

136  
29.



**UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO**

**FACULTAD DE INGENIERIA**

**DISEÑO DE UN MEDIDOR UNIVERSAL DE  
VARIABLES ELECTRICAS**

**T E S I S**

QUE PARA OBTENER EL TITULO DE:  
**INGENIERO MECANICO ELECTRICISTA  
(AREA ELECTRONICA)**  
**P R E S E N T A N :**  
**ADRIAN RESENDIZ MAGUEY  
EDGAR RICARDO RICHTER SALCEDO  
FERNANDO ANTONIO ROCHA SANCHEZ**

DIRECTORA: ING. GLORIA MATA HERNANDEZ.



MEXICO, D. F.,

ENERO 1998

TESIS CON  
FALLA DE ORIGEN

2502-11



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.

---

*A Dios por concedernos vida y salud a mi familia y a mi.*

*A María de la Luz y Jáime por ser unos padres muy buenos, el sacrificio que hacen para que sigamos estudiando mis hermanos y yo, por la motivación en seguir siempre adelante y sobre todo por su amor.*

*A Leticia, Alicia e Ismael por ser muy buenos hermanos conmigo, apoyarme en todo y además de su cariño.*

*A mi tía Magdalena por apoyarnos siempre que lo necesitamos.*

*A nuestra Directora de Tesis Ingeniera Gloria Mata Hernández por su dedicación, paciencia, amistad, consejos y sobre todo por confiar en mi.*

*A la Universidad Nacional Autónoma de México, por la oportunidad de cursar una licenciatura y ser una persona útil a México.*

*A los profesores y compañeros de la Facultad por su apoyo incondicional y amistad.*

*A Rocío por su apoyo en todo momento.*

*Gracias a todos ...*

**Adrián Reséndiz Maguey**

---

*A Dios, por todo.*

*A la Universidad Nacional Autónoma de México, por la oportunidad de cursar una licenciatura y ser una persona útil a México.*

*A mi Madre Lydia, que es el ser más maravilloso y lleno de bondades del mundo, por todo su amor, cariño, sacrificio, compañía, apoyo y confianza, que me impulsaron en seguir adelante a lo largo de mi vida.*

*A mis hermanos Miguel y Guillermo por su apoyo.*

*A mis abuelos Fernando y Lydia, por su apoyo, paciencia, comprensión y por haberme ayudado en todo, en el transcurso de mis estudios.*

*A nuestra Directora de Tesis, Ingeniera Gloria Mata Hernández, por su dedicación, paciencia, amistad y confianza.*

**Edgar Ricardo Richter Salcedo**

---

**DEDICATORIA:**

*Esta Tesis la dedico  
a mi abuela,  
Elpidia Medina y a Jesucristo  
con todo mi Cariño y Aprecio.*

**AGRADECIMIENTOS:**

*A todos las personas que nos ayudaron desinteresadamente, algunas que conocimos rápidamente y no las volvimos a ver y sobre todo a la Ingeniera Gloria Mata Hernández.*

*Un agradecimiento muy Especial a mis dos grandes amigos y compañeros por la paciencia y apoyo que me brindaron, Edgar Richter y Adrián Reséndiz y sobre todo a la vida misma que me permitió llegar hasta aquí.*

*A una persona que me hizo darme cuenta de la importancia hacer sentir a los demás importantes y sobre todo a mi mismo, y que me estimuló profundamente, María Esther Contreras Nava, a donde quiera que se encuentre.*

**Fernando Antonio Rocha Sánchez**

---

# ÍNDICE

---

PRÓLOGO	1
INTRODUCCIÓN	4
CAPÍTULO 1. DESCRIPCIÓN DEL HARDWARE	10
1.1. DESCRIPCIÓN GENERAL	10
1.2. SENSOR DE VARIABLES ELÉCTRICAS (SVE)	15
1.3. FUENTE SENOIDAL (SEN)	19
1.4. RESISTENCIA PROGRAMABLE (RES)	21
1.5. DIVISORES DE VOLTAJE PROGRAMABLES (DIV)	22
1.6. DECODIFICACIÓN, ACOPLAMIENTO Y PROTECCIONES (DAP)	24
1.7. RELEVADORES (REL5V Y REL12V)	28
1.8. AMPLIFICADOR DE GANANCIA PROGRAMABLE (AGP)	30
1.9. ADECUACIÓN PARA EL CAD (ADE)	32
1.1.0. GENERACIÓN DEL RELOJ DE MUESTREO (GRM)	34
1.11. CONTROL (CTRL)	37
1.12. FUENTE DE ALIMENTACIÓN (FTE)	41
1.13. CAD Y ALMACENAMIENTO TEMPORAL (CAD)	43
1.14. PUERTOS DE ENTRADA Y SALIDA (PTS)	46
1.15. OPTOACOPLADORES (OPT)	48
1.16. ARBITRAJE DEL BUS DE DATOS Y LÓGICA DE DECODIFICACIÓN (ALD)	50
1.17. SLOT IBM PC XT , 62 TERMINALES (SXTPC)	56

---

<b>CAPÍTULO 2. ANÁLISIS DINÁMICO</b>	<b>58</b>
2.1. MODO DE MUESTREO	58
2.2. TRANSICIÓN DEL MODO DE MUESTREO AL DE TRANSMISIÓN	62
2.3. MODO DE TRANSMISIÓN	65
2.4. TRANSICIÓN DEL MODO DE TRANSMISIÓN AL DE MUESTREO	68
<b>CAPÍTULO 3. DESCRIPCIÓN DEL SOFTWARE</b>	<b>72</b>
3.1. DESCRIPCIÓN GENERAL	72
3.2. COMANDOS PARA EL CONTROL DEL HARDWARE	74
3.3. FUNCIONES PARA EL CONTROL DEL HARDWARE	84
3.4. GRÁFICOS	87
3.5. TRANSFORMACIÓN DE UNIDADES	88
3.6. PROCESAMIENTO DE VECTORES IMAGEN DE LAS RAM	88
3.7. CONTROL DEL MENU	89
3.8. VENTANAS	89
3.9. PRINCIPAL	105
<b>CAPÍTULO 4. ESPECIFICACIONES DEL SISTEMA DE MEDICIÓN</b>	<b>107</b>
4.1. PANTALLA	107
4.2. TIEMPO	107
4.3. VOLTAJE	108
4.4. CORRIENTE	109
4.5. POTENCIA	109
4.6. IMPEDANCIA	111
4.7. RESPUESTA EN FRECUENCIA	113
<b>CONCLUSIONES</b>	<b>116</b>

---

# **PRÓLOGO**

---

En las últimas décadas, el mundo ha sufrido una acelerada transformación en el aspecto tecnológico y uno de los cambios más impactantes, es el de la instrumentación. Hoy en día existen conceptos que sólo hace algunos años parecían ciencia ficción, como lo es la instrumentación virtual.

El medidor universal se basa en este concepto de instrumentación virtual, que es la base para su diseño; en donde combinando una computadora personal e interfaces se hacen las funciones de varios instrumentos, sin contar físicamente con estos. La información del usuario o para el usuario, se muestra a través de la pantalla de una computadora, en donde se observa un panel frontal con botones para cambiar las escalas de tiempo y amplitud de las variables eléctricas y visualizadores para mostrar las magnitudes y forma de onda de estas.

El medidor universal de variables eléctricas es creado a partir de la necesidad de contar con un sistema que reúna la funcionalidad de varios instrumentos en un sólo instrumento de medición, que sea capaz de medir voltaje, corriente, impedancia, potencia, frecuencia y mostrar la forma de la señal de manera similar a un osciloscopio.

El tener varios instrumentos disponibles en un sólo sistema ahorra espacio, es más cómodo para el usuario, ya que controla todo desde su PC, es más económico, ya que cada vez más personas cuentan con una PC. Además de que se está en la posibilidad de disponer de los datos de las lecturas en archivos, con lo cual se pueden analizar con programas adicionales para manejos estadísticos, de procesamiento digital, etc.



Por otro lado, este trabajo consta de Introducción, 4 Capítulos, Conclusiones y Apéndices.

En la "Introducción", se explica que es un instrumento virtual, sus características y la importancia del desarrollo de las computadoras personales para la instrumentación virtual. Se describe el desarrollo de la instrumentación desde los primeros instrumentos analógicos hasta los modernos instrumentos digitales y la instrumentación virtual.

El capítulo uno "Descripción del Hardware", se presenta de una manera general el funcionamiento del hardware y se explica de manera detallada el funcionamiento de cada bloque con sus correspondientes diagramas.

En el capítulo dos "Análisis Dinámico", se explican y muestran los diagramas de tiempo relacionados con el control interno y la sincronización del sistema, así como los de comunicación con la PC.

En el capítulo tres "Descripción del Software", se detallan los diversos módulos que integran al software, así como la forma de utilizar los menús del medidor universal por el usuario.

El capítulo cuatro "Especificaciones del Sistema de Medición", presenta las especificaciones del sistema, tales como las características de la pantalla, rango, resolución, respuesta en frecuencia y el resultado de las pruebas realizadas al sistema.

Presentamos las conclusiones del diseño del medidor universal, en donde sintetizamos y validamos el correcto funcionamiento del sistema.

En los apéndices se incluye información adicional requerida para el diseño del sistema, incluyen el listado del programa; ecuaciones y nomenclatura; descripción de la PC enfocada a tarjetas de expansión y especificaciones técnicas.

Para este proyecto utilizamos la técnica de diseño funcional descendente Top-Down, que se utiliza a nivel componente para desarrollo de prototipos, parte de un problema al que se va particionando en módulos para simplificarlo. Esta técnica la utilizamos tanto para el diseño del hardware como para el diseño del software, siguiendo los siguientes pasos.

Para el hardware: especificación del hardware; partición del problema en módulos funcionales; selección de la familia de componentes; dar un diagrama a bloques esquemático de cada modulo; desarrollar el circuito detallando cada bloque; alambrar y probar cada bloque; conjuntar bloques en cada modulo; conjuntar módulos en un sistema final y realizar pruebas del sistema.

Para el software: especificación del software; seleccionar el lenguaje; descomponer el sistema en módulos funcionales; detallar el algoritmo para cada modulo; codificar y probar cada modulo funcional; conjuntar los módulos para obtener el sistema y realizar pruebas del sistema.

De acuerdo a lo anterior, el objetivo de este trabajo es diseñar el medidor universal de variables eléctricas como una alternativa en la medición de variables eléctricas.

---

# **INTRODUCCIÓN**

---

En la industria de la instrumentación se esta generando una gran variedad de cambios como resultado de la evolución de las computadoras personales (PC's), estos cambios están ocurriendo tanto en las áreas de hardware como de software.

Miles de científicos e ingenieros en todo el mundo usan las PC's para automatizar sus tareas de investigación, diseño y fabricación. Las compañías están creando nuevas herramientas de software e interfaces que hacen posible el desarrollo de una nueva metodología de medición. Este nuevo concepto que hace posible la construcción de un nuevo sistema de medición basado en el uso de la computadora e interfaces, es la instrumentación Virtual.

El instrumento virtual se puede definir como una computadora conectada a un módulo que adquiere, digitaliza y analiza las señales de entrada para su procesamiento; el usuario observa las señales de entrada y salida del sistema de medición a través de un panel frontal interactivo en la pantalla de la computadora, con la apariencia y sentido del instrumento original, con la ventaja de que no requiere de un instrumento físico además se puede utilizar desde un lugar remoto.

El instrumento virtual se presenta a través de un panel frontal interactivo en la pantalla de la computadora, donde el usuario observa las señales de entrada y salida del sistema de medición, con la apariencia y sentido del instrumento original, y con la ventaja de que se puede utilizar desde un lugar remoto.

Al utilizar un instrumento virtual, se utilizan sus librerías para controlar y adquirir datos a través de diversas interfaces (como el GPIB o el RS-232) o a través de tarjetas de

adquisición de datos. Los datos no sólo se pueden adquirir del mundo real, sino de archivos, de otras computadoras en red e incluso puede hacer cálculos matemáticos para simular un proceso o evento físico. Se usan las librerías de análisis para convertir estos datos en resultados entendibles, se puede incorporar procesamiento digital de señales, filtros digitales, análisis estadístico, análisis numérico, etc. Existen numerosas opciones para manipular los resultados: almacenarlos en disco, transferirlos a través de una red a otra computadora, enviarlos a impresión, etc.

La funcionalidad de un instrumento se puede dividir en tres aspectos:

- Adquisición de datos
- Análisis y procesamiento de datos
- Presentación de resultados

Los instrumentos analógicos son controlados manualmente desde un panel frontal constituido por un chasis en el que se suministran las señales de entrada y recibe las señales de salida que reflejan el análisis de estas entradas, con la desventaja de que no se pueden combinar con otros sistemas de medición.

Con la evolución de la PC, los usuarios se han equipado con una poderosa capacidad de procesamiento y flexibilidad que supera las capacidades fijas de un instrumento analógico, en cualquiera de las tres áreas: adquisición de datos, análisis y procesamiento de datos y presentación de los resultados.

Los usuarios cuentan con un suministro constante de herramientas de software fáciles de usar, las más recientes incluyen librerías de instrumentos prediseñados y de análisis de datos. Dentro de las librerías de análisis de datos se incluyen funciones estadísticas, de progresiones, de álgebra lineal, algoritmos de generación de señales en el dominio del tiempo y de la frecuencia, filtros digitales, etc. Las librerías de

instrumentos virtuales tienen funciones para la comunicación serie, usando múltiples puertos RS-232 bajo el control de interrupciones, estas contienen funciones tanto simples como sofisticadas para ofrecer la máxima potencia y flexibilidad de las diversas tarjetas de adquisición de datos (como National Instruments para IBM PC AT, ISA, Micro channel y Macintosh).

La arquitectura interna de un instrumento electrónico moderno es muy parecida a la de una computadora. Los diseñadores de instrumentos establecen sus propios estándares internos con objeto de reducir el costo de desarrollo de microprocesadores y de memorias, lo mismo se hace con librerías de software que realizan la adquisición, el análisis y la presentación de datos. Muchos de los componentes de propósito general que son reutilizables en los instrumentos, son componentes ya estandarizados en las computadoras personales, por lo que éstas son una plataforma ideal para la construcción de instrumentos virtuales.

El hardware de instrumentos virtuales incluye convertidores A/D, D/A, dispositivos de entrada/salida digital, acondicionamiento de la señal de entrada, base de tiempo, de protecciones, etc. Estos componentes son elementos modulares y son instalados como tarjetas hijas dentro de un chasis. Las PC's tienen ranuras de expansión de modo que se pueden conectar tarjetas modulares directamente al bus del procesador.

El software de un instrumento digital tradicional esta empacado en una ROM interna, este software es único para cada instrumento, aunque la mayoría del código es de librerías modulares usadas en el diseño de otros instrumentos.

El software de un instrumento virtual también usa código modular reusable, pero a diferencia de un instrumento tradicional, el cual esta limitado al uso de una memoria ROM especifica, en el instrumento virtual éste código está en forma de archivos de disco instalables. Además múltiples instrumentos virtuales pueden residir y coexistir

en la misma computadora. La pantalla y controles de un instrumento virtual es única, pero un número ilimitado de instrumentos virtuales pueden compartir la misma pantalla de una computadora. Mientras que la pantalla, los controles y el diseño del hardware son únicos para cada instrumento tradicional.

Como ya se mencionó, la plataforma ideal para la construcción de instrumentos virtuales es la computadora personal (PC), por lo que es importante conocer los componentes clave de esta, como el bus del procesador y la base de tiempo integrado.

Los componentes internos del hardware se conectan directamente al bus del procesador y todos los registros del hardware se pueden mapear en el espacio de direcciones de memoria del procesador, por esto, el procesador puede realizar control binario directo a alta velocidad en todos estos componentes. Además, las interrupciones directas del hardware hacen que el procesador se ramifique inmediatamente para manejar los servicios requeridos, se pueden transferir datos en forma binaria o en cualquier otro formato en paralelo a velocidades muy altas, además se puede usar memoria compartida interna para adquirir o generar varios buffers de datos en tiempo real. Se pueden usar técnicas de acceso directo a memoria (DMA) para analizar o preparar datos mientras se adquieren nuevos datos.

Otro componente arquitectónico clave es la base de tiempo integrado, que sincroniza la operación de los componentes del hardware de modo que los múltiples componentes puedan operar coherentemente. Todos operan con la misma base de tiempo y por eso todos tienen la misma vista de la señal de entrada/salida. De ésta manera, cada componente del hardware puede manejar por si mismo una porción específica de la señal total.

El uso de las tarjetas de adquisición de datos (DAQ) conectadas a las ranuras de expansión (slot) de la PC, es una forma de medición y generación de señales analógicas y digitales, en la cual se utilizan el bus del procesador y la base de tiempo integrado. Hay muchas ventajas en el uso de tarjetas de adquisición de datos, tienen más flexibilidad y menores precios que los instrumentos de función única, una tarjeta DAQ puede realizar una variedad de funciones, incluyendo conversiones A/D, conversiones D/A y operaciones de conteo, teniendo una amplia variedad de opciones. Las velocidades de muestreo varían desde menos de 50 kHz hasta más de 1 MHz. La resolución varía desde 8 bits hasta más de 20 bits. Otras características, tales como amplificación de señal y el número de canales de entrada disponibles varían de tarjeta a tarjeta.

Las poderosas PC's hacen que las aplicaciones de las tarjetas de adquisición de datos en tiempo real, sean una realidad, conforme aumenta el poder de procesamiento de las PC's más aplicaciones pueden usarlas en lugar de los instrumentos tradicionales.

El parámetro más importante de una tarjeta de adquisición de datos utilizada para medir señales de CD como por ejemplo: temperatura, presión, flujo y voltaje; es su exactitud. El tiempo de la medición no es crítico, ya que la señal varía relativamente en forma lenta, por lo que para la adquisición de estas señal no se requiere una alta velocidad de muestreo.

De una señal en el dominio del tiempo se puede obtener información, no sólo de su amplitud, sino también de como varía con el tiempo, por lo que la velocidad de muestreo es importante para su adquisición, con una velocidad de muestreo apropiada se pueden reproducir adecuadamente las características de esta de señal. Así los parámetros más importantes de una tarjeta de adquisición de datos para medir señales en el dominio del tiempo son además de la resolución y la exactitud, la

velocidad de muestreo. Ejemplos este tipo de señales son señales de voltaje y corriente alternos, potencia, factor de potencia, impedancia, etc. Se puede estar interesado en algunas características de la forma de onda, tal como la frecuencia, la energía, el periodo, etc.

El componente fundamental de una tarjeta DAQ, es el convertidor *A/D* (*ADC*), este convierte la señal analógica de entrada en un valor digital que el sistema almacena temporalmente en memoria durante la adquisición de datos a altas velocidades.

Un parámetro fundamental del convertidor *A/D* es el número de bits, ya que esto determina el número de niveles diferentes de la salida binaria de una conversión *A/D* y por lo tanto el más pequeño cambio de voltaje detectable. La señal digital no es una buena representación de la señal original, ya que parte de la información se pierde durante la conversión. Aumentando la resolución se puede obtener una muy buena representación de la señal analógica. Muchos de los convertidores *A/D* son de 12 bits produciendo 4096 niveles diferentes. Así el rango de entrada de un convertidor *A/D* y el número de bits son factores que influyen en la precisión de la medición.

Con base en los conceptos de instrumentación virtual, en el avance tanto en hardware como software y de la capacidad de procesamiento de las PC's, es diseñado el medidor universal de variables eléctricas como una alternativa en la medición de variables eléctricas.



---

# **CAPÍTULO 1**

## ***DESCRIPCIÓN DEL HARDWARE***

---

### **1.1. DESCRIPCIÓN GENERAL**

El medidor universal de variables eléctricas es creado como una herramienta útil tanto para los estudiantes en la enseñanza de la ingeniería, así como, para la industria donde se requiera medir en cualquier momento variables como: voltaje, corriente, frecuencia, potencia aparente, potencia real, potencia reactiva, impedancia capacitiva, impedancia inductiva e impedancia resistiva.

El medidor universal esta dividido en módulos interconectados entre sí, con la finalidad de que se pueda identificar con mayor facilidad las fallas que se presenten o para realizar cambios futuros para mejorar el sistema.

El diagrama de bloques general se presenta en la figura 1.1 y corresponde a un sólo canal.

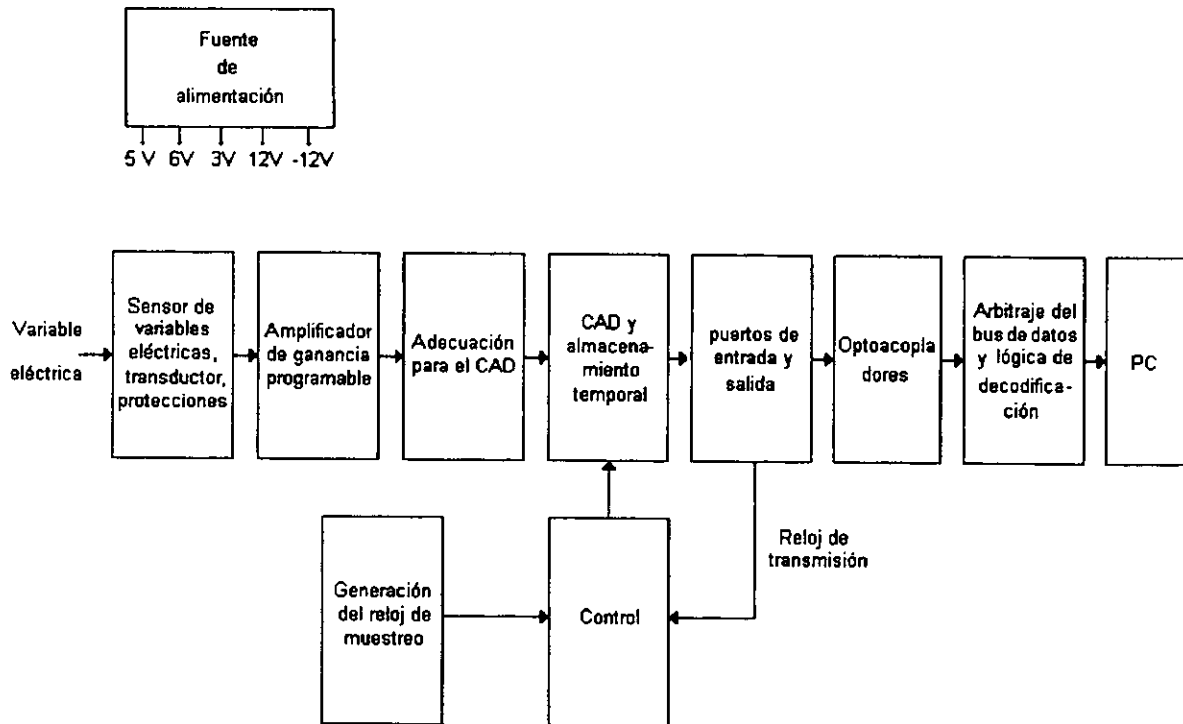


Figura 1.1

El bloque "Fuente de Alimentación" tiene el propósito de generar los voltajes: 3V, 5V, 6V, 12V y -12V que requiere el sistema para funcionar.

La variable eléctrica a medir es detectada en el bloque "Sensor de Variables Eléctricas" y es convertida a una señal de voltaje, ya que el sistema está diseñado para procesar señales de voltaje solamente, además se incluyen protecciones contra sobrevoltaje y sobrecorriente.

La señal de voltaje del bloque anterior es amplificada a un nivel adecuado en el bloque "Amplificador de Ganancia Programable", el grado de amplificación de la señal de voltaje depende de la escala vertical seleccionada.

La señal amplificada pasa a una etapa de adecuación en el bloque "Adecuación para el CAD", donde se procesa para que en la salida, la señal este dentro del rango de operación requerido por el convertidor analógico digital.

El bloque "Generación del Reloj de Muestreo" tiene el propósito de generar la señal de reloj de muestreo para los convertidores analógico digital en los canales A y B. La señal de reloj de muestreo pasa al bloque "Control" y su frecuencia depende de la escala horizontal seleccionada.

Para el bloque "CAD y Almacenamiento Temporal", se definen dos procesos: el modo de muestreo y el modo de transmisión. En el modo de muestreo el convertidor analógico digital convierte la señal analógica ya adecuada en una señal digital a una velocidad dada por el reloj de muestreo, los datos del convertidor son almacenados en memoria RAM.

En el modo de transmisión los datos almacenados en la memoria RAM son transmitidos a la PC a una velocidad dada por el reloj de transmisión generado por la PC a través de un puerto de salida.

El bloque "Control" se encarga de seleccionar, ya sea la señal de reloj de muestreo generada en el bloque "Generación del Reloj de Muestreo" o la señal de reloj de transmisión generada por la PC. Además se generan también las direcciones de memoria para las RAM y las señales de control para los convertidores y memorias RAM en los canales A y B para los modos de muestreo y transmisión.

El bloque "Puertos de Entrada y de Salida" tiene la función de establecer 3 puertos de lectura y 3 puertos de escritura de 8 bits cada uno, estos puertos son activados por medio de las líneas de selección generadas por el bloque "Arbitraje del Bus de Datos y Lógica de Decodificación".

El bloque "Optoacopladores" se encarga de aislar las señales referenciadas a la tierra que maneja la PC (bloque "Arbitraje del Bus de Datos y Lógica de Decodificación") de la tierra que maneja el medidor universal (bloque "Puertos de Entrada y Salida") por medios ópticos, de esta manera en caso de un sobrevoltaje o una sobrecorriente la PC queda protegida contra daños.

El bloque "Arbitraje del Bus de Datos y Lógica de Decodificación", se encarga de decodificar las líneas de direcciones y control provenientes del slot XT de la PC, para generar las líneas de selección de los puertos, además se dejan disponibles algunas señales de control y las líneas de alimentación del slot XT.

El diagrama general del medidor universal de variables eléctricas con todas las señales de control, se muestra en la figura 1.2.

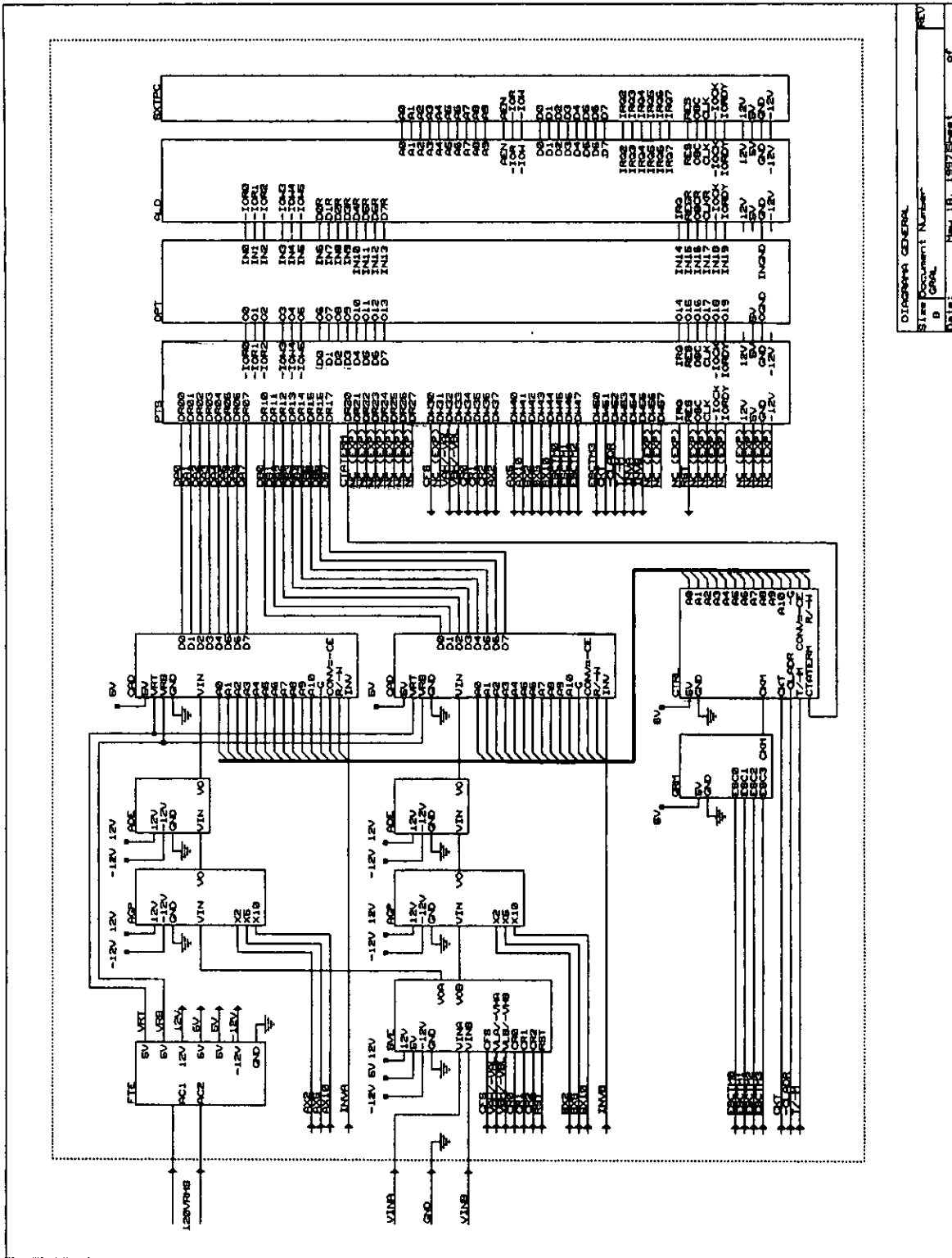


DIAGRAMA GENERAL

Sheet No. 1  
 Date: 18/12/1972  
 Author: G

Figura 1.2

## 1.2. SENSOR DE VARIABLES ELÉCTRICAS (SVE)

En este bloque es donde se detectan las variables voltaje, corriente, impedancia resistiva, impedancia capacitiva, impedancia inductiva, potencia real, potencia reactiva, potencia aparente y frecuencia (figura 1.3). Este bloque está compuesto a su vez por los siguientes bloques:

1. Fuente senoidal (SEN)
2. Resistencia programable (RES)
3. Divisores de voltaje programables (DIV)
4. Decodificación, acoplamiento y protecciones (DAP)
5. Relevadores (REL5V y REL12V)

Las variables eléctricas son convertidas a un voltaje reducido en el rango de -2.56 V a 2.56 V a través de divisores de voltaje y arreglos de resistencias controlados por relevadores, activados por señales digitales de control. Además se cuenta con elementos de protección como diodos y fusibles contra sobrecorriente y sobrevoltaje.

El sistema tiene tres bornes para el usuario: VINA (canal A), VINB (canal B) y GND, y se puede configurar el hardware de la siguiente manera:

- Los canales A y B para medir voltaje, de manera similar a un osciloscopio.
- El canal A para medir voltaje y el canal B para medir corriente.
- Para medir impedancia, entre el canal A y GND se coloca la impedancia a medir y el canal B no debe ser utilizado.
- El canal A para medir frecuencia.
- Para medir potencia, en el canal A se mide voltaje y en el canal B corriente.

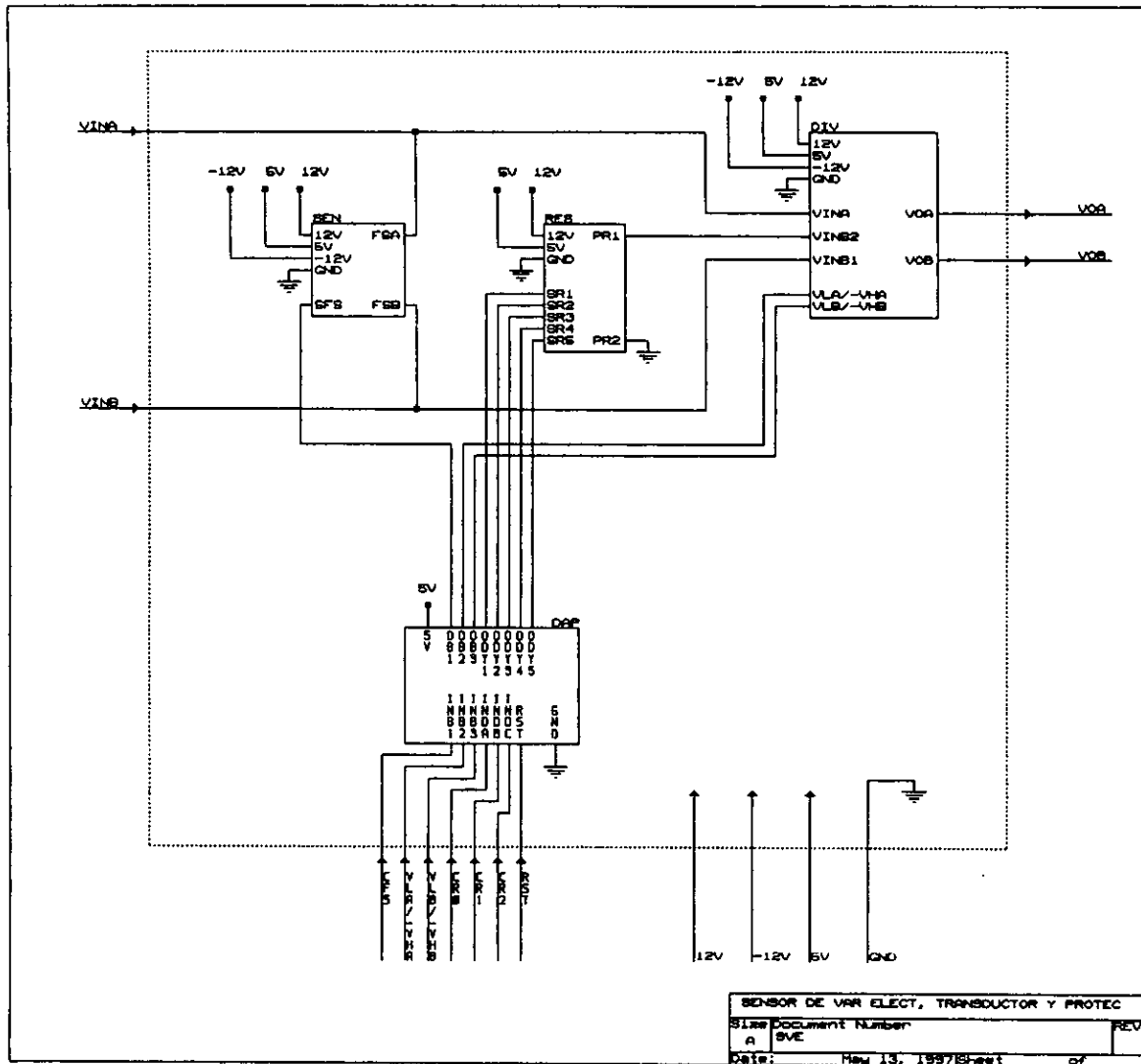


Figura 1.3

## Medición de voltaje

Cuando se mide voltaje en el canal A y en el canal B, la resistencia programable y la fuente senoidal son desconectadas por medio de relevadores para que no influyan en la medición de voltaje.

Si el voltaje de entrada es mayor de  $\pm 2.56$  V, se activa un divisor de voltaje a través de un relevador, de manera que el voltaje se divide entre 100. Si el voltaje

de entrada está entre el rango de  $-2.56\text{ V}$  a  $2.56\text{ V}$ , el divisor de voltaje no se activa.

### **Medición de corriente**

Se mide corriente solamente en el canal B. La fuente senoidal de voltaje esta desconectada mediante relevadores para que no influya en la medición. El divisor de voltaje para el canal B no está activo, así la corriente pasa directamente por la resistencia programable. Si la corriente es muy grande la resistencia programable es muy pequeña y viceversa, generándose un pequeño voltaje que el sistema detecta y que es proporcional a la corriente que esta circulando.

En un medidor de corriente la caída de potencial debe ser despreciable, por esta razón el mayor voltaje que se permite en la resistencia programable cuando esta funcionando en modo de medición de corriente es de  $0.7\text{ V}$ .

### **Medición de impedancia**

Para la medición de impedancia, se forma un circuito serie a través de relevadores, este circuito consiste de la fuente senoidal, de la resistencia programable (que es una impedancia puramente resistiva) y de la impedancia a medir. Como se conocen los parámetros de la fuente senoidal y de la resistencia programable, es posible calcular la impedancia a medir.

La fuente senoidal de voltaje es conectada al circuito por sus relevadores entre las terminales VINA y VINB, la impedancia a medir la conecta el usuario entre las terminales VINA y GND, internamente es conectada (a través de relevadores) entre VINB y GND la resistencia programable que toma un valor adecuado de acuerdo a los requerimientos de la medición.



Los divisores de voltaje tanto del canal A como del canal B no se establecen ya que la caída de voltaje es muy pequeña tanto en la impedancia a medir como en la resistencia programable.

### **Medición de potencia**

Para medir potencia en el borne VINA (canal A) se mide el voltaje y en el borne VINB (canal B) se mide la corriente. En el canal A el divisor de voltaje se puede establecer o no dependiendo de la magnitud del voltaje, en el canal B el divisor de voltaje no se establece, ya que el voltaje que se genera en la resistencia programable debido al paso de la corriente es muy pequeño. La fuente senoidal es desconectada por sus relevadores para que no influya en la medición de potencia.

Con el voltaje y la corriente se puede calcular por software la potencia aparente (S), potencia real (P), potencia reactiva (Q) y factor de potencia (fp). La deducción de las ecuaciones involucradas son detalladas en el apéndice II.

### 1.3. FUENTE SENOIDAL (SEN)

El bloque "Fuente Senoidal", que se muestra en la figura 1.4, tiene la función de generar una señal de excitación para el circuito serie cuando se mide una impedancia desconocida  $Z_A$ , si la frecuencia y la magnitud de la fuente senoidal de voltaje y la impedancia resistiva  $Z_B$  son conocidas, se puede conocer la impedancia  $Z_A$  en magnitud y ángulo de defasamiento.

En este bloque la señal de control digital SFS conecta o desconecta la fuente senoidal de voltaje a través de sus relevadores REL5V. Si la señal SFS vale 0V los relevadores REL5V permanecen en condición de desenergizados. Cuando SFS vale 5V se genera el suficiente voltaje para que los relevadores se energicen y la fuente senoidal queda conectada entre las terminales VINA y VINB.

La fuente senoidal de voltaje genera un voltaje de amplitud 1 V, este voltaje es reducido (pero no demasiado porque le afecta más el ruido) para no dañar componentes delicados. La fuente senoidal de voltaje debe tener a la salida un transformador 1:1 para desacoplar la señal de voltaje y no este referenciado a GND.

Se utiliza el XR-8038 como generador de onda senoidal, debido a que emplea un número mínimo de componentes externos, a su estabilidad en un amplio rango de temperatura, mínima distorsión de la señal en un amplio rango de frecuencia y ésta es independiente del suministro de voltaje, el análisis detallado del diseño con este generador se explica en el apéndice II.

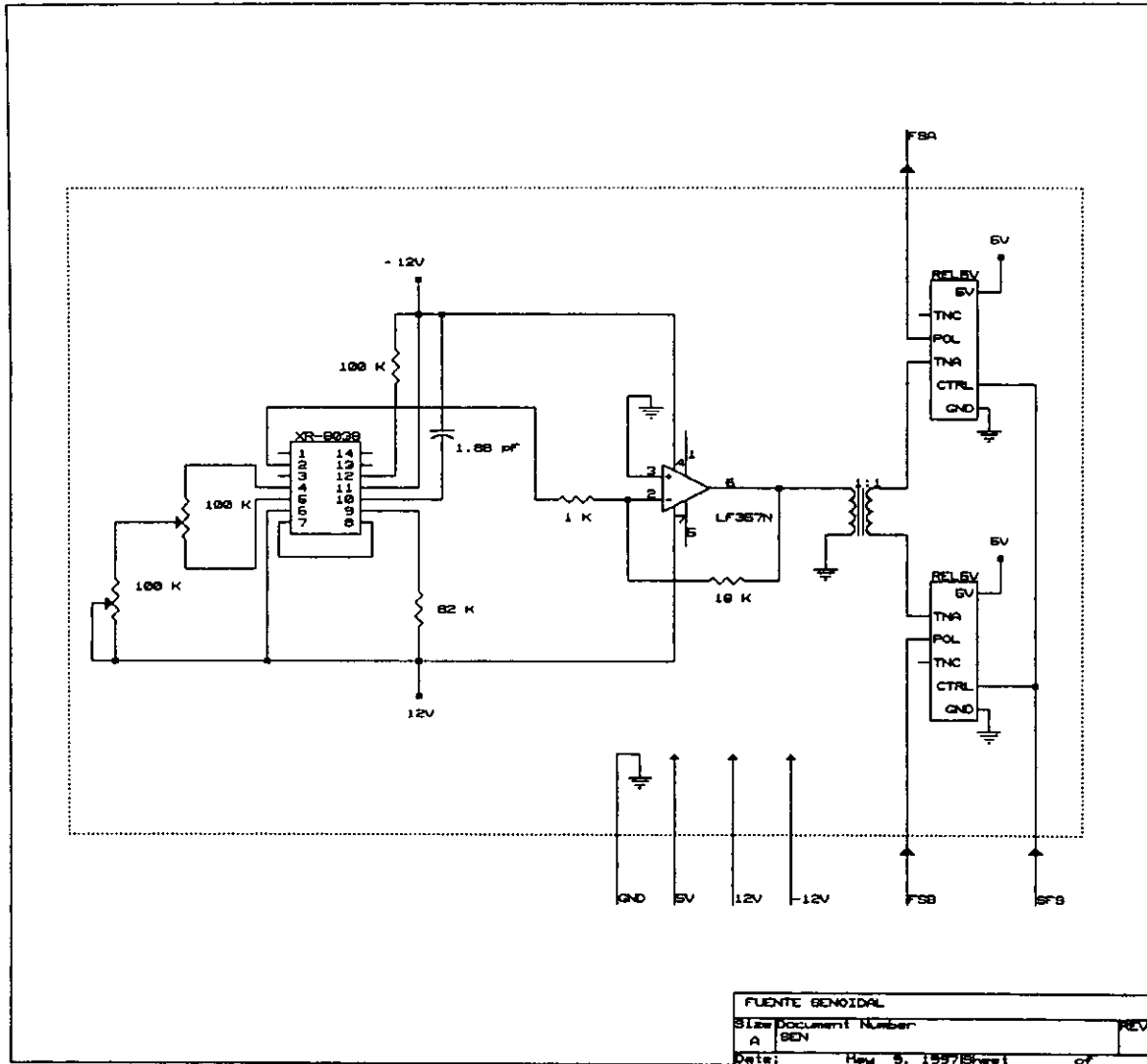


Figura 1.4

## 1.4. RESISTENCIA PROGRAMABLE (RES)

En el bloque "Resistencia Programable", el cuál se muestra en la figura 1.5, se selecciona un valor resistivo, a través de relevadores activados por señales de control digital en el canal B. Este bloque tiene las siguientes funciones:

- Seleccionar la resistencia conocida  $Z_B$  que se utiliza como referencia para medir una impedancia desconocida  $Z_A$
- Seleccionar una resistencia de valor adecuado para medir corriente, donde se genera una caída de voltaje que el sistema detecta y es una medida de la corriente que circula

La resistencia programable se selecciona a través de señales de control digital SR1, SR2, SR3, SR4, SR5 de acuerdo a la siguiente tabla:

SR5	SR4	SR3	SR2	SR1	RESISTENCIA ENTRE PR1 Y PR2
0	0	0	0	0	IDEALMENTE INFINITA
0	0	0	0	1	100 k $\Omega$
0	0	0	1	0	100 $\Omega$
0	0	1	0	0	10 $\Omega$
0	1	0	0	0	1 $\Omega$
1	0	0	0	0	0.1 $\Omega$

Cada resistencia se establece a través de su relevador correspondiente, por ejemplo: cuando se selecciona la resistencia de 100 k $\Omega$ , la señal digital SR1 vale 5V, el relevador se energiza y la resistencia de 100 k $\Omega$  queda conectada entre PR1 y PR2, mientras las señales de control SR2, SR3, SR4, SR5 están en cero lógico.

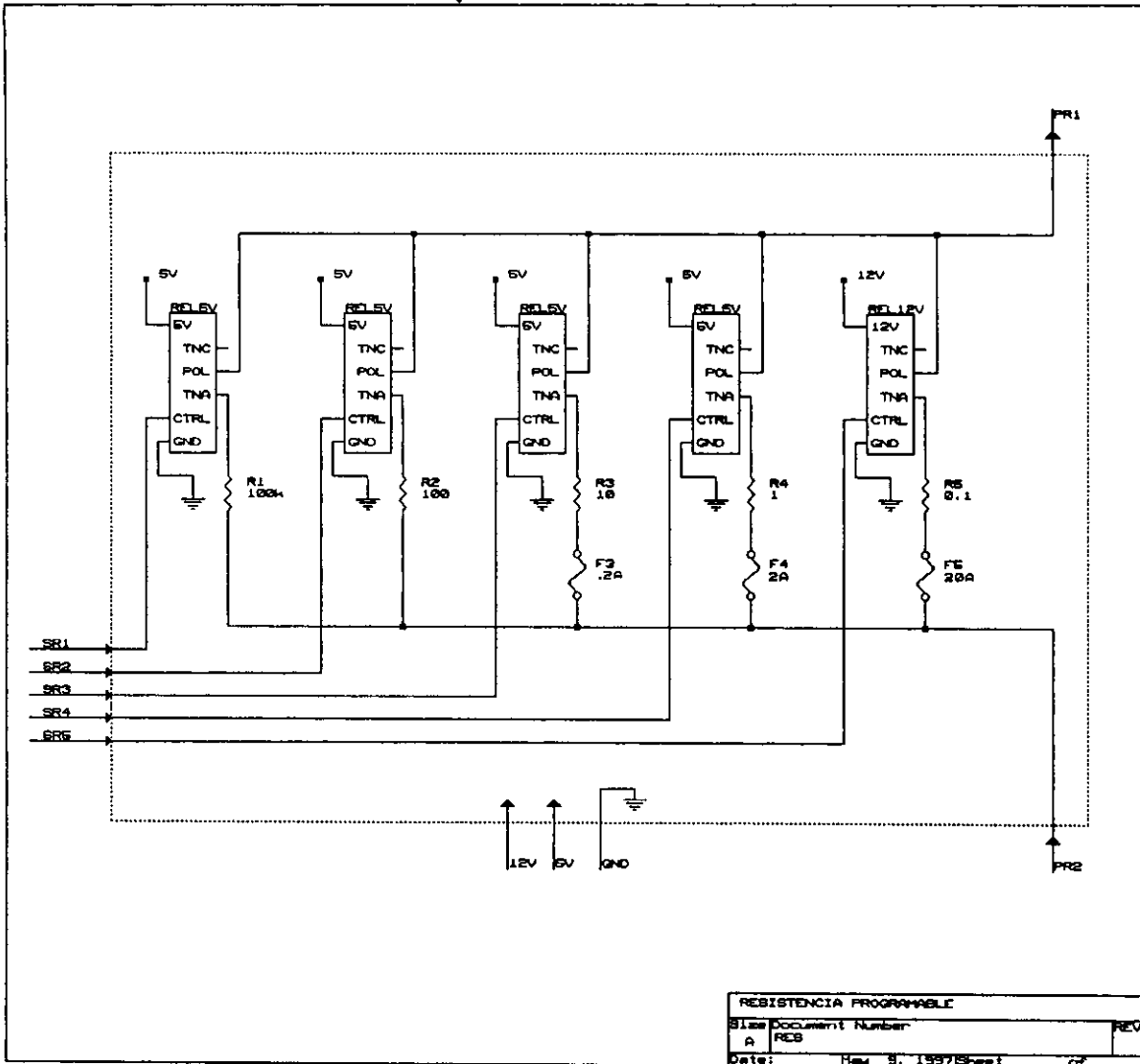


Figura 1.5

## 1.5. DIVISORES DE VOLTAJE PROGRAMABLES (DIV)

Los divisores de voltaje del canal A y B, que se muestran en la figura 1.6, tienen la finalidad de reducir el voltaje que se está midiendo entre 100, si este es "alto", es decir que su valor absoluto sea mayor a 2.56 V. Para voltajes "bajos" menores a 2.56 V, los divisores no están activos. Los divisores son establecidos a través

de relevadores controlados por las señales digitales VLA/-VHA y VLB/-VHB de manera que:

- Si VLA/-VHA = 0 lógico, entonces  $VOA = VINA/100$
- Si VLA/-VHA = 1 lógico, entonces  $VOA = VINA$
- Si VLB/-VHB = 0 lógico, entonces  $VOB = VINB/100$
- Si VLB/-VHB = 1 lógico, entonces  $VOB = VINB$

El rango operativo de las señales VOA y VOB es de -2.56 V a 2.56 V. Existen dos razones de este rango:

- Este valor es múltiplo de 256, que es el número de niveles que manejan los convertidores A/D, lo cual simplifica el manejo de las escalas.
- Por el "slew rate" de los amplificadores operacionales, es decir, con voltajes reducidos la frecuencia que puede alcanzar un amplificador operacional es mayor que con voltajes altos. Sin embargo, al disminuir mucho el voltaje existen problemas de ruido. Por lo tanto se tiene un compromiso entre el ruido y la velocidad, para el diseño de este proyecto el rango operativo de -2.56 V a 2.56 V se considero adecuado, lográndose alcanzar frecuencias de hasta 500 kHz, en señal senoidal, mientras que la señal cuadrada y la triangular se deforman un poco a esta frecuencia.

El rango absoluto de las señales VOA y VOB es de  $\pm 12$  V (sin embargo el sistema procesa información hasta  $\pm 2.56$  V). Cuando VOA y VOB están en el rango de  $\pm 12$  V, los diodos shottky están en corte, al elevarse más el voltaje los diodos entran en conducción aumentando la corriente y quemándose el fusible, con lo que el sistema queda protegido.

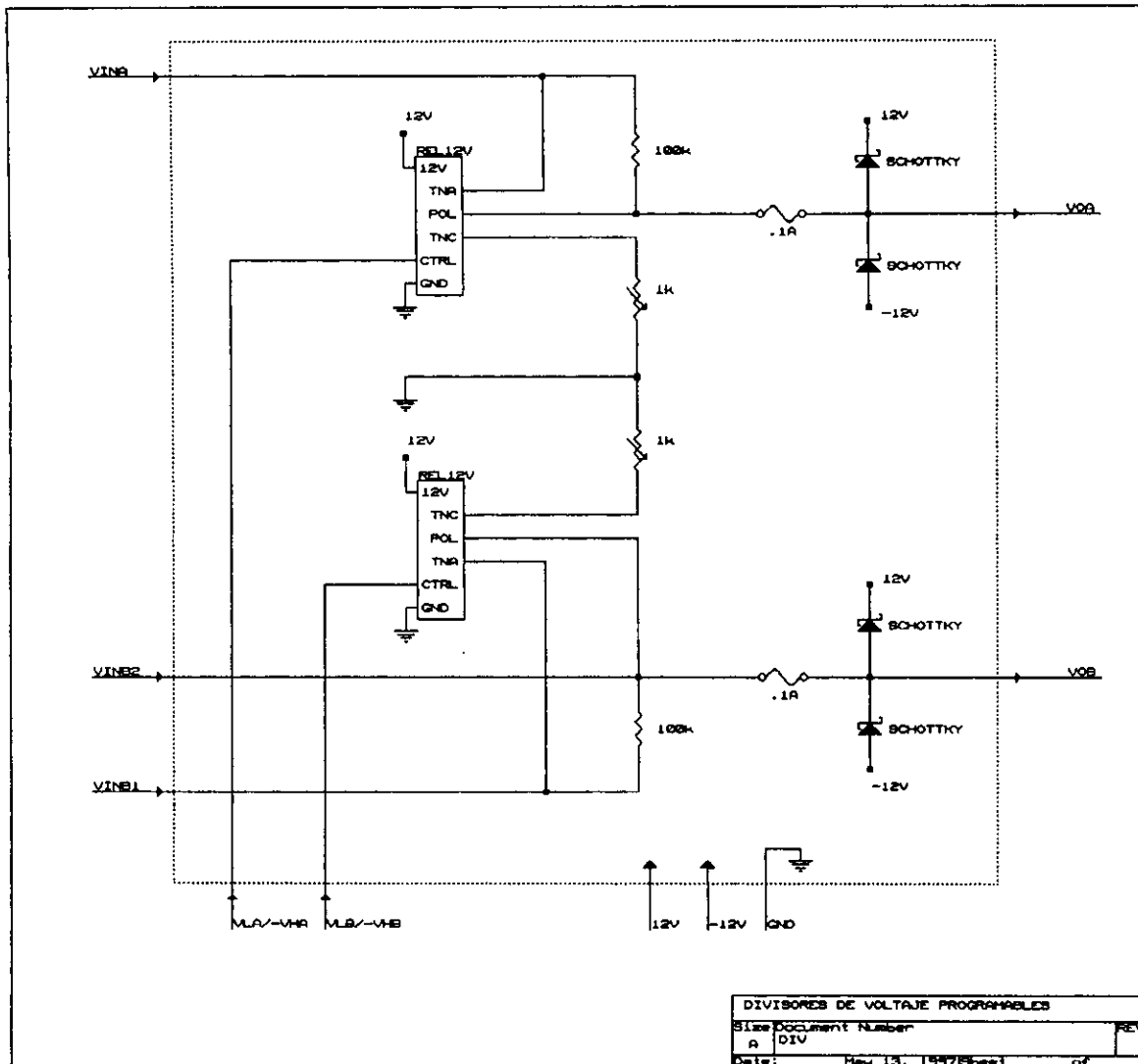


Figura 1.6

## 1.6. DECODIFICACIÓN, ACOPLAMIENTO Y PROTECCIONES (DAP)

Este bloque, que se muestra en la figura 1.7, tiene las funciones de decodificar y reforzar las señales de control de los bloques que conforman al bloque "Sensor de Variables Eléctricas", de inicializar el sistema y dejar los relevadores en un estado de protección en caso de una caída en el voltaje de alimentación.

Las señales INDC, INDB e INDA se decodifican en ODY1, ODY2, ODY3, ODY4, ODY5 a través de U3 estas señales son invertidas por U2 para que no se encuentren en lógica negada, de acuerdo a la siguiente tabla:

INDA	INDB	INDC	ODY1	ODY2	ODY3	ODY4	ODY5	RES PROG
0	0	0	0	0	0	0	0	INFINITA
1	0	0	1	0	0	0	0	100k $\Omega$
0	1	0	0	1	0	0	0	100 $\Omega$
1	1	0	0	0	1	0	0	10 $\Omega$
0	0	1	0	0	0	1	0	1 $\Omega$
1	0	1	0	0	0	0	1	0.1 $\Omega$
0	1	1	0	0	0	0	0	INFINITA
1	1	1	0	0	0	0	0	INFINITA

Todas las señales de entrada excepto RST son para el control de los relevadores, cuando hay una falla en la alimentación, es prioritario dejar los relevadores en un estado conocido, para esto existe un sensor de caída de voltaje que funciona de la siguiente manera.

Un potenciómetro esta conectado entre 5 V y GND generando un divisor de voltaje en el cursor, a casi el voltaje de umbral para que la entrada sea considerada alta por U4A y U4B, cuando hay una falla en la alimentación esta es detectada primeramente aquí, generando un 1 lógico momentáneo en la salida de las compuertas U4A y U4B, esto hace que U1 y U2 cambien al estado de alta impedancia haciendo ceros sus salidas a través de sus resistencias. Así en caso de falla en la alimentación todos los relevadores quedan en un estado conocido o de protección, que es el desenergizado.



El estado de protección es el siguiente:

- Los divisores de voltaje quedan activados
- El resistor programable se establece en infinito
- Las fuentes senoidales quedan desconectadas

Cuando el sistema se prende por primera vez no se tiene la certeza del estado de los puertos que controlan a los relevadores, por esta razón el bloque DAP no acepta su información hasta que tiene seguridad de que el programa del proyecto se ha inicializado. Por esto, desde software, al iniciar el programa se debe dar una clave de inicialización para que desde los puertos se puedan controlar los relevadores, esta clave es  $INB1=0$ ,  $INB2=0$ ,  $INB3=0$ ,  $INDA=1$ ,  $INDB=1$ ,  $INDC=1$ , llamemos a esta clave palabra de inicialización.

El integrado U5 (74LS688) es un comparador en el cual si  $P0=Q0$ ,  $P1=Q1$ ,  $P2=Q2$ ,  $P3=Q3$ ,  $P4=Q4$ ,  $P5=Q5$ ,  $P6=Q6$ ,  $P7=Q7$  entonces su salida  $P=Q$  es válida (válida en nivel bajo), con cualquiera que sea diferente entonces  $P=Q$  no es válida (permanece en nivel alto). La entrada P de este comparador se conecta de la siguiente forma P0 a INB1, P1 a INB2, P2 a INB3, P3 a INDA, P4 a INDB, P5 a INDC, P6 a GND y P7 a GND. Por otra parte la entrada Q tiene la palabra de inicialización dada por el dip switch, así  $Q0=0$ ,  $Q1=0$ ,  $Q2=0$ ,  $Q3=1$ ,  $Q4=1$ ,  $Q5=1$ ,  $Q6=0$ ,  $Q7=0$ . Cuando se da la palabra de inicialización entonces  $P=Q$  es válido.

El flip-flop tipo D, U6A, esta en cero desde que se prendió la máquina, ya que RST (que se hace válido siempre que se prende la máquina) limpió a este flip-flop, este estado se conserva hasta que el reloj cargue con un uno a U6A lo cual pasa justamente cuando se da la clave de inicialización.

Si la salida Q del flip-flop es cero los buffers de 3 estados U1 y U2 están en alta impedancia y por lo tanto los relevadores están en estado protegido. Cuando se da la clave de inicialización Q cambia a 1 y los buffers de 3 estados, U1 y U2, dejan el estado de alta impedancia y ahora los relevadores pueden ser controlados por los puertos. Así los relevadores permanecen en un estado de protección hasta que se de la palabra de inicialización.

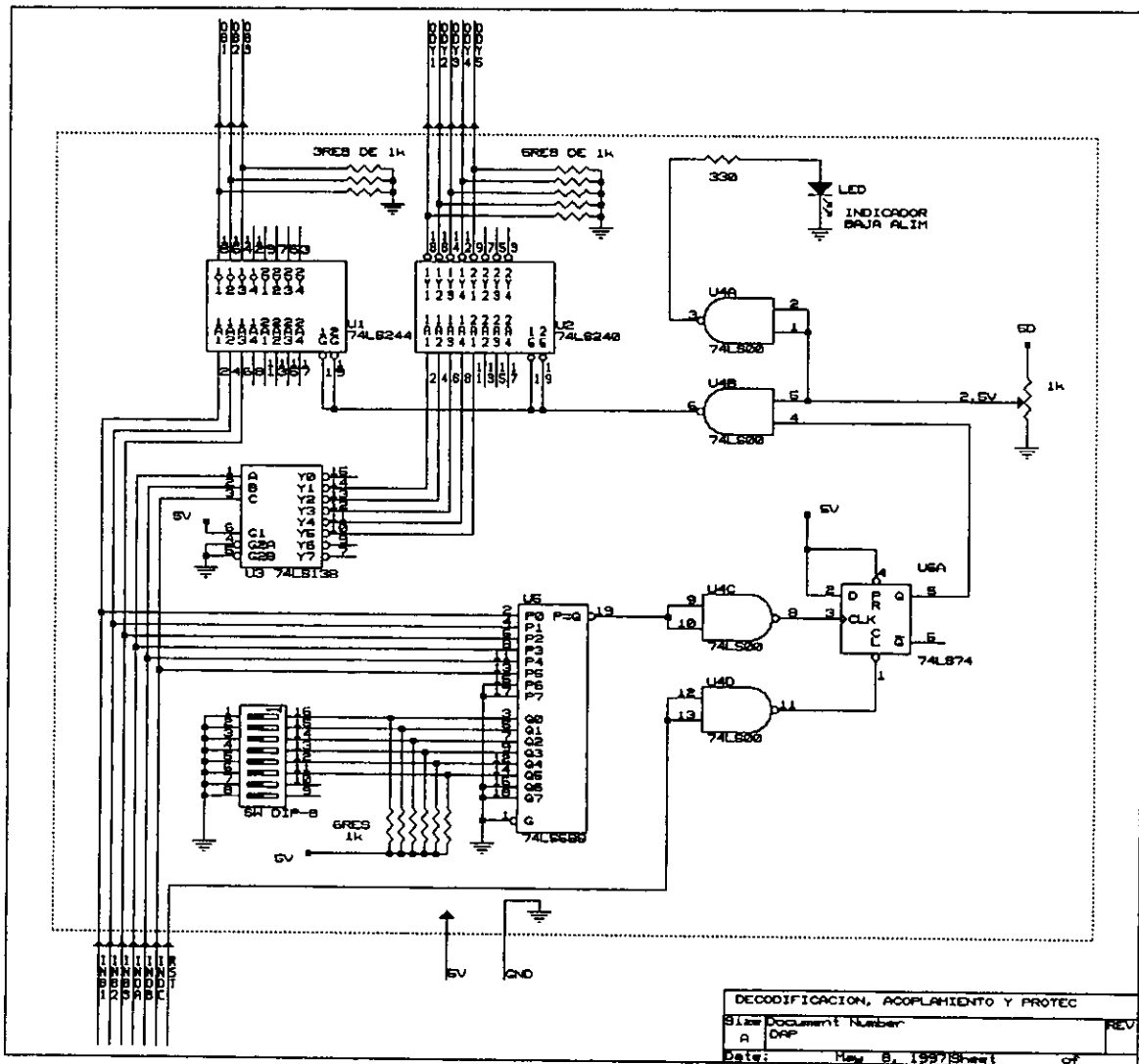


Figura 1.7

## 1.7. RELEVADORES (REL5V Y REL12V)

Los relevadores (figura 1.8) son utilizados para seleccionar la resistencia programable, conectar o desconectar los divisores de voltaje y la fuente senoidal de voltaje.

La base del transistor es la entrada de la señal digital de control CTRL. Si la señal de control CTRL es 0 V, la corriente en la base es 0 A, el transistor está en corte, la corriente en el colector es 0 A y los relevadores permanecen desenergizados, con lo que TNC permanece conectado a POL y TNA permanece abierto.

Si la señal de control CTRL es 5 V, la corriente en la base es aproximadamente 5 mA, el transistor está en saturación, la corriente en el colector es suficiente para que los relevadores se energicen, con lo que TNC queda abierto y TNA conectado a POL.

Para el bloque REL5V, el relevador requiere que el voltaje en la bobina sea de 5V, entre el polo y los tiros puede manejar 1A (HD-RELAY). Para el bloque REL12V, el relevador requiere que el voltaje en la bobina sea de 12V, entre el polo y los tiros puede manejar hasta 40A (HB-RELAY).

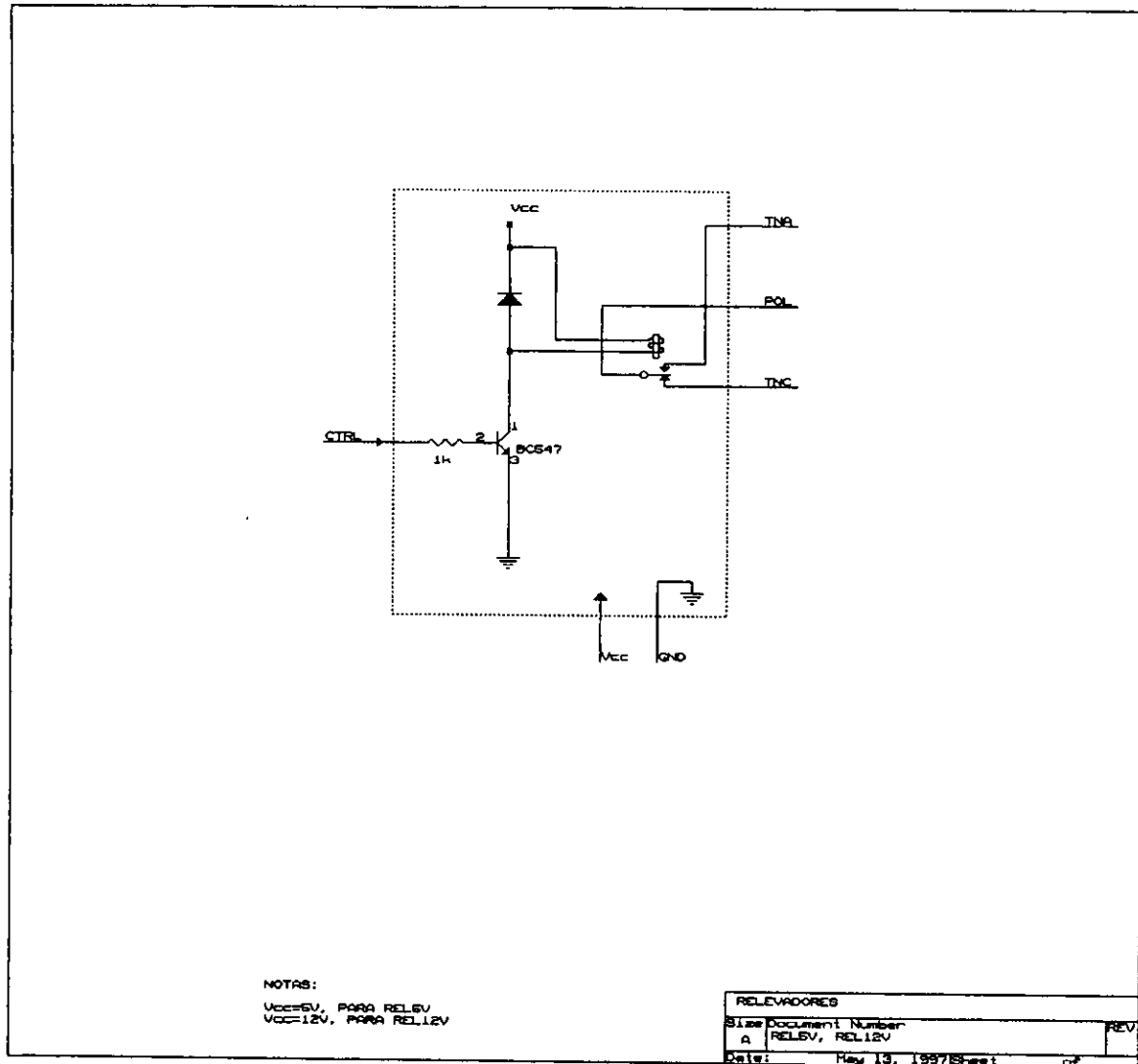


Figura 1.8

## 1.8. AMPLIFICADOR DE GANANCIA PROGRAMABLE (AGP)

En este bloque (figura 1.9) se amplifica la señal VIN ya sea por 1, 2, 5, 10, 20, 50 o 100. La ganancia se controla mediante las señales de control digital X2, X5, X10 de acuerdo con la siguiente tabla:

X10	X5	X2	GANANCIA
0	0	0	1
0	0	1	2
0	1	0	5
0	1	1	10
1	0	1	20
1	1	0	50
1	1	1	100

El funcionamiento se basa en lo siguiente:

Los amplificadores operacionales identificados como U2, U3, U4 están en configuración de inversora, la ganancia de cada uno de ellos depende de sus resistencias de realimentación y de sus resistencias de entrada, si las resistencias de realimentación se mantienen fijas, la ganancia de cada configuración inversora dependerá de las resistencias de entrada, si éstas cambian por medio de switches analógicos, se tendrá un amplificador de ganancia programable.

El circuito integrado U5 tiene 4 switches analógicos (uno de ellos, el switch 4, no se utiliza), el switch 1 controla que la ganancia sea -1 o -5 en el inversor cuyo operacional es U3, el switch 2 controla que la ganancia sea -1 o -2 en el inversor cuyo operacional es U4 y el switch 3 controla que la ganancia sea -1 o -10 en el inversor cuyo operacional es U2.

El amplificador inversor que incluye a U4, amplifica por -1 o por -2 dependiendo del estado del switch analógico 2, de acuerdo a lo siguiente:

El amplificador inversor que incluye a U4, amplifica por -1 o por -2 dependiendo del estado del switch analógico 2, de acuerdo a lo siguiente:

Cuando X2 (que esta conectado a IN2 de U5) vale 0 lógico, el switch constituido por S2 y D2 estará abierto y en la entrada de la configuración inversora se tendrá una resistencia de 10 k $\Omega$ , como la resistencia de realimentación también es de 10 k $\Omega$ , la ganancia es -1.

Cuando X2 vale 1 lógico, este switch estará cerrado poniendo en paralelo dos resistencias de 10 k $\Omega$  formándose una de 5 k $\Omega$  en la entrada, como ya se menciono, la resistencia de realimentación es de 10 k $\Omega$ , por lo tanto se tendrá una ganancia de -2 en esta etapa.

La resistencia conmutada por el switch analógico es variable y de ajuste fino, por dos razones: El switch analógico tiene una pequeña resistencia en estado de conductividad (aproximadamente de 200  $\Omega$ ) y para ajustar a un valor exacto la ganancia, ya que también hay que tomar en cuenta la tolerancia de las demás resistencias que no son variables.

Los inversores cuyos operacionales son U3 y U2 funcionan de manera similar a lo anterior, cuando sus respectivos switches están abiertos sus ganancias son -1 y cuando están cerrados estas ganancias son -5 y -10 respectivamente.

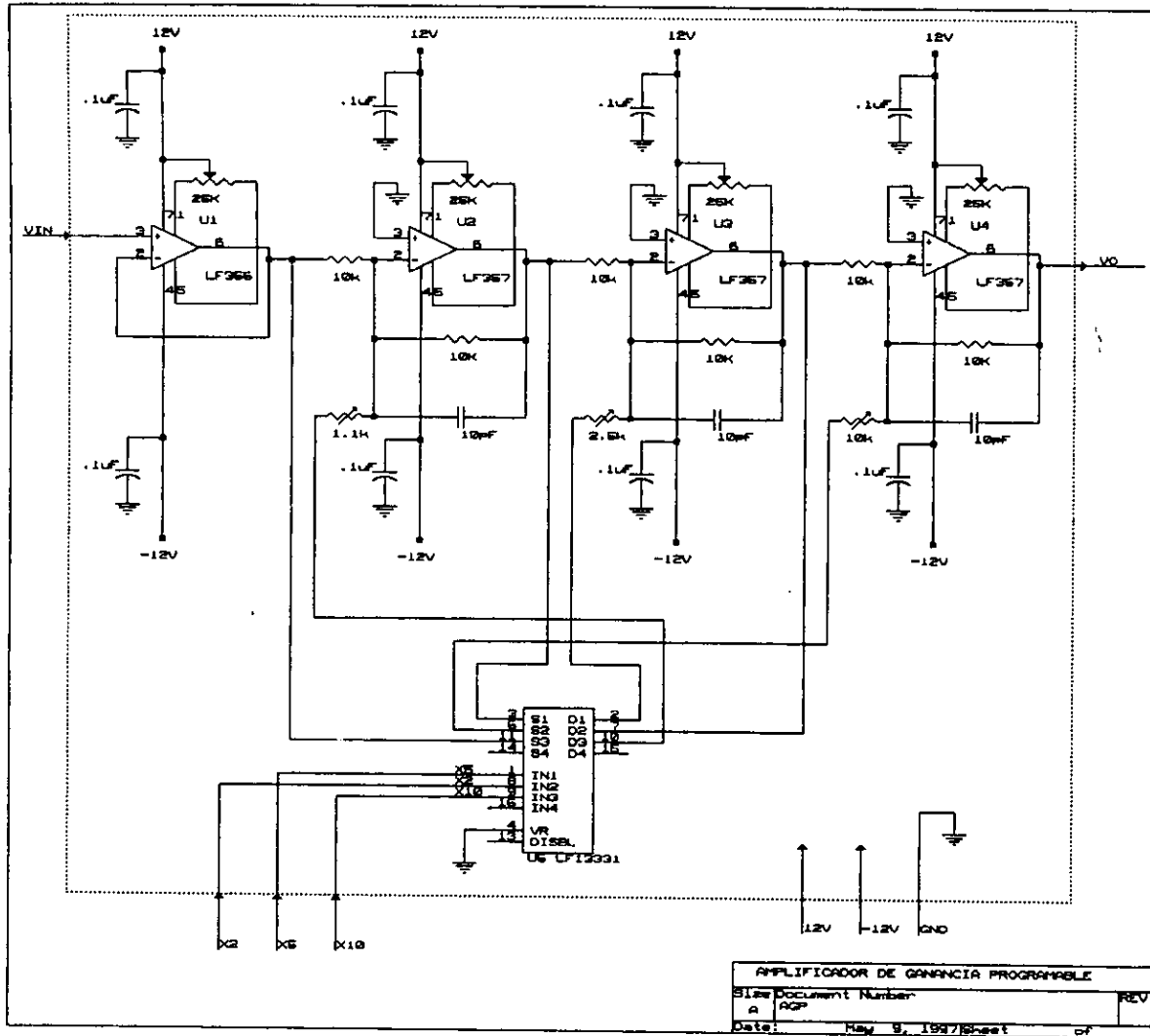


Figura 1.9

## 1.9. ADECUACIÓN PARA EL CAD (ADE)

La señal de entrada (ver figura 1.10) a este bloque VIN, se adecua de acuerdo a los requerimientos de la entrada analógica del CAD, que debe estar en el rango de 3V a 5V, protegida y libre de ruido en lo posible.

La señal VIN esta limitada a su rango operativo (-2.56V a 2.56V) a través de dos diodos en paralelo conectados a +2 V y -2 V, esto con fines de protección contra sobrevoltajes. Posteriormente en la configuración inversora, que incluye a U3, la señal es adecuada para que a su salida se tenga un rango de 3 V a 5 V. Para

señal es adecuada para que a su salida se tenga un rango de 3 V a 5 V. Para lograr esto, a  $V_{IN}$  se suma con 4 V y se multiplica por  $-1/2.56$ .

La señal de salida de U3, es filtrada antes de pasar al CAD, se utiliza un amplificador de instrumentación, el cual tiene como características: precisión, estabilidad y alta inmunidad al ruido. Este está formado por los amplificadores operacionales U4, U5 y U6. En la entrada no inversora del amplificador U4 se conecta la salida de U3 y en la entrada no inversora de U5 se conecta a GND, para eliminar el ruido en modo común que pudiera afectar la medición.

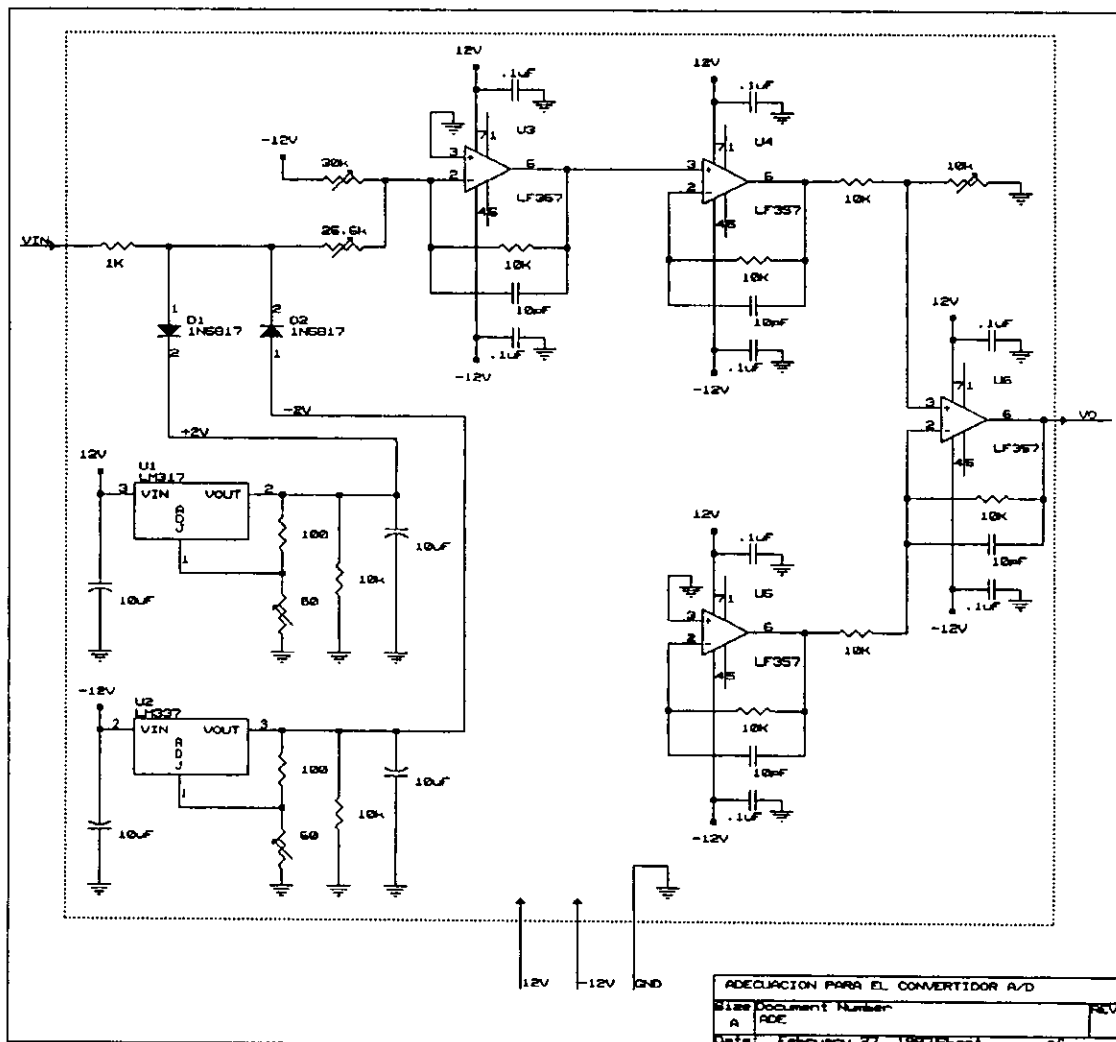


Figura 1.10



## 1.10. GENERACIÓN DEL RELOJ DE MUESTREO (GRM)

Este bloque, que se muestra en la figura 1.11, tiene la función de generar una señal de reloj de muestreo para las diferentes escalas de tiempo, a partir de una frecuencia base de 20 MHz. Para dividir la frecuencia se utilizaron 6 contadores módulo 10 conectados en cascada y 5 flip-flops JK, obteniéndose 16 diferentes frecuencias o escalas de tiempo que son las entradas de un multiplexor de 16 a 1 (0.1 $\mu$ s, 0.2 $\mu$ s, 0.5 $\mu$ s, 1 $\mu$ s, 2 $\mu$ s, 5 $\mu$ s, 10 $\mu$ s, 20 $\mu$ s, 50 $\mu$ s, 0.1ms, 0.2ms, 0.5ms, 1ms, 2ms, 5ms, 10ms). La selección de una de las escalas de tiempo es a través de 4 señales de control provenientes de los puertos de la PC.

En un osciloscopio las escalas de tiempo se incrementan de una manera aproximadamente exponencial, lo cual permite abarcar un amplio rango de medición, este incremento es aproximadamente una octava, es decir, que el valor de la escala siguiente se duplica, esto es, si se tiene inicialmente una escala unitaria, la siguiente escala será 2, la siguiente 5, la siguiente 10, la siguiente 20 y así sucesivamente. Los incrementos anteriores no son exactamente en octavas por que 1, 2, 5, 10, 20, ... son números redondos que permiten un cálculo más fácil que si fueran 1, 2, 4, 8, 16, 32, 128, etc. Para el diseño de este proyecto se considero lo anterior, no sólo para las escalas horizontales (de tiempo), sino también para las verticales (de voltaje y de corriente).

El circuito oscilador esta formado por dos inversores 74F00 (U11C y U11D) realimentados y conectados con un oscilador de cristal de 20 MHz, aquí la señal tiene una forma parecida a una senoidal, así pasa por un inversor, el cual la cuadra un poco, la frecuencia sigue en 20 MHz y el periodo en 0.5 $\mu$ s, se necesita dividir la frecuencia entre 2, 4 y 10. La señal entra a un contador sincrónico identificado como U5 (se utilizó este tipo de contadores, ya que permite trabajar a más altas frecuencias que los contadores asincrónicos), se divide la frecuencia entre 2 en QA, entre 10 en QC, para dividir entre 4 la frecuencia se utiliza la señal de QA de los contadores como entrada de reloj en los flip-flops JK, en cada una de estas salidas el ciclo de trabajo es del 50%, es decir, la parte baja es igual a la parte alta, en las siguientes etapas el funcionamiento es similar.

de QA de los contadores como entrada de reloj en los flip-flops JK, en cada una de estas salidas el ciclo de trabajo es del 50%, es decir, la parte baja es igual a la parte alta, en las siguientes etapas el funcionamiento es similar.

Si se maneja el periodo en lugar de la frecuencia, el periodo se multiplica en vez de dividirse, obteniéndose diferentes bases de tiempo que constituyen las escalas, de acuerdo a la siguiente tabla:

CI IDENTIFICACION	SALIDA	ESCALA
U5	QA	0.1 $\mu$ s
U2B	Q	0.2 $\mu$ s
U5	QC	0.5 $\mu$ s
U6	QA	1 $\mu$ s
U2A	Q	2 $\mu$ s
U6	QC	5 $\mu$ s
U7	QA	10 $\mu$ s
U3B	Q	20 $\mu$ s
U7	QC	50 $\mu$ s
U8	QA	0.1ms
U3A	Q	0.2 ms
U8	QC	0.5 ms
U9	QA	1 ms
U4B	Q	2 ms
U9	QC	5 ms
U10	QA	10 ms

Lo anterior son los periodos de muestreo, por ejemplo: si se tiene la escala de 0.1 $\mu$ s, las muestras se tomaran a intervalos de 0.1 $\mu$ s. Cuando se visualiza la señal en el monitor de la computadora cada muestra representa un pixel, así por ejemplo: se tienen 100 pixeles por división, en esta misma escala de muestreo se tendrán:

$$100 \frac{\text{pixeles}}{\text{division}} * 0.1 \frac{\mu\text{s}}{\text{pixeles}} = 10 \frac{\mu\text{s}}{\text{division}}$$

$$100 \frac{\text{píxeles}}{\text{division}} * 0.1 \frac{\mu\text{s}}{\text{píxeles}} = 10 \frac{\mu\text{s}}{\text{division}}$$

Nota: un píxel es equivalente a una muestra.

Cada una de las salidas anteriores entran a un multiplexor 16:1 (identificado como U1) con lo cual se abarcan las 16 escalas, cuyas cuatro líneas de selección A, B, C, D seleccionan la escala de tiempo deseada, de acuerdo con la siguiente tabla:

ESC3	ESC2	ESC1	ESC0	ESCALA X PIXEL
0	0	0	0	0.1μs
0	0	0	1	0.2μs
0	0	1	0	0.5μs
0	0	1	1	1μs
0	1	0	0	2μs
0	1	0	1	5μs
0	1	1	0	10μs
0	1	1	1	20μs
1	0	0	0	50μs
1	0	0	1	0.1ms
1	0	1	0	0.2 ms
1	0	1	1	0.5 ms
1	1	0	0	1 ms
1	1	0	1	2 ms
1	1	1	0	5 ms
1	1	1	1	10 ms

ESC3, ESC2, ESC1, ESC0 es el código proveniente de un puerto de la PC para seleccionar una determinada escala.

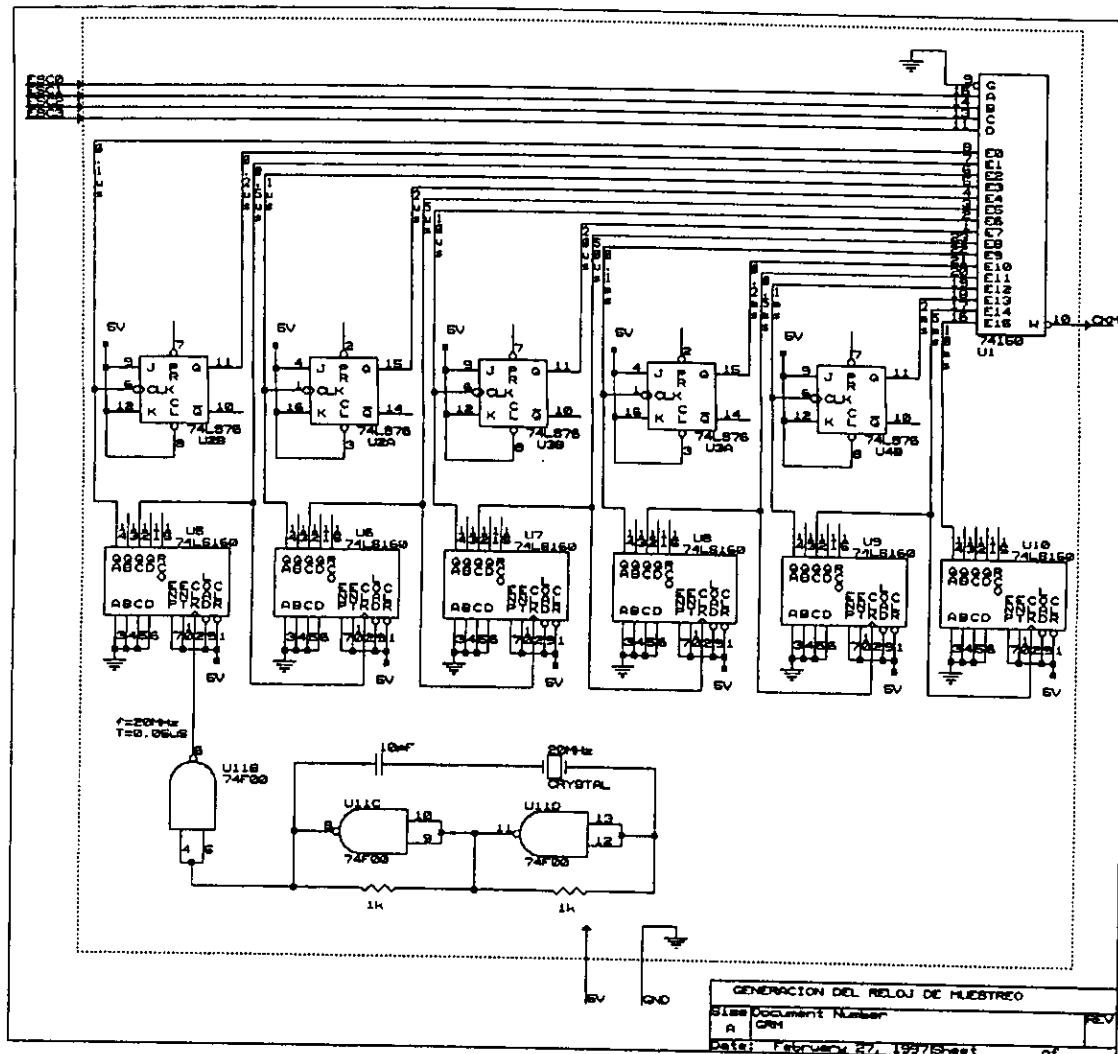


Figura 1.11

## 1.11. CONTROL (CTRL)

Este bloque, que se muestra en la figura 1.12, tiene la función de generar las señales de control para los bloques "CAD y almacenamiento temporal" y para la PC.

Para la siguiente explicación es necesario ver también la figura 1.14, la cual muestra el bloque CAD (a menos que se indique lo contrario, las identificaciones de los circuitos integrados a los que se hace referencia, son los del bloque CTRL).

Inicialmente, se establece el modo de muestreo ( $T/M = 0$ ), en el cual, el CAD va a

Inicialmente, se establece el modo de muestreo ( $T/-M = 0$ ), en el cual, el CAD va a escribir datos en la RAM (U3 del bloque CAD) a una velocidad dada por el reloj de muestreo (CKM). Primero la RAM (U3 del bloque CAD) debe ser habilitada como entrada y después el buffer de 3 estados (U2 del bloque CAD) deja el estado de alta impedancia, éste orden es con el objeto de evitar cortos circuitos. Lo anterior se detalla a continuación.

Con  $T/-M$  a cero,  $R/-W$  se hace cero a través de la AND, U6A, con un retardo provocado por esta compuerta, habilitando a la RAM para que le sean escritos datos. Después de que la señal  $T/-M$  pasa por un retardo mayor (a través de 3 compuertas AND: U6B, U6C, U6D y 3 OR: U7B, U7C, U7D) es  $RT/-RM$ , esto hace, a través de la compuerta OR, U7A, que  $-G$  sea cero con un retardo provocado por 7 compuertas, con esto el buffer de 3 estados (U2 del bloque CAD) deja el estado de alta impedancia. Los retardos anteriores son provocados para evitar cortos circuitos momentáneos entre el buffer de 3 estados y la memoria RAM.

$RT/-RM$  (que es la señal retardada de  $T/-M$ ) es la entrada de selección del multiplexor U4, así cuando se esta en modo de muestreo, MUXSAL esta dada por CKM.

Al pasar MUXSAL y CTATERM a través de la NOR, U5B, se genera el reloj (CKCON) de los contadores, cuya cuenta es la dirección de memoria de la RAM, la cuenta se incrementa hasta que llegue a un valor predeterminado (dado a través de los DIP SW), cuando se llega a éste valor, CTATERM se activa, bloqueando CKCON para que el conteo termine.

Al pasar MUXSAL y  $RT/-RM$  a través de la NOR, U5C, se genera  $CONV=-CE$  que es el reloj del convertidor A/D y la señal de habilitación de la RAM. Cuando  $CONV=-CE$  es cero, la RAM se habilita para leer la información que le manda el CAD, éstos datos son válidos justamente cuando se habilita la RAM ( $CONV=-CE = 0$ ) almacenándose correctamente en esta. El propósito de la NOR es dejar siempre habilitada a la RAM cuando se este en el modo de transmisión ( $T/-M = 1$ ), esto permite que la PC lea a esta en cualquier momento, ya que la PC y el hardware exterior no están sincronizados a través de algún reloj.

sincronizados a través de algún reloj.

Así en el modo de muestreo la memoria se va "llenando" con la información proporcionada por el CAD a una velocidad dada por el reloj de muestreo (CKM). Cuando la memoria se "lleno" hasta un valor predeterminado (128, 256, 512 o 1024 muestras/imagen) dado por el DIP SW, CTATERM se activa (se hace 1 lógico), bloqueándose CKCON no incrementándose mas las direcciones de la RAM (ADRMEM).

A CTATERM lo esta sensando continuamente la PC (a través de poleo), y cuando detecta que esta activo, la PC reconoce que la memoria ya tiene los datos de la imagen completa y que por lo tanto el modo de muestreo ha terminado. Con lo cual la PC inicializa el proceso que conforma el modo de transmisión, en el cual la memoria va a ser leída por la PC transmitiendo a esta su información, para esto el buffer de 3 estados tiene que estar en estado de alta impedancia, con el fin de evitar un corto circuito con los datos que esta mandando la RAM.

Al pasar del modo de muestreo al de transmisión primero el buffer de 3 estados debe pasar al estado de alta impedancia y después la memoria puede dar información o ser leída, de lo contrario se dará un corto circuito.

Así la PC hace que se active el modo de transmisión haciendo  $T/-M = 1$ , con esto se hace que  $-G$  pase a 1 lógico (a través de la compuerta OR, U7A) dejando en estado de alta impedancia al buffer de tres estados, después de un retardo provocado por las 3 AND y las 3 OR,  $RT/-RM$  pase también a 1, con lo cual  $R/-W$  se establece en 1 (a través de la compuerta AND) con un retardo de 7 compuertas, quedando la RAM en modo de lectura.

Después la PC hace válido  $-CLADR$  estableciéndolo en 0 lógico,  $RT/-RM$  (que es la señal retardada de  $T/-M$ ) es la entrada de selección del multiplexor, cuando se esta en modo de transmisión, MUXSAL esta dada por CKT.

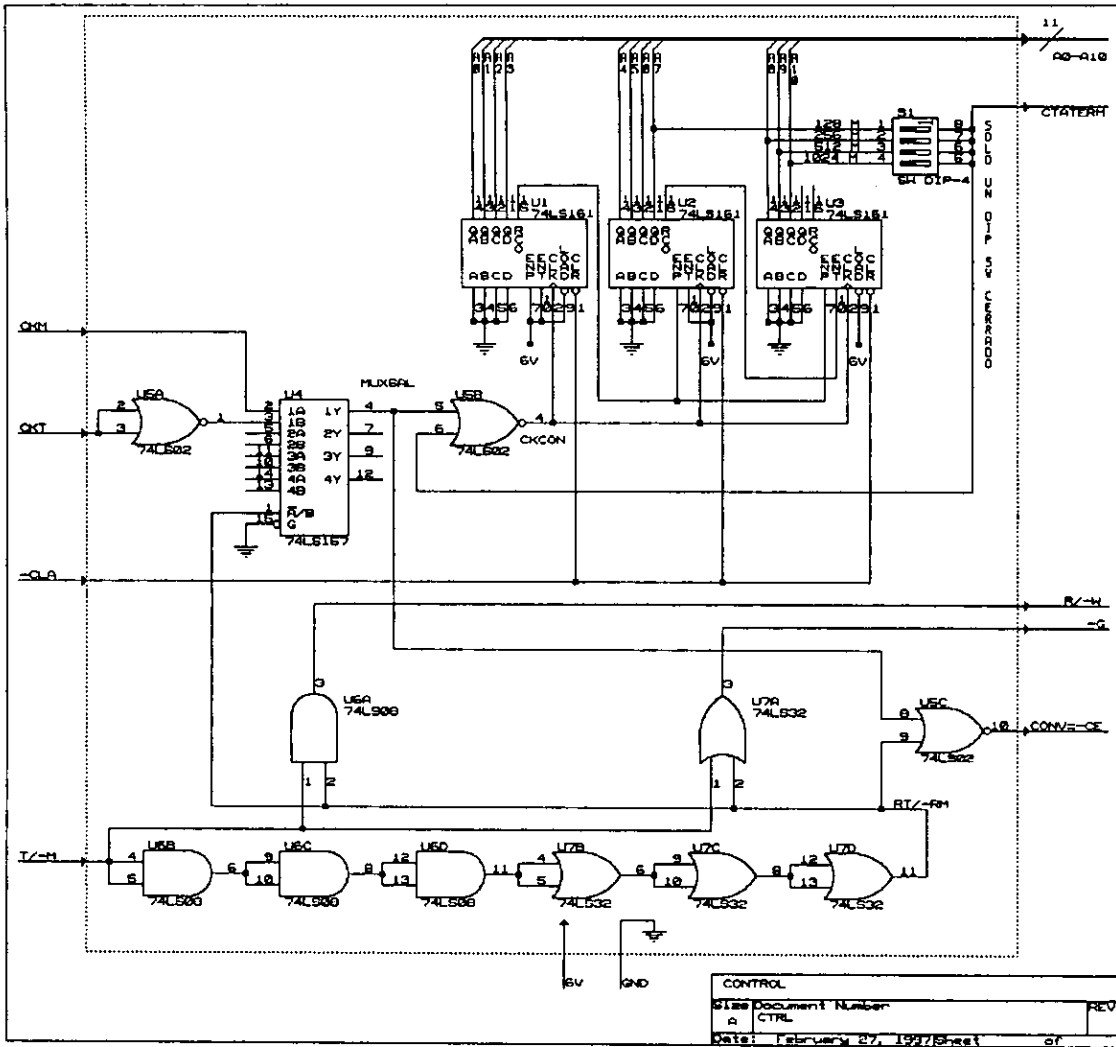


Figura 1.12

## 1.12. FUENTE DE ALIMENTACIÓN (FTE)

El bloque "Fuente de Alimentación" (figura 1.13) tiene la función de proporcionar los voltajes que necesitan las diversas tarjetas del sistema. Esta constituido por un transformador de 24 V a 3 A, 4 reguladores variables positivos LM317T, 1 regulador variable positivo LM317K, 1 regulador negativo LM337T, 2 capacitores de 5000  $\mu\text{f}$ , capacitores de 0.1  $\mu\text{f}$  y resistencias variables de ajuste fino (trimpots).

El primario del transformador se conecta a la línea de alimentación de 120  $V_{\text{RMS}}$ , en el secundario se generan 24  $V_{\text{RMS}}$ , incluyendo el TAP central (conectado a GND) se tienen en las terminales  $\pm 12 V_{\text{RMS}}$  aproximadamente, estos voltajes se rectifican a través de un puente de diodos de onda completa, cada voltaje se filtra con un capacitor de 3300  $\mu\text{f}$  conectado entre cada terminal y GND, dando como resultado dos niveles de corriente directa de +17 V y -17 V aproximadamente, con un cierto voltaje de rizo sobrepuesto, para reducir este se utilizan capacitores de 0.1  $\mu\text{f}$ .

Para obtener los voltajes 3 V, 5 V, 6 V, 12 V, se utiliza el voltaje +17 V como voltaje de entrada para los reguladores LM317T que proporcionan una corriente de hasta 1.5 A, el voltaje de salida se ajusta a través de una resistencia variable de ajuste fino para mayor precisión, también se colocan capacitores en las terminales de entrada y salida de cada regulador para eliminar las componentes de AC que pudieran existir. Además se necesita otro voltaje de +5 V para energizar varias tarjetas, se utiliza el LM317K ya que proporciona mayor corriente y mayor disipación de calor. Para obtener el voltaje de -12 V, se utiliza -17 V como voltaje de entrada para el regulador LM337T, el cuál proporciona una corriente de 1.5 A, el voltaje de salida se ajusta a través de una resistencia variable de ajuste fino. Se colocan capacitores entre las terminales de entrada y salida del regulador para eliminar las componentes de AC que pudieran existir.



Para tener una mayor estabilidad en circuito abierto, se colocan resistencias de carga de 10 K $\Omega$  a la salida de cada referencia de voltaje.

Se utilizaron estos reguladores de voltaje por sus características, su funcionamiento aceptable, bajo costo y pocos elementos externos.

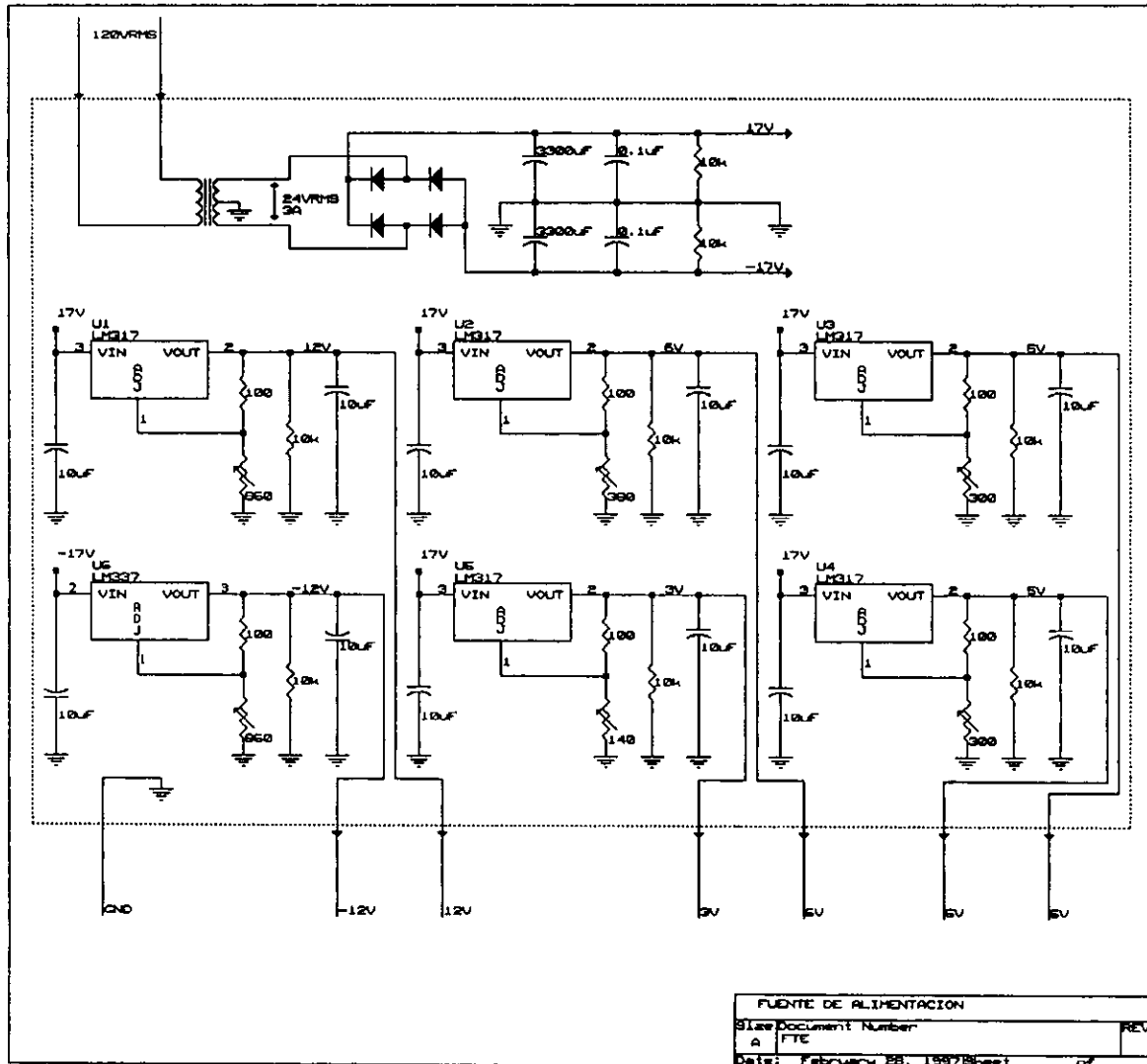


Figura 1.13

## 1.13. CAD Y ALMACENAMIENTO TEMPORAL (CAD)

En este bloque, que se muestra en la figura 1.14, las señales analógicas del canal A y el canal B ya acondicionadas, son convertidas a señales digitales a través del convertidor analógico digital, se almacenan en memoria RAM y posteriormente son transferidas hacia la PC. Básicamente se realizan dos procesos:

- Primero: consiste en el almacenamiento temporal de esta señal digitalizada en memoria RAM, aquí los datos a la salida del convertidor analógico digital se van almacenando en tiempo real en la memoria RAM, hasta que se llegue a un número predeterminado de muestras. La velocidad de almacenamiento esta dado por la señal de reloj de muestreo, el cual es función de las escalas de tiempo. Lo anterior constituye lo que se llama modo de muestreo.
- Segundo: cuando ya se llegó al número establecido de muestras, la información contenida en la memoria RAM es "vacuada a la PC" a través de los puertos. La velocidad de este "vaciado" esta determinada por la señal de reloj de transmisión, el cual es función de la velocidad de recepción de datos de los puertos para la PC. Lo anterior constituye lo que se llama modo de transmisión.

Durante el segundo proceso, dado que se está en el modo de transmisión, la señal analógica se pierde, sin embargo en la visualización en la pantalla de la computadora la retención de la vista hace que esta parezca continua. Aun y cuando se estuviera desplegando la información continuamente en tiempo real, el buffer de la pantalla tiene un período discreto de actualización, por lo que de todos modos se estaría perdiendo algo de información.

La entrada digital de control INV, le indica al convertidor analógico digital que invierta los datos digitales correspondientes a la señal analógica. Las otras señales de

entrada digital son: una señal de reloj (CONV=-CE) que puede ser ya sea la señal de reloj de muestreo o la señal de reloj de transmisión, las direcciones de memoria para la memoria RAM (A0-A10), el control para los buffer de tres estados (-G), el control de lectura o escritura de la memoria RAM (R/-W). Las entradas analógicas a este bloque son la señal VIN y las referencias de voltaje para el convertidor VRT y VRB.

Ambos bloques de conversión analógico digital y almacenamiento temporal, el del canal A y el del canal B, reciben simultáneamente la misma señal de reloj y las mismas direcciones de memoria, para que tomen muestras y transmitan los datos digitales simultáneamente. Los datos de salida (D0 a D7 válidos en el modo de transmisión) del bloque para el canal A van a un puerto paralelo de entrada y los de salida para el canal B van al otro puerto paralelo de entrada, es decir no están multiplexados.

Para comprender el funcionamiento de este bloque es necesario tener presente el bloque "Control", así como revisar el apéndice de especificaciones técnicas en la parte que se refiere al ADC0881, 74LS244 y MCM6264.

Cuando se esta en el modo de muestreo, U2, no esta en estado de alta impedancia y la memoria esta leyendo datos del bus, mientras sus direcciones se incrementan, de esta manera la información del convertidor pasa a la memoria con una determinada velocidad de muestreo hasta un valor predeterminado.

Cuando se esta en el modo de transmisión U2, esta en estado de alta impedancia evitando que los datos del convertidor y de la memoria se colapsen. La memoria deja disponibles los datos en el bus, los cuales son tomados por los puertos de la PC, mientras las direcciones son incrementadas desde cero, hasta un valor predeterminado, lo anterior sincronizado por el reloj de transmisión el cual lo genera

la PC, de esta manera los datos de la memoria (que fueron los datos generados por el convertidor) pasan a la PC a través de los puertos para su procesamiento.

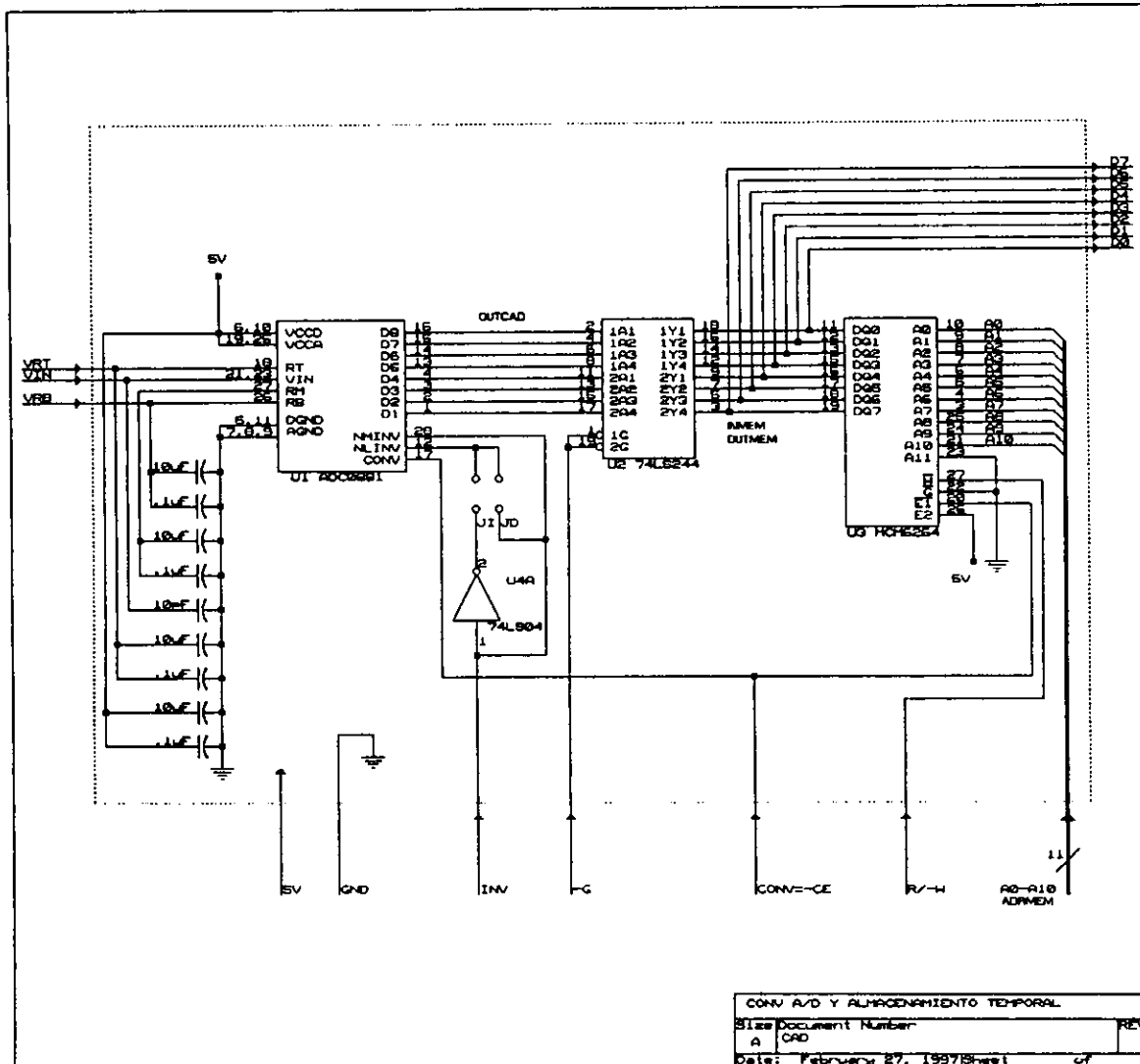


Figura 1.14

## 1.14. PUERTOS DE ENTRADA Y SALIDA (PTS)

Este bloque (figura 1.15) tiene por objetivo tener disponibles 3 puertos de lectura y 3 puertos de escritura de 8 bits cada uno. Además se deja disponible la alimentación de el slot de la PC y diversas señales de control (bus CTRLPTOS) que podrían ser útiles.

La tarjeta se conecta a los buses -IOR, -IOW, CTRLPTOS, DATPCR de la tarjeta arbitraje del bus de datos y lógica de decodificación, a través de un conector DB-25. Así la tarjeta puertos de entrada y de salida queda exterior a la PC.

Los puertos de lectura están constituidos por tres buffers octales de tres estados (U1, U2 y U3), cuando se esta dando una lectura, sólo uno de estos es habilitados a través de alguna de las líneas del bus -IOR, mientras los otros permanecen en estado de alta impedancia para evitar un corto circuito en el bus datos.

Los puertos de escritura están constituidos por tres flip-flops octales tipo D (U4, U5 y U6) cuyas entradas de reloj funcionan como líneas de carga, es decir cuando en el bus de datos, DATPCR, hay información de escritura, sólo a una entrada de reloj de estos flip-flops octales le llega una línea válida (válida en cero) del bus -IOW, con lo cual la información se quedara cargada en este flip-flop octal.

Más específicamente cuando -IOR0 es válido (válido en nivel bajo) se selecciona el puerto de lectura cuyos datos constituyen el bus DR0, las demás líneas de selección (tanto el resto del bus -IOR como todo el bus -IOW) no serán válidas, esto para evitar cortos circuitos en el bus de datos. Cuando -IOR1 es válido se selecciona el puerto de lectura cuyos datos constituyen el bus DR1 mientras que las demás líneas de selección no deben de ser válidas.

De las líneas de selección de puertos de lectura del bloque ALD, se escogieron para los puertos de lectura del bloque PTS las siguientes: -IOR0, -IOR1, -IOR2. Igualmente de las líneas de selección de puertos de escritura del bloque ALD, se escogieron para los puertos de escritura del bloque PTS las siguientes: -IOW3, -IOW4, -IOW5. Las líneas de selección no utilizadas se dejan para expansiones.

No todas las señales de control son utilizadas en el proyecto, se trató de hacer el bloque ALD y el bloque PTS genéricos, es decir, utilizables en otros proyectos, así las líneas de control no utilizadas en este si podrían serlo en otro. Las líneas de control disponibles en esta tarjeta, pero no utilizadas en el proyecto son: OSC, CLK, -/OCHCK, +/OCHRDY e IRQ.

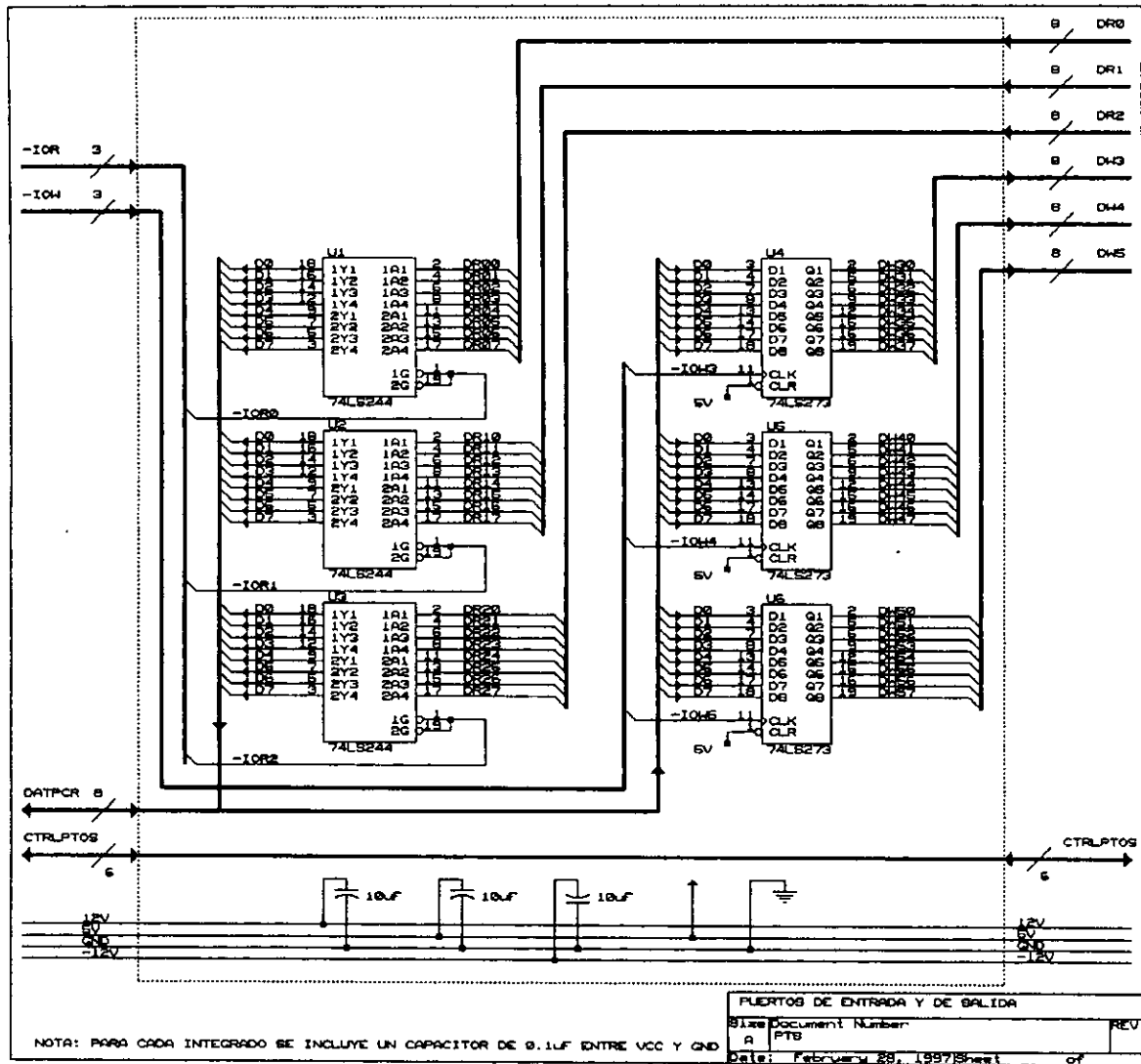


Figura 1.15

## 1.15. OPTOACOPLADORES (OPT)

Este bloque (figura 1.16) tiene la finalidad de aislar por medios ópticos la señal de entrada con respecto a la señal de salida, las tierras también se aíslan, de esta manera en caso de un sobrevoltaje o una sobrecorriente la parte del sistema que esta hacia la PC queda protegida.

Una señal digital de entrada referenciada a GND del proyecto, se convierte a la salida, en una señal digital referenciada a GND de la computadora.

El funcionamiento de este bloque es el siguiente:

Los transistores en los optoacopladores trabajan en corte o saturación, se explicará sólo el funcionamiento de U0, los demás trabajan de manera similar:

Para la señal de entrada IN0 es 1 lógico, circular una corriente que activa el diodo emisor de luz estableciendo un flujo luminoso que excitan la base del fototransistor cuya corriente de base esta determinada por el flujo incidente, estableciendo la corriente de colector y un voltaje de salida de 5 V, por lo tanto, el transistor se encuentra saturado, lo cual es considerado por el buffer U20 como 1 lógico y en la salida O0 se tendrá también 1 lógico.

Cuando la señal IN0 es 0 lógico, el diodo luminiscente esta en corte, por lo tanto, lo esta también el fototransistor, lo cual es considerado por U20 como 0 lógico y en la salida O0 se tendrá también 0 lógico.



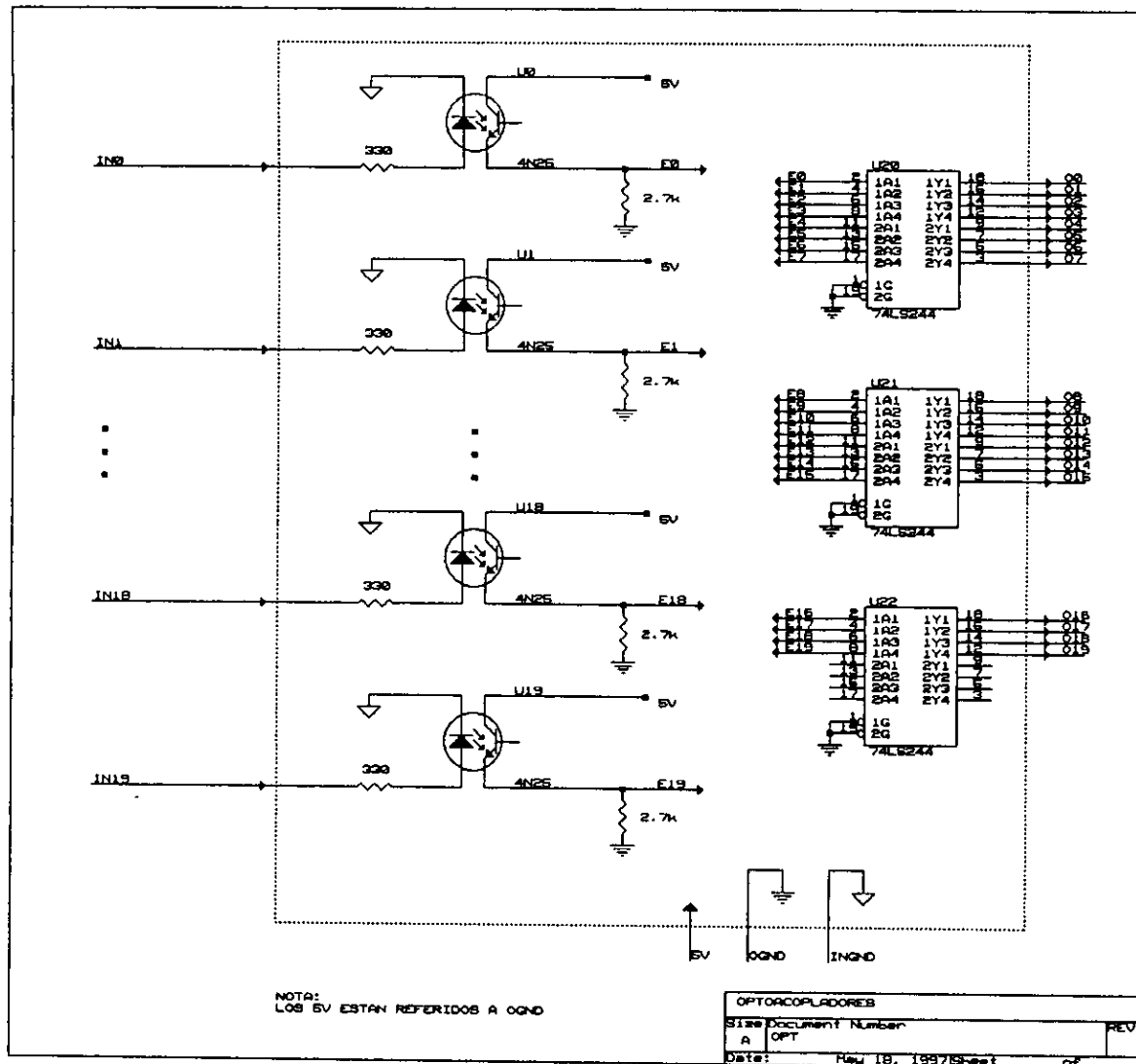


Figura 1.16

## 1.16. ARBITRAJE DEL BUS DE DATOS Y LÓGICA DE DECODIFICACIÓN (BLOQUE ALD)

Este bloque, que se muestra en la figura 1.17, tiene por objetivo generar las líneas de selección de puertos, generar diversas señales de control y dejar disponible el bus de datos de la PC de acuerdo a los requerimientos de los puertos.

La tarjeta se conecta al slot tipo XT (slot de 62 pines) de las PC, además se deja disponible la alimentación de el slot de la PC. Las líneas de direcciones y control provenientes del slot XT son decodificadas para generar las líneas de selección de los puertos y para determinar el sentido de los datos en el bus de datos (si están saliendo de la PC o entrando). Las líneas de selección generadas son:

Para entrada (lectura) -IOR0,-IOR1,-IOR2,-IOR3,-IOR4,-IOR5,-IOR6,-IOR7 (se llamara al conjunto de estas líneas, bus -IOR). Para salida (escritura) -IOW0,-IOW1,-IOW2,-IOW3,-IOW4,-IOW5,-IOW6,-IOW7 (se llamara al conjunto de estas líneas, bus -IOW). Solamente una línea de estos dos buses puede estar activa en un mismo tiempo.

Las líneas -IOR0 y -IOW0 se seleccionan con la misma dirección de memoria, lo que hace que una se active o no, es si desde la PC se puso una instrucción de salida (OUT) o una de entrada (IN) lo cual esta reflejado por las líneas de control -IOR y -IOW. Lo mismo ocurre para -IOR1 y -IOW1 hasta -IOR7 y -IOW7.

Para generar las líneas de selección (los buses -IOR y -IOW) se utilizan las líneas de direcciones de memoria (A0 a A9) y la línea AEN. En el dip-switch S1 se programa el espacio de direcciones de puertos que se trabajara en la tarjeta, la información de las líneas de direcciones de memoria se comparan (a través del comparador U1) con la información proveniente de este dip-switch. Si la comparación resulta verdadera esto significa que las líneas de direcciones de memoria contienen la misma información que la programada en el dip-switch (S1), y que la PC quiere leer o escribir en dicho espacio de puertos. Cuando la comparación es verdadera -IOS se hace válida (válida en bajo porque esta en lógica negada). La línea AEN se compara con 0 (GND), ya que AEN debe tener ese valor en una lectura/escritura de puertos, porque si vale 1 indica que un ciclo de bus DMA esta en progreso y que el controlador de DMA controlara el bus de direcciones, el bus de datos, y las líneas de lectura/escritura de

la memoria e I/O, por lo tanto, si no se incluye AEN en la comparación podrían surgir conflictos en estos buses. En resumen, cuando -IOS es válida (en cero) la PC quiere leer o escribir en los puertos específicos de esta tarjeta.

Lo anterior se baso en el mapa de memoria de la PC (ver apéndice III, la PC enfocada a tarjetas de expansión), con el cual se realizo la siguiente tabla:

A	A	A	A	A	A	A	A	A	A	A	—	—	PUERTO (PTO) QUE ES SELECCIONADO
9	8	7	6	5	4	3	2	1	0		IOR	IOW	
1	1	0	0	0	0	0	0	0	0	0	0	1	PTO DE LECTURA 0 (DR0)
1	1	0	0	0	0	0	0	0	1	0	0	1	PTO DE LECTURA 1 (DR1)
1	1	0	0	0	0	0	0	1	0	0	0	1	PTO DE LECTURA 2 (DR2)
1	1	0	0	0	0	0	0	1	1	0	0	1	FUTURA EXP PARA PTO DE LECTURA
1	1	0	0	0	0	0	1	0	0	0	0	1	FUTURA EXP PARA PTO DE LECTURA
1	1	0	0	0	0	0	1	0	1	0	0	1	FUTURA EXP PARA PTO DE LECTURA
1	1	0	0	0	0	0	1	1	0	0	0	1	FUTURA EXP PARA PTO DE LECTURA
1	1	0	0	0	0	0	1	1	1	0	0	1	FUTURA EXP PARA PTO DE LECTURA
1	1	0	0	0	0	0	0	0	0	1	1	0	PTO DE ESCRITURA 0 (DW3)
1	1	0	0	0	0	0	0	0	1	1	1	0	PTO DE ESCRITURA 1 (DW4)
1	1	0	0	0	0	0	0	1	0	1	1	0	PTO DE ESCRITURA 2 (DW5)
1	1	0	0	0	0	0	0	1	1	1	1	0	FUTURA EXP PARA PTO DE LECTURA
1	1	0	0	0	0	0	1	0	0	1	0	0	FUTURA EXP PARA PTO DE LECTURA
1	1	0	0	0	0	0	1	0	1	1	1	0	FUTURA EXP PARA PTO DE LECTURA
1	1	0	0	0	0	0	1	1	0	1	0	0	FUTURA EXP PARA PTO DE LECTURA
1	1	0	0	0	0	0	1	1	1	1	1	0	FUTURA EXP PARA PTO DE LECTURA

El valor A9 A8 A7 A6 A5 A4 A3 = 1100000 = 300 hex, es establecido en el dip switch S1 por default, con lo cual se hace referencia a la primera dirección de memoria del espacio que la PC asigna a las tarjetas prototipo (ver apéndice III, la PC enfocada a tarjetas de expansión), este valor puede cambiar si por alguna razón se crea algún conflicto con la memoria o si el modelo PC a utilizar maneja otro espacio para las tarjetas prototipo.

En la tabla anterior se hace notar que las direcciones altas (A3 a A9) controlan el espacio de puertos del proyecto, es decir el rango. Para especificar un puerto en este rango se utilizan las direcciones bajas (A0 a A2). El decodificador U5 se debe de habilitar cuando la PC indique que quiere acceder al espacio de direcciones de puertos del proyecto, es decir, cuando  $\text{-IOS}$  valga 0 lógico, y cuando la PC indique que quiere acceder a un puerto de lectura (ya que en U5 se tienen las líneas de selección  $\text{-IOR0}$  a  $\text{-IOR7}$  que son para la selección de los puertos de lectura) haciendo  $\text{-IOR}$  válido (válido en cero). Así cuando  $\text{-IOS} = 0$  y  $\text{-IOR} = 0$  se habilita U5. Ahora estando U5 habilitado, las direcciones A2 A1 A0 son decodificadas por este integrado para que la deseada línea de selección de puerto de lectura sea válida (válida en cero).

De la misma manera el decodificador U6 se debe de habilitar cuando la PC indique que quiere acceder al espacio de direcciones de puertos del proyecto, es decir cuando  $\text{-IOS}$  valga 0, y cuando la PC indique que quiere acceder a un puerto de escritura (ya que en U6 se tienen las líneas de selección  $\text{-IOR0}$  a  $\text{-IOR7}$  que son para la selección de los puertos de escritura) haciendo  $\text{-IOW}$  válido (válido en cero). Así cuando  $\text{-IOS} = 0$  y  $\text{-IOW} = 0$  se habilita U6. Ahora estando U6 habilitado, las direcciones A2 A1 A0 son decodificadas por este integrado para que la deseada línea de selección de puerto de escritura sea válida (válida en cero).

Los puertos pueden tener acceso al bus de datos de la PC para lectura o escritura cuando se cumplan las siguientes condiciones:

Que  $\text{-IOR}$  o  $\text{-IOW}$  sean válidos, es decir cuando la PC quiera hacer una lectura o escritura en el bus de datos.

Cuando  $\text{-IOS}$  sea válido, es decir cuando se seleccione el espacio de direcciones de puertos, especificado por el dip-switch identificado como S1.

Cuando ambas condiciones se cumplan, el integrado U3 (ver apéndice IV, en la parte del 74LS245) estará habilitado para controlar el sentido de la información, es decir que se va a realizar una lectura o escritura del bus de datos. Este sentido esta dado por -IOR, así cuando -IOR sea válido el sentido de la información es desde el mundo exterior hacia la PC, y cuando -IOR no sea válido el sentido de la información es de la PC al exterior. Lo anterior se puede traducir en las

siguientes ecuaciones booleanas:

$$GU3 = ( IOR \text{ OR } IOW ) \text{ AND } IOS \quad \text{en lógica no negada}$$

$$GU3 = - ( -IOR \text{ AND } -IOW ) \text{ AND } - ( -IOS ) \quad \text{teorema De Morgan}$$

$$-GU3 = - [ - ( -IOR \text{ AND } -IOW ) \text{ AND } - ( -IOS ) ]$$

$$-GU3 = (-IOR \text{ NAND } -IOW) \text{ NAND } (-IOS \text{ NAND } -IOS)$$

Donde -GU3 es la línea de habilitación del integrado U3

La ecuación anterior esta representada en el bloque ALD, a través de las compuertas NAND: U4A, U4B y U4C.

-IOR se conecta al pin DIR (dirección) de U3, controlando el sentido de la información, es decir cuando -IOR y -GU3 sean ambos válidos (en nivel bajo), U3 dejara pasar la información solo en el sentido del exterior hacia la PC. Cuando -IOR no sea válido y -GU3 sea válido, U3 dejara pasar la información solo en el sentido de la PC hacia el exterior.

Las líneas de dirección A2, A1, A0 y las señales de control de salida -IOR, -IOW, RESETDRV, OSC Y CLK son reforzadas a través de un buffer identificado como U2

También se utilizan las señales de control de entrada a la PC: -/OCHCK, /OCHRDY y las líneas de interrupción IRQ2, IRQ3, IRQ4, IRQ5, IRQ6, IRQ7, solo una de estas líneas de interrupción es utilizada en este bloque, esta línea es seleccionada manualmente a través del dip-switch S2, es decir que la tarjeta da la disponibilidad de trabajar con una sola línea de interrupción, pero teniendo la alternativa que esta puede ser cualquiera de IRQ2 a IRQ7.

Los capacitores mostrados en la parte inferior del diagrama del bloque ALD, tienen la finalidad de filtrar los pequeños componentes de AC y picos de voltaje.

No todas las señales de control son utilizadas en el proyecto, se trato de hacer a el bloque ALD y al bloque PTS genéricos, es decir utilizables en otros proyectos, así las líneas de control no utilizadas en este si podrían serlo en otro.

Las líneas de control disponibles en esta tarjeta, pero no utilizadas en el proyecto son: OSC, CLK, -/OCHCK, +/OCHRDY e IRQ.

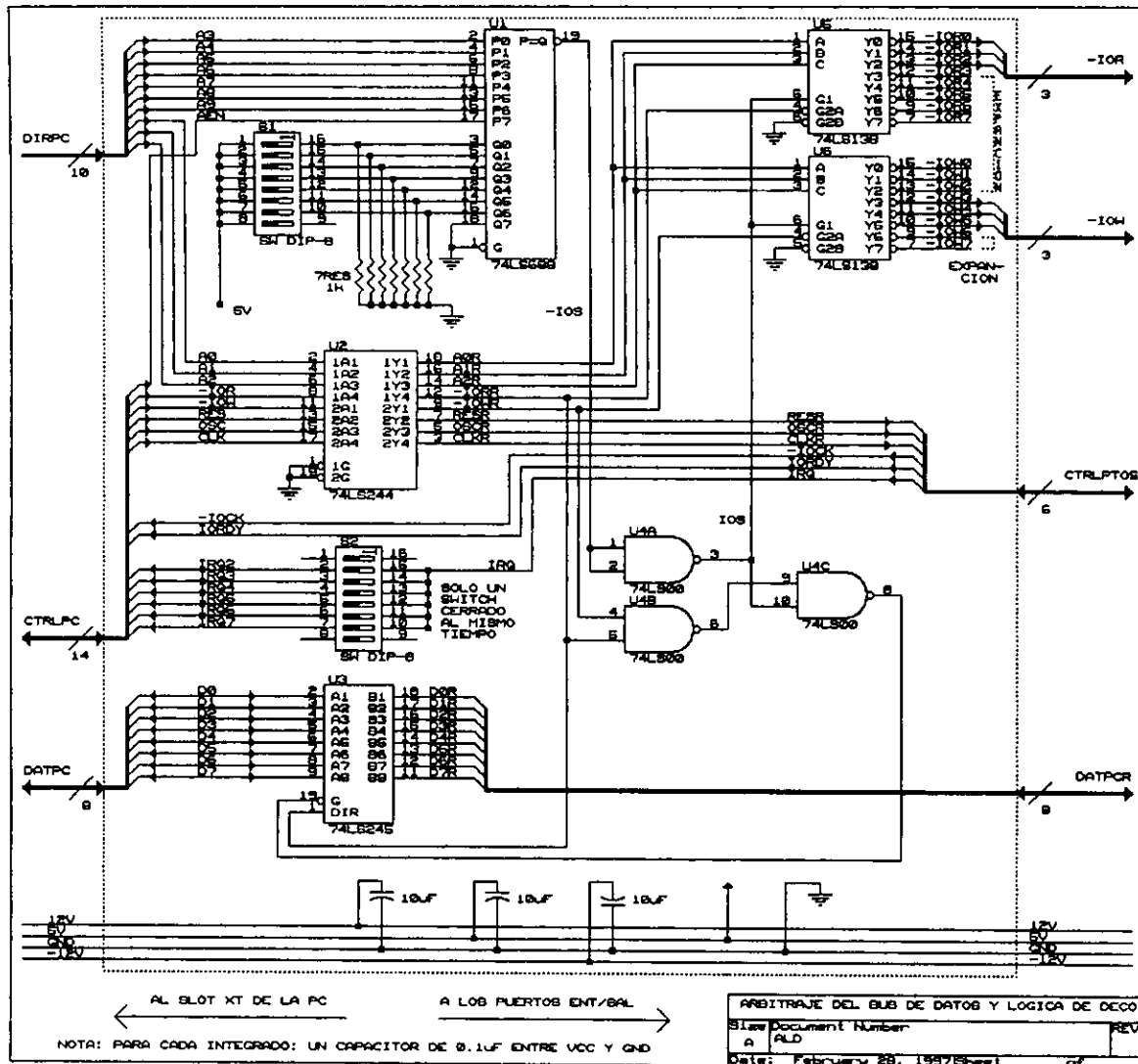


Figura 1.17

## 1.17. SLOT IBM PC XT, 62 TERMINALES (SXTPC)

Aquí es donde se conecta el bloque arbitraje del bus de datos y lógica de decodificación. De este slot se toman las líneas de direcciones de memoria A0, A1, A2, A3, A4, A5, A6, A7, A8, A9; las líneas de control AEN, -IOR, -IOW, RESETDRV, OSC, CLK, -/OCHCK, +/OCHRDY, IRQ2, IRQ3, IRQ4, IRQ5, IRQ6, IRQ7; las líneas

bidireccionales del bus de datos D0, D1, D2, D3, D4, D5, D6, D7; y las líneas de alimentación +12V, +5V, GND, -12V como se muestra en la figura 1.18.

Para una explicación detallada de estas señales y las demás del slot IBM PC XT de 62 terminales ver el apéndice III, "La PC Enfocada a Tarjetas de Expansión".

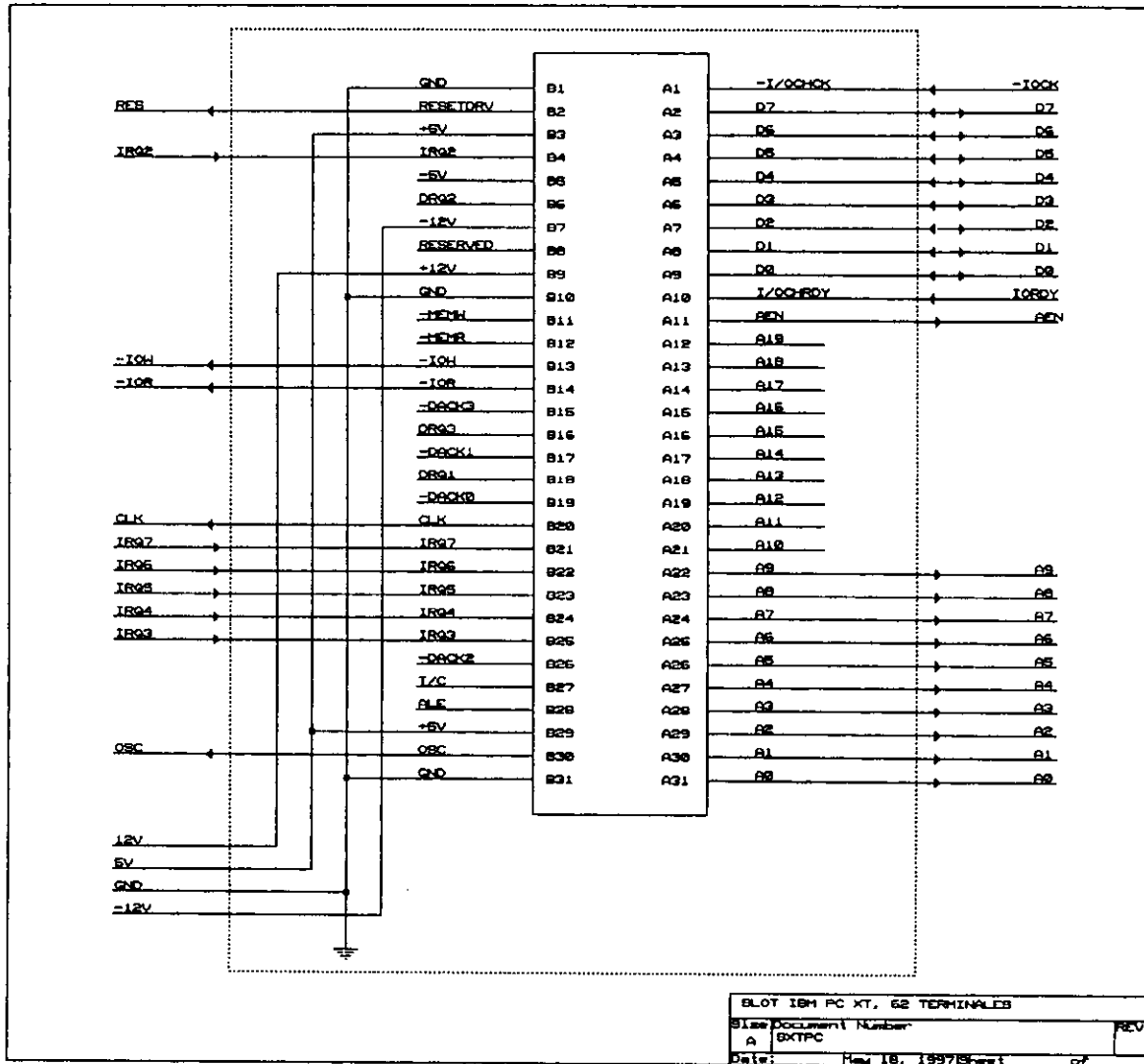


Figura 1.18



---

# **CAPÍTULO 2**

## ***ANÁLISIS DINÁMICO***

---

El análisis dinámico corresponde al análisis en tiempo de cada señal digital, con el fin de que estén en sincronía para el adecuado funcionamiento del sistema. Involucra esencialmente los bloques "CAD y Almacenamiento Temporal" y "Control". Dos bloques CAD trabajan simultáneamente uno para el canal A y el otro para el canal B. En la descripción de los modos de operación, que a continuación se detallan, se requiere hacer referencia directa a los bloques CAD y CTRL.

La especificación sobre el retardo de las compuertas que se utilizaron para los diagramas de tiempos es mostrado en el apéndice IV.

### **2.1. MODO DE MUESTREO**

El modo de muestreo esta presente hasta que se alcance un número predeterminado de muestras, en este modo los datos del convertidor analógico digital son escritos en la RAM (U3 de CAD) a una velocidad dada por el reloj de muestreo.

La explicación de las señales en el diagrama de tiempos (que se muestra en la figura 2.1), se da a continuación.

Se considera el caso más crítico, en donde el reloj de muestreo, CKM, tiene una frecuencia de 10 MHz y un período de 100 ns.

CKT (dada por la PC). Permanece en nivel bajo porque no hay reloj de transmisión en el modo de muestreo

-CLADR (dada por la PC). Permanece en alto, con lo cual se incrementan normalmente los contadores U1, U2, U3 de CTRL, que generan las direcciones de memoria.

T/-M (dada por la PC). Permanece en nivel bajo indicando que se esta en el modo de muestreo.

RT/-RM. Misma señal que T/-M pero con un retraso aproximadamente 70 ns, este retraso es el generado por las compuertas U6B, U6C, U6D, U7B, U7C y U7D de CTRL. El objetivo de este retraso es explicado posteriormente.

R/-W. Permanece en nivel bajo, ya que en el modo de muestreo se están escribiendo datos en la memoria RAM.

-G. Permanece también en nivel bajo, habilitándose el buffer de 3 estados U2 de CAD, que deja pasar la información del convertidor analógico digital para que sea escrita en la RAM.

MUXSAL. Con el multiplexor U4 de CTRL se escoge CKM, por lo que a la salida de este multiplexor, MUXSAL, se tendrá CKM pero con un retraso máximo de 14 ns.

CONV=-CE. Como RT/-RM permanece en cero, la salida de la compuerta U5C de CTRL (NOR) será MUXSAL invertida con un retraso propio de esta misma compuerta, esta salida la denominamos CONV=-CE, que es la entrada de reloj para convertidor

analógico digital (pin CONV de U1 de CAD) y también es el pin de habilitación de la memoria RAM (identificada como U3 de CAD).

OUTCAD. Son los datos de salida de convertidor analógico digital. De acuerdo a los diagramas de tiempo para el convertidor ADC0881, con un período de reloj de 100 ns en el pin CONV, se tiene que los datos de la conversión (OUTCAD) son validos de acuerdo a lo mostrado en el diagrama de tiempos en el modo de muestreo.

INMEM. Son los datos que se están escribiendo en la RAM, estos datos son los mismos que OUTCAD pero con un retraso provocado por el buffer de 3 estados (U2 de CAD). La señal digital CONV=-CE es la entrada de habilitación de la memoria U3 de CAD (pin -CE), esta señal hace que se habilite la memoria en su período bajo y se escriban (porque R/-W permanece en cero lógico) en ella los datos INMEM durante IN. Lo anterior se obtuvo de los diagramas de tiempo para la memoria MCM6264 con un período de 100 ns en el pin -CE (50 ns en alto y 50 ns en bajo). Esto es mostrado en el diagrama de tiempos en el modo de muestreo.

OUTMEM. Permanece en estado de alta impedancia ya que aquí se están escribiendo datos en la memoria.

CKCONT. Es el reloj que es la base de tiempo de los contadores que generan las direcciones de memoria de la RAM. En el modo de muestreo, este es el mismo que CKM, pero con un retraso provocado por U4 y U5B de CTRL.

ADRMEM. Son las direcciones de memoria (A0-A10) generadas por los contadores U1, U2, U3 de CTRL. Estas direcciones tienen un retraso máximo de 27 ns con respecto al flanco positivo de CKCONT, retraso provocado por los mismos contadores.

CTATERM. Indica que se ha llegado a el número predeterminado de muestras y que se debe iniciar el modo de transmisión. A lo largo del modo de muestreo permanece en nivel bajo.

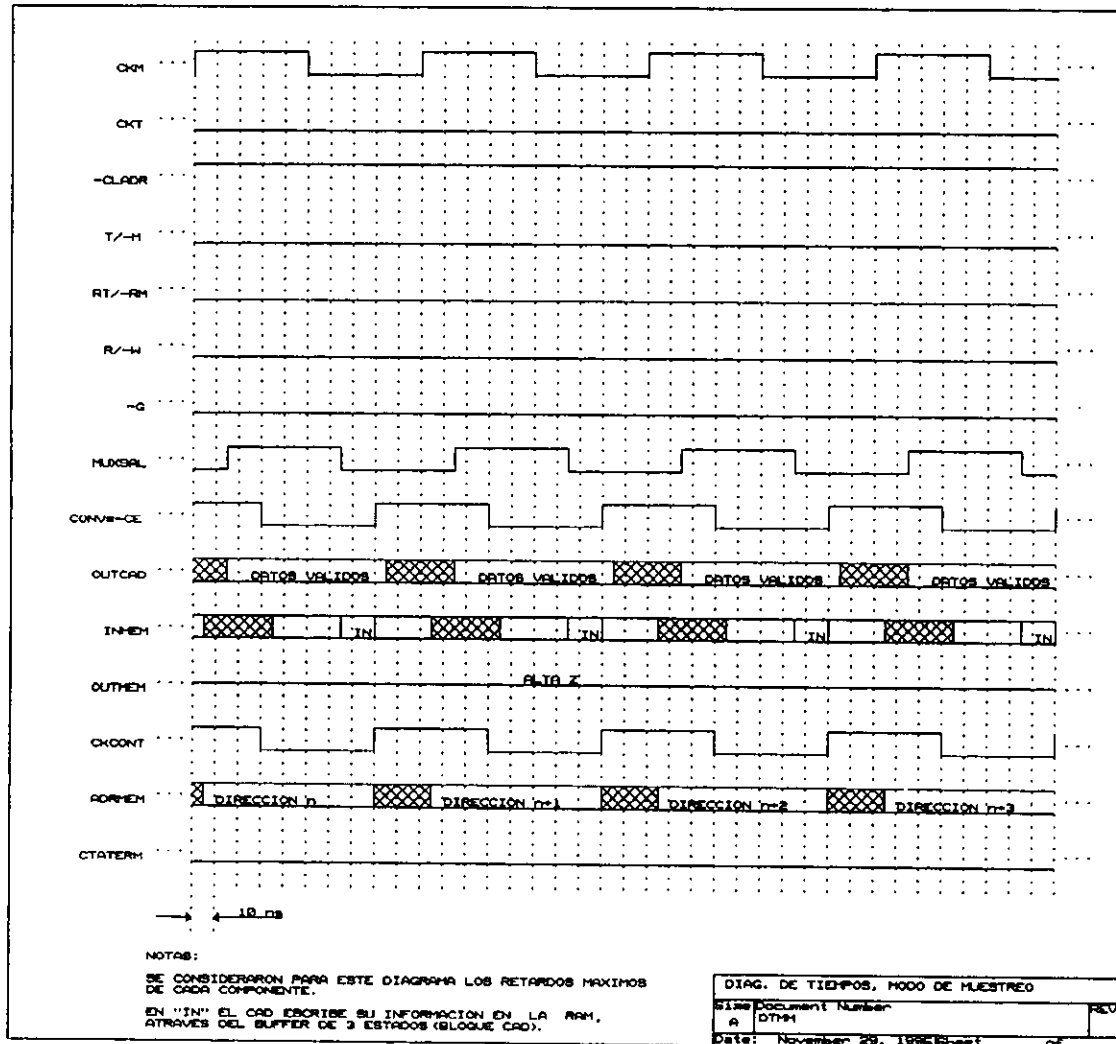


Figura 2.1

## 2.2. TRANSICIÓN DEL MODO DE MUESTREO AL DE TRANSMISIÓN

La transición del modo de muestreo al modo de transmisión (ver figuras 2.2 y 2.3) empieza cuando se ha alcanzado el número máximo de muestras y CTATERM se hace válido con lo que se bloquea el reloj de los contadores generadores de direcciones de memoria (CKCONT) para que estos no se incrementen más. Sin embargo como CONV=-CE sigue siendo el reloj de muestreo (CKM) y el buffer de tres estados no está en alta impedancia, el contenido de la primera dirección (dirección 0) de memoria es sobrescrito varias veces con la información actual del convertidor. Así el contenido de la dirección 0 será sólo basura.

A la señal CTATERM la esta sensando continuamente un puerto de entrada de la PC, cuando ésta detecta que está activa realiza los siguientes eventos en orden de ocurrencia:

Evento 1M-T: La PC establece T/-M en alto: Con lo cual primeramente, a través de U7A de CTRL, -G pasa a 1 lógico y el buffer de 3 estados, U2 de CAD, entra en estado de alta impedancia (INMEM en alta impedancia). Después de un retardo provocado por U6B, U6C, U6D, U7B, U7C, U7D de CTRL, y a través de U6A de CTRL, R/-W pasa a 1 lógico, con lo que la memoria U3 de CAD da información que puede ser leída por el sistema (OUTMEM deja el estado de alta impedancia). El que primero entre en estado de alta impedancia el buffer U2 de CAD y después la memoria de información para ser leída, es para evitar un corto circuito entre la memoria y el buffer de tres estados. A través de RT/-RM en la línea de selección de el multiplexor U4 de CTRL, se escoge también la señal CKT, la cual es el reloj que será dado por la PC en el modo de transmisión.

Evento 2M-T: La PC establece -CLADR en bajo: Con lo que la cuenta en los contadores (ADRMEM) U1, U2, U3 de CTRL se hace cero. Al hacerse cero la cuenta

---

(o dirección de memoria), CTATERM también se va a cero, con lo que se desbloquea el reloj de los contadores (CKCONT).

Evento 3M-T: La PC establece -CLADR en alto, con lo que se dejan listos a los contadores para ser incrementados en el modo de transmisión.

Evento 4M-T: La PC empieza mandar un pulso, que es el reloj de transmisión (CKT) con lo cual los contadores se empiezan a incrementar, como la memoria esta en lectura, empieza a pasar su información (que son los datos de las muestras) a la PC a través de un puerto de entrada, así, el modo de transmisión se inicia.

Por otra parte, el reloj de muestreo CKM sigue activo, pero como este no es seleccionado a través del multiplexor, después del evento 1, no tiene ninguna importancia.

Como RT/-RM se va a uno lógico después del evento 1, CONV=-CE queda bloqueado en cero a través de la compuerta U5C de CTRL. Esto tiene el objeto de que en el modo de transmisión siempre este habilitada la memoria. CONV=-CE también es el reloj del convertidor, que como ya se dijo esta bloqueado, esto no tiene importancia en el modo de transmisión, ya que su información no esta pasando a la memoria porque el buffer U2 de CAD esta en estado de alta impedancia.

Es importante observar que INMEM y OUTMEM nunca deben de dejar el estado de alta impedancia al mismo tiempo ya que comparten el mismo bus físico, de lo contrario se producirá un corto circuito. Así se deja un tiempo de protección en donde tanto INMEM como OUTMEM están en alta impedancia.

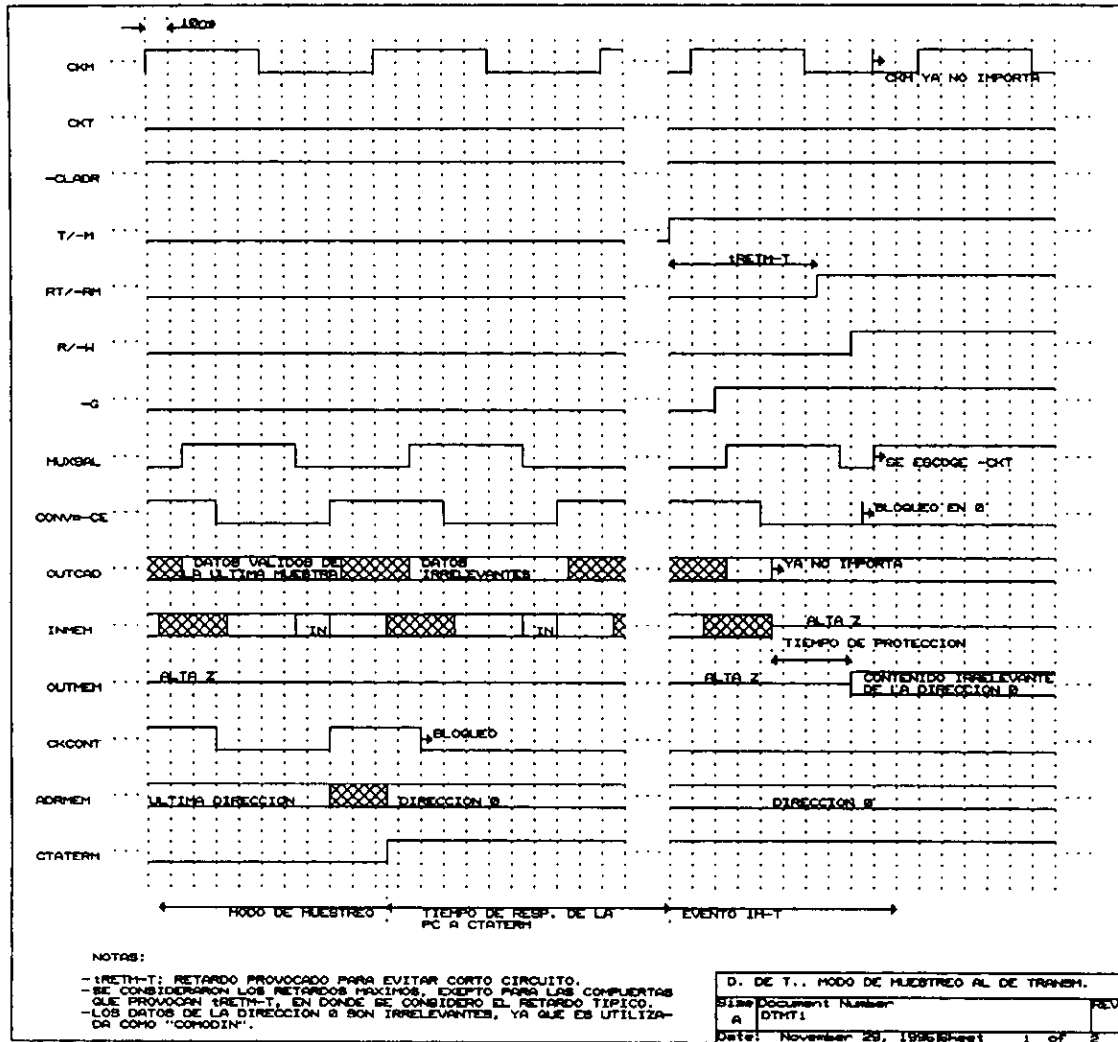


Figura 2.2

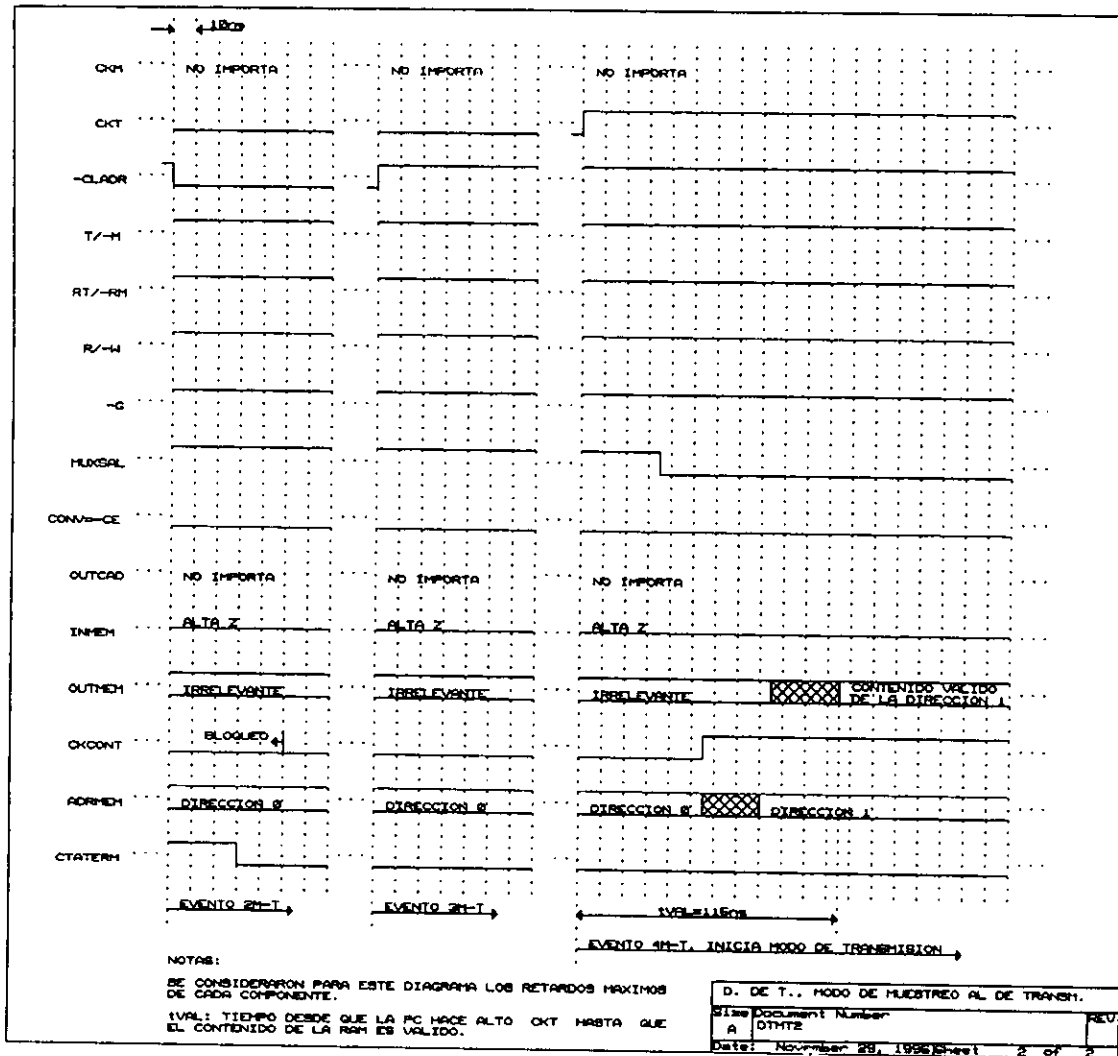


Figura 2.3

## 2.3. MODO DE TRANSMISIÓN

Es importante destacar que en este diagrama (figura 2.4) los tiempos no están en escala, ni son proporcionales a las distancias en el papel.

CKM no importa ya que no esta siendo seleccionado por el multiplexor U4 de CTRL.



---

CKT es el reloj generado por la PC a través de un programa, no es necesario que su ciclo de trabajo sea del 50%.

-CLADR (dada por la PC). Permanece en alto, con lo cual se incrementan normalmente los contadores U1, U2, U3 de CTRL, generadores de direcciones de memoria.

T/-M (dada por la PC). Permanece en nivel alto indicando que se esta en el modo de transmisión.

RT/-RM. Misma señal que T/-M pero con un retraso aproximadamente 70 ns, este retraso es el generado por las compuertas U6B, U6C, U6D, U7B, U7C y U7D de CTRL.

R/-W. Permanece en nivel alto, ya que en el modo de transmisión se están leyendo datos de la memoria RAM

-G. Permanece también en nivel alto, deshabilitando el buffer de 3 estados U2 de CAD, no dejando pasar la información del convertidor analógico digital al bus que esta ocupando la RAM, evitando así un corto circuito.

MUXSAL. Con el multiplexor U4 de CTRL se escoge CKT, por lo que a la salida de este multiplexor, MUXSAL, se tendrá CKT pero con cierto retraso.

CONV=-CE. Como RT/-RM permanece en 1, CONV=-CE queda bloqueado en cero a través de la compuerta U5C de CTRL.

OUTCAD. Son los datos de salida de convertidor analógico digital. Como el buffer que pasa su información esta en estado de alta impedancia, los datos del convertidor no importan.

INMEM. Permanece en estado de alta impedancia (a través del buffer U2 de CAD que esta en alta impedancia).

OUTMEM. Son los datos que se están leyendo de la RAM, estos datos son la información de las muestras que se tomaron en el modo de muestreo. Estos datos son validos después de tVAL (tiempo desde que la PC hace alto CKT hasta que el contenido de la RAM es valido, tVAL=115ns aprox.), después de este tiempo la PC puede leer estos datos (esto lo hace en "OUT").

CKCONT. Es el reloj que es la base de tiempo de los contadores que generan las direcciones de memoria de la RAM. En el modo de transmisión, este es el mismo que CKT, pero con cierto retraso.

ADRMEM. Son las direcciones de memoria generadas por los contadores U1, U2, U3 de CTRL. Estas direcciones tienen un retraso máximo de 27 ns con respecto al flanco positivo de CKCONT, retraso provocado por los contadores.

CTATERM. Indica que se ha alcanzado el numero predeterminado de muestras que fueron almacenadas en la RAM.

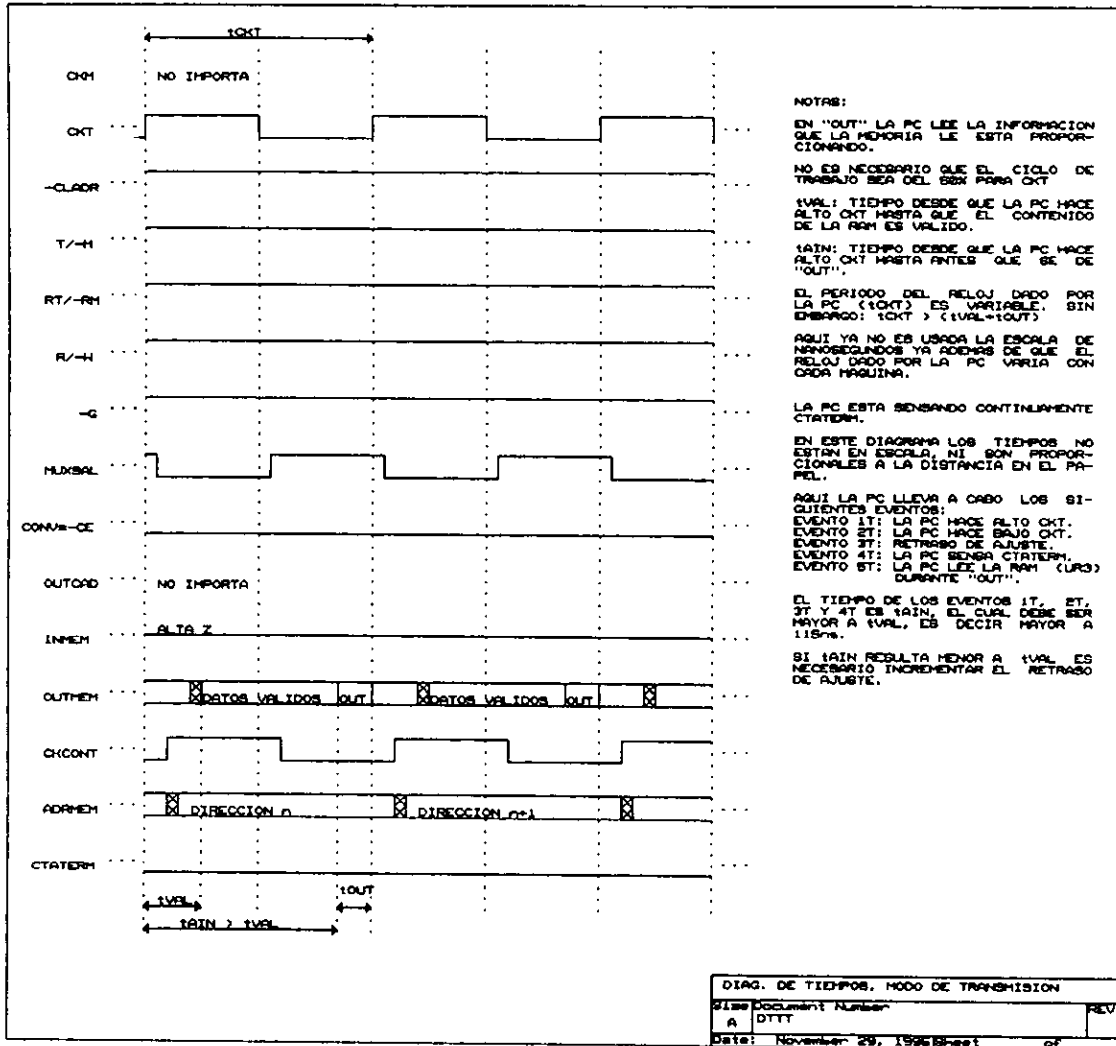


Figura 2.4

## 2.4. TRANSICIÓN DEL MODO DE TRANSMISIÓN AL DE MUESTREO

La transición del modo de transmisión al de muestreo (ver figuras 2.5 y 2.6) empieza cuando todas las muestras que fueron almacenadas en la RAM ya han sido pasadas a la PC y CTATERM se hace válido con lo que se bloquea el reloj de los contadores generadores de direcciones de memoria (CKCONT) para que estos no se

---

incrementen más. Así en ADRMEM se tendrá una dirección para la RAM de 0, como aquí la PC ya no leerá a la RAM su contenido es irrelevante.

A la señal CTATERM la esta sensando continuamente un puerto de entrada de la PC, cuando ésta detecta que está activa realiza los siguientes eventos en orden de ocurrencia:

Evento 1T-M: La PC establece T/-M en bajo: Con lo cual primeramente, a través de U6A de CTRL, R/-W pasa a 0 lógico y la RAM (U3 de CAD) entra en estado de escritura, es decir OUTMEM entra en estado de alta impedancia. Después de un retardo provocado por U6B, U6C, U6D, U7B, U7C, U7D de CTRL, y a través de U7A de CTRL, -G pasa a 0 lógico, con lo que el buffer de tres estados U2 de CAD deja el estado de alta impedancia dando los datos INMEM. La razón de que primero entre en estado de alta impedancia OUTMEM y después el buffer U2 de CAD deje el estado de alta impedancia, es para evitar un corto circuito entre la memoria y el buffer de tres estos. A través de RT/-RM en la línea de selección de el multiplexor U4 de CTRL se escoge también la señal CKM, el cual es el reloj de muestreo.

Evento 2T-M: La PC establece -CLADR en bajo: Con lo que la cuenta en los contadores U1, U2, U3 de CTRL se hace cero, por lo que en ADRMEM se tendrá la dirección 0, cuyo contenido es sólo basura. Al hacerse cero la cuenta (o dirección de memoria), CTATERM también se va a cero, con lo que se desbloquea el reloj de los contadores (CKCON).

Evento 3T-M: La PC establece -CLADR en alto, con lo que los contadores empiezan a ser incrementados y el modo de muestreo de inicia.

Es importante observar que INMEM y OUTMEM nunca deben de dejar el estado de alta impedancia al mismo tiempo ya que comparten el mismo bus físico, de lo

contrario se producirá un corto circuito. Así se deja un tiempo de protección en donde tanto INMEM como OUTMEM están en alta impedancia.

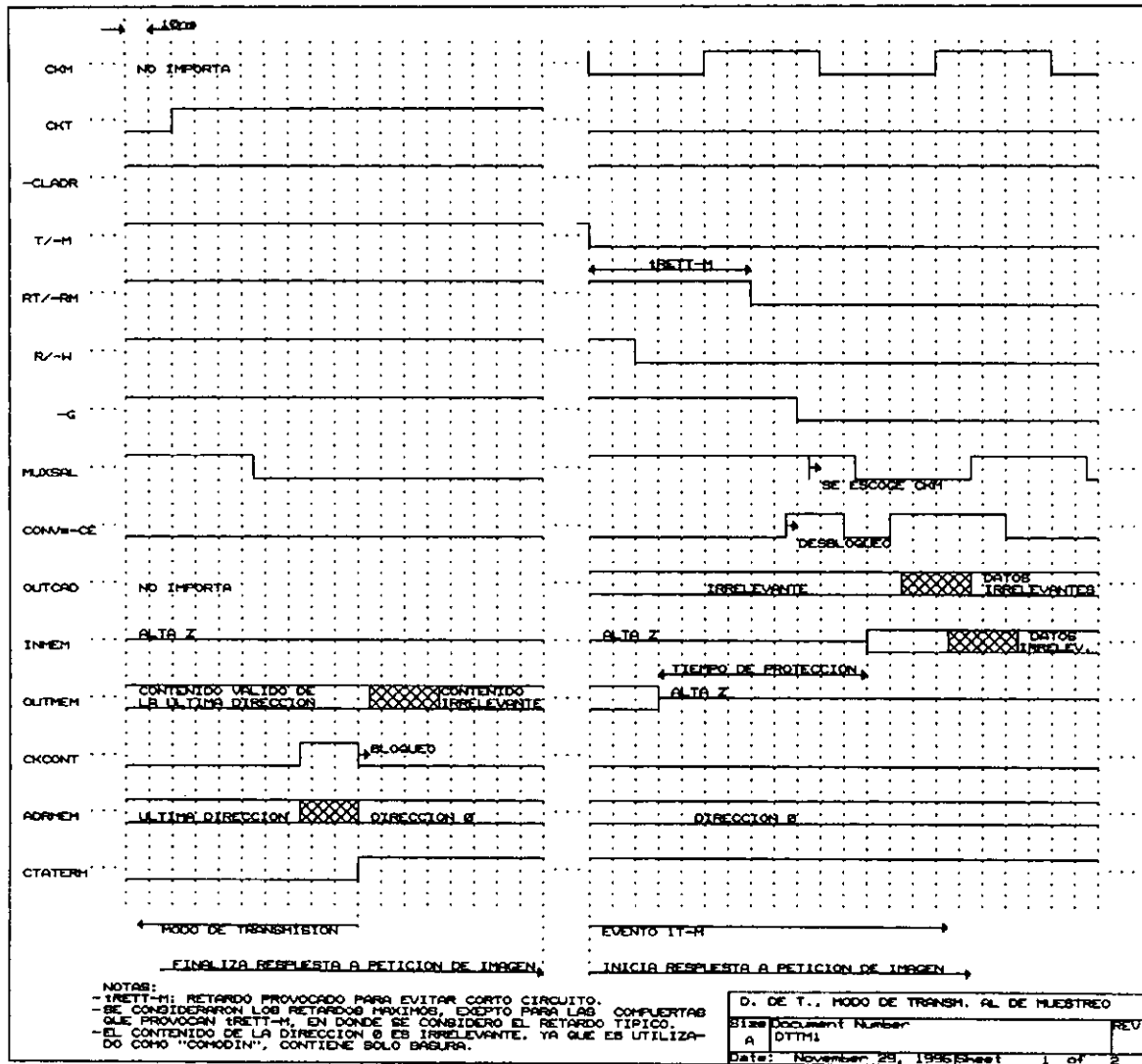


Figura 2.5

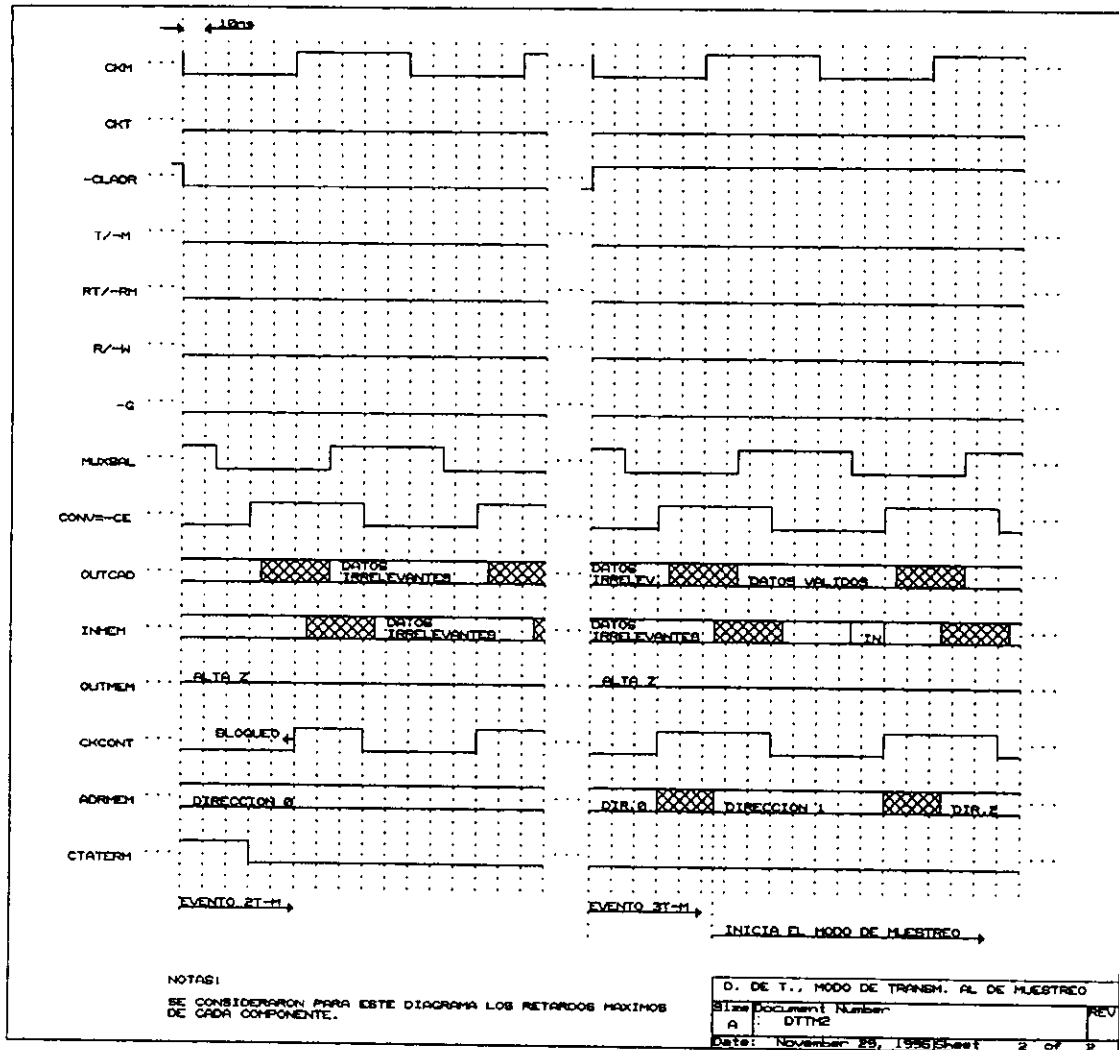


Figura 2.6

---

# CAPÍTULO 3

## *DESCRIPCIÓN DEL SOFTWARE*

---

### 3.1. DESCRIPCIÓN GENERAL

Se desarrollo el programa (ver apéndice I) en el lenguaje de alto nivel C++, que se encarga de enviar diversas señales de control al hardware. También se encarga de recibir, analizar y presentar los datos que el hardware le envía, lo anterior de acuerdo a las necesidades de medición del usuario. En el programa se cargan los vectores *imga* e *imgb*, asociados al canal A y B respectivamente, estos contienen información de las muestras de los convertidores A/D almacenadas en memorias RAM, dependiendo de la configuración utilizada por el usuario, el programa analiza estos vectores como sigue:

- **CANAL A: voltaje, CANAL B: no usar.** Presenta *imga* (señal de voltaje) con las escalas vertical y horizontal establecidas por el usuario. Así mismo determina el valor máximo, el valor mínimo, la frecuencia y el período de esta señal, analizando su código digital.
- **CANAL A: no usar, CANAL B: corriente.** Presenta *imgb* (señal de corriente) con las escalas vertical y horizontal establecidas por el usuario. Así mismo determina el valor máximo, el valor mínimo, la frecuencia y el período.
- **CANAL A: Voltaje, CANAL B: Voltaje.** Presenta *imga* e *imgb* (ambas señales de voltaje) con las escalas vertical y horizontal establecidas por el usuario. Así mismo se determina el valor máximo, el valor mínimo, la frecuencia y el período de cada

señal de voltaje, analizando el código digital asociado a cada señal.

- **CANAL A: impedancia, CANAL B: no usar.** Analiza  $im_{ga}$  (señal de voltaje asociada a la impedancia desconocida) e  $im_{gb}$  (señal de voltaje asociada a la impedancia resistiva conocida) y obtiene su relación de magnitudes y su diferencia angular, con ésta información calcula la impedancia desconocida a través de ecuaciones detalladas en el apéndice II.
- **CANAL A: voltaje, CANAL B: corriente.** Analiza  $im_{ga}$  (señal de voltaje) e  $im_{gb}$  (señal de corriente), obtiene sus magnitudes y su diferencia angular, con esta información calcula la potencia aparente en magnitud y ángulo, correspondiente al voltaje (canal A) y a la corriente (canal B) a través de ecuaciones que se detallan en el apéndice II. Con la potencia aparente el programa determina la potencia real, la potencia reactiva y el factor de potencia, a través de ecuaciones detalladas en el mismo apéndice. Además del calculo de potencia el programa presenta la señal de voltaje del canal A y la señal de corriente del canal B con las escalas vertical y horizontal establecidas por el usuario.

El programa se divide en los siguientes módulos:

- Comandos para el control del hardware
- Funciones para el control del hardware
- Gráficos
- Transformación de unidades
- Procesamiento de los vectores imagen de las RAM
- Control de menú
- Ventanas
- Principal



## 3.2. COMANDOS PARA EL CONTROL DEL HARDWARE

Es aquí donde el software envía y recibe información del hardware. Los bloques que sirven de interface entre la PC y el hardware del proyecto son el bloque ALD y el bloque PTS, en este ultimo se tienen seis puertos, tres de entrada y tres de salida. Los puertos de entrada o de lectura en el bloque PTS están denotados por los buses DR0, DR1 y DR2. Los puertos de salida o de escritura están denotados por los buses DW3, DW4 y DW5. Cada bit de estos buses es denotado como sigue:

BUS	MSB							LSB
DR0	DR07	DR06	DR05	DR04	DR03	DR02	DR01	DR00
DR1	DR17	DR16	DR15	DR14	DR13	DR12	DR11	DR10
DR2	DR27	DR26	DR25	DR24	DR23	DR22	DR21	DR20
DW3	DW37	DW36	DW35	DW34	DW33	DW32	DW31	DW30
DW4	DW47	DW46	DW45	DW44	DW43	DW42	DW41	DW40
DW5	DW57	DW56	DW55	DW54	DW53	DW52	DW51	DW50

Cada bit de cada puerto de salida es una señal de control que el software manda al hardware a través de una dirección en el espacio de puertos correspondiente al uso de tarjetas prototipo. A su vez cada bit de cada puerto de entrada es una señal que el software sensa del hardware, también a través de una dirección en el espacio de puertos. Lo anterior de acuerdo a la siguiente tabla:

BUS	DIR ESP PTOS (HEX)
DR0	300
DR1	301
DR2	302
DW3	303
DW4	304
DW5	305

En el capítulo correspondiente a la descripción del hardware se detallo las señales que cada bloque requiere para su control, así como las señales que la PC sensa del hardware. A continuación se detalla la correspondencia entre estas señales y los bytes del bloque PTS.

DR00=DA0	Bit 0 de la entrada de datos a la PC del canal A
DR01=DA1	Bit 1 de la entrada de datos a la PC del canal A
DR02=DA2	Bit 2 de la entrada de datos a la PC del canal A
DR03=DA3	Bit 3 de la entrada de datos a la PC del canal A
DR04=DA4	Bit 4 de la entrada de datos a la PC del canal A
DR05=DA5	Bit 5 de la entrada de datos a la PC del canal A
DR06=DA6	Bit 6 de la entrada de datos a la PC del canal A
DR07=DA7	Bit 7 de la entrada de datos a la PC del canal A

DR10=DB0	Bit 0 de la entrada de datos a la PC del canal B
DR11=DB1	Bit 1 de la entrada de datos a la PC del canal B
DR12=DB2	Bit 2 de la entrada de datos a la PC del canal B
DR13=DB3	Bit 3 de la entrada de datos a la PC del canal B
DR14=DB4	Bit 4 de la entrada de datos a la PC del canal B
DR15=DB5	Bit 5 de la entrada de datos a la PC del canal B
DR16=DB6	Bit 6 de la entrada de datos a la PC del canal B
DR17=DB7	Bit 7 de la entrada de datos a la PC del canal B

DR20=CTATERM	Si es 1, el contador generador de direcciones de mem. esta en cuenta máx.
DR21=Expansión	
DR22=Expansión	
DR23=Expansión	
DR24=Expansión	
DR25=Expansión	
DR26=Expansión	
DR27=Expansión	

DW30=CFS	Activa la fuente senoidal
DW31=Expansión	
DW32=VAH/-VAL	0 = no activa divisor., 1 = activa divisor., para el canal A
DW33=VBH/-VBL	0 = no activa divisor., 1 = activa divisor., para el canal B
DW34=CR0	Bit 0 del código para seleccionar el valor de la resistencia programable
DW35=CR1	Bit 1 del código para seleccionar el valor de la resistencia programable
DW36=CR2	Bit 2 del código para seleccionar el valor de la resistencia programable
DW37=AX2	Control para el AGPA, multiplica VA por 2

DW40=AX5	Control para el AGPA, multiplica VA por 5
DW41=AX10	Control para el AGPA, multiplica VA por 10
DW42=BX2	Control para el AGPB, multiplica VB por 2
DW43=BX5	Control para el AGPB, multiplica VB por 5
DW44=BX10	Control para el AGPB, multiplica VB por 10
DW45=ESCTM0	Bit 0 del código para las escalas de tiempo
DW46=ESCTM1	Bit 1 del código para las escalas de tiempo
DW47=ESCTM2	Bit 2 del código para las escalas de tiempo

DW50=ESCTM3	Bit 3 del código para las escalas de tiempo
DW51=CKT	Reloj de transmisión
DW52=-CLADR	Cuando es 0; inicializa el contador generador de direcciones de memoria
DW53=T/-M	Si es 1, modo de muestreo; si es 0, modo de transmisión
DW54=INVA	Invierte los datos digitales del canal A
DW55=INVB	Invierte los datos digitales del canal B
DW56=Expansión	
DW57=Expansión	

Para controlar individualmente cada bit de los puertos desde el software, se usan máscaras con funciones AND y OR. Por ejemplo si se quiere que DW40 sea 0 y DW41 a DW47 queden en su estado anterior se hace:

( DW47 DW46 DW45 DW44 DW43 DW42 DW41 DW40 ) =

( DW47 DW46 DW45 DW44 DW43 DW42 DW41 DW40 ) AND ( 11111110 )

Donde ( 11111110 ) es la máscara que fuerza a DW40 a 0.

Si se quiere que DW40 sea 1 y DW41 a DW47 queden en su estado anterior se hace:

$$( \text{DW47 DW46 DW45 DW44 DW43 DW42 DW41 DW40} ) = \\ ( \text{DW47 DW46 DW45 DW44 DW43 DW42 DW41 DW40} ) \text{ OR } ( 00000001 )$$

Donde ( 00000001 ) es la máscara que fuerza a DW40 a 1.

Con el control individual de cada bit de los puertos, cada señal del hardware se puede establecer individualmente desde el software. Ahora controlando un conjunto de bits, uno individualmente a la vez, se tiene lo que denominamos los comandos para cada bloque del hardware. A continuación se describirán los comandos usados para los bloques del hardware, estos comandos establecen las variables digitales (o líneas de control digital) de estos bloques a un estado que depende de los requerimientos de medición. Es importante destacar que cuando se da un comando que establece las variables digitales de cierto bloque, las variables de los otros bloques deben permanecer inalteradas, es decir en su estado anterior.

### **Comandos para el Sensor de Variables Eléctricas (SVE)**

Con estos comandos la PC establece los puertos que controlan este bloque a una configuración deseada, a través de instrucciones de escritura a los puertos (instrucción OUT).

COMANDO	ESTADO DE LAS VARIABLES DIGITALES DEL BLOQUE						
	CFS	VLA-VHA	VLB-VHB	CR0	CR1	CR2	RST
SVE_INI	0	0	0	1	1	1	0
SVE_VLA	0	1	A	A	A	A	0
SVE_VLB	0	A	1	0	0	0	0
SVE_VHA	0	0	A	A	A	A	0
SVE_VHB	0	A	0	0	0	0	0
SVE_C2LB	0	A	1	0	1	0	0
SVE_C1LB	0	A	1	1	1	0	0
SVE_C1HB	0	A	1	0	0	1	0
SVE_C2HB	0	A	1	1	0	1	0
SVE_CFSR1	1	1	1	1	0	0	0
SVE_CFSR2	1	1	1	0	1	0	0
SVE_FIN	0	0	0	1	1	1	0

"A" significa que se deja la variable en su estado anterior, es decir no es modificada, "SVE", son las siglas de Sensor de Variables Eléctricas.

La explicación de cada comando para el bloque SVE es:

COMANDO	ESTABLECE ESTE BLOQUE PARA MEDICIÓN DE
SVE_VLA	BAJOS VOLTAJES EN EL CANAL A (SIN DIVISOR)
SVE_VLB	BAJOS VOLTAJES EN EL CANAL B (SIN DIVISOR)
SVE_VHA	ALTOS VOLTAJES EN EL CANAL A (CON DIVISOR)
SVE_VHB	ALTOS VOLTAJES EN EL CANAL B (CON DIVISOR)
SVE_C2LB	MUY BAJA CORRIENTE EN EL CANAL B
SVE_C1LB	BAJA CORRIENTE EN EL CANAL B
SVE_C1HB	ALTA CORRIENTE EN EL CANAL B

SVE_C2HB	MUY ALTA CORRIENTE EN EL CANAL B
SVE_CFSR1	IMPEDANCIA (USANDO FTE.SEN. Y RES.PROG.EN R1)
SVE_CFSR2	IMPEDANCIA (USANDO FTE.SEN. Y RES.PROG.EN R2)

El comando SVE\_INI, da una clave de inicio para que los relevadores de este bloque sean establecidos a un estado seguro, esto es por si antes de ejecutarse el programa los puertos correspondientes a los relevadores están en un estado peligroso. El comando SVE\_FIN, deja los relevadores en un estado de protección, antes de salir del programa.

De la manera en que fue deducida la tabla del estado de las variables digitales del bloque SVE se dan unos ejemplos a continuación, se dará una explicación generalizada por grupo de comandos para medición de voltaje, otra por grupo de comandos para medición de corriente, y otra por grupo de comandos para medición de impedancia. El estado de las variables de los otros comandos fue deducida similarmente.

- **Medición de voltaje:** Cuando, por ejemplo, el comando SVE\_VLA es invocado se requiere medir bajos voltajes en el canal A (sin divisor), así la fuente senoidal no debe estar activa, obviamente no debe intervenir en la medición, así se establece  $CFS = 0$ . Como lo que se este midiendo en el canal B no importa,  $VLB/-VHB = A$  (estado anterior). Como en el canal A se requiere medir bajos voltajes (no se usa el divisor de voltaje)  $VLA/-VHA = 1$ . Como el resistor programable esta involucrado con el canal B, su valor no importa así que este se deja en su estado anterior,  $CR0=A$ ,  $CR1=A$ ,  $CR2 =A$ .
- **Medición de corriente:** Para medición de corriente es usado el canal B, aquí la fuente senoidal debe permanecer desactivada, así  $CFS = 0$ . Como el voltaje que se va a generar al pasar la corriente por el resistor programable es muy pequeño,

se utiliza el canal B para pequeños voltajes así  $V_{LB}/V_{HB} = 1$ . Como el canal A no importa,  $V_{LA}/V_{HA}$  permanece en su estado anterior ( $V_{LA}/V_{HA} = A$ ). El valor de la resistencia programable se establece de acuerdo a la magnitud de la corriente que se va a medir, por ejemplo para corrientes muy bajas se utiliza la resistencia  $R_2$  que es de  $100 \Omega$ , el código para establecer esta resistencia es  $CR_0=0$ ,  $CR_1=1$ ,  $CR_2=0$ . Los valores que puede tomar el resistor programable para medición de corriente son  $R_2=100\Omega$ ,  $R_3=10\Omega$ ,  $R_4=1\Omega$  y  $R_5=0.1\Omega$ .

- **Medición de impedancia:** Para la medición de impedancia son usados tanto el canal A como el B, aquí la fuente senoidal es utilizada, así  $CFS=1$ , su corriente pasa tanto por la impedancia desconocida (canal A), como por el resistor programable (canal B). Comparando las magnitudes y fases de los voltajes en estas dos impedancias se calcula el valor de la impedancia desconocida. Los dos canales se establecen para bajos voltajes, ya que la amplitud de la fuente senoidal es de  $1 V$  (por protección de los componentes a medir), por lo que  $V_{LB}/V_{HB} = 1$  y  $V_{LA}/V_{HA} = 1$ . El valor de la resistencia programable se establece de acuerdo al tipo y valor de la impedancia a medir, por ejemplo, una resistencia de  $200k\Omega$ , el resistor programable debe ser lo más cercano a esta, para que cuando pase la corriente no se genere un voltaje muy pequeño o muy alto, lo que dificultaría su medición. Así la resistencia óptima del resistor programable para este caso sería de  $R_1=100k\Omega$ , cuyo código digital es  $CR_0 = 1$ ,  $CR_1 = 0$ ,  $CR_2 = 0$ . Los valores que puede tomar el resistor programable para medición de impedancia son  $R_1=100k\Omega$  y  $R_2=100\Omega$ .

## Comandos para el Amplificador de Ganancia Programable (AGP)

En estos comandos la PC establece los puertos que controlan este bloque a una configuración deseada, a través de una instrucción de escritura a los puertos (instrucción OUT).

COMANDO	EDO. DE LAS VAR. DIG. DE ESTE BLOQUE			EXPLICACIÓN: ESTABLECE ESTE BLOQUE (CANAL A) PARA:
	AX2	AX5	AX10	
AGPA_X001	0	0	0	GANANCIA 1
AGPA_X002	1	0	0	GANANCIA 2
AGPA_X005	0	1	0	GANANCIA 5
AGPA_X010	1	1	0	GANANCIA 10
AGPA_X020	1	0	1	GANANCIA 20
AGPA_X050	0	1	1	GANANCIA 50
AGPA_X100	1	1	1	GANANCIA 100

COMANDO	EDO. DE LAS VAR. DIG. DE ESTE BLOQUE			EXPLICACIÓN: ESTABLECE ESTE BLOQUE (CANAL B) PARA:
	BX2	BX5	BX10	
AGPB_X001	0	0	0	GANANCIA 1
AGPB_X002	1	0	0	GANANCIA 2
AGPB_X005	0	1	0	GANANCIA 5
AGPB_X010	1	1	0	GANANCIA 10
AGPB_X020	1	0	1	GANANCIA 20
AGPB_X050	0	1	1	GANANCIA 50
AGPB_X100	1	1	1	GANANCIA 100

"AGPA" siglas de Amplificador de Ganancia Programable A.

"AGPB" siglas de: Amplificador de Ganancia Programable B.



Xnnn, debe entenderse como multiplica por nnn, donde nnn puede ser 001, 002, 005, 010, 020, 050 o 100.

El bloque que involucra los dos comandos anteriores tiene 4 amplificadores en configuración inversora, tres que generan la ganancia adecuada, y uno con ganancia unitaria. Como el número de inversores es par, la señal de salida no esta invertida con respecto a la señal de entrada. Tomando, por ejemplo, el comando AGPA\_X001 los 3 amplificadores que generan ganancia permanecen en 1, con lo que la ganancia total es 1 en el canal A. Para el comando AGPA\_X100 los tres amplificadores generan su ganancia correspondiente (ganancia de 2, de 5 y de 10) con lo que la ganancia total es de 100.

### Comandos para el Generador del Reloj de Muestreo (GRM)

En estos comandos la PC establece los puertos que controlan este bloque a una configuración deseada, a través de una instrucción de escritura a los puertos (instrucción OUT).

COMANDO	EDO. DE LAS VAR. DIG. DE ESTE BLOQUE			
	ESCTM3	ESCTM2	ESCTM1	ESCTM0
GRM_TM000000100	0	0	0	0
GRM_TM000000200	0	0	0	1
GRM_TM000000500	0	0	1	0
GRM_TM000001000	0	0	1	1
GRM_TM000002000	0	1	0	0
GRM_TM000005000	0	1	0	1
GRM_TM000010000	0	1	1	0
GRM_TM000020000	0	1	1	1
GRM_TM000050000	1	0	0	0
GRM_TM000100000	1	0	0	1

GRM_TM000200000	1	0	1	0
GRM_TM000500000	1	0	1	1
GRM_TM001000000	1	1	0	0
GRM_TM002000000	1	1	0	1
GRM_TM005000000	1	1	1	0
GRM_TM010000000	1	1	1	1

"GRM" siglas de: Generador del Reloj de Muestreo.

"TM" período (T) del reloj Muestreo.

Con estos comandos se establece el generador del reloj de muestreo para que se tomen muestras a un período especificado a la derecha de "TM", las unidades son nanosegundos. Por ejemplo: GRM\_TM000000100 establece el período del reloj de muestreo en 100 nanosegundos.

Una muestra es visualizada como un pixel en la pantalla, por lo tanto, y considerando el ejemplo anterior cada pixel representa 100 nanosegundos horizontalmente. Como una división esta constituida por 100 pixeles la escala será: ESC HOR = 10 microseg/DIV.

### **Comandos para el CAD y Almacenamiento Temporal (CAD)**

Para este bloque existen dos comandos: CAD\_INVA y CAD\_INVB, que hacen 1 la variable de control digital INV de el bloque CAD asociados al canal A y B respectivamente, Con lo que se invierte la señal del canal A o del canal B.

### 3.3. FUNCIONES PARA EL CONTROL DEL HARDWARE

Con las funciones se tiene un control del hardware desde un nivel más alto que de los comandos, los cuales solamente establecen el conjunto de bits correspondientes a cierto bloque. A continuación se describirá en forma general los objetivos de las funciones para el control del hardware desarrolladas en lenguaje C++, las cuales llaman a los comandos.

- A partir de un índice que indica la escala horizontal (o de tiempo), se establecen las variables de control en el hardware que genera el reloj de muestreo.
- A partir de un índice que indica la escala vertical (voltaje en A, voltaje en B o corriente en B), se establecen las variables de control en el hardware que corresponden a los bloques "Amplificador de Ganancia Programable" y "Sensor de Variables Eléctricas".
- Ya establecidas las variables de control de escala vertical y escala horizontal en el hardware, se toma un cierto número de muestras que se almacenan temporalmente en memoria RAM (modo de muestreo). Al terminar de tomar las muestras, estas se transmiten a la PC y son cargadas en vectores que denominamos *imga* e *imgb* (modo de transmisión).
- El bloque CTRL es el que maneja el protocolo de comunicación con la PC, dependiendo de las ordenes de la PC este bloque selecciona ya sea el reloj de muestreo o el de transmisión, configura y da información a los bloques CAD. La función que realiza esto, no sólo es una escritura a los puertos, sino un programa cuyo resultado en lenguaje de alto nivel se ve sólo como la carga de vectores con la información de las muestras tomadas con un rango comprendido entre 0 a 255 ya que los convertidores son de 8 bits.

Este programa se muestra a continuación en pseudocódigo, para entender su

funcionamiento es necesario haber revisado la parte del ANALISIS DINAMICO.

INICIO	
HACE T/-M = 0 HACE -CLADR = 0 HACE -CLADR = 1 ETIQUETA 1: SI CTATERM = 0 ENTONCES SALTA A ETIQUETA 1	MODO DE MUESTREO
HACE T/-M = 1 HACE -CLADR = 0 HACE -CLADR = 1 HACE i=0 ETIQUETA 2: HACE CKT = 1 HACE CKT = 0 SI CTATERM = 1 ENTONCES SALTA A ETIQUETA 3 HACE i=i+1 ASIGNA EL CONTENIDO DE LA RAM DEL CANAL A A $imga(i)$ ASIGNA EL CONTENIDO DE LA RAM DEL CANAL B A $imgb(i)$ SALTA A ETIQUETA 2 ETIQUETA 3:	MODO DE TRANSMISION
FIN.	

A continuación se da una explicación de lo anterior.

<ul style="list-style-type: none"> <li>• HACE T/M = 0</li> </ul>	Con esto se establece el modo de muestreo, en donde en memoria RAM se guardan las muestras del convertidor analógico digital.
<ul style="list-style-type: none"> <li>• HACE -CLADR = 0</li> <li>• HACE -CLADR = 1</li> </ul>	El estado de los contadores se hace cero y se dejan listos para ser incrementados.
<ul style="list-style-type: none"> <li>• ETIQUETA 1:</li> <li>• SI CTATERM = 0, SALTA A ETIQUETA 1</li> </ul>	La PC espera hasta que el número de muestras se halla completado, esto es hasta que CTATERM = 1.
<ul style="list-style-type: none"> <li>• HACE T/M = 1</li> </ul>	Al haberse completado el número de muestras, se inicia el modo de transmisión, en donde estas muestras se transmitirán a la PC. El modo de transmisión se establece con T/M = 1.
<ul style="list-style-type: none"> <li>• HACE -CLADR = 0</li> <li>• HACE -CLADR = 1</li> </ul>	El estado de los contadores se hace cero y se dejan listos para ser incrementados.
<ul style="list-style-type: none"> <li>• HACE i = 0</li> </ul>	Se inicializa el índice del vector imga[i] e imgb[i]
<ul style="list-style-type: none"> <li>• ETIQUETA 2:</li> <li>• HACE CKT = 1</li> <li>• HACE CKT = 0</li> </ul>	La PC da el reloj de transmisión, considerando el tiempo del ciclo de bus (que tiene mínimo 4 períodos de reloj de 4.77 MHz) las dos escrituras a puertos toman 2 microsegundos aprox.
<ul style="list-style-type: none"> <li>• SI CTATERM = 1, SALTA A ETIQUETA 3</li> </ul>	Si el número de datos transmitidos ya se completo, entonces termina el ciclo.
<ul style="list-style-type: none"> <li>• HACE I=I+1</li> <li>• ASIGNA EL CONTENIDO DE LA RAM CANAL A A imga(i)</li> <li>• ASIGNA EL CONTENIDO DE LA RAM CANAL B A imgb(i)</li> <li>• SALTA A ETIQUETA 2</li> </ul>	Si el número de datos transmitidos no se ha completado, entonces, el contenido de las RAM se asigna a imga(i) e imgb(i)
<ul style="list-style-type: none"> <li>• ETIQUETA 3:</li> <li>• FIN.</li> </ul>	

## 3.4. GRÁFICOS

Este modulo esta conformado por las funciones que se encargan de dar la presentación gráfica al proyecto. Los objetivos son:

- Establecer el modo gráfico y los colores que se utilizan en el proyecto. Más específicamente activa el modo gráfico VGA de alta resolución y 16 colores y se establece la paleta de colores adecuada.
- Cerrar el modo gráfico al terminar el programa.
- Dibujar botones (tipo windows) con texto en cierta posición.
- Dibujar la pantalla en la cual se visualizan las señales que representan las diferentes variables eléctricas, además de que se visualizan diferentes mensajes del o para el usuario.
- Dibujar visualizadores que muestran diversos valores de interés para el usuario.
- Escribir texto para las diversas presentaciones.
- Dibujo de rectángulos que simulan relieve o para agrupación de varios "objetos", tales como botones, visualizadores, etc. Generalmente a cada agrupación se le asocia un titulo en la parte superior.

### **3.5. TRANSFORMACIÓN DE UNIDADES**

Este grupo de funciones tiene el objeto de hacer las conversiones de unidades que se requieren en el proyecto. En función de ciertos índices, las cantidades en pixeles son multiplicadas por un factor, para obtener estas mismas cantidades en unidades reales (voltaje, corriente, etc.).

### **3.6. PROCESAMIENTO DE VECTORES IMAGEN DE LAS RAM**

Este grupo de funciones tiene por objeto analizar los vectores que contienen el código digital de las muestras tomadas por los convertidores A/D, para determinar valores útiles para el programa, así como valores y gráficos que se presentan al usuario. A continuación se da una explicación de las funciones de este modulo:

- Obtener los índices de inicio, a partir de los cuales las imágenes, tanto del canal A como del canal B, serán visualizadas. Estos índices de inicio sirven para el sincronizado de la secuencia de imágenes, de manera que se empiece a visualizar cada imagen en un punto determinado por la forma de onda de la señal.
- Visualizar las imágenes del canal A y del canal B a partir del índice de inicio.
- Calcular a partir del código digital asociado al canal A y/o al canal B: el período, la frecuencia, el valor máximo, el valor mínimo, impedancia (resistiva, capacitiva o inductiva) y potencia (aparente, real, reactiva y factor de potencia). Una vez calculados también son presentados al usuario.

## 3.7. CONTROL DEL MENÚ

En un menú, la selección de opciones en este proyecto es parecido al sistema de botones de windows 3.X. Es decir las opciones son presentadas en la pantalla en forma de botones. Cada opción es seleccionada presionando su botón asociado, a través del ratón, presionado el botón derecho o seleccionado el botón con TAB y presionando ENTER.

Dentro del programa se definen las diferentes características del menú de botones, estas son el número de opciones (o número de botones), las coordenadas de cada botón y el texto de cada botón.

## 3.8. VENTANAS

Cada ventana esta conformada por los gráficos de las funciones definidas en el modulo GRAFICOS, con un menú de botones especifico. Los objetivos de estas funciones a continuación se detallan:

- Dibujar los objetos que permanentemente están presentes en la presentación gráfica del proyecto, como relieves, títulos, líneas de agrupación. Así como la presentación original de los visualizadores que muestran las diferentes variables.
- Dibujar todos los botones que se utilizan en el proyecto en estado desactivo. Así antes de cambiar a un menú nuevo todos los botones se desactivan, posteriormente sólo se presentan como habilitados los del menú nuevo.
- Definir los diferentes tipos de menús de botones con los que interactúa el usuario:



Menú principal.	
Canal A: medición de voltaje	canal B: no usado.
Canal A: no usado	canal B: medición de corriente.
Canal A: medición de voltaje	canal B: medición de voltaje.
Canal A: medición de impedancia	canal B: no usado.
Canal A: medición de voltaje	canal B: medición de corriente.

A continuación se presentan las ilustraciones de las diversas ventanas, cada una constituye un menú de opciones para el usuario. Las opciones de cada menú son representadas por los botones activos (un botón activo no está tachado).

## Ventana, Menú Principal

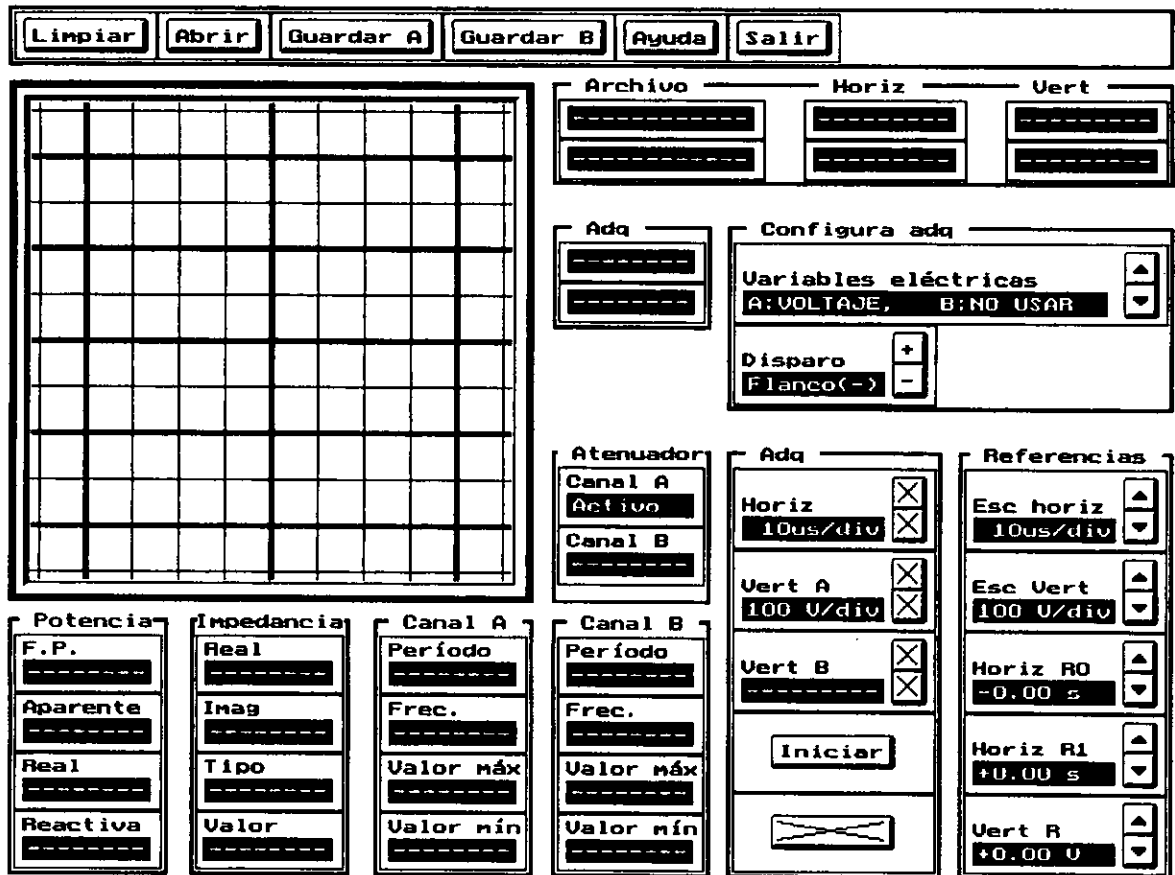


Figura 3.1

- **Limpia.** La pantalla y todos los visualizadores son establecidos en su condición inicial.
- **Abrir.** La información contenida en un archivo la muestra en pantalla. De este archivo también obtiene su nombre, su escala de tiempo (escala horizontal) y su escala vertical.
- **Guardar A.** La información en pantalla, la escala de tiempo y la escala vertical correspondientes al canal A es guardada en un archivo con cierto nombre.

- **Guardar B.** La información en pantalla, la escala de tiempo y la escala vertical correspondientes al canal B es guardada en un archivo con cierto nombre.
- **Ayuda.** Despliega información de la manera de utilizar el sistema.
- **Salir.** Salir al sistema operativo.
- **Variables eléctricas.** Se indica a que menú se tendrá acceso y se selecciona el tipo de variables eléctricas que se sensorarán durante la adquisición de datos, tiene 5 opciones:

A: VOLTAJE, B: NO USAR
A: NO USAR, B: CORRIENTE
A: VOLTAJE, B: VOLTAJE
A: IMPEDANCIA, B: NO USAR
A: VOLTAJE, B: CORRIENTE

- **Disparo.** Indica el tipo de disparo que se aplicará durante la adquisición. El disparo sirve para la sincronización de una señal, de manera que esta empiece a visualizarse desde el mismo punto, este punto puede ser el inicio de la mitad superior de la señal (flanco positivo) o el inicio de la mitad inferior de la señal (flanco negativo). Estas opciones son:  
Flanco(-)  
Flanco(+)
- **Esc horiz.** Aquí se selecciona la escala de las referencias horizontales, esta puede ser una de las siguientes opciones:

10 $\mu\text{s}/\text{div}$	20 $\mu\text{s}/\text{div}$	50 $\mu\text{s}/\text{div}$
100 $\mu\text{s}/\text{div}$	200 $\mu\text{s}/\text{div}$	500 $\mu\text{s}/\text{div}$
1 ms/div	2 ms/div	5 ms/div
10 ms/div	20 ms/div	50 ms/div
100 ms/div	200 ms/div	500 ms/div
1 s/div		

- **Esc vert.** Aquí se selecciona la escala de la referencia vertical, esta escala puede ser una de las siguientes opciones:

10 mV/div	20 mV/div	50 mV/div
100 mV/div	200 mV/div	500 mV/div
1 V/div	2 V/div	5 V/div
10 V/div	20 V/div	50 V/div
100 V/div	100 $\mu\text{A}/\text{div}$	200 $\mu\text{A}/\text{div}$
500 $\mu\text{A}/\text{div}$	1 mA/div	2 mA/div
5 mA/div	10 mA/div	20 mA/div
500 mA/div	1 A/div	2 A/div
5 A/div		

- **Horiz R0.** Con esto se cambia la posición de la referencia horizontal R0, su valor es con respecto a la referencia horizontal R1 y depende de la escala horizontal establecida para las referencias horizontales.
- **Horiz R1.** Con esto se cambia la posición de la referencia horizontal R1, su valor es con respecto a la referencia horizontal R0 y depende de la escala horizontal establecida para las referencias horizontales.
- **Vert R.** Se cambia la posición de la referencia vertical, su valor es con respecto a cero, que es centro de la pantalla verticalmente y depende de la escala vertical establecida para la referencia vertical.

- **Iniciar.** Se inicia la adquisición de datos y se cambia al menú correspondiente al tipo de variables ya seleccionadas, con esto ya se pueden empezar a utilizar los botones agrupados bajo "Adq" (todos los demás quedan inactivos).

En los visualizadores se muestran características de la señales que se están sensando de acuerdo a la variable que se esta midiendo. Si se esta utilizando el canal A y el canal B no, sólo se presenta la información de la agrupación "canal A". Si se están usando los dos canales se presenta información tanto en la agrupación "Canal A" como en la agrupación "Canal B". Cuando se mide impedancia y potencia se presenta información en las agrupaciones "Canal A" y "Canal B", así como en la agrupación "Impedancia" o "Potencia", según sea el caso. Para el caso de mediciones de voltaje, el atenuador para cada canal esta activo para escalas mayores a 1 V/div. En la agrupación "Adq", se muestra el canal o canales que se esta usando con el mismo color que su señal asociada que se esta presentando en pantalla. En la agrupación "Archivo" se presentan los nombres de los archivos que fueron abiertos con la opción "Abrir". Cuando se abre un archivo, también es visualizada en la pantalla su señal asociada con sus respectivas escalas horizontal y vertical.

Ventana, A: Voltaje, B: No Usar

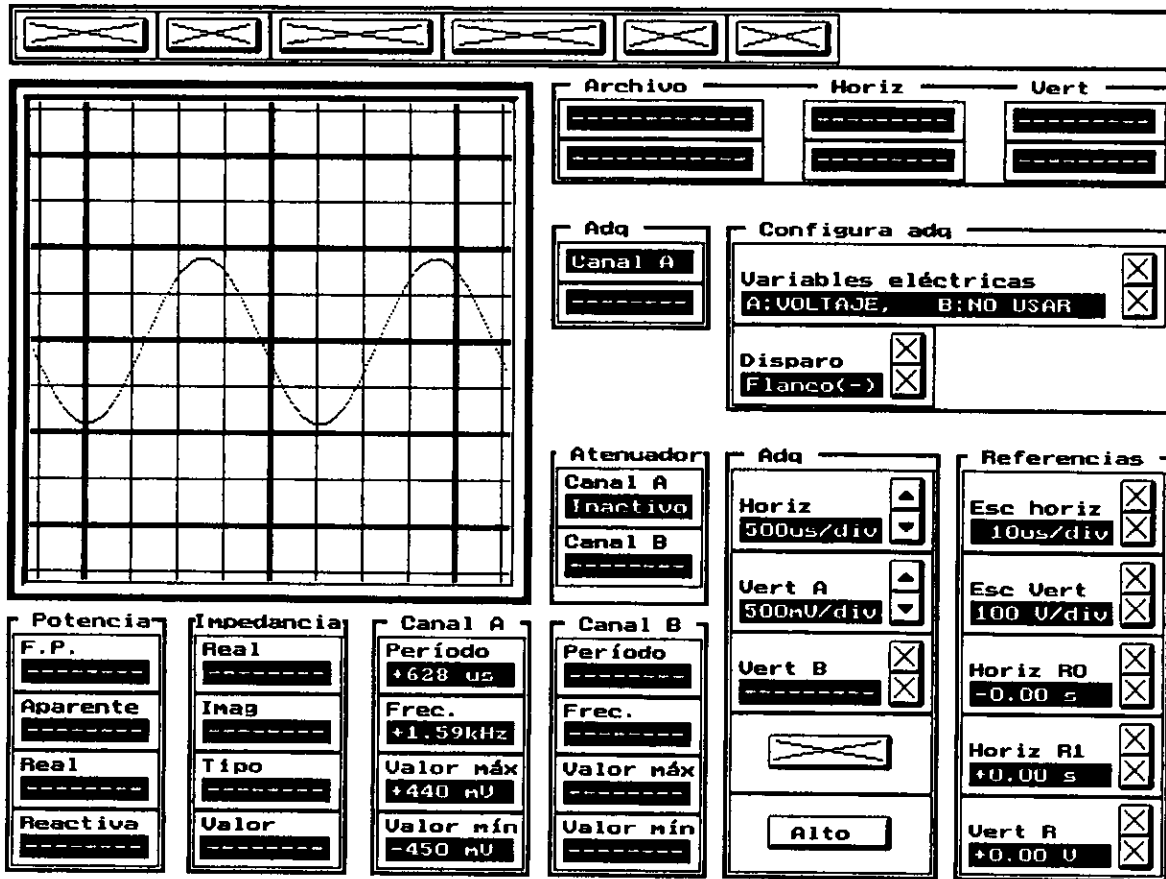


Figura 3.2

- **Horiz.** Cambia la escala horizontal (de tiempo) de la señal que esta siendo visualizada en la pantalla, se tienen disponibles las siguientes opciones:

10 $\mu$ s/div	20 $\mu$ s/div	50 $\mu$ s/div
100 $\mu$ s/div	200 $\mu$ s/div	500 $\mu$ s/div
1 ms/div	2 ms/div	5 ms/div
10 ms/div	20 ms/div	50 ms/div
100 ms/div	200 ms/div	500 ms/div
1 s/div		

- **Vert A.** Cambia la escala vertical (para este caso voltaje) de la señal del canal A que esta siendo visualizada en la pantalla, se tienen disponibles las siguientes opciones:

10 mV/div	20 mV/div	50 mV/div
100 mV/div	200 mV/div	500 mV/div
1 V/div	2 V/div	5 V/div
10 V/div	20 V/div	50 V/div
100 V/div		

- **Alto.** Termina la adquisición de datos y cambia al menú principal.

## Ventana, A: No Usar, B: Corriente

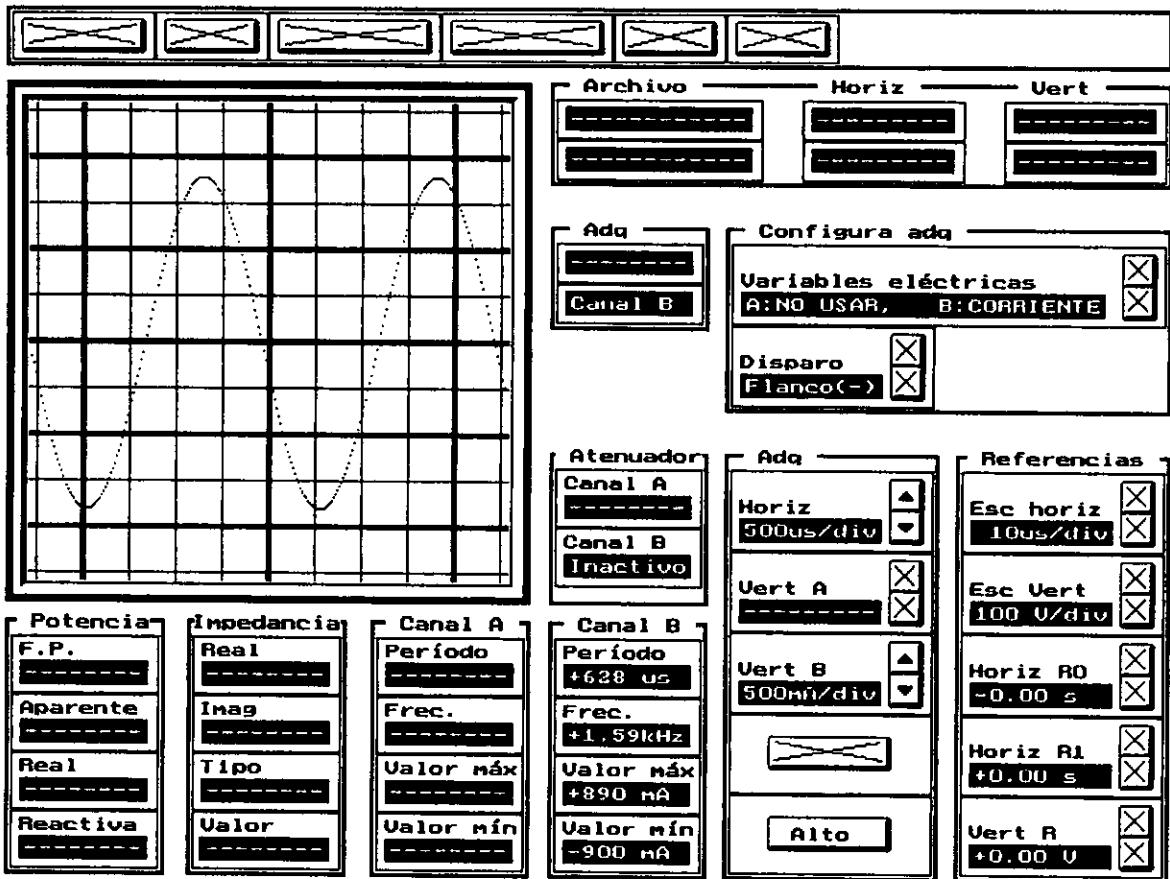


Figura 3.3

- **Horiz.** Cambia la escala horizontal (de tiempo) de la señal que esta siendo visualizada en la pantalla, se tienen disponibles las siguientes opciones:

10 $\mu$ s/div	20 $\mu$ s/div	50 $\mu$ s/div
100 $\mu$ s/div	200 $\mu$ s/div	500 $\mu$ s/div
1 ms/div	2 ms/div	5 ms/div
10 ms/div	20 ms/div	50 ms/div
100 ms/div	200 ms/div	500 ms/div
1 s/div		



- **Vert B.** Cambia la escala vertical (para este caso corriente) de la señal de canal B que esta siendo visualizada en la pantalla, se tienen disponibles las siguientes opciones:

100 $\mu$ A/div	200 $\mu$ A/div	500 $\mu$ A/div
1 mA/div	2 mA/div	5 mA/div
10 mA/div	20 mA/div	50 mA/div
100 mA/div	200 mA/div	500 mA/div
1 A/div	2 A/div	5 A/div

- **Alto.** Termina la adquisición de datos y cambia al menú principal.

En los visualizadores se muestran características de la señal que se esta sensando en el canal B. En la agrupación "Canal B", se actualizan el período, la frecuencia, el valor máximo y el valor mínimo.

## Ventana, A: Voltaje, B: Voltaje

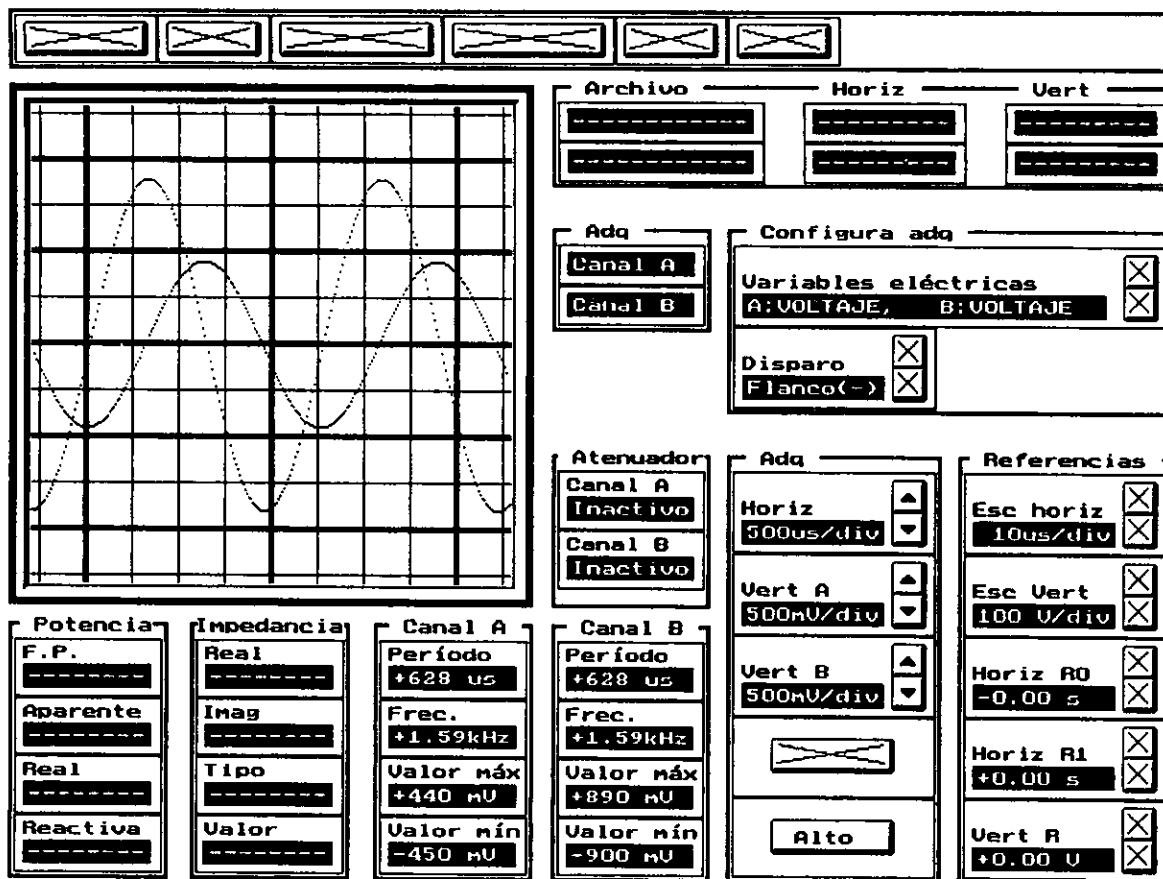


Figura 3.4

- **Horiz.** Cambia la escala horizontal (de tiempo) de las señales que están siendo visualizadas en la pantalla, se tienen disponibles las siguientes opciones:

10 $\mu\text{s}/\text{div}$	20 $\mu\text{s}/\text{div}$	50 $\mu\text{s}/\text{div}$
100 $\mu\text{s}/\text{div}$	200 $\mu\text{s}/\text{div}$	500 $\mu\text{s}/\text{div}$
1 ms/div	2 ms/div	5 ms/div
10 ms/div	20 ms/div	50 ms/div
100 ms/div	200 ms/div	500 ms/div
1 s/div		

- **Vert A.** Cambia la escala vertical (para este caso voltaje) de la señal del canal A que esta siendo visualizada en la pantalla, se tienen disponibles las siguientes opciones:

10 mV/div	20 mV/div	50 mV/div
100 mV/div	200 mV/div	500 mV/div
1 V/div	2 V/div	5 V/div
10 V/div	20 V/div	50 V/div
100 V/div		

- **Vert B.** Cambia la escala vertical (para este caso voltaje) de la señal del canal B que esta siendo visualizada en la pantalla, se tienen disponibles las siguientes opciones:

10 mV/div	20 mV/div	50 mV/div
100 mV/div	200 mV/div	500 mV/div
1 V/div	2 V/div	5 V/div
10 V/div	20 V/div	50 V/div
100 V/div		

- **Alto.** Termina la adquisición de datos y cambia al menú principal.

## Ventana, A: Impedancia, B: No Usar

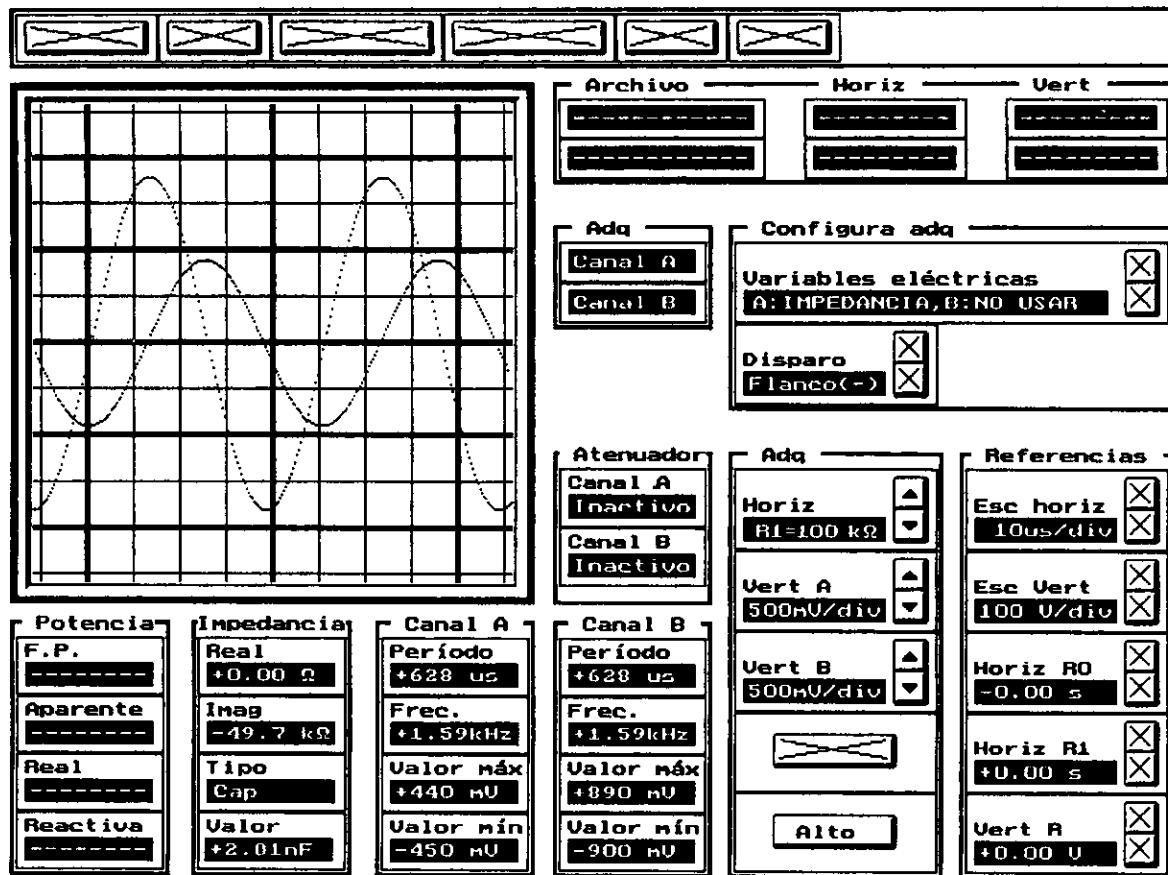


Figura 3.5

- **Horiz.** Cambia el valor del resistor programable (impedancia conocida) que se utiliza en el circuito serie implementado para determinar el valor de la impedancia desconocida. Se tiene dos opciones:

$R1=100\text{ k}\Omega$

$R2=100\ \Omega$

Aquí la escala de tiempo no la puede controlar el usuario, ya que la frecuencia de la fuente senoidal esta fija a 1.59 kHz (10000 radianes/seg.).

- **Vert A.** Cambia la escala vertical (para este caso voltaje) de la señal del canal A que esta siendo visualizada en la pantalla. Esta señal corresponde a la caída de

potencial en la impedancia desconocida. Como la amplitud de voltaje de la fuente senoidal es de 1 V , no es necesario que el atenuador este activo por esto la escala más grande es de 1 V/div. Así se tienen disponibles las siguientes opciones:

10 mV/div	20 mV/div	50 mV/div
100 mV/div	200 mV/div	500 mV/div
1 V/div		

- **Vert B.** Cambia la escala vertical (para este caso voltaje) de la señal del canal B que esta siendo visualizada en la pantalla. Esta señal corresponde a la caída de potencial en el resistor programable. Como la amplitud de voltaje de la fuente senoidal es de 1 V , no es necesario que el atenuador este activo por esto la escala más grande es de 1 V/div. Así se tienen disponibles las siguientes opciones:

10 mV/div	20 mV/div	50 mV/div
100 mV/div	200 mV/div	500 mV/div
1 V/div		

- **Alto.** Termina la adquisición de datos y cambia al menú principal.

El usuario cambia el valor del resistor programable, la escala vertical del canal A y la escala vertical del canal B, de manera que en la pantalla se presenten las señales de voltaje con la mayor información posible (que sean lo más grande posible, sin salirse de la pantalla), esto para tener la mejor precisión al momento del calculo de la impedancia. El calculo de la impedancia se basa en la información presente en la pantalla, en las escalas de voltaje, en el valor del resistor programable y en ecuaciones que se detallan en el apéndice II.

## Ventana, A: Voltaje, B: Corriente

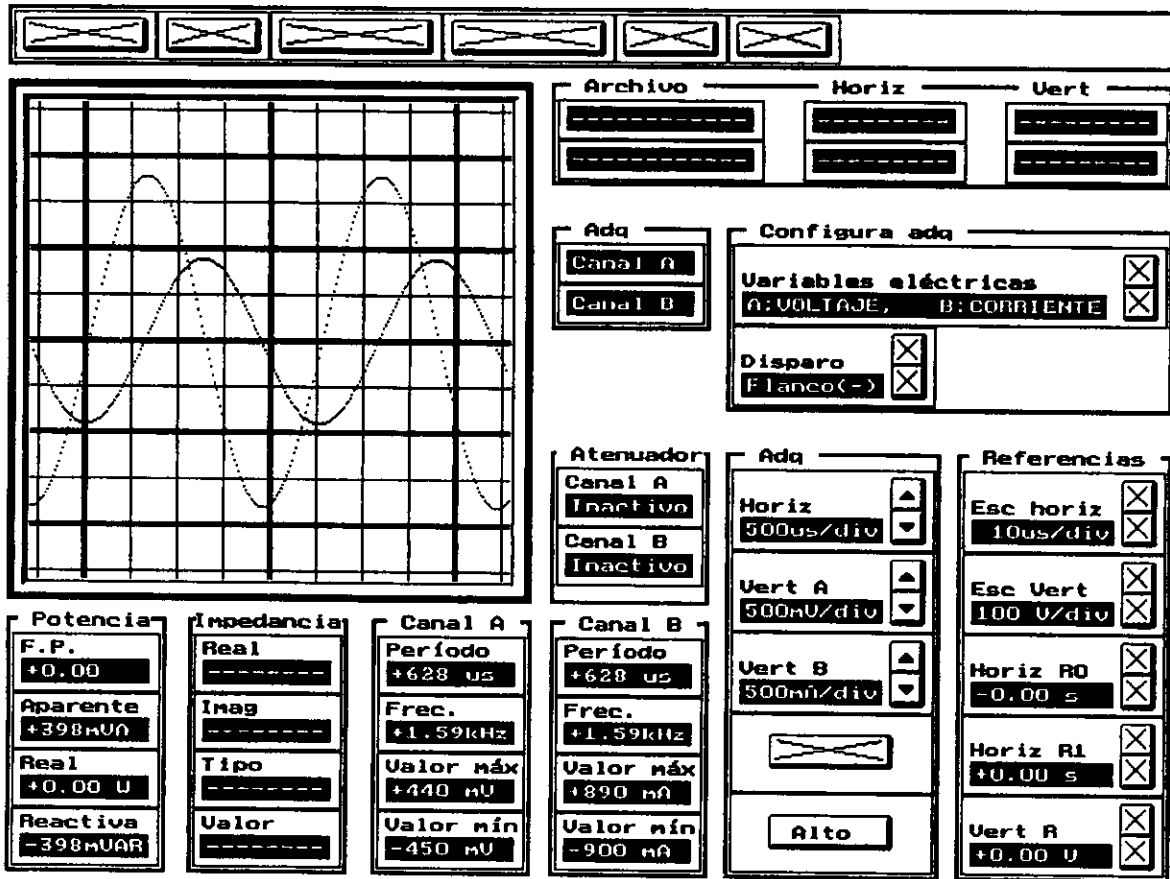


Figura 3.6

- **Horiz.** Cambia la escala horizontal (de tiempo) de las señales que están siendo visualizadas en la pantalla, se tienen disponibles las siguientes opciones:

10 $\mu$ s/div	20 $\mu$ s/div	50 $\mu$ s/div
100 $\mu$ s/div	200 $\mu$ s/div	500 $\mu$ s/div
1 ms/div	2 ms/div	5 ms/div
10 ms/div	20 ms/div	50 ms/div
100 ms/div	200 ms/div	500 ms/div
1 s/div		

- **Vert A.** Cambia la escala vertical (para este caso voltaje) de la señal del canal A que esta siendo visualizada en la pantalla, se tienen disponibles las siguientes opciones:

10 mV/div	20 mV/div	50 mV/div
100 mV/div	200 mV/div	500 mV/div
1 V/div	2 V/div	5 V/div
10 V/div	20 V/div	50 V/div
100 V/div		

- **Vert B.** Cambia la escala vertical (para este caso corriente) de la señal de canal B que esta siendo visualizada en la pantalla, se tienen disponibles las siguientes opciones:

100 $\mu$ A/div	200 $\mu$ A/div	500 $\mu$ A/div
1 mA/div	2 mA/div	5 mA/div
10 mA/div	20 mA/div	50 mA/div
100 mA/div	200 mA/div	500 mA/div
1 A/div	2 A/div	5 A/div

- **Alto.** Termina la adquisición de datos y cambia al menú principal.

El usuario cambia la escala vertical del canal A y la escala vertical del canal B, de manera que en la pantalla se presenten las señales de voltaje y corriente con la mayor información posible (que sean lo más grande posible, sin salirse de la pantalla), esto para tener la mejor precisión al momento del calculo de la potencia. El calculo de la potencia se basa en la información presente en la pantalla, en la escala de voltaje, en las escala de corriente y en ecuaciones que se detallan en el apéndice II.

### 3.9. PRINCIPAL

En el programa principal se emplean todas las funciones ya detalladas y se encarga de manejar el sistema de menús a través de instrucciones switch, por ejemplo para un menú de n opciones:

```
salir_del_menú = falso
mientras (salir_del_menú es falso)
{
  sensa opción del usuario
  switch(opción)
  {
    caso 1:
      salir_del_menú = verdadero
      break;
    caso 2:
      procedimiento 2;
      break;
    .
    .
    .
    caso n:
      procedimiento n
      break;
  }
}
```

En la instrucción switch, la ejecución del programa pasa al caso i, donde i = opción. El procedimiento i, es el asociado con a la opción i, o lo que es lo mismo, a la selección de un botón, por ejemplo, la opción 1 fue asociada con el botón salir, inclusive un



procedimiento puede dar acceso a otro menú.

Así cuando se pulsa el botón iniciar (ver figuras 3.1 a 3.6) el programa pasa a un submenú, el cual depende de la variable eléctrica escogida por el usuario.

Al entrar a cualquiera de estos submenús se establecen los valores por defecto de las escalas, se inactivan todos los botones, y se llama a la función de ventana con lo que se define el menú de botones correspondiente, con esto quedan activados sólo los botones involucrados con este submenú.

Las escalas son cargadas en el hardware. Posteriormente del hardware se cargan los vectores que contienen las muestras dadas por los convertidores A/D. De cada vector que contiene 512 muestras se selecciona un índice de inicio para sincronización, a partir del cual se visualiza la señal (se presentan 256 muestras). Una vez desplegadas las señales en la pantalla se presentan características de estas en los visualizadores. El usuario puede cambiar las escalas al mismo tiempo que el sistema esta sensando las variables eléctricas, esto es, el cambio de escala horizontal (tiempo) y el cambio de escala vertical A y/o B. Las unidades de las escalas verticales dependen del de la variable eléctrica que se esta sensando. Cada submenú tiene la opción "Alto", con lo que la ejecución del programa regresa al menú principal.

---

# CAPÍTULO 4

## *ESPECIFICACIONES DEL SISTEMA DE MEDICIÓN*

---

Aquí se presentan varias especificaciones del sistema de medición, las cuales son características de la pantalla (número de puntos, divisiones y subdivisiones), el rango, resolución y otras especificaciones para cada variable eléctrica, así como la respuesta en frecuencia del sistema.

### 4.1. PANTALLA

HORIZONTALMENTE (TIEMPO)

250 puntos, 2.5 div, 10 subdiv, 100 puntos/div, 25 puntos/subdiv, 2 subdiv/div

VERTICALMENTE

250 puntos, 5 div, 10 subdiv, 50 puntos/div, 25 puntos/subdiv, 4 subdiv/div

### 4.2. TIEMPO

ESCALA	RANGO	RESOLUCIÓN
10 $\mu$ s/div	25 $\mu$ s	100ns
20 $\mu$ s/div	50 $\mu$ s	200ns
50 $\mu$ s/div	125 $\mu$ s	500ns
100 $\mu$ s/div	250 $\mu$ s	1 $\mu$ s

1ms/div	2.5ms	10 $\mu$ s
2ms/div	50ms	20 $\mu$ s
5ms/div	12.5ms	50 $\mu$ s
10ms/div	25ms	100 $\mu$ s
20ms/div	500ms	200 $\mu$ s
50ms/div	125ms	500 $\mu$ s
100ms/div	250ms	1ms
200ms/div	500ms	2ms
500ms/div	1.25s	5ms
1 s/div	2.5 s	10ms

### 4.3. VOLTAJE

ESCALA	RANGO	RESOLUCIÓN	R	G	EP	PROTECCIÓN	DIVISOR
10mV/div	$\pm 25$ mV	200 $\mu$ V	10 <sup>12</sup>	100	200 $\mu$ V	12 V	INACTIVO
20mV/div	$\pm 50$ mV	400 $\mu$ V	10 <sup>12</sup>	50	400 $\mu$ V	12 V	INACTIVO
50mV/div	$\pm 125$ mV	1mV	10 <sup>12</sup>	20	1mV	12 V	INACTIVO
100mV/div	$\pm 250$ mV	2mV	10 <sup>12</sup>	10	2mV	12 V	INACTIVO
200mV/div	$\pm 500$ mV	4mV	10 <sup>12</sup>	5	4mV	12 V	INACTIVO
500mV/div	$\pm 1250$ mV	10mV	10 <sup>12</sup>	2	10mV	12 V	INACTIVO
1 V/div	$\pm 2.5$ V	20mV	10 <sup>12</sup>	1	20mV	12 V	INACTIVO
2 V/div	$\pm 5$ V	40mV	100k	50	40mV	1200 V	ACTIVO
5 V/div	$\pm 12.5$ V	10mV	100k	20	10mV	1200 V	ACTIVO
10 V/div	$\pm 25$ V	200mV	100k	10	200mV	1200 V	ACTIVO
20 V/div	$\pm 50$ V	400mV	100k	5	400mV	1200 V	ACTIVO
50 V/div	$\pm 125$ V	1V	100k	2	1V	1200 V	ACTIVO
100 V/div	$\pm 250$ V	2V	100k	1	2V	1200 V	ACTIVO

R: Impedancia de entrada [ $\Omega$ ].

G: Ganancia de voltaje

EP: Error de precisión por cuantificación

## 4.4. CORRIENTE

E <sub>B</sub>	RANGO	RESOLUCIÓN	R <sub>B</sub>	V <sub>B</sub>	G <sub>B</sub>	E <sub>P</sub>	PROTECCIÓN
100µA/div	±250µA	2µA	0.1	1.25	2	2µA	30 A, 12 V
200µA/div	±500µA	4µA	0.1	0.5	5	4µA	30 A, 12 V
500µA/div	±1250µA	10µA	0.1	0.25	10	10µA	30 A, 12 V
1mA/div	±2.5Ma	20µA	0.1	0.125	20	20µA	30 A, 12 V
2mA/div	±5mA	40µA	0.1	0.05	50	40µA	30 A, 12 V
5mA/div	±12.5mA	100µA	1	0.25	10	100µA	3 A, 12 V
10mA/div	±25mA	200µA	1	0.125	20	200µA	3 A, 12 V
20mA/div	±50mA	400µA	1	0.05	50	400µA	3 A, 12 V
50mA/div	±125mA	1Ma	10	0.25	10	1Ma	0.3 A, 12 V
100mA/div	±250mA	2mA	10	0.125	20	2mA	0.3 A, 12 V
200mA/div	±500mA	4mA	10	0.5	50	4mA	0.3 A, 12 V
500mA/div	±1250mA	10mA	100	0.25	10	10mA	0.3 A, 12 V
1 A/div	±2.5 A	20mA	100	0.125	20	20mA	0.3 A, 12 V
2 A/div	±5 A	40mA	100	0.05	50	40mA	0.3 A, 12 V
5 A/div	±12.5 A	10mA	100	0.01	100	10mA	0.3 A, 12 V

E<sub>B</sub>: Escala de corriente en el canal B, configurado por el usuario [A/DIV].

R<sub>B</sub>: Valor del resistor programable (resistor de prueba) [Ω].

V<sub>B</sub>: Caída máxima de voltaje, al pasar la corriente por el resistor programable [V].

G<sub>B</sub>: Ganancia de voltaje en el canal B

E<sub>P</sub>: Error de precisión por cuantificación

## 4.5. POTENCIA

Cuando el canal A esta configurado para medir voltaje y el canal B para medir corriente, el programa, a partir del voltaje en A y de la corriente en B, calcula el factor de potencia, la potencia real, la potencia aparente y la potencia reactiva.

En la siguiente tabla se da el rango y la resolución de la potencia aparente (valor

en el canal A y en el canal B, etc., no se presentan ya que son los mismos que los especificados en las tablas de voltaje y corriente dadas por separado.

EA	EB	RANGO	RESOLUCIÓN
10mV/div	100μA/div	±6.25μVA	400Pva
10mV/div	200μA/div	±12.5μVA	800pVA
10mV/div	500μA/div	±31.25μVA	2nVA
10mV/div	1mA/div	±62.5μVA	4nVA
10mV/div	2mA/div	±125μVA	8nVA
10mV/div	5mA/div	±312.5μVA	20nVA
10mV/div	10mA/div	±625μVA	40nVA
10mV/div	20mA/div	±1.25mVA	80nVA
10mV/div	50mA/div	±3.125mVA	200nVA
10mV/div	100mA/div	±6.25mVA	400nVA
10mV/div	200mA/div	±12.5mVA	800nVA
10mV/div	500mA/div	±31.25mVA	2μVA
10mV/div	1 A/div	±62.5mVA	4μVA
10mV/div	2 A/div	±125mVA	8μVA
10mV/div	5 A/div	±312.5mVA	20μVA
20mV/div	100μA/div	±12.5μVA	800pVA

.  
 .  
 .  
 .  
 .  
 .  
 .  
 .  
 .  
 .  
 .  
 .  
 .  
 .  
 .  
 .  
 .  
 .  
 .  
 .

50V/div	5 A/div	$\pm 1.5625\text{kVA}$	100mVA
100V/div	100 $\mu\text{A}$ /div	$\pm 62.5\text{mVA}$	4 $\mu\text{VA}$
100V/div	200 $\mu\text{A}$ /div	$\pm 125\text{mVA}$	8 $\mu\text{VA}$
100V/div	500 $\mu\text{A}$ /div	$\pm 312.5\text{mVA}$	20 $\mu\text{VA}$
100V/div	1Ma/div	$\pm 625\text{mVA}$	40 $\mu\text{VA}$
100V/div	2mA/div	$\pm 1.25\text{VA}$	80 $\mu\text{VA}$
100V/div	5mA/div	$\pm 3.125\text{VA}$	200 $\mu\text{VA}$
100V/div	10mA/div	$\pm 6.25\text{VA}$	400 $\mu\text{VA}$
100V/div	20mA/div	$\pm 12.5\text{VA}$	800 $\mu\text{VA}$
100V/div	50mA/div	$\pm 31.25\text{VA}$	2mVA
100V/div	100mA/div	$\pm 62.5\text{VA}$	4mVA
100V/div	200mA/div	$\pm 125\text{VA}$	8mVA
100V/div	500mA/div	$\pm 312.5\text{VA}$	20mVA
100V/div	1 A/div	$\pm 625\text{VA}$	40mVA
100V/div	2 A/div	$\pm 1.25\text{kVA}$	80mVA
100V/div	5 A/div	$\pm 3.125\text{kVA}$	200mVA

## 4.6. IMPEDANCIA

En la siguiente tabla se dan valores adecuados de configuración por parte del usuario ( $R_B$ ,  $E_A$  y  $E_B$ ) para varios valores de la magnitud de la impedancia desconocida ( $Z_A$ )

$Z_A$	$R_B$	$E_A$	$E_B$	$G_A$	$G_B$	$V_A$	$V_B$	$N_A$	$N_B$	%EP
0.1	100	0.010	0.500	100	2	0.001	0.999	10	200	10
1	100	0.010	0.500	100	2	0.010	0.990	100	198	1
10	100	0.050	0.500	20	2	0.091	0.909	182	182	0.5
100	100	0.200	0.200	5	5	0.500	0.500	250	250	0.4
1K	100	0.500	0.050	2	20	0.909	0.091	182	182	0.5
10K	100k	0.050	0.500	20	2	0.091	0.909	182	182	0.5
100K	100k	0.200	0.200	5	5	0.500	0.500	250	250	0.4

10M	100k	0.500	0.010	2	100	0.990	0.010	198	100	1
100M	100k	0.500	0.010	2	100	0.999	0.001	200	10	10

Z<sub>A</sub>: Magnitud de la impedancia desconocida, colocada por el usuario [Ω]

R<sub>B</sub>: Resistor programable, configurado por el usuario [Ω]

E<sub>A</sub>: Escala de voltaje en el canal A, configurado por el usuario [V/DIV]

E<sub>B</sub>: Escala de voltaje en el canal B, configurado por el usuario [V/DIV]

G<sub>A</sub>: Ganancia de voltaje en el canal A

G<sub>B</sub>: Ganancia de voltaje en el canal B

V<sub>A</sub>: Amplitud de la caída de voltaje (magnitud) en la reactancia desconocida [V]

V<sub>B</sub>: Amplitud de la caída de voltaje (magnitud) en el resistor programable [V]

N<sub>A</sub>: Niveles de información asociados al canal A [NIV]

N<sub>B</sub>: Niveles de información asociados al canal B [NIV]

E<sub>P</sub>: Error de precisión por cuantificación [%]

$$G_A = \frac{1 \text{ [V / DIV]}}{E_A} \text{ , normalización del voltaje}$$

$$G_B = \frac{1 \text{ [V / DIV]}}{E_B} \text{ , normalización del voltaje}$$

$$V_A = 1 \text{ [V]} \frac{R_A}{R_A + R_B}$$

$$V_B = 1 \text{ [V]} \frac{R_B}{R_A + R_B}$$

$$N_A = 2 \frac{V_A G_A 50 \text{ [NIV / DIV]}}{1 \text{ [V / DIV]}}$$

$$N_B = 2 \frac{V_B G_B 50 \text{ [NIV / DIV]}}{1 \text{ [V / DIV]}}$$

$$E_P = \frac{1}{N_{MIN}} 100\%$$

Donde N<sub>MIN</sub> es el menor entre N<sub>A</sub> y N<sub>B</sub>

Resistiva  $R_A = Z_A$

Capacitiva  $C_A = \frac{1}{\omega Z_A}$

Inductiva  $L_A = \frac{Z_A}{\omega}$

Donde  $\omega = 10000$  rad/seg es la velocidad angular de la fuente senoidal (1.59 kHz)

Para un error de precisión por cuantificación menor o igual al 1% se tienen los siguientes rangos:

Resistencia  $1 \Omega$  a  $10 M\Omega$

Capacitancia  $10 \text{ pF}$  a  $100 \mu\text{F}$

Inductancia  $100 \mu\text{H}$  a  $1 \text{ kH}$

## 4.7. RESPUESTA EN FRECUENCIA

AMP. DE GANANCIA PROG. PARA UNA GANANCIA DE 1 CONFIGURACIÓN: A:VOLTAJE, B:NO USAR; ESCALA: 1 V/div VOLTAJE DE PRUEBA: SENOIDAL DE 1 V DE AMPLITUD (DIVISOR INACTIVO)	
FRECUENCIA	VALOR MEDIDO/VALOR REAL
100 Hz	1
200 Hz	1
500 Hz	1
1 kHz	1
2 kHz	1
5 kHz	1



50 kHz	1
100 kHz	1
200 kHz	0.90
500 kHz	0.75
1 MHz	0.30
2 MHz	0.10

AMP. DE GANANCIA PROG. PARA GANANCIA DE 10	
CONFIGURACIÓN: A:VOLTAJE, B:NO USAR; ESCALA: 100mV/div	
VOLTAJE DE PRUEBA: SENOIDAL DE 100 mV DE AMPLITUD (DIVISOR INACTIVO)	
FRECUENCIA	VALOR MEDIDO/VALOR REAL
100 Hz	1
200 Hz	1
500 Hz	1
1 kHz	1
2 kHz	1
5 kHz	1
10 kHz	1
20kHz	1
50 kHz	1
100 kHz	1
200 kHz	0.75
500 kHz	0.55
1 MHz	0.20
2 MHz	0.05

AMP. DE GANANCIA PROG. PARA UNA GANANCIA DE 100	
CONFIGURACIÓN: A:VOLTAJE, B:NO USAR; ESCALA: 10 mV/div	
VOLTAJE DE PRUEBA: SENOIDAL DE 10 mV de AMPLITUD (DIVISOR INACTIVO)	
FRECUENCIA	VALOR MEDIDO/VALOR REAL
100 Hz	1
200 Hz	1
500 Hz	1
1 kHz	1
2 kHz	1
5 kHz	1
10 kHz	1
20kHz	1
50 kHz	1
100 kHz	1
200 kHz	0.60
500 kHz	0.40
1 MHz	0.10
2 MHz	0.00

---

# CONCLUSIONES

---

Con el medidor universal hemos conseguido la sustitución de varios instrumentos de medición, logrando la adquisición y medición de variables eléctricas sin contar con los instrumentos físicos. Estas variables eléctricas son: voltaje, frecuencia, corriente, impedancia y potencia. En muchos aspectos se parece a un osciloscopio, ya que cuenta con escalas horizontal (de tiempo), vertical (voltaje o corriente) y la forma de onda de las variables voltaje y corriente son graficadas. Las variables: frecuencia, impedancia y potencia son visualizadas numéricamente. Las escalas horizontal y vertical se obtuvieron a partir de las escalas mostradas por los instrumentos analógicos existentes.

Con la utilización de la técnica de diseño funcional descendente Top-Down, se organizo y simplifíco el diseño del sistema, ya que parte de un problema que se va particionando en módulos, siguiendo una secuencia organizada de pasos. Así el diseño fue sencillo, en forma modular y con componentes que se encuentran fácilmente en el mercado. Con lo anterior este proyecto puede ser de utilidad para el desarrollo de proyectos más completos. Es necesario que en México se realicen más investigaciones de este tipo de proyectos, ya que los dispositivos de instrumentación digital existentes en el mercado, son importados, muy costosos y generalmente no miden una amplia gama de variables eléctricas. Presentamos este proyecto como una alternativa más económica, versátil y sencilla a los anteriores, para su uso en laboratorios de investigación, en laboratorios a nivel educacional, para mantenimiento correctivo y/o preventivo de equipo, etc.

El medidor universal puede usarse como un equipo alternativo a aquellos utilizados en la medición de las variables eléctricas ya mencionadas. Es de gran ventaja para usuarios que cuentan con una computadora personal (AT 286 en adelante) pero no

con todos los instrumentos de medición de las variables eléctricas. Las lecturas de las variables eléctricas están disponibles en archivos en la PC, con lo cual desde un mismo lugar éstas se pueden procesar individual o conjuntamente.

El proyecto se diseñó para dos canales, sin embargo se realizó un sistema mínimo con un sólo canal, donde se mide únicamente voltaje en tiempo real. La realización de un sistema mínimo con un sólo canal demuestra la validez del diseño y la viabilidad para la construcción de un sistema completo, ya que ambos canales son iguales.

Por otra parte el medidor está diseñado en forma modular, de manera que para cualquier cambio no se tendrá que modificar todo el sistema, sino solamente la parte involucrada, lo que no altera de manera significativa la modularidad del diseño.

Se realizaron pruebas para verificar el correcto funcionamiento del proyecto, tomando como instrumento patrón un osciloscopio, comparando magnitud y frecuencia de las lecturas del osciloscopio y del medidor universal para diferentes escalas, se obtuvieron resultados satisfactorios.

Creemos que en un análisis de optimización futura, el sistema completo se puede mejorar incluyendo mejores componentes, esto es con convertidores analógicos digitales con mayor ancho de banda, convertidores A/D tipo flash más rápidos, y compuertas con menor tiempo de respuesta. Los componentes usados en este proyecto son relativamente económicos y se encuentran en México. En la conversión A/D se pueden usar dos convertidores por canal, lo que aumentaría la resolución a 512 niveles o la frecuencia de muestreo al doble.

---

# APÉNDICE I

## LISTADO DEL PROGRAMA

### (SISTEMA MÍNIMO)

---

#### PROGRAMA

```
// *****
// **** PROG IV (IVFINAL.CPP), 1997 ****
// *****

#include <stdio.h>
#include <time.h>
#include <dos.h>
#include <graphics.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#include <string.h>

// *****
// ***** COMANDOS PARA EL CONTROL DEL HARDWARE *****
// *****

// ***** VARIABLES GLOBALES *****
////////////////////////////////////
int  dw3=0;      // Variable imagen del contenido del puerto de salida 3
int  dw4=0;      // Variable imagen del contenido del puerto de salida 4
int  dw5=0;      // Variable imagen del contenido del puerto de salida 5
////////////////////////////////////
// ***** FIN. VARIABLES GLOBALES *****

// **** EXPLICACION DE LAS SEÑALES DE CONTROL ASOCIADAS A LOS PUERTOS ****
////////////////////////////////////
// DR00 = DA0      : Bit 0 de la entrada de datos a la PC del canal A.
// DR01 = DA1      : Bit 1 de la entrada de datos a la PC del canal A.
// DR02 = DA2      : Bit 2 de la entrada de datos a la PC del canal A.
// DR03 = DA3      : Bit 3 de la entrada de datos a la PC del canal A.
// DR04 = DA4      : Bit 4 de la entrada de datos a la PC del canal A.
// DR05 = DA5      : Bit 5 de la entrada de datos a la PC del canal A.
// DR06 = DA6      : Bit 6 de la entrada de datos a la PC del canal A.
// DR07 = DA7      : Bit 7 de la entrada de datos a la PC del canal A.
////////////////////////////////////
// DR10 = DB0      : Bit 0 de la entrada de datos a la PC del canal B.
// DR11 = DB1      : Bit 1 de la entrada de datos a la PC del canal B.
// DR12 = DB2      : Bit 2 de la entrada de datos a la PC del canal B.
// DR13 = DB3      : Bit 3 de la entrada de datos a la PC del canal B.
// DR14 = DB4      : Bit 4 de la entrada de datos a la PC del canal B.
// DR15 = DB5      : Bit 5 de la entrada de datos a la PC del canal B.
```

```

// DR17 = DB7      : Bit 7 de la entrada de datos a la PC del canal B.
// DR20 = CTATERM  : Si es 1, el cont. gen. de dir. de mem. esta en cta.max.
// DR21 = Expansion.
// DR22 = Expansion.
// DR23 = Expansion.
// DR24 = Expansion.
// DR25 = Expansion.
// DR26 = Expansion.
// DR27 = Expansion.
// DW30 = CFS1     : Activa la fuente senoidal 1.
// DW31 = CFS2     : Activa la fuente senoidal 2.
// DW32 = VAH/-VAL : 0 = no activa div., 1 = activa div., para el canal A.
// DW33 = VBH/-VBL : 0 = no activa div., 1 = activa div., para el canal B.
// DW34 = CRO      : Bit 0 del codigo para selec. el valor del res. prog.
// DW35 = CR1      : Bit 1 del codigo para selec. el valor del res. prog.
// DW36 = CR2      : Bit 2 del codigo para selec. el valor del res. prog.
// DW37 = AX2      : Control para el AGPA, multiplica VA por 2.
// DW40 = AX5      : Control para el AGPA, multiplica VA por 5.
// DW41 = AX10     : Control para el AGPA, multiplica VA por 10.
// DW42 = BX2      : Control para el AGPB, multiplica VB por 2.
// DW43 = BX5      : Control para el AGPB, multiplica VB por 5.
// DW44 = BX10     : Control para el AGPB, multiplica VB por 10.
// DW45 = ESCTM0   : Bit 0 del codigo para las escalas de tiempo.
// DW46 = ESCTM1   : Bit 1 del codigo para las escalas de tiempo.
// DW47 = ESCTM2   : Bit 2 del codigo para las escalas de tiempo.
// DW50 = ESCTM3   : Bit 3 del codigo para las escalas de tiempo.
// DW51 = CKT      : Reloj de transmisiçn.
// DW52 = -CLADR   : Cuando es 0, resetea el cont. gen. de dir. de mem.
// DW53 = T/-M     : Si es 1, modo de muestreo; si es 0, modo de transm.
// DW54 = INVA     : Invierte los datos digitales del canal A.
// DW55 = INVB     : Invierte los datos digitales del canal B.
// DW56 = Expansion.
// DW57 = Expansion.
// **** FIN. EXPLICACION DE LAS SEÑALES DE CONTROL ASOCIADAS A LOS PTO5 ***

// ***** DEFINICIONES *****
// Definicion de las constantes de direcciones de los puertos.
#define DR0 0x300
#define DR1 0x301
#define DR2 0x302
#define DW3 0x303
#define DW4 0x304
#define DW5 0x305
// Fin. Definicion de las constantes de direcciones de los puertos.
// Variables digitales (control de los puertos a nivel de bits).
#define CTATERM      (inportb(DR2) & 1) // BIN "00000001"
#define NEG_CLADR_0  outportb( DW5, dw5 = dw5 & 251 ) // BIN "11111011"
#define NEG_CLADR_1  outportb( DW5, dw5 = dw5 | 4 ) // BIN "00000100"
#define T_NEG_M_0    outportb( DW5, dw5 = dw5 & 247 ) // BIN "11110111"
#define T_NEG_M_1    outportb( DW5, dw5 = dw5 | 8 ) // BIN "00001000"
#define CKT_0        outportb( DW5, dw5 = dw5 & 253 ) // BIN "11111101"
#define CKT_1        outportb( DW5, dw5 = dw5 | 2 ) // BIN "00000010"
#define INVA_0       outportb( DW5, dw5 = dw5 & 239 ) // BIN "11101111"
#define INVA_1       outportb( DW5, dw5 = dw5 | 16 ) // BIN "00010000"
#define ESCTM0_0     outportb( DW4, dw4 = dw4 & 223 ) // BIN "11011111"
#define ESCTM0_1     outportb( DW4, dw4 = dw4 | 32 ) // BIN "00100000"
#define ESCTM1_0     outportb( DW4, dw4 = dw4 & 191 ) // BIN "10111111"
#define ESCTM1_1     outportb( DW4, dw4 = dw4 | 64 ) // BIN "01000000"
#define ESCTM2_0     outportb( DW4, dw4 = dw4 & 127 ) // BIN "01111111"
#define ESCTM2_1     outportb( DW4, dw4 = dw4 | 128 ) // BIN "10000000"
#define ESCTM3_0     outportb( DW5, dw5 = dw5 & 252 ) // BIN "11111110"
#define ESCTM3_1     outportb( DW5, dw5 = dw5 | 1 ) // BIN "00000001"
#define AX02_0       outportb( DW3, dw3 = dw3 & 127 ) // BIN "01111111"
#define AX02_1       outportb( DW3, dw3 = dw3 | 128 ) // BIN "10000000"
#define AX05_0       outportb( DW4, dw4 = dw4 & 254 ) // BIN "11111110"
#define AX05_1       outportb( DW4, dw4 = dw4 | 1 ) // BIN "00000001"
#define AX10_0       outportb( DW4, dw4 = dw4 & 253 ) // BIN "11111101"
#define AX10_1       outportb( DW4, dw4 = dw4 | 2 ) // BIN "00000010"
#define BX02_0       outportb( DW4, dw4 = dw4 & 251 ) // BIN "11111011"

```

```

#define BX02_0      outportb( DW4, dw4 = dw4 & 251 ) // BIN "11111011"
#define BX02_1      outportb( DW4, dw4 = dw4 | 4 ) // BIN "00000100"
#define BX05_0      outportb( DW4, dw4 = dw4 & 247 ) // BIN "11110111"
#define BX05_1      outportb( DW4, dw4 = dw4 | 8 ) // BIN "00001000"
#define BX10_0      outportb( DW4, dw4 = dw4 & 239 ) // BIN "11101111"
#define BX10_1      outportb( DW4, dw4 = dw4 | 16 ) // BIN "00010000"
#define VAH_NEG_VAL_1 outportb( DW3, dw3 = dw3 & 251 ) // BIN "11111011"
#define VAH_NEG_VAL_0 outportb( DW3, dw3 = dw3 | 4 ) // BIN "00000100"
#define VBH_NEG_VBL_1 outportb( DW3, dw3 = dw3 & 247 ) // BIN "11110111"
#define VBH_NEG_VBL_0 outportb( DW3, dw3 = dw3 | 8 ) // BIN "00001000"
// Fin. Variables digitales (control de los puertos a nivel de bits).
// Comandos.
#define GRM_TM000000100 {ESCTM3_0; ESCTM2_0; ESCTM1_0; ESCTM0_0;}
#define GRM_TM000000200 {ESCTM3_0; ESCTM2_0; ESCTM1_0; ESCTM0_1;}
#define GRM_TM000000500 {ESCTM3_0; ESCTM2_0; ESCTM1_1; ESCTM0_0;}
#define GRM_TM000001000 {ESCTM3_0; ESCTM2_0; ESCTM1_1; ESCTM0_1;}
#define GRM_TM000002000 {ESCTM3_0; ESCTM2_1; ESCTM1_0; ESCTM0_0;}
#define GRM_TM000005000 {ESCTM3_0; ESCTM2_1; ESCTM1_0; ESCTM0_1;}
#define GRM_TM000010000 {ESCTM3_0; ESCTM2_1; ESCTM1_1; ESCTM0_0;}
#define GRM_TM000020000 {ESCTM3_0; ESCTM2_1; ESCTM1_1; ESCTM0_1;}
#define GRM_TM000050000 {ESCTM3_1; ESCTM2_0; ESCTM1_0; ESCTM0_0;}
#define GRM_TM000100000 {ESCTM3_1; ESCTM2_0; ESCTM1_0; ESCTM0_1;}
#define GRM_TM000200000 {ESCTM3_1; ESCTM2_0; ESCTM1_1; ESCTM0_0;}
#define GRM_TM000500000 {ESCTM3_1; ESCTM2_0; ESCTM1_1; ESCTM0_1;}
#define GRM_TM001000000 {ESCTM3_1; ESCTM2_1; ESCTM1_0; ESCTM0_0;}
#define GRM_TM002000000 {ESCTM3_1; ESCTM2_1; ESCTM1_0; ESCTM0_1;}
#define GRM_TM005000000 {ESCTM3_1; ESCTM2_1; ESCTM1_1; ESCTM0_0;}
#define GRM_TM010000000 {ESCTM3_1; ESCTM2_1; ESCTM1_1; ESCTM0_1;}
#define AGPA_X001 {AX10_0; AX05_0; AX02_0;}
#define AGPA_X002 {AX10_0; AX05_0; AX02_1;}
#define AGPA_X005 {AX10_0; AX05_1; AX02_0;}
#define AGPA_X010 {AX10_0; AX05_1; AX02_1;}
#define AGPA_X020 {AX10_1; AX05_0; AX02_1;}
#define AGPA_X050 {AX10_1; AX05_1; AX02_0;}
#define AGPA_X100 {AX10_1; AX05_1; AX02_1;}
#define AGPB_X001 {BX10_0; BX05_0; BX02_0;}
#define AGPB_X002 {BX10_0; BX05_0; BX02_1;}
#define AGPB_X005 {BX10_0; BX05_1; BX02_0;}
#define AGPB_X010 {BX10_0; BX05_1; BX02_1;}
#define AGPB_X020 {BX10_1; BX05_0; BX02_1;}
#define AGPB_X050 {BX10_1; BX05_1; BX02_0;}
#define AGPB_X100 {BX10_1; BX05_1; BX02_1;}
// Fin. Comandos.
// ***** FIN. DEFINICIONES *****

// *****
// ***** FIN. COMANDOS PARA EL CONTROL DEL HARDWARE *****
// *****

// *****
// ***** FUNCIONES PARA EL CONTROL DEL HARDWARE *****
// *****

// ***** DECLARACION DE FUNCIONES Y VARIABLES GLOBALES *****
// *****
void cargaimgab(int *ptrimga,int *ptrimgb);//(ST) v. interrup por activ m t
void cargaadqhoriz(int ind); //(ST)esc tiempo
void cargaadqvertva(int ind); //(ST)esc volt a, f. atenuador
// void cargaadqvertvb(int ind); // ---> fucion no desarrollada, esc volt b
// void cargaadqvertcb(int ind); // ---> fucion no desarrollada, esc corr b
// *****
int imga[512]; // Vector en donde se guarda la imagen A de la RAM ext.
int imgb[512]; // Vector en donde se guarda la imagen B de la RAM ext.
// ***** FIN. DECLARACION DE FUNCIONES Y VARIABLES GLOBALES *****

// ***** FUNCIONES *****
// *****
void cargaimgab(int *ptrimga,int *ptrimgb)

```

```

// Periodo maximo de CKT: 4 microseg., 486DX2@50MHz
// Periodo maximo de CKT: 10 microseg., 486DX2@25MHz
int i;
T_NEG_M_0;
NEG_CLADR_0;
NEG_CLADR_1;
ETIQUETA1:
if (CTATERM==0) goto ETIQUETA1;
T_NEG_M_1;
NEG_CLADR_0;
NEG_CLADR_1;
i=0;
ETIQUETA2:
CKT_1;
CKT_0;
if (CTATERM==1) goto ETIQUETA3;
i=i+1;
*(ptrimga+i)=inportb(DR0);
goto ETIQUETA2;
ETIQUETA3;
} // Fin. void cargaimgab(int *ptrimga,int *ptrimgb)
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void cargaadqhoriz(int ind)
{
// El valor de GRM_TMxxxxxxxx es el periodo de muestreo en nanoseg, cada
// muestra se representa en la pantalla como un pixel (horizontalmente).
// Cada division horizontal se compone de 100 pixeles (o muestras).
switch(ind)
{
case 0: GRM_TM000000100; // 100 [ns/muestra] = 10 [us/div]
break;
case 1: GRM_TM000000200; // 200 [ns/muestra] = 20 [us/div]
break;
case 2: GRM_TM000000500; // 500 [ns/muestra] = 50 [us/div]
break;
case 3: GRM_TM000001000; // 1000 [ns/muestra] = 100 [us/div]
break;
case 4: GRM_TM000002000; // 2000 [ns/muestra] = 200 [us/div]
break;
case 5: GRM_TM000005000; // 5000 [ns/muestra] = 500 [us/div]
break;
case 6: GRM_TM000010000; // 10000 [ns/muestra] = 1 [ms/div]
break;
case 7: GRM_TM000020000; // 20000 [ns/muestra] = 2 [ms/div]
break;
case 8: GRM_TM000050000; // 50000 [ns/muestra] = 5 [ms/div]
break;
case 9: GRM_TM000100000; // 100000 [ns/muestra] = 10 [ms/div]
break;
case 10: GRM_TM000200000; // 200000 [ns/muestra] = 20 [ms/div]
break;
case 11: GRM_TM000500000; // 500000 [ns/muestra] = 50 [ms/div]
break;
case 12: GRM_TM001000000; // 1000000 [ns/muestra] = 100 [ms/div]
break;
case 13: GRM_TM002000000; // 2000000 [ns/muestra] = 200 [ms/div]
break;
case 14: GRM_TM005000000; // 5000000 [ns/muestra] = 500 [ms/div]
break;
case 15: GRM_TM010000000; // 10000000 [ns/muestra] = 1 [s/div]
break;
}
} // Fin. cargaadqhoriz(int ind)
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void cargaadqvertva(int ind)
{
switch(ind)
{
case 0: VAH_NEG_VAL_0; // 10 [mV/DIV]
AGPA_X100;
break;
case 1: VAH_NEG_VAL_0; // 20 [mV/DIV]
AGPA_X050;
}
}

```



```

                break;
    case 2: VAH_NEG_VAL_0; // 50 [mV/DIV]
            AGPA_X020;
            break;
    case 3: VAH_NEG_VAL_0; // 100 [mV/DIV]
            AGPA_X010;
            break;
    case 4: VAH_NEG_VAL_0; // 200 [mV/DIV]
            AGPA_X005;
            break;
    case 5: VAH_NEG_VAL_0; // 500 [mV/DIV]
            AGPA_X002;
            break;
    case 6: VAH_NEG_VAL_0; // 1 [V/DIV]
            AGPA_X001;
            break;
    case 7: VAH_NEG_VAL_1; // 2 [V/DIV]
            AGPA_X050;
            break;
    case 8: VAH_NEG_VAL_1; // 5 [V/DIV]
            AGPA_X020;
            break;
    case 9: VAH_NEG_VAL_1; // 10 [V/DIV]
            AGPA_X010;
            break;
    case 10:VAH_NEG_VAL_1; // 20 [V/DIV]
            AGPA_X005;
            break;
    case 11:VAH_NEG_VAL_1; // 50 [V/DIV]
            AGPA_X002;
            break;
    case 12:VAH_NEG_VAL_1; // 100 [V/DIV]
            AGPA_X001;
            break;
    }
} // Fin. cargaadqvertva(int ind)
////////////////////////////////////////////////////////////////////
// ***** FIN. FUNCIONES *****
//
// ***** FIN. FUNCIONES PARA EL CONTROL DEL HARDWARE *****
//
//
// ***** GRAFICOS *****
//
// ***** PROTOTIPOS DE FUNCIONES Y VARIABLES GLOBALES *****
////////////////////////////////////////////////////////////////////
void iniamb_func(void); // (ST)
void finamb_func(void); // (ST)
void boton_func(int edo,int posx,int posy,char *text); // (ST)
void pant_func(int posx,int posy); // (ST)
void display_func(int posx,int posy,char *text); // (ST)
void textfond_func(int posx,int posy,char *text); // (ST)
void textdisp_func(int posx,int posy,char *text); // (ST)
void relieve_func(int posx1,int posy1,int posx2,int posy2); // (ST)
void grupo_func(int posx1,int posy1,int posx2,int posy2); // (ST)
////////////////////////////////////////////////////////////////////
int ambtrab_cfondo;
int boton_color;
int pant_cfondo;
int pant_cejepri;
int pant_cejesec;
int display_cfondo;
int display_ctext;
int textfond_ctext;
int relieve_grueso;
int grupo_color;

```

```

int grupo_grueso;
// **** FIN. PROTOTIPO DE FUNCIONES Y VARIABLES GLOBALES ****

// ***** FUNCIONES *****
// *****
void iniamb_func(void)
{
    int gdriver=9,gmode = 2;
    // Se activa el modo grafico VGA de alta resolucion y 16 colores
    initgraph(&gdriver, &gmode, "");
    // Se activa el color del fondo, se pierde el color numero 0 (negro)
    setbkcolor(ambtrab_cfondo);
    // Redefinicion del color negro, ya que este se perdio
    setpalette(6, BLACK);
} // Fin. iniamb_func(void)
// *****
void finamb_func(void)
{
    closegraph();
} // Fin. finamb_func(void)
// *****
void textfond_func(int posx,int posy,char *text)
{
    int barx1,barx2,bary1,bary2;
    barx1=posx;
    bary1=posy;
    barx2=posx+textwidth(text)-1;
    bary2=posy+textheight(text)-1;
    setfillstyle(1,ambtrab_cfondo);
    bar(barx1,bary1,barx2,bary2);
    setcolor(textfond_ctext);
    outtextxy(posx,posy,text);
} // Fin. textfond_func(int posx,int posy,char *text)
// *****
void textdisp_func(int posx,int posy,char *text)
{
    int barx1,barx2,bary1,bary2;
    barx1=posx;
    bary1=posy;
    barx2=posx+textwidth(text)-1;
    bary2=posy+textheight(text)-1;
    setfillstyle(1,display_cfondo);
    bar(barx1,bary1,barx2,bary2);
    setcolor(display_ctext);
    outtextxy(posx,posy,text);
} // Fin. textdisp_func(int posx,int posy,char *text)
// *****
void display_func(int posx,int posy,char *text)
{
    int barx1,barx2,bary1,bary2;
    int csom;
    int tamx=2,tamy=2;
    posx=posx+3;
    posy=posy+3;
    // Color del sombreado de la barra de texto
    if (ambtrab_cfondo<9 || ambtrab_cfondo>13)
        csom=8;
    else
        csom=ambtrab_cfondo-8;
    // Se establece el tamaño de la barra de texto
    barx1=posx-tamx;
    bary1=posy-tamy;
    barx2=posx+textwidth(text)+tamx-1;
    bary2=posy+textheight(text)+tamy-1;
    // Se visualiza el texto
    setfillstyle(1,display_cfondo);
    bar(barx1,bary1,barx2,bary2);
    setcolor(display_ctext);
    outtextxy(posx,posy,text);
    // Se visualiza el marco
    setcolor(csom);
    line(barx1-1,bary1-1,barx2+1,bary1-1);
}

```

```

line(barx1-1,bary1-1,barx1-1,bary2+1);
setcolor(15);
line(barx1-1,bary2+1,barx2+1,bary2+1);
line(barx2+1,bary1-1,barx2+1,bary2+1);
} // Fin. display_func(int posx,int posy,char *text)
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void relieve_func(int posx1,int posy1,int posx2,int posy2)
{
int csom;
register int i;
// Color del sombreado del relieve
if (ambtrab_cfondo<9 || ambtrab_cfondo>13)
    csom=8;
else
    csom=ambtrab_cfondo-8;
// Se dibuja el relieve
for (i=0; i<=relieve_grueso-1; i=i+1)
{
setcolor(csom);
line(posx1+i,posy2-i,posx2-i,posy2-i);
line(posx2-i,posy1+i,posx2-i,posy2-i);
setcolor(15);
line(posx1+i,posy1+i,posx2-i,posy1+i);
line(posx1+i,posy1+i,posx1+i,posy2-i);
}
} //Fin. relieve_func(int posx1,int posy1,int posx2,int posy2)
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void grupo_func(int posx1,int posy1,int posx2,int posy2)
{
int i;
for (i=0; i<=grupo_grueso-1; i=i+1)
{
setcolor(grupo_color);
rectangle(posx1+i,posy1+i,posx2-i,posy2-i);
}
} //Fin. grupo_func(int posx1,int posy1,int posx2,int posy2)
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void boton_func(int edo,int posx,int posy,char *text)
{
// Simulacion de boton, Edgar R Richter S, 1996
// Estados del boton. 0, normal; 1, senalado; 2, presionado; 3, inactivo
// Requiere una variable global, boton_color, para el color del boton
register int i,j;
int barx1,barx2,bary1,bary2;
int k,l,lc,dlx1,dlx2,dly1,dly2,offset,csup,csom,tamx=0,tamy=0;
char botdat[3][4][16] =
{{{ 15,15,15,15, 15,15,15,15, 15,15, 7, 7, 15,15, 7, 7 },
{ 15,15,15, 8, 15,15, 8, 8, 7, 7, 8, 8, 7, 7, 8, 8 },
{ 15,15, 7, 7, 15,15, 7, 7, 15, 8, 8, 8, 8, 8, 8, 8, 8 },
{ 7, 7, 8, 8, 7, 7, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8 }},
{{ 6, 6, 6, 6, 6,15,15,15, 6,15, 7, 7, 6,15, 7, 7 },
{ 6, 6, 6, 6, 15,15, 8, 6, 7, 8, 8, 6, 7, 8, 8, 6 },
{ 6,15, 7, 7, 6,15, 8, 8, 6, 8, 8, 8, 6, 6, 6, 6 },
{ 7, 8, 8, 6, 8, 8, 8, 6, 8, 8, 8, 6, 6, 6, 6, 6 }},
{{ 6, 6, 6, 6, 6, 8, 8, 8, 6, 8, 8, 8, 6, 8, 7, 7 },
{ 6, 6, 6, 8, 8, 8, 8, 8, 7, 7, 8, 8, 7, 7, 8, 8 },
{ 6, 8, 7, 7, 6, 8, 7, 7, 6, 8, 8, 8, 8, 8, 8, 8 },
{ 7, 7, 8, 8, 7, 7, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8 }}}};
posx=posx+5;
posy=posy+6;
// Color del boton
if (boton_color<1 || boton_color>5)
{
    csup=7;
    csom=8;
}
else
{
    csup=boton_color+8;
    csom=boton_color;
}
for (i=0; i<=2; i=i+1)
    for (j=0; j<=3; j=j+1)

```

```

        for (k=0; k<=15; k=k+1)
            if (botdat[i][j][k]==7)
                botdat[i][j][k]=csup;
            else
                if (botdat[i][j][k]==8)
                    botdat[i][j][k]=csom;
// Se establece el offset del texto cuando esta presionado el boton
if (edo==2)
    offset=1;
else
    offset=0;
// Se establece el tamaño de la barra de texto
barx1=posx-tamx;
bary1=posy-tamy-1;
barx2=posx+textwidth(text)+tamx;
bary2=posy+textheight(text)+tamy-1;
// Se visualiza el texto
setfillstyle(1,csup);
bar(barx1,bary1,barx2,bary2);
if (edo==3)
    {
        setcolor(15);
        edo=0;
    }
else
    setcolor(6);
outtextxy(posx+offset, posy+offset, text);
setcolor(6);
// Se visualizan las esquinas del boton
for (i=0; i<=3; i=i+1)
    for (j=0; j<=3; j=j+1)
        {
            putpixel(barx1+i-4,bary1+j-4,botdat[edo][0][j*4+i]);
            putpixel(barx2+i+1,bary1+j-4,botdat[edo][1][j*4+i]);
            putpixel(barx1+i-4,bary2+j+1,botdat[edo][2][j*4+i]);
            putpixel(barx2+i+1,bary2+j+1,botdat[edo][3][j*4+i]);
        }
// Se visualizan los lados del boton
for (i=0; i<=3; i=i+1)
    {
        setcolor(botdat[edo][0][i*4+3]);
        line(barx1,bary1+i-4,barx2,bary1+i-4);
        setcolor(botdat[edo][1][i*4+1]);
        line(barx2+i+1,bary1,barx2+i+1,bary2);
        setcolor(botdat[edo][3][i*4]);
        line(barx1,bary2+i+1,barx2,bary2+i+1);
        setcolor(botdat[edo][2][i]);
        line(barx1+i-4,bary1,barx1+i-4,bary2);
    }
// Se visualiza el marco del boton
setcolor( 6);
line(barx1-4,bary1-5,barx2+4,bary1-5);
line(barx1-4,bary2+5,barx2+4,bary2+5);
line(barx2+5,bary1-4,barx2+5,bary2+4);
line(barx1-5,bary1-4,barx1-5,bary2+4);
// Se visualiza la línea punteada de selección
if (edo == 1 || edo == 2)
    {
        j=0;
        k=0;
        l=strlen(text);
        for (i=0; i<=l-2; i=i+1 )
            if (*(text+i)==32 && *(text+i+1)!=32 && j==0 )
                j=i+1;
        for (i=l-1; i>=1; i=i-1 )
            if (*(text+i)==32 && *(text+i-1)!=32 && k==0 )
                k=l-i;
        dlx1=barx1+j*8+tamx-1;
        dlx2=barx2-k*8-tamx;
        dly1=bary1+tamy;
        dly2=bary2-tamy;
        for (i=dlx1; i<=dlx2; i=i+2)
            {

```

```

        putpixel(i+offset,dly1-1+offset,8);
        putpixel(i+offset,dly2+1+offset,8);
    }
    for (i=dly1; i<=dly2; i=i+2)
    {
        putpixel(dlx1-1+offset,i+offset,8);
        putpixel(dlx2+1+offset,i+offset,8);
    }
} // Fin. boton_func(int edo,int posx,int posy,char *text)
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void pant_func(int posx,int posy)
{
    // Pantalla, Edgar R Richter S, 1996
    // Requiere las siguientes variables globales:
    // colorfondo, pant_cejepri, pant_cejeseq, pant_cfondo
    register int i;
    int margen=2,marco=2;
    int barx1,barx2,bary1,bary2;
    int cuadx1,cuady1;
    int csom;
    posx=posx+2;
    posy=posy+2;
    // Color del sombreado de la pantalla
    if (ambtrab_cfondo<9 || ambtrab_cfondo>13)
        csom=8;
    else
        csom=ambtrab_cfondo-8;
    // Se establece el tamaño de la barra del fondo de la pantalla
    barx1=posx;
    bary1=posy;
    barx2=posx+258+2*margen;
    bary2=posy+258+2*margen;
    // Se establece el inicio de la cuadrícula;
    cuadx1=posx+margen;
    cuady1=posy+margen;
    // Fondo de la pantalla
    setfillstyle(1,pant_cfondo);
    bar(barx1,bary1,barx2,bary2);
    // Se dibuja el marco de la pantalla
    for (i=1; i<=marco; i=i+1)
    {
        setcolor(csom);
        line(barx1-i,bary1-i,barx2+i,bary1-i);
        line(barx1-i,bary1-i,barx1-i,bary2+i);
        setcolor(15);
        line(barx1-i,bary2+i,barx2+i,bary2+i);
        line(barx2+i,bary1-i,barx2+i,bary2+i);
    }
    // Se dibuja la cuadrícula
    setcolor(pant_cejeseq);
    line(cuadx1,cuady1+ 4,cuadx1+258,cuady1+ 4);
    line(cuadx1,cuady1+ 54,cuadx1+258,cuady1+ 54);
    line(cuadx1,cuady1+104,cuadx1+258,cuady1+104);
    line(cuadx1,cuady1+154,cuadx1+258,cuady1+154);
    line(cuadx1,cuady1+204,cuadx1+258,cuady1+204);
    line(cuadx1,cuady1+254,cuadx1+258,cuady1+254);
    line(cuadx1+ 4,cuady1,cuadx1+ 4,cuady1+258);
    line(cuadx1+ 54,cuady1,cuadx1+ 54,cuady1+258);
    line(cuadx1+ 79,cuady1,cuadx1+ 79,cuady1+258);
    line(cuadx1+104,cuady1,cuadx1+104,cuady1+258);
    line(cuadx1+154,cuady1,cuadx1+154,cuady1+258);
    line(cuadx1+179,cuady1,cuadx1+179,cuady1+258);
    line(cuadx1+204,cuady1,cuadx1+204,cuady1+258);
    line(cuadx1+254,cuady1,cuadx1+254,cuady1+258);
    setcolor(pant_cejepri);
    line(cuadx1,cuady1+ 29,cuadx1+258,cuady1+ 29);
    line(cuadx1,cuady1+ 79,cuadx1+258,cuady1+ 79);
    line(cuadx1,cuady1+129,cuadx1+258,cuady1+129);
    line(cuadx1,cuady1+179,cuadx1+258,cuady1+179);
    line(cuadx1,cuady1+229,cuadx1+258,cuady1+229);
    line(cuadx1+ 29,cuady1,cuadx1+ 29,cuady1+258);
    line(cuadx1+129,cuady1,cuadx1+129,cuady1+258);

```

```

line(cuadx1+229,cuady1,cuadx1+229,cuady1+258);
} // Fin. pant_func(int posx,int posy)
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// ***** FIN. FUNCIONES *****

// *****
// ***** FIN. GRAFICOS *****
// *****

// *****
// ***** TRANSFORMACION DE UNIDADES *****
// *****

// ***** PROTOTIPOS DE FUNCIONES Y VARIABLES GLOBALES *****
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void prefijosiu(double valor,char *ptrcadena); // (ST)
double tseg_func(int indh,double tpix); // (ST)
double vertvolts func(int indv,double vertpix); // (ST)
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// ***** FIN. PROTOTIPOS DE FUNCIONES Y VARIABLES GLOBALES *****

// ***** FUNCIONES *****
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void prefijosiu(double valor,char *ptrcadena)
{
int pot;
static char buffer[10],poten[4],salida[7];
sprintf(buffer,"%+1.2e\n",valor);
buffer[9]='\0';
poten[0]=' ';
poten[1]=buffer[6];
poten[2]=buffer[7];
poten[3]=buffer[8];
poten[4]='\0';
salida[0]=buffer[0];
salida[1]=buffer[1];
salida[6]='\0';
pot=atoi(poten);
switch (pot)
{
case -12:salida[2]='.';
salida[3]=buffer[3];
salida[4]=buffer[4];
salida[5]='p';
break;
case -11:salida[2]=buffer[3];
salida[3]='.';
salida[4]=buffer[4];
salida[5]='p';
break;
case -10:salida[2]=buffer[3];
salida[3]=buffer[4];
salida[4]=' ';
salida[5]='p';
break;
case -9: salida[2]='.';
salida[3]=buffer[3];
salida[4]=buffer[4];
salida[5]='\n';
break;
case -8: salida[2]=buffer[3];
salida[3]='.';
salida[4]=buffer[4];
salida[5]='\n';
break;
case -7: salida[2]=buffer[3];
salida[3]=buffer[4];
salida[4]=' ';
salida[5]='\n';
}
}

```

```
break;
case -6: salida[2]='.';
        salida[3]=buffer[3];
        salida[4]=buffer[4];
        salida[5]='u';
        break;
case -5: salida[2]=buffer[3];
        salida[3]='.';
        salida[4]=buffer[4];
        salida[5]='u';
        break;
case -4: salida[2]=buffer[3];
        salida[3]=buffer[4];
        salida[4]=' ';
        salida[5]='u';
        break;
case -3: salida[2]='.';
        salida[3]=buffer[3];
        salida[4]=buffer[4];
        salida[5]='m';
        break;
case -2: salida[2]=buffer[3];
        salida[3]='.';
        salida[4]=buffer[4];
        salida[5]='m';
        break;
case -1: salida[2]=buffer[3];
        salida[3]=buffer[4];
        salida[4]=' ';
        salida[5]='m';
        break;
case 0:  salida[2]='.';
        salida[3]=buffer[3];
        salida[4]=buffer[4];
        salida[5]=' ';
        break;
case 1:  salida[2]=buffer[3];
        salida[3]='.';
        salida[4]=buffer[4];
        salida[5]=' ';
        break;
case 2:  salida[2]=buffer[3];
        salida[3]=buffer[4];
        salida[4]=' ';
        salida[5]=' ';
        break;
case 3:  salida[2]='.';
        salida[3]=buffer[3];
        salida[4]=buffer[4];
        salida[5]='k';
        break;
case 4:  salida[2]=buffer[3];
        salida[3]='.';
        salida[4]=buffer[4];
        salida[5]='k';
        break;
case 5:  salida[2]=buffer[3];
        salida[3]=buffer[4];
        salida[4]=' ';
        salida[5]='k';
        break;
case 6:  salida[2]='.';
        salida[3]=buffer[3];
        salida[4]=buffer[4];
        salida[5]='M';
        break;
case 7:  salida[2]=buffer[3];
        salida[3]='.';
        salida[4]=buffer[4];
        salida[5]='M';
        break;
case 8:  salida[2]=buffer[3];
        salida[3]=buffer[4];
```

```

        salida[4]=' ';
        salida[5]='M';
        break;
case 9:  salida[2]='.';
        salida[3]=buffer[3];
        salida[4]=buffer[4];
        salida[5]='G';
        break;
case 10: salida[2]=buffer[3];
        salida[3]='.';
        salida[4]=buffer[4];
        salida[5]='G';
        break;
case 11: salida[2]=buffer[3];
        salida[3]=buffer[4];
        salida[4]=' ';
        salida[5]='G';
        break;
    }
    if (pot>=12 || pot<=-12)
    {
        salida[1]='-';
        salida[2]='-';
        salida[3]='-';
        salida[4]='-';
        salida[5]='-';
    }
*(ptrcadena+0)=salida[0];
*(ptrcadena+1)=salida[1];
*(ptrcadena+2)=salida[2];
*(ptrcadena+3)=salida[3];
*(ptrcadena+4)=salida[4];
*(ptrcadena+5)=salida[5];
*(ptrcadena+6)=salida[6];
} // Fin. prefijosiu(double valor,char *ptrcadena)
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
double tseg_func(int indh,double tpix)
{
    double tseg;
    switch(indh)
    {
        case 0: tseg = 100e-09      * tpix;
                break;
        case 1: tseg = 200e-09      * tpix;
                break;
        case 2: tseg = 500e-09      * tpix;
                break;
        case 3: tseg = 1000e-09     * tpix;
                break;
        case 4: tseg = 2000e-09     * tpix;
                break;
        case 5: tseg = 5000e-09     * tpix;
                break;
        case 6: tseg = 10000e-09    * tpix;
                break;
        case 7: tseg = 20000e-09    * tpix;
                break;
        case 8: tseg = 50000e-09    * tpix;
                break;
        case 9: tseg = 100000e-09   * tpix;
                break;
        case 10: tseg = 200000e-09  * tpix;
                break;
        case 11: tseg = 500000e-09  * tpix;
                break;
        case 12: tseg = 1000000e-09 * tpix;
                break;
        case 13: tseg = 2000000e-09 * tpix;
                break;
        case 14: tseg = 5000000e-09 * tpix;
                break;
        case 15: tseg = 10000000e-09 * tpix;
                break;
    }
}

```



```

    }
    return (tseg);
} // Fin. tseg_func(int indh,double tpix)
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
double vertvolts_func(int indv,double vertpix)
{
    double vertvolts;
    switch(indv)
    {
        case 0: vertvolts = 0.0002 * vertpix;
                break;
        case 1: vertvolts = 0.0004 * vertpix;
                break;
        case 2: vertvolts = 0.001 * vertpix;
                break;
        case 3: vertvolts = 0.002 * vertpix;
                break;
        case 4: vertvolts = 0.004 * vertpix;
                break;
        case 5: vertvolts = 0.01 * vertpix;
                break;
        case 6: vertvolts = 0.02 * vertpix;
                break;
        case 7: vertvolts = 0.04 * vertpix;
                break;
        case 8: vertvolts = 0.1 * vertpix;
                break;
        case 9: vertvolts = 0.2 * vertpix;
                break;
        case 10: vertvolts = 0.4 * vertpix;
                break;
        case 11: vertvolts = 1 * vertpix;
                break;
        case 12: vertvolts = 2 * vertpix;
                break;
    }
    return(vertvolts);
} // Fin. vertvolts_func(int indv,double vertpix)
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// ***** FIN. FUNCIONES *****

// ***** FIN. TRANSFORMACION DE UNIDADES *****

// ***** PROCESAMIENTO DE LOS VECTORES IMAGEN DE RAM *****

// ***** PROTOTIPOS DE FUNCIONES Y VARIABLES GLOBALES *****
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
int indiniimga(int disparo_ind); // (ST) falta incluir disparo
void visimga(int ini); // (ST) visualiza imagen A
void varvoltimga(int ini,int indh,int indv); // (ST) falta atenuador
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// int indinimgb(int disparo_ind); // ---> nd
// void visimgb(int ini); // ---> nd
// void varvoltimgb(int ini,int indh,int indv); // ---> nd
// void varcorrimgb(int ini,int indh,int indv); // ---> nd
// void varpotimgab(int ini,int indh,int indva, int indvb); // ---> nd
// void varimpimgab(int ini,int indh,int indva, int indvb); // ---> nd
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
int visimga c; // Color de visualizacion de la senal A
int imgaant[512]; // Vect auxiliar a imga.
int colaant[512]; // Vect que tiene el color antes de poner imga
int imgbant[512]; // Vect auxiliar a imgb.
int colbant[512]; // Vect que tiene el color antes de poner imgb
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// ***** FIN. PROTOTIPOS DE FUNCIONES Y VARIABLES GLOBALES *****

```

```

// ***** FUNCIONES *****
////////////////////////////////////
int indiniimga(int disparo_ind)
{
    register int i;
    int ini=0;
    int horizflancopix[512];
    int auxmax=0,auxmin=0,vertmedpix,nflancos=-1,vertmaxpix=0,vertminpix=999;
    int noruidol=4; // ventana antiruido en para el flanco de subida o bajada
    for (i=1; i<=255; i=i+1)
    {
        if (imga[i]>vertmaxpix)
            vertmaxpix=imga[i];
        if (imga[i]<vertminpix)
            vertminpix=imga[i];
    }
    vertmedpix=(vertmaxpix+vertminpix)/2;
    for (i=1; i<=255; i=i+1)
    {
        if (imga[i]>vertmedpix+noruidol && auxmax==0)
        {
            auxmax=1;
            auxmin=0;
            nflancos=nflancos+1;
            horizflancopix[nflancos]=i;
        }
        if (imga[i]<vertmedpix-noruidol && auxmin==0)
        {
            auxmax=0;
            auxmin=1;
            nflancos=nflancos+1;
            horizflancopix[nflancos]=i;
        }
    }
    switch(disparo_ind)
    {
        case 0: ini=horizflancopix[nflancos-1+auxmax];
                break;
        case 1: ini=horizflancopix[nflancos-1+auxmin];
                break;
    }
    return (ini);
} // Fin. int indiniimga()
////////////////////////////////////
void visimga(int ini)
{
    register int i;
    int idx,iini,dx=18,dy=57;
    for (i=0; i<=255; i=i+1)
    {
        idx=i+dx;
        iini=i+ini;
        putpixel(idx, 256-imgaant[i]+ dy, colaant[i]);
        colaant[i]=getpixel(idx, 256-imga[iini]+ dy);
        putpixel(idx, 256-imga[iini]+ dy, visimga_c);
        imgaant[i]=imga[iini];
    }
} // Fin. void visimga(int ini)
////////////////////////////////////
void varvoltimga(int ini,int indh,int indv)
{
    // imga debe representar valores de voltajes del canal A. Analiza el
    // vector imga y a partir de este y las escalas calcula: el periodo,
    // la frecuencia, el val max y el val min. El resultado es visualizado.
    register int i;
    int horizflancopix[512];
    int sum=0,auxmax=0,auxmin=0,vertmedpix,nflancos=-1;
    double vertmaxpix=0,vertminpix=999,semitpix,tpix;
    double tseg,fhz,vertminvolts,vertmaxvolts;
    char cadena[7],cadvisual[8];
    int noruidol=4; // ventana antiruido en para el flanco de subida o bajada
    int noruido2=4; // periodo minimo [pix], si es menor se considera ruido
}

```

```

for (i=ini; i<=(ini+255); i=i+1)
{
    if (imga[i]>vertmaxpix)
        vertmaxpix=imga[i];
    if (imga[i]<vertminpix)
        vertminpix=imga[i];
}
vertmedpix=(vertmaxpix+vertminpix)/2;
for (i=1; i<=511; i=i+1)
{
    if (imga[i]>vertmedpix+noruido1 && auxmax==0)
    {
        auxmax=1;
        auxmin=0;
        nflancos=nflancos+1;
        horizflancopix[nflancos]=i;
    }
    if (imga[i]<vertmedpix-noruido1 && auxmin==0)
    {
        auxmax=0;
        auxmin=1;
        nflancos=nflancos+1;
        horizflancopix[nflancos]=i;
    }
}
sum=0;
for (i=2; i<=nflancos-1; i=i+1)
    sum=sum+abs(horizflancopix[i+1]-horizflancopix[i]);
if (nflancos>=3)
    semitpix=double(sum)/(double(nflancos)-2); // semiperiodo [pixeles]
if (vertmaxpix>=1 && vertmaxpix<=254)
    vertmaxpix=vertmaxpix-128; // voltaje max [pixeles]
else
    vertmaxpix=1e99; // voltaje max indefinido
if (vertminpix>=1 && vertminpix<=254)
    vertminpix=vertminpix-128; // voltaje min [pixeles]
else
    vertminpix=1e99; // voltaje min indefinido
if (semitpix>=noruido2 && nflancos>=2 &&
    vertmaxpix<=254 && vertminpix<=254)
    tpix=2*semitpix; // periodo [pixeles]
else
    tpix=1e99; // periodo indefinido
tseg=tseg_func(indh,tpix); // periodo [s]
if (tseg<=50)
    fhz=1/tseg; // frecuencia [Hz]
else
    fhz=1e99; // frecuencia indefinida
vertmaxvolts=vertvolts_func(indv,vertmaxpix); // voltaje max [V]
vertminvolts=vertvolts_func(indv,vertminpix); // voltaje min [V]
prefijosiu(tseg,cadena);
if (cadena[5]!='-')
{
    for (i=0; i<=5; i=i+1)
        cadvisual[i]=cadena[i];
    cadvisual[6]='s';
    cadvisual[7]=' ';
}
else
{
    for (i=0; i<=7; i=i+1)
        cadvisual[i]='-';
}
cadvisual[8]='\0';
textdisp_func(213,362,cadvisual);
prefijosiu(fhz,cadena);
if (cadena[5]!='-')
{
    for (i=0; i<=5; i=i+1)
        cadvisual[i]=cadena[i];
    cadvisual[6]='H';
    cadvisual[7]='z';
}

```

```

else
{
    for (i=0;i<=7;i=i+1)
        cadvisual[i]='-';
}
cadvisual[8]='\0';
textdisp_func(213,394,cadvisual);
prefijosiu(vertmaxvolts,cadena);
if (cadena[5]!='-')
{
    for (i=0;i<=5;i=i+1)
        cadvisual[i]=cadena[i];
    cadvisual[6]='V';
    cadvisual[7]=' ';
}
else
{
    for (i=0;i<=7;i=i+1)
        cadvisual[i]='-';
}
cadvisual[8]='\0';
textdisp_func(213,426,cadvisual);
prefijosiu(vertminvolts,cadena);
if (cadena[5]!='-')
{
    for (i=0;i<=5;i=i+1)
        cadvisual[i]=cadena[i];
    cadvisual[6]='V';
    cadvisual[7]=' ';
}
else
{
    for (i=0;i<=7;i=i+1)
        cadvisual[i]='-';
}
cadvisual[8]='\0';
textdisp_func(213,458,cadvisual);
} // Fin. varvoltimga(int indh,int indv)
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// ***** FIN. FUNCIONES *****
//
// ***** FIN PROCESAMIENTO DE LOS VECTORES IMAGEN DE RAM *****
//
//
// ***** CONTROL DEL MENU *****
//
// ***** PROTOTIPOS DE FUNCIONES Y VARIABLES GLOBALES *****
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
int  menbot_func(void); // (ST)
int  menbot_def;
int  menbot_nop;
int  menbot_posx[25];
int  menbot_posy[25];
char *menbot_text[25][10];
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// ***** FIN. PROTOTIPO DE FUNCIONES Y VARIABLES GLOBALES *****
//
// ***** FUNCIONES *****
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
int menbot_func(void)
{
    // Menu de botones. Edgar R Richter S, 1996
    char mousec[7][7]=
        {{ 6, 6, 6, 6, 6, 6, 6 },
         { 6,15,15,15,15,15, 6 },
         { 6,15,15,15, 6, 6, 6 },

```

```

    { 6,15,15,15,15, 6, 6 },
    { 6,15, 6,15,15,15, 6 },
    { 6,15, 6, 6,15,15, 6 },
    { 6, 6, 6, 6, 6, 6 } });
register int i;
int j,k,keyb,retardo,salida,cursora,mousex,mousey,mouseb;
static int mousebp,mousexa,mouseya,cursor,keyba;
static int mouseca[7][7];
static int aux1,aux2;
union REGS inreg,outreg;
inreg.x.ax=3;
if (menbot_def==1)
{
    cursor=1;
    keyba=0;
    for (i=1; i<=menbot_nop; i=i+1)
        boton_func (0,menbot_posx[i],menbot_posy[i],menbot_text[i][i]);
    boton_func(1,menbot_posx[cursor],menbot_posy[cursor],
        menbot_text[cursor][1]);
    cursora=cursor;
}
if (menbot_def==0)
{
    keyb = 0;
    cursora=cursor;
    int86(0x33,&inreg,&outreg);
    mousex=outreg.x.cx;
    mousey=outreg.x.dx*2.4;
    if (aux1==0)
    {
        mousexa=mousex;
        mouseya=mousey;
        for (i=0; i<=6; i=i+1)
            for (j=0; j<=6; j=j+1)
                mouseca[i][j]=getpixel(mousex+i,mousey+j);
        aux1=1;
    }
    if (aux2==0 && (mousexa!=mousex || mouseya!=mousey))
        aux2=1;
    if (mousey>479)
        mousey=479;
    mouseb=outreg.x.bx;
    if (mousebp==1)
        for (i=0; i<=6; i=i+1)
            for (j=0; j<=6; j=j+1)
                mouseca[i][j]=getpixel(mousexa+i,mouseya+j);
    if (aux2==1)
    {
        if (mousexa>=mousex && mouseya>=mousey) // -x, -y
            for (i=0; i<=6; i=i+1)
                for (j=0; j<=6; j=j+1)
                {
                    putpixel(mousexa+i,mouseya+j,mouseca[i][j]);
                    mouseca[i][j]=getpixel(mousex+i,mousey+j);
                    putpixel(mousex+i,mousey+j,mouseca[i][j]);
                }
        if (mousexa<mousex && mouseya>mousey) // +x, -y
            for (i=6; i>=0; i=i-1)
                for (j=0; j<=6; j=j+1)
                {
                    putpixel(mousexa+i,mouseya+j,mouseca[i][j]);
                    mouseca[i][j]=getpixel(mousex+i,mousey+j);
                    putpixel(mousex+i,mousey+j,mouseca[i][j]);
                }
        if (mousexa<=mousex && mouseya<=mousey) // +x, +y
            for (i=6; i>=0; i=i-1)
                for (j=6; j>=0; j=j-1)
                {
                    putpixel(mousexa+i,mouseya+j,mouseca[i][j]);
                    mouseca[i][j]=getpixel(mousex+i,mousey+j);
                    putpixel(mousex+i,mousey+j,mouseca[i][j]);
                }
        if (mousexa>mousex && mouseya<mousey) // -x, +y

```

```

        for (i=0; i<=6; i=i+1)
            for (j=6; j>=0; j=j-1)
                {
                    putpixel(mousex+i,mouseya+j,mouseca[i][j]);
                    mouseca[i][j]=getpixel(mousex+i,mousey+j);
                    putpixel(mousex+i,mousey+j,mousec[i][j]);
                }
    }
    mousexa=mousex;
    mouseya=mousey;
    if (kbhit())
        keyb=getch();
    mousebp=0;
    if (mouseb==1)
        {
            for (i=1; i<=menbot_nop; i=i+1)
                if (menbot_posx[i]<=mousex &&
                    mousex<=menbot_posx[i]+textwidth(menbot_text[i][1])+9 &&
                    menbot_posy[i]<=mousey &&
                    mousey<=menbot_posy[i]+textheight(menbot_text[i][1])+9 )
                    {
                        mousebp=1;
                        cursor=i;
                        boton_func(0,menbot_posx[cursor],menbot_posy[cursor],
                            menbot_text[cursor][1]);
                        for (j=0; j<=6; j=j+1)
                            for (k=0; k<=6; k=k+1)
                                putpixel(mousexa+j,mouseya+k,mouseca[j][k]);
                        keyb=13;
                        delay(50);
                    }
        }
    if (keyb == 0)
        {
            if (keyba==13)
                boton_func(1,menbot_posx[cursor],menbot_posy[cursor],
                    menbot_text[cursor][1]);
            keyba=keyb;
        }
    if (keyb != 0)
        {
            if (keyba==13 && keyb==13)
                retardo=0;
            else
                retardo=70;
            keyba=keyb;
            if (keyb == 9 || keyb == 77 || keyb == 80)
                {
                    cursor=cursor+1;
                    if (cursor>menbot_nop)
                        cursor=1;
                    boton_func(0,menbot_posx[cursor],menbot_posy[cursor],
                        menbot_text[cursor][1]);
                    boton_func(1,menbot_posx[cursor],menbot_posy[cursor],
                        menbot_text[cursor][1]);
                }
            if (keyb == 72 || keyb == 75)
                {
                    cursor=cursor-1;
                    if (cursor<1)
                        cursor=menbot_nop;
                    boton_func(0,menbot_posx[cursor],menbot_posy[cursor],
                        menbot_text[cursor][1]);
                    boton_func(1,menbot_posx[cursor],menbot_posy[cursor],
                        menbot_text[cursor][1]);
                }
            if (keyb == 13)
                {
                    boton_func(2,menbot_posx[cursor],menbot_posy[cursor],
                        menbot_text[cursor][1]);
                    delay(retardo);
                    boton_func(1,menbot_posx[cursor],menbot_posy[cursor],
                        menbot_text[cursor][1]);
                }
        }

```

```

        salida=cursor;
    }
}
if (salida<0 || salida>menbot_nop)
    salida=0;
return(salida);
} // Fin. menbot_func(void)
////////////////////////////////////
// ***** FIN. FUNCIONES *****
// *****
// ***** FIN. CONTROL DEL MENU *****
// *****

// *****
// ***** VENTANAS *****
// *****

// ***** PROTOTIPOS DE FUNCIONES *****
void ventfija_func(void); // (ST)
void desactivabotones_func(void); // (ST)
void ventmenprin_func(void); // (ST)
void ventmenvn_func(void); // default // (ST)
////////////////////////////////////
// void ventmennc_func(void); // nd
// void ventmenvv_func(void); // nd
// void ventmenin_func(void); // nd
// void ventmenvc_func(void); // nd
// ***** FIN. PROTOTIPOS DE FUNCIONES *****

// ***** FUNCIONES *****
////////////////////////////////////
void ventfija_func(void)
{
    // Ventana fija, sin opciones
    // Tambien se presentan los valores iniciales en los displays
    pant_func(13,52); // Pantalla
    relieve_func ( 8, 47,284,323); // Pantalla
    relieve_func ( 8, 6, 84, 32); // Limpiar
    relieve_func ( 85, 6,145, 32); // Abrir
    relieve_func (146, 6,238, 32); // Guardar A
    relieve_func (239, 6,331, 32); // Guardar B
    relieve_func (332, 6,392, 32); // Ayuda
    relieve_func (393, 6,453, 32); // Salir
    textfond_func(401,147,"Variables electricas"); // Var elect
    display_func (401,157,"A:VOLTAJE, B:NO USAR "); // Var elect
    relieve_func (396,131,630,174); // Var elect
    relieve_func (396,385,505,428); // Iniciar
    relieve_func (396,429,505,472); // Alto
    textfond_func( 13,349,"F.P. "); // f.p
    display_func ( 13,359,"-----"); // f.p
    relieve_func ( 8,345, 87, 376); // f.p
    textfond_func( 13,381,"Aparente "); // Aparente
    display_func ( 13,391,"-----"); // Aparente
    relieve_func ( 8,377, 87,408); // Aparente
    textfond_func( 13,413,"Real "); // Real
    display_func ( 13,423,"-----"); // Real
    relieve_func ( 8,409, 87,440); // Real
    textfond_func( 13,445,"Reactiva "); // Reactiva
    display_func ( 13,455,"-----"); // Reactiva
    relieve_func ( 8,441, 87,472); // Reactiva
    textfond_func(111,349,"Real "); // Real
    display_func (111,359,"-----"); // Real
    relieve_func (106,345,185,376); // Real
    textfond_func(111,381,"Imag "); // Imag
    display_func (111,391,"-----"); // Imag
    relieve_func (106,377,185,408); // Imag
    textfond_func(111,413,"Tipo "); // Tipo
}

```

```

display_func (111,423,"-----"); // Tipo
relieve_func (106,409,185,440); // Tipo
textfond_func(111,445,"Valor "); // Valor
display_func (111,455,"-----"); // Valor
relieve_func (106,441,185,472); // Valor
textfond_func(210,349,"Periodo "); // Periodo A
display_func (210,359,"-----"); // Periodo A
relieve_func (205,345,284,376); // Periodo A
textfond_func(210,381,"Frec. "); // Frec A
display_func (210,391,"-----"); // Frec A
relieve_func (205,377,284,408); // Frec A
textfond_func(210,413,"Valor max"); // Valor max A
display_func (210,423,"-----"); // Valor max A
relieve_func (205,409,284,440); // Valor max A
textfond_func(210,445,"Valor min"); // Valor min A
display_func (210,455,"-----"); // Valor min A
relieve_func (205,441,284,472); // Valor min A
textfond_func(306,257,"Canal A "); // Atenuador A
display_func (306,267,"Activo "); // Atenuador A
relieve_func (301,253,380,284); // Atenuador B
textfond_func(306,289,"Canal B "); // Atenuador B
display_func (306,299,"-----"); // Atenuador B
relieve_func (301,285,380,316); // Atenuador B
textfond_func(306,349,"Periodo "); // Periodo B
display_func (306,359,"-----"); // Periodo B
relieve_func (301,345,380,376); // Periodo B
textfond_func(306,381,"Frec. "); // Frec B
display_func (306,391,"-----"); // Frec B
relieve_func (301,377,380,408); // Frec B
textfond_func(306,413,"Valor max"); // Valor max B
display_func (306,423,"-----"); // Valor max B
relieve_func (301,409,380,440); // Valor max B
textfond_func(306,445,"Valor min"); // Valor min B
display_func (306,455,"-----"); // Valor min B
relieve_func (301,441,380,472); // Valor min B
display_func (306, 57,"-----"); // Archivo 1 nombre
relieve_func (301, 53,412, 74); // Archivo 1 nombre
display_func (306, 79,"-----"); // Archivo 2 nombre
relieve_func (301, 75,412, 96); // Archivo 2 nombre
display_func (306,135,"-----"); // Adq nombre A
relieve_func (301,131,380,152); // Adq nombre A
display_func (306,157,"-----"); // Adq nombre B
relieve_func (301,153,380,174); // Adq nombre B
display_func (439, 57,"-----"); // Archivo 1 horiz
relieve_func (434, 53,521, 74); // Archivo 1 horiz
display_func (439, 79,"-----"); // Archivo 2 horiz
relieve_func (434, 75,521, 96); // Archivo 2 horiz
display_func (548, 57,"-----"); // Archivo 1 vert
relieve_func (543, 53,630, 74); // Archivo 1 vert
display_func (548, 79,"-----"); // Archivo 2 vert
relieve_func (543, 75,630, 96); // Archivo 2 vert
textfond_func(401,269,"Horiz "); // Aqd horiz
display_func (401,279," 10us/div"); // Adq horiz
relieve_func (396,253,505,296); // Adq horiz
textfond_func(401,313,"Vert A "); // Adq vert A
display_func (401,323,"100 V/div"); // Adq vert A
relieve_func (396,297,505,340); // Adq vert A
textfond_func(401,357,"Vert B "); // Adq vert B
display_func (401,367,"-----"); // Adq vert B
relieve_func (396,341,505,384); // Adq vert B
textfond_func(401,191,"Disparo "); // Disparo
display_func (401,201,"Flanco(-)"); // Disparo
relieve_func (396,175,505,218); // Disparo
textfond_func(526,269,"Esc horiz "); // Ref esc horiz
display_func (526,279," 10us/div"); // Ref esc horiz
relieve_func (521,253,630,296); // Ref esc horiz
textfond_func(526,313,"Esc Vert "); // Ref esc vert
display_func (526,323,"100 V/div"); // Ref esc vert
relieve_func (521,297,630,340); // Ref esc vert
textfond_func(526,357,"Horiz R0 "); // Ref horiz R0
display_func (526,367,"-0.00 s "); // Ref horiz R0
relieve_func (521,341,630,384); // Ref horiz R0
textfond_func(526,401,"Horiz R1 "); // Ref horiz R1

```



```

display_func (526,411,"+0.00 s "); // Ref horiz R1
relieve_func (521,385,630,428); // Ref horiz R1
textfond_func(526,445,"Vert R "); // Ref vert R
display_func (526,455,"+0.00 V "); // Ref vert R
relieve_func (521,429,630,472); // Ref vert R
grupo_func (393,244,508,475); // Grupo adq
textfond_func(403,242," Adq "); // Grupo adq
grupo_func (393,122,633,221); // Grupo configura adq
textfond_func(403,120," Configura adq "); // Grupo configura adq
grupo_func (518,244,633,475); // Grupo referencias
textfond_func(524,242," Referencias "); // Grupo referencias
grupo_func (202,336,287,475); // Grupo canal A
textfond_func(210,334," Canal A "); // Grupo canal A
grupo_func (298,336,383,475); // Grupo canal B
textfond_func(306,334," Canal B "); // Grupo canal B
grupo_func ( 5,336, 90,475); // Grupo potencia
textfond_func(11,334," Potencia"); // Grupo potencia
grupo_func (103,336,188,475); // Grupo impedancia
textfond_func(106,334,"Impedancia"); // Grupo impedancia
grupo_func (298,244,383,326); // Grupo atenuador
textfond_func(301,242," Atenuador"); // Grupo adq
grupo_func (298,122,383,177); // Grupo adq nom
textfond_func(308,120," Adq "); // Grupo adq nom
grupo_func (298, 44,633, 99); // Grupo archivo
textfond_func(308, 42," Archivo "); // Grupo archivo nom
textfond_func(442, 42," Horiz "); // Grupo archivo horiz
textfond_func(550, 42," Vert "); // Grupo archivo vert
grupo_func ( 5, 44,287,326); // Grupo pantalla
grupo_func ( 5, 3,633, 35); // Grupo menu sup
} // Fin. ventfija_func(void)
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void desactivabotones_func(void)
{
boton_func(3, 13, 10,"Limpiar"); // Limpiar
boton_func(3, 90, 10,"Abrir"); // Abrir
boton_func(3,151, 10,"Guardar A"); // Guardar A
boton_func(3,244, 10,"Guardar B"); // Guardar B
boton_func(3,337, 10,"Ayuda"); // Ayuda
boton_func(3,398, 10,"Salir"); // Salir
boton_func(3,417,397,"Iniciar"); // Iniciar
boton_func(3,417,441," Alto "); // Alto
boton_func(3,607,135,"-"); // var elect
boton_func(3,607,152,""); // var elect
boton_func(3,482,179,"+"); // Disparo
boton_func(3,482,196,"-"); // Disparo
boton_func(3,482,257,"-"); // Adq horiz
boton_func(3,482,274,""); // Adq horiz
boton_func(3,482,301,"-"); // Adq vert A
boton_func(3,482,318,""); // Adq vert A
boton_func(3,482,345,"-"); // Adq vert B
boton_func(3,482,362,""); // Adq vert B
boton_func(3,607,257,"-"); // Ref esc horiz
boton_func(3,607,274,""); // Ref esc horiz
boton_func(3,607,301,"-"); // Ref esc vert
boton_func(3,607,318,""); // Ref esc vert
boton_func(3,607,345,"-"); // Ref horiz R0
boton_func(3,607,362,""); // Ref horiz R0
boton_func(3,607,389,"-"); // Ref horiz R1
boton_func(3,607,406,""); // Ref horiz R1
boton_func(3,607,433,"-"); // Ref vert R
boton_func(3,607,450,""); // Ref vert R
} // Fin. desactivabotones_func(void)
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void ventmenprin_func(void)
{
// Ventana con el menu principal
// Ventana con 20 opciones (dadas por la funcion, menbot_func() ):
// menbot_func() devuelve 0 si no es seleccionada alguna opcion
menbot_def=1;
menbot_nop=20; // numero de opciones
menbot_posx[ 1]=398; // Salir
menbot_posy[ 1]= 10; // Salir
menbot_text[ 1][1]="Salir"; // Salir
}

```

```

menbot_posx[ 2]=607; // Inc var elect
menbot_posy[ 2]=135; // Inc var elect
menbot_text[ 2][1]="-"; // Inc var elect
menbot_posx[ 3]=607; // Dec var elect
menbot_posy[ 3]=152; // Dec var elect
menbot_text[ 3][1]=""; // Dec var elect
menbot_posx[ 4]=482; // Disparo(+)
menbot_posy[ 4]=179; // Disparo(+)
menbot_text[ 4][1]="+"; // Disparo(+)
menbot_posx[ 5]=482; // Disparo(-)
menbot_posy[ 5]=196; // Disparo(-)
menbot_text[ 5][1]="-"; // Disparo(-)
menbot_posx[ 6]=417; // Disparo(-)
menbot_posy[ 6]=397; // Inic
menbot_text[ 6][1]="Iniciar"; // Inic
menbot_posx[ 7]=607; // Inc ref esc horiz
menbot_posy[ 7]=257; // Inc ref esc horiz
menbot_text[ 7][1]="-"; // Inc ref esc horiz
menbot_posx[ 8]=607; // Dec ref esc horiz
menbot_posy[ 8]=274; // Dec ref esc horiz
menbot_text[ 8][1]=""; // Dec ref esc horiz
menbot_posx[ 9]=607; // Dec ref esc horiz
menbot_posy[ 9]=301; // Inc ref esc vert
menbot_text[ 9][1]="-"; // Inc ref esc vert
menbot_posx[10]=607; // Inc ref esc vert
menbot_posy[10]=318; // Dec ref esc vert
menbot_text[10][1]=""; // Dec ref esc vert
menbot_posx[11]=607; // Dec ref esc vert
menbot_posy[11]=345; // Inc ref horiz R0
menbot_text[11][1]="-"; // Inc ref horiz R0
menbot_posx[12]=607; // Inc ref horiz R0
menbot_posy[12]=362; // Dec ref horiz R0
menbot_text[12][1]=""; // Dec ref horiz R0
menbot_posx[13]=607; // Dec ref horiz R0
menbot_posy[13]=389; // Inc ref horiz R1
menbot_text[13][1]="-"; // Inc ref horiz R1
menbot_posx[14]=607; // Dec ref horiz R1
menbot_posy[14]=406; // Dec ref horiz R1
menbot_text[14][1]=""; // Dec ref horiz R1
menbot_posx[15]=607; // Inc ref vert R
menbot_posy[15]=433; // Inc ref vert R
menbot_text[15][1]="-"; // Inc ref vert R
menbot_posx[16]=607; // Dec ref vert R
menbot_posy[16]=450; // Dec ref vert R
menbot_text[16][1]=""; // Dec ref vert R
menbot_posx[17]= 13; // Limpiar
menbot_posy[17]= 10; // Limpiar
menbot_text[17][1]="Limpiar"; // Limpiar
menbot_posx[18]= 90; // Abrir
menbot_posy[18]= 10; // Abrir
menbot_text[18][1]="Abrir"; // Abrir
menbot_posx[19]=151; // Abrir
menbot_posy[19]= 10; // Guardar A
menbot_text[19][1]="Guardar A"; // Guardar A
menbot_posx[20]=337; // Guardar A
menbot_posy[20]= 10; // Ayuda
menbot_text[20][1]="Ayuda"; // Ayuda
menbot_func(); // Ayuda
menbot_def=0; // Carga config vent
} // Fin. ventmenprin_func(void)
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void ventmenvn_func(void)
{
// Ventana con el menu A:voltaje, B:no usar
// Ventana con 5 opciones (dadas por la funcion, menbot_func() ):
// menbot_func() devuelve 0 si no es seleccionada alguna opcion
menbot_def=1;
menbot_nop=5; // numero de opciones
menbot_posx[ 1]=482; // Inc adq horiz
menbot_posy[ 1]=257; // Inc adq horiz
menbot_text[ 1][1]="-"; // Inc adq horiz
menbot_posx[ 2]=482; // Dec adq horiz
menbot_posy[ 2]=274; // Dec adq horiz

```

```

menbot_text[ 2][1]=""; // Dec adq horiz
menbot_posx[ 3]=482; // Inc adq vert A
menbot_posy[ 3]=301; // Inc adq vert A
menbot_text[ 3][1]="-"; // Inc adq vert A
menbot_posx[ 4]=482; // Dec adq vert A
menbot_posy[ 4]=318; // Dec adq vert A
menbot_text[ 4][1]=""; // Dec adq vert A
menbot_posx[ 5]=417; // Alto
menbot_posy[ 5]=441; // Alto
menbot_text[ 5][1]=" Alto "; // Alto
menbot_func(); // Carga conf vent
menbot_def=0;
} // Fin. ventmenvn_func(void)
////////////////////////////////////
// ***** FIN. FUNCIONES *****

// *****
// ***** FIN. VENTANAS *****
// *****

// *****
// ***** PRINCIPAL *****
// *****

// ***** VARIABLES GLOBALES *****
int ref_c; // Color de visualizacion de las referencias
int arch1_c; // Color de visualizacion del archivo 1 (superior)
int arch2_c; // Color de visualizacion del archivo 1 (inferior)
// ***** FIN VARIABLES GLOBALES *****

// ***** FUNCIONES *****
void main()
{
int i,j,irp,ini,velr,keyb,opcionmenprin,opcionmenprinant,salirmenprin;
int opcionmenvn,salirmenvn,presimga,presarch,archhoriz_ind,archvertv_ind;
int confvarel_ind,adqhoriz_ind,adqvertva_ind,refeschor_ind,refescver_ind;
int adqvertvb_ind,adqvertcb_ind,refhorr0,refhorr0ant,refhorrl;
int refhorrlant,refverr,refverrant,salirayuda,manauto_ind,disparo_ind;
int imgaux[512],colr0ant[256],colrlant[256],colrant[256];
double tpixr,tsegr,vertpixr,vertvoltsr;
void *panant;
unsigned int size;
FILE *archivo;
clock_t tref1,tref2;
char archvarel;
char cadena[33];
char *disparo_text[2][10]=
    {"Flanco(+)",
     "Flanco(-)"};
char *confvarel_text[5][25]=
    {"A:VOLTAJE, B:NO USAR "},
    {"A:NO USAR, B:CORRIENTE"},
    {"A:VOLTAJE, B:VOLTAJE "},
    {"A:IMPEDANCIA,B:NO USAR "},
    {"A:VOLTAJE, B:CORRIENTE"};
char *adqhoriz_text[16][10]=
    {" 10us/div"},
    {" 20us/div"},
    {" 50us/div"},
    {"100us/div"},
    {"200us/div"},
    {"500us/div"},
    {" 1ms/div"},
    {" 2ms/div"},
    {" 5ms/div"},
    {" 10ms/div"},
    {" 20ms/div"},
    {" 50ms/div"},
    {"100ms/div"},

```

```

        {"200ms/div"},
        {"500ms/div"},
        {" 1 s/div"};
char *adqvertva_text[13][10]=
    {" 10mV/div"},
    {" 20mV/div"},
    {" 50mV/div"},
    {"100mV/div"},
    {"200mV/div"},
    {"500mV/div"},
    {" 1 V/div"},
    {" 2 V/div"},
    {" 5 V/div"},
    {" 10 V/div"},
    {" 20 V/div"},
    {" 50 V/div"},
    {"100 V/div"};
char *refeschor_text[16][10]=
    {" 10us/div"},
    {" 20us/div"},
    {" 50us/div"},
    {"100us/div"},
    {"200us/div"},
    {"500us/div"},
    {" 1ms/div"},
    {" 2ms/div"},
    {" 5ms/div"},
    {" 10ms/div"},
    {" 20ms/div"},
    {" 50ms/div"},
    {"100ms/div"},
    {"200ms/div"},
    {"500ms/div"},
    {" 1 s/div"};
char *refescver_text[13][10]=
    {" 10mV/div"},
    {" 20mV/div"},
    {" 50mV/div"},
    {"100mV/div"},
    {"200mV/div"},
    {"500mV/div"},
    {" 1 V/div"},
    {" 2 V/div"},
    {" 5 V/div"},
    {" 10 V/div"},
    {" 20 V/div"},
    {" 50 V/div"},
    {"100 V/div"};
presimga=0;
presarch=0;
velr=1;
trefl=0;
tref2=clock();
confvarel_ind=0;
disparo_ind=1;
adqhoriz_ind=0;
adqvertva_ind=12;
refeschor_ind=0;
refescver_ind=12;
refhorr0=146;
refhorr0ant=146;
refhorr1=146;
refhorrlant=146;
refverr=185;
refverrant=185;
salirmenprin=0;
archivo=fopen("COLORES.DAT","r");
fgets(cadena,80,archivo);
cadena[3]='\0';
ambtrab_cfondo=atoi(cadena);
fgets(cadena,80,archivo);
cadena[3]='\0';
boton_color=atoi(cadena);

```

```

fgets(cadena,80,archivo);
cadena[3]='\0';
pant_cfondo=atoi(cadena);
fgets(cadena,80,archivo);
cadena[3]='\0';
pant_cejepri=atoi(cadena);
fgets(cadena,80,archivo);
cadena[3]='\0';
pant_cejesecc=atoi(cadena);
fgets(cadena,80,archivo);
cadena[3]='\0';
display_cfondo=atoi(cadena);
fgets(cadena,80,archivo);
cadena[3]='\0';
display_ctext=atoi(cadena);
fgets(cadena,80,archivo);
cadena[3]='\0';
textfond_ctext=atoi(cadena);
fgets(cadena,80,archivo);
cadena[3]='\0';
relieve_grueso=atoi(cadena);
fgets(cadena,80,archivo);
cadena[3]='\0';
grupo_color=atoi(cadena);
fgets(cadena,80,archivo);
cadena[3]='\0';
grupo_grueso=atoi(cadena);
fgets(cadena,80,archivo);
cadena[3]='\0';
visimga_c=atoi(cadena);
fgets(cadena,80,archivo);
cadena[3]='\0';
ref_c=atoi(cadena);
fgets(cadena,80,archivo);
cadena[3]='\0';
arch1_c=atoi(cadena);
fgets(cadena,80,archivo);
cadena[3]='\0';
arch2_c=atoi(cadena);
fclose(archivo);
iniamb_func();
ventfija_func();
desactivabotones_func();
ventmenprin_func();
size = imagesize(17,56,17+258,56+258);
panant = malloc(size);
cargaadqhoriz(adqhoriz_ind);
cargaadqvertva(adqvertva_ind);
for (i=0; i<=255; i=i+1)
{
    colr0ant[i]=pant_cejepri;
    colrlant[i]=pant_cejepri;
    colrant[i]=pant_cejepri;
}
for (i=0; i<512; i=i+1)
{
    imgaant[i]=128;
    colaant[i]=pant_cejepri;
    imgbant[i]=128;
    colbant[i]=pant_cejepri;
}
while (salirmenprin!=1)
{
    opcionmenprin=menbot_func();
    switch(opcionmenprin)
    {
        case 1:
            salirmenprin=i;
            break;
        case 2:
            confvarel_ind=confvarel_ind-1;
            if (confvarel_ind<0)
                confvarel_ind=0;
    }
}

```

```

        textdisp_func(404,160,confvarel_text[confvarel_ind][0]);
        break;
    case 3:
        confvarel_ind=confvarel_ind+1;
        if (confvarel_ind>4)
            confvarel_ind=4;
        textdisp_func(404,160,confvarel_text[confvarel_ind][0]);
        break;
    case 4:
        disparo_ind=disparo_ind-1;
        if (disparo_ind<0)
            disparo_ind=0;
        textdisp_func(404,204,disparo_text[disparo_ind][0]);
        break;
    case 5:
        disparo_ind=disparo_ind+1;
        if (disparo_ind>1)
            disparo_ind=1;
        textdisp_func(404,204,disparo_text[disparo_ind][0]);
        break;
    case 6:
        if (confvarel_ind==0)
        {
            presimga=1;
            i=display_ctext;
            display_ctext=visimga_c;
            textdisp_func(309,138,"Canal A ");
            display_ctext=i;
            salirmenvn=0;
            desactivabotones_func();
            ventmenvn_func();
            textdisp_func(404,282,adqhoriz_text[adqhoriz_ind][0]);
            textdisp_func(404,326,adqvertva_text[adqvertva_ind][0]);
            while(salirmenvn!=1)
            {
                opcionmenvn=menbot_func();
                switch(opcionmenvn)
                {
                    case 1:
                        adqhoriz_ind=adqhoriz_ind-1;
                        if (adqhoriz_ind<0)
                            adqhoriz_ind=0;
                        textdisp_func(404,282,
                            adqhoriz_text[adqhoriz_ind][0]);
                        break;
                    case 2:
                        adqhoriz_ind=adqhoriz_ind+1;
                        if (adqhoriz_ind>15)
                            adqhoriz_ind=15;
                        textdisp_func(404,282,
                            adqhoriz_text[adqhoriz_ind][0]);
                        break;
                    case 3:
                        adqvertva_ind=adqvertva_ind-1;
                        if (adqvertva_ind<0)
                            adqvertva_ind=0;
                        if (adqvertva_ind> 6)
                            textdisp_func(309,270,"Activo ");
                        if (adqvertva_ind<= 6)
                            textdisp_func(309,270,"Inactivo");
                        textdisp_func(404,326,
                            adqvertva_text[adqvertva_ind][0]);
                        break;
                    case 4:
                        adqvertva_ind=adqvertva_ind+1;
                        if (adqvertva_ind> 12)
                            adqvertva_ind= 12;
                        if (adqvertva_ind> 6)
                            textdisp_func(309,270,"Activo ");
                        if (adqvertva_ind<= 6)
                            textdisp_func(309,270,"Inactivo");
                        textdisp_func(404,326,
                            adqvertva_text[adqvertva_ind][0]);
                }
            }
        }

```

```

        break;
    case 5:
        salirmenvn=1;
        salirmenprin=0;
        desactivabotones_func();
        ventmenprin_func();
        break;
    }
    //////////////////////////////////////
    cargaadqhoriz(adqhoriz_ind);
    cargaadqvertva(adqvertva_ind);
    cargaimgab(& imga[0], & imgb[0]);
    //senocargaimgab(& imga[0], & imgb[0]);
    ini=indiniimga(disparo_ind);
    visimga(ini);
    if ((clock()-trefl)>20) // actualiza val cada seg aprox
    {
        trefl=clock();
        varvoltimga(ini, adqhoriz_ind, adqvertva_ind);
    }
    //////////////////////////////////////
}
break;
case 7:
case 8:
switch(opcionmenprin)
{
    case 7:
        refeschor_ind=refeschor_ind-1;
        if (refeschor_ind<0)
            refeschor_ind=0;
        break;
    case 8:
        refeschor_ind=refeschor_ind+1;
        if (refeschor_ind>15)
            refeschor_ind=15;
        break;
}
textdisp_func(529, 282, refeschor_text[refeschor_ind][0]);
tpixr=abs(refhorr1-refhorr0);
tsegr=tseg_func(refeschor_ind, tpixr);
prefijosiu(tsegr, cadena);
cadena[6]='s';
cadena[7]=' ';
cadena[8]=' ';
cadena[9]='\0';
textdisp_func(529, 414, cadena);
cadena[0]='-';
textdisp_func(529, 370, cadena);
break;
case 9:
case 10:
    if (opcionmenprin==9)
    {
        refescver_ind=refescver_ind-1;
        if (refescver_ind<0)
            refescver_ind=0;
    }
    else
    {
        refescver_ind=refescver_ind+1;
        if (refescver_ind>12)
            refescver_ind=12;
    }
    textdisp_func(529, 326, refescver_text[refescver_ind][0]);
    vertpixr=57+128-refverr;
    vertvoltsr=vertvolts_func(refescver_ind, vertpixr);
    prefijosiu(vertvoltsr, cadena);
    cadena[6]='V';
    cadena[7]=' ';
    cadena[8]=' ';
    cadena[9]='\0';
}

```

```

        textdisp_func(529,458,cadena);
        break;
case 11:
case 12:
case 13:
case 14:
    if (opcionmenprinant==opcionmenprin && (clock()-tref2)<3)
        velr=5;
    else
        velr=1;
    opcionmenprinant=opcionmenprin;
    tref2=clock();
    switch(opcionmenprin)
    {
        case 11:
        case 12:
            switch(opcionmenprin)
            {
                case 11:
                    refhorr0=refhorr0-velr;
                    if (refhorr0<18)
                        refhorr0=refhorrl;
                    break;
                case 12:
                    refhorr0=refhorr0+velr;
                    if (refhorr0>refhorrl)
                        refhorr0=18;
                    break;
            }
            for (i=0;i<=255;i=i+3)
            {
                irp=i+57;
                putpixel(refhorr0ant,irp,colr0ant[i]);
                colr0ant[i]=getpixel(refhorr0,irp);
                putpixel(refhorr0,irp,ref_c);
            }
            refhorr0ant=refhorr0;
            break;
        case 13:
        case 14:
            switch(opcionmenprin)
            {
                case 13:
                    refhorrl=refhorrl-velr;
                    if (refhorrl<refhorr0)
                        refhorrl=274;
                    break;
                case 14:
                    refhorrl=refhorrl+velr;
                    if (refhorrl>274)
                        refhorrl=refhorr0;
                    break;
            }
            for (i=0;i<=255;i=i+3)
            {
                irp=i+57;
                putpixel(refhorrlant,irp,colrlant[i]);
                colrlant[i]=getpixel(refhorrl,irp);
                putpixel(refhorrl,irp,ref_c);
            }
            refhorrlant=refhorrl;
            break;
    }
    tpixr=abs(refhorrl-refhorr0);
    tsegr=tseg_func(refeschor_ind,tpixr);
    prefijosiu(tsegr,cadena);
    cadena[6]='s';
    cadena[7]=' ';
    cadena[8]=' ';
    cadena[9]='\0';
    textdisp_func(529,414,cadena);
    cadena[0]='-';
    textdisp_func(529,370,cadena);

```



```

        break;
case 15:
case 16:
    if (opcionmenprinant==opcionmenprin && (clock()-tref2)<3)
        velr=5;
    else
        velr=1;
    opcionmenprinant=opcionmenprin;
    tref2=clock();
    switch(opcionmenprin)
    {
        case 15:
            refverr=refverr-velr;
            if (refverr<57)
                refverr=312;
            break;
        case 16:
            refverr=refverr+velr;
            if (refverr>312)
                refverr=57;
            break;
    }
    for (i=0;i<=255;i=i+3)
    {
        irp=i+18;
        putpixel(irp,refverrant,colrant[i]);
        colrant[i]=getpixel(irp,refverr);
        putpixel(irp,refverr,ref_c);
    }
    refverrant=refverr;
    vertpixr=57+128-refverr;
    vertvoltsr=vertvolts_func(refescver_ind,vertpixr);
    prefijosiu(vertvoltsr,cadena);
    cadena[6]='V';
    cadena[7]=' ';
    cadena[8]=' ';
    cadena[9]='\0';
    textdisp_func(529,458,cadena);
    break;
case 17:
    ventfija_func();
    presimga=0;
    presarch=0;
    confvarel_ind=0;
    disparo_ind=1;
    adqhoriz_ind=0;
    adqvertva_ind=12;
    refeschor_ind=0;
    refescver_ind=12;
    refhorr0=146;
    refhorr0ant=146;
    refhorrl=146;
    refhorrlant=146;
    refverr=185;
    refverrant=185;
    cargaadqhoriz(adqhoriz_ind);
    cargaadqvertva(adqvertva_ind);
    for (i=0; i<=255; i=i+1)
    {
        colr0ant[i]=pant_cejepri;
        colrlant[i]=pant_cejepri;
        colrant[i]=pant_cejepri;
    }
    for (i=0; i<512; i=i+1)
    {
        imgaant[i]=128;
        colaant[i]=pant_cejepri;
        imgbant[i]=128;
        colbant[i]=pant_cejepri;
    }
    break;
case 18:
    getimage(17,56,17+258,56+258,panant);

```

```

setfillstyle(1,display_cfondo);
bar(17,56,17+258,56+258);
textdisp_func(18+8* 0,57+8* 0,"Abrir");
if (presarch==2)
{
textdisp_func(18+8* 0,57+8* 2,
"Espacio de Archivos saturado.");
textdisp_func(18+8* 0,57+8* 31,
"Pulse cualquier tecla para cont.");
getch();
putimage(17,56,panant,COPY_PUT);
}
else
{
textdisp_func(18+8* 0,57+8* 2,"Archivo:");
cadena[0]='_';
for (i=1;i<=12;i=i+1)
cadena[i]=' ';
cadena[13]='\0';
textdisp_func(18+8* 9,57+8* 2,cadena);
i=0;
keyb=0;
while (keyb!=13)
{
keyb=getch();
if (keyb!=8)
{
if (keyb!=13)
{
cadena[i]=keyb;
if (i<=11)
i=i+1;
}
}
else
{
if (i>=1)
i=i-1;
cadena[i]=32;
cadena[i+1]=32;
}
cadena[i]='_';
textdisp_func(18+8* 9,57+8* 2,cadena);
}
for (i=0;i<=11;i=i+1)
if (cadena[i]!='_')
cadena[i]='\0';
if ((archivo=fopen(cadena,"r"))== NULL)
{
textdisp_func(18+8* 0,57+8* 4,
"No se puede abrir el archivo");
textdisp_func(18+8* 0,57+8* 31,
"Pulse cualquier tecla para cont.");
getch();
putimage(17,56,panant,COPY_PUT);
}
else
{
putimage(17,56,panant,COPY_PUT);
fscanf(archivo,"%c", &archvarel);
fscanf(archivo,"%d", &archhoriz_ind);
fscanf(archivo,"%d", &archvertv_ind);
switch(presarch)
{
case 0:
for (i=0;i<=255;i=i+1)
{
fscanf(archivo,"%d",&j);
putpixel(i+18,256-j+57,arch1_c);
}
textdisp_func(309,60,"");
i=display_ctext;
display_ctext=arch1_c;
}
}
}
}

```

```

        textdisp_func(309,60,cadena);
        display_ctext=i;
        textdisp_func(442,60,
        refeschor_text[archhoriz_ind][0]);
        textdisp_func(551,60,
        refescver_text[archvertv_ind][0]);
        presarch=1;
        break;
    case 1:
        for (i=0;i<=255;i=i+1)
            {
                fscanf(archivo,"%d",&j);
                putpixel(i+18,256-j+57,arch2_c);
            }
        textdisp_func(309,82,"");
        i=display_ctext;
        display_ctext=arch2_c;
        textdisp_func(309,82,cadena);
        display_ctext=i;
        textdisp_func(442,82,
        refeschor_text[archhoriz_ind][0]);
        textdisp_func(551,82,
        refescver_text[archvertv_ind][0]);
        presarch=2;
        break;
    }
    }
    fclose(archivo);
}
break;
case 19:
getimage(17,56,17+258,56+258,panant);
setfillstyle(1,display_cfondo);
bar(17,56,17+258,56+258);
textdisp_func(18+8* 0,57+8* 0,"Guardar A");
if (presimga==0)
{
    textdisp_func(18+8* 0,57+8* 2,
    "La imagen del canal A no esta");
    textdisp_func(18+8* 0,57+8* 4,
    "presente.");
    textdisp_func(18+8* 0,57+8* 31,
    "Pulse cualquier tecla para cont.");
    getch();
    putimage(17,56,panant,COPY_PUT);
}
else
{
    textdisp_func(18+8* 0,57+8* 2,"Archivo:");
    cadena[0]=' ';
    for (i=1;i<=12;i=i+1)
        cadena[i]=' ';
    cadena[13]='\0';
    textdisp_func(18+8* 9,57+8* 2,cadena);
    i=0;
    keyb=0;
    while (keyb!=13)
        {
            keyb=getch();
            if (keyb!=8)
                {
                    if (keyb!=13)
                        {
                            cadena[i]=keyb;
                            if (i<=11)
                                i=i+1;
                        }
                }
            else
                {
                    if (i>=1)
                        i=i-1;
                    cadena[i]=32;
                }
        }
}
}

```

```

        cadena[i+1]=32;
    }
    cadena[i]='_';
    textdisp_func(18+8* 9,57+8* 2,cadena);
}
for (i=0;i<=11;i=i+1)
    if (cadena[i]!='_')
        cadena[i]='\0';
putimage(17,56,panant,COPY_PUT);
archivo=fopen(cadena,"w");
fprintf(archivo,"V\n");
fprintf(archivo,"%d\n",adqhoriz_ind);
fprintf(archivo,"%d\n",adqvertva_ind);
ini=indiniimga(disparo_ind);
for (i=ini;i<=ini+255;i=i+1)
    fprintf(archivo,"%d\n",imga[i]);
fclose(archivo);
}
break;
case 20:
getimage(17,56,17+258,56+258,panant);
setfillstyle(1,display_cfondo);
bar(17,56,17+258,56+258);
archivo=fopen("AYUDA.DAT","r");
i=0;
salirayuda=0;
while (!feof(archivo) && salirayuda==0)
{
fgets(cadena,80,archivo);
cadena[32]='\0';
textdisp_func(18+8* 0,57+8* i,cadena);
i=i+1;
if (i>=30)
{
textdisp_func(18+8* 0,57+8* 31,
"Continuar: [ENTER]; Salir:[ESC]");
if (getch()==27)
salirayuda=1;
else
{
i=0;
setfillstyle(1,display_cfondo);
bar(17,56,17+258,56+258);
}
}
}
if (salirayuda==0)
{
textdisp_func(18+8* 0,57+8* 31,
"Pulse cualquier tecla para cont.");
getch();
}
fclose(archivo);
putimage(17,56,panant,COPY_PUT);
break;
default:
break;
}
}
}
////////////////////////////////////
finamb_func();
}
// ***** FIN. FUNCIONES *****
// *****
// ***** FIN. PRINCIPAL *****
// *****

```

## ARCHIVO DE AYUDA (AYUDA.DAT)

LIMPIAR. LA PANTALLA Y TODOS  
LOS VISUALIZADORES SON  
ESTABLECIDOS EN SU CONDICIÓN  
INICIAL.

ABRIR. LA INFORMACIÓN CONTENIDA  
EN UN ARCHIVO LA CARGA EN  
PANTALLA. DE ESTE ARCHIVO  
TAMBIÉN OBTIENE SU NOMBRE, SU  
ESCALA DE TIEMPO (ESCALA  
HORIZONTAL) Y SU ESCALA  
VERTICAL.

GUARDAR A. LA INFORMACIÓN EN  
PANTALLA, LA ESCALA DE TIEMPO Y  
LA ESCALA VERTICAL  
CORRESPONDIENTES AL CANAL A ES  
GUARDADA EN UN ARCHIVO CON  
CIERTO NOMBRE.

GUARDAR B. LA INFORMACIÓN EN  
PANTALLA, LA ESCALA DE TIEMPO Y  
LA ESCALA VERTICAL  
CORRESPONDIENTES AL CANAL B ES  
GUARDADA EN UN ARCHIVO CON  
CIERTO NOMBRE.

AYUDA. DESPLIEGA INFORMACIÓN DE  
LA MANERA DE UTILIZAR EL  
SISTEMA.

SALIR. SALIR AL SISTEMA  
OPERATIVO.

VARIABLES ELÉCTRICAS. SE INDICA  
A QUE MENÚ SE TENDRÁ ACCESO Y  
SE SELECCIONA EL TIPO DE  
VARIABLES ELÉCTRICAS QUE SE  
SENSARÁN DURANTE LA ADQUISICIÓN  
DE DATOS, TIENE 4 OPCIONES:  
A: VOLTAJE, B: NO USAR  
A: NO USAR, B: CORRIENTE  
A: VOLTAJE, B: VOLTAJE  
A: IMPEDANCIA, B: NO USAR  
A: VOLTAJE, B: CORRIENTE

DISPARO. INDICA EL TIPO DE  
DISPARO QUE SE APLICARÁ DURANTE  
LA ADQUISICIÓN. EL DISPARO  
SIRVE PARA LA SINCRONIZACIÓN DE  
UNA SEÑAL, DE MANERA QUE ESTA  
EMPIECE A VISUALIZARSE DESDE EL  
MISMO PUNTO, ESTE PUNTO PUEDE  
SER EL INICIO DE LA MITAD  
SUPERIOR DE LA SEÑAL (FLANCO  
POSITIVO) O EL INICIO DE LA  
MITAD INFERIOR DE LA SEÑAL  
(FLANCO NEGATIVO)

ESC HORIZ. SE SELECCIONA LA  
ESCALA DE LAS REFERENCIAS  
HORIZONTALES

ESC VERT. SE SELECCIONA LA  
ESCALA DE LA REFERENCIA  
VERTICAL

HORIZ RO. CON ESTO SE CAMBIA LA  
POSICIÓN DE LA REFERENCIA

HORIZONTAL RO, SU VALOR ES CON RESPECTO A LA REFERENCIA HORIZONTAL R1 Y DEPENDE DE LA ESCALA HORIZONTAL ESTABLECIDA PARA LAS REFERENCIAS HORIZONTALES.

HORIZ R1. CON ESTO SE CAMBIA LA POSICIÓN DE LA REFERENCIA HORIZONTAL R1, SU VALOR ES CON RESPECTO A LA REFERENCIA HORIZONTAL RO Y DEPENDE DE LA ESCALA HORIZONTAL ESTABLECIDA PARA LAS REFERENCIAS HORIZONTALES.

VERT R. SE CAMBIA LA POSICIÓN DE LA REFERENCIA VERTICAL, SU VALOR ES CON RESPECTO A CERO, QUE ES CENTRO DE LA PANTALLA VERTICALMENTE Y DEPENDE DE LA ESCALA VERTICAL ESTABLECIDA PARA LA REFERENCIA VERTICAL.

INICIAR. SE INICIA LA ADQUISICIÓN DE DATOS Y SE CAMBIA AL MENÚ CORRESPONDIENTE AL TIPO DE VARIABLES YA SELECCIONADAS, CON ESTO YA SE PUEDEN EMPEZAR A UTILIZAR LOS BOTONES AGRUPADOS BAJO "ADQ" (TODOS LOS DEMÁS QUEDAN INACTIVOS).

EN LOS VISUALIZADORES SE MUESTRAN CARACTERÍSTICAS DE LA SEÑALES QUE SE ESTÁN SENSANDO DE ACUERDO A LA VARIABLE QUE SE ESTA MIDIENDO. SI SE ESTA UTILIZANDO EL CANAL A Y EL CANAL B NO, SOLO SE PRESENTA LA INFORMACIÓN DE LA AGRUPACIÓN "CANAL A". SI SE ESTÁN USANDO LOS DOS CANALES SE PRESENTA INFORMACIÓN TANTO EN LA AGRUPACIÓN "CANAL A" COMO EN LA AGRUPACIÓN "CANAL B". CUANDO SE MIDE IMPEDANCIA Y POTENCIA SE PRESENTA INFORMACIÓN EN LAS AGRUPACIONES "CANAL A" Y "CANAL B", ASÍ COMO EN LA AGRUPACIÓN "IMPEDANCIA" O "POTENCIA", SEGÚN SEA EL CASO. PARA EL CASO DE MEDICIONES DE VOLTAJE, EL ATENUADOR PARA CADA CANAL ESTA ACTIVO PARA ESCALAS MAYORES A 1 V/DIV. EN LA AGRUPACIÓN "ADQ", SE MUESTRA EL CANAL O CANALES QUE SE ESTA USANDO CON EL MISMO COLOR QUE SU SEÑAL ASOCIADA QUE SE ESTA PRESENTANDO EN PANTALLA. EN LA AGRUPACIÓN "ARCHIVO" SE PRESENTAN LOS NOMBRES DE LOS ARCHIVOS QUE FUERON ABIERTOS CON LA OPCIÓN "ABRIR". CUANDO SE ABRE UN ARCHIVO, TAMBIÉN ES VISUALIZADA EN LA PANTALLA SU SEÑAL ASOCIADA CON SUS RESPECTIVAS ESCALAS HORIZONTAL Y VERTICAL.

---

## ARCHIVO DE DEFINICION DE COLORES (COLORES.DAT)

```
7  ambtrab_cfondo
3  boton_color // valores entre 1-5,8 ,default 3
6  pant_cfondo
15 pant_cejepri
1  pant_cejeseq
6  display_cfondo
14 display_ctext
6  textfond_ctext
1  relieve_grueso
6  grupo_color
2  grupo_grueso
11 visimga_c
14 ref_c
15 arch1_c
14 arch2_c
```

---

# APÉNDICE II

## *ECUACIONES Y*

## *NOMENCLATURA*

---

### ECUACIONES PARA EL CÁLCULO DE POTENCIA

Cuando el sistema esta configurado para medir en el CANAL A voltaje y en el CANAL B corriente, es posible calcular por software la potencia aparente (S), la potencia activa o real (P), la potencia reactiva (Q) y el factor de potencia (f. p.), de acuerdo al siguiente análisis:

La potencia compleja S se define como el producto de los fasores de voltaje y corriente:

$$S = V_A I_B^*$$

También se puede representar de la siguiente forma:

$$S = V_A e^{j\phi_A} I_B e^{-j\phi_B}$$

Donde:

$V_A$  y  $V_B$  son las amplitudes de  $V_A$  y  $V_B$

$\phi_A$  y  $\phi_B$  son los ángulos de defasamiento de  $V_A$  y  $V_B$



Reduciendo términos semejantes, obtenemos:

$$S = V_A I_B e^{j(\phi_A - \phi_B)}$$

Aplicando la identidad de Euler, obtenemos:

$$S = V_A I_B \left[ \cos(\phi_A - \phi_B) + j \operatorname{Sen}(\phi_A - \phi_B) \right]$$

Otra forma en que se define la potencia compleja  $S$ , es la siguiente:

$$S = P + jQ$$

La parte real es la potencia real o activa  $P$ .

La parte imaginaria es la potencia reactiva  $Q$ .

Igualando términos semejantes en la última y penúltima ecuación se obtiene:

$$P = V_A I_B \left[ \cos(\phi_A - \phi_B) \right]$$

$$Q = V_A I_B \left[ \operatorname{Sen}(\phi_A - \phi_B) \right]$$

Designemos a  $\theta$  como la diferencia angular entre el voltaje y la corriente.

$$\theta = \phi_A - \phi_B$$

Así:

$$P = V_A I_B \cos \theta$$

$$Q = V_A I_B \operatorname{Sen} \theta$$

La potencia aparente  $S$ , es la magnitud de la potencia compleja, así:

$$S = V_A I_B$$

El factor de potencia f.p., se define como la potencia real entre la potencia aparente.

$$\text{f.p.} = \frac{P}{S}$$

$$\text{f.p.} = \cos \theta$$

Resumiendo:

Factor de potencia:	$\text{f.p.} = \cos \theta$
Potencia Aparente:	$S = V_A I_B$
Potencia Real o Activa:	$P = V_A I_B \cos \theta$
Potencia Reactiva:	$Q = V_A I_B \text{ Sen } \theta$

## ECUACIONES PARA EL CÁLCULO DE IMPEDANCIA

El diseño se basó en el siguiente circuito:

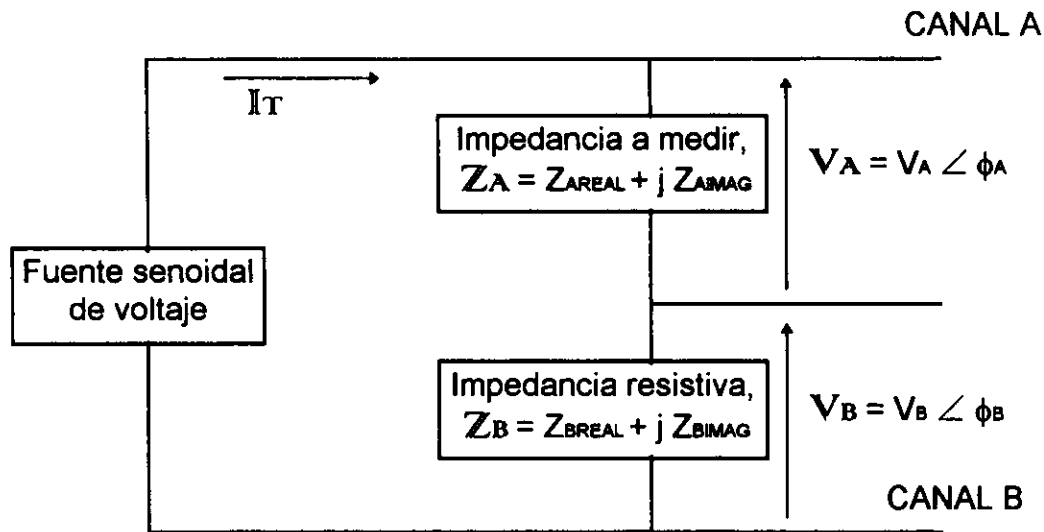


Figura II.1

Si se conocen los parámetros de la fuente senoidal de voltaje y la impedancia resistiva  $Z_B$ , entonces se puede conocer la impedancia a medir  $Z_A$ . En otras palabras  $Z_A$  está en función de la velocidad angular  $\omega$  de la fuente senoidal, de las magnitudes  $V_A$  y  $V_B$ , los ángulos de defasamiento  $\phi_A$  y  $\phi_B$ , de acuerdo al siguiente análisis:

En el circuito serie (Figura II.1), se muestra una resistencia de referencia  $Z_B$ , que es el resistor programable y una impedancia a medir  $Z_A$ , conectadas a una fuente senoidal de voltaje, la cuál produce una corriente de prueba  $I_T$  que es la misma en todo el circuito, obteniéndose:

$$I_T = \frac{V_A}{Z_A} = \frac{V_B}{Z_B}$$

Despejando  $Z_A$ :

$$Z_A = Z_B \frac{V_A}{V_B}$$

Expresando los fasores en términos de su magnitud y ángulo de defasamiento:

$$V_A = V_A \angle \phi_A$$

$$V_B = V_B \angle \phi_B$$

Así:

$$Z_A = Z_B \left[ \frac{V_A}{V_B} \angle (\phi_A - \phi_B) \right]$$

Empleando la identidad de Euler, obtenemos la siguiente ecuación:

$$Z_A = Z_B \left[ \frac{V_A}{V_B} e^{j(\phi_A - \phi_B)} \right]$$

Desarrollando:

$$Z_A = Z_B \left[ \frac{V_A}{V_B} (\cos(\phi_A - \phi_B) + j \operatorname{Sen}(\phi_A - \phi_B)) \right]$$

Como  $Z_B$  es puramente resistiva,  $Z_{BIMAG} = 0$ .

$$Z_B = Z_{BREAL} + j Z_{BIMAG} = Z_{BREAL}$$

Si designamos,  $Z_{BREAL} = R_B$ , el valor del resistor programable y  $\delta = \phi_A - \phi_B$

$$Z_A = R_B \left[ \frac{V_A}{V_B} (\cos \delta + j \operatorname{Sen} \delta) \right]$$

$$\mathbf{Z}_A = \left( R_B \frac{V_A}{V_B} \cos \delta \right) + j \left( R_B \frac{V_A}{V_B} \operatorname{Sen} \delta \right) = Z_{\text{AREAL}} + j Z_{\text{AIMAG}}$$

Así:

Parte Real de la impedancia desconocida	$Z_{\text{AREAL}} = R_B \frac{V_A}{V_B} \cos \delta$
Parte imaginaria de la impedancia desconocida	$Z_{\text{AIMAG}} = R_B \frac{V_A}{V_B} \operatorname{Sen} \delta$

A continuación se consideran tres casos, que la impedancia a medir sea resistiva, inductiva y capacitiva.

Si la impedancia a medir es puramente resistiva,  $V_A$  esta en fase con  $V_B$ , por lo tanto  $\delta = 0$ .

$$\mathbf{Z}_A = \left( R_B \frac{V_A}{V_B} \cos 0 \right) + j \left( R_B \frac{V_A}{V_B} \operatorname{Sen} 0 \right)$$

$$\mathbf{Z}_A = R_B \frac{V_A}{V_B}$$

Como  $Z_A$  se considera una impedancia puramente resistiva, se cumple:

$$\mathbf{Z}_A = R_A$$

Donde:

$R_A$  es la resistencia correspondiente a  $Z_A$ .

Igualando términos en la última y penúltima ecuación se obtiene:

$$R_A = R_B \frac{V_A}{V_B}$$

Si la impedancia a medir es puramente inductiva,  $V_A$  se adelanta  $90^\circ$  con respecto a  $V_B$ , por lo tanto  $\delta = 90^\circ$ .

$$Z_A = \left( R_B \frac{V_A}{V_B} \cos 90^\circ \right) + j \left( R_B \frac{V_A}{V_B} \operatorname{Sen} 90^\circ \right)$$

$$Z_A = j R_B \frac{V_A}{V_B}$$

Como  $Z_A$  se considera una impedancia puramente inductiva, se cumple lo siguiente:

$$Z_A = j\omega L_A$$

Donde:

$\omega$  es la velocidad angular de la fuente senoidal de voltaje.

$L_A$  es la inductancia correspondiente a  $Z_A$ .

Igualando términos en la última y penúltima ecuación se obtiene:

$$\omega L_A = R_B \frac{V_A}{V_B}$$

Despejando  $L_A$ :

$$L_A = \frac{R_B V_A}{\omega V_B}$$

Si la impedancia a medir es puramente capacitiva,  $V_A$  se atrasa  $90^\circ$  con respecto a  $V_B$ , por lo tanto  $\delta = -90^\circ$ .

$$Z_A = \left( R_B \frac{V_A}{V_B} \cos -90^\circ \right) + j \left( R_B \frac{V_A}{V_B} \operatorname{Sen} -90^\circ \right)$$

$$Z_A = -j R_B \frac{V_A}{V_B}$$

Como  $Z_A$  se considera una impedancia puramente capacitiva, se cumple lo siguiente:

$$Z_A = -j \frac{1}{\omega C_A}$$

Donde:

$\omega$  es la velocidad angular de la fuente senoidal de voltaje.

$C_A$  es la capacitancia correspondiente a  $Z_A$ .

Igualando términos en la última y penúltima ecuación se obtiene:

$$\frac{1}{\omega C_A} = R_B \frac{V_A}{V_B}$$

Despejando  $C_A$ :

$$C_A = \frac{V_B}{\omega R_B V_A}$$

Idealmente para un elemento resistivo  $\delta = 0^\circ$ , para un inductivo  $\delta = 90^\circ$  y para un capacitivo  $\delta = -90^\circ$ , pero en la práctica esto no sucede, así, para la determinación del tipo de elemento se considera lo siguiente:

$\delta$	Tipo	Valor
$-90^\circ$ a $-60^\circ$	Capacitivo	$C_A = \frac{V_B}{\omega R_B V_A}$
$-15^\circ$ a $15^\circ$	Resistivo	$R_A = R_B \frac{V_A}{V_B}$
$60^\circ$ a $90^\circ$	Inductivo	$L_A = \frac{R_B V_A}{\omega V_B}$
$-60^\circ$ a $-15^\circ$ y $15^\circ$ a $60^\circ$	Indeterminado	

## DISEÑO DE LA FUENTE SENOIDAL DE VOLTAJE

Se emplea el circuito integrado XR-8038 como fuente senoidal de voltaje (ver figura II.2), este integrado emplea un número mínimo de componentes externos: dos potenciómetros de  $100\text{ k}\Omega$  ( $R_1$  y  $R_2$ ), un capacitor  $C$  y dos resistencias ( $R_3$  y  $R_4$ ); el rango de frecuencias de operación lo fija el capacitor  $C$ .

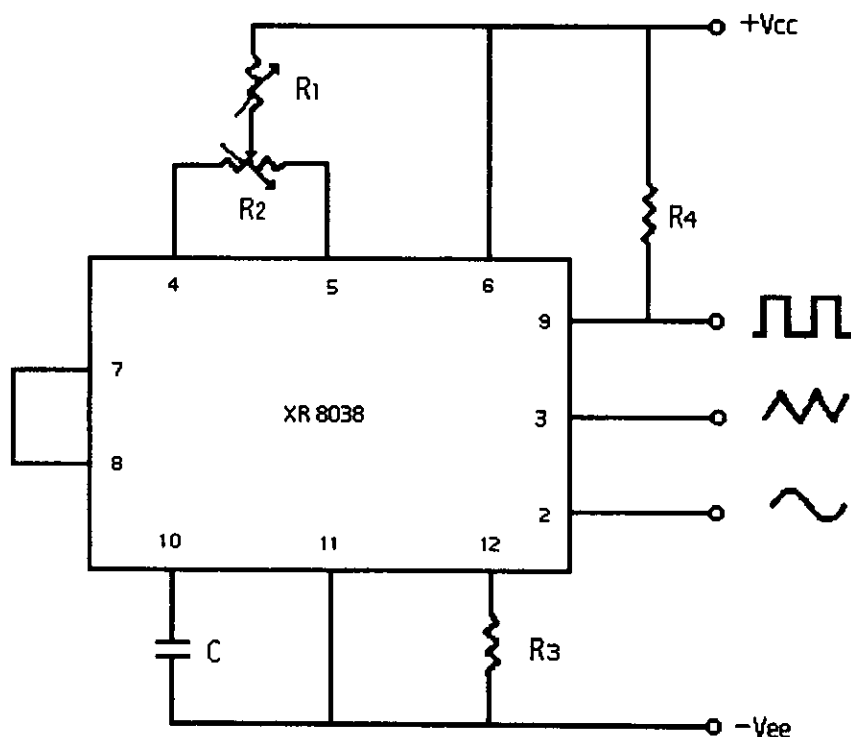


Figura II.2

De acuerdo a recomendaciones del manual del XR-8038, los valores de los componentes externos son:  $R_4 = 82\text{ k}\Omega$ ,  $R_3 = 100\text{ k}\Omega$ ,  $R_1 =$  potenciómetro de  $100\text{ k}\Omega$ ,  $R_2 =$  potenciómetro de  $100\text{ k}\Omega$ .

El voltaje de polarización del circuito es  $\pm 12\text{ V}$ . La salida senoidal que proporciona el generador es  $0.11\text{ V}$  de amplitud en el pin 2, por lo que se amplifica para obtener una amplitud de  $1\text{ V}$ . La simetría de la onda senoidal es ajustada con el potenciómetro



externo  $R_2$ , con este se puede variar el ciclo de trabajo de la señal del 2 % al 98 % y se conecta entre el pin 4 y pin 5. El potenciómetro  $R_1$ , con el que se varía la frecuencia de la señal senoidal, se conecta entre  $V_{cc}$  y el cursor del potenciómetro  $R_2$ . La resistencia  $R_3$  sirve para reducir la distorsión de la onda senoidal y se conecta entre el pin 11 y -Vee. El capacitor  $C$  se conecta entre el pin 10 y -Vee. La resistencia  $R_4$  se conecta entre  $V_{cc}$  y el pin 9 que es la salida de onda cuadrada.

Se utilizó este circuito debido a su estabilidad en un amplio rango de temperatura, mínima distorsión de la señal en un amplio rango de frecuencia y está es independiente del suministro de voltaje.

La frecuencia de operación se determina de acuerdo a los elementos externos con la ecuación siguiente:

$$f = \frac{1}{\frac{5}{3} R_1 C \left[ 1 + \frac{R_2}{2R_1 - R_2} \right]}$$

despejando la capacitancia  $C$ :

$$C = \frac{1}{\frac{5}{3} R_1 f \left[ 1 + \frac{R_2}{2R_1 - R_2} \right]}$$

Sustituyendo, los valores de los potenciómetros  $R_1$  y  $R_2$  por su valor central, para obtener el valor del capacitor  $C$  para una frecuencia de 1.59 kHz .

$$C = \frac{1}{\frac{5}{3} 50k\Omega 1.59kHz \left[ 1 + \frac{50k\Omega}{100k\Omega - 50k\Omega} \right]} = 3.77 \text{ nF}$$

## COMPENSACIÓN EN FRECUENCIA, LF356 Y LF357

Los amplificadores operacionales LF356 y LF357 se pueden utilizar en aplicaciones de bajo ruido, debido a sus características (ver hojas de especificaciones del LF356 Y LF357 en el apéndice IV).

La capacitancia parásita de entrada reduce el ancho de banda de la señal, esta tiene un valor de 3 pf que limita la frecuencia de operación del amplificador, para disminuir este efecto se conecta un capacitor en paralelo a la resistencia de realimentación (ver la figura II.3) de acuerdo a la relación siguiente:

$$R_2 C_2 = R_1 C_1$$

Despejando  $C_2$ :

$$C_2 = C_1 \left( \frac{R_1}{R_2} \right)$$

Si  $R_2 = R_1 = 10 \text{ k}\Omega$ .

$$C_2 = 3 \text{ pf} \left( \frac{10 \text{ k}\Omega}{10 \text{ k}\Omega} \right) = 3 \text{ pF}$$

Pero en la práctica se utilizo un capacitor de 10 pF, de acuerdo al diagrama mostrado.

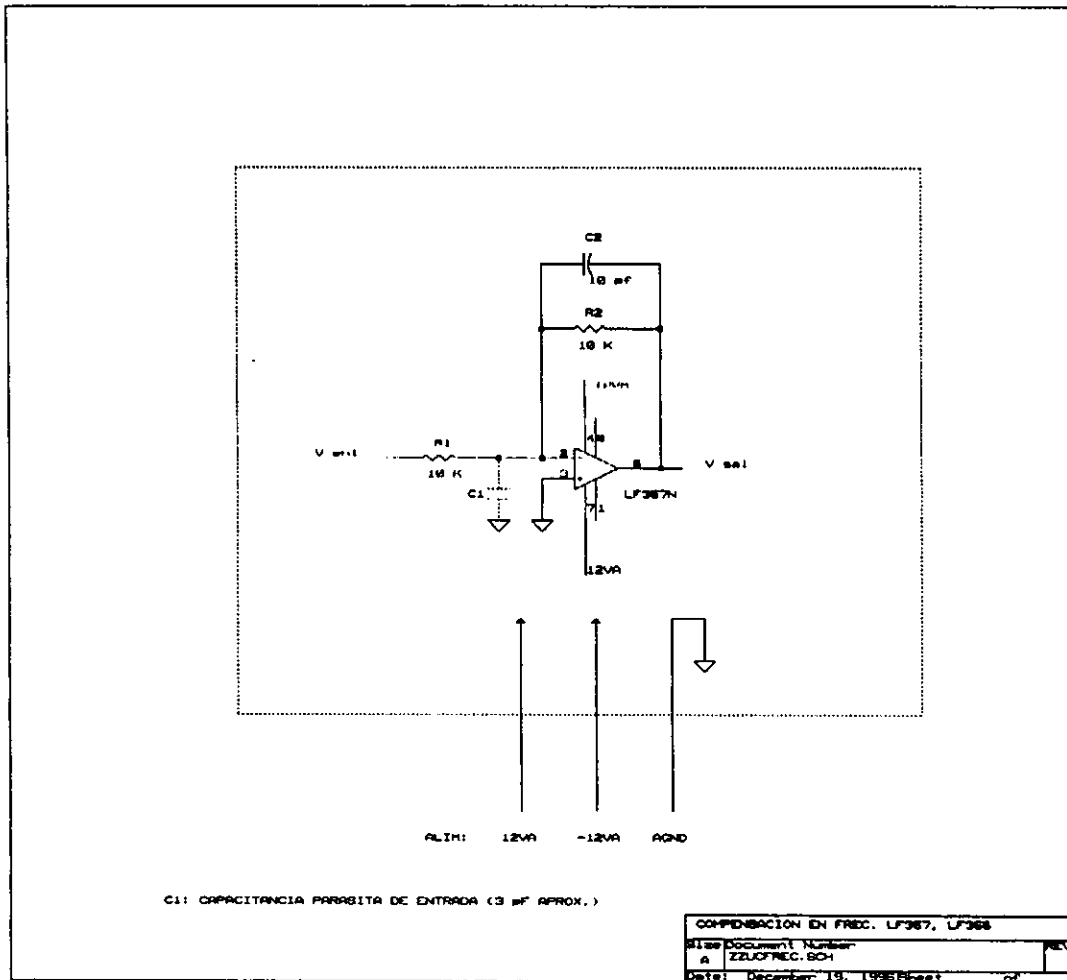


Figura II.3

## COMPENSACIÓN DE OFFSET, LF356 Y LF357

De modo ideal, sin un voltaje de entrada aplicado a un amplificador operacional, el voltaje en la salida es cero, pero en la práctica la salida está en el rango de algunos milivolts para una señal de entrada cero, este voltaje se deriva de la diferencia entre componentes internos y debido al voltaje de entrada de apagado  $V_{io}$ .

La corriente de entrada de apagado  $I_{io}$  es una cantidad que contribuye también al voltaje de error de salida, ya que existe una pequeña corriente de polarización de CD

en cada una de las terminales de entrada, si las corrientes son desiguales su diferencia es la corriente de entrada de apagado, la corriente  $I_{io}$  esta en el rango de nanoamperes.

Para compensar el voltaje de error, se coloca un potenciómetro (ver hojas de especificaciones del LF356 Y LF357 en el apéndice IV) de  $25\text{ k}\Omega$  entre las terminales 1 y 5 para ajustar la salida a  $0\text{ V}$  cuando no hay señal de entrada, el cursor se conecta a la fuente de alimentación positiva de acuerdo a la siguiente figura.

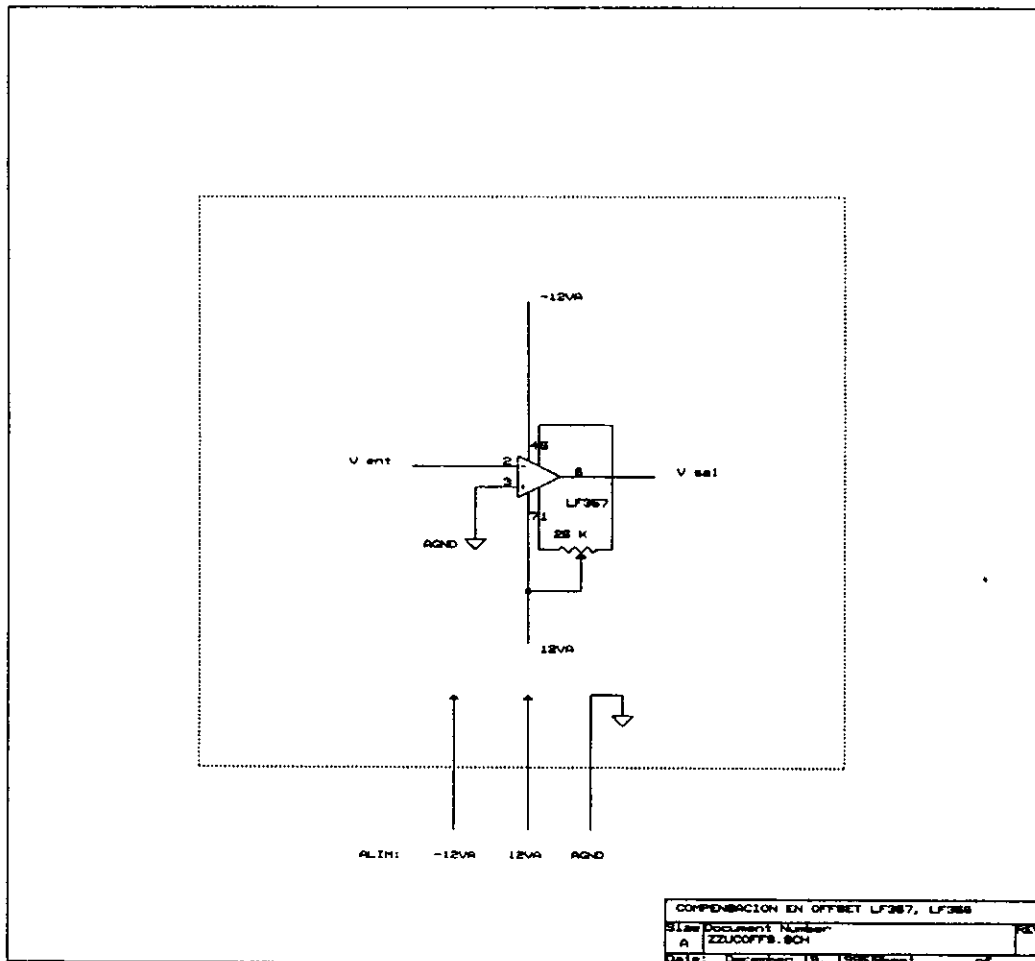


Figura II.4

---

## NOMENCLATURA

**GND:** Es la tierra de referencia de los elementos, dispositivos y sistemas cuyas variables eléctricas se quieren medir.

**VINA:** Borne correspondiente al canal A referenciado a GND.

**VINB:** Borne correspondiente al canal B referenciado a GND.

**CFS:** señal de control digital en el bloque "Sensor de Variables Eléctricas", cuando es válida en alto (1 lógico) activa la fuente senoidal.

**VLA-VHA, VLB-VHB:** señales de control digitales en el bloque "Sensor de Variables Eléctricas", cuando están en alto (1 lógico) no se activan los divisores de voltaje y cuando están en bajo (0 lógico) se activan los divisores de voltaje para altos voltajes en los canales A y B respectivamente.

**CR2, CR1, CR0:** Señales de control digitales en el bloque "Sensor de Variables Eléctricas", por medio de las cuáles se selecciona el valor del resistor programable.

**RST:** Señal de control digital proveniente del slot de la PC, la cuál es válida cuando se inicia el proceso de encendido de la PC y el estado del sistema es desconocido. Esta señal activa los relevadores a un estado conocido de protección.

**AX2, AX5, AX10:** Son las señales de control digital para el bloque "Amplificador de Ganancia Programable", cuando son válidas la señal correspondiente al canal A se multiplica ya sea por 2, por 5 o por 10.

---

**BX2, BX5, BX10:** Son las señales de control digital para el bloque "Amplificador de Ganancia Programable", cuando son válidas la señal correspondiente al canal B se multiplica ya sea por 2, por 5 o por 10.

**VRT:** Referencia de voltaje mayor (5 V) para las resistencias internas de los convertidores analógico digital correspondientes a los canales A y B.

**VRB:** Referencia de voltaje menor (3 V) para las resistencias internas de los convertidores analógico digital correspondientes a los canales A y B

**ESCTM3, ESCTM2, ESCTM1, ESCTM0:** Son las señales de control digitales para seleccionar las escalas de tiempo en el bloque "Generación del Reloj de Muestreo".

**CKM:** Señal del reloj de muestreo, salida del bloque "Generación del Reloj de Muestreo".

**CKT:** Señal del reloj de transmisión dado por una línea de un puerto de escritura de la PC.

**-CLADR:** Establece en ceros a los contadores que generan las direcciones de memoria en el bloque "Control", para empezar el modo de muestreo o de transmisión.

**T/-M:** Señal de control digital en el bloque "Selección de CK de Muestreo y Generación de Direcciones", cuando está en alto (1 lógico) indica que se está en el modo de transmisión y cuando está en bajo (0 lógico) indica que se está en el modo de muestreo.

---

**CTATERM:** Señal que detiene la cuenta de los contadores que generan las direcciones de memoria cuando han llegado a su cuenta máxima en el bloque "Control".

**R/-W:** Señal del bloque "Control", que establece a las memorias RAM en los bloques "CAD", en lectura si es 1 lógico y en escritura si es 0 lógico respectivamente.

**-G:** Señal del bloque "Control", que establece a los buffers de tres estados en los bloques "CAD" en estado de alta impedancia (-G en 1 lógico) o en estado de conductividad (-G en 0 lógico) para evitar cortocircuito en el bus de datos.

**CONV= -CE:** Señal de reloj de muestreo para los convertidores analógico digital correspondientes al canal A y B, ésta señal también habilita a las memorias RAM sincronizandolas con los convertidores en los bloques "CAD".

**A10, A9, A8, A7, A6, A5, A4, A3, A2, A1, A0:** Son las señales que constituyen las direcciones de las memorias RAM, en los bloques "CAD".

**INVA, INVB:** Señales de control que indican a los bloques "CAD" que inviertan los datos digitales de las señales analógicas de los canales A y B.

**DA0-DA7:** Información digital correspondiente a la señal analógica del CANAL A y es la salida del bloque "CAD".

**DB0-DB7:** Información digital correspondiente a la señal analógica del CANAL B y es la salida del bloque "CAD".

**DR00-DR07:** Son las líneas del puerto 0 que es de lectura, estas líneas constituyen el bus DR0 en el bloque "Puertos de Entrada y Salida".

---

**DR10-DR17:** Son las líneas del puerto 1 que es de lectura, estas líneas constituyen el bus DR1 en el bloque "Puertos de Entrada y Salida".

**DR20-DR27:** Son las líneas del puerto 2 que es de lectura, estas líneas constituyen el bus DR2 en el bloque "Puertos de Entrada y Salida".

**DW30-DW37:** Son las líneas del puerto 3 que es de escritura, estas líneas constituyen el bus DW3 en el bloque "Puertos de Entrada y Salida".

**DW40-DW47:** Son las líneas del puerto 4 que es de escritura, estas líneas constituyen el bus DW4 en el bloque "Puertos de Entrada y Salida".

**DW50-DW57:** Son las líneas del puerto 5 que es de escritura, estas líneas constituyen el bus DW5 en el bloque "Puertos de Entrada y Salida".

**-IOR2, -IOR1, -IOR0:** Son las líneas de selección de los puertos de lectura, estas líneas constituyen el bus -IOR en el bloque "Arbitraje del Bus de Datos y Lógica de Decodificación".

**-IOW2, -IOW1, -IOW0:** Son las líneas de selección de los puertos de escritura, éstas líneas constituyen el bus -IOW en el bloque "Arbitraje del Bus de Datos y Lógica de Decodificación".

**RES, OSC, CLK:** Son las mismas señales especificadas para el slot XT, pero reforzadas con buffers.

**-IOCK, IORDY:** Son las mismas señal especificadas para el slot XT.



**IRQ:** Es la línea de interrupción que puede ser IRQ2, IRQ3, IRQ4, IRQ5, IRQ6, IRQ7 dependiendo de la posición de un dip-switch, en el bloque "Arbitraje del Bus de Datos y Lógica de Decodificación".

**D7, D6, D5, D4, D3, D2, D1, D0:** Son las mismas líneas del bus de datos para el slot XT pero reforzadas con buffers.

Para las líneas del slot XT, ver el apéndice III: La PC enfocada a tarjetas de expansión.

---

# **APÉNDICE III**

## ***LA PC ENFOCADA A TARJETAS DE EXPANSIÓN***

---

### **CONCEPTOS GENERALES Y ESTANDARES DE EXPANSIÓN DE LA PC.**

El bus de expansión proporciona un medio de adhesión no sólo para dispositivos periféricos propios de la PC, sino también para una amplia variedad de sistemas electrónicos externos. Así muchos fabricantes desarrollan productos de control, de adquisición de datos, de instrumentación, etc., que trabajan junto con la PC a través del bus de expansión.

Un bus de expansión proporciona acceso completo a una gran variedad de señales del sistema PC, estas señales pueden ser divididas como sigue: bus de líneas de direcciones, bus de líneas de datos, señales de control de lectura y escritura, señales de petición de interrupción, señales de reconocimiento de DMA y de petición de DMA, señales de control general, señales de reloj y líneas de alimentación.

Dos estándares fundamentales son empleados en el esquema del bus de expansión de la PC convencional. El original y más extensamente usado, se basa en el "Industry Standard Architecture" (ISA) comúnmente conocido como el Bus de Expansión de la PC. Este esquema utiliza entre uno y dos conectores directos.

El primer conector (62 líneas) proporciona acceso al bus de 8 bits, a la mayoría de las señales de control y líneas de alimentación, mientras que el segundo conector (36 líneas) da acceso al bus de datos restante junto con algunas señales de control adicionales.

Existe gran variedad de tarjetas diseñadas para diferentes propósitos y que apoyan a la computadora en tareas especiales, algunos ejemplos son: tarjetas de Entrada/Salida analógica, tarjetas de Entrada/Salida digital, tarjetas de Entrada/Salida digitales opto-aisladas, tarjetas con líneas de salida adaptadas con relevadores, tarjetas de expansión de memoria, tarjetas manejadoras de disco duro, tarjetas para redes, tarjetas prototipo, tarjetas para comunicación serial, tarjetas Modem, tarjetas de instrumentación especializada, etc.

## CARACTERÍSTICAS FÍSICAS DE LAS TARJETAS DE EXPANSIÓN DE LA PC

Las tarjetas de expansión del sistema PC tienden a ser pequeñas. Sin embargo, el máximo disponible del adaptador y del conector para la PC (y PS/2) es:

ESTÁNDAR	BUS	ALTURA(mm)	LARGO(mm)	ANCHO(mm)
ISA	XT	107	3.35	12.7
ISA	AT	107	3.35	12.7
MCA	PS/2	96	3.35	12.7

Es importante notar que aunque la XT 286 esta basada en una tarjeta principal AT, sus expansiones están sujetas a las restricciones de altura generales impuestas en las tarjetas XT (4.2 pulgadas o 107 mm máximo).

Otra dificultad es que algunas tarjetas XT pueden fallar al operar en el equipo de AT debido a la interrupción del circuito DMA.

## **BUS DE EXPANSIÓN DE LA PC (ESTANDAR ISA)**

Como ya se mencionó, este esquema consta de dos conectores. El primer conector (62 líneas) proporciona acceso al bus de 8 bits, a la mayoría de las señales de control y líneas de alimentación, mientras que el segundo conector (36 líneas) da acceso al bus de datos restante junto con algunas señales de control adicionales. Aplicaciones que requieren únicamente 8 bits de datos y el estándar de señales de control pueden usar únicamente el primer conector. En aplicaciones que requieren acceso a 16 bits (no disponible en maquinas PC antiguas y PC-XT) pueden ser usados ambos conectores.

Una tarjeta de expansión puede ser diseñada para conectarse únicamente al conector de 62 líneas o puede hacer pareja con el conector de 36 líneas. En las máquinas antiguas sólo se tenía adaptado el conector de 62 líneas (que tenían el bus de 8 bits); las tarjetas diseñadas con este conector son conocidas como tarjetas de expansión de 8 bits o tarjetas de expansión PC/XT. La maquina AT, sin embargo proporciona acceso completo a los 16 bits del bus de datos junto con las señales de control, por lo que se requiere el conector de 36 líneas.

Las tarjetas que son diseñadas para hacer uso de los dos conectores son conocidas generalmente como tarjetas de expansión de 16 bits o tarjetas de expansión AT. Las tarjetas diseñadas para la AT (están específicamente diseñadas para tomar ventaja de la disponibilidad del bus de datos de 16 bits) ofrecen una considerable ventaja en la velocidad sobre las que están basadas en el bus de datos de 8 bits,

---

proporcionados por el conector de expansión de la original XT. En algunas aplicaciones, esta ventaja en la velocidad puede ser crítica.

## **CONECTOR DEL BUS DE EXPANSIÓN (ISA) DE 62 LINEAS**

El conector del bus de expansión de 62 líneas es adaptado a la tarjeta principal. El lado derecho (vista superior) del conector es referido como A (líneas enumeradas de A1 a A31) mientras el lado izquierdo es referido como B (líneas enumeradas de B1 a B31). El bus de datos y direcciones son agrupados en el lado A del conector mientras el bus de control y líneas de alimentación ocupan el lado B.

Algunas PC antiguas no utilizaban letras A y B en el conector de expansión, de un lado utilizaban números nones del 1 al 61 y del otro lado números pares del 2 al 62.

La figura III.1 describe cada una de las señales presentes en el conector del bus de expansión de 62 líneas.

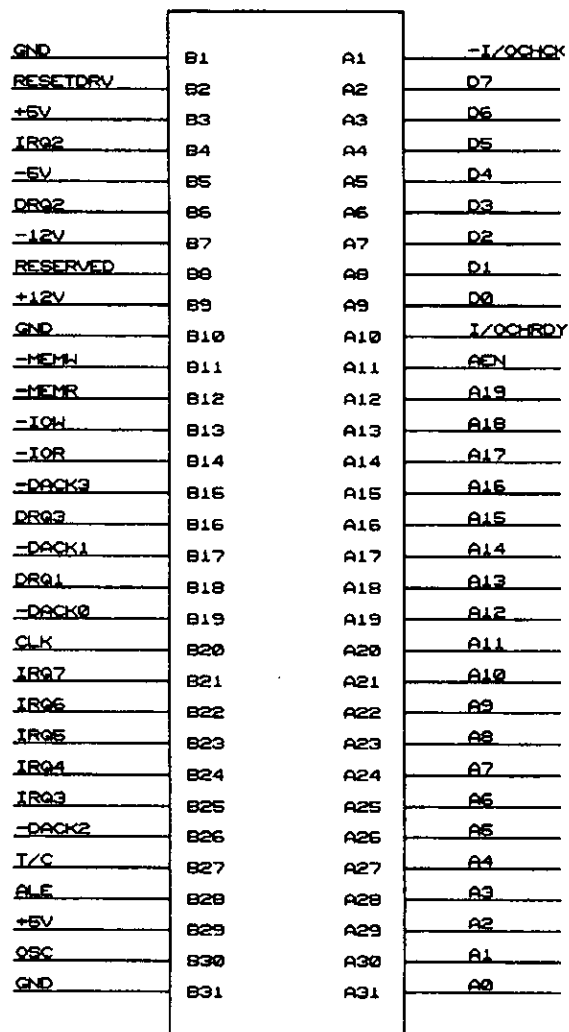


Figura III.1

**OSC (Oscilador).** Señal de salida con frecuencia de 14.31818 MHz, ciclo de trabajo de 50% y período de 70 ns aproximadamente. Esta es la señal de más alta frecuencia en el bus PC y todas las otras señales de tiempo son generadas de esta señal.

**CLK (Reloj).** Señal de salida generada dividiendo OSC por 3, dando una frecuencia de 4.77 MHz. Su período es 210 ns con un tiempo alto de 70 ns y uno bajo de 140 ns.

**RESETDRV (Manejador de Reset).** Señal de salida puesta en activo en nivel alto, en la secuencia de encendido de la PC. Permanece activa hasta que todos los niveles han alcanzado su rango de operación y entonces pasa a inactiva. Su función es dar un reset a las interfaces y dispositivos I/O unidos al bus PC para llevarlos a un estado conocido.

**A0-A19.** Líneas de dirección son señales de salida que son usadas para direccionar memoria y dispositivos I/O.

**D0-D7.** Líneas de datos bidireccionales usadas para transmitir datos entre el microprocesador, memoria y dispositivos I/O.

**ALE (Latch de direcciones habilitado).** Señal de salida, indica que el bus de direcciones es ahora válido para el comienzo de un ciclo de bus. Esta señal va a activo justo antes de que el bus de direcciones es válido y cae a inactiva en nivel bajo, justo después de que el bus de direcciones es válido.

**I/OCHCK (Chequeo del canal I/O).** Señal de entrada (nivel bajo) usada para reportar condiciones de error en interfaces unidas al bus. Esta señal cuando es baja generará una interrupción no mascarable (NMI) al el microprocesador.

**I/OCHRDY (Canal I/O listo).** Señal de entrada que es usada para extender el ciclo de bus, de manera que las memorias y puertos I/O que no son bastante rápidas para responder a el ciclo de bus normal de 4 periodos (840 ns) pueden también estar unidas a el bus PC del sistema.

**IRQ2 hasta IRQ7 (Petición de interrupción 2-7).** Señales de entrada usadas para generar petición de interrupción a el microprocesador. IRQ2 es la señal de más alta prioridad e IRQ7 es la de menor prioridad.

**IOR (Lectura I/O).** Señal de salida activa en nivel bajo, usada para indicar a los puertos I/O que el presente ciclo de bus ya iniciado es un ciclo de lectura I/O y que la dirección en el bus de direcciones es una dirección de puerto I/O.

**IOW (Escritura I/O).** Señal de salida activa en nivel bajo, usada para indicar a los puertos I/O que el presente ciclo de bus ya iniciado es un ciclo de escritura I/O y que la dirección en el bus de direcciones es una dirección de puerto I/O.

**MEMW (Escritura de memoria).** Señal de salida activa en nivel bajo, usada para escribir datos desde el bus del sistema en memoria. Manejada por el bus de control durante el ciclo de bus ya iniciado, esta señal es usada para indicar que el bus de direcciones contiene la dirección de una localidad de memoria a la cual el dato en el bus de datos le será escrita.

**MEMR (Lectura de memoria).** Señal de salida activa en nivel bajo, usada para pedir leer datos de la memoria. Manejada por el bus de control durante el ciclo de bus ya iniciado, esta señal es usada para indicar que el bus de direcciones contiene la dirección de una localidad de memoria desde la cual el dato será leído.

**DRQ1 hasta DRQ3.** Líneas de entrada activas en nivel alto, usadas por la interface para pedir un ciclo DMA. Si un dispositivo o una interface quiere transferir datos entre el mismo y la memoria sin la intervención del CPU, la petición es inicializada activando la línea DRQ.

**DACK0 hasta DACK3.** Señales de salida activas en nivel bajo, provienen del controlador de DMA 8237 para indicar que la correspondiente DRQ ha sido concedida y el controlador de DMA tomara el bus y procederá con el ciclo DMA pedido.



**AEN (dirección habilitada).** Señal de salida activa en nivel alto, indica que un ciclo de bus DMA esta en progreso. Cuando esta activo, el controlador de DMA controlara el bus de direcciones, el bus de datos y las líneas de lectura/escritura de la memoria e I/O.

**T/C (Terminal de conteo).** Señal de salida activa en nivel alto, proviene del controlador de DMA para indicar que uno de los canales DMA ha alcanzado su preprogramado número de ciclos de transferencia. Esta señal es típicamente usada para terminar un bloque de datos transferidos por DMA.

**+5V, -5V, +12V, -12V, GND.** Alimentación y tierra.

## **CARACTERÍSTICAS ELÉCTRICAS DEL BUS DE LA PC**

En el diseño de una tarjeta de expansión para la PC se deben considerar los requerimientos de la alimentación y la distribución de las líneas de alimentación.

Todas las líneas presentes del slot son compatibles con TTL. En el caso de las señales de salida de la tarjeta principal, el máximo de carga impuesto por una tarjeta de expansión deberá ser limitada a no más de dos dispositivos LS de TTL.

La disponibilidad de la alimentación para la tarjeta de expansión depende de el rango de suministro de la fuente de alimentación del sistema; los requerimientos de la tarjeta principal y las demandas de otras tarjetas. Cuando se diseñan tarjetas de expansión, los limites recomendados por tarjeta para cada línea de alimentación son las siguientes:

Línea de voltaje	Conexión	Máximo de corriente
+5 Volts	B3 y B29	1.5 mA
+5 Volts	B5	100 mA
+12 Volts	B9	500 mA
-12 Volts	B7	100 mA

Cuando varias tarjetas son adaptadas se debe estimar y calcular la corriente y potencia total que demanda cada línea de alimentación. Esto debe ser sin exceder el rango de la fuente de suministro. En algunos casos debe de ser menor que 1.5 Watts.

Como una guía, la siguiente tabla de datos da referencia de la potencia de las líneas de alimentación para los sistemas XT, XT-286 y AT.

Sistema	XT	XT- 286	AT
Total de potencia (Watts)	130[W]	157[W]	192[W]
Voltaje	Rango de corriente max (A)		
+5 Volts	15[A]	20[A]	20[A]
-5 Volts	0.3[A]	0.3[A]	0.3[A]
+12 Volts	4.2[A]	4.2[A]	7.3[A]
-12 Volts	0.25[A]	0.25[A]	0.3[A]

Un sistema principal completo (incluyendo coprocesador matemático) requiere aproximadamente 4A en la línea de +5V y 2A en la línea de +12V. Una tarjeta de vídeo VGA y dos manejadores de unidad de disco flexibles demandaran adicionalmente 2.4A y 1.8A en la línea de +5V y 12V respectivamente. Un ventilador requiere de 0.3A en la línea de alimentación de +12V. La carga total es de 6.4A en la línea de +5V y 4.1A en la línea de +12V. Con una XT estándar la fuente de suministro reserva 8.6A disponibles de la línea de +5V y únicamente 100mA disponibles de la fuente de +12V.

---

## DISEÑO DE PUERTOS PARA LA PC

Cuando los puertos pueden ser accesados por el microprocesador como direcciones de memoria, se da lo que se conoce como mapa de memoria de Entrada/Salida, comúnmente conocida como mapa de puertos. Las instrucciones IN y OUT del microprocesador de Intel proporcionan la transferencia de datos para los puertos de Entrada/Salida.

La figura III.2 muestra un diagrama de bloques de un ejemplo de operación de entrada/salida muy sencillo. La lógica de decodificación da el pulso de selección llamado chip select recibiendo señales del bus de direcciones y la señales de control IOR e IOW. Un pulso de selección (chip select) habilita el puerto de entrada para dar acceso a los datos desde el dispositivo de entrada. Otro pulso de selección (chip select) habilita el puerto de salida para dar acceso al dato dado por el procesador. El puerto de entrada incluye un circuito integrado tres estados para aislar la entrada del bus excepto durante el ciclo de lectura. En la salida se usan dispositivos de memoria (registros) para guardar el dato hasta que la salida relativamente lenta lo acepte.

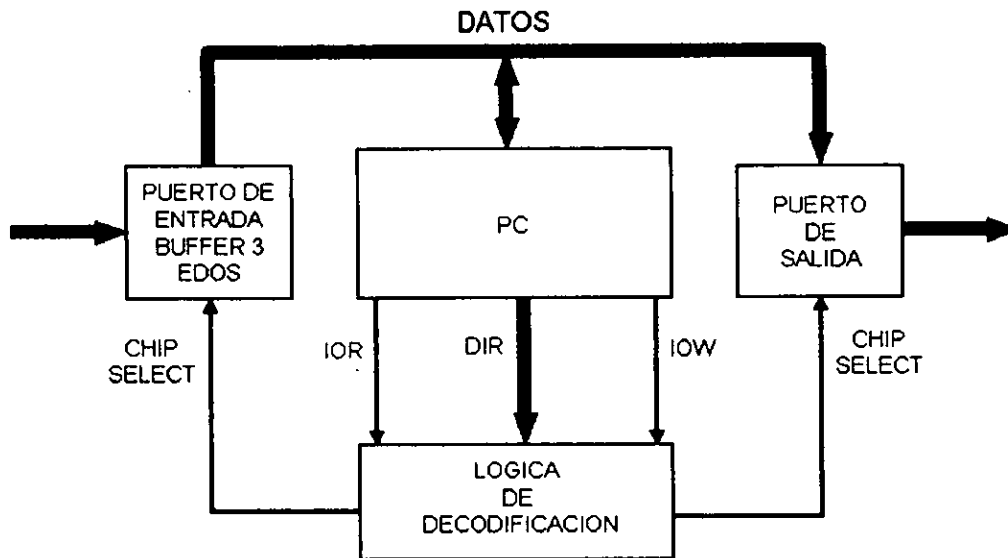


Figura III.2

Hay dos tipos de programas que controlan la entrada/salida: el condicional e incondicional. La transferencia incondicional lleva o trae datos del puerto de entrada/salida sin determinar si el puerto envía o recibe datos. Para esta transferencia de datos, el procesador asume que el dispositivo de entrada/salida está leyendo o enviando datos. Pueden ocurrir errores en las transferencias de datos si el programador no es cuidadoso. Si el microprocesador envía datos más rápidamente que la capacidad de recepción del puerto de salida, los datos se perderán. También en el puerto de entrada se pueden tener errores si los datos son leídos por el microprocesador con velocidades elevadas.

Para evitar estos problemas, los puertos son configurados para una transferencia de datos condicional, usando un protocolo de comunicación. Así el estado de la información es enviado incondicionalmente desde el dispositivo de entrada/salida hacia el microprocesador antes de la transferencia del dato. El estado de la información se encuentra en un bit, llamado bandera que indica la condición de disposición de un puerto de entrada/salida. El software examina el estado de la bandera y si se indica que el dispositivo todavía no está listo incrementa el tiempo asociado con la

operación Entrada/Salida. Si varios dispositivos de entrada salida son usados en un sistema con programa de entrada/salida, el proceso necesita verificar la bandera por cada dispositivo en turno, este proceso es conocido como de poleo.

Un ejemplo de uso de un protocolo de comunicación, en la operación de entrada/salida se da a continuación. Un dispositivo de entrada tiene datos disponibles en el puerto 1 para la transmisión a la microcomputadora. Para indicar la disponibilidad de datos, el dispositivo de entrada envía un bit de bandera en el puerto de entrada 0. La microcomputadora periódicamente accesa la bandera de estado en el puerto de entrada 0 y la examina. Si es 1, el dato esta disponible y la microcomputadora acepta el dato en el puerto de entrada 1. Si es 0, se le da más tiempo al dispositivo, hasta que el dato este disponible y esta bandera sea 1.

## MAPA DE DIRECCIONES DE PUERTOS DE LA PC

Para generar las señales de selección (chip select), se necesita conocer las direcciones del puerto de Entrada/Salida y el mapa de asignación del puerto. El diseño de la PC proporciona 10 bits para las direcciones de los puertos desde A0 hasta A9 para direccionar un total de 1024 puertos. El mapa de direcciones de puertos esta dividido en dos partes. Las direcciones desde 000H a 1FFH están asignadas para la tarjeta principal. El espacio de las direcciones de 200H a 3FFH se reservan para las tarjetas de expansión.

La siguiente tabla muestra las direcciones de los puertos que están destinados a específicos tipos de tarjetas.

Se tienen únicamente 32 direcciones (300H a 31FH) designadas a puertos para las tarjetas prototipos.

Rango de direcciones (hex)	Uso
000-1FF	Usada por la tarjeta de sistema base
200	No usado o reservado
201	Control de juego
202-20F	No usado o reservado
210-217.	Unidad de expansión
218-277.	No usado o reservado
278-27F	Segundo puerto de impresora
280-2F7	No usado o reservado
2F8-2FF	Comunicación asíncrona (COM2)
300-31F	Tarjetas prototipo
320-32F	Disco duro
330-377	No usado o reservado
378-37F	Puerto de impresora
380-38C	Comunicaciones SLDC
38D-38F	No usado o reservado
390-399	Comunicación síncrona binaria (2)
3A0-3A9	Comunicación síncrona binaria (1)
3AA-3AF	No usado o reservado
3B0-3BF	Desplegado monocromático IBM e impresora.
3C0-3CF	No usado o reservado
3D0-3DF	Color y gráficos
3E0-3EF	No usado o reservado
3F0-3F7	Diskette drive 5 1/4 pulg.
3F8-3FF	Comunicación asíncrona (COM1)

---

## DESCRIPCIÓN DEL CICLO DE BUS PARA LA PC

Un ciclo de bus es aquel en el cual el microprocesador lee o escribe en el bus de datos. Los ciclos de bus son normalmente de 4 ciclos de reloj pero el diseño de la PC inserta tiempos adicionales que constituyen el llamado ciclo de espera. Si un dispositivo no responde es necesario utilizar el pin IOCHRDY para alargar (genera el ciclo de espera) el ciclo de bus del sistema.

El ciclo de lectura del bus se inicia cada vez que el procesador ejecuta una instrucción de entrada (IN). El ciclo de escritura del bus se inicia cada vez que el procesador ejecuta una instrucción de salida (OUT).

### Descripción del ciclo de lectura

Durante T1 (ver la figura III.3) la señal ALE es activada, lo cual indica en el flanco de bajada que los bits A0 hasta A15 del bus de direcciones contienen una dirección válida de puertos. Durante T2, la señal -IOR del bus de control es activada, lo cual indica que se está direccionando al puerto de entrada para poder utilizar el bus de datos para la transferencia de datos. En el comienzo de T4, el procesador lee el dato del bus de datos y la señal de -IOR es desactivada.

### Descripción del ciclo de escritura

Durante T1 la señal ALE es activada, lo cual indica en el flanco de bajada que el bus de direcciones contiene una dirección válida de puertos. Después, la señal -IOW del bus de control se activa durante T2, para seleccionar al puerto de salida que deberá tomar el dato del bus. Más tarde durante T2, el procesador colocará en el bus el dato para el puerto de salida. En el tiempo de T4, la señal -IOW es desactivada.

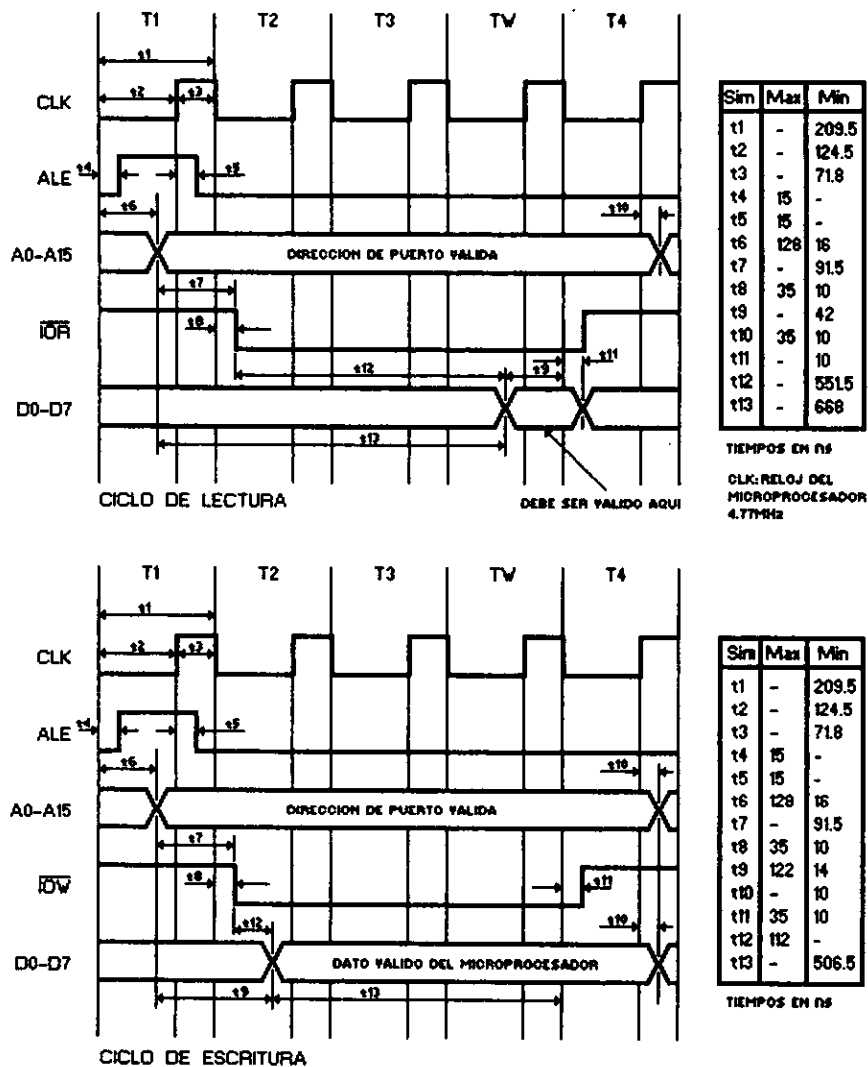


Figura III.3

Para un ciclo de lectura el dato debe de ser válido después de 551.5 ns después de que -IOR es válido y permanecer así por lo menos 52 ns. Si el periférico es demasiado lento para tener el dato listo después de los 551.5 ns se debe insertar un ciclo de espera a través de IOCHRDY. Para un ciclo de escritura el dato debe ser leído por el periférico antes de 506.5 ns desde que -IOW se hace válido. Si el periférico es demasiado lento para esto es necesario insertar un ciclo de espera a través de IOCHRDY.



# APÉNDICE IV

## ESPECIFICACIONES TÉCNICAS

### RETARDOS TÍPICOS Y MÁXIMOS DE ALGUNOS COMPONENTES DIGITALES

COMPONENTE	SÍMBOLO	PARÁMETRO	TÍPICO	MÁXIMO
74LS02	tPLH	ENTRADA A SALIDA	10ns	15ns
	tPHL		10ns	15ns
74LS08	tPLH	ENTRADA A SALIDA	8ns	15ns
	tPHL		10ns	20ns
74LS32	tPLH	ENTRADA A SALIDA	14ns	22ns
	tPHL		14ns	22ns
74LS157	tPLH	DATO A SALIDA	9ns	14ns
	tPHL		9ns	14ns
	tPLH	SELECT A SALIDA	15ns	23ns
	tPHL		18ns	27ns
74LS161	tPLH	RELOJ A Q	13ns	24ns
	tPHL		18ns	27ns
	tPLH	-MR a Q	20ns	28ns
	tPLH	DATO A SALIDA	12ns	18ns
74LS244	tPHL		12ns	18ns
	tPZH	HABILITACIÓN DE LA SALIDA A NIVEL ALTO	15ns	23ns
	tPZL	HABILITACIÓN DE LA SALIDA A NIVEL BAJO	20ns	30ns
	tPLZ	DESHABILITACIÓN DE LA SALIDA DESDE NIVEL ALTO	15ns	25ns

	tPHZ	DESHABILITACIÓN DE LA SALIDA DESDE NIVEL BAJO	10ns	18ns
--	------	---	------	------

tPLH: Retardo del cambio de bajo a alto

tPHL: Retardo del cambio de alto a bajo

-MR: Para el 74LS161 entrada MASTER RESET (activa en bajo).



# ADC0881

## 8-Bit 20 MSPS Flash A/D Converter

### General Description

The ADC0881 is a monolithic flash Analog-to-Digital converter capable of converting a video signal into a stream of 8-bit digital words at 20 MegaSamples Per Second (MSPS). Since the ADC0881 is a flash converter, a sample-and-hold circuit is not required.

The ADC0881 consists of 255 clocked latching comparators, precision resistive divider, encoding logic, and an output register. A single convert clock controls the conversion operation. The unit can be configured to give either true or inverted outputs, in binary or offset two's complement coding. All digital I/O is TTL compatible.

### Key Specifications

■ Resolution	8 bits
■ Conversion rate	DC to 20 MSPS (min)
■ Full power bandwidth	7 MHz (min)
■ Small signal bandwidth (-3 dB)	60 MHz (min)
■ Linearity error	½ LSB (max)
■ Analog input range	+3 to +5V
■ Differential gain	1%
■ Differential phase	0.5°
■ Power dissipation (V <sub>CC</sub> = +5V)	600 mW
■ Power supply	+5V

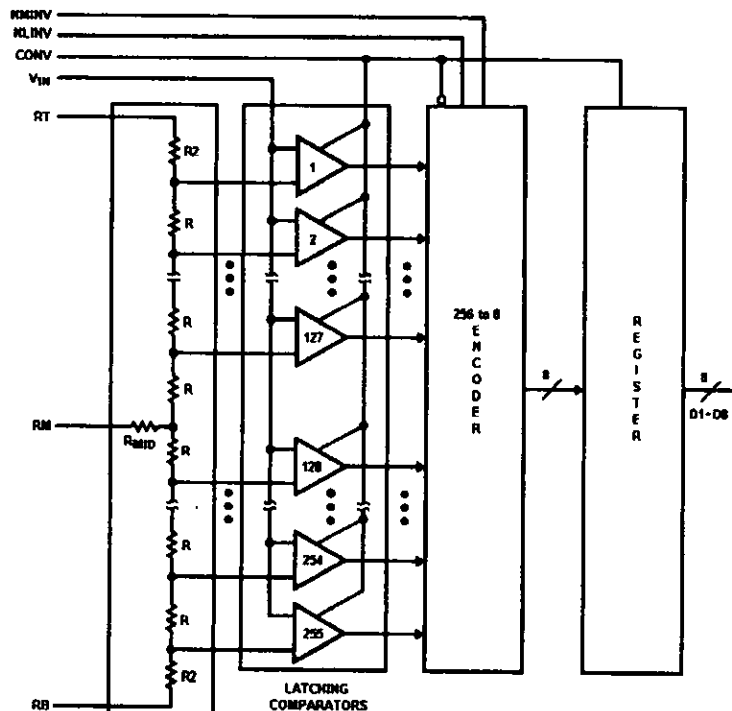
### Applications

- Video digitizing
- Medical imaging
- High energy physics
- Digital television
- Radar
- High speed data links

### Features

- Drop-in replacement for TDC1058
- Pin for pin replacement for CXA1086P and ADC-304
- No sample-and-hold circuit required
- Selectable data format
- Available in plastic DIP, CERDIP and PLCC
- Single supply operation

### Block Diagram



TL/H/11082-1

**Absolute Maximum Ratings** (Notes 1 & 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V <sub>CCA</sub> , V <sub>CCD</sub> )	-0.5V to +7.0V
V <sub>CCA</sub> -V <sub>CCD</sub>	±0.5V
AGND-DGND	±0.5V
Voltage at Logic Control Inputs (CONV, NM <sub>INV</sub> , NL <sub>INV</sub> ) (Note 3)	-0.5V to (V <sub>CC</sub> + 0.5V)
Voltage at Digital Outputs (D1-D8) (Note 3)	-0.5V to (V <sub>CC</sub> + 0.5V)
Voltage at Analog Inputs (V <sub>IN</sub> , V <sub>RT</sub> , V <sub>RB</sub> ) (Note 3)	-0.5V to (V <sub>CC</sub> + 0.5V)
V <sub>RT</sub> -V <sub>RB</sub>	±2.2V
Applied Current at Digital Outputs (Note 4)	±50 mA
Input Current at Logic Control Inputs (Note 4)	±50 mA
Input Current for V <sub>IN</sub> , V <sub>RT</sub> , V <sub>RB</sub> (Note 4)	±100 mA
Power Dissipation at T <sub>A</sub> = 25°C (Note 5)	500V
ESD Rating	500V

Storage Temperature Range	-65°C to +150°C
Soldering Temperature	
All Packages (10 seconds)	300°C

**Operating Ratings** (Notes 1 & 2)

Temperature Range	T <sub>MIN</sub> ≤ T <sub>A</sub> ≤ T <sub>MAX</sub>
ADC0881CCJ/TDC1058B6C	0°C ≤ T <sub>A</sub> ≤ +70°C
ADC0881CCN/TDC1058N6C	0°C ≤ T <sub>A</sub> ≤ +70°C
ADC0881CCV/TDC1058R3C	0°C ≤ T <sub>A</sub> ≤ +70°C
Supply Voltages (V <sub>CCA</sub> = V <sub>CCD</sub> = V <sub>CC</sub> )	4.75V to 5.25V
AGND-DGND Voltage	±0.1V
V <sub>RT</sub> Most Positive Reference Voltage (Note 6)	5.0V to (V <sub>CCA</sub> + 0.1V)
V <sub>RB</sub> Most Negative Reference Voltage (Note 6)	3.0V to 2.65V
V <sub>RT</sub> -V <sub>RB</sub>	1.8V to 2.2V
Analog Input Voltage (V <sub>IN</sub> )	V <sub>RT</sub> to V <sub>RB</sub>
t <sub>PWL</sub>	≥ 19 ns
t <sub>PWH</sub>	≥ 27 ns

**Converter Electrical Characteristics** The following specifications apply for V<sub>CCA</sub> = V<sub>CCD</sub> = V<sub>CC</sub> = +5.25V, V<sub>RB</sub> = 3.0V, V<sub>RT</sub> = 5.0V, V<sub>AGND</sub> = V<sub>DGND</sub> = GND, t<sub>PWL</sub> = 19 ns and t<sub>PWH</sub> = 27 ns unless otherwise specified. **Boldface limits apply for T<sub>A</sub> = T<sub>MIN</sub> to T<sub>MAX</sub>**; all other limits T<sub>A</sub> = 25°C.

Symbol	Parameter	Conditions	Typical	Limit	Units (Limit)
<b>STATIC CHARACTERISTICS</b>					
	Resolution			<b>8</b>	Bits(min)
	Integral Linearity Error			<b>0.2</b>	%FS(max)
	Differential Linearity Error			<b>0.2</b>	%FS(max)
	Code Size (Note 7)		100	<b>25</b> <b>178</b>	%LSB %LSB(min) %LSB(max)
	Offset Error, Top	V <sub>IN</sub> = V <sub>RT</sub>		±10	mV(max)
	Offset Error, Bottom	V <sub>IN</sub> = V <sub>RB</sub>		-16	mV(max)
	Offset Error Temperature Coefficient			±20	µV/°C(max)
I <sub>REF</sub>	Reference Current			<b>30</b>	mA(max)
R <sub>REF</sub>	Total Reference Resistance			<b>67</b>	Ω(min)
V <sub>IN</sub>	Analog Input Voltage Range			V <sub>RB</sub> V <sub>RT</sub>	V(min) V(max)
I <sub>IN</sub>	Analog Input Constant Bias Current			<b>250</b>	µA(max)
R <sub>IN</sub>	Analog Input Equivalent Resistance	V <sub>IN</sub> = V <sub>RB</sub>		<b>80</b>	kΩ(min)
C <sub>IN</sub>	Analog Input Capacitance	V <sub>IN</sub> = V <sub>RB</sub>		<b>80</b>	pF(max)
<b>DYNAMIC CHARACTERISTICS</b>					
S/(N+D)	Signal-to-Noise + Distortion Ratio Peak-to-Peak Signal/RMS Noise	V <sub>IN</sub> = 2.0 V <sub>p-p</sub> 10 MHz BW	f <sub>IN</sub> = 1.248 MHz	<b>54</b>	dB(min)
			f <sub>IN</sub> = 2.438 MHz	<b>53</b>	dB(min)
S/(N+D)	Signal-to-Noise + Distortion Ratio RMS Signal/RMS Noise	V <sub>IN</sub> = 2.0 V <sub>p-p</sub> 10 MHz BW	f <sub>IN</sub> = 1.248 MHz	<b>45</b>	dB(min)
			f <sub>IN</sub> = 2.438 MHz	<b>44</b>	dB(min)

**Converter Electrical Characteristics** The following specifications apply for  $V_{CCA} = V_{CCD} = V_{CC} = +5.25V$ ,  $V_{RB} = 3.0V$ ,  $V_{RT} = 5.0V$ ,  $V_{AGND} = V_{DGND} = GND$ ,  $t_{PWL} = 19\text{ ns}$  and  $t_{PWH} = 27\text{ ns}$  unless otherwise specified. **Boldface limits apply for  $T_A = T_{MIN}$  to  $T_{MAX}$ ; all other limits  $T_A = 25^\circ\text{C}$ .** (Continued)

Symbol	Parameter	Conditions	Typical	Limit	Units (Limit)
<b>DYNAMIC CHARACTERISTICS</b>					
BW	Bandwidth, Full Scale Input	$V_{IN} = 2.0 V_{D-P}$ , No Spurious or Missing Codes		<b>7</b>	MHz(min)
BW <sub>SS</sub>	-3 dB Small Signal Bandwidth	$V_{IN} = -20\text{ dB}$ , FS = $0.2 V_{D-P}$		<b>60</b>	MHz(min)
	Full Scale Transient Response			<b>70</b>	ns(min)
t <sub>STO</sub>	Sampling Time Offset	$V_{CC} = +4.75V$		<b>-2</b>	ns(min)
				<b>10</b>	ns(max)
	Aperture Jitter			<b>60</b>	ps <sub>rms</sub> (max)
	Differential Phase Error	$f_s = 4 \times \text{NTSC}$	0.5	<b>1.0</b>	Degree(max)
	Differential Gain Error	$f_s = 4 \times \text{NTSC}$	1.0	<b>2.0</b>	%(max)

**DC Electrical Characteristics** The following specifications apply for  $V_{CCA} = V_{CCD} = V_{CC} = +5.25V$ ,  $V_{RB} = 3.0V$ ,  $V_{RT} = 5.0V$ ,  $V_{AGND} = V_{DGND} = GND$ ,  $t_{PWL} = 19\text{ ns}$  and  $t_{PWH} = 27\text{ ns}$  unless otherwise specified. **Boldface limits apply for  $T_A = T_{MIN}$  to  $T_{MAX}$ ; all other limits  $T_A = 25^\circ\text{C}$ .**

Symbol	Parameter	Conditions	Typical (Note 7)	Limit	Units (Limit)
V <sub>IN(1)</sub>	Logical "1" Input Voltage			<b>2.4</b>	V(min)
V <sub>IN(0)</sub>	Logical "0" Input Voltage			<b>0.4</b>	V(max)
I <sub>IN(1)</sub>	Logical "1" Input Current	$V_{IN} = 2.4V$		<b>50</b>	$\mu\text{A}(\text{max})$
		$V_{IN} = V_{CC} = +5.25V$		<b>-200</b>	$\mu\text{A}(\text{min})$
I <sub>IN(0)</sub>	Logical "0" Input Current	$V_{IN} = 0.4V$		<b>1.0</b>	mA(max)
V <sub>OUT(1)</sub>	Logical "1" Output Voltage	$V_{CC} = 4.75V$ , I <sub>OUT</sub> = $-400\ \mu\text{A}$		<b>2.4</b>	V(min)
V <sub>OUT(0)</sub>	Logical "0" Output Voltage	$V_{CC} = 4.75V$ , I <sub>OUT</sub> = $4.0\text{ mA}$		<b>0.5</b>	V(max)
	Short-Circuit Output Current	Output High, One Pin to Ground, One Second Duration Max		<b>-40</b>	mA(max)
C <sub>I</sub>	Digital Input Capacitance	f = 1 MHz		15	pF(max)
I <sub>CC</sub>	V <sub>CCA</sub> and V <sub>CCD</sub> Supply Current	(Note 8)		<b>160</b>	mA(max)

**AC Electrical Characteristics** The following specifications apply for  $V_{CCA} = V_{CCD} = V_{CC} = +5.25V$ ,  $V_{RD} = 3.0V$ ,  $V_{RT} = 5.0V$ ,  $V_{AGND} = V_{DGND} = GND$ ,  $t_{PWL} = 19\text{ ns}$ , and  $t_{PWH} = 27\text{ ns}$  unless otherwise specified. **Boldface limits apply for  $T_A = T_{MIN}$  to  $T_{MAX}$ ; all other limits  $T_A = 25^\circ\text{C}$ .**

Symbol	Parameter	Conditions	Typical (Note 7)	Limit	Units (Limit)
$f_S$	Maximum Conversion Rate	$V_{CC} = +4.75V$		<b>20</b>	MSPS(min)
$t_{PWL}$	Convert (CONV) Pulse Width, Low			<b>19</b>	ns(min)
$t_{PWH}$	Convert (CONV) Pulse Width, High			<b>27</b>	ns(min)
$t_D$	Output Delay	$V_{CC} = +4.75V$ , Load 1, Figure 4		<b>38</b>	ns(max)
$t_{HO}$	Output Hold Time	Load 1, Figure 4		<b>5</b>	ns(min)

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is functional, but do not guarantee specific performance limits. For guaranteed specifications and test conditions, see the Electrical Characteristics. The guaranteed specifications apply only for the test conditions listed. Some performance characteristics may degrade when the device is not operated under the listed test conditions.

Note 2: All voltages are measured with respect to AGND and DGND, unless otherwise specified.

Note 3: Applied voltage must be current-limited to the specified range.

Note 4: Forcing voltage must be limited to specified range.

Note 5: The maximum power dissipation must be derated at elevated temperatures and is dictated by  $T_{Jmax}$  (maximum junction temperature),  $\theta_{JA}$  (package junction to ambient thermal resistance),  $\theta_{JC}$  (package junction to case thermal resistance), and  $T_A$  (ambient temperature). The maximum allowable power dissipation at any temperature is  $P_{Dmax} = (T_{Jmax} - T_A)/\theta_{JA}$  or the number given in the Absolute Maximum Ratings, whichever is lower. For this device,  $T_{Jmax} = 175^\circ\text{C}$ , and the typical thermal resistances ( $\theta_{JA}$  and  $\theta_{JC}$ ) of the ADC0881 follow:

Suffixes	Package Number	$\theta_{JA}$ ( $^\circ\text{C}/\text{W}$ )	$\theta_{JC}$ ( $^\circ\text{C}/\text{W}$ )
CCJ	J28A	50	12
CCN	N28A	45	17
CCV	V28A	65	14

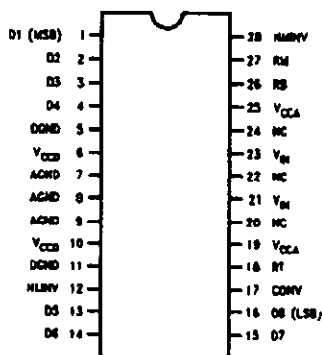
Note 6:  $V_{R1}$  must be more positive than  $V_{R2}$ , and the voltage reference differential must be within the specified range.

Note 7: Code size is the size of the individual codes, from code transition to code transition. It is often expressed as a percentage of the ideal code size. The ideal code size is given by:  $\text{Input Voltage Range}/2^N$ . Where N is the number of bits of resolution of the A/D converter.

Note 8: Worst case, all digital inputs and outputs LOW.

### Connection Diagrams

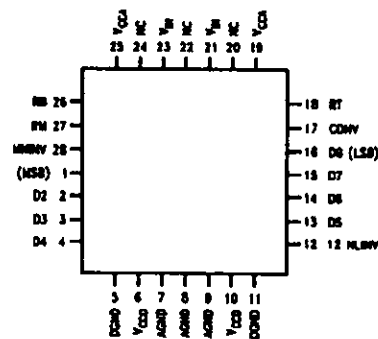
28-Lead CERDIP—J Package  
28-Lead Plastic DIP—N Package



Top View

TL/H/11082-2

28-Lead Plastic Chip Carrier—V Package



Top View

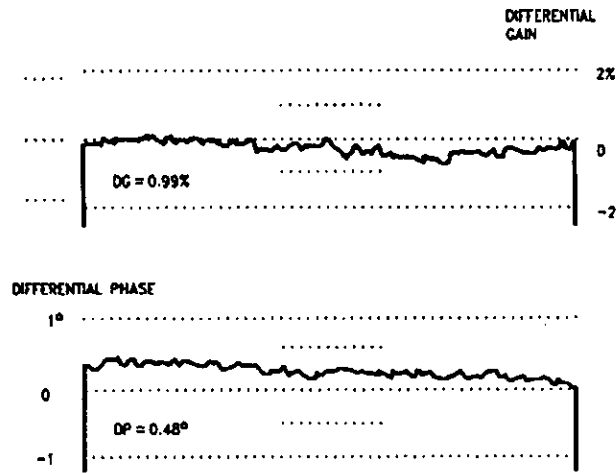
TL/H/11082-3

### Ordering Information

Commercial ( $0^\circ\text{C} \leq T_A \leq 70^\circ\text{C}$ )	Package
ADC0881CCN/TDC1058N8C	N28B, 28-Pin Plastic
ADC0881CCJ/TDC1058B8C	J28A, 28-Pin CERDIP
ADC0881CCV/TDC1058R3C	V28A, 28-Lead Plastic Chip Carrier

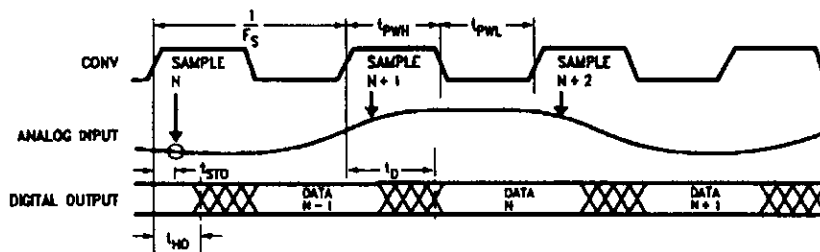
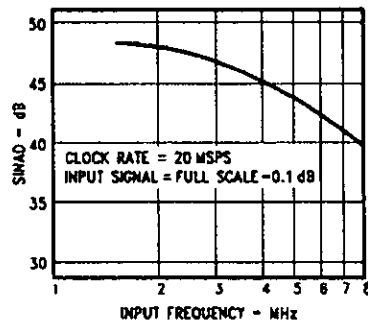
**Typical Performance Characteristics** ( $T_A = 25^\circ\text{C}$ ,  $V_{CC} = +5\text{V}$ )

**Differential Phase and Gain**



Convert Frequency = 14.3181800 MHz  
 Analog Input = 3.57954550 MHz

**SINAD (Signal to Noise + Distortion Ratio) vs Input Frequency**



**FIGURE 1. Timing Diagram**

TL/H/11082-6

**Typical Performance Characteristics** (Continued)

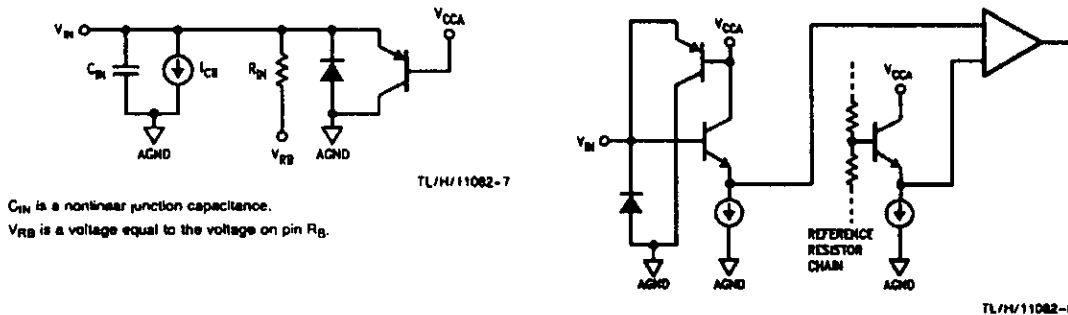
**TABLE I. Output Coding Table**

Input Voltage	Binary		Offset Two's Complement	
	True	Inverted	True	Inverted
	NMINV = HIGH NLINV = HIGH	NMINV = LOW NLINV = LOW	NMINV = LOW NLINV = HIGH	NMINV = HIGH NLINV = LOW
5.0000V	0000 0000	1111 1111	1000 0000	0111 1111
4.9922V	0000 0001	1111 1110	1000 0001	0111 1110
⋮	⋮	⋮	⋮	⋮
4.0078V	0111 1111	1000 0000	1111 1111	0000 0000
4.0000V	1000 0000	0111 1111	0000 0000	1111 1111
3.9922V	1000 0001	0111 1110	0000 0001	1111 1110
⋮	⋮	⋮	⋮	⋮
3.0156V	1111 1110	0000 0001	0111 1110	1000 0001
3.0078V	1111 1111	0000 0000	0111 1111	1000 0000

Note 1: NMINV and NLINV are to be considered DC controls, they may be tied to V<sub>CC</sub> for a logic "1" or to ground for a logic "0".

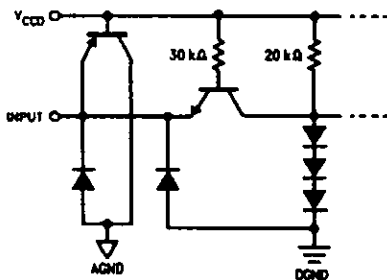
Note 2: Voltages are code midpoints.

**Simplified Input and Output Circuits**



$C_{IN}$  is a nonlinear junction capacitance.  
 $V_{RB}$  is a voltage equal to the voltage on pin  $R_B$ .

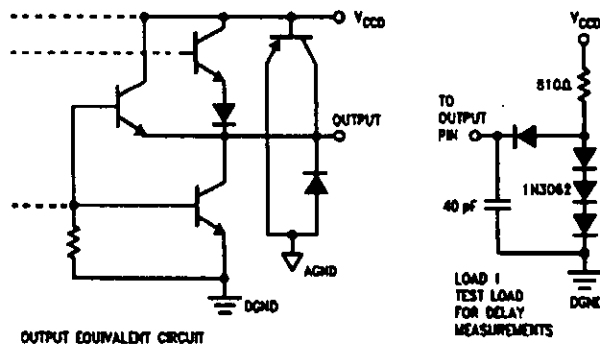
**FIGURE 2. Simplified Analog Input Circuits**



**FIGURE 3. Simplified Digital Convert Input Circuit**



**Simplified Input and Output Circuits (Continued)**



TL/H/11082-10

**FIGURE 4. Simplified Digital Output Circuit**

**Pin Descriptions**

- V<sub>CCD</sub> (6,10)** These are the digital power supply pins. Normally, +5 V<sub>DC</sub> should be applied and bypassed to digital ground with a 0.1 μF ceramic capacitor in parallel with a 10 μF tantalum capacitor.
- V<sub>CCA</sub> (19,25)** These are the analog power supply pins. Normally, +5 V<sub>DC</sub> should be applied and bypassed to analog ground with a 0.1 μF ceramic capacitor in parallel with a 10 μF tantalum capacitor.
- AGND (7,8,9)** Analog ground pins.
- DGND (5,11)** Digital ground pins.
- RT (18)** This pin connects to the top of the reference resistor. Normally this pin is connected to +5 V<sub>DC</sub> reference and should be bypassed to AGND with a 0.1 μF ceramic capacitor in parallel with a 10 μF tantalum capacitor.
- RM (27)** This pin is connected to the middle of the reference resistor. A voltage may be applied to this pin to trim the converter's integral linearity (see text).
- RB (26)** This pin is connected to the bottom of the reference resistor. Normally +3V is applied to this pin and should be bypassed to analog ground with a 0.1 μF ceramic capacitor in parallel with a 10 μF tantalum capacitor.
- V<sub>IN</sub> (21,23)** These are the analog signal input pins, the two pins are internally connected. The input signal range is from +3V to +5V.
- CONV (17)** A TTL convert signal is applied to this pin. A conversion is initiated at the rising edge of the signal.
- NMINV (28)** This is the Not Most Significant Bit Invert pin. A TTL signal at this pin controls the format of the output data (see Table I).
- NLINV (12)** This is the Not Least Significant Bit Invert pin. A TTL signal at this pin controls the format of the output data (see Table I).
- D1-D8** These are the digital output pins. D1 outputs the MSB data while D8 outputs the LSB data.
- NC** No connection.

**Application Hints**

**OPERATION**

The ADC0881 has three functional sections: a resistor/comparator array, encoding logic, and output registers. The resistor/comparator array compares the input signal with 255 reference voltages to produce an N-of-255 code (or thermometer code, since all the comparators whose reference is more negative than the input signal will be in logic "1" state and all those whose reference is more positive will be in logic "0" state). The encoding logic converts the N-of-255 code into the user's choice of coding. The output register holds the output constant between updates.

**REFERENCE**

The ADC0881 converts analog signals in the range  $V_{RB} \leq V_{IN} \leq V_{RT}$  into digital form. Nominally,  $V_{RB}$  is set to 3V and  $V_{RT}$  is set to 5V. However, the specifications of the ADC0881 are guaranteed as long as the following three reference operating conditions are met:

1. the voltage applied across the reference resistor chain ( $V_{RT} - V_{RB}$ ) is within the range of 1.8V to 2.2V,
2.  $V_{RT} \leq (V_{CCA} + 0.1V)$
3.  $V_{RB} \geq 2.65V$ .

Therefore, if the supply voltage is expected to drop below 4.9V, the reference voltages should be lowered accordingly. For instance, if the system design allows the supply voltage to drop to the minimum recommended value of 4.75V,  $V_{RT}$  should be set to 4.75V and  $V_{RB}$  should be set to 2.75V. At worst case,  $V_{RT}$  may be set at 4.85V and  $V_{RB}$  set at 2.85V. This assumes a 2 V<sub>p-p</sub> maximum input signal that spans from  $V_{IN} = V_{RB}$  to  $V_{IN} = V_{RT}$ . These reference voltages will allow the ADC0881 to give fully guaranteed performance over the full supply range. See the "Electrical Characteristics" table for further information.

Linearity is guaranteed with no adjustment; however, a mid-point tap,  $R_{M}$ , allows trimming of converter integral linearity as well as the creation of a nonlinear transfer function. Note that ADC0881's integral nonlinearity is  $\pm \frac{1}{2}$  LSB maximum. If the maximum non-linearity occurs at midscale then the circuit of Figure 5 will allow the user to null out the linearity error at midscale. This adjustment may improve the overall integral linearity of the converter to less than  $\pm \frac{1}{2}$  LSB.

### Application Hints (Continued)

The characteristic impedance seen at this node is approximately  $220\Omega$  and should be driven from a low-impedance source. Note that any load applied to this node will affect linearity, and any noise introduced at this point will degrade the overall quantization Signal to Noise Ratio. Due to the slight variation in the reference current with clock and input signals,  $R_T$  and  $R_B$  should be low-impedance-to-ground points. For circuits in which the reference is not varied, a bypass capacitor ( $0.01\ \mu\text{F}$  to  $0.1\ \mu\text{F}$ ) to ground is recommended. If the reference inputs are exercised dynamically (as in an automatic gain control circuit) a low-impedance reference source is required. The reference voltages may be varied dynamically at up to 5 MHz; however, device performance is specified with fixed reference voltages as defined in the "Electrical Characteristics" tables.

#### ANALOG INPUT AND SOURCE IMPEDANCE CONSIDERATIONS

For precise quantization, the ADC0881 uses latching comparators. For optimum overall system performance the source impedance of the driving circuit must be less than  $25\Omega$ . If the input signal is between the  $V_{RT}$  and  $V_{RB}$  references, the output will be a binary number from 0 to 255. When a signal outside the recommended input voltage range is applied, the output will remain at either full-scale value. The input signal will not damage the ADC0881 if it remains within the range specified in the "Absolute Maximum Ratings" table. Both analog input pins ( $V_{IN}$ ) are connected together internally and therefore either one or both may be used.

#### CONVERT

The ADC0881 requires an external convert (CONV) signal. Because the ADC0881 is a flash converter it does not require a track-and-hold circuit. A sample is taken (the outputs of the comparators are latched) within  $t_{STO}$  (Sampling Time Offset) after a rising edge on the CONV pin. The result is encoded on the falling edge, and then transferred to the output registers on the next rising edge. The output becomes valid  $t_D$  (Output Delay Time) after the rising edge of CONV and remains valid for at least  $t_{HO}$  (Output Hold Time) after the rising edge of CONV. Therefore, the value of sample N becomes valid  $t_D$  after the rising edge of clock N+1 and remains valid until  $t_{HO}$  after the rising edge of clock N+2. (See Figure 1, Timing Diagram.)

#### OUTPUT FORMAT CONTROL

Two output format control pins, NMINV and NLINV, are provided. These controls are for DC (i.e., steady state) use. They permit the output coding to be either straight binary or offset two's complement, in either true or inverted sense, according to the "Output Coding Table". (See Table 1.) These active low pins may be tied to  $V_{CC}$  (through a  $4.7k$  resistor) for a logic "1" or DGND for a logic "0".

#### OUTPUTS

The outputs of the ADC0881 are TTL compatible, capable of driving four low-power Schottky TTL (54/74 LS) loads or the equivalent. The outputs hold the previous data for a minimum of  $t_{HO}$  after the rising edge of the convert signal.

#### POWER SUPPLIES

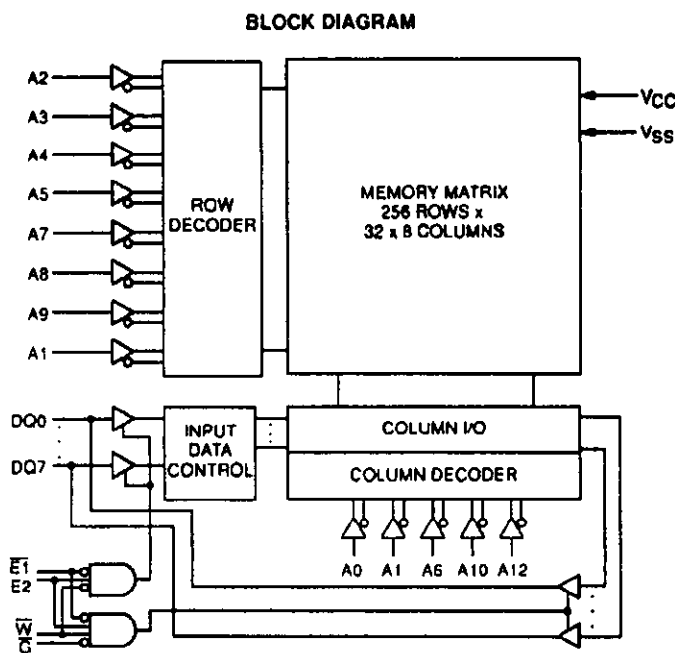
The ADC0881 operates from a single +5V supply voltage. All power and ground pins must be connected.  $V_{CCA}$  and  $V_{CCD}$  should be bypassed with a  $0.1\ \mu\text{F}$  ceramic capacitor in parallel with a  $10\ \mu\text{F}$  tantalum capacitor to the analog and digital grounds respectively. A ferrite bead may be used to provide high frequency isolation between the two supply pins.

## 8K x 8 Bit Fast Static RAM

The MCM6264 is fabricated using Motorola's high-performance silicon-gate CMOS technology. Static design eliminates the need for external clocks or timing strobes, while CMOS circuitry reduces power consumption and provides for greater reliability.

This device meets JEDEC standards for functionality and pinout, and is available in plastic dual-in-line and plastic small-outline J-leaded packages.

- Single 5 V  $\pm$ 10% Power Supply
- Fully Static — No Clock or Timing Strokes Necessary
- Fast Access Times: 15, 20, 25 and 35 ns
- Equal Address and Chip Enable Access Times
- Output Enable ( $\bar{G}$ ) Feature for Increased System Flexibility and to Eliminate Bus Contention Problems
- Low Power Operation: 110–140 mA Maximum ac
- Fully TTL-Compatible — Three-State Output



## MCM6264



P PACKAGE  
300-MIL PLASTIC  
CASE 710B-01



NJ PACKAGE  
300-MIL SOJ  
CASE 810B-03

### PIN ASSIGNMENT

NC	1	28	VCC
A12	2	27	W
A7	3	26	E2
A6	4	25	A8
A5	5	24	A9
A4	6	23	A11
A3	7	22	$\bar{G}$
A2	8	21	A10
A1	9	20	E1
A0	10	19	DQ7
DQ0	11	18	DQ6
DQ1	12	17	DQ5
DQ2	13	16	DQ4
VSS	14	15	DQ3

### PIN NAMES

A0–A12	Address Input
DQ0–DQ7	Data Input/Data Output
W	Write Enable
$\bar{G}$	Output Enable
E1, E2	Chip Enable
NC	No Connection
VCC	Power Supply (+5 V)
VSS	Ground

TRUTH TABLE (X = don't care)

E1	E2	Ø	W	Mode	V <sub>CC</sub> Current	Output	Cycle
H	X	X	X	Not Selected	I <sub>SB1</sub> , I <sub>SB2</sub>	High-Z	-
X	L	X	X	Not Selected	I <sub>SB1</sub> , I <sub>SB2</sub>	High-Z	-
L	H	H	H	Output Disabled	I <sub>CCA</sub>	High-Z	-
L	H	L	H	Read	I <sub>CCA</sub>	O <sub>out</sub>	Read Cycle
L	H	X	L	Write	I <sub>CCA</sub>	High-Z	Write Cycle

The device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit.

The CMOS memory circuit has been designed to meet the dc and ac specifications shown in the tables, after thermal equilibrium has been established. The circuit is in a test socket or mounted on a printed circuit board and transverse air flow of at least 500 linear feet per minute is maintained.

ABSOLUTE MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Power Supply Voltage Relative to V <sub>SS</sub>	V <sub>CC</sub>	-0.5 to +7.0 V	V
Voltage Relative to V <sub>SS</sub> For Any Pin Except V <sub>CC</sub>	V <sub>in</sub> , V <sub>out</sub>	-0.5 to V <sub>CC</sub> + 0.5	V
Output Current (per IO)	I <sub>out</sub>	± 30	mA
Power Dissipation	P <sub>D</sub>	1.0	W
Temperature Under Bias	T <sub>total</sub>	-10 to +85	°C
Operating Temperature	T <sub>A</sub>	0 to +70	°C
Storage Temperature—Plastic	T <sub>stg</sub>	-55 to +125	°C

NOTE: Permanent device damage may occur if ABSOLUTE MAXIMUM RATINGS are exceeded. Functional operation should be restricted to RECOMMENDED OPERATING CONDITIONS. Exposure to higher than recommended voltages for extended periods of time could affect device reliability.

DC OPERATING CONDITIONS AND CHARACTERISTICS

(V<sub>CC</sub> = 5.0 V ± 10%, T<sub>A</sub> = 0 to 70°C, Unless Otherwise Noted)

RECOMMENDED OPERATING CONDITIONS

Parameter	Symbol	Min	Typ	Max	Unit
Supply Voltage (Operating Voltage Range)	V <sub>CC</sub>	4.5	5.0	5.5	V
Input High Voltage	V <sub>IH</sub>	2.2	—	V <sub>CC</sub> + 0.3*	V
Input Low Voltage	V <sub>IL</sub>	-0.5**	—	0.8	V

\*V<sub>IH</sub> (max) = V<sub>CC</sub> + 0.3 V dc; V<sub>IH</sub> (max) = V<sub>CC</sub> + 2.0 V ac (pulse width ≤ 20 ns)  
 \*\*V<sub>IL</sub> (min) = -0.5 V dc; V<sub>IL</sub> (min) = -2.0 V ac (pulse width ≤ 20 ns)

DC CHARACTERISTICS

Parameter	Symbol	Min	Max	Unit
Input Leakage Current (All inputs, V <sub>I1</sub> = 0 to V <sub>CC</sub> )	I <sub>log(I)</sub>	—	± 1	µA
Output Leakage Current (E = V <sub>IH</sub> or G = V <sub>IH</sub> , V <sub>out</sub> = 0 to V <sub>CC</sub> )	I <sub>log(O)</sub>	—	± 1	µA
Output Low Voltage (I <sub>OL</sub> = 8.0 mA)	V <sub>OL</sub>	—	0.4	V
Output High Voltage (I <sub>OH</sub> = -4.0 mA)	V <sub>OH</sub>	2.4	—	V

POWER SUPPLY CURRENTS

Parameter	Symbol	-15	-20	-25	-35	Unit
AC Active Supply Current (I <sub>out</sub> = 0 mA, V <sub>CC</sub> = Max, I = I <sub>max</sub> )	I <sub>CCA</sub>	140	130	120	110	mA
AC Standby Current (E = V <sub>IH</sub> or E2 = V <sub>IL</sub> , V <sub>CC</sub> = Max, I = I <sub>max</sub> )	I <sub>SB1</sub>	40	35	30	30	mA
Standby Current (E1 ≥ V <sub>CC</sub> - 0.2 V or E2 ≤ V <sub>SS</sub> + 0.2 V, V <sub>I1</sub> ≤ V <sub>SS</sub> + 0.2 V, or ≥ V <sub>CC</sub> - 0.2 V)	I <sub>SB2</sub>	20	20	20	20	mA

CAPACITANCE (f = 1 MHz, dV = 3 V, T<sub>A</sub> = 25°C, Periodically sampled rather than 100% tested)

Characteristic	Symbol	Max	Unit
Address and Data Input Capacitance	C <sub>in</sub>	6	pF
Control Pin Input Capacitance (E, E2, $\bar{G}$ , $\bar{W}$ )	C <sub>in</sub>	6	pF
Output Capacitance	C <sub>out</sub>	7	pF

**AC OPERATING CONDITIONS AND CHARACTERISTICS**  
(V<sub>CC</sub> = 5 V ± 10%, T<sub>A</sub> = 0 to + 70°C, Unless Otherwise Noted)

Input Timing Measurement Reference Level ..... 1.5 V      Output Timing Measurement Reference Level ..... 1.5 V  
 Input Pulse Levels ..... 0 to 3 V      Output Load ..... Figure 1A Unless Otherwise Noted  
 Input Rise/Fall Time ..... 5 ns

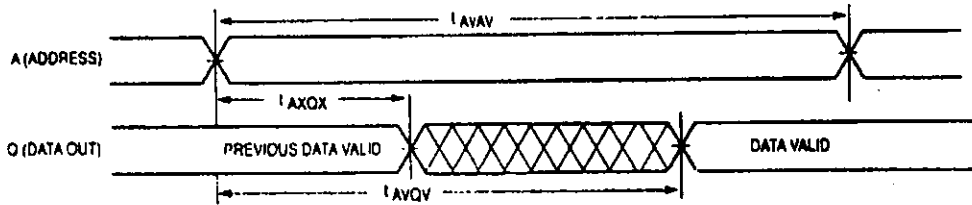
READ CYCLE (See Notes 1 and 2)

Parameter	Symbol		- 15		- 20		- 25		- 35		Unit	Notes
	Std	AR	Min	Max	Min	Max	Min	Max	Min	Max		
Read Cycle Time	t <sub>RDV</sub>	t <sub>RC</sub>	15	—	20	—	25	—	35	—	ns	3
Address Access Time	t <sub>AVD</sub>	t <sub>AA</sub>	—	15	—	20	—	25	—	35	ns	
Enable Access Time	t <sub>ELD</sub>	t <sub>ACS</sub>	—	15	—	20	—	25	—	35	ns	4
Output Enable Access Time	t <sub>GLO</sub>	t <sub>OE</sub>	—	8	—	10	—	11	—	12	ns	
Output Hold from Address Change	t <sub>AOX</sub>	t <sub>OH</sub>	4	—	4	—	4	—	4	—	ns	
Enable Low to Output Active	t <sub>ELO</sub>	t <sub>OLZ</sub>	4	—	4	—	4	—	4	—	ns	5,6,7
Output Enable Low to Output Active	t <sub>GLO</sub>	t <sub>OLZ</sub>	0	—	0	—	0	—	0	—	ns	5,6,7
Enable High to Output High-Z	t <sub>EHQ</sub>	t <sub>CHZ</sub>	0	8	0	9	0	10	0	11	ns	5,6,7
Output Enable High to Output High-Z	t <sub>GHQ</sub>	t <sub>CHZ</sub>	0	7	0	8	0	9	0	10	ns	5,6,7
Power Up Time	t <sub>ELCO</sub>	t <sub>PU</sub>	0	—	0	—	0	—	0	—	ns	
Power Down Time	t <sub>EMCO</sub>	t <sub>PO</sub>	—	15	—	20	—	25	—	35	ns	

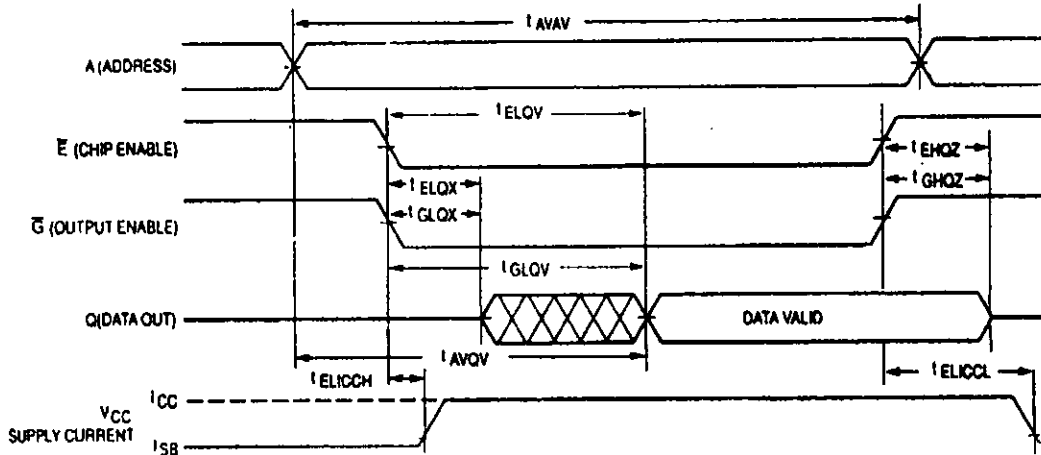
NOTES:

1.  $\bar{W}$  is high for read cycle.
2. E1 and E2 are represented by E in this data sheet. E2 is of opposite polarity to E.
3. All timings are referenced from the last valid address to the first transitioning address.
4. Addresses valid prior to or coincident with E going low.
5. At any given voltage and temperature, t<sub>EHQ</sub> max < t<sub>ELO</sub> min, and t<sub>GHQ</sub> max < t<sub>GLO</sub> min, both for a given device and from device to device.
6. Transition is measured ± 500 mV from steady-state voltage with load of Figure 1B.
7. This parameter is sampled and not 100% tested.
8. Device is continuously selected (E1 = V<sub>IL</sub>, E2 = V<sub>IH</sub>,  $\bar{G}$  = V<sub>IL</sub>).

**READ CYCLE 1 (See Note 8)**



**READ CYCLE 2 (See Note 4)**



**AC TEST LOADS**

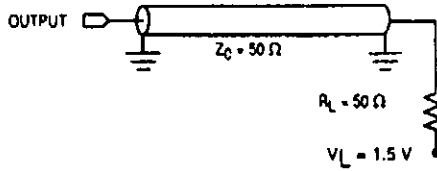


Figure 1A

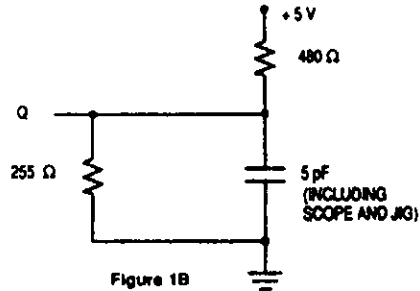
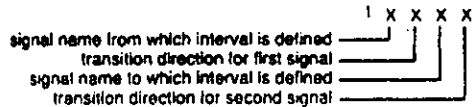


Figure 1B

**TIMING PARAMETER ABBREVIATIONS**



- The transition definitions used in this data sheet are
- H = transition to high
- L = transition to low
- V = transition to valid
- X = transition to invalid or don't care
- Z = transition to off (high impedance)

**TIMING LIMITS**

The table of timing values shows either a minimum or a maximum limit for each parameter. Input requirements are specified from the external system point of view. Thus, address setup time is shown as a minimum since the system must supply at least that much time (even though most devices do not require it). On the other hand, responses from the memory are specified from the device point of view. Thus, the access time is shown as a maximum since the device output will be valid no later than that time.

**WRITE CYCLE 1 (W Controlled)** (See Notes 1, 2, and 3)

Parameter	Symbol		- 15		- 20		- 25		- 35		Unit	Notes
	Std	Alt	Min	Max	Min	Max	Min	Max	Min	Max		
Write Cycle Time	t <sub>WAV</sub>	t <sub>WC</sub>	15	—	20	—	25	—	35	—	ns	4
Address Setup Time	t <sub>WAL</sub>	t <sub>AS</sub>	0	—	0	—	0	—	0	—	ns	
Address Valid to End of Write	t <sub>AWH</sub>	t <sub>AW</sub>	12	—	15	—	17	—	20	—	ns	
Write Pulse Width	t <sub>WLNH</sub> t <sub>WLEH</sub>	t <sub>WP</sub>	12	—	15	—	17	—	20	—	ns	
Write Pulse Width, High (Output Enable devices)	t <sub>WLNH</sub> t <sub>WLEH</sub>	t <sub>WP</sub>	10	—	12	—	15	—	17	—	ns	5
Data Valid to End of Write	t <sub>DVWH</sub>	t <sub>DW</sub>	7	—	8	—	10	—	12	—	ns	
Data Hold Time	t <sub>WDH</sub>	t <sub>DH</sub>	0	—	0	—	0	—	0	—	ns	
Write Low to Output High-Z	t <sub>WLQZ</sub>	t <sub>WZ</sub>	0	7	0	8	0	10	0	12	ns	6,7,8
Write High to Output Active	t <sub>WHQX</sub>	t <sub>WH</sub>	4	—	4	—	4	—	4	—	ns	6,7,8
Write Recovery Time	t <sub>WHX</sub>	t <sub>WR</sub>	0	—	0	—	0	—	0	—	ns	

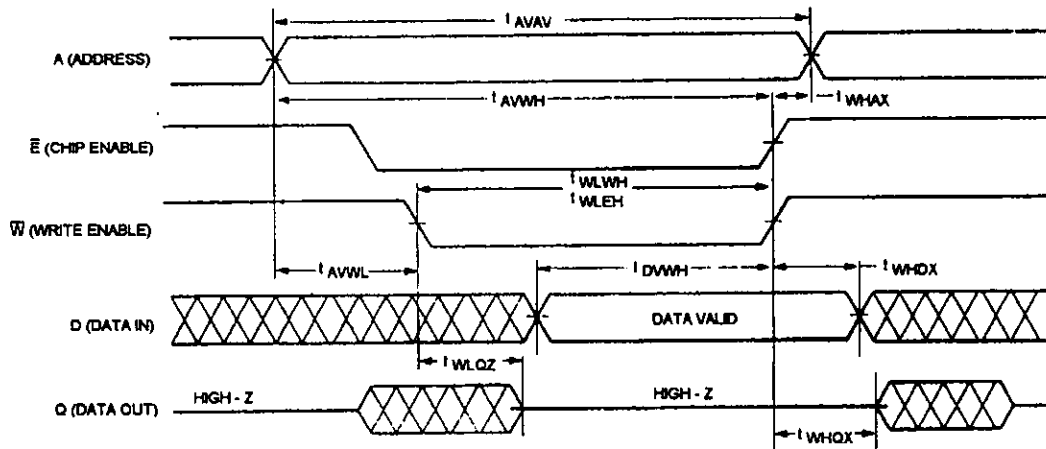
**WRITE CYCLE 2 (E Controlled)** (See Notes 1 and 2)

Parameter	Symbol		- 15		- 20		- 25		- 35		Unit	Notes
	Std	Alt	Min	Max	Min	Max	Min	Max	Min	Max		
Write Cycle Time	t <sub>WAV</sub>	t <sub>WC</sub>	15	—	20	—	25	—	35	—	ns	4
Address Setup Time	t <sub>WEL</sub>	t <sub>AS</sub>	0	—	0	—	0	—	0	—	ns	
Address Valid to End of Write	t <sub>WEH</sub>	t <sub>AW</sub>	12	—	15	—	20	—	25	—	ns	
Enable to End of Write	t <sub>ELEH</sub> t <sub>ELWH</sub>	t <sub>EW</sub>	10	—	12	—	15	—	25	—	ns	9,10
Data Valid to End of Write	t <sub>DVEH</sub>	t <sub>DW</sub>	7	—	8	—	10	—	15	—	ns	
Data Hold Time	t <sub>EDH</sub>	t <sub>DH</sub>	0	—	0	—	0	—	0	—	ns	
Write Recovery Time	t <sub>EHX</sub>	t <sub>WR</sub>	0	—	0	—	0	—	0	—	ns	

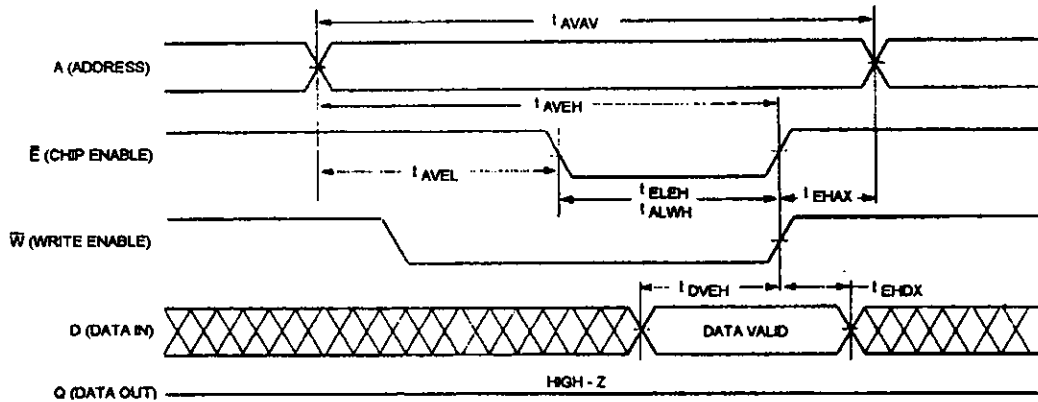
**NOTES:**

1. A write occurs during the overlap of E low and W low.
2. E1 and E2 are represented by E in this data sheet. E2 is of opposite polarity to E.
3. If  $\bar{G}$  goes low coincident with or after W goes low, the output will remain in a high-impedance state.
4. All timings are referenced from the last valid address to the first transitioning address.
5. If  $\bar{G} > V_{IH}$ , the output will remain in a high-impedance state.
6. At any given voltage and temperature, t<sub>WLQZ</sub> max < t<sub>WHQX</sub> min, both for a given device and from device to device.
7. Transition is measured ± 500 mV from steady-state voltage with load of Figure 1B.
8. This parameter is sampled and not 100% tested.
9. If E goes low coincident with or after W goes low, the output will remain in a high-impedance state.
10. If E goes high coincident with or before W goes high, the output will remain in a high-impedance state.

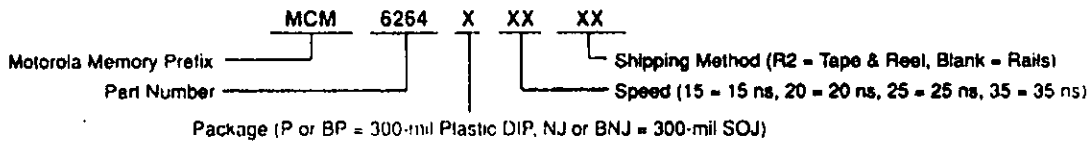
WRITE CYCLE 1 (See Notes 1, 2, and 3)



WRITE CYCLE 2 (See Notes 1 and 2)



ORDERING INFORMATION (Order by Full Part Number)



Full Part Numbers—	MCM6264P15	MCM6264NJ15	MCM6264NJ15R2
	MCM6264P20	MCM6264NJ20	MCM6264NJ20R2
	MCM6264BP25	MCM6264BNJ25	MCM6264BNJ25R2
	MCM6264BP35	MCM6264BNJ35	MCM6264BNJ35R2





# LF155/LF156/LF157 Series Monolithic JFET Input Operational Amplifiers

## General Description

These are the first monolithic JFET input operational amplifiers to incorporate well matched, high voltage JFETs on the same chip with standard bipolar transistors (BI-FET™ Technology). These amplifiers feature low input bias and offset currents/low offset voltage and offset voltage drift, coupled with offset adjust which does not degrade drift or common-mode rejection. The devices are also designed for high slew rate, wide bandwidth, extremely fast settling time, low voltage and current noise and a low 1/f noise corner.

## Advantages

- Replace expensive hybrid and module FET op amps
- Rugged JFETs allow blow-out free handling compared with MOSFET input devices
- Excellent for low noise applications using either high or low source impedance—very low 1/f corner
- Offset adjust does not degrade drift or common-mode rejection as in most monolithic amplifiers
- New output stage allows use of large capacitive loads (5,000 pF) without stability problems
- Internal compensation and large differential input voltage capability

## Applications

- Precision high speed integrators
- Fast D/A and A/D converters
- High impedance buffers
- Wideband, low noise, low drift amplifiers
- Logarithmic amplifiers

- Photocell amplifiers
- Sample and Hold circuits

## Common Features

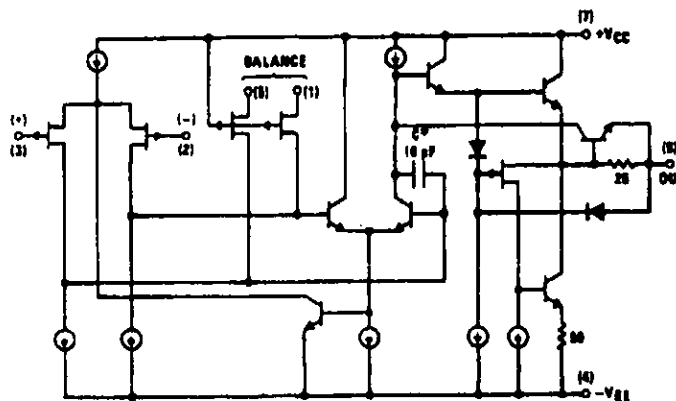
(LF155A, LF156A, LF157A)

- Low input bias current 30 pA
- Low Input Offset Current 3 pA
- High input impedance 10<sup>12</sup>Ω
- Low input offset voltage 1 mV
- Low input offset voltage temp. drift 3 μV/°C
- Low input noise current 0.01 pA/√Hz
- High common-mode rejection ratio 100 dB
- Large dc voltage gain 106 dB

## Uncommon Features

	LF155A	LF156A	LF157A (A <sub>v</sub> = 6)	Units
■ Extremely fast settling time to 0.01%	4	1.5	1.5	μs
■ Fast slew rate	5	12	50	V/μs
■ Wide gain bandwidth	2.5	5	20	MHz
■ Low input noise voltage	20	12	12	nV/√Hz

## Simplified Schematic



\*3 pF in LF157 series.

TL/11/6648-1

### Absolute Maximum Ratings

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/Distributors for availability and specifications.

(Note 8)

	LF155A/6A/7A	LF155/6/7	LF355B/6B/7B LF355/6/7	LF355A/6A/7A
Supply Voltage	±22V	±22V	±22V	±18V
Differential Input Voltage	±40V	±40V	±40V	±30V
Input Voltage Range (Note 2)	±20V	±20V	±20V	±16V
Output Short Circuit Duration	Continuous	Continuous	Continuous	Continuous
$T_{MAX}$				
H-Package	150°C	150°C	115°C	115°C
N-Package			100°C	100°C
M-Package			100°C	100°C
Power Dissipation at $T_A = 25^\circ\text{C}$ (Notes 1 and 9)				
H-Package (Still Air)	560 mW	560 mW	400 mW	400 mW
H-Package (400 LF/Min Air Flow)	1200 mW	1200 mW	1000 mW	1000 mW
N-Package			670 mW	670 mW
M-Package			380 mW	380 mW
Thermal Resistance (Typical) $\theta_{JA}$				
H-Package (Still Air)	180°C/W	180°C/W	160°C/W	160°C/W
H-Package (400 LF/Min Air Flow)	65°C/W	65°C/W	85°C/W	65°C/W
N-Package			130°C/W	130°C/W
M-Package			185°C/W	195°C/W
(Typical) $\theta_{JC}$				
H-Package	23°C/W	23°C/W	23°C/W	23°C/W
Storage Temperature Range	-65°C to +150°C	-65°C to +150°C	-65°C to +150°C	-65°C to +150°C
Soldering Information (Lead Temp.)				
Metal Can Package				
Soldering (10 sec.)	300°C	300°C	300°C	300°C
Dual-In-Line Package				
Soldering (10 sec.)		260°C	260°C	260°C
Small Outline Package				
Vapor Phase (60 sec.)			215°C	215°C
Infrared (15 sec.)			220°C	220°C
See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" for other methods of soldering surface mount devices.				
ESD tolerance (100 pF discharged through 1.5 k $\Omega$ )	1200V	1200V	1200V	1200V

### DC Electrical Characteristics (Note 3) $T_A = T_I = 25^\circ\text{C}$

Symbol	Parameter	Conditions	LF155A/6A/7A			LF355A/6A/7A			Units
			Min	Typ	Max	Min	Typ	Max	
$V_{OS}$	Input Offset Voltage	$R_B = 50\Omega$ , $T_A = 25^\circ\text{C}$ Over Temperature		1	2 2.5		1	2 2.3	mV mV
$\Delta V_{OS}/\Delta T$	Average TC of Input Offset Voltage	$R_B = 50\Omega$		3	5		3	5	$\mu\text{V}/^\circ\text{C}$
$\Delta TC/\Delta V_{OS}$	Change in Average TC with $V_{OS}$ Adjust	$R_B = 50\Omega$ , (Note 4)		0.5			0.5		$\mu\text{V}/^\circ\text{C}$ per mV
$I_{OS}$	Input Offset Current	$T_I = 25^\circ\text{C}$ , (Notes 3, 5) $T_I \leq T_{HIGH}$		3	10 10		3	10 1	$\mu\text{A}$ nA
$I_B$	Input Bias Current	$T_I = 25^\circ\text{C}$ , (Notes 3, 5) $T_I \leq T_{HIGH}$		30	50 25		30	50 5	$\mu\text{A}$ nA
$R_{IN}$	Input Resistance	$T_I = 25^\circ\text{C}$		10 <sup>12</sup>			10 <sup>12</sup>		$\Omega$
$A_{VOL}$	Large Signal Voltage Gain	$V_B = \pm 15\text{V}$ , $T_A = 25^\circ\text{C}$ $V_O = \pm 10\text{V}$ , $R_L = 2\text{k}$ Over Temperature	50 25	200		50 25	200		V/mV V/mV
$V_O$	Output Voltage Swing	$V_B = \pm 15\text{V}$ , $R_L = 10\text{k}$ $V_B = \pm 15\text{V}$ , $R_L = 2\text{k}$	$\pm 12$ $\pm 10$	$\pm 13$ $\pm 12$		$\pm 12$ $\pm 10$	$\pm 13$ $\pm 12$		V V

**DC Electrical Characteristics** (Note 3)  $T_A = T_J = 25^\circ\text{C}$  (Continued)

Symbol	Parameter	Conditions	LF155A/6A/7A			LF355A/6A/7A			Units
			Min	Typ	Max	Min	Typ	Max	
$V_{CM}$	Input Common-Mode Voltage Range	$V_B = \pm 15\text{V}$	$\pm 11$	+15.1 -12		$\pm 11$	+15.1 -12		V V
CMRR	Common-Mode Rejection Ratio		85	100		85	100		dB
PSRR	Supply Voltage Rejection Ratio	(Note 6)	85	100		85	100		dB

**AC Electrical Characteristics**  $T_A = T_J = 25^\circ\text{C}$ ,  $V_B = \pm 15\text{V}$

Symbol	Parameter	Conditions	LF155A/355A			LF156A/356A			LF157A/357A			Units
			Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
SR	Slew Rate	LF155A/6A; $A_V = 1$ , LF157A; $A_V = 5$	3	5		10	12		40	80		V/ $\mu\text{s}$ V/ $\mu\text{s}$
GBW	Gain Bandwidth Product			2.5		4	4.5		15	20		MHz
$t_s$	Settling Time to 0.01%	(Note 7)		4			1.5			1.5		$\mu\text{s}$
$e_n$	Equivalent Input Noise Voltage	$R_S = 100\Omega$ $f = 100\text{ Hz}$ $f = 1000\text{ Hz}$		25 20			15 12			18 12		nV/ $\sqrt{\text{Hz}}$ nV/ $\sqrt{\text{Hz}}$
$i_n$	Equivalent Input Noise Current	$f = 100\text{ Hz}$ $f = 1000\text{ Hz}$		0.01 0.01			0.01 0.01			0.01 0.01		pA/ $\sqrt{\text{Hz}}$ pA/ $\sqrt{\text{Hz}}$
$C_{IN}$	Input Capacitance			3			3			3		pF

**DC Electrical Characteristics** (Note 3)

Symbol	Parameter	Conditions	LF155/6/7			LF255/6/7 LF355B/6B/7B			LF355/6/7			Units
			Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
$V_{OS}$	Input Offset Voltage	$R_S = 50\Omega$ , $T_A = 25^\circ\text{C}$ Over Temperature		3 5	7		3 5	6.5		3 10	13	mV mV
$\Delta V_{OS}/\Delta T$	Average TC of Input Offset Voltage	$R_S = 50\Omega$		5			5			5		$\mu\text{V}/^\circ\text{C}$
$\Delta TC/\Delta V_{OS}$	Change in Average TC with $V_{OS}$ Adjust	$R_S = 50\Omega$ , (Note 4)		0.5			0.5			0.5		$\mu\text{V}/^\circ\text{C}$ per mV
$i_{OS}$	Input Offset Current	$T_J = 25^\circ\text{C}$ , (Notes 3, 5) $T_J \leq T_{HGH}$		3 20	20		3 20	1		3 50	2	pA nA
$i_b$	Input Bias Current	$T_J = 25^\circ\text{C}$ , (Notes 3, 5) $T_J \leq T_{HGH}$		30 100	50		30 100	5		30 200	8	pA nA
$R_{IN}$	Input Resistance	$T_J = 25^\circ\text{C}$		$10^{12}$			$10^{12}$			$10^{12}$		$\Omega$
$A_{VOL}$	Large Signal Voltage Gain	$V_B = \pm 15\text{V}$ , $T_A = 25^\circ\text{C}$ $V_O = \pm 10\text{V}$ , $R_L = 2k$ Over Temperature	50 25	200		50 25	200		25 15	200		V/mV V/mV
$V_O$	Output Voltage Swing	$V_B = \pm 15\text{V}$ , $R_L = 10k$ $V_B = \pm 15\text{V}$ , $R_L = 2k$	$\pm 12$ $\pm 10$	$\pm 13$ $\pm 12$		$\pm 12$ $\pm 10$	$\pm 13$ $\pm 12$		$\pm 12$ $\pm 10$	$\pm 13$ $\pm 12$		V V
$V_{CM}$	Input Common-Mode Voltage Range	$V_B = \pm 15\text{V}$	$\pm 11$	+15.1 -12		$\pm 11$	+15.1 -12		+10	+15.1 -12		V V
CMRR	Common-Mode Rejection Ratio		85	100		85	100		80	100		dB
PSRR	Supply Voltage Rejection Ratio	(Note 6)	85	100		85	100		80	100		dB

**DC Electrical Characteristics**  $T_A = T_J = 25^\circ\text{C}$ ,  $V_B = \pm 15\text{V}$

Parameter	LF155A/155, LF255, LF355A/355B		LF355		LF155A/155, LF255/355B		LF355A/355		LF157A/157 LF257/357B		LF357A/357		Units
	Typ	Max	Typ	Max	Typ	Max	Typ	Max	Typ	Max	Typ	Max	
Supply Current	2	4	2	4	5	7	5	10	5	7	5	10	mA

**AC Electrical Characteristics**  $T_A = T_J = 25^\circ\text{C}$ ,  $V_B = \pm 15\text{V}$

Symbol	Parameter	Conditions	LF155/255/ 355/355B	LF156/256, LF356B	LF156/256/ 356/356B	LF157/257, LF257B	LF157/257/ 357/357B	Units
			Typ	Min	Typ	Min	Typ	
SR	Slew Rate	LF155/6: $A_V = 1$ , LF157: $A_V = 5$	5	7.5	12	30	50	V/ $\mu\text{s}$ V/ $\mu\text{s}$
GBW	Gain Bandwidth Product		2.5		5		20	MHz
$t_s$	Settling Time to 0.01%	(Note 7)	4		1.5		1.5	$\mu\text{s}$
$e_n$	Equivalent Input Noise Voltage	$R_B = 100\Omega$ $f = 100\text{ Hz}$ $f = 1000\text{ Hz}$	25 20		15 12		15 12	nV/ $\sqrt{\text{Hz}}$ nV/ $\sqrt{\text{Hz}}$
$i_n$	Equivalent Input Current Noise	$f = 100\text{ Hz}$ $f = 1000\text{ Hz}$	0.01 0.01		0.01 0.01		0.01 0.01	pA/ $\sqrt{\text{Hz}}$ pA/ $\sqrt{\text{Hz}}$
$C_{IN}$	Input Capacitance		3		3		3	pF

**Notes for Electrical Characteristics**

Note 1: The maximum power dissipation for these devices must be derated at elevated temperatures and is dictated by  $T_{MAX}$ ,  $\theta_{JA}$ , and the ambient temperature,  $T_A$ . The maximum available power dissipation at any temperature is  $P_d = (T_{MAX} - T_A) / \theta_{JA}$  or the ZPC  $P_{dMAX}$ , whichever is less.

Note 2: Unless otherwise specified the absolute maximum negative input voltage is equal to the negative power supply voltage.

Note 3: Unless otherwise stated, these test conditions apply:

	LF155A/5A/7A LF155/5/7	LF255/5/7	LF355A/5A/7A	LF355B/5B/7B	LF355/5/7
Supply Voltage, $V_B$	$\pm 15\text{V} \leq V_B \leq \pm 20\text{V}$	$\pm 15\text{V} \leq V_B \leq \pm 20\text{V}$	$\pm 15\text{V} \leq V_B \leq \pm 18\text{V}$	$\pm 15\text{V} \leq V_B \leq \pm 20\text{V}$	$V_B = \pm 15\text{V}$
$T_A$	$-55^\circ\text{C} \leq T_A \leq +125^\circ\text{C}$	$-25^\circ\text{C} \leq T_A \leq +85^\circ\text{C}$	$0^\circ\text{C} \leq T_A \leq +70^\circ\text{C}$	$0^\circ\text{C} \leq T_A \leq +70^\circ\text{C}$	$0^\circ\text{C} \leq T_A \leq +70^\circ\text{C}$
$T_{HIGH}$	$+125^\circ\text{C}$	$+85^\circ\text{C}$	$+70^\circ\text{C}$	$+70^\circ\text{C}$	$+70^\circ\text{C}$

and  $V_{OS}$ ,  $t_s$  and  $I_{OS}$  are measured at  $V_{CM} = 0$ .

Note 4: The Temperature Coefficient of the adjusted input offset voltage changes only a small amount (0.5 $\mu\text{V}/^\circ\text{C}$  typically) for each mV of adjustment from its original unadjusted value. Common-mode rejection and open loop voltage gain are also unaffected by offset adjustment.

Note 5: The input bias currents are junction leakage currents which approximately double for every 10 $^\circ\text{C}$  increase in the junction temperature,  $T_J$ . Due to limited production test time, the input bias currents measured are correlated to junction temperature. In normal operation the junction temperature rises above the ambient temperature as a result of internal power dissipation,  $P_d$ .  $T_J = T_A + \theta_{JA} P_d$  where  $\theta_{JA}$  is the thermal resistance from junction to ambient. Use of a heat sink is recommended if input bias current is to be kept to a minimum.

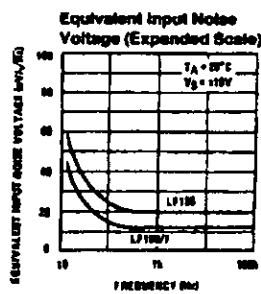
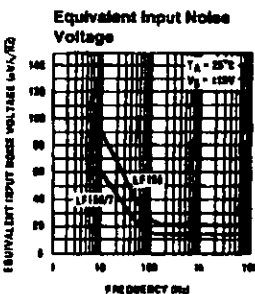
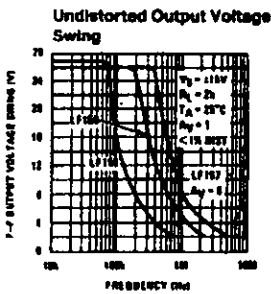
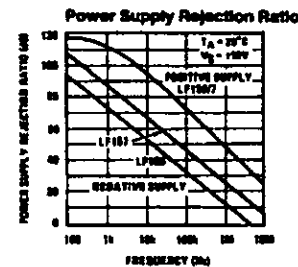
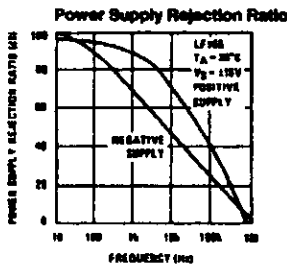
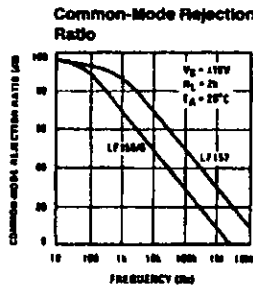
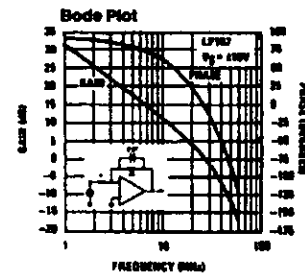
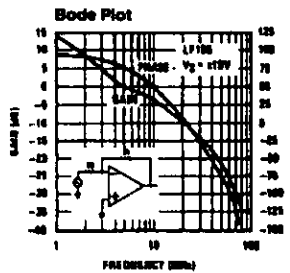
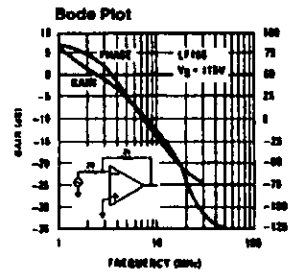
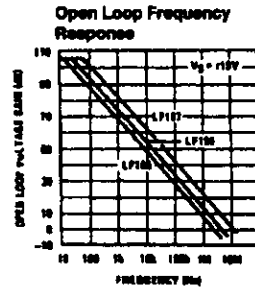
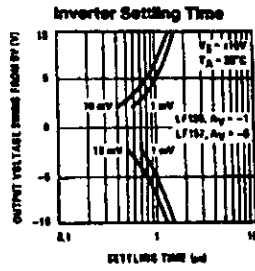
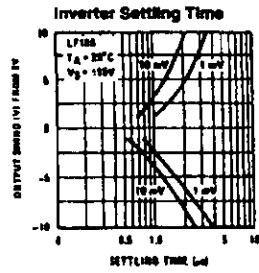
Note 6: Supply Voltage Rejection is measured for both supply magnitudes increasing or decreasing simultaneously, in accordance with common practice.

Note 7: Settling time is defined here, for a unity gain inverter connection using 2 k $\Omega$  resistors for the LF155/5. It is the time required for the output voltage (the voltage at the inverting input pin on the amplifier) to settle to within 0.01% of its final value from the time a 10V step input is applied to the inverter. For the LF157,  $A_V = -5$ , the feedback resistor from output to input is 2 k $\Omega$  and the output step is 10V (See Settling Time Test Circuit).

Note 8: Refer to RETS155AX for LF155A, RETS155X for LF155, RETS155MAX for LF155A, RETS155X for LF155, RETS157A for LF157A and RETS157X for LF157 military specifications.

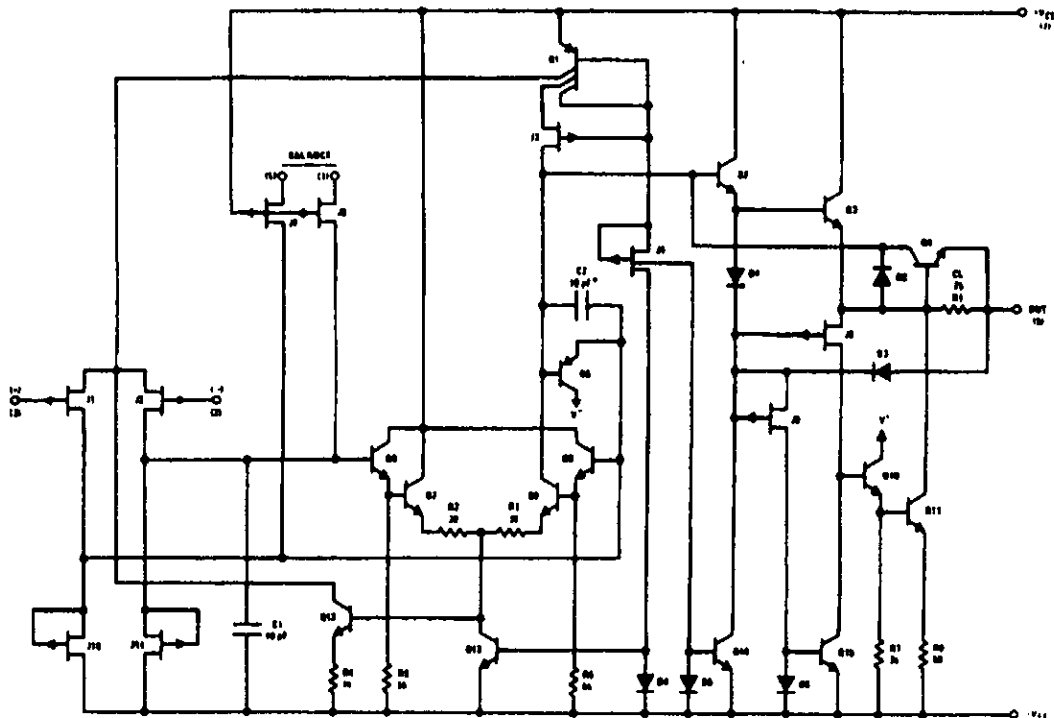
Note 9: Max. Power Dissipation is defined by the package characteristics. Operating the part near the Max. Power Dissipation may cause the part to operate outside guaranteed limits.

Typical AC Performance Characteristics (Continued)



TL770046-11

Detailed Schematic

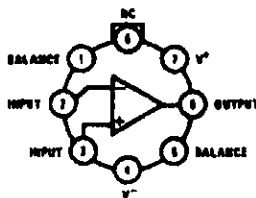


\*C = 3 pF in LF157 series.

TL/H/5646-13

Connection Diagrams (Top Views)

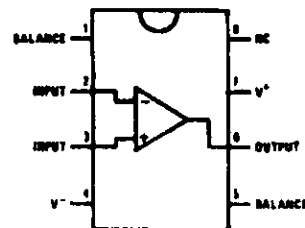
Metal Can Package (H)



TL/H/5646-14

Order Number LF155AH, LF156AH, LF157AH, LF155H, LF156H, LF157H, LF256H, LF256H, LF267H, LF355AH, LF356AH, LF357AH, LF356BH, LF355H, LF356H, LF357H, LM155AH/883, LM155H/883, LM156AH/883, LM156H/883, LM157AH/883 or LM157H/883\*  
See NS Package Number H08C

Dual-In-Line Package (M and N)



TL/H/5646-20

Order Number LF355M, LF356M, LF357M, LF355BM, LF356BM, LF356BN, LF356BN, LF357BN, LF355N, LF356N or LF357N  
See NS Package Number M08A or N08E



# DM54ALS244A/DM74ALS244A

## Octal TRI-STATE® Bus Driver

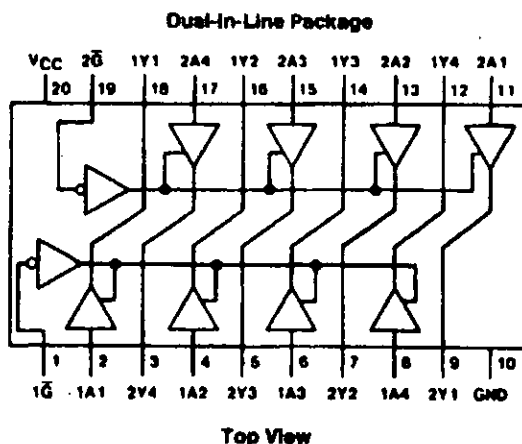
### General Description

This octal TRI-STATE bus driver is designed to provide the designer with flexibility in implementing a bus interface with memory, microprocessor, or communication systems. The output TRI-STATE gating control is organized into two separate groups of four buffers, and both control inputs enable the respective outputs when set logic low. The TRI-STATE circuitry contains a feature that maintains the buffer outputs in TRI-STATE (high impedance state) during power supply ramp-up or ramp-down. This eliminates bus glitching problems that arise during power-up and power-down.

### Features

- Advanced low power oxide-isolated ion-implanted Schottky TTL process
- Functional and pin compatible with the DM54/74LS counterpart
- Improved switching performance with less power dissipation compared with the DM54/74LS counterpart
- Switching response specified into 500Ω and 50 pF load
- Switching response specifications guaranteed over full temperature and V<sub>CC</sub> supply range
- PNP input design reduces input loading
- Low level drive current:  
54ALS = 12 mA, 74ALS = 24 mA

### Connection Diagram



TL/F/6212-1

Order Number DM54ALS244AJ, DM74ALS244AWM, DM74ALS244AN or DM74ALS244ASJ  
See NS Package Number J20A, M20B, M20D or N20A

### Function Table

Input		Output
$\bar{G}$	A	Y
L	L	L
L	H	H
H	X	Z

H = High Level Logic State  
L = Low Level Logic State  
X = Don't Care (Either Low or High Level Logic State)  
Z = High Impedance (Off) State



# DM54ALS245A/DM74ALS245A Octal TRI-STATE® Bus Transceiver

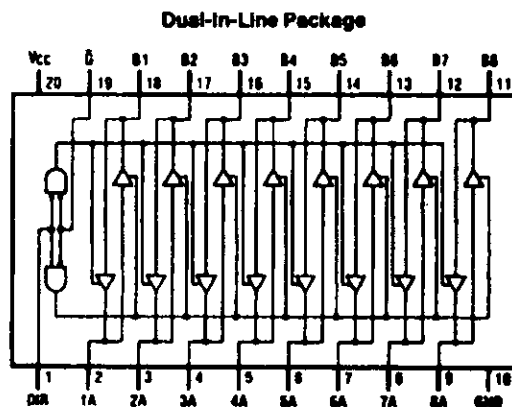
## General Description

This advanced low power Schottky device contains 8 pairs of TRI-STATE logic elements configured as octal bus transceivers. These circuits are designed for use in memory, microprocessor systems and in asynchronous bidirectional data buses. Two way communication between buses is controlled by the (DIR) input. Data transmits either from the A bus to the B bus or from the B bus to the A bus. Both the driver and receiver outputs can be disabled via the ( $\bar{G}$ ) enable input which causes outputs to enter the high impedance mode so that the buses are effectively isolated.

## Features

- Advanced oxide-isolated, ion-implanted Schottky TTL process
- Non-inverting logic output
- Glitch free bus during power up and down
- TRI-STATE outputs independently controlled on A and B buses
- Low output impedance to drive terminated transmission lines to 133 $\Omega$
- Switching response specified into 500 $\Omega$ /50 pF
- Specified to interface with CMOS at  $V_{OH} = V_{CC} - 2V$
- PNP inputs to reduce input loading
- Switching specifications guaranteed over full temperature and  $V_{CC}$  range

## Connection Diagram



Order Number DM54ALS245AJ, DM74ALS245AWM, DM74ALS245AWN or DM74ALS245ASJ  
See NS Package Number J20A, M20B, M20D or N20A

TL/F/6213-1

## Function Table

Control Inputs		Operation
$\bar{G}$	DIR	
L	L	B Data to A Bus
L	H	A Data to B Bus
H	X	Hi-Z

H = High Logic Level  
L = Low Logic Level  
X = Either High or Low Logic Level



---

APÉNDICE I. LISTADO DEL PROGRAMA (SISTEMA MÍNIMO)	118
APÉNDICE II. ECUACIONES Y NOMENCLATURA	153
APÉNDICE III. LA PC ENFOCADA A TARJETAS DE EXPANSIÓN	171
APÉNDICE IV. ESPECIFICACIONES TÉCNICAS	186
BIBLIOGRAFÍA	210

---

# **BIBLIOGRAFÍA**

---

**Circuitos Integrados Lineales y amplificadores Operacionales**

Robert F. Coughlin, Frederick F. Driscoll

Segunda Edición

Prentice Hall

**Guía para Mediciones Electrónicas y Prácticas de Laboratorio**

Stanley Wolf

Prentice Hall

**Electrónica, Teoría de Circuitos**

Robert Boylestad, Louis Nashelsky

Cuarta Edición

Prentice Hall

**Lógica Digital y Diseño de Computadoras**

M. Morris Mano

Prentice Hall

**Interfacing. A Laboratory Approach Using the Microcomputer for instrumentation, data analysis, and Control.**

Stephen E. Derenzo

Prentice Hall

**Interfacing Sensors to the IBM PC**

Willis J. Tompkins, John G. Webster

Prentice Hall

**Upgrading and Repairing PCs**

Scott Mueller

Quinta Edición

QUE

**C++. Manual de Referencia**

Herbert Schildt  
Mcgraw-Hill

**Borland C++. Manual de Bolsillo**

Luis Joyanes Aguilar  
Mcgraw-Hill

**Apuntes de Análisis de Circuitos Eléctricos, Primera Parte**

Wilbert Arcila Rodriguez, J. Lino Vidal Macedo  
Editado por la División de Ingeniería Mecánica y Eléctrica, Departamento de  
Ingeniería de Control (Facultad de Ingeniería, U.N.A.M.)

**Turbo Assembler, Quick Reference Guide**

Borland International, Inc.

**PCL-750, Prototype Development Card. User's Manual**

Advantech Co., Ltd.

**PCL-812, Enhanced Multi-Lab Card. User's Manual**

Advantech Co., Ltd.

**Labview, Programación Gráfica para Instrumentación, 1992**

National Instruments

**Data Acquisition Seminar, August 1994 Edition**

National Instruments

**Instrumentation Newsletter, Volume 7, Number 1, Spring 1995**

National Instruments

**Optoelectronics Device Data**

Motorola

**Memory Device Data**

Motorola

**Fast And LS TTL Data**

Motorola

**Data Acquisition, Databook, 1993**

National Semiconductor

**Operational Amplifier, Databook, 1993**

National Semiconductor

**Saber Electrónica, Número 10, Año 6, Enero/Diciembre 1993**

Editorial Saber del Uruguay, S. R. L.