000V.

Ganancias del Amplificador

Para este procedimiento de ajuste se usa un generador de ondas senoidal y un osciloscopio. Se alimenta al sistema una señal senoidal de 1 KHz, de frecuencia. La lectura de la señal amplificada se hace conectando el osciloscopio en la terminal 1 de CI5A.

El ajuste se hace conforme a la siguiente tabla.

Ganancia (Vpp)	Señal de entrada (vPP)	Canal	Modo	Resistencia de ajuste	Lectura
0.5	1	1	Simple	R17	0.5
1	1	1	Simple	R18	1
2	1	1	Simple	R19	2
4	1	1	Simple	R20	4
8	1	1	Simple	R21	8
10	0.5	1,9	Diferencial	R22	5
50	0.1	1,9	Diferencial	R23	5
100	0.05	1,9	Diferencial	R24	5

Ajuste del cero en el convertidor A/D

Programar el sistema con una ganancia unitaria y modo de entrada simple.

En el canal se alimenta un voltaje de DC de 0.61 mV de amplitud. Con el potenciómetro R-35 se ajusta V_{OS} tal que la lectura obtenida oscile entre 0 y 1.2 mV.

Ajuste de escala completa

Con ganancia unitaria y modo simple se alimenta al sistema una señal de D.C. por el canal 1. Aplique la entrada a un voltaje de 4.9982 V, ahora ajuste V_{ref} con R32 tal que la salida del convertidor oscile entre 0,111,111, 111,110 (4.998) y 0,111,111,111,111 (5.000).

Cambie el voltaje de entrada a un voltaje negativo cercano a -5V tal que la lectura oscile entre 1,000,000,000,000 y 1,000,000,000,001; lea el voltaje de entrada y llámelo V_x . Este voltaje debería ser $(-V_f + V_f/8192)$. V_f es el voltaje de escala completa igual a 5.000.

Calcule:

$$A = V_x - (-V_f - V_f / 8192)$$

Ahora aplique el voltaje $V_x=A/2$ y ajuste V_{ref} tal que la salida digital oscile entre 1,000,000,000,000 y 1,000,000,000,001.

APENDICE 4

LISTA DE COMPONENTES

CIRCUITOS INTEGRADOS

CI0	4050	CI31	74LS14
CI1	4051	CI32	74LS14
CI2	4052	CI33	74LS244
CI3	TL082	CI34	7489
CI4	TL082	CI35	7489
CI5	TL082	CI36	7489
CI6	4051	CI37	74LS240
CI7	LM317	CI38	74LS240
CI8	LM337	CI39	74LS240
CI9	ADC1205	CI40	DAC1230
CI10	TL072	CI41	TL082
CI11	74LS74	CI42	LF355N
CI12	74LS04	CI43	LF355N
CI13	8253	CI44	74LS155
CI14	74LS75	CI45	74LS73
CI15	74LS74	CI46	74LS73
CI16	74LS02	CI47	74LS04
CI17	74LS02	CI48	74LS00
CI18	74LS00	CI49	74LS04
CI19	74LS245	CI50	74LS00
CI20	74LS244	CI51	74LS00
CI21	74LS244	CI52	74LS02
CI22	74LS75	CI53	74LS02
CI23	74LS85	CI54	74LS85
CI24	74LS138	CI55	74LS244
CI25	74LS123	CI56	74LS244
CI26	8255	CI57	74LS193
CI27	8253	CI58	8051
CI28	74LS14		
CI29	74LS14		
CI30	74LS14		

RESISTENCIAS

R1, R2, R3, R4, R5 10 K OHMS.

R6	100.0 K	OHMS	R48	4.7 K	OHMS
R7	15.0 K	OHMS	R49	4.7 K	OHMS
R8	50.0 K	OHMS	R50	4.7 K	OHMS
R9	4.7 M	OHMS	R51	4.7 K	OHMS
R10	1.0 M	OHMS	R52	4.7 K	OHMS
R11	84.0 K	OHMS	R53	4.7 K	OHMS
R12	60.0 K	OHMS	R54	4.7 K	OHMS
R13	39.0 K	OHMS	R55	4.7 K	OHMS
R14	18.0 K	OHMS	R56	15.0 K	OHMS
R15	8.2 K	OHMS	R57	15.0 K	OHMS
R16	4.7 K	OHMS	R58	15.0 K	OHMS
R17	1.0 K	OHMS	R59	6.8 K	OHMS
R18	2.0 K	OHMS	R60	6.8 K	OHMS
R19	5.0 K	OHMS	R61	6.8 K	OHMS
R20	5.0 K	OHMS	R62	6.8 K	OHMS
R21	20.0 K	OHMS	R63	6.8 K	OHMS
R22	20.0 K	OHMS	R64	6.8 K	OHMS
R23	500.0 K	OHMS	R65	6.8 K	OHMS
R24	1.0 M	OHMS	R67	6.8 K	OHMS
R25	1.0 K	OHMS	R68	6.8 K	OHMS
R26	1.0 K	OHMS	R69	6.8 K	OHMS
R27	220.0	OHMS	R70	1.0 K	OHMS
R28	680.0	OHMS	R71	1.0 K	OHMS
R29	12.0 K	OHMS	R72	1.0 K	OHMS
R30	47.0 K	OHMS	R73	1.0 K	OHMS
R31	4.7 K	OHMS	R74	1.0 K	OHMS
R32	5.0 K	OHMS	R75	1.0 K	OHMS
R33	22.0 K	OHMS	R76	1.0 K	OHMS
R34	470.0 K	OHMS	R77	1.0 K	OHMS
R35	10.0 K	OHMS	R78	1.0 K	OHMS
R36	10.0 K	OHMS	R79	4.7 K	OHMS
R37	0.8 K	OHMS	R80	47.0	OHMS
R38	0.8 K	OHMS	R81	20.0 K	OHMS
R39	39.0 K	OHMS	R82	1.0 M	OHMS
R40	4.7 K	OHMS	R83	4.7 K	OHMS
R41	5.6 K	OHMS	R84	820.0	OHMS
R42	10.0 K	OHMS	R85	1.0 K	OHMS
R43	10.0 K	OHMS	R86	10.0 K	OHMS
R44	10.0 K	OHMS	R87	10.0 K	OHMS
R45	3.3 K	OHMS	R88	2.7 K	OHMS
R46	4.7 K	OHMS	R89	8.2 K	OHMS
R47	4.7 K	OHMS	R90	56.0 K	OHMS
R91	5.0 K	OHMS			
R92	17.0 K	OHMS			
R93	560.0	OHMS			

R94	8.2 K	OHMS
R95	820.0	OHMS
R96	8.2	OHMS
R97	1.0 K	OHMS
R98	1.0 K	OHMS
R99	1.0 K	OHMS
R100	1.0 K	OHMS
R102	1.0 K	OHMS
R103	1.0 K	OHMS
R104	1.0 K	OHMS
R105	1.0 K	OHMS
R106	1.0 K	OHMS
R107	1.0 K	OHMS
R108	1.0 K	OHMS
R109	4.7 K	OHMS
R110	4.7 K	OHMS
R111	4.7 K	OHMS
R112	4.7 K	OHMS
R113	10.0 K	OHMS
R114	15.0 K	OHMS
R115	15.0 K	OHMS

NOTA: Resistencias de precisión al 1% y 1/2 Watt

CAPACITORES

C1	4.0	pf
C2	150.0	nf
C3	0.1	µf
C4	0.1	µf
C5	0.1	µf
C6	10.0	µf
C7	1.0	µf
C8	1.0	µf
C9	10.0	µf
C10	0.1	µf
C13	38.0	pf
C14	1.0	nf
C15	1.0	nf
C16	22.0	µf
C17	22.0	µf

NOTA: Capacitores a 35 V.

DIODOS

D1	5.6 V
D2	6.8 V
D3	6.8 V
D4	6.8 V
D5	6.8 V

NOTA: Diodos de 1/2 Watt.

APENDICE 5

PROGRAMA DE CONTROL DE LA TARJETA DE OPCIONES MULTIPLES

PROGRAM SINCRONIA

```

(*N+);
(*F+);
(*R+);

```

```

USES DOS,CRT,PRINTER,TURBO3;
  ( Etiqueta para salir al sistema operativo )
LABEL EXIT;

```

```

  ( ETIQUETA DE REGREBO AL MENU )
LABEL REGRESAR;
LABEL UNO;
LABEL DOS;
LABEL TRES;

```

```

CONST
  INT_A      = *0A;
  INT_B      = *0B;
  IRQ0       = *FE;
  NOT_IRQ0   = *01;
  IRQ2       = *FB;
  NOT_IRQ2   = *04;
  IRQ3       = *F7;
  NOT_IRQ3   = *08;
  IMR_B259   = *21;

```

```

TYPE
  INT_VECTOR = ARRAY [0..15] OF INTEGER;
  ITEMPONTER = ^INTEGER;

```

```

VAR
  VECTORB, VECTORA           :
  POINTER;
  MUESTRA, MUESTRA_B        :
  INT_VECTOR;
  FRECUENCIA, FREQ_MAX, FRECUENCIA_REAL, NUM_SEG_R,
  TIEMPO_MAX, TIEMPO_MUESTREO           : REAL;
  CANALA, CUENTA, COORX, COORY, DIV_FRECUENCIA, GANANCIA,
  GANAN_MODO, I, K, MODO, NUM_CANALES, NUM_SEG_I,
  NUM_MUESTRAS, POR_SEG_POR_CANAL_NUM_MUESTRAS,
  BUFFER, OFFSET, SECUENCIA, SEGMENTO, VALIDACION :
  INTEGER;
  APUNTADOR                 :
  ITEMPONTER;
  BUF : ARRAY [1..10240] OF INTEGER;
  NOMBRE                       : STRING
[20];
  FTEXT                       : TEXT;

```

```

  ( VARIABLES PARA INICIAR GRAFICAS )

```

```

OPCION, OPCION2 : STRING [10] ;
DIGITALE : ARRAY [1..8] OF REAL ;
DIGITAL : ARRAY [1..16] OF STRING [10] ;
GITAL : ARRAY [1..16] OF INTEGER ;
Y : ARRAY [1..16] OF INTEGER ;
X : ARRAY [1..10] OF INTEGER ;
GD, GM, A, VARI
INTEGER ;
PUERTON, VARI2, VARI3
INTEGER ;
N, M
INTEGER ;
OPCIONA : CHAR ;
OPCION2A : CHAR ;

( VARIABLES PARA LA LECTURA DE CADA CANAL )
DATO0, DATO1
INTEGER ;

( VARIABLES QUE SE ASIGNAN A CADA UNO DE LOS PUERTOS )
PUERTO0, PUERTO1, PUERTO2, PUERTO3, PUERTO4
INTEGER ;
PUERTO5, PUERTO6, PUERTO7, PUERTO8, PUERTO9
INTEGER ;
PUERTO10, PUERTO11, PUERTO12, PUERTO13, PUERTO14
INTEGER ;
PUERTO15, PUERTOR1, DDATORR
INTEGER ;

( VABLES. DE AYUDA EN EL PROGRAMA PRINCIPAL )
PUERTO, DATO, PUERTOR, VREF
INTEGER ;

( VABLES. PARA LA ESCRITURA EN FORMA ANALOGICA )
VOLO, VOL1, VOLTAJEI, VOLTAJE
INTEGER ;

PCONT1, PCONT2, PCONT3
INTEGER ;
GITAL1, GITAL2, GITAL3, GITAL4, GITAL5, GITAL6
INTEGER ;
GITAL7, GITAL8, GITAL9, GITAL10, GITAL11
INTEGER ;
GITAL12, GITAL13, GITAL14, GITAL15, GITAL16
INTEGER ;
GITALE1, GITALE2, GITALE3, GITALE4, GITALES
INTEGER ;
GITALE6, GITALE7, GITALE8
INTEGER ;

QDATO1, QDATO2, QPUERA, QPUERB, QPUERC, QDATOP
INTEGER ;

```

```

    QPUERTOP, IPUERTO
INTEGER ;

( VABLES. PARA LOS CONTADORES )
CONT1, CONT2, CONT3, QDATO           : WORD ;
QDATOE1, QDATOE11, QDATO11, QDATOB1, CONT22 : REAL ;
DATOR, DATOOO, DATOO1               : REAL ;
VOLTAJEO, VOLTAJE1                  : REAL ;

( VABLES. PARA LAS SALIDAS DIGITALES )
OPRIME, OPF                          : CHAR ;
DIGITAL1, DIGITAL2, DIGITAL3, DIGITAL4, DIGITAL5 : CHAR ;
DIGITAL6, DIGITAL7, DIGITAL8, DIGITAL9 : CHAR ;
DIGITAL10, DIGITAL11, DIGITAL12, DIGITAL13 : CHAR ;
DIGITAL14, DIGITAL15, DIGITAL16 : CHAR ;

( VABLES. PARA LAS ENTRADAS DIGITALES )
DIGITALE1, DIGITALE2, DIGITALE3, DIGITALE4 : CHAR ;
DIGITALE5, DIGITALE6, DIGITALE7, DIGITALE8 : CHAR ;

( VABLES. PARA CONTINUAR DESPUES DE VER LA PRESENTACION )
D, CONTP1, CONTP2, CONTP3, OPCIONIA : CHAR ;
OPCION1 : STRING

[10] ;
    CANAL, QCONT, LPUERTO, LPUERTO : CHAR ;
    CONTINUAR, QPOT, QPUERTO, LPUERTO : CHAR ;
    OPC : STRING

[10];

PROCEDURE PRESENTACION ;
BEGIN
    CLRSCR;
    GOTOXY (16,2); WRITELN
('*****');
    GOTOXY (16,3); WRITELN ('*
*');
    GOTOXY (16,4); WRITELN ('*          TARJETA DE FUNCIONES
MULTIPLES *');
    GOTOXY (16,5); WRITELN ('*
*');
    GOTOXY (16,6); WRITELN ('*          DIVISION DE EDUCACION
CONTINUA DE *');
    GOTOXY (16,7); WRITELN ('*          LA FACULTAD DE INGENIERIA DE
LA UNAM *');
    GOTOXY (16,8); WRITELN ('*
*');
    GOTOXY (16,9); WRITELN ('*          PROGRAMA DE APOYO A LA
TITULACION *');
    GOTOXY (16,10); WRITELN ('*
*');
    GOTOXY (16,11); WRITELN
('*****');

```

```

REPEAT UNTIL KEYPRESSED;
CLRSCR;
END;

```

```

PROCEDURE MENUP;
BEGIN

```

```

  CLRSCR;
  GOTOXY (15,4); Writeln
('*****');
  GOTOXY (15,5); Writeln ('*
*');
  GOTOXY (15,6); Writeln ('*
*')
                                MENU PRINCIPAL
  GOTOXY (15,7); Writeln ('*
*');
  GOTOXY (15,8); Writeln ('*
*')
                                TECLEE LA OPCION DESEADA
  GOTOXY (15,9); Writeln ('*
*')
                                PARA PROGRAMAR LA TFM
  GOTOXY (15,10); Writeln ('*
*');
  GOTOXY (15,11); Writeln ('*
*')
                                CANALES ANALOGICOS
  ..... A.
  GOTOXY (15,12); Writeln ('*
*')
                                CANALES DIGITALES
  ..... B.
  GOTOXY (15,13); Writeln ('*
*')
                                SALIR A DOS
  ..... C.
  GOTOXY (15,14); Writeln ('*
*');
  GOTOXY (15,15); Writeln
('*****');
END;

```

```

PROCEDURE ANALOGICOS;
( ESTA SUBROUTINA PRESENTA EL MENU DE OPCIONES PARA SEÑALES
ANALOGICAS )
BEGIN

```

```

  CLRSCR;
  GOTOXY (15,2); Writeln
('*****');
  GOTOXY (15,3); Writeln ('*
*');
  GOTOXY (15,4); Writeln ('*
*')
                                MENU PARA SEÑALES
ANALOGICAS
  GOTOXY (15,5); Writeln ('*
*');
  GOTOXY (15,6); Writeln ('*
*')
                                TECLEE LA OPCION
DESEADA
  GOTOXY (15,7); Writeln ('*
*');
  GOTOXY (15,8); Writeln ('*
*')
                                PROGRAMAR ENTRADAS
ANALOGICAS ... A.

```

```

    GOTOXY (15,9); WRITELN ('*          PROGRAMAR SALIDAS ANALOGICAS
... B.   *');
    GOTOXY (15,10); WRITELN ('*          REGRESAR AL MENU PRINCIPAL
..... C. *');
    GOTOXY (15,11); WRITELN ('*
*');
    GOTOXY (15,12); WRITELN
('*****');
END;

```

```

PROCEDURE DIGITALES;
( ESTA SUBRRUTINA PRESENTA EL MENU DE OPCIONES PARA SEÑALES
DIGITALES )
BEGIN

```

```

    CLRSCR;
    GOTOXY (15,2); WRITELN
('*****');
    GOTOXY (15,3); WRITELN ('*
*');
    GOTOXY (15,4); WRITELN ('*          MENU PARA SEÑALES
DIGITALES *');
    GOTOXY (15,5); WRITELN ('*
*');
    GOTOXY (15,6); WRITELN ('*          TECLEE LA OPCION
DESEADA *');
    GOTOXY (15,7); WRITELN ('*
*');
    GOTOXY (15,8); WRITELN ('*          VER EL ESTADO DEL CONTADOR
*');
    GOTOXY (15,9); WRITELN ('*          CONTADOR 1
..... A. *');
    GOTOXY (15,10); WRITELN ('*          CONTADOR 2
..... B. *');
    GOTOXY (15,11); WRITELN ('*          CONTADOR 3
..... C. *');
    GOTOXY (15,12); WRITELN ('*          LEER PUERTO C
..... D. *');
    GOTOXY (15,13); WRITELN ('*          ESCRIBIR AL PUERTO A
..... E. *');
    GOTOXY (15,14); WRITELN ('*          ESCRIBIR AL PUERTO B
..... F. *');
    GOTOXY (15,15); WRITELN ('*          MENU PRINCIPAL
..... G. *');
    GOTOXY (15,16); WRITELN ('*
*');
    GOTOXY (15,17); WRITELN
('*****');
END;

```

```

PROCEDURE MENSAJE;
BEGIN
    CLRSCR;

```

```

GOTOXY (20,21); WRITELN('  LOS DATOS DEBEN ESTAR
COMPRENDIDOS ');
GOTOXY (20,23); WRITELN('  ENTRE -5.0 V. Y 4.9975 V.
');
WRITELN(CHR(7));
DELAY(2000);
END;

```

```

PROCEDURE ESCRITURA;
( ESTE PROCEDIMIENTO ESCRIBE UN VOLTAJE EN UNO DE LOS 2
CANALES )
( ANALOGICOS DE SALIDA, EL VOLTAJE DEBE ESTAR EN EL RANGO
DE
-5.0 V. Y 4.9975 V., EN CASO CONTRARIO MANDARA UN
MENSAJE )
( Y SE QUEDARA ESPERANDO UN NUEVO DATO.
)
BEGIN
REPEAT
CLRSCR;
(*I-)
REPEAT
GOTOXY(41,10); WRITELN('  :10);
GOTOXY(15,10); WRITELN(' EL CANAL A PROGRAMAR ES : ');
GOTOXY(41,10); READLN(PUERTO);
IPUERTO := IORESULT;
IF (( IPUERTO <> 0 ) OR ( PUERTO > 2 )) THEN
BEGIN
GOTOXY(15,12); WRITELN(' EL DATO NO ES CORRECTO
');
GOTOXY(15,13); WRITELN(' SOLO PUEDE SER 1 O 2
');
DELAY(1000);
GOTOXY(15,12); WRITELN('  :80);
END;
UNTIL (( IPUERTO = 0 ) AND ( PUERTO < 3 ));
(*I+)
(*I-)
REPEAT
GOTOXY(15,5); WRITELN(' EL VOLTAJE A PROGRAMAR ES ');
GOTOXY(44,15); READLN(VOLTAJE);
VOLTAJEI := IORESULT;
IF ( VOLTAJEI <> 0 ) THEN MENSAJE;
IF (( VOLTAJE < -5.0 ) AND ( VOLTAJE > 4.9975 )) THEN
MENSAJE;
UNTIL (( VOLTAJEI = 0 ) AND ( VOLTAJE >= -5.00 ) AND
( VOLTAJE <= 4.9975 ));
(*I+)
DATOO := ( 2048 * (VOLTAJE / VREF));
DATO := ( 2048 + TRUNC (DATOO));
IF ( PUERTO = 1 ) THEN PORTW [#300] := DATO;
IF ( PUERTO = 2 ) THEN PORTW [#302] := DATO;

```

```

GOTOXY(15,17); WRITELN(' DESEAS PROGRAMAR OTRO CANAL
(S/N)');
GOTOXY(49,17); REDLN (CANAL);
UNTIL (( CANAL = 'N' ) OR ( CANAL = 'n' ));
GOTO DOSS;
END;

```

```

PROCEDURE CONTADOR1;
( PROC. QUE PERMITE CONOCER EL EDO. DEL CONTADOR 1 )
BEGIN
VARI := 100;
CLRSCR;
REPEAT
CONT1 := PORT[#328];
CONT1 := ( PORT[#328] SHL 8 ) OR CONT1;
CONT1 := NOT(CONT1);
GOTOXY(15,15);
WRITELN ( ' EL ESTADO DEL CONTADOR 1 ES : ',CONT1);
IF ( CONT1 = VARI ) THEN VARI := 100 + VARI;
IF ( CONT1 < 35000 ) THEN PORTW [#300] := #0000 ;
IF ( CONT1 > 35000 ) THEN PORTW [#300] := #0FFF ;
UNTIL KEYPRESSED;
END;

```

```

PROCEDURE CONTADOR2;
( PROC. QUE PERMITE CONOCER EL EDO. DEL CONTADOR 2 )
BEGIN
CLRSCR;
REPEAT
CONT2 := PORT [#329];
CONT2 := ( PORT[#329] SHL 8 ) OR CONT2 ;
CONT2 := NOT(CONT2);
GOTOXY(15,15);
CONT2 := TRUNC( CONT2/10 );
WRITELN(' EL ESTADO DEL CONTADOR 2 ES : ',CONT2);
IF ( CONT2 < 40960 ) THEN PORTW [#302] := (CONT2);
UNTIL KEYPRESSED;
END;

```

```

PROCEDURE CONTADOR3;
( PROC. QUE PERMITE CONOCER EL EDO. DEL CONTADOR 3 )
BEGIN
CLRSCR;
REPEAT
CONT3 := PORT [#32A];
CONT3 := (PORT [#32A] SHL 8) OR CONT3 ;
CONT3 := NOT(CONT3);
GOTOXY(15,15);
WRITELN ( ' EL ESTADO DEL CONTADOR 3 ES : ',CONT3);
CON22 := 65535 * SIN(CONT3);
WRITELN (CONT22);
CONT3 := TRUNC (CONT22);

```

```

CONT3 := 4 * CONT3;
WRITELN (CONT3);
IF ( CONT3 < 65000 ) THEN PORTW [#304] := (CONT3);
UNTIL KEYPRESSED;
END;

```

```

PROCEDURE MENSAJE1;
BEGIN
GOTOXY(5,25); WRITELN(' EL DATO NO ES VALIDO ');
DELAY(500);
END;

```

```

PROCEDURE CONTEO;
BEGIN
QDATO := ((1* GITAL[1]) + (2* GITAL[2]) + (4* GITAL[3])
+ (8* GITAL[4]) + (16* GITAL[5]) + (32* GITAL[6])
+ (64* GITAL[7]) + (128* GITAL[8]));
PORTE[#320] := QDATO;
END;

```

```

PROCEDURE PUERTO A;
BEGIN
CLRSCR;
GOTOXY(20,2);
WRITELN(' ACTIVO = 1 ',' :7,' DEBACTIVO = 0');
GOTOXY(16,21);
WRITELN(' SE PROGRAMA ESTOS CANALES CUANDO SE ASIGNA ');
GOTOXY(21,22);
WRITELN(' LOS PRIMEROS 8 CANALES DIGITALES ');
BEGIN
REPEAT
GOTOXY(54,5); WRITELN(' ':10);
GOTOXY(15,5);
WRITELN(' CANAL DIGITAL 1 (1/0)
..... ');
GOTOXY(54,5); READLN( DIGITAL[1]);
Y[1] := LENGTH (DIGITAL[1]);
IF ( DIGITAL[1] = '1' ) THEN GITAL[1] := 1;
IF ( DIGITAL[1] = '0' ) THEN GITAL[1] := 0;
UNTIL (( Y[1] = 1 ) AND ( DIGITAL[1] = '1' )
OR ( DIGITAL[1] = '0' ));
CONTEO;
END;
BEGIN
REPEAT
GOTOXY(54,7); WRITELN(' ':10);
GOTOXY(15,7);
WRITELN(' CANAL DIGITAL 2 (1/0)
..... ');
GOTOXY(54,7); READLN( DIGITAL[2]);
Y[2] := LENGTH (DIGITAL[2]);
IF ( DIGITAL[2] = '1' ) THEN GITAL[2] := 1;

```

```

        IF ( DIGITAL[2] = '0' ) THEN BITAL[2] := 0;
    UNTIL (( Y[2] = 1 ) AND ( DIGITAL[2] = '1' )
    OR ( DIGITAL[2] = '0' ));
    CONTEQ;
END;
BEGIN
    REPEAT
        GOTOXY(54,9); WRITELN(' ',10);
        GOTOXY(15,9);
        WRITELN (' CANAL DIGITAL 3 (1/0)
        ..... ');
        GOTOXY(54,9); READLN( DIGITAL[3]);
        Y[3] := LENGTH (DIGITAL[3]);
        IF ( DIGITAL[3] = '1' ) THEN BITAL[3] := 1;
        IF ( DIGITAL[3] = '0' ) THEN BITAL[3] := 0;
        UNTIL (( Y[3] = 1 ) AND ( DIGITAL[3] = '1' )
        OR ( DIGITAL[3] = '0' ));
        CONTEQ;
    END;
    BEGIN
        REPEAT
            GOTOXY(54,11); WRITELN(' ',10);
            GOTOXY(15,11);
            WRITELN (' CANAL DIGITAL 4 (1/0)
            ..... ');
            GOTOXY(54,11); READLN( DIGITAL[4]);
            Y[4] := LENGTH (DIGITAL[4]);
            IF ( DIGITAL[4] = '1' ) THEN BITAL[4] := 1;
            IF ( DIGITAL[4] = '0' ) THEN BITAL[4] := 0;
            UNTIL (( Y[4] = 1 ) AND ( DIGITAL[4] = '1' )
            OR ( DIGITAL[4] = '0' ));
            CONTEQ;
        END;
        BEGIN
            REPEAT
                GOTOXY(54,13); WRITELN(' ',10);
                GOTOXY(15,13);
                WRITELN (' CANAL DIGITAL 5 (1/0)
                ..... ');
                GOTOXY(54,13); READLN( DIGITAL[5]);
                Y[5] := LENGTH (DIGITAL[5]);
                IF ( DIGITAL[5] = '1' ) THEN BITAL[5] := 1;
                IF ( DIGITAL[5] = '0' ) THEN BITAL[5] := 0;
                UNTIL (( Y[5] = 1 ) AND ( DIGITAL[5] = '1' )
                OR ( DIGITAL[5] = '0' ));
                CONTEQ;
            END;
            BEGIN
                REPEAT
                    GOTOXY(54,15); WRITELN(' ',10);
                    GOTOXY(15,15);

```

```

WRITELN (' CANAL DIGITAL 6 (1/0)
..... ');
GOTOXY(54,15); READLN( DIGITAL[6]);
Y[6] := LENGTH (DIGITAL[6]);
IF ( DIGITAL[6] = '1' ) THEN GITAL[6] := 1;
IF ( DIGITAL[6] = '0' ) THEN GITAL[6] := 0;
UNTIL (( Y[6] = 1 ) AND ( DIGITAL[6] = '1' )
OR ( DIGITAL[6] = '0' ));
CONTEO;
END;
BEGIN
REPEAT
GOTOXY(54,17); WRITELN(' ':10);
GOTOXY(15,17);
WRITELN (' CANAL DIGITAL 7 (1/0)
..... ');
GOTOXY(54,17); READLN( DIGITAL[7]);
Y[7] := LENGTH (DIGITAL[7]);
IF ( DIGITAL[7] = '1' ) THEN GITAL[7] := 1;
IF ( DIGITAL[7] = '0' ) THEN GITAL[7] := 0;
UNTIL (( Y[7] = 1 ) AND ( DIGITAL[7] = '1' )
OR ( DIGITAL[7] = '0' ));
CONTEO;
END;
BEGIN
REPEAT
GOTOXY(54,19); WRITELN(' ':10);
GOTOXY(15,19);
WRITELN (' CANAL DIGITAL 8 (1/0)
..... ');
GOTOXY(54,19); READLN( DIGITAL[8]);
Y[8] := LENGTH (DIGITAL[8]);
IF ( DIGITAL[8] = '1' ) THEN GITAL[8] := 1;
IF ( DIGITAL[8] = '0' ) THEN GITAL[8] := 0;
UNTIL (( Y[8] = 1 ) AND ( DIGITAL[8] = '1' )
OR ( DIGITAL[8] = '0' ));
CONTEO;
END;
END;
PROCEDURE CONTEO1;
BEGIN
QDATO1 := ((1* GITAL[9]) + (2* GITAL[10]) + (4* GITAL[11])
+ (8* GITAL[12]) + (16* GITAL[13]) + (32*
GITAL[14])
+ (64* GITAL[15] + (128* GITAL[16]));
PORT [#321] := QDATO1 ;
END;
PROCEDURE PUERTO;
BEGIN
CLRSCR;

```

```

GOTOXY(20,2);
WRITELN(' ACTIVO = 1 ', ' :7, ' DEBACTIVO = 0 ');
GOTOXY(16,21);
WRITELN(' SE PROGRAMA ESTOS CANALES CUANDO SE ASIGNAN ');
GOTOXY(21,22);
WRITELN(' LOS SIGUIENTES 8 CANALES DIGITALES ');
BEGIN
  REPEAT
    GOTOXY(54,5); WRITELN(' ':10);
    GOTOXY(15,5);
    WRITELN(' CANAL DIGITAL 9 (1/0)
    ..... ');
    GOTOXY(54,5); READLN( DIGITAL[9]);
    Y[9] := LENGTH (DIGITAL[9]);
    IF ( DIGITAL[9] = '1' ) THEN GITAL[9] := 1;
    IF ( DIGITAL[9] = '0' ) THEN GITAL[9] := 0;
    UNTIL (( Y[9] = 1 ) AND ( DIGITAL[9] = '1' )
    OR ( DIGITAL[9] = '0' ));
    CONTEO1;
  END;
  BEGIN
    REPEAT
      GOTOXY(54,7); WRITELN(' ':10);
      GOTOXY(15,7);
      WRITELN(' CANAL DIGITAL 10 (1/0)
      ..... ');
      GOTOXY(54,7); READLN( DIGITAL[10]);
      Y[10] := LENGTH (DIGITAL[10]);
      IF ( DIGITAL[10] = '1' ) THEN GITAL[10] := 1;
      IF ( DIGITAL[10] = '0' ) THEN GITAL[10] := 0;
      UNTIL (( Y[10] = 1 ) AND ( DIGITAL[10] = '1' )
      OR ( DIGITAL[10] = '0' ));
      CONTEO1;
    END;
    BEGIN
      REPEAT
        GOTOXY(54,9); WRITELN(' ':10);
        GOTOXY(15,9);
        WRITELN(' CANAL DIGITAL 11 (1/0)
        ..... ');
        GOTOXY(54,9); READLN( DIGITAL[11]);
        Y[11] := LENGTH (DIGITAL[11]);
        IF ( DIGITAL[11] = '1' ) THEN GITAL[11] := 1;
        IF ( DIGITAL[11] = '0' ) THEN GITAL[11] := 0;
        UNTIL (( Y[11] = 1 ) AND ( DIGITAL[11] = '1' )
        OR ( DIGITAL[11] = '0' ));
        CONTEO1;
      END;
      BEGIN
        REPEAT
          GOTOXY(54,11); WRITELN(' ':10);
          GOTOXY(15,11);

```

```

WRITELN(' CANAL DIGITAL 12 (1/0)
..... ');
GOTOXY(54,11); READLN( DIGITAL[12]);
Y[12] := LENGTH (DIGITAL[12]);
IF ( DIGITAL[12] = '1' ) THEN GITAL[12] := 1;
IF ( DIGITAL[12] = '0' ) THEN GITAL[12] := 0;
UNTIL (( Y[12] = 1 ) AND ( DIGITAL[12] = '1' )
OR ( DIGITAL[12] = '0' ));
CONTEO1;
END;
BEGIN
REPEAT
GOTOXY(54,13); WRITELN(' ':10);
GOTOXY(15,13);
WRITELN(' CANAL DIGITAL 13 (1/0)
..... ');
GOTOXY(54,13); READLN( DIGITAL[13]);
Y[13] := LENGTH (DIGITAL[13]);
IF ( DIGITAL[13] = '1' ) THEN GITAL[13] := 1;
IF ( DIGITAL[13] = '0' ) THEN GITAL[13] := 0;
UNTIL (( Y[13] = 1 ) AND ( DIGITAL[13] = '1' )
OR ( DIGITAL[13] = '0' ));
CONTEO1;
END;
BEGIN
REPEAT
GOTOXY(54,15); WRITELN(' ':10);
GOTOXY(15,15);
WRITELN(' CANAL DIGITAL 14 (1/0)
..... ');
GOTOXY(54,15); READLN( DIGITAL[14]);
Y[14] := LENGTH (DIGITAL[14]);
IF ( DIGITAL[14] = '1' ) THEN GITAL[14] := 1;
IF ( DIGITAL[14] = '0' ) THEN GITAL[14] := 0;
UNTIL (( Y[14] = 1 ) AND ( DIGITAL[14] = '1' )
OR ( DIGITAL[14] = '0' ));
CONTEO1;
END;
BEGIN
REPEAT
GOTOXY(54,17); WRITELN(' ':10);
GOTOXY(15,17);
WRITELN(' CANAL DIGITAL 15 (1/0)
..... ');
GOTOXY(54,17); READLN( DIGITAL[15]);
Y[15] := LENGTH (DIGITAL[15]);
IF ( DIGITAL[15] = '1' ) THEN GITAL[15] := 1;
IF ( DIGITAL[15] = '0' ) THEN GITAL[15] := 0;
UNTIL (( Y[15] = 1 ) AND ( DIGITAL[15] = '1' )
OR ( DIGITAL[15] = '0' ));
CONTEO1;
END;

```

```

BEGIN
  REPEAT
    GOTOXY(54,19); WRITELN(' ':10);
    GOTOXY(15,19);
    WRITELN(' CANAL DIGITAL 16 (1/0)
    ..... ');
    GOTOXY(54,19); READLN ( DIGITAL[16]);
    Y[16] := LENGTH (DIGITAL[16]);
    IF ( DIGITAL[16] = '1' ) THEN GITAL[16] := 1;
    IF ( DIGITAL[16] = '0' ) THEN GITAL[16] := 0;
    UNTIL (( Y[16] = 1 ) AND ( DIGITAL[16] := '1' )
    OR ( DIGITAL[16] = '0' ));
    CONTEO1;
  END;
END;

PROCEDURE PUERTOCEN;
BEGIN
  N := 1;
  REPEAT
    DIGITALE[N] := 0;
    N := N + 1;
  UNTIL ( N = 9 );
  N := 8;
  QDATOE1 := ( QDATO2 / 2 );
  QDATOE1 := ( QDATOE1 * 2 );
  REPEAT
    QDATOB1 := QDATOE1;
    QDATOE1 := ( QDATOB1 / 2 );
    QDATOE11 := TRUNC (QDATOE1);
    QDATO11 := (QDATOE11 * 2);
    DIGITALE[N] := ( QDATOB1 - QDATO11 );
    QDATOE1 := QDATOE11 ;
    N := N - 1;
    IF (( QDATOB1 = 1 ) OR ( QDATOB1 = 0 ))
      THEN DIGITALE[N] := QDATOB1;
  UNTIL ( QDATOB1 = 1 ) OR ( QDATOB1 = 0 );
END;

PROCEDURE PUERTOC;
BEGIN
  CLRSCR;
  GOTOXY(20,2); WRITELN(' ACTIVO = 1',' ':17,' DESACTIVO =
  0 ');
  PUERTOCEN;

  GOTOXY(54,5); WRITELN(' ':10);
  IF ( DIGITALE[1] = 1 ) THEN DIGITALE1 := '1';
  IF ( DIGITALE[1] = 0 ) THEN DIGITALE1 := '0';
  GOTOXY(15,5);
  WRITELN (' CANAL DIGITAL 1 (1/0) .....
  ', DIGITALE1);

```

```

GOTOXY(54,7); WRITELN(' ':10);
IF ( DIGITALEC2J = 1 ) THEN DIGITALE2 := '1';
IF ( DIGITALEC2J = 0 ) THEN DIGITALE2 := '0';
GOTOXY(15,7);
WRITELN (' CANAL DIGITAL 2 (1/0) .....
',DIGITALE2);

GOTOXY(54,9); WRITELN(' ':10);
IF ( DIGITALEC3J = 1 ) THEN DIGITALE3 := '1';
IF ( DIGITALEC3J = 0 ) THEN DIGITALE3 := '0';
GOTOXY(15,9);
WRITELN (' CANAL DIGITAL 3 (1/0) .....
',DIGITALE3);

GOTOXY(54,11); WRITELN(' ':10);
IF ( DIGITALEC4J = 1 ) THEN DIGITALE4 := '1';
IF ( DIGITALEC4J = 0 ) THEN DIGITALE4 := '0';
GOTOXY(15,11);
WRITELN (' CANAL DIGITAL 4 (1/0) .....
',DIGITALE4);

GOTOXY(54,13); WRITELN(' ':10);
IF ( DIGITALEC5J = 1 ) THEN DIGITALE5 := '1';
IF ( DIGITALEC5J = 0 ) THEN DIGITALE5 := '0';
GOTOXY(15,13);
WRITELN (' CANAL DIGITAL 5 (1/0) .....
',DIGITALE5);

GOTOXY(54,15); WRITELN(' ':10);
IF ( DIGITALEC6J = 1 ) THEN DIGITALE6 := '1';
IF ( DIGITALEC6J = 0 ) THEN DIGITALE6 := '0';
GOTOXY(15,15);
WRITELN (' CANAL DIGITAL 6 (1/0) .....
',DIGITALE6);

GOTOXY(54,17); WRITELN(' ':10);
IF ( DIGITALEC7J = 1 ) THEN DIGITALE7 := '1';
IF ( DIGITALEC7J = 0 ) THEN DIGITALE7 := '0';
GOTOXY(15,17);
WRITELN (' CANAL DIGITAL 7 (1/0) .....
',DIGITALE7);

GOTOXY(54,19); WRITELN(' ':10);
IF ( DIGITALEC8J = 1 ) THEN DIGITALE8 := '1';
IF ( DIGITALEC8J = 0 ) THEN DIGITALE8 := '0';
GOTOXY(15,19);
WRITELN (' CANAL DIGITAL 8 (1/0) .....
',DIGITALE8);
END;

```

PROCEDURE SALIR;

```

( PROCEDIMIENTO PARA SALIR AL DOS )
BEGIN
  CLRSCR;
END;

PROCEDURE INICIALIZAR;
( ESTA SUBROUTINA SIRVE PARA PREGUNTAR AL USUARIO LOS DATOS
CON LOS QUE SE
PROGRAMA LA FORMA DE OPERAR LA TARJETA: MODO, GANANCIA, NUM.
DE CANALES,
FRECUENCIA DE MUESTREO, TIEMPO DURANTE EL CUAL SE MUESTREA Y
NOMBRE DEL
ARCHIVO DE ALMACENAMIENTO DE LAS MUESTRAS )
BEGIN
  GOTOXY (1,5);
  WRITELN ('SENALES EN MODO DIFERENCIAL (0) O EN MODO COMUN
(1)? ');
  COORX:=WHEREX;
  COORY:=WHEREY;
  (#I-)
  REPEAT
    READLN (MODO);
    BUFFER:=IORESULT;
    IF (BUFFER <> 0) OR (MODO > 1) OR (MODO < 0) THEN
      BEGIN
        GOTOXY (COORX,COORY);
        CLREOL;
        GOTOXY (COORX,COORY);
      END;
    UNTIL (BUFFER=0) AND (MODO=1) OR (MODO=0);

    GOTOXY (1,7);
    WRITELN ('CUAL ES EL FACTOR DE GANANCIA (DEL 0 AL 7) ?');
    COORX:=WHEREX;
    COORY:=WHEREY;
    REPEAT
      READLN (GANANCIA);
      BUFFER:= IORESULT;
      IF (BUFFER<>0) OR (GANANCIA > 7) OR (GANANCIA < 0) THEN
        BEGIN
          GOTOXY (COORX,COORY);
          CLREOL;
          GOTOXY (COORX,COORY);
        END;
      UNTIL (BUFFER = 0) AND (GANANCIA <= 7) AND (GANANCIA
>= 0);

      GANANCIA:= GANANCIA SHL 1;
      GANAN_MODO:= GANANCIA OR MODO;
      GOTOXY(1,9);
      IF MODO = 1 THEN WRITELN ('EN MODO SENCILLO SE
DISPONE DE 16 CANALES,')

```

```

ELSE WRITELN ('EN MODO DIFERENCIAL SE DISPONE DE 8
CANALES,');
WRITELN ('CUANTOS CANALES SE USARAN?');
COORDX:=WHEREX;
COORDY:=WHEREY;
IF MODO=0 THEN
BEGIN
REPEAT
READLN (NUM_CANALES);
BUFFER:=IORESULT;
IF (BUFFER <> 0) OR (NUM_CANALES > 8) OR
(NUM_CANALES < 1) THEN
BEGIN
GOTOXY (COORDX,COORDY);
CLREOL;
GOTOXY(COORX,COORY);
END
UNTIL (BUFFER=0) AND (NUM_CANALES <= 8) AND
(NUM_CANALES >= 1)
END
ELSE
BEGIN
REPEAT
READLN(NUM_CANALES);
BUFFER:=IORESULT;
IF (BUFFER <> 0) OR (NUM_CANALES > 16) OR
(NUM_CANALES < 1) THEN
BEGIN
GOTOXY (COORDX,COORDY);
CLREOL;
GOTOXY (COORDX,COORDY);
END;
UNTIL (BUFFER = 0) AND (NUM_CANALES <> 16) AND
(NUM_CANALES >= 1)
END;
FREC_MAX:=0.0005 * NUM_CANALES;
FREC_MAX:= 1/ FREC_MAX;
GOTOXY (1,12);
WRITELN ('CUAL ES LA FRECUENCIA DE MUESTREO?');
COORDX:=WHEREX;
COORDY:=WHEREY;
REPEAT
READLN(FRECUENCIA);
BUFFER:=IORESULT;
IF (BUFFER <> 0) OR (FRECUENCIA > FREC_MAX) OR
(FRECUENCIA < 0) THEN
BEGIN
GOTOXY(COORX,COORY);
CLREOL;
GOTOXY(COORX,COORY);
END;

```

```

    UNTIL (BUFFER = 0) AND (FRECUENCIA <= FREQ_MAX) AND
(FRECUENCIA >= 0);
    DIV_FRECUENCIA:=ROUND (20000/FRECUENCIA);
    FRECUENCIA_REAL:=20000/DIV_FRECUENCIA;
    TIEMPO_MAX:= TRUNC
(MEMAVAIL*0.5/(NUM_CANALES*FRECUENCIA));
    GOTOXY(1,14);
    WRITELN('FRECUENCIA REAL DE
MUESTREO=',FRECUENCIA_REAL;8;3);
    WRITELN('DURANTE CUANTO TIEMPO SE MUESTREARA?');
    COORX:=WHEREX;
    COORY:=WHEREY;
    REPEAT
        READLN (TIEMPO_MUESTREO);
        BUFFER:=IORESULT;
        IF (BUFFER <> 0) OR (TIEMPO_MUESTREO>TIEMPO_MAX)
OR (TIEMPO_MUESTREO<0)
            THEN
                BEGIN
                    GOTOXY(COORX,COORY);
                    CLREOL;
                    GOTOXY(COORX,COORY);
                END;
                UNTIL (BUFFER=0) AND
(TIEMPO_MUESTREO<=TIEMPO_MAX) AND
(TIEMPO_MUESTREO >= 0);
                GOTOXY(1,17);
                WRITELN ('COMO SE LLAMARA EL ARCHIVO DE
ALMACENAMIENTO DE LAS MUESTRAS?');
                WRITELN ('(INCLUYA EL DRIVE)');
                REPEAT
                    READLN (NOMBRE);
                    BUFFER:=IORESULT;
                    UNTIL BUFFER=0;
                    (*I+)
                END;
            END;

PROCEDURE INTB;EXTERNAL;
(* LA SUBROUTINA DE ATENCION DE LAS INTERRUPCIONES DEL
CONV. A/D ES
DECLARADA RUTINA ESCRITA EN ENSAMBLADOR *)

PROCEDURE INSTALAB;
(* ESTA SUBROUTINA SIRVE PARA INSTALAR EL VECTOR DE INT.
DEL CONV. A/D
Y HABILITAR LAS INTERRUPCIONES PROVENIENTES DEL
MISMO *)
    BEGIN
        GET INTVEC(INT_B,VECTORB);
        SET INTVEC(INT_B,@INTB);
        PORT [IMR_8259] := PORT[IMR_8259] AND IRQ3;
    END;

```

```
PROCEDURE REINSTALAB;  
( ESTA SUBRRUTINA SIRVE PARA DESHABILITAR LAS  
INTERRUPCIONES DEL CONV.
```

```
A/D )
```

```
BEGIN
```

```
PORT[IMR_8259] := PORT[IMR_8259] OR NOT_IRQ3;  
END;
```

```
PROCEDURE INTA;EXTERNAL;  
( LA SUBRRUTINA DE ATENCION DE LAS INTERRUPCIONES DEL  
CRONOMETRO ES
```

```
DECLARADA RUTINA ESCRITA EN ENSAMBLADOR )
```

```
PROCEDURE INSTALAA;  
( ESTA SUBRRUTINA SIRVE PARA INSTALAR EL VECTOR DE INT.  
DEL CRONOMETRO
```

```
Y HABILITAR LAS INTERRUPCIONES PROVENIENTES DEL MISMO  
)
```

```
BEGIN
```

```
GET INTVEC(INT_A,VECTORA);  
GET INTVEC(INT_A,@INTA);  
PORT [IMR_8259] := PORT [IMR_8259] AND IRQ2;  
END;
```

```
PROCEDURE REINSTALAA;  
( ESTA SUBRRUTINA SIRVE PARA DESHABILITAR LAS  
INTERRUPCIONES DEL CRONOMETRO )
```

```
BEGIN
```

```
PORT [IMR_8259] := PORT [IMR_8359] OR NOT_IRQ2;  
END;
```

```
PROCEDURE INICIA;EXTERNAL;
```

```
PROCEDURE PSANALOG;
```

```
BEGIN
```

```
PRESENTACION;
```

```
INICIALIZAR;
```

```
( SE CALCULAN EL NUMERO DE MUESTRAS POR CANAL Y SE  
BLANQUEA EL VECTOR  
MUESTRA [I] )
```

```
NUM_MUESTRAS := TRUNC ( TIEMPO_MUESTREO *  
FRECUENCIA_REAL );  
FOR I := 0 TO NUM_CANALES - 1 DO  
MUESTRA [I] := 0;
```

```
INICIA ;
```

```

INSTALAB;
INSTALAA;
( SE GUARDAN EL SEGMENTO Y OFFSET DE LA DIRECCION EN
MEMORIA HACIA
LA QUE APUNTARA LA VARIABLE AUXILIAR APUNTADOR,
PARA ALMACENAMIENTO
DE LAS MUESTRAS )

SEGMENTO := SEG (HEAPPTR^);
OFFSET := OFS(HEAPPTR^);
APUNTADOR := PTR (SEGMENTO,OFFSET);

( SE CALCULA EL NUMERO DE SEGMENTOS DE MEMORIA QUE
HABRA QUE UTILIZAR
PARA ALMACENAR LAS MUESTRAS, CADA SEGMENTO CONTIENE
65536 BYTES )

NUM_SEG_R := NUM_MUESTRAS * NUM_CANALES / 32768;
NUM_SEG_I := TRUNC ( NUM_SEG_R );

( A CONTINUACION SE ENTRA EN UN CICLO ITERATIVO DE
MUESTREO EN EL QUE LAS
MUESTRAS SON ALMACENADAS EN LA MEMORIA RAM DE LA
COMPUTADORA, CADA
VEZ QUE LA BANDERA VALIDACION ES IGUAL A UNO )

K := 0;
VALIDACION := 0;
REPEAT
IF NUM_SEG_I > 0 THEN
POR_SEG_POR_CANAL_NUM_MUESTRAS := 32768 DIV NUM_CANALES
ELSE POR_SEG_POR_CANAL_NUM_MUESTRAS := TRUNC
(NUM_SEG_R *(32768 DIV NUM_CANALES));
CUENTA := 0;

REPEAT
IF VALIDACION = 1 THEN
BEGIN
FOR I := 0 TO NUM_CANALES - 1 DO
BEGIN
( LA MUESTRA DEL CANAL I ES ALMACENADA EN LA
LOCALIDAD DE MEMORIA
A LA QUE APUNTA LA VARIABLE APUNTADOR )
APUNTADOR^ := MUESTRA [I];
( LA DIRECCION A LA QUE APUNTA APUNTADOR ES
ACTUALIZADA )
APUNTADOR := PTR (SEGMENTO + K,
OFS(APUNTADOR^ ) + 2 );
END;
VALIDACION := 0;

```

```

END;
UNTIL CUENTA > POR_SEG_POR_CANAL_NUM_MUESTRAS;
< CADA VEZ QUE SE TERMINA UN SEGMENTO, SE PROCEDE A
INICIAR EL
    ALMACENAMIENTO OTRO SEGMENTO }
    K:=K+1;
    NUM_SEG_R:= NUM_SEG_R - 1;
    NUM_SEG_I:= NUM_SEG_I - 1;
UNTIL NUM_SEG_I < 0;

TIMERTIK;

REINSTALAA;

REINSTALAB;

WRITELN(' TERMINO MUESTRED ');

APUNTADOR:= PTR(SEGMENTO,OFFSET);
ASSIGN (FTEXT,NOMBRE);
SETTEXTBUF (FTEXT,BUF);
REWRITE (FTEXT);
NUM_SEG_R:= NUM_MUESTRAS * NUM_CANALES / 32768;
NUM_SEG_I:= TRUNC (NUM_SEG_R);
K:=0;

< A CONTINUACION SE ENTRA EN UN CICLO ITERATIVO PARA
EL ALMACENAMIENTO
DE LAS MUESTRAS, SACANDOLAS DE LAS LOCALIDADES DE
MEMORIA RAM A UN
ARCHIVO EN WDISC }
SECUENCIA := 0;
REPEAT
    IF NUM_SEG_I > 0 THEN
POR_SEG_POR_CANAL_NUM_MUESTRAS:=32768 DIV NUM_CANALES
    ELSE POR_SEG_POR_CANAL_NUM_MUESTRAS:=
TRUNC(NUM_SEG_R * (32768 DIV NUM_CANALES));
    CUENTA := 0;
    REPEAT
        I := 0;
        < SE PROCEDE A ABRIR EL ARCHIVO EN MEMORIA
MAGNETICA }
        WRITE (FTEXT,SECUENCIA,' ');
        SECUENCIA := SECUENCIA +1;
        CUENTA := CUENTA + 1;
        REPEAT
            WRITE (FTEXT,APUNTADOR^,' ');
            APUNTADOR :=
PTR(SEGMENTO+K, OFS (APUNTADOR^)+2);
            I := I + 1;
            UNTIL I >= NUM_CANALES;
            WRITELN (FTEXT);

```

```

        UNTIL
        CUENTA > POR_SEG_POR_CANAL_NUM_MUESTRAS;
        K := K+1;
        NUM_SEG_R:=NUM_SEG_R - 1;
        NUM_SEG_I:=NUM_SEG_I - 1;
        UNTIL NUM_SEG_I < 0 ;
        ( SE PROCEDE A CERRAR EL ARCHIVO EN MEMORIA
MAGNETICA )
END;

( ***** INICIA EL PROGRAMA PRINCIPAL
***** )

BEGIN
  ( LOCALIDADES QUE SE ASIGNARON A CADA PUERTO )
  PUERTO0 := #300 ;
  PUERTO1 := #302 ;
  PCONT1  := #320 ;
  PCONT2  := #321 ;
  PCONT3  := #322 ;
  VREF    := 5 ;

  ( INICIALIZACION DEL PUERTO )
  PORT [#323] := #89 ;

  ( PALABRAS DE CONTROL DE LOS CONTADORES Y ESCRITURA A C/U
)
  PORT [#32B] := #7B ;
  PORT [#329] := #FF ;
  PORT [#329] := #FF ;

  PORT [#32B] := #8B ;
  PORT [#32A] := #FF ;
  PORT [#32A] := #FF ;

  PORT [#32B] := #3B ;
  PORT [#32B] := #FF ;
  PORT [#32B] := #FF ;

  M := 1;
  REPEAT
    BITAL[M] := 0;
    M := M + 1;
  UNTIL ( M = 17 );

  PRESENTACION;
  REPEAT
    GOTOXY(16,22); WRITELN ( ' OPRIMA G Y <RETURN> PARA
CONTINUAR ' );
    GOTOXY(58,22); WRITELN ( ' ' ;10);
    GOTOXY(58,22); READLN (OPC);
  UNTIL (( XL13 = 1) AND (( OPC = 'G' ) OR ( OPC = 'g' )));

```

```

GOTOXY(55,9);

REGRESAR;
MASCARA;
REPEAT
  GOTOXY(55,9); WRITELN(' ',10);
  GOTOXY(55,9); READLN(OPCION);
  X[2] := LENGTH (OPCION);
  IF (( OPCION = 'a') OR ( OPCION = 'A')) THEN OPCIONA :=
'A';
  IF (( OPCION = 'b') OR ( OPCION = 'B')) THEN OPCIONA :=
'B';
  IF (( OPCION = 'c') OR ( OPCION = 'C')) THEN OPCIONA :=
'C';
  UNTIL (( X[2] = 1 ) AND (( OPCIONA = 'A') OR ( OPCIONA =
'B' ) OR
( OPCIONA = 'C')));

DOSS;
CASE OPCIONA OF
'A' :
  BEGIN
    ANALOGICOS;
    REPEAT
      GOTOXY(53,5); WRITELN(' ',10);
      GOTOXY(53,5); READLN(OPCION1);
      X[3] := LENGTH (OPCION1);
      IF (( OPCION1 = 'a') OR ( OPCION1 = 'A')) THEN
OPCION1A := 'A';
      IF (( OPCION1 = 'b') OR ( OPCION1 = 'B')) THEN
OPCION1A := 'B';
      IF (( OPCION1 = 'c') OR ( OPCION1 = 'C')) THEN
OPCION1A := 'C';
      UNTIL (( X[3] = 1) AND (( OPCION1A = 'A') OR
(OPCION1A = 'B') OR
(OPCION1A = 'C')));
      CASE OPCION1A OF
'A' :
      BEGIN
        PSANALOG;
        GOTO DOSS;
      END;
'B' :
      BEGIN
        UNO ;
        ESCRITURA;
        GOTO DOSS;
      END;
'C' :
      BEGIN
        GOTO REGRESAR;
      END;

```

```

END;
'B' ;
TRES;
BEGIN
  DIGITALES;
  REPEAT
    GOTOXY(53,5); WRITELN(' ' ;10);
    GOTOXY(53,5); READLN(OPCION2);
    X[4] := LENGTH (OPCION2);
    IF ((OPCION2 = 'a' ) OR (OPCION2 = 'A')) THEN
OPCION2A := 'A';
    IF ((OPCION2 = 'b' ) OR (OPCION2 = 'B')) THEN
OPCION2A := 'B';
    IF ((OPCION2 = 'c' ) OR (OPCION2 = 'C')) THEN
OPCION2A := 'C';
    IF ((OPCION2 = 'd' ) OR (OPCION2 = 'D')) THEN
OPCION2A := 'D';
    IF ((OPCION2 = 'e' ) OR (OPCION2 = 'E')) THEN
OPCION2A := 'E';
    IF ((OPCION2 = 'f' ) OR (OPCION2 = 'F')) THEN
OPCION2A := 'F';
    IF ((OPCION2 = 'g' ) OR (OPCION2 = 'G')) THEN
OPCION2A := 'G';
  UNTIL (( X[4] = 1) AND ((OPCION2A = 'A') OR (OPCION2A
= 'B') OR
(OPCION2A = 'C') OR (OPCION2A = 'D') OR (OPCION2A =
'E') OR
(OPCION2A = 'F') OR (OPCION2A = 'G')));
  CASE OPCION2A OF
    'A' ;
  BEGIN
    CONTADOR1;
    GOTOXY(15,20);
    WRITELN(' OPRIMA G PARA CONTINUAR ');
    REPEAT
      GOTOXY(40,20); WRITELN(' ');
      GOTOXY(40,20); READLN(CONTP1);
    UNTIL (( CONTP1 = 'G' ) OR (CONTP1 = 'g'));
    GOTO TRES;
  END;
    'B' ;
  BEGIN
    CONTADOR2;
    GOTOXY(15,20);
    WRITELN(' OPRIMA G PARA CONTINUAR ');
    REPEAT
      GOTOXY(40,20); WRITELN(' ');
      GOTOXY(40,20); READLN(CONTP2);
    UNTIL (( CONTP2 = 'G' ) OR (CONTP2 = 'g'));
    GOTO TRES;
  END;
  END;

```

```

END;
'C' ;
BEGIN
  CONTADOR3;
  GOTOXY(15,20);
  WRITELN(' OPRIMA G PARA CONTINUAR ');
  REPEAT
    GOTOXY(40,20); WRITELN(' ');
    GOTOXY(40,20); READLN(CONTP3);
  UNTIL (( CONTP3 = 'G') OR ( CONTP3 = 'g'));
  GOTO TRES;
END;
'D' ;
BEGIN
  CLRSCR;
  QDATO2 := PORT [#326];
  PUERTO;
  GOTOXY(18,24); WRITELN(' OPRIMA G Y <RETURN>
  PARA CONTINUAR ');
  REPEAT
    GOTOXY(18,25); WRITELN(' ');
    GOTOXY(18,25); READLN(LPUERTO);
  UNTIL (( LPUERTO = 'G') OR (LPUERTO = 'g'));
  GOTO TRES;
END;
'E' ;
BEGIN
  CLRSCR;
  PUERTO;
  GOTOXY(18,24); WRITELN(' OPRIMA G Y <RETURN>
  PARA CONTINUAR ');
  REPEAT
    GOTOXY(18,25); WRITELN(' ');
    GOTOXY(18,25); READLN(LPUERTO);
  UNTIL ((LPUERTO = 'G') OR (LPUERTO = 'g'));
  GOTO TRES;
END;
'F' ;
BEGIN
  CLRSCR;
  PUERTO;
  GOTOXY(18,24); WRITELN(' OPRIMA G Y <RETURN>
  PARA CONTINUAR ');
  REPEAT
    GOTOXY(18,25); WRITELN(' ');
    GOTOXY(18,25); READLN(LPUERTO);
  UNTIL ((LPUERTO = 'G') OR (LPUERTO = 'g'));
  GOTO TRES;
END;
'G' ;
BEGIN
  GOTO REGRESAR;

```

```
      END;  
    END;  
  'C'  
  BEGIN  
    CLRSCR;  
    GOTO EXIT;  
  END;  
END;  
EXIT;  
END.  
.
```

SECCION ENSAMBLADOR

```

data segment word public
extrn muestra: word;
extrn gagan_modo: word;
extrn num_canales: word;
extrn div_frecuencia: word;
extrn validacion: word;
extrn cuenta: word;
data ends

```

```

code segment byte public
assume cs:code, ds:data
public Inicia
public IntA
public IntB

```

```

en_ganan_modo db
en_num_canales db
en_bandera db
en_j db
en_div_frecuen dw
en_validacion db

```

```

Inicia proc far
push ax
push bx
push cx
push dx
push si
push di
push bp
push ds
push es

```

```
cli
```

```

mov ax,ganan_modo
mov cs:en_ganan_modo,al; se pasan la ganancia y modo
al cs
mov ax,num_canales
mov cs:en_num_canales,al; se pasan el num. de canales
al cs
mov ax,div_frecuencia
mov cs:en_div_frecuen,ax; se pasan el num. a
decrementar por el cont. 1 a cs.
mov ax,validacion
mov cs:en_validacion,al; se pasa el valor de la
bandera de validacion al cs
mov dx,038CH
mov al,cs:en_ganan_modo
out dx,al; programa modo y ganancia

```

```

mov dx,03B7H
mov al,16H
out dx,al
mov al,74H
out dx,al; programa cont.0 en modo 3 y cont.1 en modo
2
mov dx,03B4H
mov al,32H
out dx,al; el cont.0 generará una señal cuadrada de 20
KHz
mov dx,03B5H
mov ax,cs:en_div_frecuen
out dx,al; se envía el LSB de la cuenta a decrementar
mov al,ah
out dx,al; se envía el MSB de la cuenta a decrementar

```

```

mov dx,03B8H
xor al,al
out dx,al; se resetea el flip-flop del cronómetro

```

```

pop es
pop ds
pop bp
pop di
pop si
pop dx
pop cx
pop bx
pop ax
sti
ret

```

Inicia endp

```

IntA proc far
push ax
push bx
push cx
push dx
push si
push di
push bp
push ds
push es

```

```

mov dx,03B8H
xor al,al
out dx,al; se resetea el flip-flop del cronómetro

```

```

mov al,20H
mov dx,20H
out dx,al; se envía la EOI al 8259A (PIC)
sti

```

```

;se entra en un ciclo iterativo para el muestreo de
;las señales de los canales

mov  cl,cs:en_num_canales
xor  bx,bx
mov  cs:en_j,0H
lazo:call  conver
mov  cs:en_bandera,0H
dec  cl
loop:cmp  cs:en_bandera,00H
je   loop
cnp  cl,0H
jne  lazo

```

```

;se incrementa CUENTA que es una variable comun a
;PASCAL y Ensamblador, en la que lleva el conteo del
;num. de muestras

```

```

inc  cuenta
mov  validacion,01H

```

```

pop  es
pop  ds
pop  bp
pop  di
pop  si
pop  dx
pop  cx
pop  bx
pop  ax

```

```

;sti
iret

```

```

IntA endp

```

```

conver proc near

```

```

;se pide inicio de la conversion de al muestra del
;canal j

```

```

mov  dx,0380H
mov  al,cs:en_j
out  dx,al
inc  cs:en_j
ret

```

```

conver endp

```

```

IntB proc far

```

```

;se realiza la lectura de la conversion de la muestra
;del canal j

```

```

push ax
push dx
push si
push di
push bp
push ds
push es

mov dx,03B0H
in al,dx
mov ah,al
in al,dx; lee el resultado de la conversión
mov muestra(bx),ax; el resultado es asignado a la
;variable MUESTRA que es común a PASCAL y
;Ensamblador

inc bx
inc bx; se incrementó dos veces bx
mov cs:en_bandera,01H

mov al,20H
mov dx,20H
out dx,al; se envía la EOI al 8259A (PIC)

pop es
pop ds
pop bp
pop di
pop si
pop dx
pop ax

sti
iret
IntB endp

code ends
end

```

APENDICE 6

EVALUACION DE COSTOS

Como todo proyecto de ingeniería, la elaboración de costos es un punto muy importante, en donde se deben de tomar en cuenta los siguientes aspectos:

- A) Diseño de ingeniería
- B) Costos de integración de componentes
- C) Mano de obra
- D) Tiempo
- E) Depreciación del equipo usado
- F) Costos fijos (luz, agua, renta, etc.)

Evaluando y resumiendo lo anterior tenemos:

- A) El diseño de ingeniería no repercute en los costos debido a que este trabajo no persigue fines lucrativos.
- B) La suma de los costos de los componentes asciende a:.....\$ 476,000.00
- C),D),E),F) Estos aspectos forman parte del costo de la elaboración del circuito impreso y el ensamble de los componentes ascendiendo a un total de:.....\$ 375,000.00

Sumando los puntos anteriores el costo total de la tarjeta asciende a:.....\$ 871,000.00

NOTA: Este costo unitario fué calculado para un pedido mínimo de 50 tarjetas.
Los precios son en pesos mexicanos.

Por otro lado el valor obtenido de las tarjetas de adquisición de datos que se encuentran en el mercado son los siguientes:

- 1 Tarjeta de 8 canales analógicos entrada 1 canal digital salida. \$955,000.00 M.N.
- 1 Tarjeta de 16 canales analógicos de entrada 2 canales analogicos de salida. \$1,325,250.00 M.N.

1 Tarjeta de 16 canales analógicos de entrada y 2 canales de entrada y de salida digital y ganancia programable.

\$3,225,325.00 M.N.

A todo lo anterior hay que sumarle el paquete de software el cual tiene un costo de: \$927,000.00 M.N.

Evaluando los costos de la elaboración de la tarjeta de funciones múltiples, se tiene adicionalmente una mayor ventaja sobre las demás tarjetas, las cuales son:

- a) Ingeniería 100% nacional
- b) Reducción de costos por volumen
- c) Mayores opciones de uso

De lo anterior podemos resumir y determinar que es posible la elaboración de dicha tarjeta y darle la mejor aplicación en el campo que lo requiera, obteniendo con ello un gran beneficio.

(Los precios anteriores son cotizaciones hechas en JUL/91)