

ESCUELA POLITECNICA NACIONAL

FACULTAD DE INGENIERIA ELECTRICA

**TARJETA DE ADQUISICION DE DATOS ANALOGICOS
UTILIZANDO UN INTERFAZ PERIFERICO PROGRAMABLE (PPI)
CON PROGRAMA EN ENTORNO VISUAL**

**TESIS PREVIA A LA OBTENCION DEL TITULO DE INGENIERO EN LA
ESPECIALIZACION DE ELECTRONICA Y CONTROL**

CARLOS GERMAN PILLAJO ANGOS

Quito, Noviembre de 1997

AGRADECIMIENTO

En este sueño hecho realidad, doy gracias a mi amigo incondicional, quien desde el inicio de mi carrera me dio luz en el camino, dándome aliento y fuerza creadora a la vez. "Eres único gracias Padre Celestial"

A mis padres que hicieron de mi un luchador incansable, en especial a mi madre por el amor que me ha brindado, por su fortaleza y por infundirme constancia en el trabajo.

Un agradecimiento sincero al Ing. Marco Barragan y, a todos los amigos, compañeros y demás personas que colaboraron con sus consejos oportunos, especialmente a J.G.

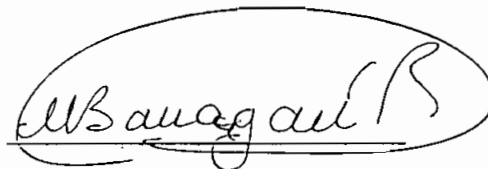
DEDICATORIA

Con este trabajo quiero despertar en Ustedes, queridos compañeros, el deseo de aplicar los conocimientos adquiridos con el afán de superación.

Este trabajo es para Ustedes.

CERTIFICACION

Certifico que el presente trabajo ha sido
realizado en su totalidad por el señor
Carlos Germán Pillajo Angos

A handwritten signature in black ink, enclosed within a hand-drawn oval. The signature appears to read "M. Barragán" with a stylized flourish at the end.

Ing. Marco Barragán.

Director de Tesis.

INDICE

Página #

CAPITULO I

1.1	INTRODUCCION	1
1.2	REQUISITOS FUNCIONALES	3
1.2.1.	REQUISITOS DE HARDWARE	4
1.2.1.1	CARACTERISTICAS GENERALES DE LOS COMPUTADORES PERSONALES	4
1.2.1.2	MEMORIA	7
1.2.1.3	CLASIFICACION BASICA DE LOS COMPUTADORES SEGUN SU MICROPROCESADOR	10
1.2.2	LAS SEÑALES PERMITIDAS	
1.2.2.1	BUSES	13
	- LOS BUSES LOCALES	21
1.2.2.2	DESCRIPCION DE LAS SEÑALES DEL - BUS DEL SISTEMA	24
1.3	ENLACE DE LA TARJETA AL PC	
1.3.1	EL CONECTOR	32
	- TAMAÑO DE LA TARJETA	33
	- POTENCIA DE LA TARJETA MADRE	33
	- DESACOPLAMIENTO DE POTENCIA	34
1.3.2	PORTICOS DEL COMPUTADOR PERSONAL	35

CAPITULO II

DISEÑO DEL HARDWARE

2.1	ESPECIFICACIONES INICIALES	36
2.1.1	ESTUDIO DE LA INTERFAZ PROGRAMABLE	37

- DESCRIPCION GENERAL	37
- FUNCIONES DE LOS PINES	38
- El registro de control	40
- En el Modo 0	41
- En el Modo 1	42
- En el Modo 2	44
2.2 REQUERIMIENTOS DE HARDWARE	45
2.3 SEÑALES Y DIRECCIONES UTILIZADAS POR LA TARJETA ADAIO....	48
2.4 DESCRIPCION DEL HARDWARE	
2.4.1 DESCRIPCION DEL DISEÑO	49
2.4.1.1 BLOQUE DE DECODIFICACIÓN	50
2.4.1.2 SEÑALES DEL PPI	52
2.4.1.3 BLOQUE DE CONVERSION D/A	53
- CONVERTOR DIGITAL ANALOGICO (DAC0830), DE 8 BITS	54
- Consideraciones Digitales	56
- Consideraciones análogas	57
- Consideraciones del Amplificador Operacional	57
2.4.1.4 BLOQUE DE CONVERSION A/D	59
- Reloj del conversor	61
2.5. DISPOSICION DE ELEMENTOS EN LA TARJETA	62

CAPITULO III

DISEÑO DEL SOFTWARE

3.1 ESPECIFICACION Y SELECCION DEL PROGRAMA	64
3.2 PROGRAMAS DE COMUNICACION	67

3.2.1	INSTRUCCIONES PARA EL MANEJO DE PORTICOS I/O	68
3.2.2	LIBRERIA DE ENLACE DINAMICO (DLL)	68
3.2.3	BLOQUE DE PROGRAMA DE COMUNICACION (Lenguaje G) ...	70
3.3	PROGRAMAS EN ENTORNO VISUAL	72
3.3.1	PROGRAMA EN QBASIC	72
3.3.2	PROGRAMA EN VISUAL BASIC (V3.0)	74
	- REALIZACION DEL ARCHIVO DE AYUDA	80
3.3.3	BLOQUES DE PROGRAMA EN LAB VIEW	82

CAPITULO IV

PRUEBAS Y RESULTADOS

4.1	PROGRAMAS DE COMPROBACION DE DIRECCIONES	84
4.2	PRUEBAS DE ACOPLAMIENTO Y FUNCIONAMIENTO	
4.2.1	PROGRAMAS DE GENERACION DE DATOS	84
	4.2.1.1 LISTADO DEL PROGRAMA EN TURBO C++	85
	4.2.1.2 LISTADO DEL PROGRAMA EN QBASIC	86
	4.2.1.3 LISTADO DEL PROGRAMA EN VISUAL BASIC V3.0	87
	4.2.1.4 LISTADO DEL PROGRAMA EN LAB VIEW	87
4.2.2	PROGRAMAS DE ADQUISICION DE DATOS.	
	4.2.2.1 PROGRAMA DE ADQUISICION EN QBASIC	89
	4.2.2.2 PROGRAMA ADQUISICION EN VISUAL BASIC V3.0	90
	4.2.2.3 PROGRAMA DE ADQUISICION EN LAB VIEW	91
4.3	RESULTADOS OBTENIDOS	93
4.3.1	RESULTADO DE LAS PRUEBAS DE CORRECTA OPERACION .	94
4.3.2	RESULTADO DE LAS PRUEBAS DE	
	FRECUENCIA DE OPERACION	94

4.3.3	RESULTADOS DE LA FRECUENCIA DE ADQUISICION	95
4.4	ANALISIS TECNICO-ECONOMICO	96
4.4.1	ANALISIS TECNICO	97

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES.

5.1	CONCLUSIONES	99
5.2	RECOMENDACIONES	101
	BIBLIOGRAFIA	102

ANEXOS

A	GRAFICOS DE LA TARJETA.	
A.1	Diagrama de Bloques del Diseño	103
A.2	Diagrama esquemático del Diseño	104
A.3	Gráfico de la Disposición de elementos en la tarjeta	105
B	MANUAL DEL USUARIO.	
B.1	CARACTERISTICAS GENERALES	106
B.2	CONFIGURACION	107
B.3	INSTALACION	108
B.4.	CONEXION	109
	TABLERO DE CONEXIONES Y PRUEBAS DE LA TARJETA ADAIO	112
C	SOFTWARE	
C.1.	ASIGNACION DE DIRECCION BASE	115
C.2.	ASIGNACION DE DIRECCION DE SEÑALES ANALOGAS	115
D	CONSIDERACIONES TECNICAS	116
E	DIAGRAMAS DE FLUJO DE LOS PROGRAMAS	117
F	HOJAS TECNICAS DE LOS PRINCIPALES ELEMENTOS UTILIZADOS.	

CAPITULO I

1.1 INTRODUCCION

Son muchas las ocasiones en las cuales se necesita controlar o conocer el estado de algunos elementos, dispositivos o sistemas externos al computador, y los periféricos que éste posee no son suficientes o adecuados para ello. Así por ejemplo el puerto paralelo es empleado para realizar algunos de estos propósitos y , aunque supera muchas expectativas que se tienen con respecto al control de sistemas, en ocasiones se ve también limitado en cuanto a la direccionabilidad de las señales o al número de variables físicas que se pueden controlar simultáneamente.

Debido a que, en su configuración básica, el computador personal no dispone de puertos de entrada y salida analógicos que permitan realizar acciones de supervisión y control directamente sobre los procesos físicos existiendo, sin embargo, tarjetas que se comercializan para este efecto, pero con muy altos costos.

Buscando superar las deficiencias antes mencionadas se propone una nueva alternativa como lo es una tarjeta de adquisición y emisión de datos análogos la misma que busca dar una visión de la utilización de los modernos elementos como son los periféricos programables los que se caracterizan por su universalidad, estos componentes son vistos por el procesador como unidades de

entrada y salida, agrupando de esta manera un cúmulo de alternativas para el control y lectura más directa sobre un mayor número de variables física externas localizadas en dispositivos ajenos al computador.

En vista de lo mencionado anteriormente un primer objetivo del presente trabajo es la utilización de un Interfaz Periférico Programable (PPI) en el diseño y construcción de una tarjeta de adquisición de datos, a la que se le ha denominado ADAIO, que incluye: 16 pórticos analógicos, 8 de entrada y 8 de salida, los cuales funcionan con voltajes normalizados entre 0 y $+10V_{DC}$, pórticos que pueden ser conectados al bus interno, para tarjetas de extensión, de cualquier computador personal compatible que posea ranuras de expansión tipo ISA.

La utilización de la tarjeta ADAIO acoplada como nuevo hardware de un computador personal, por otro lado, abre un espectro de aplicaciones muy amplio para el seguimiento de variables, puesto que, en la actualidad existen una gran cantidad de lenguajes de programación en entorno visual.

En el primer capítulo se da una breve información previa para el entendimiento de algunos detalles importantes que permiten un conocimiento y uso de señales en los computadores, indicándose aquellas señales que se utilizaron para el desarrollo de la tarjeta de adquisición de datos ADAIO, se describe brevemente los buses utilizados por los PC's, se proporciona información del funcionamiento del Interfaz Periférico Programable (PPI).

En el segundo capítulo se describen las partes que componen la tarjeta ADAIO, los puntos estratégicos para su construcción y los aspectos básicos de la misma,

se da énfasis en el acoplamiento de las señales y el modo de programación empleada para el PPI.

En el capítulo III se presentan algunos programas a fin de comprobar el funcionamiento de la tarjeta, en diferentes ámbitos de programación como lo son en: a) Programación convencional (Qbasic), b) En lenguajes en entorno visual bajo windows, como lo son Visual Basic V3.0 y Lab View

En el capítulo IV se presentará las pruebas realizadas y los resultados obtenidos en lo relativo al acoplamiento y funcionamiento en los entornos de programación, mencionados concluyéndose con un análisis técnico económico.

En el capítulo V, se presentará las conclusiones y recomendaciones resultantes del desarrollo del presente proyecto de tesis.

1.2 REQUISITOS FUNCIONALES

Para una mejor comprensión de los requisitos funcionales necesarios para el diseño y construcción de la tarjeta ADAIO, primero se empieza estudiando las características básicas con las que está constituido un computador personal o compatible, cómo ve éste a los elementos externos, se estudian los tipos de memoria, luego se ve la clasificación de los computadores personales de acuerdo a su microprocesador con lo cual se empieza a enmarcar la compatibilidad del hardware deseado. Más adelante se estudian los tipos de buses existentes de los cuales se escoge el que permita una conexión y una mayor compatibilidad con los equipos existentes a nuestro alcance y continuación se describe las señales

del bus utilizado.

Para terminar con el capítulo se ven además las características mecánico-eléctricas permitidas para las ranuras (slots) de los computadores.

1.2.1. REQUISITOS DE HARDWARE

Según lo establecido, a continuación se hace una breve mención de las características importantes de los Computadoras Personales, a fin de determinar el hardware compatible en el cual se desenvuelve el presente trabajo.

1.2.1.1 CARACTERISTICAS GENERALES DE LOS COMPUTADORES PERSONALES

CPU (Central Processing Unit).- Realmente la Unidad Central de Procesamiento es en sí EL COMPUTADOR; los demás son dispositivos que hacen que el trabajo del computador pueda ser visto o controlado por los humanos. La CPU es el circuito electrónico encargado de dirigir el flujo de la información entre él y los demás dispositivos externos; normalmente la CPU es un solo circuito integrado (un CHIP MICROPROCESADOR), existiendo algunas familias en el mercado, de las cuales interesan los de la línea fabricada por INTEL llamada familia iapx86 compuesta de varias pastillas de circuitos integrados (chips) muy similares como son: el 80286, 80368, 80486 y 80586; estos microprocesadores determinan cuanta memoria

puede utilizar el sistema, que clase de programas puede correr y que tan veloz puede avanzar.

Memoria.- Aunque el computador y toda su capacidad de trabajo están definidos por la CPU, ella sola no puede hacer operación alguna. Se sabe que un computador procesa la información de acuerdo a las instrucciones dadas por los humanos. Estas instrucciones deben ser almacenadas en memoria la cual esta formada por circuitería electrónica. Hay diferentes tipos de memoria: convencional, alta, extendida, expandida, y cada una resuelve un problema diferente, así por ejemplo, algunos programas no corren si no cuenta con cierta cantidad mínima de una clase determinada de memoria.

Bus.- Los datos o señales que fluyen entre la CPU y la memoria son transportados a través del Bus, clasificándose según ellas en bus de datos, bus de direcciones y bus de control.

El número máximo de datos que puede manipular el computador en una operación esta determinado por su tamaño de palabra (bus de datos), este puede ser de 8,16 o 32 bits, independientemente de que tan grande sea la palabra del computador, los datos se deben transportar al microprocesador, obviamente, un bus más ancho permitirá la entrada de más datos en menos tiempo. Considérese, por ejemplo, una 8088 de 8 Mhz contra una 8086 de 8 Mhz, la única diferencia entre las dos es que la primera tiene una ruta de datos de 8 bits y la segunda de 16 bits.

El bus de direcciones y de control determinan qué clase de tarjetas del

circuito de expansión pueden funcionar en la máquina, en esto, todos los buses conocidos son compatibles en diferentes grado uno con otro excepto los de micro canal.

Respecto a los buses en orden cronológico tenemos: a) El bus ISA (Industry Standard Architecture) es el nuevo nombre dado al bus compatible que utilizan la mayor parte de los computadores, b) El de Micro Canal el cual fue introducido en 1987 por IBM para sus máquinas PS/2 ,y, c)El bus EISA (Extended Industry Standard Architecture) es la respuesta al Micro Canal dada por los fabricantes ajenos a IBM.

BIOS.- (Basic Input/Output System) es el programa de sistema a bajo nivel que determina la compatibilidad de la máquina. Este programa se encuentra en una memoria sólo de lectura (ROM), el cual se localiza en la tarjeta del sistema (tarjeta madre). Como se conoce, el Sistema Operativo DOS no se comunica directamente con el equipo sino que emite comandos a través del BIOS.

Tarjeta de Vídeo .- La tarjeta de vídeo afecta el tipo de programas que se pueden correr y qué tan rápidamente llegan los datos a la pantalla. Se le puede remplazar fácilmente, y hasta el computador personal más antiguo puede utilizar cualquier versión desde una tarjeta monocromática hasta una VGA.

Las tarjetas de vídeo han mejorado ofreciendo más colores, mostrando más puntos en la pantalla y siendo más rápidas.

Puerto Paralelo.- El puerto paralelo común ofrece sólo una de las formas de añadir entradas y salidas, accesibles al usuario de un computador personal. Los circuitos y el conector para el puerto paralelo generalmente se encuentran en una tarjeta de expansión, y debido a que los circuitos de puerto no son complejos, las tarjetas de expansión casi siempre incluyen funciones adicionales.

Otro nombre dado al puerto paralelo es puerto de impresora, reflejo de su uso más común.

Puerto Serial.- El puerto serial del Computador Personal, conocido también como puerto RS-232, es muy útil, ya que permite la comunicación, no sólo con otros computadores, sino también con otros dispositivos tales como mouses, modems y por supuesto microcontroladores.

1.2.1.2 MEMORIA

En vista de la importancia de la memoria y como se mencionó existen bloques de memoria los que se clasifican en: convencional, alta, extendida y expandida. Las tres primeras se distinguen sólo por las direcciones que ocupan en el mapa de memoria del sistema. La memoria convencional ocupa los primeros 640K La memoria alta también reside en el primer megabyte entre los 640K y los 1.024K (1MB). La memoria extendida comienza en 1Mb y continua hasta el tope de la memoria física presente en el sistema.

Los computadores con microprocesadores 80286 y otros posteriores pueden dar acceso a más memoria llamada memoria extendida. Los Computadores pueden

usar 640K de memoria convencional al correr el Sistema Operativo y programas para DOS. Estos primeros 640K están localizados en la tarjeta madre o en tarjetas de expansión.

Muchos computadores compatibles disponibles con un microcontrolador 80386 y 80486 alojan hasta 4 MB directamente en la tarjeta madre. El resto está configurado en memoria extendida o expandida.

La memoria expandida, sin embargo, está fuera del espacio normal de direcciones del procesador y para tener acceso a la misma hay que usar un programa especial llamado administrador de memoria expandida (expanded memory manager) o EMM, este programa divide la memoria expandida en una serie de unidades de 16K llamadas páginas, cada página separa 64K de espacio de direcciones sin usar en el área de memoria alta, para que sirvan como un cuadro de página de EMS (Especificación de Memoria Expandida).

Los diseñadores del primer computador IBM PC decidieron reservar el área de memoria alta (una vez más, el espacio de memoria entre los 640K y 1MB), para los buffers de video, la ROM del sistema y la RAM instaladas en tarjetas adaptadoras.

Espacio de Memoria para vídeo.- La memoria de vídeo es la que utilizan las tarjetas de vídeo para llevar un registro de lo que se va a mostrar en pantalla. Cuando un programa coloca un carácter o dibuja un círculo en la pantalla, en realidad está haciendo cambios a esta memoria de vídeo.

IBM separó 128K de la memoria principal para la memoria de vídeo, aunque la mayor parte de las tarjetas de vídeo no necesita realmente usar tanto espacio de memoria. En la siguiente tabla se indican algunas tarjetas de vídeo popularizadas con su capacidad de memoria.

TARJETA	ESPACIO DE ACCESO A MEMORIA UTILIZADO	MEMORIA TOTAL DE LA TARJETA (K)	ROM EN LA TARJETA
Monichrome Display Adapter (MDA)	B0000-B1000(4K)	4	Nada en el espacio de direcciones de memoria.
Color Graphics	B8000-BC000(16K)	16	Nada en el espacio de direcciones de memoria.
Enhanced Graphics Adapter (EGA)	A0000-BFFFF(128K) (B0000-B1000 pueden desactivarse)	256	C0000-C3FFF (16K)
Vídeo Graphics Array (VGA)	A0000-BFFFF(128K) (B0000-B1000 pueden desactivarse)	256	C0000-C5FFF (24K)
Super VGA	A0000-BFFFF(128K)	512-1024K	C0000-C7FFF (32K)

ROM (Memoria para lectura solamente).- En este segmento de la memoria principal los programas son cargados sólo una vez con un dispositivo especial que puede ser un PROM(Memory Only Read Programmer), un programador EPROM. Pues bien, a diferencia de la RAM normal, la memoria ROM no pierde los datos de su memoria cuando se apaga la máquina. En esencia puede decirse que ROM contiene los programas que le dicen al sistema cómo usar una tarjeta de circuito.

Debe insistirse de que el sistema operativo no se comunica directamente con el equipo sino que emite comandos a través de BIOS, de tal manera que determina cuan compatible es un Computador Personal.

Memoria Extendida.- Un Computador con microprocesador 80286 puede realmente dirigirse a 16 megabytes, así como uno con 80386 u 80486

pueden **dirigirse a 4 gigabytes de RAM. La memoria RAM superior a 1 MB es memoria extendida.** Estos chips tienen acceso a memoria mayor que 1024K (1 MB), pero únicamente en modalidad protegida.

Decimal	Memoria	Hex
1408K-1 (1408=1024+384)	Memoria Extendida	15FFFF
1024k-1		FFFFF
	Área reservada: ROMs y Buffers	BFFFF
768K-1		
	Area reservada: Memoria de vídeo	9FFFF
640K-1		
	Area de datos de programas	3B400
247K		
	Aplicaciones DOS	11400
69K		
	TSRs	0D800
54K		
	DOS	00400
1K		
	Area BIOS	00000
0K		

Tabla 1.2.1

En la tabla 1.2.1 se puede observar el mapa de memoria básico de un computador personal.

1.2.1.3 CLASIFICACION BASICA DE LOS COMPUTADORES SEGUN SU MICROPROCESADOR

En busca de realizar un hardware compatible con todo computador personal , y tomando en cuenta que, en el mercado de los computadores se utilizan varios circuitos microprocesadores, los mismos que en su mayoría están basadas en una línea fabricada por INTEL llamada familia iapx86, conformada

por:

El 8088 es el anterior y menos poderoso que el 80286 y se utilizan en la mayor parte de las máquinas de clase XT. Este microprocesador viene con lo que se acostumbra llamar paquete de 40 patas DIP (Dual In-line Package —Paquete de doble hilera—). Los computadores Turbo AT/XT compatibles pueden tener velocidad de procesamiento de sus datos a 8 Mhz.

El 80286 , diseñado por Intel en 1981. Este microprocesador viene en un cuadrado de plástico llamado PGA (Pin Grid Array —Disposición reticular de patas—), contiene mucho más poder que el 8088, debido a esto, éste microprocesador se calienta más rápidamente al trabajar y , por lo tanto, se requiere de dispositivos adicionales de enfriamiento como disipadores de calor.

El 80386 , en sus dos versiones 80386DX y el 80386SX , es un poderoso miembro de la familia del 80x86. Fue introducido en 1985 y viene en un paquete PGA. Contiene abundancia de funciones de programación, incluyendo la habilidad de multitareas en programas bajo DOS con la ayuda de un programa supervisor. Su ruta de datos de 32 bits acelera el acceso a la información, aunque sólo pocas tarjetas de expansión utilizan plenamente la función de 32 bits. La 386SX es idéntica a el 386DX excepto que tiene una ruta de datos externos de 16 bits.

El 80486 , en sus dos versiones 486DX y 80486SX es una especie de mejora al 386. Combina a su predecesor afinado con dos circuitos integrados que aceleran los sistemas 386: el controlador caché 385 y el coprocesador numérico 387, su

microcódigo es más grande y más rápido, de manera que un 386 de 25 MHz en conjunto con un 385 y un 387 ejecutan por segundo solo la mitad de las instrucciones que un 486 de 25 Mhz.

El 486SX tiene una ruta de datos de 32 bits, controlador de memoria y 8K de caché, igual que el 80486DX , pero no tiene coprocesador numérico.

El 80486DX2, el chip Overdrive y la duplicación de reloj. A diferencia de los coprocesadores de punto flotante de un tiempo atrás (las familias 8087, 187, 287 y 387), el 80487SX no colabora con el procesador principal anterior, sino que se toma el control de todo el sistema por completo, desconectando efectivamente al 80486SX. Intel llevó las cosas un poco más allá con un chip llamado Overdrive, éste circuito integrado Overdrive posee al propiedad de correr simultáneamente a dos velocidades de reloj.

Cuando transfiere datos a o desde el bus, la memoria o cualquiera de los otros elementos aparte de sí mismo, lo hace a 25 Mhz; pero las operaciones internas —operaciones aritméticas, toma de decisiones, operaciones lógicas, y similares— son efectuadas a dos veces la velocidad del reloj — ! 50 Mhz internamente ! El efecto neto es que el chip Overdrive hace todo al menos tan velozmente como el 80486DX de 25 Mhz, y muchas operaciones a doble velocidad. El resultado es que este chip Overdrive acelera la velocidad del computador de treinta y tres hasta un cincuenta por ciento aproximadamente.

El 80486DX2, al igual que el chip Overdrive, corre externamente a "X" Mhz, pero internamente a "2X" Mhz.

De lo mencionado anteriormente, se puede decir que para el desarrollo adecuado del presente proyecto de tesis se necesita un computador que tenga un microprocesador de rápida respuesta, que trabaje con multitarea, que posea una gran capacidad de memoria RAM, y que tenga un coprocesador matemático incorporado para una mayor velocidad en sus procesos matemáticos utilizados en operaciones por programas con entorno visual.

1.2.2 LAS SEÑALES PERMITIDAS

Luego de haber determinado el hardware básico para el funcionamiento de la tarjeta ADAIO la próxima interrogante que se plantea en el desarrollo es ¿ Cuáles son las señales básicas que utiliza la tarjeta, de tal manera que sea adaptable a todo computador compatible?. Para responder a esta interrogante previamente se estudiará el conjunto de señales permitidas en las ranuras de los computadores llamadas buses.

1.2.2.1 BUSES

La estructura de ranuras o slots, para conectar tarjetas de expansión a un sistema básico de microcomputador, ideado primero por Apple y luego masificado por IBM en su línea de Computadores Personales, XT's y demás clones compatibles, ha demostrado que sigue siendo conveniente desde el punto de vista conceptual y funcional. A través de cada ranura se logran llevar todas las señales necesarias para realizar las diferentes interfaces del sistema

(impresoras, unidades de disco, pantallas, modems, scanner, etc.). El conjunto de estas señales recibe el nombre de BUS y tiene, en sus diferentes configuraciones, un determinado número de señales con funciones específicas y una forma física estándar con el fin de poder conectar y/o desconectar en él diferentes modelos de tarjetas con diversas aplicaciones.

Esto, ha permitido el desarrollo y versatilidad de los computadores personales, adaptándolos a muchas aplicaciones teniendo como base un sistema principal compuesto por un microprocesador (CPU), una memoria básica y una serie de circuitos de entrada y salida mínimos para que funcione el sistema, con las diferentes tarjetas de expansión, hasta formar un sistema de acuerdo a las necesidades específicas de cada usuario.

El bus ISA.- (Industry Standard Architecture), fue el primer bus utilizado en los Computadores Personales y XT de IBM como sistema de arquitectura abierta. Esto permitió la proliferación de clones y una variedad de tarjetas de expansión e interfaces de bajo costo que se encuentran en el mercado. De los buses ISA hay dos versiones: el de 8 bits y el de 16 bits.

El bus ISA de 8 bits.- El bus de 8 bits salió en el año 1982 y se conservó en los computadores XT. Este bus se encuentra en forma de conector en línea de doble lado y tiene 62 contactos, 31 por cada lado. Sus contactos están numerados por un lado como A1 hasta A31 y por el otro lado como B1 hasta B31. En la figura 1.2.2.1.a) se muestra la estructura física, la numeración y el nombre de cada una de las señales.

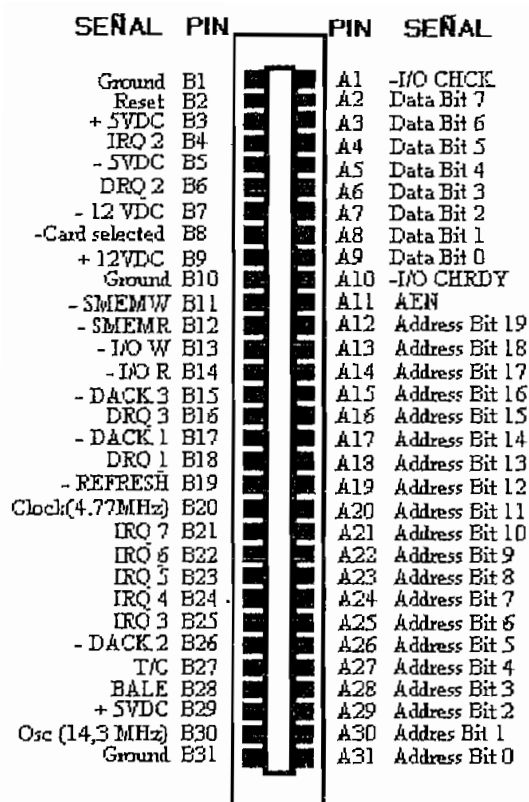


Figura 1.2.2.1 a) Bus ISA de 8

Como se puede observar en la figura anterior, en una ranura se encuentra el bus de datos, el bus de direcciones, las señales de control los voltajes de alimentación. Casi todas estas señales van conectadas al microprocesador a través de circuitos acopladores (drivers o buffers) con el fin de protegerlo de posibles cortocircuitos o conexiones equivocadas. Como podemos ver, en el bus ISA de 8 bits hay un bus de datos de

8 bits y un bus de direcciones de 20

líneas lo que permite un direccionamiento hasta de 1 Mbyte.

El bus contiene seis señales de interrupción (IRQ2 a IRQ7), tres canales de DMA (Acceso Directo a Memoria) y una señal de reloj de 4.77 Mhz. Aunque cada conector en el bus se supone que trabaja de la misma forma. El pin del oscilador (B30), entrega una señal de 14.3 Mhz. Cuando se debe hacer reset en el computador, el pin RESET DRV (B2) reinicia todo el sistema. Cuando se recibe una dirección válida, el pin AEN(A11), le indica al sistema que se puede decodificar esta dirección. El pin -I/O CHCK o I/O Channel Check (A1), avisa a los circuitos de la tarjeta principal, que ha ocurrido un error en la tarjeta de expansión; el pin -I/O CHRDY o Channel Ready (A10) se activa cuando una tarjeta de expansión está lista, si este pin está en nivel bajo (0), el

microprocesador extiende el ciclo del bus generando estados de espera (wait states).

Las seis señales de interrupción por hardware (IRQ2 a IRQ7), son utilizadas por las tarjetas de expansión para demandar atención por parte del microprocesador. Las interrupciones 0 y 1 no están disponibles en el bus ya que ellas tienen las prioridades más altas del temporizador principal y el teclado. Las señales I/O Read y I/O Write indican que el microprocesador o el controlador de DMA quieren transferir datos hacia o desde el bus de datos. Las señales de lectura y escritura de memoria (-MEMP y -MEMW) le indican a la tarjeta de expansión que la CPU o el DMA van a leer o escribir datos a la memoria principal. El bus de las XT tiene tres señales de requisitos de DMA (DRQ1 a DRQ3), que le permiten a la tarjeta de expansión transferir datos hacia o desde la memoria.

El bus ISA de 16 bits.- Con el avance de la tecnología, las limitaciones del bus ISA de 8 bits se evidenciaron rápidamente. Las seis interrupciones disponibles se coparon con la unidad de disco, el disco duro, los puertos seriales y el puerto paralelo, dejando pocas posibilidades de expansión. Así mismo, de los tres canales de DMA, la unidad de disco y el disco duro ocupaban dos, quedando uno solo libre.

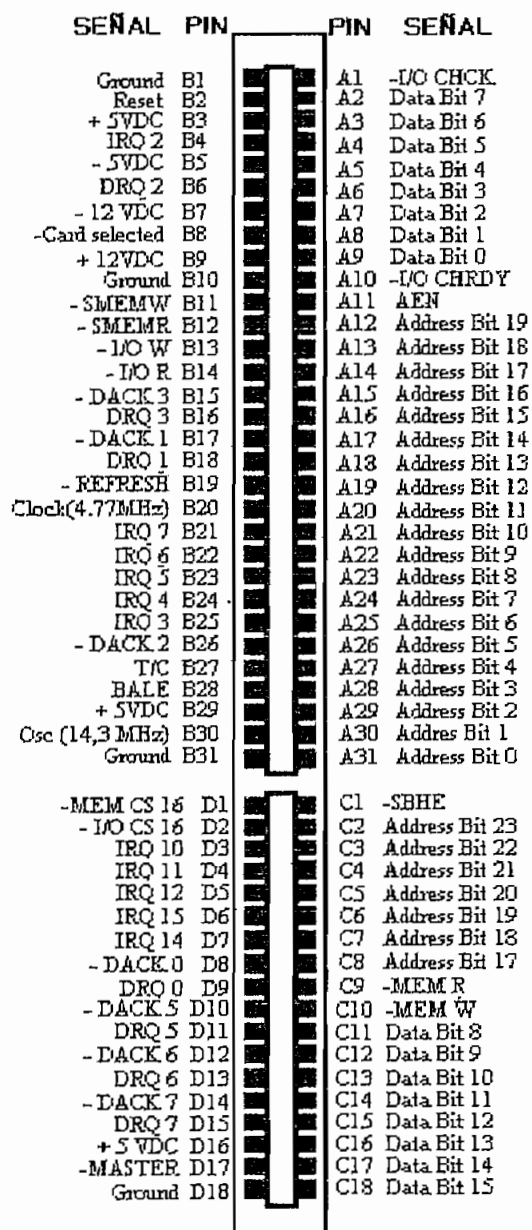


Figura 1.2.2.1.b) Bus ISA de 16

La capacidad de direccionamiento de sólo 1 MB y el bus de datos de 8 bits también se constituyeron en una desventaja, que hacían que los equipos fueran muy lentos para las nuevas aplicaciones generales en el desarrollo de nuevo software. Con la aparición del microprocesador 80286 de 16 bits y los computadores AT en 1984, se diseñó un nuevo bus, el ISA de 16 bits, que utilizaba en gran parte el bus anterior ISA de 8 bits agregándole un segundo conector de 36 pines alineado con el primero con nuevas señales, (figura 1.2.2.1.b) De esta forma, se podían utilizar la gran cantidad de tarjetas periféricas y de expansión que existían en el mercado.

En esencia, se agregaron otros 8 bits de datos, más direcciones, cinco interrupciones, cuatro canales de DMA y algunas señales de control. Así mismo, se incrementó la velocidad a 8.33 Mhz.

Se debe mencionar que el bus ISA de 16 bits o bus AT, fue ampliamente utilizado por los fabricantes de clones y compatibles, éste bus todavía se encuentra en la

mayoría de los computadores personales fabricados actualmente, ya que para él se fabrica una gran cantidad de tarjetas para periféricos y de expansión.

El bus MCA de 32 bits.- Con la introducción de los microprocesadores de 32 bits como el 80386 y el 80486, el bus ISA de 16 bits se quedó nuevamente atrás y se creó la necesidad de un nuevo bus. En 1987, IBM diseñó el bus MCA (Micro Channel Architecture) o Micro Canal para su línea de computadores PS/2.

Su diseño buscaba, además del objetivo de mejorar el rendimiento de los equipos, para el cual se necesitaban fabricar sus propias tarjetas de expansión y periféricos, razón por la cual no tuvo acogida técnica.

La principal diferencia entre los buses ISA y MCA consiste en la configuración física, ya que el último tiene una separación entre los pines de 0.050" con el fin de acomodar más conexiones en el mismo espacio, esto genera una incompatibilidad total entre los dos sistemas.

El bus EISA de 32 bits.- En 1988 y 1989 fue desarrollando el bus EISA (Extended Industry Standard Architecture) de 32 bits para satisfacer las necesidades de mayor velocidad y desempeño que proporcionaban los microprocesadores 80386 y 80486.

Para lograr la compatibilidad entre los dos sistemas ISA y EISA, se diseñó un conector, con diferente altura o nivel de tal manera que la transferencia entre los dos buses se realice automáticamente dependiendo de la forma de la ranura en la tarjeta.

Como se puede observar en la figura 1.2.2.1.c), el bus EISA tiene 30 líneas de

direccionamiento, 32 bits de datos, 15 niveles de interrupción y 7 canales de DMA.

Una de las principales ventajas del bus EISA es su capacidad de manejo autónomo del bus, que en pocas palabras podría explicarse como la capacidad para permitir el intercambio de información entre dos periféricos sin la intervención de la CPU. Aunque la velocidad del bus EISA se conservó en 8.33 Mhz, para permitir la compatibilidad con ISA, su velocidad para transferencia de los datos es mucho mayor debido al aumento en la anchura del bus; esta llega a ser de unos 33 Megabits por segundo.

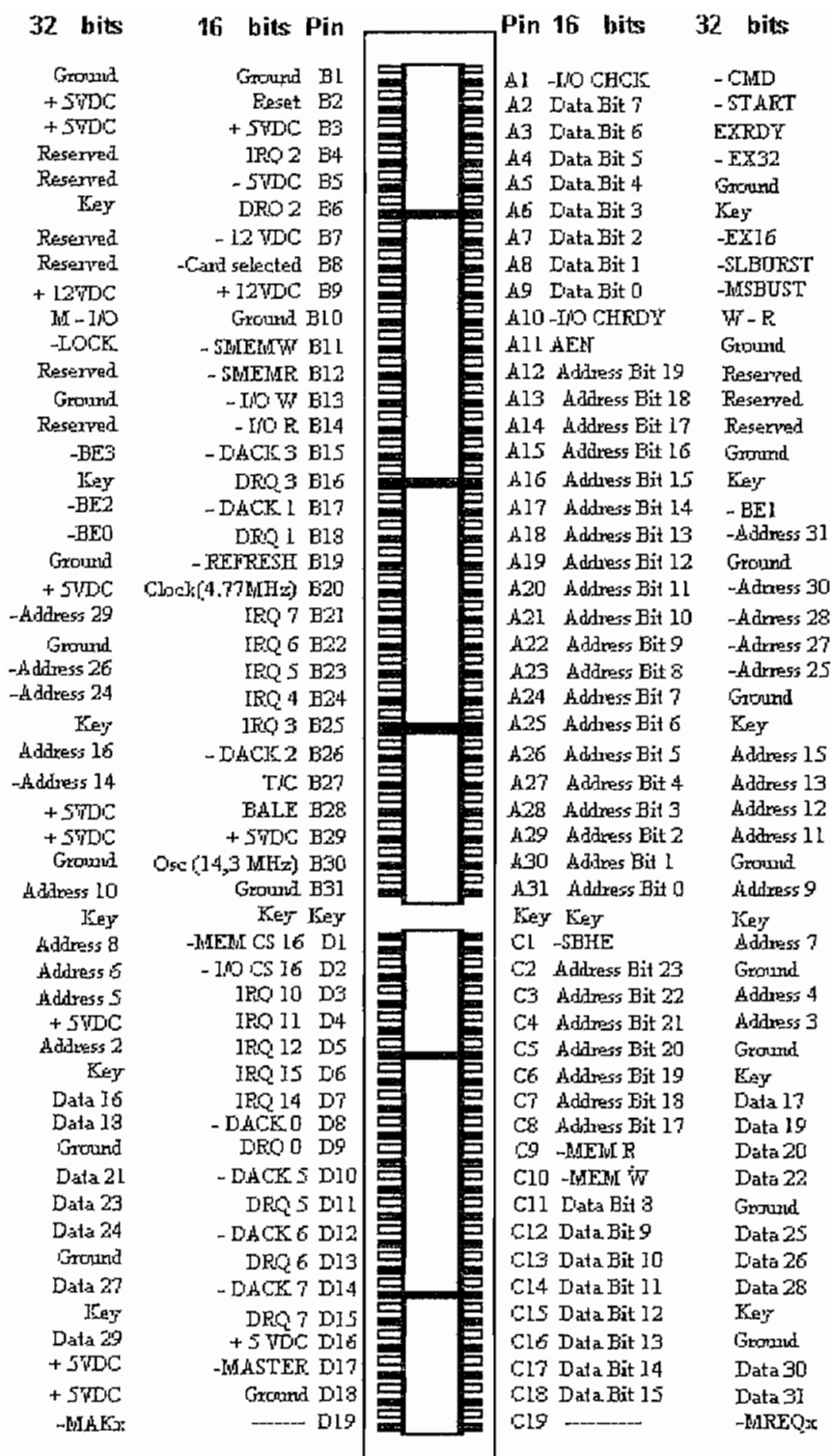


Figura 1.2.2.1.c) Bus EISA

Desafortunadamente, muchas tarjetas de interface tipo EISA no son más rápidas que sus similares tipo ISA, debido a que sus fabricantes no utilizan todas las propiedades del bus. Así mismo, la baja velocidad del disco duro comparada con otros periféricos, es un problema que no depende del bus en sí.

LOS BUSES LOCALES

Los buses locales ofrecen una mayor velocidad en las operaciones que involucran gráficas, vídeo y sonido presentes en la gran cantidad de aplicaciones con multimedia y otras similares, tan populares actualmente. La baja velocidad de los buses ISA, MCA y EISA sólo se podía superar conectando directamente las líneas de datos, las líneas de direccionamiento y las de memoria al microprocesador. De esta forma, se pueden lograr velocidades hasta de 66 Mhz en comparación a los 8 o 10 Mhz de los otros buses. Los dos tipos principales de buses locales son Vídeo Local y el PCI, que se explicarán más adelante.

El bus local de vídeo (VL) bus .- Este tipo de bus fue propuesto en 1992 por la Vídeo Electronics Standards Association y por eso se le llama VESA Vídeo Local Bus, éste bus utiliza el mismo conector de 62 pines del bus MCA, por lo tanto las tarjetas que utilizan éste bus tienen dos conectores, uno en cada extremo y para cada bus. Los conectores o ranuras de la tarjeta madre, se colocan alineados con los conectores o ranuras ISA. El bus VL es expandible de 32 a 64 bits permitiendo su utilización en sistemas con el microprocesador Pentium, por medio de otro conector tipo MCA alineado con el primero.

El bus local PCI .- El último bus es el llamado bus local PCI (Peripheral

Component Interconnect) propuesto por Intel en 1992 y apoyado por la mayoría de los fabricantes de computadores personales, quienes lo han adaptado en los equipos mas recientes, tales como los computadores que utilizan el microprocesador Power PC 604.

El bus PCI, de 32 bits, es expandible a 64 bits y soporta la lógica de 3.3 voltios junto a la normal de 5 voltios. Las ranuras PCI estarán dedicadas para los periféricos más críticos que pueden estar conectados directamente a la tarjeta principal o por medio de tarjetas de interface como son las unidades para multimedia (CD-ROM), tarjetas para gráficas y vídeo, unidades de disco duro y tarjetas para conexión en red.

De todas maneras, los sistemas seguirán incorporando ranuras ISA, EISA y VESA, ya que el bus PCI es un complemento y no reemplaza totalmente los buses tradicionales.

Del estudio realizado anteriormente se puede empezar diferenciando las máquinas por dos cosas: su compatibilidad de programas y su compatibilidad de equipo. La compatibilidad de programas queda determinada por la clase de microprocesador que utilizan.

La compatibilidad de equipo plantea la pregunta, "¿ Se puede tomar una tarjeta de circuitos electrónicos, ponerla en la computadora y que funcione ? " Buscando contestar esta pregunta se observa que los computadores XT, AT y compatibles comparten un bus común que se encuentra en las ranuras de la tarjeta principal para insertar tarjetas de expansión, este bus es del tipo ISA . Sin

embargo, la línea PS/2, utilizan un bus nuevo y completamente incompatible llamado MCA —Micro Channel Architecture (Arquitectura de Micro Canal).

MICROPROCESADOR	BUS DE PC (ISA)	ARQUITECTURA DE MICRO CANAL (MCA)	DE EXTENDED STANDARD ARCHITECTURE (EISA)	INDUSTRY
Intel 8088, 80188, 80186 NECV20, V30	8086, IBM PC, XT, Portátil, PS/2 Modelo 30 y Modelo 50 Compaq Deskpro, Portátil Toshiba T1000,T1200 AT&T 6300	Ninguna de esta categoría	Ninguna de esta categoría	de esta
Intel 80286	IBM AT, XT286, AT/370 Compaq Deskpro 286 Zenith Z248,LP286 SuperSport 286 AT&T 6300+ Toshiba 1600, 3100,3200 AST Premiun/286	IBM PS/2 Modelo 50 IBM PS/2 Modelo 50Z IBM PS/2 Modelo 60	Ninguna en esta categoría	de esta
Intel 80386, 80386SX, 80486	Compaq Deskpro 386, 386S Zenith Z-386 AT&T 6386 AST Premiun/386 ALR Flex Cache 20	IBM PS/2 Modelo 55 IBM PS/2 Modelo 70 IBM PS/2 Modelo 70 IBM PS/2 Modelo 80 IBM PS/2 Modelo 95	ALR Business VEISA Compaq Deskpro 486/33	de esta

Tabla 1.2.2.1. d) Relación entre los tipos de microprocesadores y buses existentes en los diferentes computadores.

De lo expuesto en la tabla anterior se observa que por más nueva que sea una computadora siempre existirá las señales básicas las cuales están agrupadas en los buses y, de los cuales el bus ISA de 8 bits es el más antiguo y por ende el más compatible , ya que toda tarjeta con un microprocesador de la familia INTEL iax86 llevará consigo una ranura para bus tipo ISA.

En consecuencia de todo lo anterior, el tipo de bus en el que se desarrollará el presente trabajo de tesis, es el bus tipo ISA de 8 bits, por ser el más general encontrado en el mercado de los Computadores Personales, a continuación se

realiza un estudio de las señales permitidas en las ranuras de la tarjeta del sistema.

1.2.2.2 DESCRIPCION DE LAS SEÑALES DEL BUS DEL SISTEMA

Todos las ranuras de la tarjeta madre tienen buses con idénticas señales, todas estas tienen niveles lógicos TTL, excepto por las líneas de fuentes y tierras que son provistos por el bus en el conector. El bus del 8088 es conformado por la unión de señales que soportan el acceso directo a memoria, interrupciones, tiempo y control de I/O, lectura-escritura de memoria, generación de estados de espera, refresco de memoria y detección de error. A continuación se da una serie de definiciones que detallan la descripción u funcionamiento de todas la señales de los 62 pines.

DEFINICION DE LAS SEÑALES

OSC (Oscilador)

Esta señal es sólo de salida con una frecuencia de 14,31818 MHz y un período de aproximadamente 70 ns. Tiene una relación de trabajo por ciclo de 50%.

Esta señal es la de más alta frecuencia de todas las otras señales de tiempo y la frecuencia de ésta puede ser ajustada en tiempo con el ajuste de un capacitor que esta localizado en la base de la tarjeta madre, ésta señal puede ser dividida en 4.

Además se debe tener cuidado cuando esta señal es usada como reloj de otro bus ya que puede haber una dessincronización de ésta con otras.

CLK (Reloj)

Esta señal se deriva de la señal del oscilador, esta señal de salida se obtiene de dividir la señal osciladora para tres la cual da una frecuencia de 4,77 MHz. Esta señal no es simétrica con un tiempo en alto de 70 ns. y un tiempo en bajo de 140 ns. Esta señal es sincronizada con respecto a las de control de lectura-escritura en memoria y puede ser usada para generar estados de espera en el bus. Un ciclo de bus es 4 períodos de reloj (aproximadamente 840 ns.)

RESET DRV (MANEJADOR DEL RESET)

Esta señal es únicamente de salida, utilizada para inicializar el sistema lógico tanto en el encendido del equipo, como en una falla de voltaje, además ésta es utilizada para inicializar a los elementos de I/O y traerlos a un estado lógico conocido de operación.

A0 a A19

Señales de salida desde la A0 a la A19, que constituyen el bus de direcciones del sistema, sirven para direccionar las memorias y dispositivos de I/O (A0 es el bit menos significativo y A19 es el más significativo). Estas 20 líneas son manejadas por el microprocesador, como por el controlador de DMA. Con 20 líneas se puede direccionar hasta 1M byte de memoria del sistema, pero no todos los espacios de direcciones son permitidos en el bus del sistema.

El procesador a través del uso de las instrucciones IN y OUT, pueden usar las direcciones fuera de los 64K para localizar los pórticos de I/O, estas direcciones de los pórticos son llevados en su bus a través de las líneas A0 y A15, mientras que las

líneas 16 a 19 no son usadas y están inactivas; sin embargo, un computador personal solamente direcciona pórticos de I/O usando las líneas desde A0 hasta A9, en el rango de 0200H a 03FFH HEX, válidas para el bus del sistema.

D0 a D7

Estas 8 líneas son bidireccionables y constituyen el bus de datos, señales utilizadas para transmitir datos entre el microprocesador, la memoria y pórticos de I/O (D0 es el menos significativo y D7 es el bit más significativo). Cuando el microprocesador inicia un ciclo de escritura los datos son llevados en su bus para la escritura, ya sea en memoria o en pórticos de I/O. Estos datos son validados antes que termine el pulso de las señales de control **IOW** o **MEMW**, dichos pulsos son usados como reloj en el bus de datos.

Además durante los ciclos de DMA el bus de datos es utilizado para transferir información directamente entre los pórticos de I/O y memoria sin la intervención del microprocesador.

ALE (Habilitación de retención de la dirección)

Esta es una señal de salida, manejada por el controlador del bus y es utilizada para indicar que la dirección del bus es válida para iniciar un ciclo del bus. Esta señal se activa en alto antes que el bus de direcciones empiece y se desactiva después que éste bus inició.

Esta señal es utilizada para retener la información del bus de direcciones del microprocesador, se debe notar que el bus de datos no contiene información de direcciones válidas y , por lo tanto, la señal **ALE** no puede ser utilizada para demultiplexar las direcciones.

Esta señal **ALE** es una buena señal de referencia para sincronizar el inicio de los ciclos de cada bus, tomando en cuenta que ésta señal no se activa durante los ciclos de DMA.

I/O CH CK (Chequeo de canal de entrada/salida)

Esta es una señal sólo de entrada que se activa en nivel bajo, utilizada para el reporte de una condición de error en el interface añadido. Cuando esta señal detecta un nivel lógico bajo generará una interrupción no mascarable (**NMI**) al microprocesador.

I/O CH RDY (Lectura del canal de entrada salida)

Esta es una señal solamente de entrada que se utiliza para extender la longitud de los ciclos del bus, ya que cuando la memoria o los pórticos de I/O, no son demasiado rápidos para responder a un ciclo normal del bus se pueden añadir hasta 4 ciclos de reloj en el bus del sistema.

Si la memoria o un pórtico de I/O quiere extender un ciclo del bus, forzará la señal I/O CH RDY a un nivel lógico bajo cuando se decodifique su dirección y reciba los comandos MEMR, MEMW, IOR ó IOW.

Esta señal debe ser cuidadosamente controlada ya que sólo añade estados de espera al ciclo del bus en incrementos de 210 ns hasta un máximo de 840 ms.

IRQ2 al IRQ7 (Pedido de interrupción del 2 al 7)

Estas 6 señales son sólo de entrada, y utilizadas para pedir una interrupción al micro 8088 desde el bus del sistema, y van directamente al controlador de interrupciones

de la tarjeta principal del microprocesador. Los programas básicos de configuración del sistema inicializan al controlador de interrupciones y activan al IRQ2 como el de alta prioridad y el IRQ7 como el de baja prioridad.

IOR (Lectura de entrada-salida)

Esta es una señal de salida del controlador del bus, utilizada para indicar la presencia de la dirección de algún pórtico de I/O en este bus de direcciones. La dirección del pórtico de I/O debería responder localizando sus datos de lectura en el bus del sistema, aproximadamente 30 nanosegundos después del pulso de la señal IOR para asegurar que el proceso de transferencia de datos sea válido.

Cuando ocurren ciclos de DMA, la señal **IOR** es controlada por el controlador de DMA. En este caso el bus de direcciones no contiene ninguna dirección de pórticos de I/O pero, en cambio contiene una dirección de memoria donde los datos de los pórticos serán escritos. Los pórticos de I/O no son seleccionados por una dirección sino por una señal .

IOW (Escritura de I/O)

Esta señal se activa en nivel lógico bajo, es una señal sólo de salida y es manejada por el controlador del bus. Indica las direcciones , y los datos a ser escritos dentro del pórtico de I/O, en sus respectivos buses. Cuando la señal se activa en bajo el bus de datos no es válido así que los datos del pórtico deben ser retenidos usando el pulso de esta señal.

Cuando ocurre un ciclo de DMA, la señal IOW es activada por el controlador de DMA

y es utilizada para escribir los datos desde la memoria ,los mismos que son actualizados en el bus de datos.

MEMW (Escritura de memoria)

Esta es una señal que es activada en nivel lógico bajo, utilizada para escribir los datos del bus del sistema a la memoria, es activada por el controlador del bus e indica que el bus de direcciones contiene la localidad de memoria en la cual los datos van ha ser escritos.

Además durante los ciclos de DMA esta señal es manejada por el controlador del DMA. y es utilizada para escribir datos en el bus desde los pórticos de I/O a memoria.

MEMR (Lectura de memoria)

Es una señal que se activa en nivel lógico bajo, es sólo de salida utilizada cuando se requiere una lectura de datos de memoria, esta señal es activada por el controlador del bus. Indica que bus de direcciones contienen una dirección válida de memoria y que la localidad de memoria especificada es utilizada para la lectura de datos del sistema.

Además durante los ciclos de DMA ésta señal es activada por su controlador e indica que la dirección de la localidad de memoria debe manejar el bus de datos.

DRQ1 al DRQ3 (Requerimiento de DMA del 1 al 3)

Estas tres líneas son activadas en un nivel lógico alto, son señales de entrada utilizadas por el interfaz cuando requiere ciclos de DMA. Si un elemento o interfaz

quiere transferencia de datos entre él y la memoria sin intervención del microprocesador éste requerimiento es inicializado por la activación de la línea DRQ. Estas líneas van directamente al controlador de DMA en la tarjeta del microprocesador donde son priorizadas antes que el ciclo del DMA sea dado. El programa de configuración básica del sistema del computador personal activa el controlador del DMA, de modo que la señal DRQ1 es la de mayor prioridad y la señal DRQ3 es la de menor prioridad.

DACK0 A DACK3 (Actualización del DMA de 0 a 3)

Estas cuatro señales son activadas en nivel lógico bajo, son señales de salida utilizadas por el controlador para reconocer las peticiones de DMA. La señal DACK0 es utilizada para el refresco de memorias dinámicas.

AEN (Habilitación de direcciones)

Esta señal es de salida activada en nivel lógico alto utilizada por el controlador DMA. Indica que los ciclos de DMA están en proceso, en la tarjeta del microprocesador esta señal es usada para deshabilitar los buses de dirección y de control, además habilita el bus de control y direcciones del controlador DMA.

En el bus del sistema, su propósito es deshabilitar las direcciones de los pórticos de I/O decodificados durante los ciclos DMA, así que las direcciones de memoria no son utilizadas tal como las direcciones de los pórticos de I/O durante los ciclos de DMA. Esto es posible porque las señales IOW e IOR se activan con direcciones de memoria durante los ciclos de DMA.

TC(Contador de terminal)

Esta señal es activada en alto, es una señal de salida utilizada por el controlador DMA, para indicar que ha terminado una transferencia de datos a un bloque DMA.

POLARIZACION DEL BUS

Además de las señales descritas anteriormente el bus del sistema provee de los niveles de voltaje para la polarización de las tarjetas externas conectadas a cualquiera de las ranuras. (Como se pudo observar en la figura 1.2.2.1. a)

+ 5 V DC

Este nivel de polarización del sistema es habilitado en dos pines (B3, B29) ubicados al borde del conector y, es regulado a +/- 5% (+4.75 a +5.25 VDC).

+ 12 V DC

Este nivel de polarización del sistema es habilitado en un pin (B9) ubicado al borde del conector y, es regulado a +/- 5% (+11.4 a +12.6 VDC).

- 5 V DC

Este nivel de polarización del sistema es habilitado en un pin (B5) ubicado al borde del conector y, es regulado a +/- 10% (-4.5 a -5.5 VDC).

- 12 V DC

Este nivel de polarización del sistema es habilitado en un pin (B12) ubicado al borde del conector y, es regulado a +/- 10% (-10.8 a -13.2 VDC).

GND (TIERRA)

El sistema provee de la señal de tierra en tres pines del conector. (B1, B10 y B31)

1.3 ENLACE DE LA TARJETA AL PC

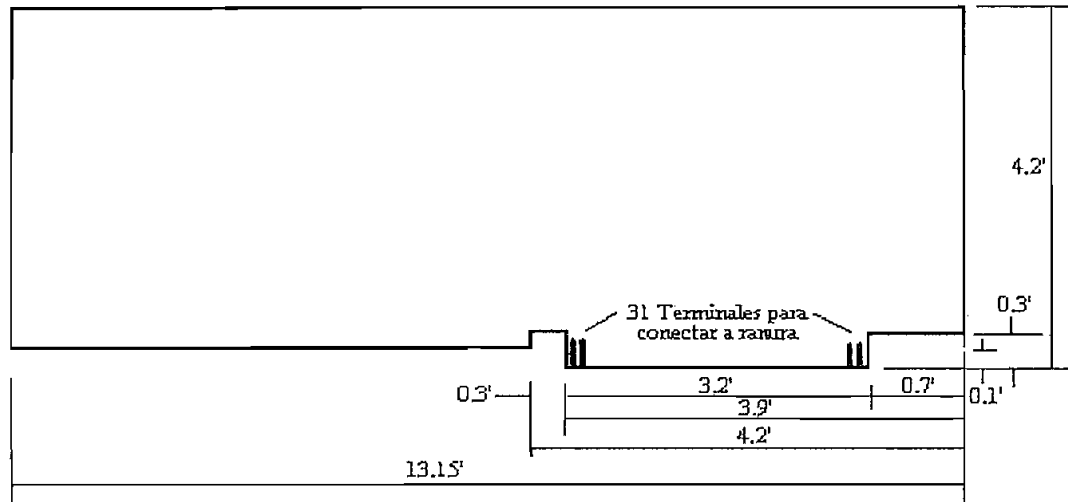
Luego de conocer la función de cada una de las señales y su posición física en la ranura de bus tipo ISA y, para establecer un adecuado enlace entre una tarjeta externa y el computador personal, a continuación se realiza un estudio de los aspectos físicos en la construcción de la tarjeta, tales como las características mecánicas, las dimensiones físicas estándares para la ranura del bus tipo ISA de 8 bits; a demás, se determina la dirección de cada uno de los pórticos utilizados y la dirección de pórtico permitida para la conexión de una tarjeta prototipo.

1.3.1 EL CONECTOR

La mayoría de diseños de interfaces se hacen en base a tarjetas predefinidas que deben ingresar en una ranura del sistema, por lo general, toda tarjeta madre de un computador personal posee ranuras para bus tipo ISA (de 8 bits) para la conexión de tarjetas externas. Las ranuras o conectores para éste bus bits son capaces de llevar 62 señales (31 en cada lado), las que pueden ser utilizadas por tarjetas de diseño. El espacio que existe entre cada ranura dentro de la tarjeta madre es de una pulgada.

Además cada una de las señales están dispuestas en los pines de la ranura, los cuales están espaciados en un décimo de pulgada, y cada ranura puede soportar una tarjeta con un área de aproximadamente 10.42cm * 33.02cm.

El la figura 1.4 a) se indican los tamaños estándares de la tarjeta para que estos calcen justo en las ranuras disponibles para este bus en la tarjeta madre del sistema.



* Dimensiones dadas en pulgadas

Figura 1.4.a) Dimensiones normalizadas para tarjetas tipo ISA de 8 Bits

TAMAÑO DE LA TARJETA

Las dimensiones máximas de una tarjeta pueden apreciarse en la siguiente figura 1.4.a). La tarjeta debe ser construida de un material con un espesor mínimo de 0.060 pulgadas para asegurar un buen contacto en el conector, además los bordes del conector de la tarjeta experimental deben ser de buen material conductor para tener un contacto confiable.

POTENCIA DE LA TARJETA MADRE

Hay 4 niveles de potencia permitidos en cada ranura de la tarjeta madre. En la tabla 1.4.b) se dan los niveles de tolerancia de potencia permitida para cada tarjeta adicional, asumiendo que existe una configuración estándar.

FUENTE DC (VDC)	MAX Vdc (VDC)	MIN Vdc (VDC)	Corriente (A) (Amps)	POTENCIA (Watts)	Corriente Típica/ranura (Amps)
+5	5.25	4.80	7	35	0.7
-5	5.50	4.60	0.3	1.5	0.03
+12	12.6	11.52	2	24	0.10
-12	13.2	10.92	0.25	3	0.05

Tabla 1.4.b) Niveles de potencia permitida para una tarjeta adicional

DESACOPLAMIENTO DE POTENCIA

Un problema común en el diseño de circuitos electrónicos es la inapropiada distribución y desacoplamiento de potencia. La mayoría de elementos de estos circuitos tienen diferentes requerimientos de potencia, que dependen de la operación que estén realizando en ese momento. El propósito de introducir elementos de desacoplamiento es suministrar, la potencia requerida de un elemento por corto tiempo. Puesto que el sistema de potencia con sus cables y alambres añaden inductancia a la fuente, esta no puede responder los transitorios de potencia requeridos. Para resolver este problema al diseñar se añaden capacitores de desacoplamiento en los puntos claves, para que estos transitorios de potencia sean absorbidos por ellos y no por la fuente.

Para grandes y lentas fluctuaciones de potencia, un banco de capacitores son usados para almacenar la potencia requerida. Estos capacitores de desacoplamiento son usualmente capacitores de tantalio con capacitancias de 8 a 20 microfaradios. Estos elementos deberían ser colocados entre el nivel de polarización y tierra, en las cercanías del conector de la tarjeta y en los bordes de la tarjeta.

En alta frecuencia un pequeño transitorio de potencia se presenta, por lo que se deben utilizar capacitores de desacoplamiento a alta frecuencia, estos elementos son típicamente de tantalio en el rango de 0.1 y 0.01 microfaradios.

1.3.2 PORTICOS DEL COMPUTADOR PERSONAL

En el diseño del Computador Personal original, IBM incorporó un direccionamiento de 10 bits, desde los bits A0 hasta A9, para un total de 1024 puertos direccionables. El mapa de los puertos esta dividido en dos partes: la primera de ellas (512 puertos) se localiza en la tarjeta madre, mientras que la segunda (otros 512 puertos) están orientados hacia tarjetas que se insertan en las ranuras del computador. La tabla 1.4.2 a) muestra algunas de las direcciones que emplean los periféricos más comunes, así como el rango de direcciones que tenemos disponibles para el desarrollo de experimentos.

RANGO Hexadecimal	UTILIZACION
000-00F	Chip DMA 8237A-5
020-021	Chip de Interrupciones 8259A
040-043	Temporizador 8253-5
060-063	PPI 8255A-5
080-083	Registro de página DMA
0AX	Registro de máscara NMI
0CX	Reservado
0EX	Reservado
100-1FF	No Utilizable
200-20F	Control de Juegos
210-217	Unidad de Expansión
220-24F	Reservado
278-27F	Reservado (Adaptador paralelo secundario)
2F0-2F7	Reservado
2F8-2FF	Comunicaciones Asincrónicas
300-31F	Tarjetas de Prototipos
320-32F	Disco Duro
378-37F	Impresora
3B0-3BF	Monitor IBM Monocromo
3C0-3CF	Reservado
3D0-3DF	Color/Gráficas
3E0-3F7	Reservado
3F0-3F7	Unidad de Disco Flexible
3F8-3FF	Comunicaciones Asincrónicas.

Tabla 1.4.2.a)

CAPITULO II

DISEÑO DEL HARDWARE

En el presente capítulo se hacen las consideraciones necesarias de hardware para realizar el proyecto de tesis, especificaciones cubiertas durante el desarrollo de la tarjeta ADAIO, la cual está compuesta básicamente por bloques funcionales de conversión, los que serán oportunamente detallados, dando una explicación de las consideraciones tomadas para su diseño.

En lo que al hardware respecta, se analizan primeramente las señales utilizadas por la tarjeta prototipo ADAIO, luego se hace una descripción de la construcción de cada uno de los bloques de conversión, revisando cada uno de los elementos utilizados y planteando las soluciones que permitieron resolver los problemas presentados en el transcurso del proyecto.

Al final de este capítulo se plantean todas las consideraciones previas que se tomaron para la realización del circuito impreso.

2.1 ESPECIFICACIONES INICIALES

Como se mencionó en el capítulo I, se implementará una tarjeta para una ranura de extensión con bus tipo ISA, la que contiene señales suficientes y elementales, que se encuentra en todos los Computadores Personales compatibles, logrando de esta manera una universalidad y sobre todo una compatibilidad en hardware, con modelos anteriores que utilicen microprocesadores de la familia INTEL 80x86. Para los requerimientos básicos en cuanto a software, se plantea como

mínimo un computador que posea el programa Windows 3.1 en modo extendido o versiones superiores de windows.

Los pórtilos analógicos tanto de salida como de entrada funcionan con voltajes normalizados entre 0 y +10 VDC, el flujo de datos tanto de entrada como de salida del Computador Personal controla el PPI, por lo que se optimizará el uso de los diferentes pórtilos por medio de software.

2.1.1 ESTUDIO DE LA INTERFAZ PROGRAMABLE

A continuación se realiza un estudio del PPI utilizado como parte fundamental del desarrollo de la tesis y puesto que es un objetivo del presente trabajo la aplicación del mismo, para la adquisición y emisión de datos.

DESCRIPCION GENERAL

Este circuito integrado es fabricado por INTEL para utilizarse como un circuito de soporte del microprocesador; la realidad es que ha tenido gran acogida y utilidad en sistemas en los cuales se requiere del manejo de muchas líneas de entrada/salida. En la figura 2.1.1.a) se muestra la distribución de pines y la organización interna de esta pastilla, la cual contiene tres puertos paralelos programables de entrada/salida, de ocho bits cada uno.

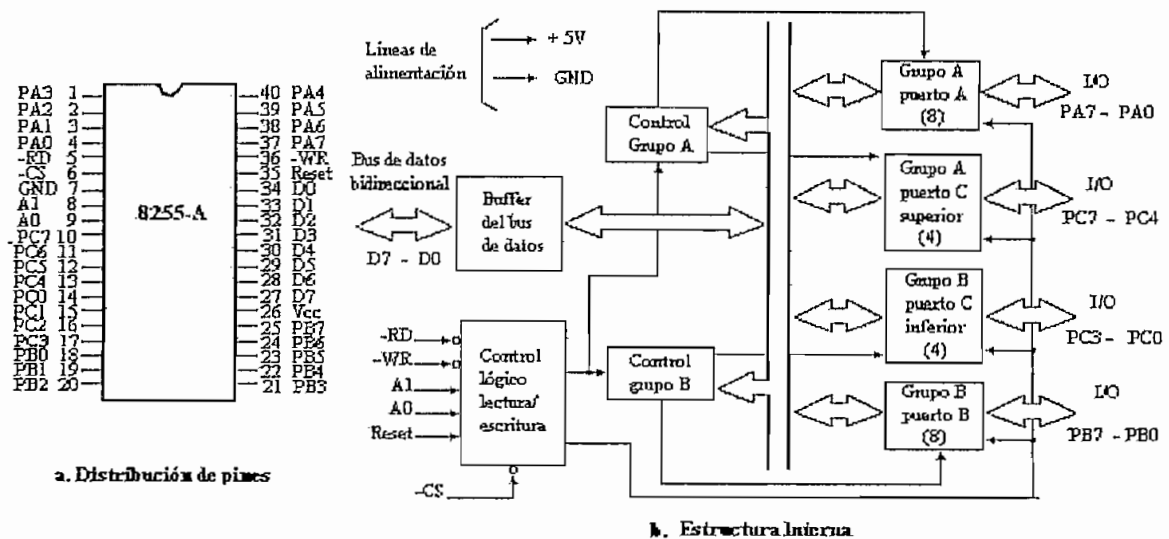


Figura 2.1.1.a) Interfase Programable (PPI) 8255

Hay que observar que 24 de los 40 pines del 8255 son líneas de entrada/salida. Más específicamente. PA0 -PA7 corresponden al puerto A, PB0-PB7 al puerto B y PC0-PC7 al puerto C. Las 8 líneas del puerto C se dividen en dos grupos iguales, cada uno de 4 líneas (PC7-PC4 y PC3-PC0) ;estos grupos de líneas pueden trabajar en combinación con los puertos A y B o como dos puertos individuales de 4 bits.

En la figura 2.1.1.a) se puede observar que la estructura interna del PPI se encuentra dividida en dos grupos: el A y el B, cada uno de los cuales tiene su propio elemento de control. El grupo A está compuesto por el puerto A y los cuatro bits más significativos del puerto C, mientras que el grupo B esta compuesto por el puerto B y los cuatro bits menos significativos del puerto C. El sentido (entrada, salida o bidireccional) y la función (datos o control) de las líneas de todos los puertos se programan durante la operación normal del dispositivo mediante una palabra de control de 8 bits que envía la Unidad Central de Proceso al PPI.

Internamente, el 8255 dispone de cuatro registros de ocho bits cada uno, tres de los cuales se dedican al almacenar la información que entra o sale por los puertos A,B y C. El cuarto se denomina registro de control, éste se dedica a realizar las funciones de control; programando este registro se configuran los grupos A y B, definiéndose el comportamiento de los puertos y el funcionamiento general del dispositivo.

FUNCIONES DE LOS PINES.- Además de las 24 líneas de entrada/salida asociadas a los puertos A,B y C, la función de las líneas restantes del 8255 se

puede resumir en los siguientes términos:

Vcc y GND. Líneas de polarización del circuito integrado: la fuente de alimentación ($V_{cc}=5V$) y el punto de referencia ($GND=0V$)

D7 a D0. Líneas de datos, comunican el 8255 con el bus bidireccional de datos del sistema microprocesador.

/WR (Write). Se activa en bajo, y permite que la CPU realice una operación de escritura sobre el PPI.

/RD (Read). Se activa en bajo, y permite que la CPU realice una operación o ciclo de lectura sobre el PPI

/CS (Chip Select). Se activa en bajo, y se destina para la activación del circuito integrado. El 8255 se comunica con la CPU cuando $CS=0$ o se aísla cuando $CS=1$.

RESET. Se activa en alto, inicializa el PPI y borra (sitúa en 0's) todos los registros internos, incluyendo el de control, configurando como entradas todas la líneas de los puertos.

A0 y A1. Estas señales, en combinación con /RD y /WR seleccionan el puerto o registro del 8255 sobre el cual se va a realizar una operación de lectura o escritura; normalmente, se conectan a las líneas A0 y A1 del bus de direcciones del sistema. La tabla 2.1.1.b) resume el sentido de las transferencias de información que se realiza entre el bus de datos del sistema microprocesador y los distintos registros internos del 8255, de acuerdo con el estado lógico de las señales de control A1, A0, /RD, /WR y /CS.

**Transferencia de información entre
CPU y el PPI 8255**

A1	A0	-RD	-WR	-CS	OPERACION
0	0	0	1	0	A => Bus de Datos
0	1	0	1	0	B => Bus de Datos
1	0	0	1	0	C => Bus de Datos
1	1	0	1	0	Condición inválida
0	0	1	0	0	Bus de datos => A
0	1	1	0	0	Bus de datos => B
1	0	1	0	0	Bus de datos => C
1	1	1	0	0	Bus de datos => Control
X	X	X	X	1	Bus de datos en Hi-Z
X	X	1	1	0	Bus de datos en Hi-Z

X = no importa

Tabla 2.1.1.b)

El registro de control.- En el momento de aplicar potencia por primera vez al circuito, el 8255 se encuentra en estado de reset, por lo tanto todas las líneas de los puertos y del bus pueden dar datos flotantes o en estado de alta impedancia (Hi-Z), por esta razón, antes de utilizar los puertos de entrada y de salida, el circuito integrado debe inicializarse escribiendo en el registro de control un valor de 8 bits, denominado **palabra de control**, la cual configura el funcionamiento general del PPI, el cual a su vez admite tres modos diferentes de funcionamiento denominados Modo 0, Modo 1 y Modo 2.

En la figura 2.1.1.c) se resume la función de cada uno de los bits que componen el registro de control. Este registro solamente puede ser escrito, es decir, no es posible leer su contenido desde la CPU. Cuando el bit más significativo (D7) es 1, el registro de control determina el modo de trabajo de los tres puertos según los restantes bits. En caso de que éste bit sea cero, los restantes bits del registro de

control se utilizan para sacar un 1 o un 0 por las líneas del puerto C.

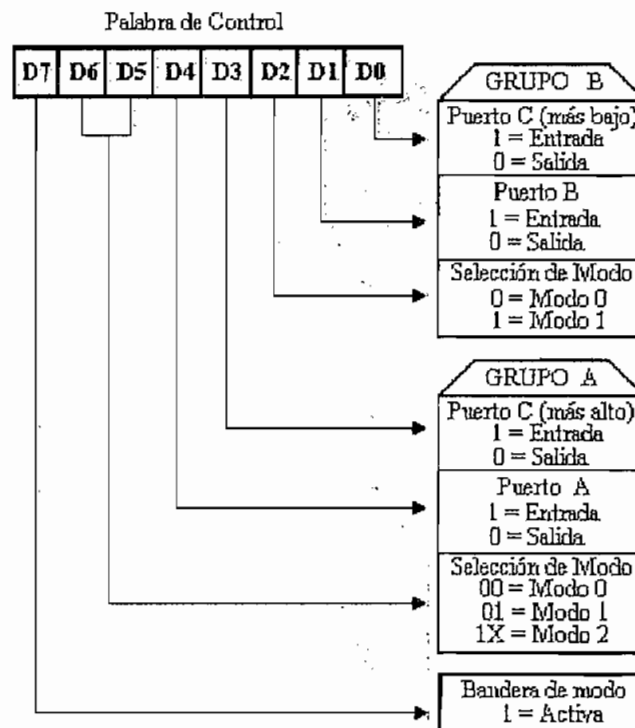


Figura 2.1.1.c) Palabra de Control

En el **Modo 0** (entradas/salidas básicas), cualquiera de los puertos A,B y C puede trabajar como entrada o como salida; los datos son escritos hacia o leídos desde un puerto específico. En este modo, las entradas y salidas tienen 16 configuraciones posibles; en la figura 2.1.1.d) se ilustran dos de ellas, al igual que las palabras de control correspondientes. En el primer caso, actúan como entradas tanto el pódico A, como el pódico C, en cambio el pódico B actúa como salida. En el segundo, A, B y las dos mitades de C actúan como puertos de entrada.

Una característica importante de este modo es que las líneas en los pódicos de salida tienen capacidad para retener el dato, hasta que se escribe un nuevo dato, mientras que las líneas en los pódicos de entrada no poseen esta

propiedad, es decir el dato de interés deben estar presente al momento de la lectura del puerto.

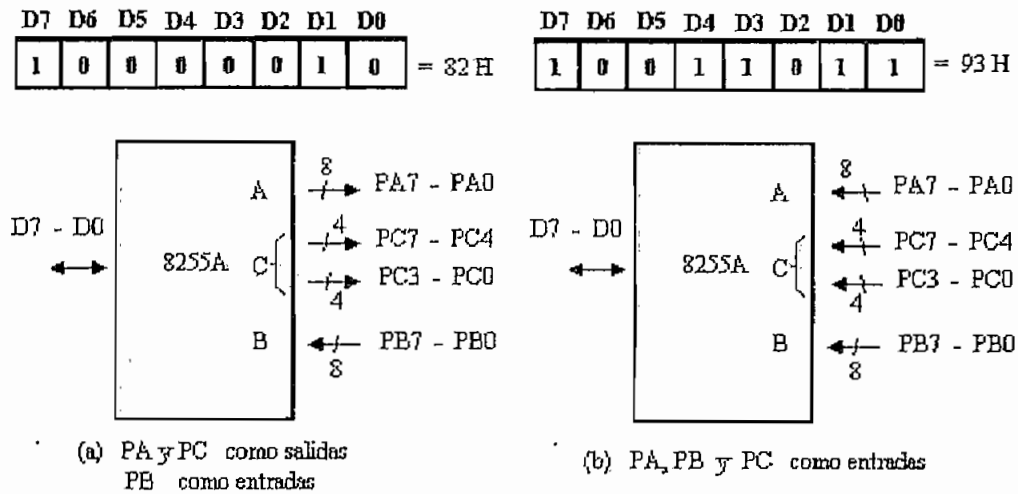


Figura 2.1.1.d) PPI 8255 configurado en modo 0

En el Modo 1 (Entradas/salidas validadas), los puertos A y B pueden trabajar como entradas o salidas, mientras las líneas del puerto C controlan la transferencia de datos entre los periféricos y el PPI. Una parte del puerto C (líneas PC3, PC4 y PC5) están asociadas con el puerto A, mientras que otras (PC0, PC1 y PC2) proporcionan señales de control para el puerto B. Las líneas no utilizadas de C (PC6 y PC7) pueden actuar como líneas I/O convencionales. En este modo, las entradas y salidas tienen la propiedad de retener la información

Cuando los puertos están configurados como ENTRADAS, (figura 2.1.1.e), allí la señal /STB (Strobe o validación) sirve para cargar en los registros internos del PPI el dato que el periférico envía hacia el puerto A o B. Al cargarse un dato en los puertos del PPI, se activa la señal de conformidad IBF (Input Buffer Full), la cual indica al periférico que el buffer de entrada de datos está lleno. La CPU

explora periódicamente las líneas IBF y cuando encuentra alguna activa (alta), reconoce que en el puerto que controla hay un dato disponible. Al leer ese puerto, las líneas IBF y STB se desactivan. Las líneas INTR se utilizan para interrumpir la CPU cuando el puerto (A o B) dispone de información y está listo para ser leído.

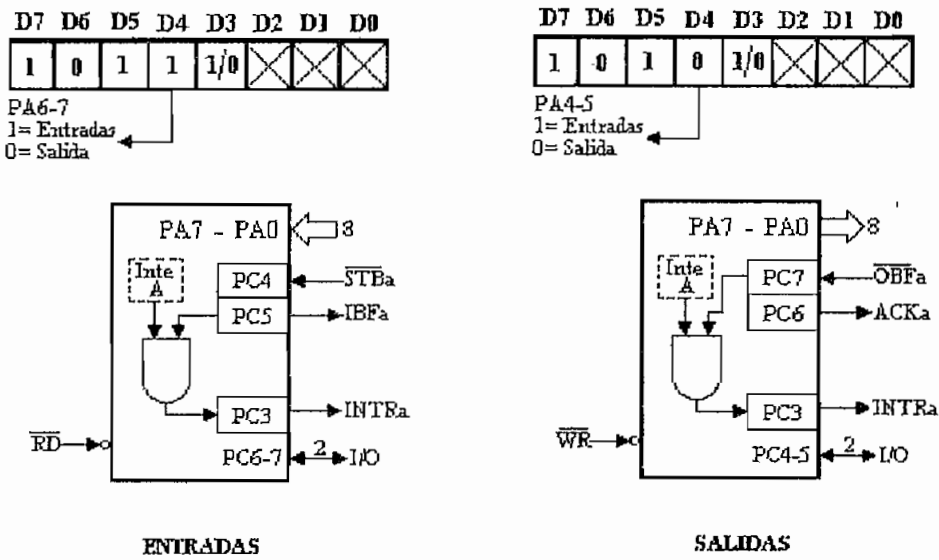


Figura 2.1.1.e) Programación del registro de control en el modo 1

Cuando los puertos están configurados como SALIDAS, la línea /OBF (Output Buffer Full), es una señal que la CPU coloca en un nivel lógico bajo, indicando que escribió un dato en el puerto correspondiente, y éste debe ser leído por el periférico. Esta línea sólo retornará a un estado lógico alto cuando el periférico responda colocando la línea /ACK en un estado lógico bajo, indicando que recibió el dato. Aquí, también las líneas INTR se utilizan para interrumpir la CPU, pero cuando el periférico aceptó el dato. Los dos grupos pueden ser configurados independientemente como entradas o salidas, admitiendo entonces cuatro combinaciones diferentes.

En el **Modo 2** (bus bidireccional), el puerto A puede trabajar en forma bidireccional, es decir actuar como entrada y salida de datos, mientras una parte de las líneas del puerto C (PC3 a PC7) controlan el diálogo con los periféricos, como se puede ver en la figura 2.1.1.e), las tres líneas restantes del puerto C pueden actuar como líneas I/O convencionales o como líneas auxiliares del puerto B, y éste último puede actuar en el Modo 0, o en el Modo 1. En el Modo 2, tanto las entradas como las salidas tienen cerrojo (capacidad para retener los datos).

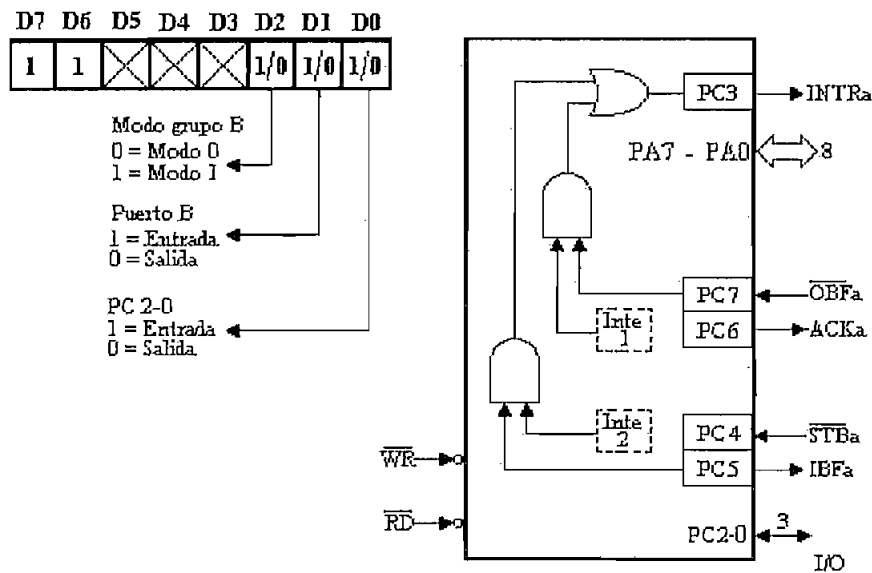


Figura 2.1.1.e) PPI configurado en Modo 2

De lo mencionado anteriormente, se puede determinar que para el presente trabajo de tesis, el modo más idóneo para la programación del PPI es el modo 0, porque éste permite una gran flexibilidad en la programación para poder seleccionar el número de entradas y salidas activas deseadas, para lo cual se utiliza el pórtilo C como líneas de control, de esta manera: las líneas del bloque inferior del pórtilo C actúan como líneas de control del pórtilo de entrada

análogo/digital ubicado en el pórtico B y el pórtico C superior como líneas de control del pórtico salida digital/análoga ubicado en el pórtico A.

Se ha determinado que para el desarrollo del proyecto los registros del PPI se configuren de la siguiente manera: a) un registro de dirección para el módulo de salidas análogas (denominado PA) que es el registro base; b) un registro de dirección para el módulo de entradas análogas (denominado PB) que es el registro base más uno; c) un registro de dirección para la decodificación de las diferentes entradas-salidas análogas (denominado PC), que es el registro base más dos, y, d) un registro de control el cual ha de llevar la palabra 82H colocado por programa de esta manera se inicializa el modo de funcionamiento de los pórticos anteriores, este registro denominado palabra de control, es el registro base más tres.

2.2 REQUERIMIENTOS DE HARDWARE

A fin de determinar los requisitos para establecer un enlace entre las tarjetas externas prototipo con el hardware del sistema a continuación se mencionará, como trabaja el sistema de un computador compatible con pórticos I/O.

Como bien se conoce que la mayoría de elementos y adaptadores en el Computador Personal son controlados utilizando pórticos de I/O, éstos pórticos son direccionados mediante registros de memoria del microcontrolador. Los datos pueden ser enviados a estos pórticos utilizando la instrucción OUT del conjunto de instrucciones del microcontrolador, o también pueden ser leídos de estos pórticos utilizando la instrucción IN.

La arquitectura del microcontrolador soporta un espacio de direcciones asignado

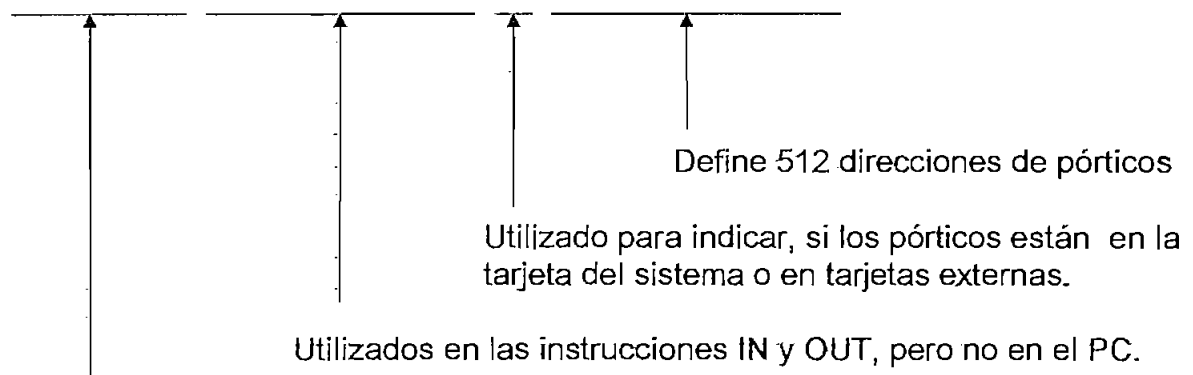
a p rticos de I/O desde 0 a 65.535 (0 H a FFFF H), pero en el dise o del computador personal no se utiliza todo el espacio direccionable (2^{16}), solamente se emplean los 10 bits m s bajos, de esta manera, los bits del 0 al 9 son utilizados para decodificar la direcci n de p rticos.

El bit 9 tiene especial significado para el direccionamiento de p rticos de I/O, cuando este bit esta inactivo (0L), los datos de una tarjeta externa no pueden ser le dos en el bus del sistema y, por lo tanto, se habilitan solamente los datos de los elementos y de los p rticos que est n en la tarjeta del sistema. Cuando este bit esta activo (1L), habilita los datos de las tarjetas externas, esto significa que de los 1024 (2^{10}) p rticos direccionables en el computador, los p rticos de entrada de datos se dividen equitativamente en 512 (2^9) p rticos que deben existir s lo en la tarjeta del sistema y 512 p rticos deben existir s lo en las tarjetas externas.

Se debe tomar en cuenta que esta restricci n no es aplicada a los p rticos de salida, cualquiera de los 1024 p rticos direccionables pueden ser usados como direcciones de p rticos de salida, sin embargo, las direcciones de p rticos de salida usados en la tarjeta del sistema no deber n ser reproducidas en las tarjetas externas.

En la figura 2.2.1, se ilustra como se utiliza en el computador el direccionamiento de p rticos:

19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



No son utilizados para el direccionamientos de p[or]ticos en la arquitectura del 8088.

Figura 2.2.1

Las direcciones de p[or]ticos de I / O puede ser dividido en dos partes. La primera parte es el espacio de direcciones hexadecimal de 0000 a 01FF, esta parte reside en la tarjeta del sistema, estas direcciones de p[or]ticos son utilizados para direccionar elementos integrados en la tarjeta del sistema. La segunda parte de las direcciones de p[or]ticos, es el espacio de direcciones hexadecimal de 0200 a 03FF, estos espacios de direcciones son utilizados para la decodificaci[on] de p[or]ticos sobre el bus del sistema y habilitar las tarjetas externas.

De lo mencionado anteriormente se puede determinar que para realizar un enlace entre las se[~]ales utilizadas por el computador personal y una tarjeta externa se tiene que realizar un adecuado direccionamiento de los p[or]ticos de I/O para lo cual se debe construir un circuito de decodificaci[on] tomando en cuenta que la direcci[on] asignada a tarjetas externas prototipo esta dada apartir de la 300H a 31FH, como se mencion[on] en el cap[itu]lo 1 (ac[ap]ite 1.4.2).

2.3 SEÑALES Y DIRECCIONES UTILIZADAS POR LA TARJETA ADAIO

Las señales disponibles en las ranuras de expansión con bus tipo ISA de ocho bits para tarjetas de extensión del computador personal son 62, de las cuales la tarjeta ADAIO utiliza las señales indicadas en la tabla 2.3.

Conector bus ISA (#)	Tipo	A (Lado de Componentes)	Tipo	B (Lado de Sueidas)
1			--	GND
2	I/O	Dato D 7	O	RESET DRV
3	I/O	Dato D 6	--	+ 5 VDC
4	I/O	Dato D5		
5	I/O	Dato D4	--	- 5 VDC
6	I/O	Dato D3		
7	I/O	Dato D2	—	- 12 VDC
8	I/O	Dato D1		
9	I/O	Dato D0 (LSB)	--	+ 12 VDC
10		Señal no Usada	--	GND
11	O	AEN		
12				
13			O	- IOW
14			O	- IOR
15 a 21		Señales no Usadas		
22	I/O	Dirección A9		
23	I/O	Dirección A8		
24	I/O	Dirección A7		
25	I/O	Dirección A6		
26	I/O	Dirección A5		
27	I/O	Dirección A4		
28	I/O	Dirección A3		
29	I/O	Dirección A2	—	+ 5 VDC
30	I/O	Dirección A1		
31	I/O	Dirección A0	--	GND

Tabla 2.3.

2.4 DESCRIPCION DEL HARDWARE

El proyecto de la tarjeta ADAIO tiene como bloques funcionales, básicamente 3, (figura 2.4) estos son: a) Bloque de decodificación de direcciones, b) Bloque de Conversión Digital/Análoga (de salidas), c) Bloque de Conversión Análoga/Digital

(de entradas); pero en el transcurso del proyecto se ha desarrollado la implementación de un nuevo bloque, el cual es una ampliación de pórticos digitales bidireccionales (véase Capítulo V , conclusiones y recomendaciones)

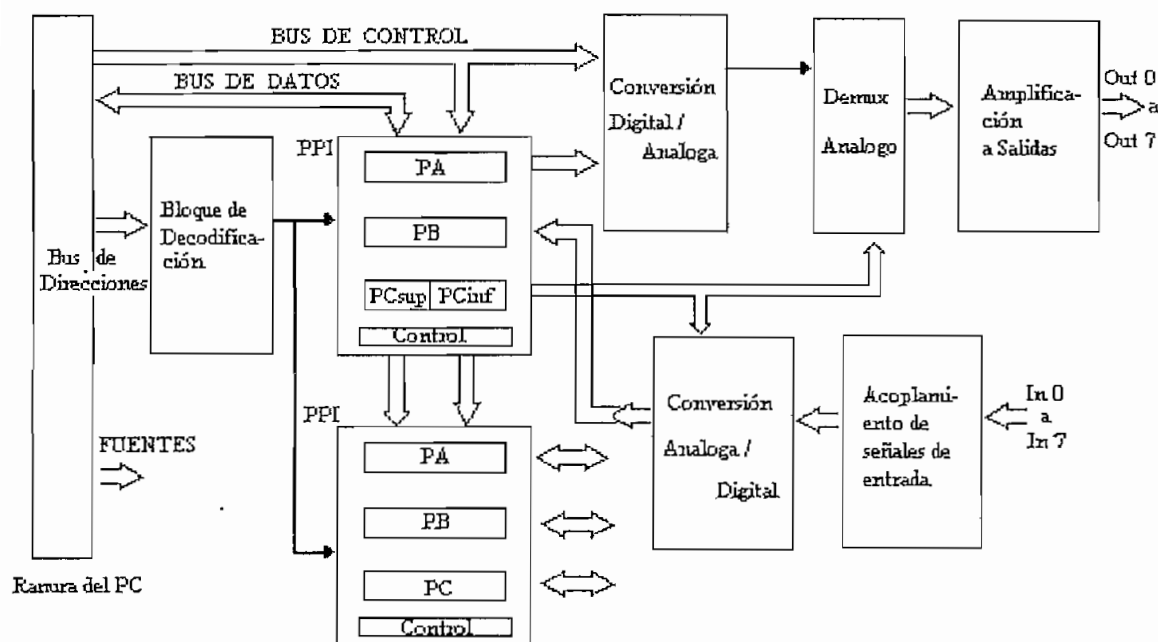


Figura 2.4. Diagrama de Bloques de la tarjeta ADAIO

2.4.1 DESCRIPCION DEL DISEÑO

Como se puede apreciar, la parte modular del proyecto es el PPI y, por la descripción anterior también se observa que la tarjeta tiene muy definidos los bloques de decodificación, de entradas análogas y de salidas análogas.

Cabe mencionar que el PPI utilizado posee 4 registros internos de 8 bits y se lo programa en modo cero, por lo tanto, todos los módulos deberán tener una precisión de 8 bits.

Con estos antecedentes se procede a la descripción detallada de cada uno de los bloques:

2.4.1.1 BLOQUE DE DECODIFICACIÓN

Se utiliza un arreglo de decodificación seleccionable por bloques de dirección, por medio de los puentes (J1, J2) como se puede apreciar en la figura 2.4.1.1.a), de esta manera se puede colocar la tarjeta en cualquier dirección dentro del bloque de direcciones (300H a 31FH).

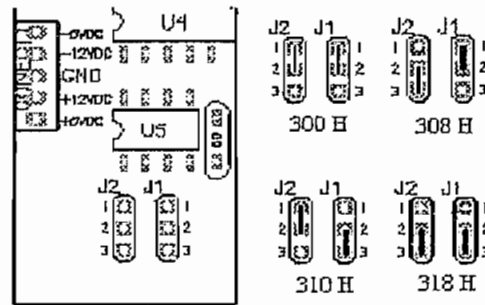


Figura 2.4.1.1.a) Colocación de puentes en J1, J2 para el seteo de dirección base

Este arreglo de decodificación utiliza las señales de dirección desde la A0 a la A9, señales que están localizadas físicamente en las posiciones A31 a A22, respectivamente en la ranura de expansión de todo computador personal compatible utilizado para el efecto. En la tabla 2.4.1.1.a), se observan las posibles direcciones base que puede ocupar la tarjeta de experimentación ADAIO.

A9	A8	A7	A6	A5	A4 J1	A3 J2	A2	A1	A0	Dirección Base del PPI (HEX)
1	1	0	0	0	0	0	0	0	0	300
1	1	0	0	0	0	1	0	0	0	308
1	1	0	0	0	1	0	0	0	0	310
1	1	0	0	0	1	1	0	0	0	318

Tabla 2.4.1.1.a) Direcciones posibles de para la tarjeta ADAIO

En la decodificación se utilizan inversores para las señales desde la A9 a la A0, de esta forma (figura 2.4.1.1.b):

- ⇒ Las señales A8 y A9 van conectadas directamente a dos entradas de una compuerta lógica NAND de 8 entradas;
- ⇒ Las señales A7, A6 y A5 van conectadas a la compuerta anterior por medio de inversores;
- ⇒ En cuanto a las señales A4 y A3 se utilizan para seleccionar el bloque direccionable, selección realizada manualmente por medio de jumpers, también van conectadas a la compuerta NAND, el resultado de una buena combinación (dirección válida) de las señales anteriores da un nivel lógico bajo;
- ⇒ Señal que se combina (se suma), por medio de una compuerta OR, con la señal A2;
- ⇒ El resultado de la combinación anterior con la señal habilitadora AEN determina una señal válida (OL), para la activación del respectivo PPI, el cual funcionará cuando la señal Chip Selec este en nivel lógico Bajo ($CS=0$);
- ⇒ Las señales A0 y A1 van ligados directamente al PPI, pues este utiliza 4 direcciones, las que sirven para direccionar sus registros internos, siendo la dirección base la dirección asignada al Pórtico A (PA), la dirección base + 1 asignada al pórtico B (PB) , la dirección base + 2 asignada al pórtico C (PC) y la dirección base + 3 asignada a la palabra de control, éste registro es sólo de escritura más no de lectura y en éste se programa el modo de funcionamiento del PPI utilizado.

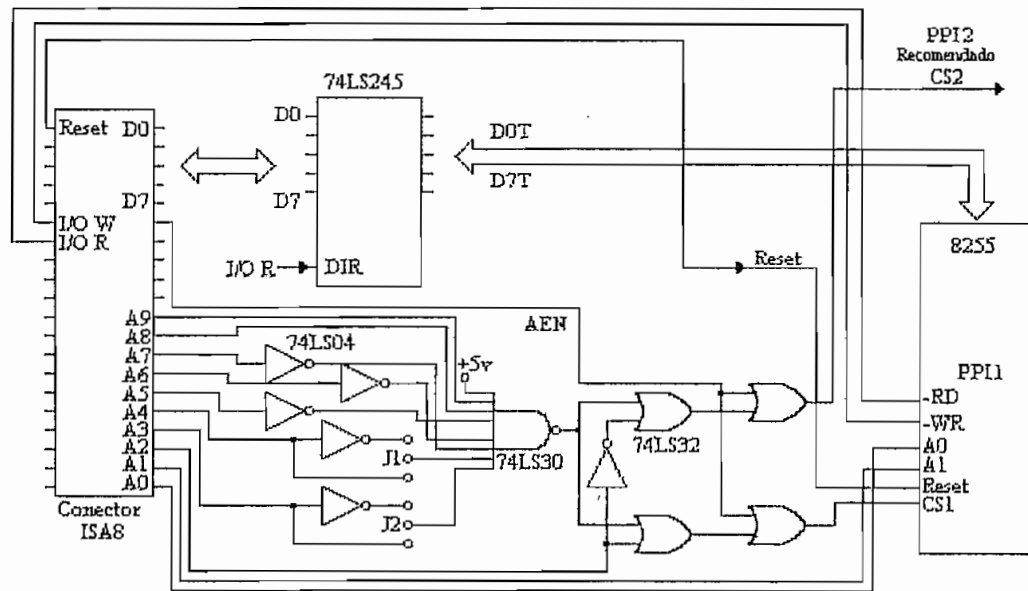


Figura 2.4.1.1.b) Bloque de direccionamiento de la tarjeta ADAIO

2.4.1.2 SEÑALES DEL PPI

En cuanto a las señales del PPI utilizadas en el proyecto, como se puede ver en la figura anterior, se conectan directamente de la señal del RESET de la ranura al RESET del PPI. Las señales /IOWR y /IORD se conectan respectivamente en los pines indicados en el PPI. La señal habilitadora para el inicio de funcionamiento es la del pin /CS, la cual se da cuando existe una dirección válida. El bus de datos D7 a D0 localizado en las posiciones A2 a A9 de la ranura se conecta a través de un buffer bidireccional (74LS245) al bus de datos del PPI, correspondiendo DOT a D7T a los pines 34 a 27 respectivamente, con la finalidad de dar mayor consistencia y seguridad a las señales de los datos, evitando de esta manera la pérdida de información.

En cuanto a las señales de control interno para los demás bloques, se utilizan las señales del pórtico C (PC), las cuales serán activadas por medio de software, de

la siguiente manera: del bloque superior del p \acute{o} rtico C (PC) las se \acute{n} ales PC4, PC5 y PC6 controlar \acute{a} n la activaci \acute{o} n y seleccionar \acute{a} n la se \acute{n} al de salida an \acute{a} loga que se estar \acute{a} enviando al PA; es decir, enviar \acute{a} las se \acute{n} ales al decodificador del bloque de salida digital/an \acute{a} loga. El bloque inferior del p \acute{o} rtico C (PC), es decir las se \acute{n} ales PC0, PC1 y PC2 ser \acute{a} n las l \acute{i} neas de control del decodificador interno que tiene el ADC0808 utilizado como elemento principal del bloque de entrada an \acute{a} logo/digital, como se puede observar en el diagrama esquem \acute{a} tico total de la tarjeta (Anexo 1).

El PPI ser \acute{a} programado en modo cero, esto es, seteado por medio de software, escribiendo el n \acute{u} mero 82H en la palabra de control, de tal manera que el PA act \acute{u} e como p \acute{o} rtico de salida, PB como p \acute{o} rtico de entrada y el PC tanto el bloque inferior como el superior act \acute{u} en como l \acute{i} neas de control de los bloques internos. Se estableci \acute{o} \acute{e} ste modo de programaci \acute{o} n porque ofrece la gran ventaja de que en este modo todas las salidas internamente tienen un cerrojo o latch.

2.4.1.3 BLOQUE DE CONVERSION D/A

En el gr \acute{a} fico de la figura 2.4.1.3 se puede ver que para la construcci \acute{o} n del bloque de conversi \acute{o} n digital an \acute{a} loga se utiliz \acute{o} el conversor DAC0830 en conjunci \acute{o} n con un amplificador operacional LM358 que funciona como convertidor de corriente a voltaje; la se \acute{n} al resultante de esta conversi \acute{o} n se la env \acute{i} a a un demultiplexer an \acute{a} logo 4051 (de 1 a 8), el cual se encarga de enviar la se \acute{n} al an \acute{a} loga a la respectiva salida dependiendo de la decodificaci \acute{o} n planteada en las se $\acute{n$ ales habilitadoras.

Como el voltaje de referencia para el convertidor es de + 5VDC y, se desea una

señal análoga de salida de +10VDC, entonces cada señal análoga que sale del multiplexer va a un amplificador no inversor con ganancia dos el cual duplica la señal de salida análoga, cumpliendo de esta manera con uno de los objetivos planteados.

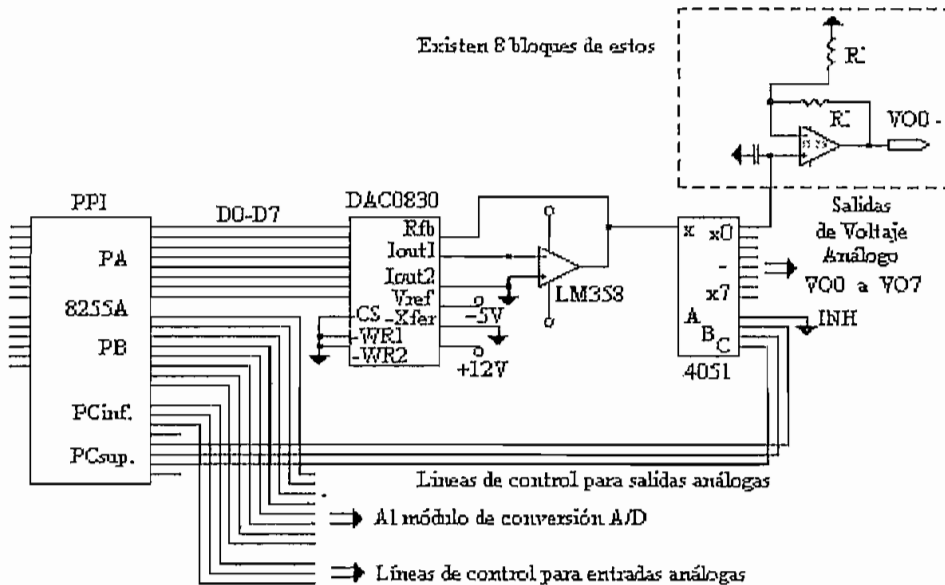


Figura 2.4.1.3 Diagrama del bloque de salida conversión D/A

A continuación se da una descripción detallada de los elementos utilizados en este bloque, y se explican las causas por la que se utilizó cada uno de los elementos de este bloque.

CONVERSION DIGITAL ANALOGICO (DAC0830), DE 8 BITS

De las hojas técnicas del convertidor DAC0830 (Ver anexos de hojas de datos), se determinó que éste circuito integrado es un convertidor digital-análogo de 8 bits con doble buffer, que permite enviar la salida de voltaje correspondiente a una palabra digital, mientras sostiene la próxima palabra digital, y esto permite la simultánea actualización de cualquier número de conversiones.

Las características principales son:

- Doble buffer para un flujo sostenido de datos digitales.
- Interface directo para todos los microprocesadores populares.
- Linealidad específica con cero a plena escala.
- Puede operar “solo” sin microprocesador si así fuese el diseño.

En cuanto a los pines utilizados por el convertidor se tiene:

- CS :Chip Select (Se activa en bajo), El -CS en combinación con el ILE habilitará el -WR1
- ILE :Input Latch Enable. (Se activa en alto)
- WR1 :Write 1. Se activa en bajo siendo usado para cargar los bits de los datos de entrada digital en el latch de entrada. Los datos en el latch de entrada son retenidos cuando -WR1 está en alto. Para actualizar el latch de entrada -CS y -WR1 deben estar en bajo mientras ILE está en alto.
- WR2 :Write 2. (Se activa en bajo) , esta señal en combinación con -XFER, causa que los 8 bits de datos los cuales son habilitados en el latch de entrada se transfieran al registro de conversión.
- XFER :Transfer control signal (Se activa en bajo). Es señal de transferencia de control que habilitará -WR2.
- DIO-DI7 :Entradas digitales donde, DIO es el bit menos significativo(LSB) y DI7 el bit más significativo (MSB).
- Iout1 :La corriente máxima que ocurre para un código digital de todos unos en el registro de conversión y es cero para todos los ceros en el registro de control.

Rfb :Resistencia de realimentación, se encuentra dentro de la pastilla para utilizarla como lazo de realimentación, para un amplificador operacional externo, el cual es usado para proveer un voltaje de salida para el convertidor. La resistencia interna de esta pastilla siempre debería ser utilizada, en vez de una resistencia externa.

Vref :Voltaje de referencia de entrada. Esta entrada conecta un voltaje de precisión externo de referencia. Puede ser seleccionado entre +10 a -10V.

Este convertidor es muy utilizado en la industria de los microprocesadores, ya que el doble buffer permite la flexibilidad para el mayor número de controles, pues los 20 pines de este elemento también son compatibles con el convertidor DAC1230 de 12 bits, lo cual puede permitir ampliar la resolución.

Consideraciones Digitales.- Entre las razones más importantes por las que se escogió esta pastilla integrada para la conversión Digital-Análoga, son:

- a) Cualquier sistema de conversión puede sostener simultáneamente los datos en un registro y la próxima palabra de datos en el registro de entrada, esto permite una rápida actualización a la salida del convertidor.
- b) Lo más importante es que el doble-buffer permite cualquier número de convertidores en un sistema, para ser actualizados en sus nuevas salidas análogas simultáneamente, esto mediante una señal común (strobe).

Además se tomó en consideración para su implementación, que el flujo de conversión sea continuo; es decir que el microconvertidor sea fácilmente configurado para permitir una salida continua análoga que refleje el estado de

una entrada digital. Siendo ésta la aplicación más utilizada donde el convertor es usado en un lazo de retroalimentación continuo, esto se logra simplemente puenteando a tierra los pines correspondientes a las señales -CS, -WR1, -WR2 y -XFER, a demás el pin ILE a alto; todo esto permite a ambos registros internos aplicar a la entrada digital datos, y directamente afectar a la salida del convertor análogo.

Consideraciones análogas.- Como el principal propósito de cualquier convertor digital analógico es proveer una exacta cantidad de salida análoga, la cual representa a la palabra digital, en el caso del DAC0830, la salida IOOUT1 es una corriente directamente proporcional al producto de aplicar un voltaje de referencia por la correspondiente palabra digital. Para una aplicación versátil una segunda salida es proporcional al complemento de la entrada digital. Básicamente:

$$IOOUT1 = [VREF / 15 K\Omega] * [Entrada digital / 256]$$

Donde la entrada digital es decimal (base 10), equivalente a aplicar una palabra de 8 bits (0 a 255). VREF es el voltaje al pin 8 y 15 K Ω es el valor normal de la resistencia interna Rfb.

Por todo lo mencionado anteriormente se ha utilizado este convertor resaltando su beneficio para la aplicación en curso por que permite la simultánea actualización de cualquier número de conversiones, posee un doble buffer para un flujo sostenido de datos digitales, y además puede operar "solo" sin microprocesador ya que así se plantea en este diseño.

Consideraciones del Amplificador Operacional.- El amplificador utilizado deberá tener capacidad de voltaje offset nula y, un bajo valor de corriente de

polarización (bias) de entrada, tan bajo como sea posible. El producto de esta corriente por la resistencia de realimentación produce un error en la salida del voltaje, error que puede ser significativo en medidas bajas de voltaje.

Por lo tanto, se ha seleccionado el operacional LM358, el cual es un integrado que contiene dos operacionales, con compensación de frecuencia interna, puede funcionar con diferentes rangos de alimentación, la entrada de corriente de polarización (bias) es baja ($45nA_{DC}$) y a demás compensada con la temperatura; es compatible con señales lógicas, baja entrada de voltaje offset ($2mV_{DC}$) y baja corriente offset ($5nA_{DC}$)

En cuanto al Multiplexer/Demultiplexer de 8 canales analógicos, posee swiches análogos los cuales tienen baja impedancia en el encendido y muy baja fuga de corriente en el apagado; el control de las señales análogas puede ser activado por señales digitales.

Esta pastilla está polarizada en la tarjeta de la siguiente manera: $V_{DD}=12V$, $V_{SS}=0V$ y $V_{EE}=-5V$, con lo cual se consigue que las salidas análogas se encuentren en el rango de 0 a 5V, las cuales pueden ser seleccionadas por señales digitales. Cuando un "1" lógico está presente en la entrada habilitadora (INH pin 6) todos los canales están apagados.

El CD4051BM es un simple demultiplexer/multiplexer de 8 canales, que tiene 3 entradas de control binario A,B,C y una entrada habilitadora; las tres señales digitales seleccionan uno de los ocho canales de tal manera que la entrada seleccionada se conecte a su salida respectiva.

En la figura 2.4.1.3 se puede observar que para conseguir un voltaje de salida de

+10VDC, a cada canal de salida análoga se ha conectado un amplificador no inversor con ganancia dos, para lo cual se utiliza un circuito integrado LM324, ya que éste contiene 4 operacionales, los que tienen alta ganancia, compensación interna de frecuencia, baja fuga de corriente de fuente la que es independiente del voltaje suministrado, elimina la necesidad de dos fuentes ya que se polariza a los operacionales con +12VDC, esto se realiza para eliminar la salida de voltajes negativos precautelando a dispositivos externos conectados a esta tarjeta.

2.4.1.4 BLOQUE DE CONVERSION A/D

Este bloque de conversión utiliza el pórtico B del PPI el cual funciona como entradas de datos. Las señales del PPI se unen al conversor ADC0808 por medio de un buffer unidireccional 74LS244 con el fin de darle mayor consistencia a los datos de entrada. El conversor ADC0808 de 8 bits, opera en modo de conversión continua. Se escogió este conversor por poseer 8 entradas análogas, las que, por medio de un multiplexer interno se las puede seleccionar una a una por software, por medio de las líneas del pórtico C (bloque inferior). Las señales análogas que recibe el conversor tienen como referencia +5VDC pero, como se desea señales de entrada de +10VDC, entonces, previa a cada entrada se coloca un divisor de tensión. Además, para precautelar todo este bloque se utiliza un diodo zener de +10V por cada señal de entrada, la que no debe ser mayor al voltaje zener, pues en caso contrario la señal será recortada a +10VDC.

A continuación se da una descripción de los elementos que conforman este bloque.

Como el elemento principal del bloque de conversión A/D se tiene al circuito integrado ADC0808, que es un componente de adquisición de datos con la siguientes características:

- Elemento de 8 bits, multiplexado a 8 canales;
- Tiene un tiempo típico de conversión de 100 μ s
- Ofrece un fácil acople con microprocesadores, provisto por el latch y un decodificador multiplexado;
- Además, ofrece alta velocidad, alta precisión, mínima dependencia de la temperatura, y consumo de mínima potencia
- Sus salidas se ajustan a los niveles de voltaje de lógica TTL.
- Es un conversor análogo de 8 bits

Por último, en la figura 2.4.1.4.a) se puede observar la disposición interna de este circuito integrado, en donde se puede apreciar que éste, fue diseñado para dar una exacta conversión en un amplio rango de temperaturas.

La conversión continua puede ser cumplida conectando el pin de fin de conversión (EOC salida) con el pin de inicio de conversión (START entrada). Si se utiliza en este modo un pulso de conversión externa, se debería aplicar después de aplicada la alimentación.

En la figura 2.4.1.4.b) podemos observar el diagrama del bloque de conversión análoga/digital.

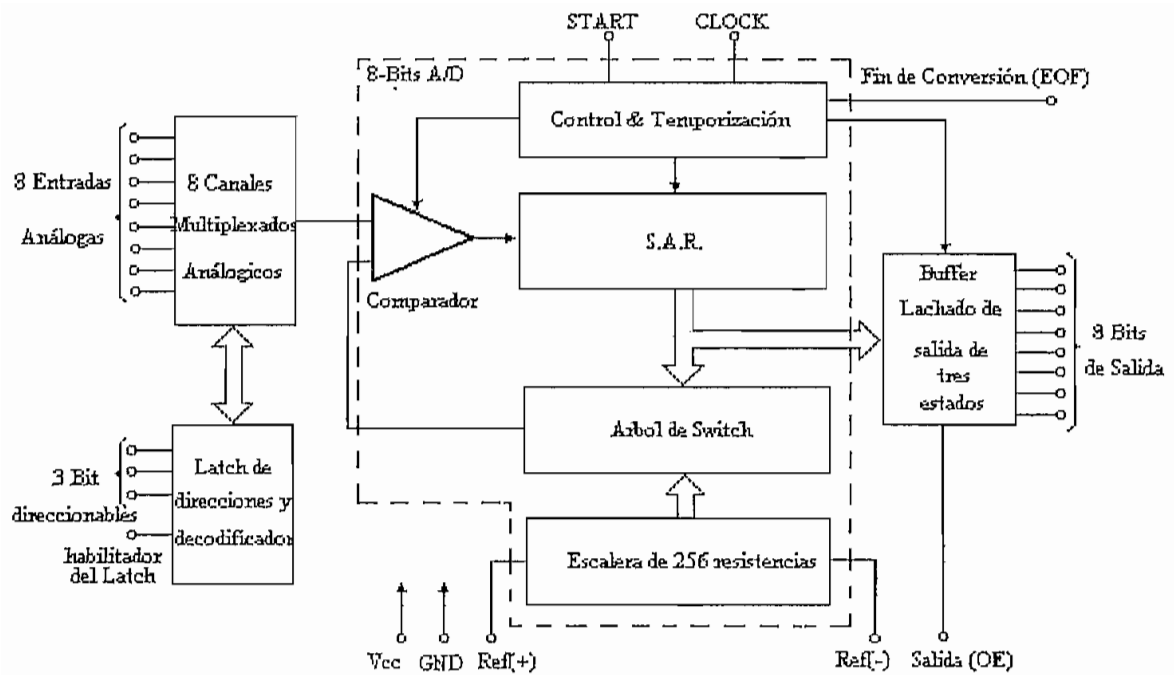


Figura 2.4.1.4.a) Diagrama de bloques del convertor A/D

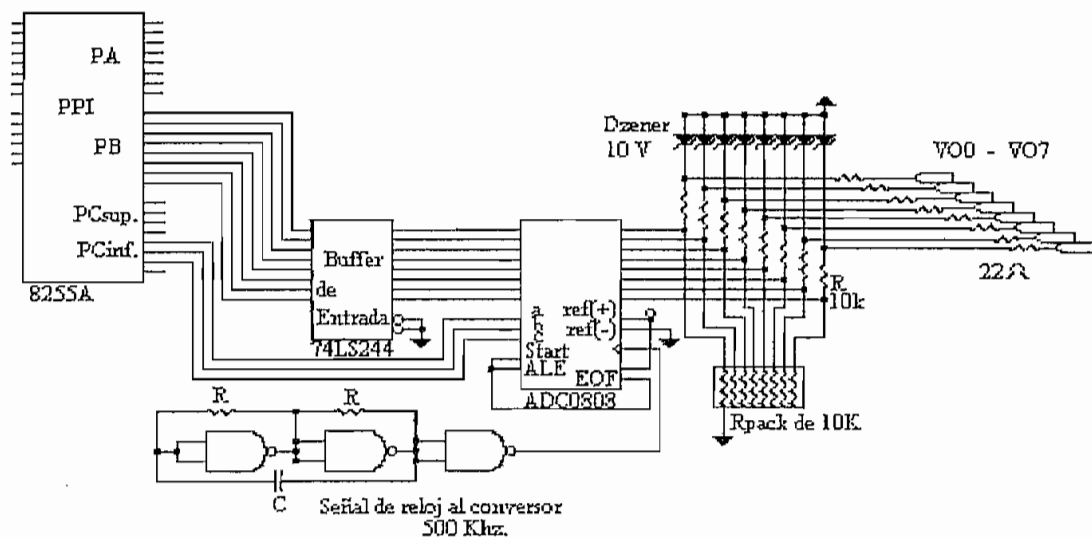


Figura 2.4.1.4.b) Diagrama esquemático del bloque de conversión A/D

Reloj del convertor: En cuanto a la señal de reloj para el convertor ADC0808, se utiliza una compuerta lógica NAND (74LS04), conectada como se indica en la figura anterior (figura 2.4.1.4.b), la frecuencia del reloj será de 500 Mhz, para lo

cual se necesita dos resistencias de $1K\Omega$ y un condensador de $820pF$.

2.5. DISPOSICION DE ELEMENTOS EN LA TARJETA.

Para plasmar todo el diseño anteriormente citado, luego de realizar el diagrama esquemático (Anexo A1) en ORCAD (archivo ADAIO.SCH), se realizó la generación del archivo ADAIO.NET necesario para el enlace entre el diseño esquemático y el diseño de circuitos impresos, archivo denominado ADAIO.PCB, realizado en TANGO.

Las consideraciones para la disposición física de los elementos fueron :

- El buffer bidireccional 74LS245, así como los circuitos integrados que conforman el bloque de decodificación, deben estar dispuestos físicamente lo más cerca a la ranura de expansión en las líneas correspondientes a datos y direcciones, respectivamente.
- Que los Circuitos de Interfaz Periférico Programable estén lo más cerca posible uno respecto a otro, ya que éstos comparten tanto líneas de control como, líneas de datos
- Los bloques de entradas análogas y de protección deben estar colocados lo más lejos a la ranura de expansión, para evitar daños si hubiese sobrecalentamiento por el ingreso de voltajes superiores a $+10VDC$.
- Los circuitos integrados que conforman el bloque de salidas análogas deben estar colocados lo más cerca posible entre ellos, pues estos comparten líneas de polarización y de decodificación.

Todas estas consideraciones se toman con el fin de optimizar el espacio y por ende el tamaño total de la tarjeta. Cabe anotar que la tarjeta fue ruteada a doble

lado (los diagramas de este ruteo se lo puede ver en el anexo A2).

La tarjeta fue implementada con un ancho en la pista (track) de 10 unidades (1 unidad equivale a 1 milésima de pulgada), con agujero metalizado, y en fibra de vidrio la cual es resistente a altas temperaturas, tecnología no disponible todavía en el Ecuador, razón por la cual su implementación física se la realizó en Colombia.

CAPITULO III

DISEÑO DEL SOFTWARE

Como se puede apreciar en el capítulo anterior el diseño del hardware va ligado al diseño de software y, dependiendo de éste su posterior aplicación. En el presente capítulo se presentan programas básicos que se han desarrollado tomando en cuenta las consideraciones necesarias para demostrar el funcionamiento del presente proyecto de tesis.

3.1 ESPECIFICACION Y SELECCION DEL PROGRAMA.

Debido a que la tarjeta ADAIO es de tipo universal, es decir, compatible con cualquier computador tipo XT/AT/286/.../586. Se han desarrollado programas básicos en diferentes lenguajes, tales como : Qbasic, Visual Basic, Lab View.

Se han seleccionado estos lenguajes de programación, ya que cada uno de éstos tiene una lógica de funcionamiento diferente, así por ejemplo:

a.- En un lenguaje de programación convencional, tal como el Qbasic, se realizan los programas de tal manera que se ejecuten secuencialmente, bien sea en forma descendente o estructurada, siguiendo el flujo dado por las sentencias de control previamente escritas en el código fuente.

Se seleccionó este lenguaje de programación por ser uno de los lenguajes universales básicos y, además por ser:

- Un lenguaje de programación que puede ejecutarse en cualquier tipo de

computador PC compatible, desde el más básico como lo es un XT de 8MHz ; por ende sus requerimientos son : memoria de 1Mb en RAM; y espacio libre en disco duro de 1.2Mby;

- Tiene instrucciones tales como IN, OUT, etc., que permiten el acceso directo a localidades de memoria, necesarios para el manejo de elementos periféricos.
- Existen varios proyectos realizados en este lenguaje, lo cual permite un rápido acoplamiento y utilización de librerías existentes para las aplicaciones deseadas.

b.- En Visual Basic (V 3.0), la lógica de flujo es totalmente, distinta, y se compone de diversas porciones de código que son activadas de acuerdo a un determinado suceso. Estos sucesos ocurren en la pantalla según decisión del usuario, es decir, el programa está monitoreando constantemente las acciones sobre la pantalla. De esta forma el programa no sigue la secuencia fija de control, sino la secuencia depende de las decisiones del usuario.

Se seleccionó este lenguaje de programación por ser en la actualidad uno de los lenguajes de entorno visual más difundidos, mediante el cual se puede realizar aplicaciones comerciales y, además por que:

- Este lenguaje de programación puede ejecutarse en un computador con medianas características tales como: Microprocesador 486 de 66MHz,

con 12Mby libres en disco duro y 4 Mb de memoria RAM.

- Tiene una gran librería de ejemplos, los cuales hacen más fácil su aprendizaje, y por ende es apto para toda persona que guste de la programación aplicada a la electrónica.
- Este lenguaje de programación no tiene instrucciones tales como IN o OUT, propias para el manejo de localidades de memoria que permitan el acceso a la programación de elementos periféricos pero, permite incorporar Librerías de Enlace Dinámico DLL (Dynamic-link library) (Ver numeral 3.2.2 LIBRERIA DE ENLACE DINAMICO (DLL))

c.- En Lab View (V 3.1.1), se utiliza un lenguaje de programación gráfica (lenguaje G), cuya lógica de funcionamiento es crear programas en diagramas de bloque. Lab View es un programa de desarrollo de aplicaciones, para lo cual tiene librerías y funciones designada específicamente para la adquisición de datos e instrumentos de control, además éste lenguaje de programación promueve el concepto de programación modular.

Se seleccionó este lenguaje de programación por ser en la actualidad uno de los lenguajes de instrumentación aplicados al control, mediante el cual se puede realizar aplicaciones dedicadas y, además por que:

- Es un programa que puede ejecutarse en cualquier máquina con buenas características tales como: Microprocesador 486 a 100mhz o superior, 8

Mby en memoria RAM y capacidad libre en disco de 40Mb (mínimo).

- Tiene gran librería de ejemplos dedicados a la adquisición de datos.
- Tiene Instrucciones de manejo de pórticos, lo cual hace posible la utilización de este lenguaje con cualquier tarjeta universal para el control de elementos periféricos.

Por todo lo mencionado anteriormente se han desarrollado pequeños programas en cada uno de estos lenguajes con la finalidad de demostrar la funcionalidad y el acoplamiento de la tarjeta ADAIO a cualquier computador, sin importar la lógica de programación.

A continuación se hará referencia a la comunicación ó enlace entre el hardware (circuitería electrónica) de la tarjeta con el software (programas) que prueban el correcto funcionamiento de la tarjeta.

3.2 PROGRAMAS DE COMUNICACION

Después de haber diseñado el hardware de la Tarjeta ADAIO se deben realizar programas básicos que puedan servir como manejadores (drivers) ó como programas de referencia para posteriores aplicaciones. Estas aplicaciones dependerán en su mayor grado de la utilización particular que se de a la tarjeta ADAIO, por ende los programas deberán ser dedicados.

Antes de indicar el diseño y funcionamiento de cada uno de los programas se hará referencia a las instrucciones, librerías o bloques de programas (lenguaje G), utilizados para establecer la comunicación entre la tarjeta y el programa.

3.2.1 INSTRUCCIONES PARA EL MANEJO DE PORTICOS I/O

En todo lenguaje de programación convencional existen instrucciones o funciones que permiten ingresar o sacar datos a través de los puertos. Así en Qbasic se tiene las siguientes instrucciones INP o OUT utilizadas de la siguiente manera:

`X = INP (Port)`

Esta instrucción permite almacenar en la variable "X" el resultado de leer la información existente en el puerto "Port"; por lo tanto, esta instrucción se utiliza para leer el pórtico de entrada. La variable "X" puede ser tratada internamente para poder luego ser graficada, tabulada, etc.

`OUT (Port), Y`

Esta instrucción permite sacar datos por el puerto "Port", los datos de salida están ubicados en la variable Y, en este caso primero se tiene que tratar la variable que se desee enviar al puerto.

3.2.2 LIBRERIA DE ENLACE DINAMICO (DLL)

Una de las desventajas de Visual Basic (V 3.0), es que éste lenguaje de programación de entorno visual no tiene instrucciones que manejan directamente la información de los puertos, por lo tanto, para resolver este problema se compiló una librería de enlace dinámico realizada en Turbo C++ para Windows (V 3.0). Esta librería se la denominó "PPI.DLL", cuyo listado es:

```

/*****
//      Realizado por CARLOS G. PILLAJO A.
//      All CopyRight CGPA, 1997
/*****/
#include <windows.h>
#include <dos.h>

#define EXPORTAPI _far _pascal _export _loadds
/*****/
FUNCION:
    LibMain(HANDLE, WORD, WORD, LPSTR)
PROPOSITO:
    Llamada por LIBENTRY, la cual es llamada
    por Windows cuando la DLL es cargada
*****/

int EXPORTAPI LibMain( HANDLE    hModule,
                      WORD      wDataSeg,
                      WORD      cbHeapSize,
                      LPSTR     lpszCmdLine )
{
    hModule;
    wDataSeg;
    cbHeapSize;
    lpszCmdLine;
    return 1;
}
/*****/
Procedimiento para la salida de la
Librería de Enlace de Datos (DLL)
*****/
VOID EXPORTAPI WEP (int bSystemExit)
{
    bSystemExit;
}

/*****/
FUNCIONES DE ESCRITURA - LECTURA DE PORTICOS
*****/

void EXPORTAPI Writeppi(int x,int y)
{
    outportb(x,y);
}

int EXPORTAPI Readppi(int x)
{

```

```
int valor;  
valor=inportb(x);  
return(valor);  
}
```

Mediante esta librería Visual Basic puede acceder a la información de los pórticos tanto en operaciones de escritura como de lectura. Cabe recordar que la operación "Writeppi" es una subrutina que envía un byte a un puerto de salida, no regresando un valor al programa, en cambio "Readppi" es una función, debido a que regresa un byte desde un puerto de entrada. (Véase Figura 3.2. 1)

Para que Visual Basic establezca comunicación con las instrucciones dadas en la librería dinámica se deberá guardar este archivo "PPI.DLL" en la dirección dada por el programa en el módulo destinado para asignación de variables (MODULE1.BAS), de esta manera:

```
Declare Sub Writeppi Lib "C:\ADAIO\PPI.DLL" (ByVal x As Integer, ByVal y As Integer)  
  
Declare Function Readppi Lib "C:\ADAIO\PPI.DLL" (ByVal x As Integer) As Integer
```

Figura 3.2. 1

3.2.3 BLOQUE DE PROGRAMA DE COMUNICACION (Lenguaje G)

En Lab View existen bloques de programa que sirven para el manejo de puertos de, estos bloques son accesibles por el usuario fácilmente, sólo se deben localizar en la ventana de diagrama del VI en el menú (figura 3.2.3.a).

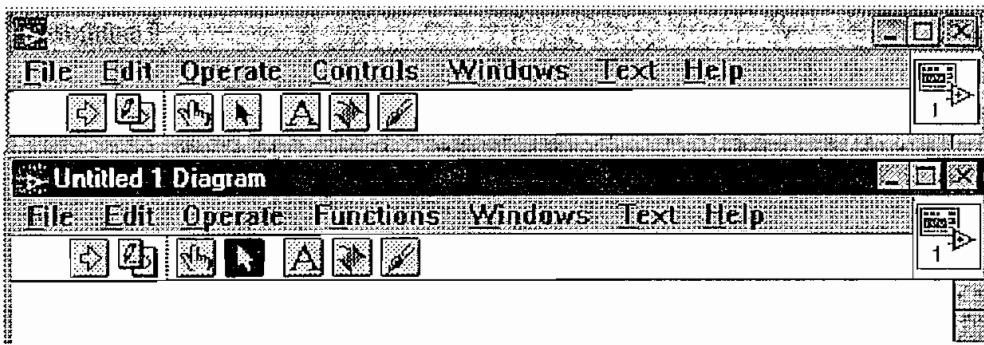


Figura 3.2.3.a Pantallas principales de Lab. View

En la opción "Funtions", existen las funciones con las cuales se puede programar el VI, se abre esta opción y se debe ingresar a la opción "Utility" y dentro de esta a "System" en donde se encuentran los bloques Out.vi o In.vi, como se puede visualizar en la figura 3.2.3.b

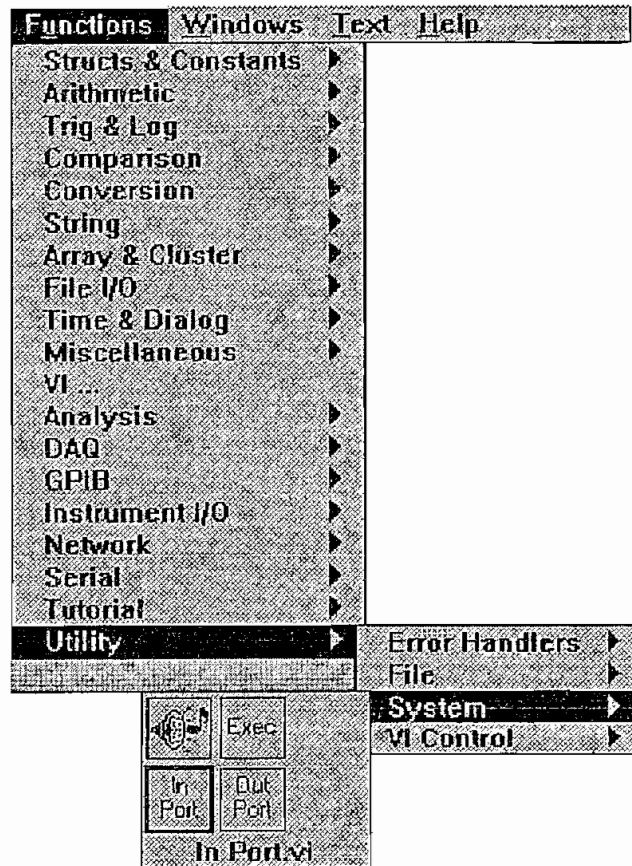


Figura 3.2.3.b Menú de la opción "Funtion" en Lab. View

Luego de haber revisado brevemente las instrucciones, librerías y bloques de programas que serán utilizados para el diseño del software, a continuación se describirá brevemente cada uno de los programas realizados.

3.3 PROGRAMAS EN ENTORNO VISUAL

La tarjeta ADAIO es una tarjeta ampliadora de pórticos, la cual da pórticos análogos y digitales tanto de entrada como de salida , para el funcionamiento de elementos externos, por lo tanto, esta tarjeta es de uso general limitada sólo por la imaginación del programador. Con el fin de demostrar la compatibilidad y la universalidad de esta tarjeta se han desarrollado los siguientes programas, que prueban su funcionamiento, los cuales pueden ser utilizados en proyectos posteriores.

3.3.1 PROGRAMA EN QBASIC.

El programa que puede servir como base para posteriores aplicaciones es el denominado "PPI.BAS", el cual se encuentra en el disquete adjunto a la tarjeta, las características principales de este programa son: (Ver anexo E flujo del programa PPI.BAS)

- Posee menús en pantalla lo cual permite un fácil acceso.
- Navegación y puesta en funcionamiento ya sea de adquisición o emisión de datos por medio teclas habilitadoras.
- Ayuda en línea de cada una de las acciones.

Este programa realiza las siguientes funciones secuenciales dependiendo en el menú que se encuentre.

- a) En el menú principal se puede escoger ya sea: "SISTEMA, PORTICOS, GRAFICOS, FUNCIONES, DIGITALES , y AYUDA.
- b) Dentro del menú SISTEMA se tiene: -Acceso al DOS dentro de éste se puede escoger si desea aplicar un comando de DOS o si se desea salir temporalmente al Sistema Operativo; -Acerca del Programa, al escoger esta opción se muestra una pantalla de información; -Continuar, esta opción es habilitada cuando la ejecución del programa ha sido parada; -Salir, sirve para abandonar la ejecución del programa.
- c) Dentro de menú PORTICOS se tienen los pórticos análogos que pueden ser habilitados uno a la vez ya sea pórticos análogos de entrada o pórticos análogos de salida, y dentro de cada uno de estos se puede escoger el pórtico deseado que va desde el 0 al 7 tanto para entradas como para salidas.
- d) Dentro del menú GRAFICOS se tiene que elegir si se quiere graficar la adquisición de datos o la generación de datos, siendo la emisión de datos secuencial en base a las teclas cursoras de posición.
- e) Dentro del menú FUNCIONES se tienen las opciones de escala, la cual setea el tiempo de adquisición dadas en divisiones por segundo (div/seg), y grillas la cual setea si el gráfico es visto con grillas o no.

- f) En el menú DIGITALES, se puede escoger si se desea realizar una prueba de éstos pórticos como entradas o salidas.
- g) En el menú AYUDA, se tiene una referencia rápida de las diferentes funciones antes descritas.

3.3.2 PROGRAMA EN VISUAL BASIC (V3.0)


Siendo este uno de los paquetes utilizados en la actualidad y de uso comercial para todo tipo de aplicaciones se ha implementado un programa que al igual que el anterior puede servir como base para posteriores aplicaciones en cuanto a la adquisición y emisión de datos se refiere. Esto se lo puede hacer gracias a que ya se ha superado e implementado un enlace entre éste programa y el direccionamiento de datos a registros de memoria. (Véase el acápite 3.2.2).

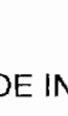
El programa denominado ADAIO.MAK, el cual se encuentra en el disquete adjunto a la tarjeta, (ver anexo E flujo del programa ADAIO.MAK) posee las siguientes características principales:

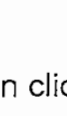
- Tiene íconos que activan las diferentes funciones como son: configuración inicial, la adquisición, la emisión de señales análogas, pruebas de pórticos digitales, información del sistema e icono de salida lo cual permite un fácil acceso.
- Utiliza una forma MDI (Interface de Múltiple Documento), la cual actúa como la forma patrón para toda la aplicación.

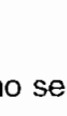
- Posee enlace con archivos ayuda (*.hlp).



Este programa realiza las siguientes funciones dependiendo del suceso activado:

a)  **ICONO DE INICIO**, al realizar un click en éste ícono se da comienzo a la configuración de la dirección base la cual servirá para el buen funcionamiento.

b)  **ICONO DE SALIDAS ANALOGAS**, representado por una pantalla y flecha de salida; al realizar un click en este ícono se abre una forma hija en la cual se presenta el chequeo de cual de las señales de salida se desea activar (una a la vez); a demás se puede escoger el tipo de onda a enviar, y se puede visualizar el voltaje enviado, por medio de un cuadro de dibujo.

c)  **ICONO DE ENTRADAS ANALOGICAS**, representado por una pantalla y un flecha de entrada, al efectuar el suceso de click con el ratón en éste icono se habre una forma hija en la cual se presenta el chequeo de la señal de entrada que se desea visualizar (una a la vez).

d)  **ICONO DE PORTICOS DIGITALES**, al realizar un click en este icono se presenta una pantalla en el cual se puede setear el funcionamiento de cada uno de los puentes, visualizar la palabra de control y el comportamiento de los puentes.

- e)  ICONO DE INFORMACION, el cual trae una pantalla en donde se puede ver la información básica del sistema en el cual esta funcionando el programa de prueba de la tarjeta.
- f)  ICONO DE SALIDA, al pulsar éste ícono, se da por terminado el programa de prueba y sale del mismo seteando los pórtricos a valores dados por defecto tales como : el PPI 1 programado en modo 0, colocando el número 82H (hex) en la palabra de control, lo cual implica que los pórtricos A y C sean de salida y el pórtrico B de entrada; el PPI 2 programado en modo 0, colocando el número 80H (hex) en la palabra de control, consiguiéndose de esta manera que todos los pórtricos estén configurados como salidas.

El formato visual en el cual sea ha desarrollado el programa hace precisamente de él un programa muy amigable para su utilización y entendimiento. Los puntos de relevancia que se deben tomar en cuenta en el desarrollo de éste programa de detallan brevemente a continuación.

◆ PANTALLA DE PRESENTACION

Esta pantalla se despliega al inicio del programa de prueba, en la cual se encuentra el título del proyecto, los nombres de los autores (figura 3.3.2.a), luego de haberse cargado ésta ventana, se presenta otra la cual muestra una forma MDI que será la pantalla principal en donde están contenidos los iconos antes descritos.

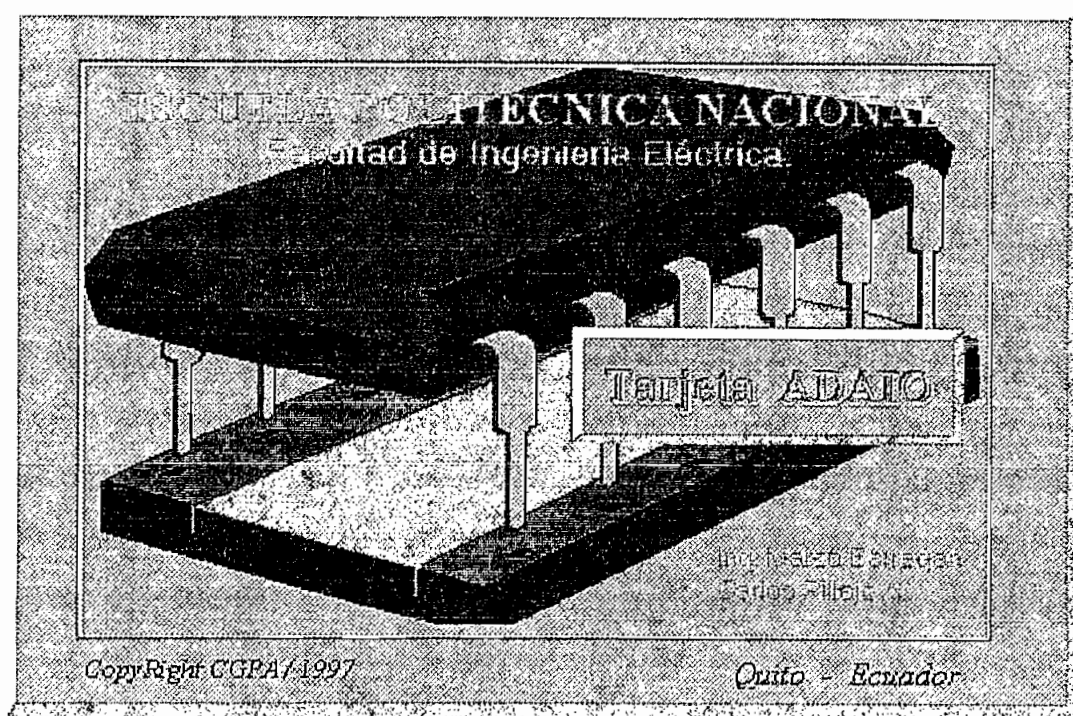


Figura 3.3.2.a Pantalla de presentación en Visual Basic V3.0

◆ PANTALLA DE CONFIGURACION

En ésta se puede ver las posibles direcciones que puede tomar la tarjeta, de las cuales sólo una de ellas puede ser activada, direcciones que van íntimamente ligadas con la posición física de los jumper (J1, J2). Al aceptar la dirección escogida, se presentará un cuadro de diálogo en el cual se pedirá que confirme la dirección base, debido a que de ésta dirección dependerá el buen funcionamiento del programa de prueba.

Al descargarse esta pantalla activará los pórtricos de ésta tarjeta a valores predefinidos tales como: el PPI 1 programado en modo 0 y colocando el número 82H (hex) en la palabra de control, lo cual implica que los pórtricos A y C sean de salida y el pórtrico B de entrada; el PPI 2 programado en modo 0 y colocando el número 80H (hex) en la palabra de control,

consiguiéndose de esta manera que todos los pódicos estén configurados como salidas.

Además se cargará el archivo de ayuda ligado con este programa (AYUDATEC.HLP).

◆ PANTALLA DE SALIDAS ANALOGAS

Al realizar un click en el icono de salidas análogas se presenta una pantalla (figura 3.3.2.b), la cual contiene elementos de programación dentro de los cuales existe las subrutinas necesarias para la generación de datos; el chequeo de los pódicos análogos de salida, la visualización de los datos enviados; un botón para iniciar el envío de datos al pódico seleccionado, y un botón para parar éste suceso.

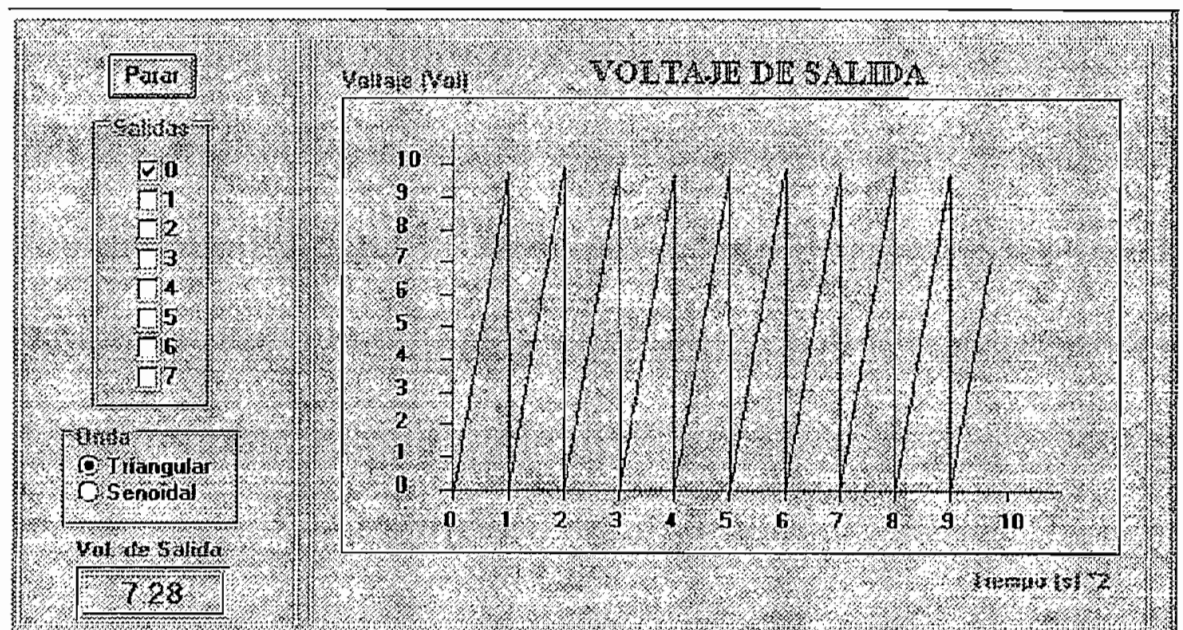


Figura 3.3.2.b) Pantalla de salidas análogas

- ◆ PANTALLA DE ENTRADAS ANALOGAS

Luego de efectuar un click en el ícono de entrada análogas se presenta una pantalla (figura 3.3.2.c), dentro de la cual existen elementos de programa los que contienen las subrutinas necesarias para la adquisición de datos; el chequeo de los pórticos análogos de entrada; la visualización de los datos recibidos; un botón para iniciar la adquisición de datos del pórtico seleccionado, y un botón para parar éste suceso.

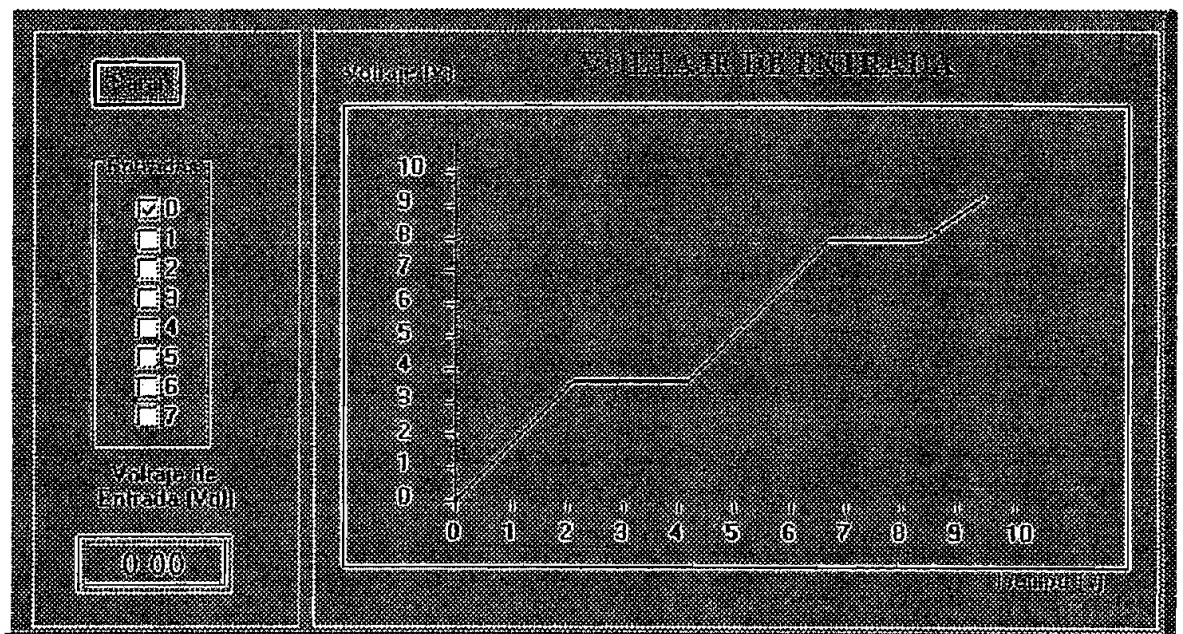


Figura 3.3.2.c) Pantalla de entradas análogas

- ◆ PANTALLA DE PRUEBA DE PORTICOS DIGITALES

Al hacer un click en el ícono del pórticos digitales, se presenta una pantalla (figura 3.3.2.d), en donde se puede setear el comportamiento de cada uno de los pórticos del segundo PPI, el cual será programado en su modo básico (MODO 0 entrada o salidas básicas) y, además se visualizará el comportamiento de cada uno de estos.

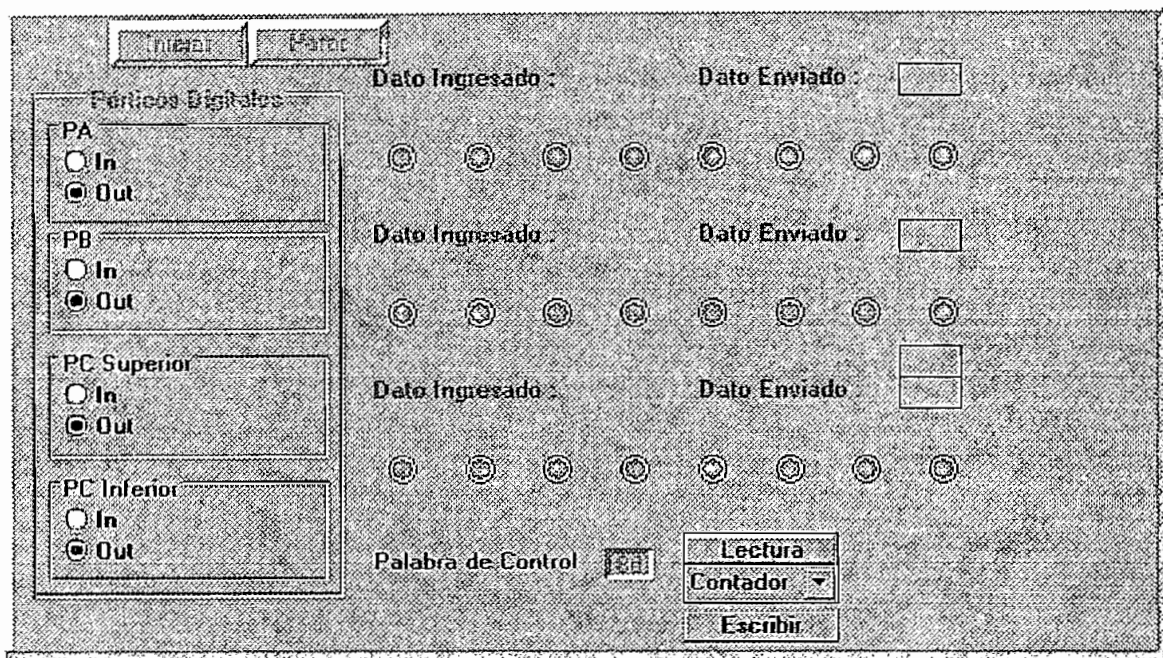


Figura 3.3.2.d) Pantalla de prueba de Pórticos Digitales

◆ REALIZACION DEL ARCHIVO DE AYUDA

Para la realización de este archivo se debe tener acceso a algunos utilitarios necesarios para la elaboración de cualquier archivo de ayuda (*.HLP). Los utilitarios básicos son:

- ◇ Un procesador de palabras bajo windows, que pueda guardar sus archivos en formato RTF (Rich Text Format), preferible Word 6.0. En éste utilitario se digitará el texto que aparecerá en las ayudas ;
- ◇ Un programa graficador, el cual pueda crear archivos en formato genérico de dibujo (archivos *.BMP), éste utilitario se puede utilizar únicamente cuando se tienen gráficos que se incluirán en las ayudas. Un utilitario puede ser el Paint Brush.
- ◇ Un utilitario que realice la edición de hipergráficos, es decir un

EDITOR DE HOSTPOST, el cual realizar archivos *.SHG, necesario cuando se necesita enlazar un dibujo con un texto de ayuda; este utilitario se denomina SHED.EXE y se lo puede encontrar en el subdirectorio ...\\HC de visual basic V3.0

- ◊ El último utilitario y el más importante es el que enlaza la información generada en los anteriores, construyendo archivos *.HPJ, los cuales son archivos generadores previos al archivo de ayuda propiamente dicho los archivos *.HLP; este utilitario contiene un compilador de ayuda tal como el HC31.EXE, localizado igualmente en Visual Basic.

Luego de saber manejar los utilitarios anteriores, los pasos previos para la realización de un archivo de ayuda es:

- a) Realizar el texto que servirá de lectura en el archivo ayuda, realizarlo en Word y guardarlo en formato RTF, teniendo en cuenta lo siguiente, que si se desea realizar un salto de página de información asociado a un texto, este texto debe tener una palabra clave la cual debe estar oculta y con formato de doble subrayado todo el texto ligado con esta palabra; si se desea realizar una presentación de texto en pantalla (popup text), la palabra clave debe estar oculta y con simple subrayado todo el texto asociado a esa palabra;
- b) Si la ayuda tiene gráficos (archivos *.BMP), éstos son asociados en último utilitario; y,
- c) Luego que ya se tiene todo el texto y los gráficos requeridos, para la

ayuda se procede a compilarlos en el último utilitario antes mencionado.

3.3.3 BLOQUES DE PROGRAMA EN LAB VIEW

Los bloques de programa realizados en este lenguaje se hallan en el disquete asociado con este proyecto, los mismos que se encuentran dentro de un subdirectorío llamado VIADAIO, el cual contiene una librería denominada ADAIO.LLB, dentro de la cual hay pequeños bloques de programa utilizados sobre todo en las pruebas a realizarse en el siguiente capítulo, en donde se los describirá.

CAPITULO IV

PRUEBAS Y RESULTADOS

Antes de describir las pruebas realizadas para la comprobación del correcto funcionamiento de la tarjeta, se debe tomar en cuenta que la mayor parte de éstas fueron realizadas en un computador con las siguientes características: Microprocesador AMI 586, con 16 Mby de memoria RAM, Disco de 1.2 Gby; también se realizaron pruebas de compatibilidad de direcciones en otro computador con microprocesador INTEL386, con 2 Mby de RAM y disco duro de 270 Mby.

Las pruebas en su mayor parte se realizaron en el laboratorio de Sistemas de Control, utilizando los siguientes instrumentos:

- Un osciloscopio TEKTRONIX (modelo 5113) que permite grabar la señal, lo cual es útil para bajas frecuencias;
- Un generador de ondas SIMPSON (modelo 420), el cual permite ingresar ondas de diferentes tipos a la tarjeta;
- Un voltímetro digital TMC-500, para poder visualizar la lectura tanto de emisión como de adquisición de datos

Una precaución que se debe tener para realizar las pruebas es que debe existir un punto de referencia común entre todos los elementos, esto se consiguió uniendo las tierras de cada uno de los instrumentos, tanto del

osciloscopio, el generador y el computador.

A continuación se describirán las pruebas realizadas.

4.1 PROGRAMAS DE COMPROBACION DE DIRECCIONES.

Con el fin de comprobar el correcto funcionamiento de la tarjeta ADAIO, según lo indicado en el capítulo anterior, se elaboraron programas que servirán para probar la compatibilidad de hardware y software dado por una adecuada decodificación del bloque de direcciones.

La ejecución de los programas ya descritos, fue perfecta, por lo tanto, el bloque de decodificación de direcciones está bien elaborado.

4.2 PRUEBAS DE ACOPLAMIENTO Y FUNCIONAMIENTO.

Luego de haber determinado y comprobado el acoplamiento entre software y hardware dado por el bloque de decodificación de direcciones, las siguientes pruebas de funcionamiento dadas por la ejecución de pequeños programas adicionales estuvieron encaminados a determinar: la frecuencia de generación de datos, y la frecuencia de adquisición de datos, con el fin de establecer las características técnicas en cuanto a la velocidad de procesamiento de información que posee la tarjeta ADAIO.

4.2.1 PROGRAMAS DE GENERACION DE DATOS.

Con el fin de determinar la frecuencia, a la cual puede operarse la tarjeta ADAIO, y comparar la velocidad de respuesta de programas compilados con diferentes compiladores, se han realizado programas que sacan una señal rampa a un

puerto de salidas analógicas.

Los programas se han realizado en: Turbo C++, Qbasic, Visual Basic V3.0, Lab View V3.1.1

4.2.1.1 LISTADO DEL PROGRAMA EN TURBO C++

```
/* PROGRAMA PARA SACAR UNA RAMPA A UN PUERTO ANALOGICO */  
  
#include <stdio.h>  
  
#include <dos.h>  
  
#define PORT 768 ; Define el p rtico de salida an logo  
  
#define COUNT 10000  
  
#define UPP 255 ; Define el valor m ximo enviado al p rtico  
  
void main(void)  
{  
  
unsigned register int nsal,dato;  
  
nsal=0;  
  
outp (PORT+3,130); // Define al PPI 1 el modo de  
funcionamiento de sus p rticos  
  
do  
  
{  
  
outp (PORT+2,0); // Define el primer p rtico an logo  
  
outp (PORT,dato); // Env a los datos al p rtico definido  
anteriormente  
  
dato=dato +1; // Incremento del dato enviado
```

```

if( dato>255)

dato=0;

nsal++

}

while (nsal<=COUNT);

}

```

4.2.1.2 LISTADO DEL PROGRAMA EN QBASIC

'PROGRAMA PARA SACAR UNA RAMPAA UN PUERTO ANALOGICO

CONTA = 10000

UPP = 255

SCREEN 0

CLS

LOCATE 2, 10

PRINT " PRUEBA DE PUERTO ANALOGICO"

PRINT ".....GENERANDO DATOS AL OSCILOSCOPIO....."

OUT &H303, &H82 ' Inicialización del funcionamiento del PPI
1

DO ' Inicio de un lazo

FOR DATO = 0 TO UPP STEP 1 ' Emisión de datos al pÓrtico del 1 al 255

OUT &H302, &H0 ' Selección el primer puerto anÁlogo.

OUT &H300, DATO ' Saca el dato al puerto escogido

NEXT DATO

NSAL = NSAL + 1

LOOP WHILE NSAL <= COUNT

4.2.1.3 LISTADO DEL PROGRAMA EN VISUAL BASIC V3.0

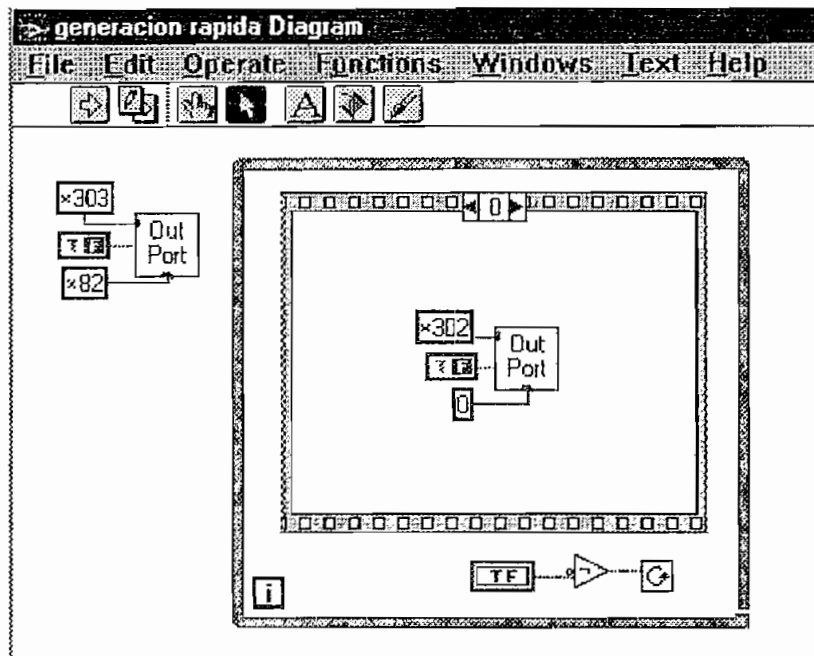
El siguiente programa de prueba de generación de datos a través de un pÓrtico del PPI se lo puede implementar en dentro de la ventana de programación de un COMMAND BUTTON de la siguiente manera, seteando a la variable "dirbase a la dirección base del PPI 1"

```
Sub Command3_Click ()  
  
Do  
  
Writeppi dirbase + 2, 0           ' Selección del pÓrtico de salida anÁloga  
  
Writeppi dirbase, dato          ' Envio del dato  
  
dato = dato + 1                 ' Incremento el valor a enviar  
  
If dato > 255 Then dato = 0  
  
x = DoEvents()  
  
Loop  
  
End Sub
```

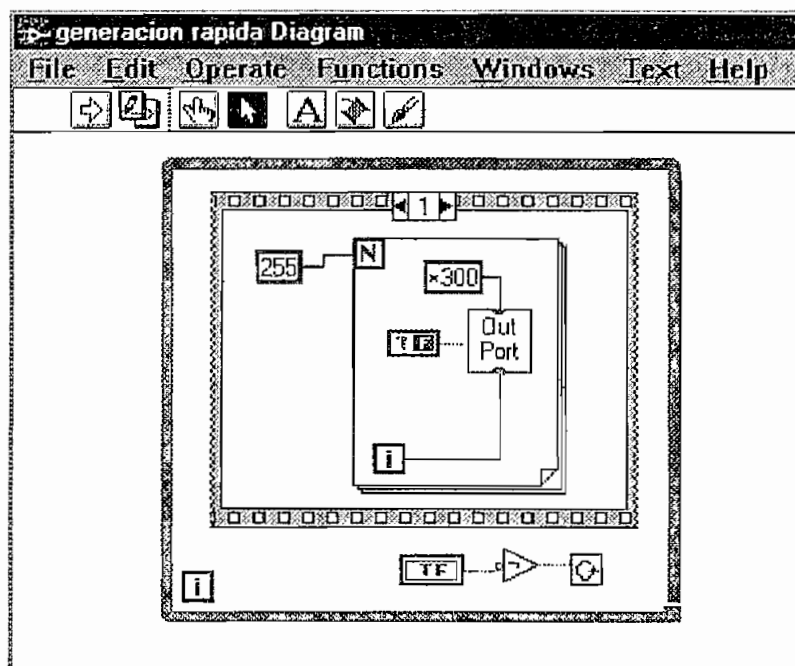
4.2.1.4 LISTADO DEL PROGRAMA EN LAB VIEW

Este programa posee lenguaje gráfico, por lo cual se presentan las pantallas que

componen el programa de prueba de emisión de datos, estas pantallas pueden ser visualizadas en la ventana de diagramas del programa, para lo cual se utiliza una secuencia de dos frames.



Figuras 4.2.1.1 a y b) Frame 0 y 1 en la ventana de diagramas del programa de generación.



4.2.2 PROGRAMAS DE ADQUISICION DE DATOS.

Con el fin de realizar las pruebas de adquisición de datos para los puertos de entrada analógicos, se ingresaron señales de distintas formas y frecuencias para cada uno de los puertos de entrada.

Los datos resultantes de la adquisición pueden ser redireccionados a un archivo, el cual puede ser utilizado para construir un gráfico de la señal obtenida, ó pueden ser simplemente chequeados para ser comparados con un determinado nivel de referencia el cual permitiría realizar una acción de control.

Todas las acciones anteriores y demás tales como cálculo, graficación y manipulación matemática de los datos de entrada, están limitadas sólo por la habilidad del programador.

Los programas descritos a continuación permiten ingresar una señal a un puerto de entradas analógicas y realizar un muestreo de dicha señal.

Los programas se han realizado en: Qbasic, Visual Basic V3.0, Lab View V3.1.1

4.2.2.1 PROGRAMA DE ADQUISICION EN QBASIC

Para la adquisición de datos en el programa PPI.BAS se definió una función de adquisición de esta manera:

```
DECLARE FUNCTION DatAdq! (Nivel!)
```

```
.....
```

```
FUNCTION DatAdq (Nivel)
```

```
DatAdq = INP(Dirb + 1)
```

‘ Coloca el dato que está dentro del
‘ registro Dirb + 1 en la variable DatAdq.

```
END FUNCTION
```

Función que será encargada de recoger el dato (valores de 0 a 255), que da el conversor A/D, para luego manipular estos datos y enviarlos a un gráfico.

4.2.2.2 PROGRAMA ADQUISICION EN VISUAL BASIC V3.0

Dentro del programa ADAIO.MAK, se tiene la siguiente subrutina que se encargará de la adquisición, ésta se encuentra dentro de la pantalla de entradas analógicas dada por la forma FmrSaln, dentro del elemento del programa Timer1.

```
Sub Timer1_Timer ()
```

```
***** Subrutina de adquisición de datos *****
```

```
Static x, y, z
```

```
Label5.Caption = Hex$(FlagIn(Signal))
```

```
Writeppi (Dirbase + 2), FlagIn(Signal1) 'Habilitación del pórtilo por
```

```
Signal = Signal + 1 'el cual va ingresar el dato
```

```
If Signal = Contaln Then Signal = 0
```

```
' Adquisición del dato localizado en el registro Dirbase + 1
```

```
y = Readppi(Dirbase + 1)
```

```
' Manipulación de dato adquirido, para la graficación
```

```
pnivelo.Caption = Format((y / 255 * 10), "0.00")
```

```
z = y * 100 / 255
```

```
x = x + .075
```

```
Picture1.Line -(x, z), RGB(0, 0, 255)
```

```
If x >= 100 Then
```

```
x = 0
```

```
Picture1.Cls
```

```
EscalaIn -20, 120, 120, -20
```

```
End If
```

```
End Sub
```

4.2.2.3 PROGRAMA DE ADQUISICION EN LAB VIEW

Los bloques de programación del lenguaje gráfico que componen el programa de prueba de adquisición de datos, vienen dados en la ventana de diagramas del mismo (figura 4.2.2.3.a), en donde se puede apreciar:

- Un estructura de Lazo While que contiene un bloque de programa (In Port.vi), bloques de operaciones aritmética y variables que representan indicadores o controles.
- El bloque de programa In Port.vi se utiliza para leer el dato que se encuentra

en la dirección del registro especificado, dada por el usuario (en este caso dirección 301H, del pórtico B del PPI, pórtico utilizado como entradas.), seteado este bloque con una variable digital en posición de False (F), para que se pueda leer sólo un byte.

- Las funciones aritméticas de multiplicación y división se utilizan para el acondicionamiento de los datos de entrada, de tal manera que se pueda leer el dato correspondiente al voltaje que esta ingresando, ya que si: en el registro se ingresa un voltaje de 0 VDC equivalga al dato 0 y un voltaje de 10 VDC equivaldrá al dato 255.
- En éste diagrama también se puede ver variables de doble precisión (DBL's) que representan indicadores, los cuales se pueden ver en el panel frontal del programa (figura 4.2.2.3.b); éstos no sirven para visualizar los datos correspondientes al voltaje de entrada.

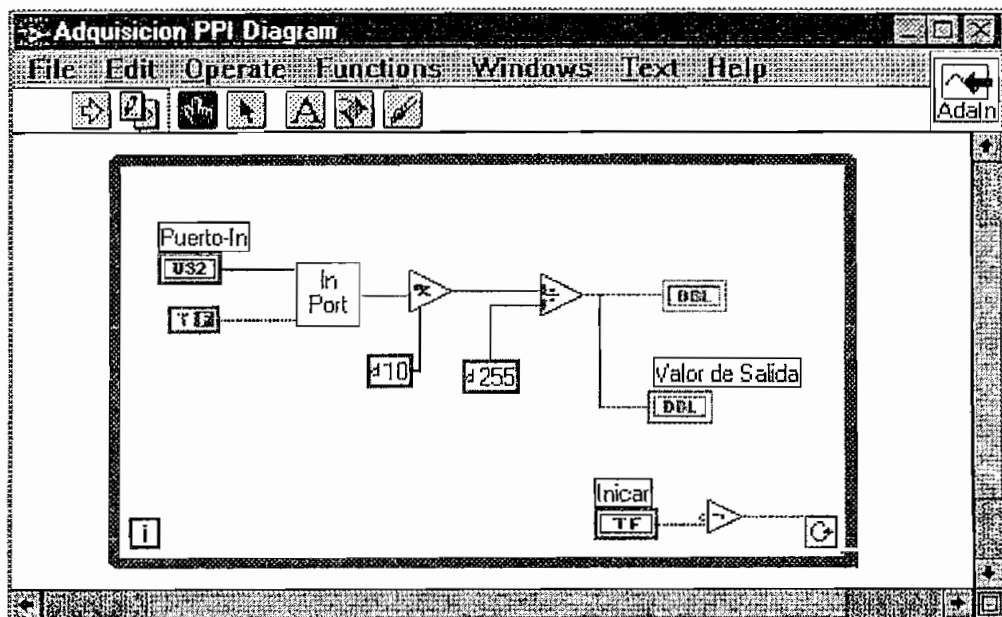


Figura 4.2.2.3.a) Ventana de diagrama del programa de Adquisición de Datos.

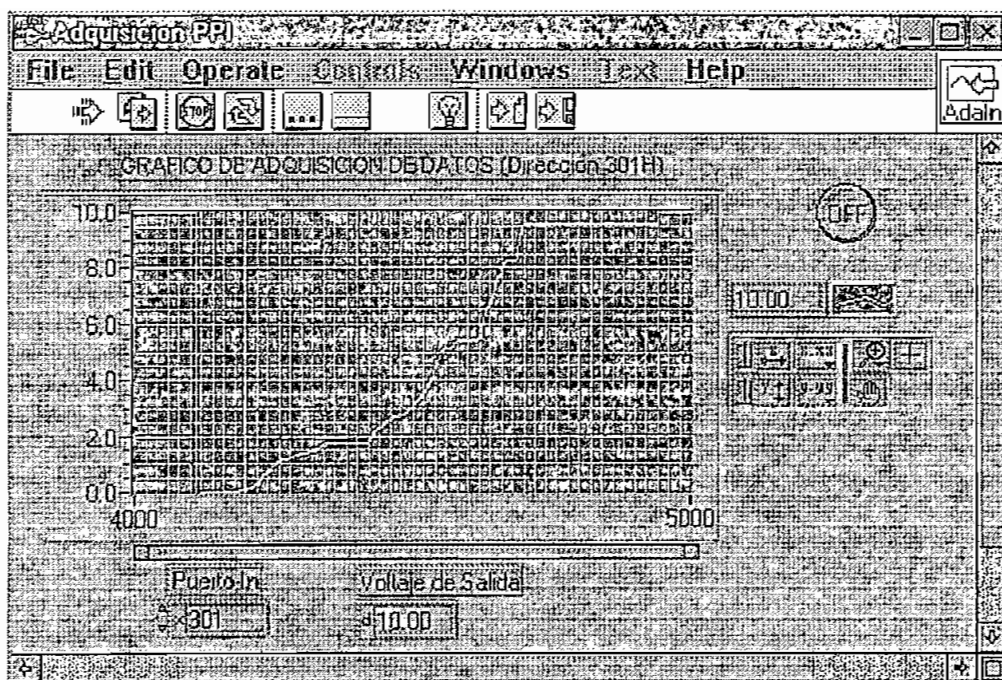


Figura 4.2.2.3.b) Panel de Control del programa de Adquisición de Datos.

4.3 RESULTADOS OBTENIDOS.

La tarjeta ADAIO con las pruebas realizadas anteriormente demostró no solo operar adecuadamente sin alterar el funcionamiento interno del computador, sino que además es óptima en la utilización de registros de memoria. Cabe recordar que la mayor parte de éstas fueron realizadas en computadores con las siguientes características:

- Microprocesador AMI 586, con 16 Mby de memoria RAM, Disco de 1.2 Gby, y
- Microprocesador INTEL386, con 2 Mby de RAM y disco duro de 270 Mby.

Una prueba clara de este hecho es que se colocó dos tarjetas en un mismo computador sin existir interferencia o cruce de información, ya que la una tarjeta

se direccionó en la dirección 300H y la otra en la dirección 310H.

4.3.1 RESULTADO DE LAS PRUEBAS DE CORRECTA OPERACION

En el computador con microprocesador AMI586, no mostró ninguna alteración en su funcionamiento, pudiéndose observar que todos los puertos analógicos tanto de entrada como de salida operaban adecuadamente, sin perturbar el funcionamiento de otros programas, ni mucho menos de otras tarjetas.

4.3.2 RESULTADO DE LAS PRUEBAS DE FRECUENCIA DE OPERACION

Los programas listados en el acápite 4.2.1 prueban la frecuencia de emisión de la tarjeta ADAIO, tomando en cuenta que estos datos fueron emitidos por los programas: Turbo C++, Qbasic, Visual Basic V 3.0 y Lab View, obteniéndose su respuesta en un osciloscopio.

Se debe indicar que, en estos programas no se han incluido retardos para la salida de datos, por lo tanto las frecuencias indicadas son las máximas que se pueden alcanzar para el computador en el que se realizaron las pruebas.

Los resultados obtenidos para los diferentes programas en cuanto a la frecuencia de emisión se tiene en la tabla 4.2.3.a.

RESULTADOS DE GENERACION DE DATOS CON LA TARJETA ADAIO		
	Frecuencia Máxima (Hz)	Frecuencia Máxima (Hz)
	(1 Salida Activa)	(8 Salidas Activas)
Turbo C++	1562.5	227.3
Q-Basic	71.4	27.8
Visual Basic	90.9	43.5
Lab. View	526.3	

Tabla 4.2.3.a Cuadro de frecuencias de generación máxima

El tiempo de retardo que se le debe dar en software para la activación y envío de cada una de las señales esta dada en la tabla 4.2.3.b

MINIMO TIEMPO DE ESPERA PARA ENVIO ENTRE SEÑAL Y SEÑAL (Medido en Visual Basic V3.0)								
SEÑAL	1	2	3	4	5	6	7	8
En mseg.	0	80	100	120	140	160	180	200

Tabla 4.2.3.b Tiempo máximo de retardo para la activación entre señales de salida

4.3.3 RESULTADOS DE LA FRECUENCIA DE ADQUISICION

Para determinar la frecuencia de adquisición se utilizaron los programas descritos en el capítulo anterior, realizando pruebas de adquisición de todos los pódicos uno a la vez.

Se introdujeron señales dadas por un generador de ondas (marca SIMPSON del laboratorio), el cual permitió ingresar diferentes tipos de onda a la tarjeta ADAIO.

Se llegó a determinar que la máxima frecuencia de adquisición no depende del lenguaje de programación; además se pudo fijar que la tarjeta puede funcionar para control de procesos en donde se controle fenómenos físicos (temperatura, presión, nivel de fluidos, etc), que tengan un período de repetición máximo de 0.1 segundos.

El tiempo de retardo que se le debe dar en software para la activación y recepción entre cada una de las señales de entrada es de 100 mseg.

Luego de determinar el correcto funcionamiento se debe realizar un estudio técnico económico que determine si este proyecto da la factibilidad de implementarlo a nivel comercial.

4.4 ANALISIS TECNICO-ECONOMICO

A continuación se dará un enfoque económico general, y dado que la tarjeta ADAIO es modular, ésta puede implementarse según el módulo que se desee, para lo cual se tienen los siguientes cuadros de costos:

Módulo básico de direccionamiento y pórtricos digitales constituido por:

Item	Cantidad	Referencia	Código	Precio Unita (dólares US).	Precio Total
1	13	C0-C13	0.1 uF	0.15	1.95
5	2	J1,J2	Puentes	0.25	0.50
17	1	U8	74LS04	0.63	0.63
18	1	U9	74LS30	0.63	0.63
19	1	U10	74LS32	0.63	0.63
10	1	U0	74LS245	1.00	1.00
24	4	BU8-U10,BU16	BASE-14	0.13	0.50
26	1	BU0	BASE-20	0.18	0.18
7	1	JP1	Conect. 34 macho	0.38	0.38
11	1	U1	8255A	3.00	3.00
28	1	BU1	BASE-40	7.50	7.50
SUBTOTAL:					16.88

Módulo de Entradas Analógicas constituido por:

Item	Cantidad	Referencia	Código	Precio Unita. (Dólares US)	Precio Total
2	1	CK1	820 pF	0.15	0.15
3	8	DZ0-DZ7	5.1 VDC	0.50	4.00
6	1	JP0	Conect. 20 macho	0.25	0.25
8	16	R16-R24	10 k	0.05	0.80
9	2	R1K,R2K	1 k	0.05	0.10
11	1	U2	8255A	7.50	7.50
15	1	U6	74LS244	0.83	0.83
16	1	U7	ADC0808	7.50	7.50
22	1	U14	74LS00	0.63	0.63
23	1	U15	10 Rpack.	0.50	0.50
24	1	BU14	BASE-14	0.13	0.13
28	1	BU2	BASE-40	0.38	0.38
29	1	BU7	BASE-28	0.20	0.20
26	1	BU0	BASE-20	0.18	0.18
SUBTOTAL:					23.13

Módulo de Salidas Analógicas constituido por:

Item	Cantidad	Referencia	Código	Precio Unita. (Dólares US)	Precio Total
1	8	CA1-CA8	0.1 uF	0.15	1.20
8	8	RA1-RA2	10 k	0.05	0.40
12	1	U3	DAC0830	8.25	8.25
13	1	U4	CD4051	1.00	1.00
14	1	U5	LM358	0.75	0.75
20	2	U11,U12	LM324	0.88	1.75
21	1	U13	10 Rpack	0.50	0.50
24	2	BU11-BU12	BASE-14	0.13	0.25
25	1	BU4	BASE-16	0.15	0.15
26	2	BU3	BASE-20	0.18	0.35
27	1	BU5	BASE-8	0.13	0.13
				SUBTOTAL:	14.73

Por lo dispuesto anteriormente el costo total de la tarjeta, implementando todos los módulos se presenta en el siguiente cuadro:

Costo por módulo	Dólares
Círcuito Impreso	30.00
M. de direccionamiento y pórtricos digitales	16.88
M. de entradas análogas	23.13
M. de salidas análogas	14.73
COSTO TOTAL por cada tarjeta:	84.74

4.4.1 ANALISIS TECNICO

En resumen la tarjeta ADAIO presenta las siguientes características técnicas:

- **Puertos Análogos de salida:** hasta un máximo de 8 puertos, multiplexados de 8 bits y activados por programa, cuya señal de salida debe estar entre 0 y 10 VDC.

⇒ Frecuencia máxima de generación depende del programa. Ver tabla 4.3.2.a

⇒ Tiempo máximo de retardo en software para la activación y envío de

cada una de las señales esta dada en la tabla 4.2.3.b.

- **Puertos Análogos de entrada:** hasta un máximo de 8 puertos de 8 bits, activados por programa, cuya señal de entrada debe estar entre 0 y 10 VDC.

⇒ La máxima frecuencia de adquisición no depende del lenguaje de programación

⇒ Tiempo máximo de retardo en software para la activación y recepción entre cada una de las señales de entrada es de 100 mseg.

- **Puertos Digitales:** Acceso a 3 puertos digitales de 8 bits totalmente programables. Ver Capítulo V (Manual del usuario).

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES.

Después de realizar las pruebas experimentales de la tarjeta ADAIO, como se pudo observar en los resultados, considerando sus características técnicas, el número de pórtilos análogos tanto de entrada como de salida y, por último su costo en general, se puede decir que esta tarjeta puede ser una herramienta poderosa para quien incursione en el campo de la electrónica y computación, ya que, por medio de ésta se puede controlar un sin número de elementos externos.

Luego de implementar el diseño tanto en software como en hardware se podría establecer las siguientes conclusiones:

5.1 CONCLUSIONES.

- ◆ Después de haber realizado las pruebas de funcionamiento se puede afirmar que se ha construido una tarjeta cuyas características técnicas la vuelven apta para ser implementada con fines didácticos, ya que la tarjeta ADAIO permite una gran facilidad de integración para cualquier computador personal.
- ◆ Los puertos disponibles de la tarjeta ADAIO permiten que ésta se constituya como parte integrante de un computador personal tipo PC compatible, el cual puede servir para el control de procesos en donde se controle fenómenos físicos (temperatura, presión, nivel de fluidos, etc), que tengan un período de repetición máximo de 0.1 segundos.

- ◆ La facilidad en el ajuste de direcciones, permite que dentro de un mismo computador puedan existir hasta un máximo de 4 tarjetas, siempre y cuando el computador tenga ranuras de extensión tipo ISA de 8 ó de 16 bits.
- ◆ La tarjeta ADAIO está diseñada de tal manera que cumple con requisitos estándares, ya que el uso de señales normalizadas permite su fácil integración en cualquier tipo de sistema.
- ◆ Se debe resaltar que ésta tarjeta ha funcionado perfectamente en ambientes multiusuario, multitarea como lo es en Windows 95, lo que permite realizar actividades adicionales en el mismo computador.
- ◆ La tarjeta ADAIO también constituye una tarjeta amplificador de pórticos digitales programables, lo cual implica que proyectos anteriores realizados con el pórtico paralelo pueden ser implementados igualmente y con mayor facilidad utilizando ésta tarjeta.
- ◆ El usuario de la tarjeta ADAIO puede escoger el lenguaje que más domine para realizar proyectos personales a partir de ésta. Esto es gracias a la gran cantidad de lenguajes de programación existentes para computadores personales PC's compatibles.
- ◆ La tarjeta ADAIO por su diseño modular permite una gran flexibilidad en cuanto a costos se refiere, dependiendo de la aplicación específica que el usuario quiera darle a ésta. (Ver análisis técnico-económico. Capítulo IV). El costo de implementar toda la tarjeta es de \$ 84.74 (dólares)

5.2 RECOMENDACIONES.

Con el presente proyecto realizado "TARJETA ADAIO", se pueden realizar con ésta un sin número de proyectos en el campo del control computarizado, por lo que se recomienda:

- ◇ Desarrollar proyectos adicionales acoplados a esta tarjeta, pero para ello se considera que se necesitan conocimientos básicos de electrónica en cuanto a hardware se refiere; en cuanto a software su está limitada únicamente por la habilidad del programador.
- ◇ Realizar librerías en los programas de entorno visual tal como: Visual V3.0 ó Lab View, que permitan la fácil operación e integración de la tarjeta a la aplicación realizada por el usuario.
- ◇ Implementar módulos de control que puedan facilitar el uso de la tarjeta ADAIO en aplicaciones específicas, tales como:
 - ⇒ Módulos de potencia que tengan relés para el manejo de elementos de potencia.
 - ⇒ Módulos de entradas optoacopladas.
 - ⇒ Módulos de amplificación de señal para el manejo de sensores, etc.
- ◇ Realizar las correcciones necesarias en software, debido a que los voltajes de referencia para el conversor D/A se han tomado de la fuente del computador, los cuales pueden variar de uno a otro.

- ◇ Implementar proyectos anteriores utilizados con el p rtico paralelo, esta vez en la tarjeta ADAIO, debido a que ella tambi n constituye una tarjeta ampliadora de p rticos digitales programables.
- ◇ Tener cuidado cuando se realiza proyectos a partir de esta tarjeta, para lo cual se debe leer detenidamente el MANUAL de la misma, especialmente en el ac pite relacionado a conexi n con elementos externos.

BIBLIOGRAFIA

- Mark Minasi, GUIA COMPLETA DE MANTENIMIENTO Y ACTUALIZACION DE LA PC, Ed. Ventura, 1993.
- Lewis C. Eggebrecht, INTERFACING TO THE IBM PERSONAL COMPUTER, Ed. Howard W. Sanms & Co., 1987.
- Gary Cornell, MANUAL DE VISUAL BASIC V3.0 PARA WINDOWS, Ed. McGraw-Hill, 1994.
- Hofacker Winfriend, MICROCOMPUTADORES MANUALES, 1992
- ECG Semicondutores, MASTER REPLACEMENT GUIDE, Edici n 1996
- Linear Circuits, DATA ACQUISITION AND CONVERSION, Texas Instruments, 1989.
- Texas Instruments, LABVIEW FOR WINDOWS, 1994.
- Del Pozo Andrade, HERRAMIENTA GENERADORA DE SISTEMAS DE AYUDA PARA WINDOWS, Tesis Sistemas 1996.

ANEXOS

A GRAFICOS DE LA TARJETA.

A.1 Diagrama de Bloques del Diseño

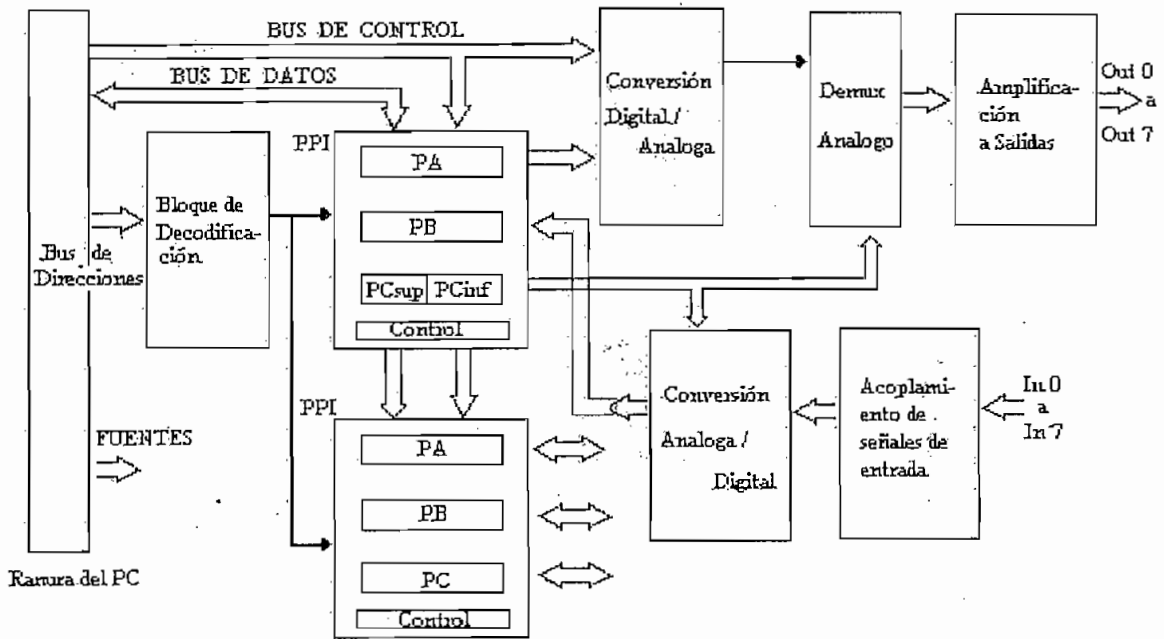


Figura A.2.1 Diagrama de bloques de la tarjeta ADAIO.

A.2 Diagrama esquemático del Diseño.

A.3 Gráfico de la Disposición de elementos en la tarjeta.

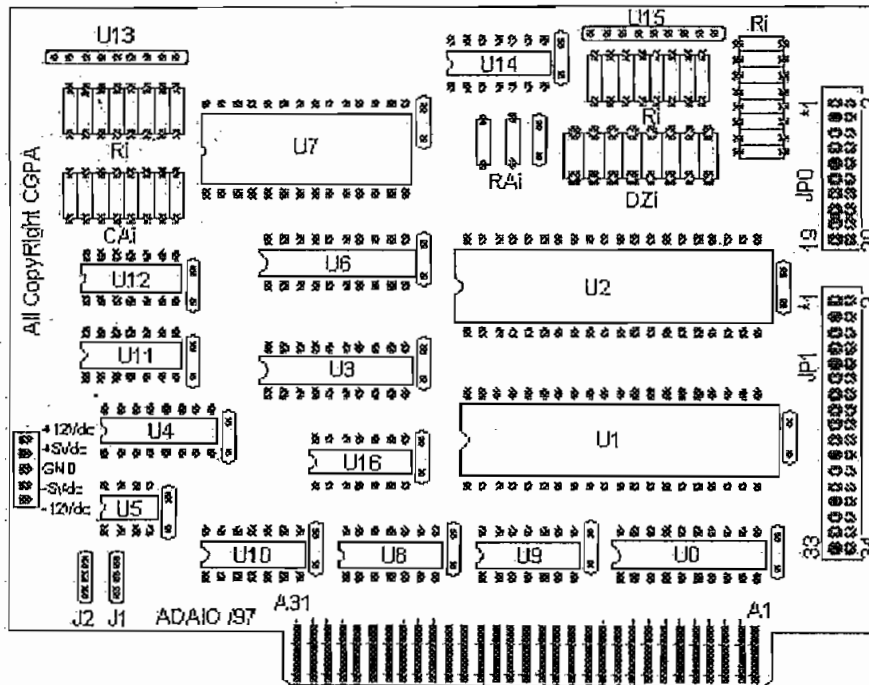


Figura A.3 Disposición de elementos en la tarjeta ADAIO

B MANUAL DEL USUARIO.

B.1 CARACTERISTICAS GENERALES

En el diagrama de bloques mostrado en la figura A.2.1 se puede observar que

La tarjeta pone a disposición del usuario los siguientes recursos:

- 8 Entradas Analógicas (0 a 10VDC), conectadas a una conversor A/D de 8 bits, a través de diodos de protección para cada entrada, con desacoplamiento de impedancia.
- 8 Salidas Analógicas (0 a 10VDC), multiplexadas con punto de tierra común, señales que provienen de un conversor D/A de 8 bits, correspondiendo el valor digital 00H a la señal de 0 VDC y FFH a la de 10 VDC
- Pórticos digitales programables por el usuario, los cuales dependen del modo de programación que sea seleccionado para el segundo PPI, por lo tanto se puede tener:

En el modo cero (entradas / salidas básicas).-

- * 3 Pórticos digitales de salida de 8 bits, ó;
- * 3 Pórticos digitales de entrada de 8 bits, ó ;
- * Las respectivas combinaciones con los pórticos anteriores

En este modo los datos de salida son retenidos.

En Modo uno (entradas / salidas validadas).-

- * 2 Pórticos digitales de entrada (PA ó PB), de 8 bits ó;
- * 2 Pórticos digitales de salida (PA ó PB), de 8 bits.

Los pórticos digitales (PA, PB) son controlados por el pórtilo PC del PPI.

En este modo los datos tanto de entrada como de salida tienen pueden ser retenidos.

En el Modo 2 (bus bidireccional).-

- * Un pórtilo bidireccional (PA), controlado por las líneas superiores del

pórtico C (PC3 a PC7)

- * Un pórtico de entrada ó salida que puede trabajar en modo 0 ó modo 1.

B.2 CONFIGURACION

La tarjeta ADAIO posee una circuitería básica la cual se puede asociar a cualquier computador PC compatible, permitiendo al usuario configurar la dirección de la tarjeta, para el efecto la tarjeta tiene 2 Jumper de selección (J1, J2) como se puede ver en la figura A.2.2

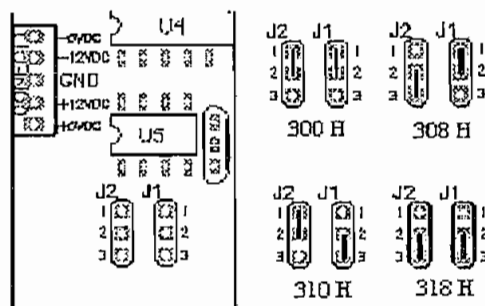


Figura A.2.2 Colocación de puentes en J1, J2 para el seteo de dirección base

- La tarjeta ADAIO debe ser instalada en una ranura de ocho bits para tarjetas de extensión de un computador XT/AT/286...../586 compatible. Además ésta tarjeta provee salidas de la fuente de + 5VDC, para usos externos.
- La tarjeta se ha diseñado para facilitar su mantenimiento, por lo que, no se requieren calibraciones internas de ningún tipo.
- La tarjeta ocupa 8 direcciones consecutivas para direccionar los puertos disponibles, a partir de la dirección base, véase en la tabla **Tabla B.2.1 1**

	PPI 1	Utilizado para señales analógicas
Dirección Base	PA	Utilizado para salidas análogas (1 a 8)
Dirección Base + 1	PB	Utilizado para entradas analógicas (1 a 8)
Dirección Base + 2	PC	Utilizado para determinar que entrada/salida análoga se desea.
Dirección Base + 3	WC	Palabra de control (82H)
	PPI 2	Utilizado para señales digitales

Dirección Base + 4	PA	Pórtico de I/O ó Bidireccional
Dirección Base + 5	PB	Pórtico de I/O
Dirección Base + 6	PC	Pórtico de I/O ó de Control.
Dirección Base + 7	WC	Palabra de control para determinar el modo de funcionamiento del PPI 2.

Tabla B.2.1 1

B.3 INSTALACION

Antes de instalar la tarjeta ADAIO se debe leer cuidadosamente el acápite anterior de seteo de direcciones permitidas y definir la dirección apropiada en concordancia con sus requerimientos.

Es muy importante definir correctamente la dirección que se asignará a la tarjeta ADAIO, para que ésta no entre en conflicto con otras tarjetas previamente instaladas.

Para instalar la tarjeta siga los siguientes pasos:

- a.- Apague el computador, y todo periférico a éste conectado,
- b.- Remueva la cubierta del computador
- c.- Verifique que una de las láminas localizadas en la parte posterior del computador, correspondiente a una ranura, esté libre, si no lo esta remuévala.
- d.- Seleccione la dirección adecuada que se dará a la tarjeta (ver figura de configuración anterior)
- e.- Instale apropiadamente los cables que se utilizaran en su aplicación. (Ver acápite B.4 acerca de la conexión con elementos periféricos)
- f.- Instale la tarjeta en una ranura para bus tipo ISA. (de 8 o 16 bits, la tarjeta

sólo ocupara la de 8 bits).

- g.- Asegúrese de la correcta ubicación de la tarjeta y cables.
- h.- Encienda el computador y verifique el correcto funcionamiento de la tarjeta utilizando para este efecto cualquiera de los programas adjuntos a la tarjeta (Ver sección software de la tarjeta ADAIO).
- i.- Si la operación es correcta, apague el computador y restituya la cubierta.
- j.- Si la operación no es correcta apague el computador y verifique la dirección base dada a la tarjeta y vuelva a ejecutar los pasos verificando la conexión de los cables desde el paso (e).

B.4. CONEXION

Para establecer un enlace entre el PC y un periférico de aplicación específico la tarjeta ADAIO POSEE DOS CONECTORES EXTERNOS Y DOS INTERNOS, dispuestos de la siguiente manera:

- Un conector interno de 62 pines el cual sirve para conectar la tarjeta ADAIO con la ranura del computador. Esta tarjeta utiliza la fuente del computador para activar alimentar todos los circuitos integrados que la conforma.
- Un conector interno de 5 pines, por medio del cual se tiene acceso a las fuentes(+12Vdc, + 5Vdc, GND, - 5Vdc, - 12Vdc), que provee el computador. Para aplicaciones externas se debe tener cuidado al utilizar estas fuentes, por lo que se recomienda utilizarlas con protección externa adecuada (fusibles de 1 A).
- Un conector macho de 20 pines, el cual, se utiliza para sacar las señales análogas, tanto de entrada como de salida. Conector denominado JP0. (Ver figura A.2), las señales están dispuestas de la tabla B.4.1

# Pin del conector JP0	Señales	# Pin del conector JP0	Señales
1	Vout 0	11	GND
2	Vout 1	12	+ 5 VDC
3	Vout 2	13	Vin 0
4	Vout 3	14	Vin 1
5	Vout 4	15	Vin 2
6	Vout 5	16	Vin 3
7	Vout 6	17	Vin 4
8	Vout 7	18	Vin 5
9	GND	19	Vin 6
10	+5 VDC	20	Vin 7

Tabla B.4.1

- Un conector macho de 34 pines, por medio del cual se tiene acceso externamente a todas las señales de PPI 2, conector denominado JP1. (Ver figura A.2), las señales están dispuestas de la tabla B.4.2

# Pin del conector JP1	Señales	# Pin del conector JP1	Señales
1	PA 0	18	GND
2	PA 1	19	GND
3	PA 2	20	GND
4	PA 3	21	NC
5	PA 4	22	NC
6	PA 5	23	+ 5 VDC
7	PA 6	24	+ 5 VDC
8	PA 7	25	+ 5 VDC
9	PB 0	26	+ 5 VDC
10	PB 1	27	PC 0
11	PB 2	28	PC 1
12	PB 3	29	PC 2
13	PB 4	30	PC3
14	PB 5	31	PC4
15	PB 6	32	PC5
16	PB 7	33	PC6
17	GND	34	PC7

Tabla B.4.2

Los cables conectores utilizados deben ser cables RIBBON de 34 ó 20 hilos, respectivamente, con una particularidad, de que **el hilo rojo del cable debe conectarse al pin # 1 del conector macho de la tarjeta.**

Los cables son de fácil elaboración, pues existen en el mercado local, cables

ribbon de 34 ó 20 hilos y sus respectivos conectores hembra, para éste propósito se deben ensamblar los cables como se indican en la figura B.4.1

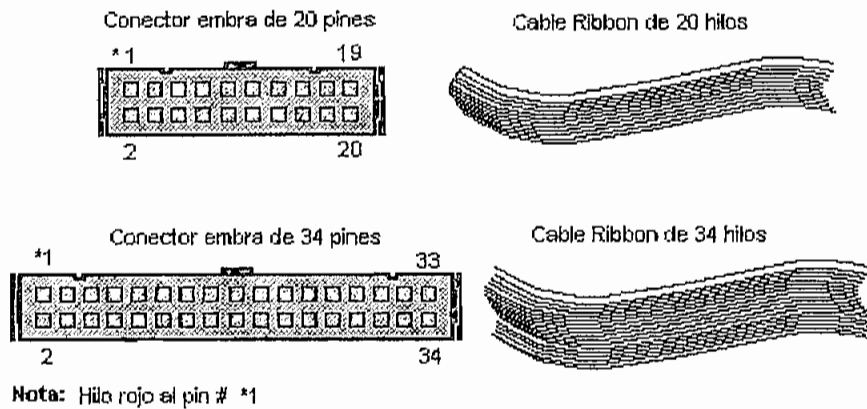


Figura B.4.1

Se recomienda que para la implementación de proyectos externos se conecte el un extremo del cable a la tarjeta según los conectores indicados anteriormente, y en el otro extremo del cable; se separen los hilos y se conecten separadamente en un protoboard, lo cual permitirá medir cada una de las señales a utilizar

Consideraciones que se deben tomar en la conexión de elementos externos.-

Los puertos de Entrada y Salida de elementos NMOS como el 8255A fueron contruidos de tal manera que permite una corriente de salida 2 mA en conexión de sumidero o <<sinking>> a 0.5V, y de 0.2mA en conexión fuente o sourcing a 2.4V.

Por lo anterior es recomendable, realizar con ellos las conexiones indicadas en la figura B.4.2. **Conexiones que permiten el manejo de cargas de potencia desde una salida del 8255** en conexión de sumidero *sinking*, y en conexión de

fuente o *sourcing*.

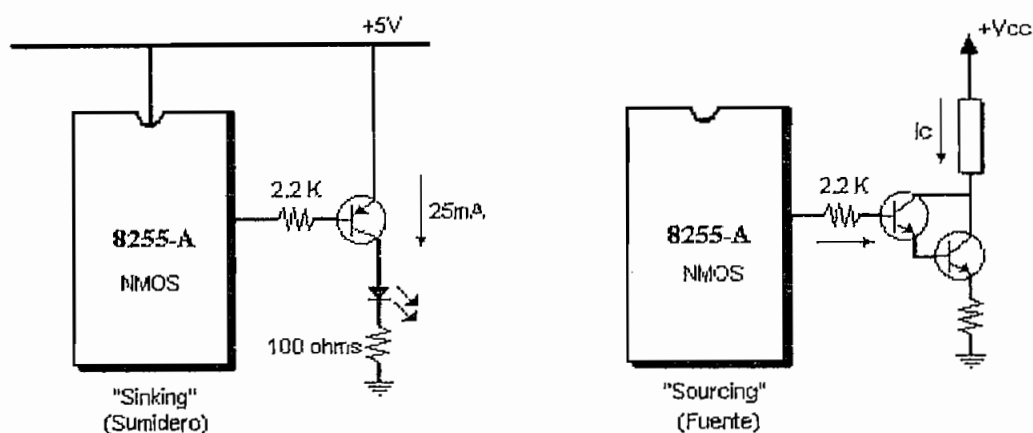


Figura B.4.2. Manejo adecuado de cargas de potencia desde salidas NMOS

TABLERO DE CONEXIONES Y PRUEBAS DE LA TARJETA ADAIO

Se ha desarrollado un tablero de conexiones que tiene elementos necesarios para probar el buen funcionamiento de la tarjeta ADAIO. Como se sabe la tarjeta tiene puertos digitales (A, B y C), y 8 analógicos de entrada y 8 de salida, los cuales se encuentra claramente diferenciados; además se puede identificar claramente cuales son los bits más y menos significativos en cada caso (el terminal 0 es el bit menos significativo); esto se da, para poder probar más rápidamente cada uno de los pórtricos, lográndose una mayor agilidad al momento de experimentar con nuevos prototipos, a partir de este tablero.

De la misma manera, la alimentación proveniente de la fuente (5 voltios) se encuentra en esta tarjeta; este voltaje se puede aprovechar para aquellos casos en los cuales la demanda de corriente no es muy elevada. Si la corriente es mayor lo recomendable es utilizar una fuente externa a la tarjeta, recordando que

es necesario que tanto la tierra del circuito como la del computador estén unidas.

Además en el tablero se tiene:

- 8 leds para cada pórtricos (A, B y C), con el fin de probar la salida de los pórtricos digitales, cuando estos estén programados de esta manera. En el tablero se lo puede identificar como "Outd" a los conectores de éstos pórtricos.
- 3 dip switch de 8 con el fin de probar la entrada de los pórtricos digitales, cuando estos estén programados de esta manera. En el tablero se lo puede identificar como "Ind" a los conectores de éstos pórtricos.
- 2 conectores de 8 zócalos cada uno que servirán para conectar tanto las entradas como las salidas análogas identificadas como "Ina" y "Outa" respectivamente. En estos conectores se puede identificar claramente el número de la señal.

El tablero de conexiones se encuentra unido al computador a través de dos cables *ribbon* uno de 34 líneas para las señales digitales y, otro de 20 líneas para las señales análogas.

⇒ Para probar los pórtricos de salida digitales, se instaló la configuración que se indica en la figura B.4.3. para cada pórtrico.

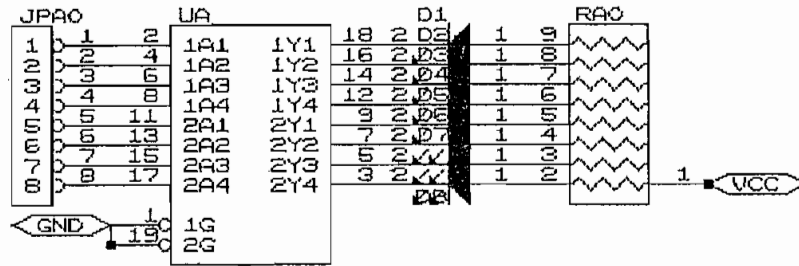


Figura B.4.3 Diagrama de elementos para probar el p3rtico de salidas digitales

Como se puede observar, la salida del puerto se lleva a un circuito integrado UA (74LS240), el cual es un buffer de 8 l3neas. Este es necesario debido a que las salidas del puerto no pueden manejar directamente corrientes elevadas (m3ximo 4mA por l3nea), adem3s se conecta una banco de resistencias RAO (330 Homs), y un banco de diodos D0 a D8, los cuales nos indican en forma visual el bit activo del p3rtico.

El buffer est3 en capacidad de manejar directamente 600 mA en total, lo que los convierte en aptos para controlar LEDs y, es el indicado para manejar cargas inductivas como rel3s, motores paso a paso, etc.

⇒ Para probar los p3rticos de entrada digitales, se instal3 la configuraci3n que se indica en la figura B.4.4 para cada p3rtico.

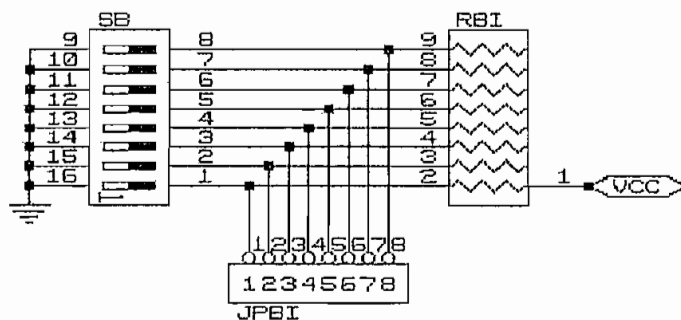


Figura B.4.4 Diagrama de elementos para probar el p3rtico de entradas digitales

Como se puede observar en la figura anterior B.4.4 se ha implementado una serie de 8 interruptores DIP, para cada puerto, de tal manera que se puede leer su estado; los suiches conmutan la línea que entra al computador entre estados lógicos altos y bajos.

Lo ideal es tener resistencias *pull-up* ($RBI = 4.7K\Omega$) conectadas al puerto, tal como se muestra en la figura anterior, cuando se cierra uno de los interruptores la línea del puerto correspondiente pasa a un estado lógico bajo.

C SOFTWARE

C.1. ASIGNACION DE DIRECCION BASE

La dirección base asignada en los programas de aplicación deben tener plena concordancia con el hardware Ver B.2 CONFIGURACION

C.2. ASIGNACION DE DIRECCION DE SEÑALES ANALOGAS

Para acceder a las señales análogas localizadas en el conector externo de 20 pines, por medio de programa se debe :

- Escribir en el registro que pertenece al PC (dirección base + 2), del PPI 1 un número que habilitará la señal análoga deseada.
- Si se desea salidas análogas, se deberá escribir en el PA del PPI 1, (Dirección base) un número que corresponderá a un voltaje deseado, de esta forma el número 0 equivale a 0Vdc y el número 255 equivaldrá a 10 Vdc.

- Si se desea adquirir entradas análogas, el voltaje adquirido por medio del conversor A/D se traducirá a un número que es localizado en el PB del PPI 1, (Dirección base + 1).

Pórtico #	Análogo	Localizado en el pin # de PA, del PPI 1. (hardware)	# HEX que debe ser escrito en el PC(dirección base +2), del PPI 1. (software).
	Vout 0	PA0	00H
	Vout 1	PA1	10H
	Vout 2	PA2	20H
	Vout 3	PA3	30H
	Vout 4	PA4	40H
	Vout 5	PA5	50H
	Vout 6	PA6	60H
	Vout 7	PA7	70H
	Vin 0	PB0	00H
	Vin 1	PB1	01H
	Vin 2	PB2	02H
	Vin 3	PB3	03H
	Vin 4	PB4	04H
	Vin 5	PB5	05H
	Vin 6	PB6	06H
	Vin 7	PB7	07H

D CONSIDERACIONES TECNICAS

La tarjeta ADAIO presenta las siguientes características técnicas:

- **Puertos Análogos de salida:** hasta un máximo de 8 puertos, multiplexados de 8 bits y activados por programa, cuya señal de salida debe estar entre 0 y 10 VDC.

⇒ Frecuencia máxima de generación depende del programa. Ver tabla

4.3.2.a

RESULTADOS DE GENERACION DE DATOS CON LA TARJETA ADAIO		
	Frecuencia Máxima (Hz)	Frecuencia Máxima (Hz)
	(1 Salida Activa)	(8 Salidas Activas)
Turbo C++	1562.5	227.3
Q-Basic	71.4	27.8
Visual Basic	90.9	43.5
Lab. View	526.3	

Tabla 4.2.3.a Cuadro de frecuencias de generación máxima

⇒ Tiempo máximo de retardo en software para la activación y envío de cada una de las señales esta dada en la tabla 4.2.3.b.

MINIMO TIEMPO DE ESPERA PARA ENVIO ENTRE SEÑAL Y SEÑAL (Medido en Visual Basic V3.0)								
SEÑAL	1	2	3	4	5	6	7	8
En mseg.	0	80	100	120	140	160	180	200

Tabla 4.2.3.b Tiempo máximo de retardo para la activación entre señales de salida

- **Puertos Análogos de entrada:** hasta un máximo de 8 puertos de 8 bits, activados por programa, cuya señal de entrada debe estar entre 0 y 10 VDC.

⇒ La máxima frecuencia de adquisición no depende del lenguaje de programación

Tiempo máximo de retardo en software para la activación y recepción entre cada una de las señales de entrada es de 100 mseg.

E DIAGRAMAS DE FLUJO DE LOS PROGRAMAS

F HOJAS TECNICAS DE LOS PRINCIPALES ELEMENTOS UTILIZADOS.

DIAGRAMA ESQUEMATICO DEL FLUJO DEL PROGRAMA PPI.BAS

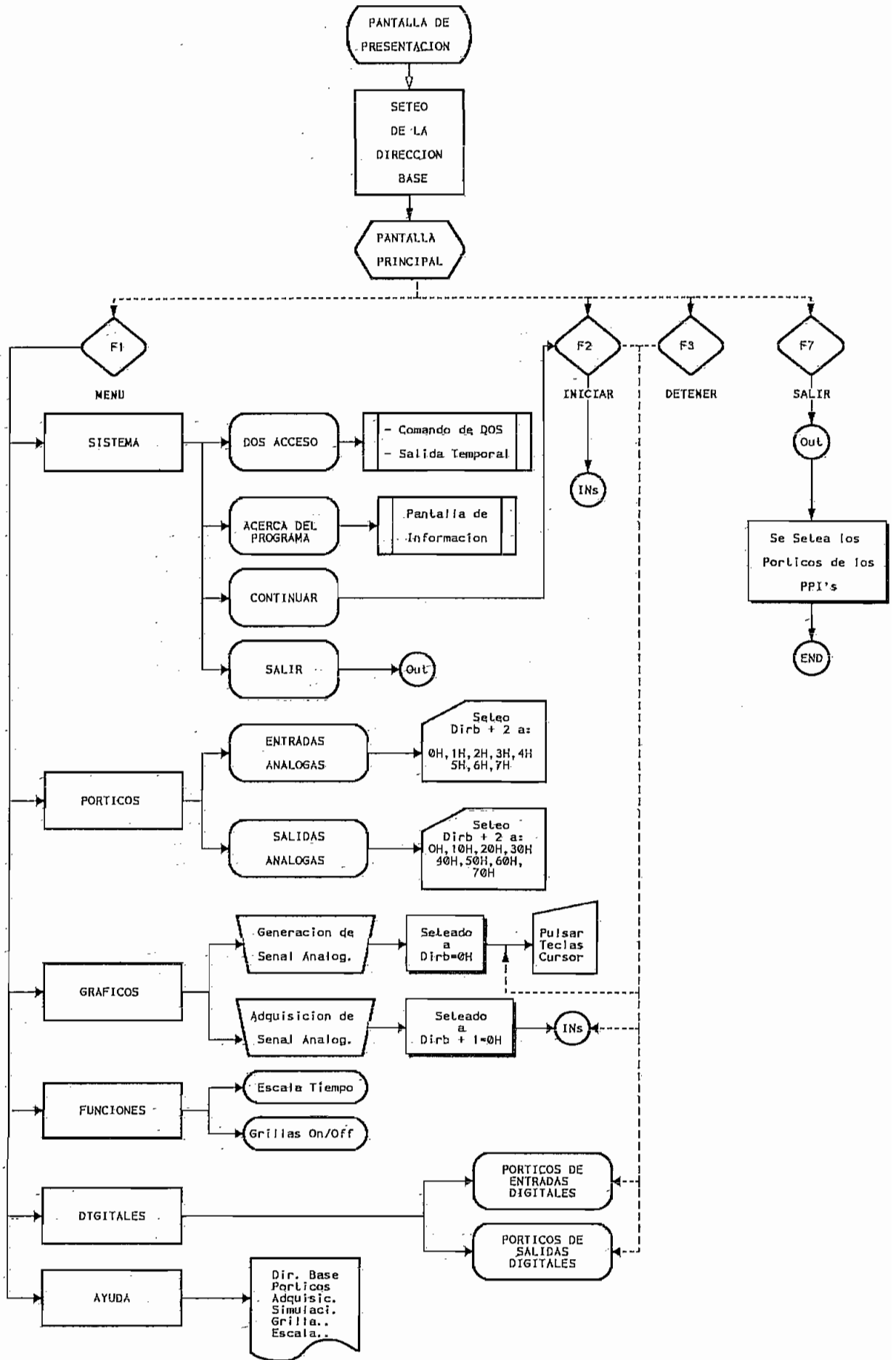
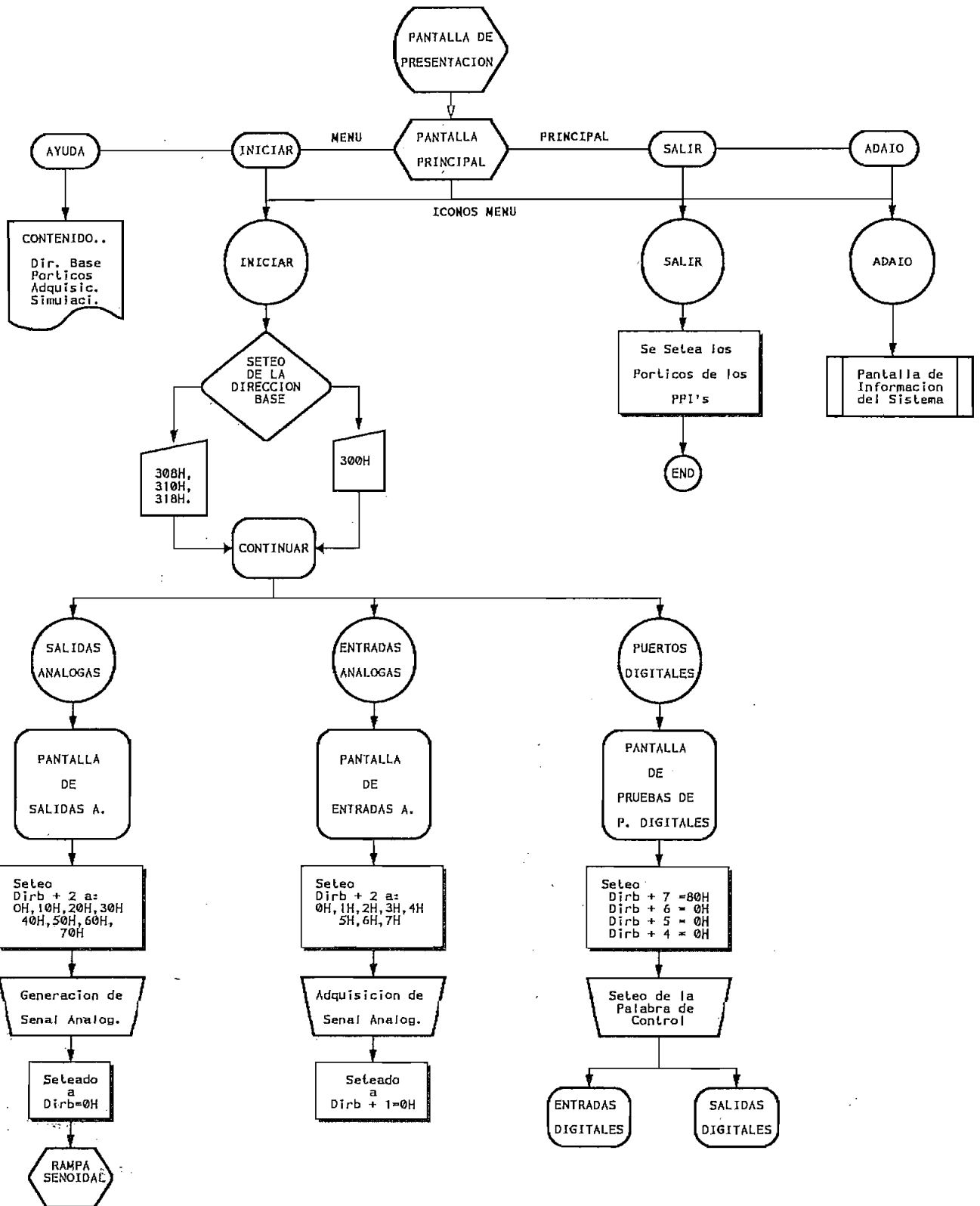


DIAGRAMA ESQUEMATICO DE FLUJO DEL PROGRAMA ADAIO.MAK





8255A/8255A-5 PROGRAMMABLE PERIPHERAL INTERFACE

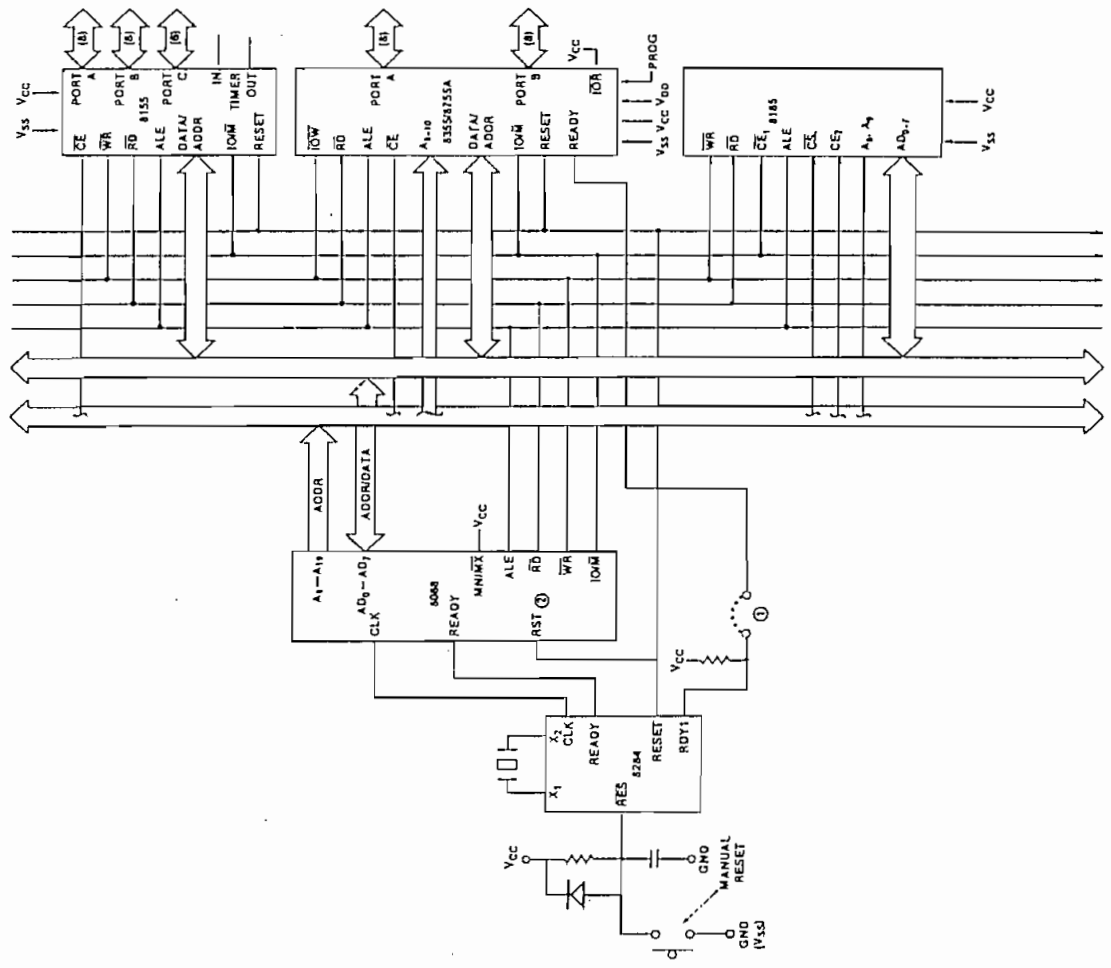
- MCS-85™ Compatible 8255A-5
- Direct Bit Set/Reset Capability Easing Control Application Interface
- 24 Programmable I/O Pins
- 40-Pin Dual In-Line Package
- Completely TTL Compatible
- Reduces System Package Count
- Fully Compatible with Intel® Micro-processor Families
- Improved Timing Characteristics
- Improved DC Driving Capability

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

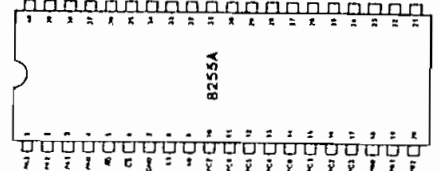
8088 FIVE CHIP SYSTEM

Figure 11b shows a five chip system containing:

- 1.25K Bytes RAM
- 2K Bytes ROM
- 38 I/O Pins
- 1 Interval Timer
- 2 Interrupt Levels



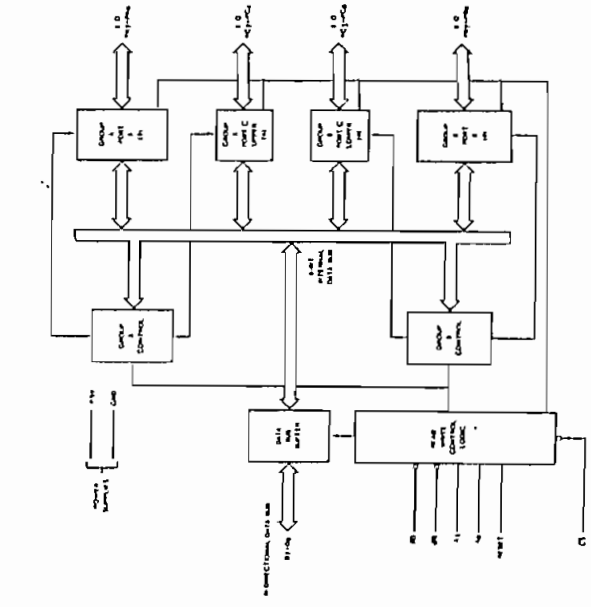
PIN CONFIGURATION



PIN NAMES

Pin	Function
1-4	DATA BUS (BI DIRECTIONAL)
5	RESET INPUT
6	CHIP SELECT
7	READY INPUT
8	WRITE INPUT
9-12	PORT A (BIT)
13-16	PORT B (BIT)
17-20	PORT C (BIT)
21-24	...
25-28	...
29-32	...
33-36	...
37-40	...

8255A BLOCK DIAGRAM



8255A FUNCTIONAL DESCRIPTION

General

The 8255A is a programmable peripheral interface (PPI) device designed for use in Intel® microcomputer systems. Its function is that of a general purpose I/O component to interface peripheral equipment to the microcomputer system bus. The functional configuration of the 8255A is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the 8255A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control buses and in turn, issues commands to both of the Control Groups.

(CS)

Chip Select. A "low" on this input pin enables the communication between the 8255A and the CPU.

(RD)

Read. A "low" on this input pin enables the 8255A to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to "read from" the 8255A.

(WR)

Write. A "low" on this input pin enables the CPU to write data or control words into the 8255A.

(A₀ and A₁)

Port Select 0 and Port Select 1. These input signals, in conjunction with the RD and WR inputs, control the selection of one of the three ports or the control word registers. They are normally connected to the least significant bits of the address bus (A₀ and A₁).

8255A BASIC OPERATION

A ₁	A ₀	RD	WR	CS	INPUT OPERATION (READ)
0	0	0	1	0	PORT A ← DATA BUS
0	1	0	1	0	PORT B ← DATA BUS
1	0	0	1	0	PORT C ← DATA BUS
					OUTPUT OPERATION (WRITE)
0	0	1	0	0	DATA BUS → PORT A
0	1	1	0	0	DATA BUS → PORT B
1	0	1	0	0	DATA BUS → PORT C
1	1	1	0	0	DATA BUS → CONTROL
X	X	X	X	1	DISABLE FUNCTION
1	1	0	1	0	DATA BUS → 3-STATE
X	X	1	1	0	DATA BUS → 3-STATE

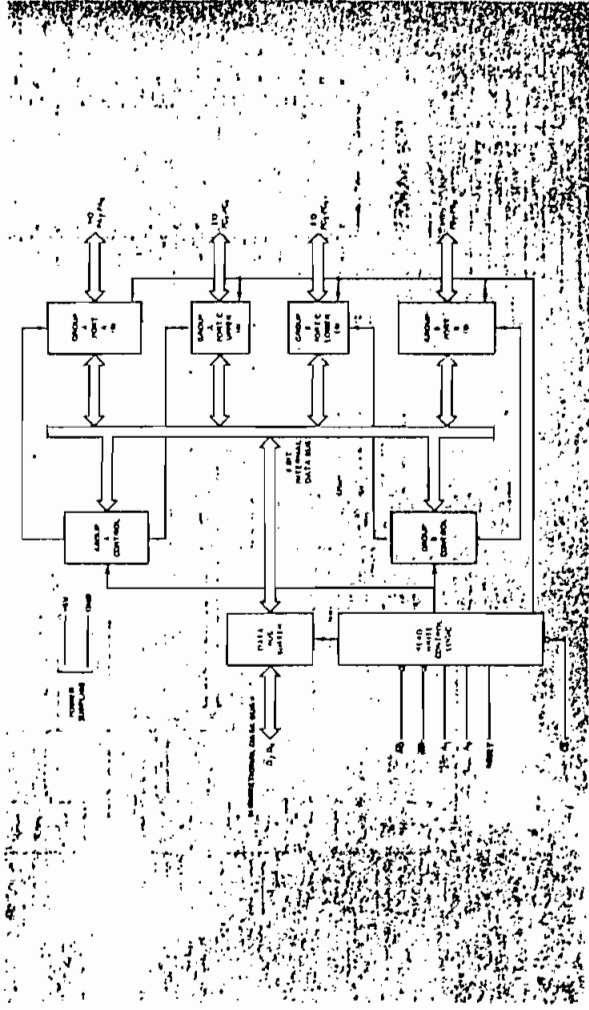


Figure 1. 8255A Block Diagram Showing Data Bus Buffer and Read/Write Control Logic Functions

(RESET)

Reset. A "high" on this input clears the control register and all ports (A, C, B) are set to the input mode.

Group A and Group B Controls

The functional configuration of each port is programmed by the system software. In essence, the CPU "outputs" a control word to the 8255A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 8255A.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A — Port A and Port C upper (C7-C4)

Control Group B — Port B and Port C lower (C3-C0)

The Control Word Register can only be written into. No Read operation of the Control Word Register is allowed.

Ports A, B, and C

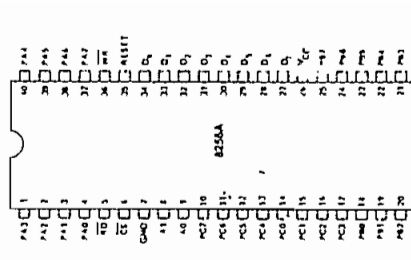
The 8255A contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 8255A.

Port A. One 8-bit data output latch/buffer and one 8-bit data input latch.

Port B. One 8-bit data input/output latch/buffer and one 8-bit data input buffer.

Port C. One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B.

PIN CONFIGURATION



PIN NAMES

U ₁ , O ₁	DATA BUS (BI DIRECTIONAL)
RESET	RESET INPUT
CS	CHIP SELECT
RD	READ INPUT
WR	WRITE INPUT
A ₀ , A ₁	PORT ADDRESS
P _{A7} -P _{A0}	PORT A (BIT)
P _{B7} -P _{B0}	PORT B (BIT)
P _{C7} -P _{C0}	PORT C (BIT)
V _{CC}	+5 VOLTS
GND	0 VOLTS

Figure 2. 8255A Block Diagram Showing Group A and Group B Control Functions

8255A OPERATIONAL DESCRIPTION

Mode Selection

There are three basic modes of operation that can be selected by the system software:

- Mode 0 — Basic Input/Output
- Mode 1 — Strobed Input/Output
- Mode 2 — Bi-Directional Bus

When the reset input goes "high" all ports will be set to the input mode (i.e., all 24 lines will be in the high impedance state). After the reset is removed the 8255A can remain in the input mode with no additional initialization required. During the execution of the system program any of the other modes may be selected using a single output instruction. This allows a single 8255A to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance: Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.

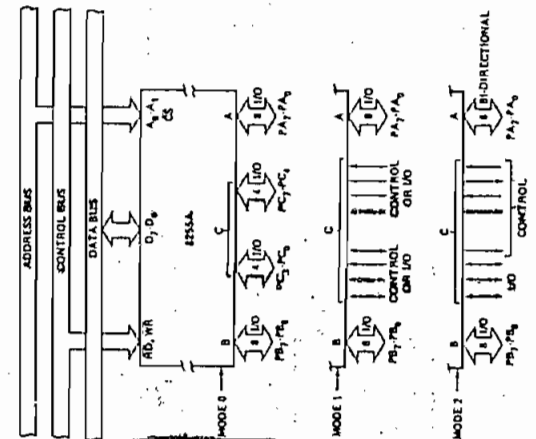


Figure 3. Basic Mode Definitions and Bus Interface

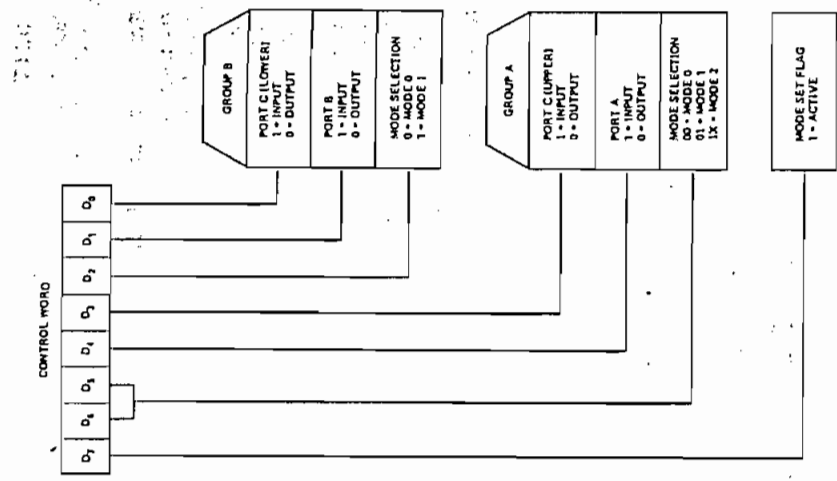


Figure 4. Mode Definition Format

The mode definitions and possible mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 8255A has taken into account things such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTPUT instruction. This feature reduces software requirements in Control-based applications.

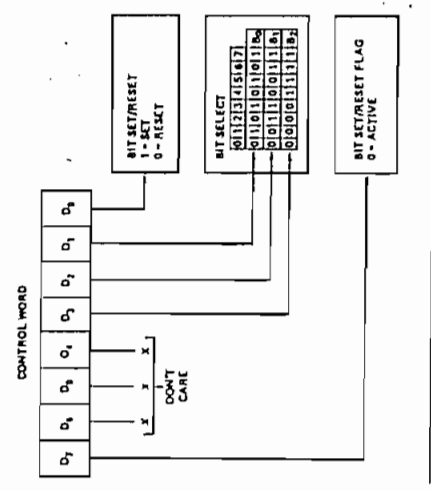
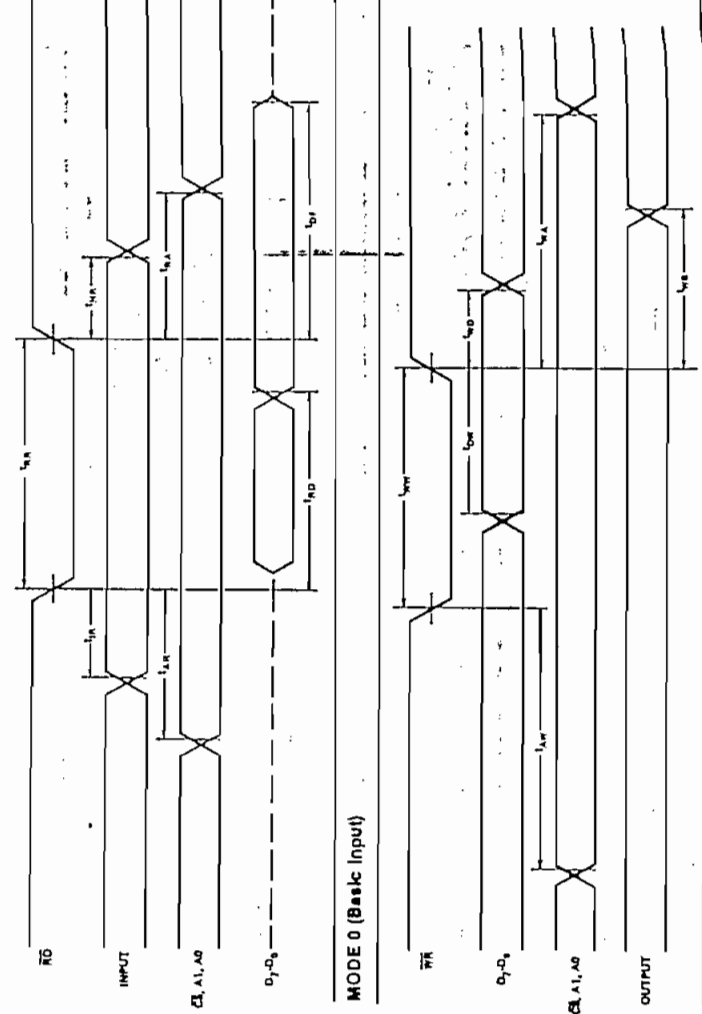


Figure 5. Bit Set/Reset Format

Operating Modes

- Mode 0 Basic Functional Definitions:
- Two 8-bit ports and two 4-bit ports.
 - Any port can be input or output.
 - Outputs are latched.
 - Inputs are not latched.
 - 16 different Input/Output configurations are possible in this Mode.

MODE 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to or read from a specified port.



MODE 0 (Basic Output)

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

Interrupt Control Functions

When the 8255A is programmed to operate in mode 1 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

This function allows the Programmer to disallow or allow a specific I/O device to interrupt the CPU without affecting any other device in the interrupt structure.

INTE flip-flop definition:

- (BIT-SET) — INTE is SET — Interrupt enable
- (BIT-RESET) — INTE is RESET — Interrupt disable

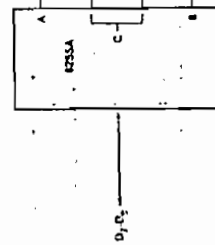
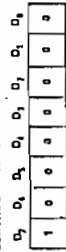
Note: All Mask flip-flops are automatically reset during mode selection and device Reset.

MODE 0 Port Definition

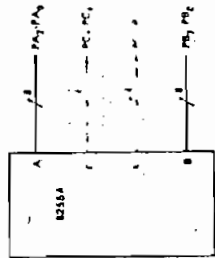
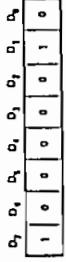
D4	GROUP A			GROUP B		
	PORT A	PORT C (UPPER)	#	PORT B	PORT C (LOWER)	
0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT	
0	OUTPUT	OUTPUT	1	OUTPUT	INPUT	
0	OUTPUT	OUTPUT	2	INPUT	OUTPUT	
0	OUTPUT	OUTPUT	3	INPUT	INPUT	
0	OUTPUT	INPUT	4	OUTPUT	OUTPUT	
0	OUTPUT	INPUT	5	OUTPUT	INPUT	
0	OUTPUT	INPUT	6	INPUT	OUTPUT	
0	OUTPUT	INPUT	7	INPUT	INPUT	
1	INPUT	OUTPUT	8	OUTPUT	OUTPUT	
1	INPUT	OUTPUT	9	OUTPUT	INPUT	
1	INPUT	OUTPUT	10	INPUT	OUTPUT	
1	INPUT	OUTPUT	11	INPUT	INPUT	
1	INPUT	INPUT	12	OUTPUT	OUTPUT	
1	INPUT	INPUT	13	OUTPUT	INPUT	
1	INPUT	INPUT	14	INPUT	OUTPUT	
1	INPUT	INPUT	15	INPUT	INPUT	

MODE 0 Configurations

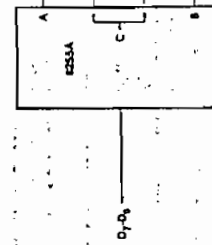
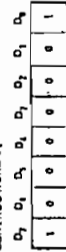
CONTROL WORD #0



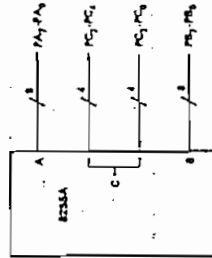
CONTROL WORD #2



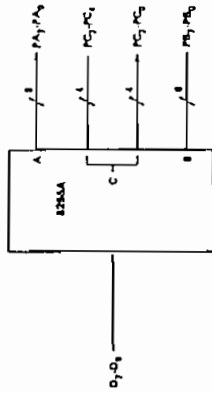
CONTROL WORD #1



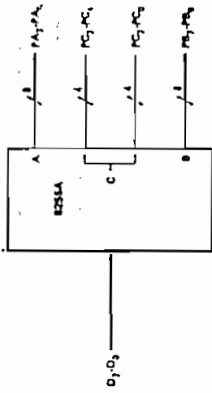
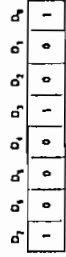
CONTROL WORD #3



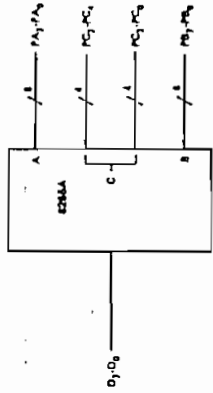
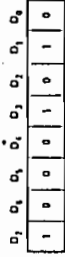
CONTROL WORD #4



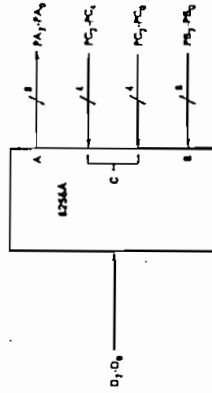
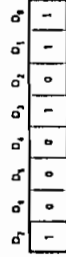
CONTROL WORD #5



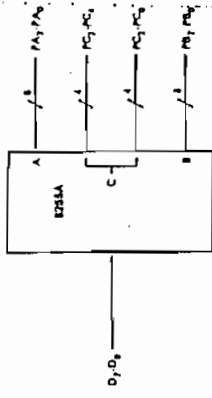
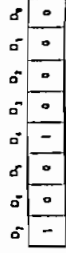
CONTROL WORD #6



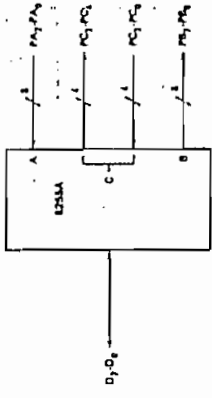
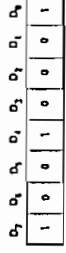
CONTROL WORD #7



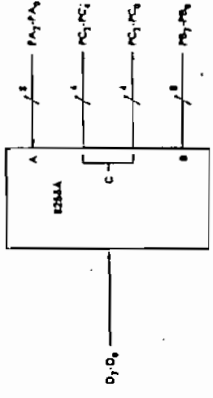
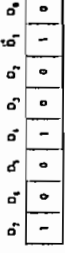
CONTROL WORD #8



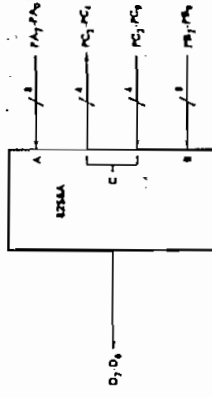
CONTROL WORD #9

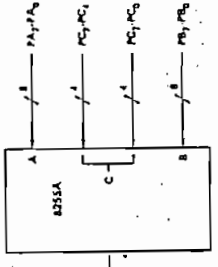
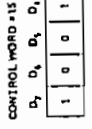
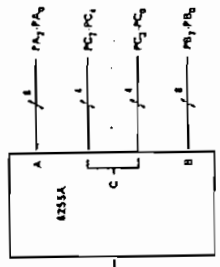
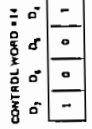
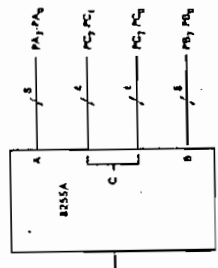
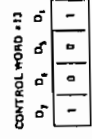
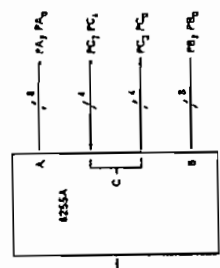
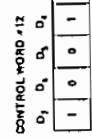


CONTROL WORD #10



CONTROL WORD #11





Input Control Signal Definition

STB (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that the data has been loaded into the input latch; in essence, an acknowledgement. IBF is set by STB input being low and is reset by the rising edge of the RD input.

INTR (Interrupt Request)

A "high" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set by the STB is a "one", IBF is a "one" and INTE is a "one". It is reset by the falling edge of RD. This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

INTE A

Controlled by bit set/reset of PC₄.

INTE B

Controlled by bit set/reset of PC₂.

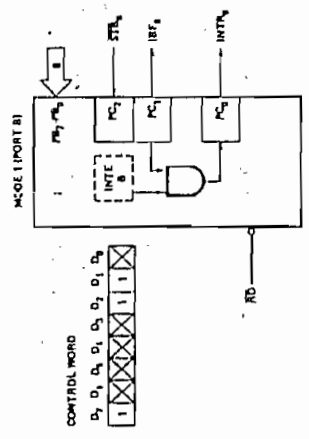
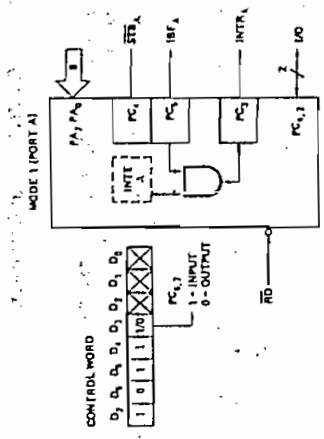


Figure 6. MODE 1 Input

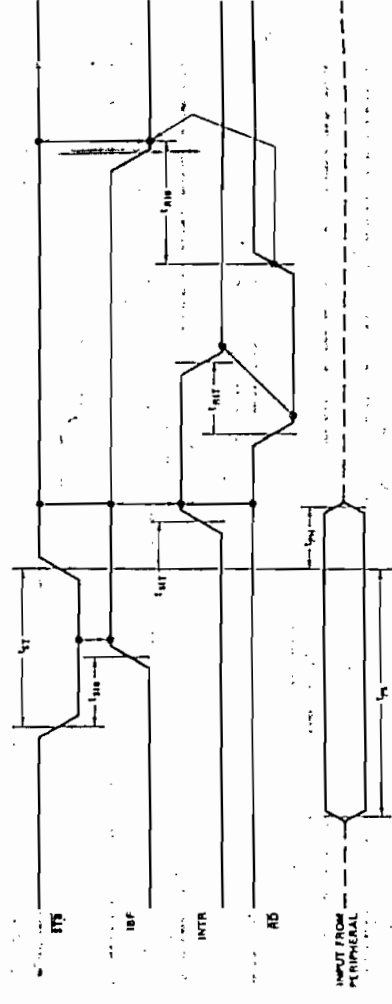


Figure 7. MODE 1 (Strobed Input)

Operating Modes

MODE 1 (Strobed Input/Output). This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "handshaking" signals. In mode 1, port A and Port B use the lines on port C to generate or accept these "handshaking" signals.

Mode 1 Basic Functional Definitions:

- Two Groups (Group A and Group B)
- Each group contains one 8-bit data port and one 4-bit control/data port.
- The 8-bit data port can be either input or output. Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit data port.

Output Control Signal Definition

OBF (Output Buffer Full FIF). The OBF output will go "low" to indicate that the CPU has written data out to the specified port. The OBF FIF will be set by the rising edge of the WR input and reset by ACK input being low.

ACK (Acknowledge Input). A "low" on this input informs the 8255A that the data from port A or port B has been accepted. In essence, a response from the peripheral device indicating that it has received the data output by the CPU.

INTR (Interrupt Request). A "high" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set when ACK is a "one", OBF is a "one" and INTE is a "one". It is reset by the falling edge of WR.

INTE A

Controlled by bit set/reset of PC6.

INTE B

Controlled by bit set/reset of PC2.

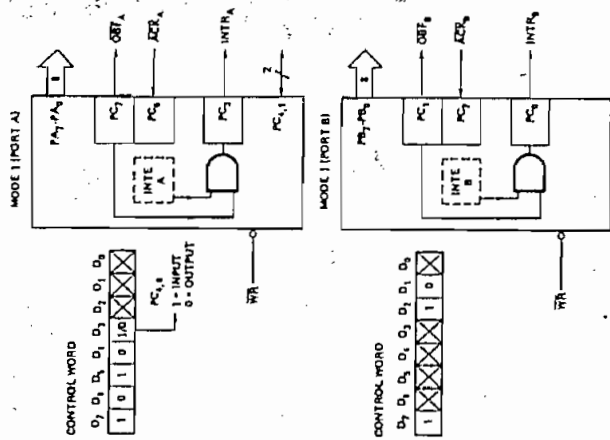


Figure 8. MODE 1 Output

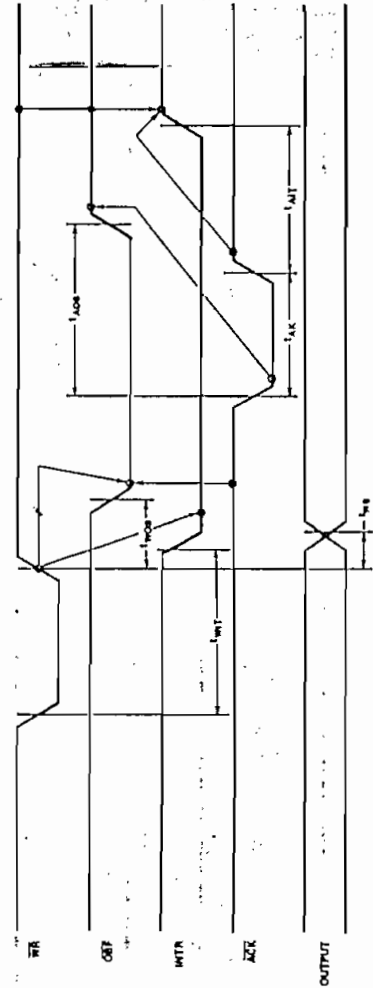


Figure 9. Mode 1 (Strobed Output)

Combinations of MODE 1

Port A and Port B can be individually defined as input or output in Mode 1 to support a wide variety of strobed I/O applications.

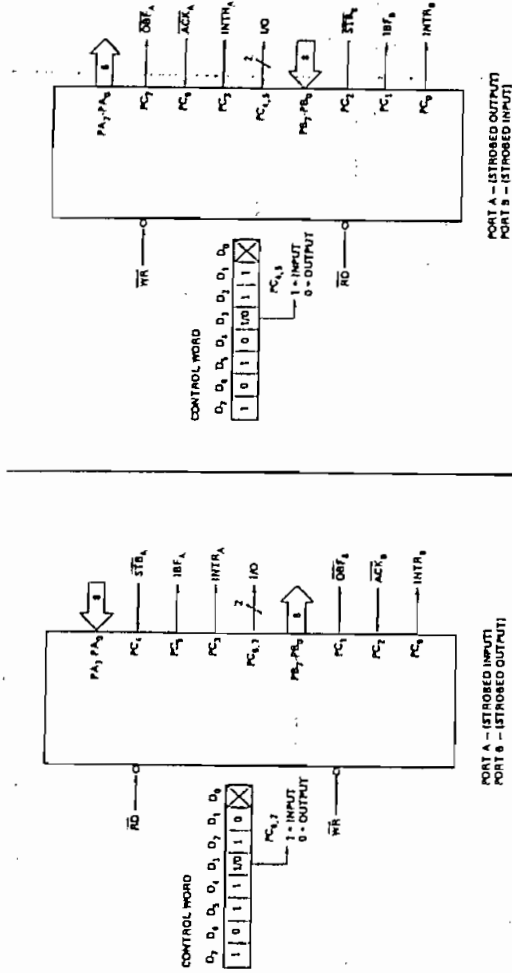


Figure 10. Combinations of MODE 1

Operating Modes

MODE 2 (Strobed Bidirectional Bus I/O). This functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bidirectional bus I/O). "Handshaking" signals are provided to maintain proper bus flow discipline in a similar manner to MODE 1. Interrupt generation and enable/disable functions are also available.

MODE 2 Basic Functional Definitions:

- Used in Group A only.
- One 8-bit, bi-directional bus Port (Port A) and a 5-bit control Port (Port C).
- Both inputs and outputs are latched.
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional bus port (Port A).

Bidirectional Bus I/O Control Signal Definition

INTR (Interrupt Request). A high on this output can be used to interrupt the CPU for both input or output operations.

Output Operations

OBF (Output Buffer Full). The OBF output will go "low" to indicate that the CPU has written data out to port A.

ACK (Acknowledge). A "low" on this input enables the tri-state output buffer of port A to send out the data. Otherwise, the output buffer will be in the high impedance state.

INTE 1 (The INTE Flip-Flop Associated with OBF). Controlled by bit set/reset of PC6.

Input Operations

STB (Strobe Input)

STB (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full FIF). A "high" on this output indicates that data has been loaded into the input latch.

INTE 2 (The INTE Flip-Flop Associated with IBF). Controlled by bit set/reset of PC4.

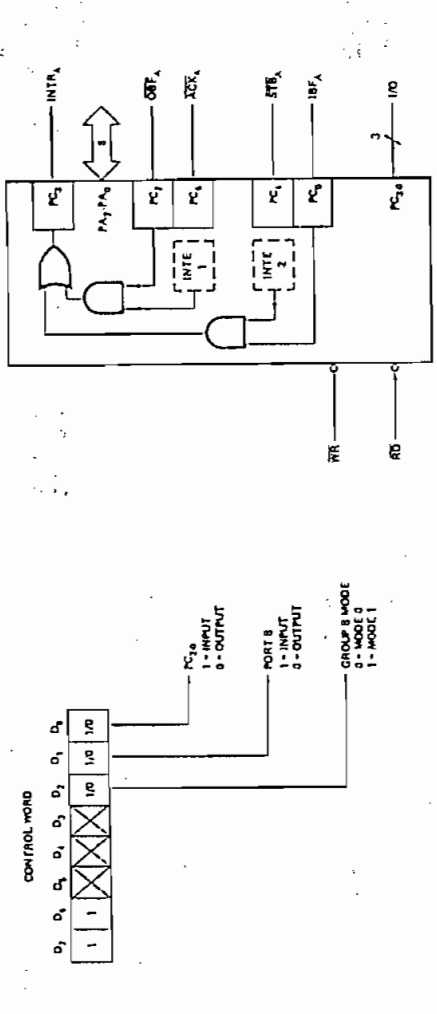


Figure 11. MODE Control Word

Figure 12. MODE 2

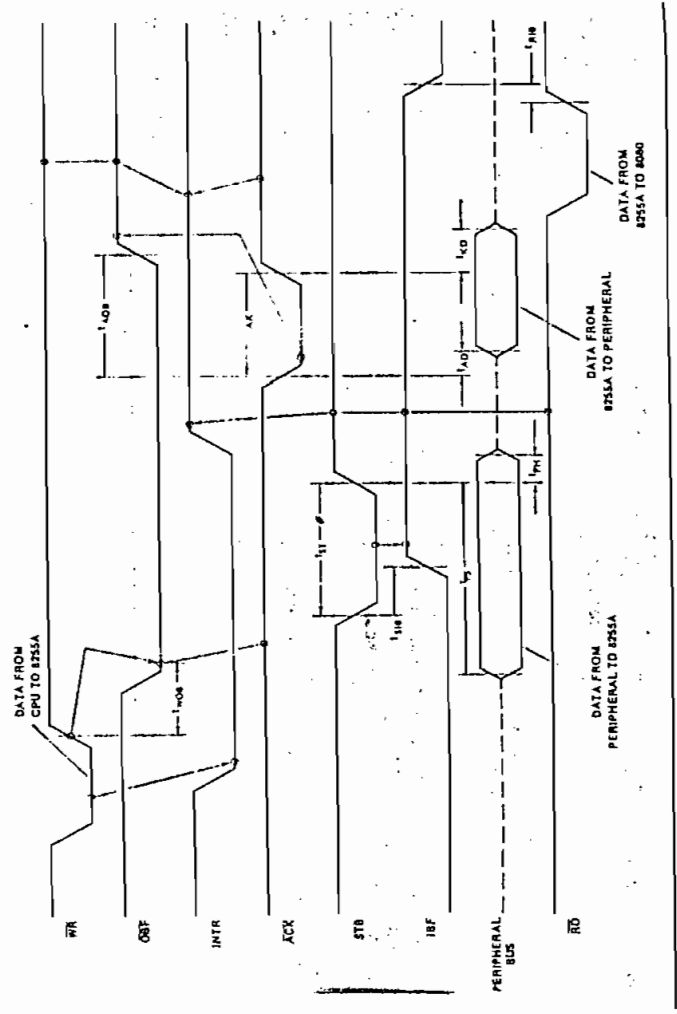
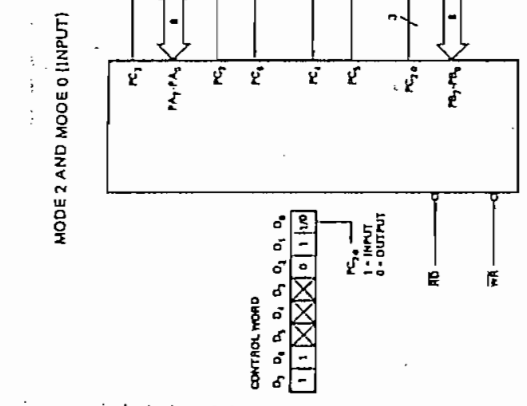
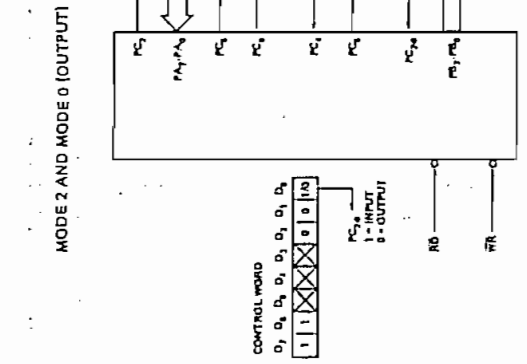
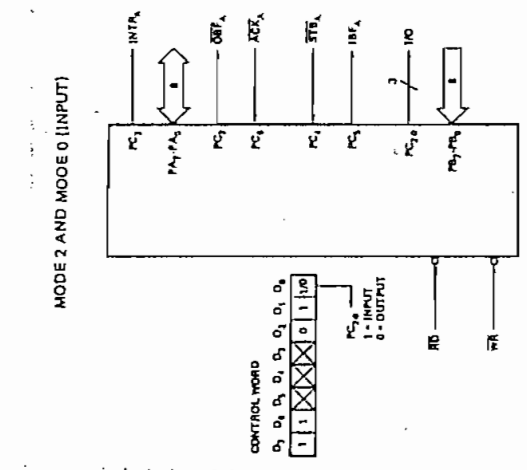


Figure 13. MODE 2 (Bidirectional)

NOTE: Any sequence where WR occurs before ACK and STB occurs before RD is permissible.
 (INTR = IBF · MASK · STB · RD + OBF · MASK · ACK · WR)

Figure 14. MODE 2 Combinations

Mode Definition Summary

	MODE 0		MODE 1		MODE 2	
	IN	OUT	IN	OUT	GROUP A ONLY	GROUP B ONLY
PA0	IN	OUT	IN	OUT	←	←
PA1	IN	OUT	IN	OUT	←	←
PA2	IN	OUT	IN	OUT	←	←
PA3	IN	OUT	IN	OUT	←	←
PA4	IN	OUT	IN	OUT	←	←
PA5	IN	OUT	IN	OUT	←	←
PA6	IN	OUT	IN	OUT	←	←
PA7	IN	OUT	IN	OUT	←	←
PB0	IN	OUT	IN	OUT	←	←
PB1	IN	OUT	IN	OUT	←	←
PB2	IN	OUT	IN	OUT	←	←
PB3	IN	OUT	IN	OUT	←	←
PB4	IN	OUT	IN	OUT	←	←
PB5	IN	OUT	IN	OUT	←	←
PB6	IN	OUT	IN	OUT	←	←
PB7	IN	OUT	IN	OUT	←	←
PC0	IN	OUT	INTRB	INTRB	I/O	I/O
PC1	IN	OUT	IBFB	OBFB	I/O	I/O
PC2	IN	OUT	STB	ACK-B	I/O	I/O
PC3	IN	OUT	INTRA	INTRA	INTRA	INTRA
PC4	IN	OUT	STB	I/O	STB	STB
PC5	IN	OUT	IBFA	I/O	ACK-A	ACK-A
PC6	IN	OUT	I/O	I/O	OBFA	OBFA
PC7	IN	OUT	I/O	I/O		

MODE 0 OR MODE 1 ONLY

Special Mode Combination Considerations

There are several combinations of modes when not all of the bits in Port C are used for control or status. The remaining bits can be used as follows:

If Programmed as Inputs — All input lines can be accessed during a normal Port C read.

If Programmed as Outputs — Bits in C upper (PC₇-PC₄) must be individually accessed using the bit set/reset function.

Bits in C lower (PC₃-PC₀) can be accessed using the bit set/reset function or accessed as a three-state source current into Port C.

Source Current Capability on Port B and Port C

Any set of eight output buffers, selected randomly from Ports B and C can source 1mA at 1.5 volts. This feature allows the 8255 to directly drive Darlington type drivers and high-voltage displays that require such source current.

Reading Port C Status

In Mode 0, Port C transfers data to or from the peripheral device. When the 8255 is programmed to function in Modes 1 or 2, Port C generates or accepts "hand-shaking" signals with the peripheral device. Reading the contents of Port C

APPLICATIONS OF THE 8255A

The 8255A is a very powerful tool for interfacing peripheral equipment to the microcomputer system. It represents the optimum use of available pins and is flexible enough to interface almost any I/O device without the need for additional external logic.

Each peripheral device in a microcomputer system usually has a "service routine" associated with it. The routine manages the software interface between the device and the CPU. The functional definition of the 8255A is programmed by the I/O service routine and becomes an extension of the system software. By examining the I/O devices interface characteristics for both data transfer and timing, and matching this information to the examples and tables in the detailed operational description, a control word can easily be developed to initialize the 8255A to exactly "fit" the application. Figures 17 through 23 present a few examples of typical applications of the 8255A.

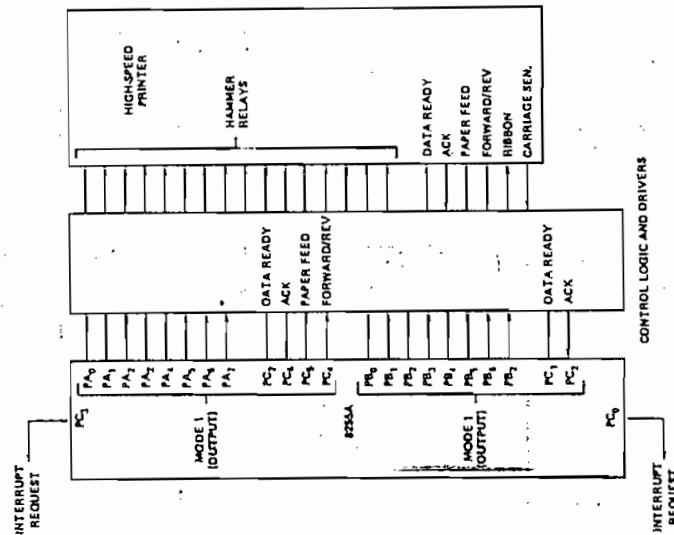


Figure 17. Printer Interface

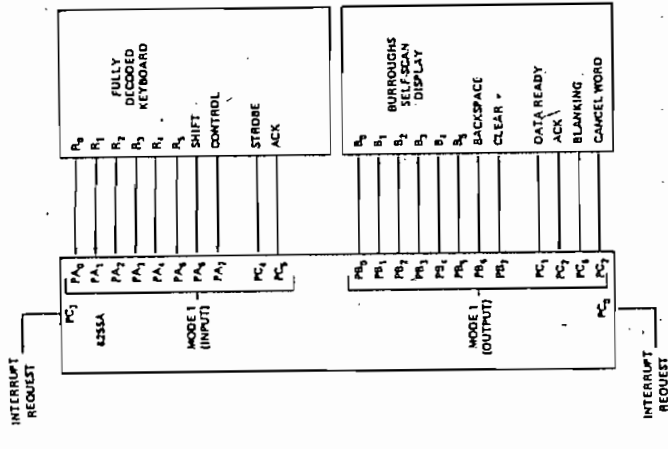


Figure 18. Keyboard and Display Interface

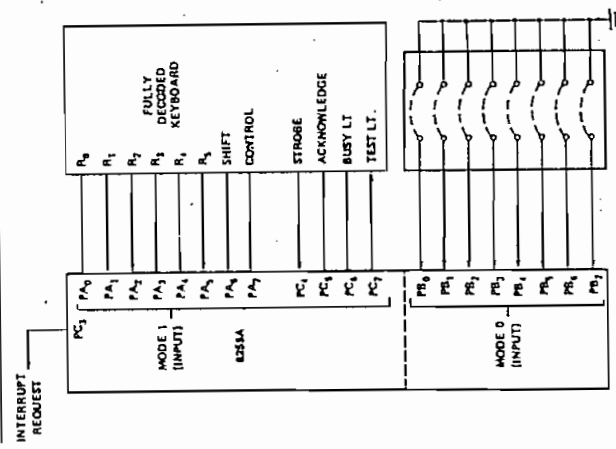


Figure 19. Keyboard and Terminal Address Interface

allows the programmer to test or verify the "status" of each peripheral device and change the program flow accordingly. There is no special instruction to read the status information from Port C. A normal read operation of Port C is executed to perform this function.

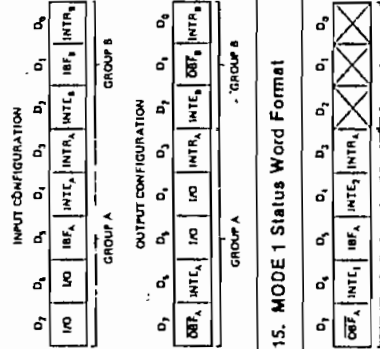
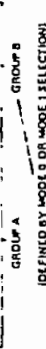


Figure 15. MODE 1 Status Word Format

Figure 16. MODE 2 Status Word Format



AC0830/DAC0831/DAC0832 8-Bit μ P Compatible, Double-Buffered D to A Converters

General Description

The DAC0830 is an advanced CMOS/Si-Cr 8-bit multiplying DAC designed to interface directly with the 8080, 8048, 85, Z80, and other popular microprocessors. A deposit-silicon-chromium R-2R resistor ladder network provides a reference current and provides the circuit with excellent temperature tracking characteristics (0.05% of Full Scale temperature maximum linearity error over temperature). The circuit uses CMOS current switches and control logic to achieve low power consumption and low output leakage current errors. Special circuitry provides TTL logic input voltage level compatibility.

Double buffering allows these DACs to output a voltage corresponding to one digital word while holding the next digital word. This permits the simultaneous updating of any number of DACs.

The DAC0830 series are the 8-bit members of a family of processor-compatible DACs (MICRO-DAC™). For applications demanding higher resolution, the DAC1000 series and the DAC1208 and DAC1230 (12-bits) are available alternatives.

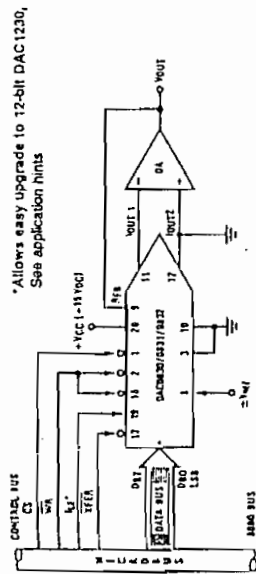
Features

- Double-buffered, single-buffered or flow-through digital data inputs
- Easy interchange and pin-compatible with 12-bit DAC1230 series
- Direct interface to all popular microprocessors
- Linearity specified with zero and full scale adjust only—NOT BEST STRAIGHT LINE FIT.
- Works with $\pm 10V$ reference-full 4-quadrant multiplication
- Can be used in the voltage switching mode
- Logic inputs which meet TTL voltage level specs (1.4V logic threshold)
- Operates "STAND ALONE" (without μ P) if desired
- Available in 20-pin small-outline or molded chip carrier package

Key Specifications

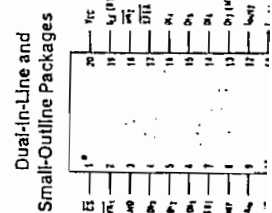
- Resolution: 1 μ s, 8 bits, 8, 9, or 10 bits
- Linearity: 0.0002% FS/°C (guaranteed over temp.)
- Gain Tempco: 20 mW
- Low power dissipation: 5 to 15 VDC
- Single power supply

Typical Application

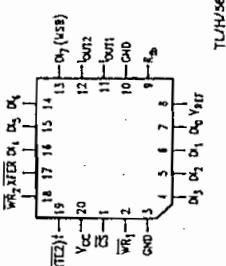


*Allows easy upgrade to 12-bit DAC1230. See application hints

Connection Diagrams (Top Views)



Molded Chip Carrier Package



This is necessary for the 12-bit DAC1230 series to permit interchanging from an 8-bit to a 12-bit DAC with NO PC board changes and no software changes. See applications section.

Absolute Maximum Ratings (Notes 1 & 2)

- Lead Temperature (soldering, 10 sec): 260°C
- Dual-In-Line Package (plastic): 300°C
- Dual-In-Line Package (ceramic): 300°C
- Surface Mount Package: 215°C
- Vapor Phase (60 sec): 220°C
- Infrared (15 sec): 220°C
- Operating Voltage (V_{CC}): 17 VDC
- Voltage at Any Digital Input: V_{CC} to GND
- Voltage at V_{REF} Input: $\pm 25V$
- Storage Temperature Range: -65°C to +150°C
- Package Dissipation: 500 mW
- At T_A = 25°C (Note 3)
- V_{CC} Voltage Applied to V_{OUT1} or I/O_{UT2} (Note 4): -100 mV to V_{CC}
- ESD Susceptibility (Note 14): 800V

Operating Conditions

- Temperature Range: T_{MIN} \leq T_A \leq T_{MAX}
- Part numbers with 'LCN' suffix: 0°C to +70°C
- Part numbers with 'LCWM' suffix: 0°C to +70°C
- Part numbers with 'LCV' suffix: 0°C to +70°C
- Part numbers with 'LJ' suffix: -40°C to +85°C
- Part numbers with 'LJ' suffix: -55°C to +125°C
- Voltage at Any Digital Input: V_{CC} to GND

Electrical Characteristics V_{REF} = 10,000 V_{OC} unless otherwise noted. Boldface limits apply over temperature. T_{MIN} \leq T_A \leq T_{MAX}. For all other limits T_A = 25°C.

Parameter	Conditions	See Note	V _{CC} = 4.75 VDC V _{CC} = 15.75 VDC		Limit Units
			Typ (Note 12)	Tested Limit (Note 5)	
Resolution			8	8	8 bits
Linearity Error Max	Zero and full scale adjusted -10V \leq V _{REF} \leq +10V	4, 8	0.05	0.05	% FSR
IN0830LJ & LCJ			0.2	0.2	% FSR
IN0832LJ & LCJ			0.05	0.05	% FSR
IN0830LCN, LCWM & LCV			0.1	0.1	% FSR
IN0831LCN			0.2	0.2	% FSR
IN0832LCN, LCWM & LCV			0.1	0.1	% FSR
Differential Nonlinearity Max	Zero and full scale adjusted -10V \leq V _{REF} \leq +10V	4, 8	0.1	0.1	% FSR
IN0830LJ & LCJ			0.4	0.4	% FSR
IN0832LJ & LCJ			0.1	0.1	% FSR
IN0830LCN, LCWM & LCV			0.2	0.2	% FSR
IN0831LCN			0.4	0.4	% FSR
Monotonicity	-10V \leq V _{REF} \leq +10V LJ & LCJ LCN, LCWM & LCV	4	8	8	bits
Gain Error Max	Using Internal R _{FB} -10V \leq V _{REF} \leq +10V	7	± 0.2	± 1	% FS
Gain Error Tempco Max	Using Internal R _{FB}		0.0002	0.0006	% FS/°C
Power Supply Rejection	All digital inputs latched high V _{CC} = 14.5V to 15.5V 11.5V to 12.5V 4.5V to 5.5V		0.0002 0.0006 0.013	0.0025 0.015	% FSR/V
Reference Input	Max		15	20	k Ω
	Min		15	10	k Ω
Input Feedthrough Error	V _{REF} = 20 Vp-p, f = 100 kHz All data inputs latched low		3		mVp-p

Electrical Characteristics $V_{REF} = 10,000$ VDC unless otherwise noted. Boldface limits apply over temperature, $T_{MIN} \leq T_A \leq T_{MAX}$. For all other limits $T_A = 25^\circ\text{C}$. (Continued)

Parameter	Conditions	See Note	$V_{CC} = 4.75$ VDC $V_{CC} = 15.75$ VDC		$V_{CC} = 5$ VDC $\pm 5\%$ $V_{CC} = 12$ VDC $\pm 5\%$ to 15 VDC $\pm 5\%$		Limit Units
			Typ (Note 12)	Tested Limit (Note 5)	Design Limit (Note 6)	Design Limit (Note 6)	
Output Leakage Current Max	All data inputs latched low	10		100	100		
				50	100		
Output Leakage Current Min	All data inputs latched high			100	100		
				50	100		
Input Capacitance	All data inputs latched low		45				
			115				
Input Capacitance	All data inputs latched high		130				
			30				

DIGITAL AND DC CHARACTERISTICS

Parameter	Conditions	Typ	Tested Limit	Design Limit
Digital Input Voltages	Logic Low	LJ	0.6	
		LJ	0.8	
		LCJ	0.7	
		LCJ	0.8	0.8
		LCN, LCWM, LCV	0.95	
Digital Input Currents	Logic High	LJ & LCJ	2.0	2.0
		LCN, LCWM, LCV	1.9	2.0
Digital Input Currents	Digital inputs < 0.8V	LJ & LCJ	-50	-200
		LCN, LCWM, LCV	-160	-200
Supply Current Drain	Digital inputs > 2.0V	LJ & LCJ	0.1	± 10
		LCN, LCWM, LCV	± 8	± 10
Supply Current Drain	Logic High	LJ & LCJ	1.2	3.5
		LCN, LCWM, LCV	1.7	2.0

Electrical Characteristics $V_{REF} = 10,000$ VDC unless otherwise noted. Boldface limits apply over temperature, $T_{MIN} \leq T_A \leq T_{MAX}$. For all other limits $T_A = 25^\circ\text{C}$. (Continued)

Symbol	Parameter	Conditions	See Note	$V_{CC} = 15.75$ VDC		$V_{CC} = 12$ VDC $\pm 5\%$ to 15 VDC $\pm 5\%$		$V_{CC} = 4.75$ VDC		$V_{CC} = 5$ VDC $\pm 5\%$	
				Typ (Note 12)	Tested Limit (Note 5)	Design Limit (Note 6)	Design Limit (Note 6)	Typ (Note 12)	Tested Limit (Note 5)	Design Limit (Note 6)	Design Limit (Note 6)
	Current Setting Time	$V_{IL} = 0V, V_{IH} = 5V$		1.0				1.0			
	Write and XFER Pulse Width Min	$V_{IL} = 0V, V_{IH} = 5V$	11	100	250	320	320	375	600	900	900
	Data Setup Time Min	$V_{IL} = 0V, V_{IH} = 5V$	9	100	250	320	320	375	600	900	900
	Data Hold Time Min	$V_{IL} = 0V, V_{IH} = 5V$	9	30	30	30	30	50	50	50	50
	Control Setup Time Min	$V_{IL} = 0V, V_{IH} = 5V$	9	110	250	320	320	600	900	1100	1100
	Control Hold Time Min	$V_{IL} = 0V, V_{IH} = 5V$	9	0	0	10	10	0	0	0	0

IC CHARACTERISTICS

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.

Note 2: All voltages are measured with respect to GND, unless otherwise specified.

Note 3: The maximum power dissipation must be derated at elevated temperatures and is dictated by T_{JMAX} , θ_{JA} , and the ambient temperature, T_A . The maximum allowable power dissipation at any temperature is $P_D = (T_{JMAX} - T_J)/\theta_{JA}$, or the number given in the Absolute Maximum Ratings, whichever is lower. For this device, $T_{JMAX} = 125^\circ\text{C}$ (plastic) or 150°C (ceramic), and the typical junction-to-ambient thermal resistance of the J package this number is 120°C/W . For the L package, this number increases to 100°C/W and for the V package this number is 120°C/W .

Note 4: For current switching applications, both I_{OUT1} and I_{OUT2} must go to ground or the "Virtual Ground" of an operational amplifier. The linearity error is compensated by approximately $V_{OS} + V_{REF}$. For example, if $V_{REF} = 10V$ then a 1 mV offset, V_{OS} , on I_{OUT1} or I_{OUT2} will introduce an additional 0.01% linearity error.

Note 5: Tested limits are guaranteed to National's AOCL (Average Outgoing Quality Level).

Note 6: Guaranteed, but not 100% production tested. These limits are not used to calculate outgoing quality levels.

Note 7: Guaranteed at $V_{REF} = \pm 10$ VDC and $V_{REF} = \pm 1$ VDC.

Note 8: The unit "FSR" stands for "Full Scale Range." "Linearity Error" and "Power Supply Rejection" specs are based on this unit to eliminate dependence on a particular V_{REF} value and to indicate the true performance of the part. The "Linearity Error" specification of the DAC0830 is "0.05% of FSR (MAX)". This guarantees that after performing a zero and full scale adjustment (see Sections 2.5 and 2.6), the plot of the 256 analog voltage outputs will each be within $\pm 0.05\%$ of V_{REF} of a straight line which passes through zero and full scale.

Note 9: Boldface tested limits apply to the LJ and LCJ suffix parts only.

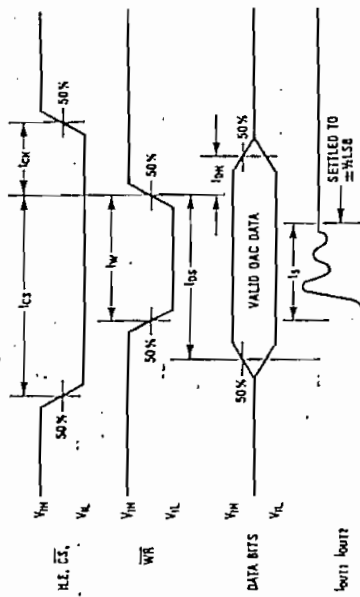
Note 10: A 100nA leakage current with $I_{FB} = 20k$ and $V_{REF} = 10V$ corresponds to a zero error of $(100 \times 10^{-9} \times 20 \times 10^3) \times 100/10$ which is 0.02% of FS.

Note 11: The entire write pulse must occur within the valid data interval for the specified I_{FB} , I_{OS} , I_{OH} , and I_S to apply.

Note 12: Typicals are at 25°C and represent most likely parametric norm.

Note 13: Human body model, 100 pF discharged through a 1.5 k Ω resistor.

Switching Waveform



TJ/H/5606-2

Definition of Package Pinouts

Control Signals (All control signals level actuated)
CS: Chip Select (active low). The \overline{CS} in combination with ILE will enable \overline{WR}_1 .

ILE: Input Latch Enable (active high). The ILE in combination with \overline{CS} enables \overline{WR}_1 .

\overline{WR}_1 : Write 1. The active low \overline{WR}_1 is used to load the digital input data bits (DI) into the input latch. The data in the input latch is latched when \overline{WR}_1 is high. To update the input latch— \overline{CS} and \overline{WR}_1 must be low while ILE is high.

\overline{WR}_2 : Write 2 (active low). This signal, in combination with \overline{XFER} , causes the 8-bit data which is available in the input latch to transfer to the DAC register.

\overline{XFER} : Transfer control signal (active low). The \overline{XFER} will enable \overline{WR}_2 .

Other Pin Functions

DI₀-DI₇: Digital Inputs. DI₀ is the least significant bit (LSB) and DI₇ is the most significant bit (MSB). DAC Current Output 1, IOUT1 is a maximum for a digital code of all 1's in the DAC register, and is zero for all 0's in DAC register.

IOUT₂: DAC Current Output 2. IOUT₂ is a constant (I full minus IOUT₁), or IOUT₁ + IOUT₂ = constant (I full scale for a fixed reference voltage).

RIP: Feedback Resistor. The feedback resistor is provided on the IC chip for use as the shunt

feedback resistor for the external op amp which is used to provide an output voltage for the DAC. This on-chip resistor should always be used (not an external resistor) since it matches the resistors which are used in the on-chip R-2R ladder and tracks these resistors over temperature.

Reference Voltage Input. This input connects an external precision voltage source to the internal R-2R ladder. V_{REF} can be selected over the range of +10 to -10V. This is also the analog voltage input for a 4-quadrant multiplying DAC application.

Digital Supply Voltage. This is the power supply pin for the part. V_{CC} can be from +5 to +15VDC. Operation is optimum for +15VDC.

The pin 10 voltage must be at the same ground potential as IOUT₁ and IOUT₂ for current switching applications. Any difference of potential (VOS pin 10) will result in a linearity change of

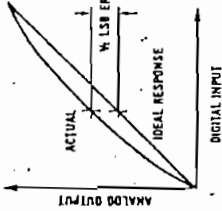
$$V_{OS} \text{ pin } 10$$

$$3V_{REF}$$

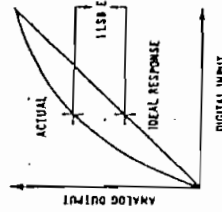
For example, if $V_{REF} = 10V$ and pin 10 is 9mV offset from IOUT₁ and IOUT₂ the linearity change will be 0.03%.

Pin 3 can be offset $\pm 100mV$ with no linearity change, but the logic input threshold will shift.

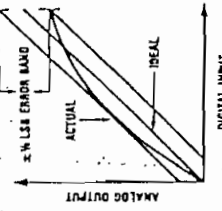
Linearity Error



a) End point test after zero and its adj.



b) Best straight line



c) Shifting is adj. to pass best straight line test

Definition of Terms

Resolution: Resolution is directly related to the number of switches or bits within the DAC. For example, the DAC0830 has 28 or 256 steps and therefore has 8-bit resolution.

Linearity Error: Linearity Error is the maximum deviation from a straight line passing through the endpoints of the DAC transfer characteristic. It is measured after adjusting for zero and full-scale. Linearity error is a parameter intrinsic to the device and cannot be externally adjusted.

National's linearity "end point test" (a) and the "best straight line" test (b,c) used by other suppliers are illustrated above. The "end point test" greatly simplifies the adjustment procedure by eliminating the need for multiple iterations of checking the linearity and then adjusting full scale until the linearity is met. The "end point test" guarantees that linearity is met after a single full scale adjust (One adjustment vs. multiple iterations of the adjustment.) The "end point test" uses a standard zero and F.S. adjustment procedure and is a much more stringent test for DAC linearity.

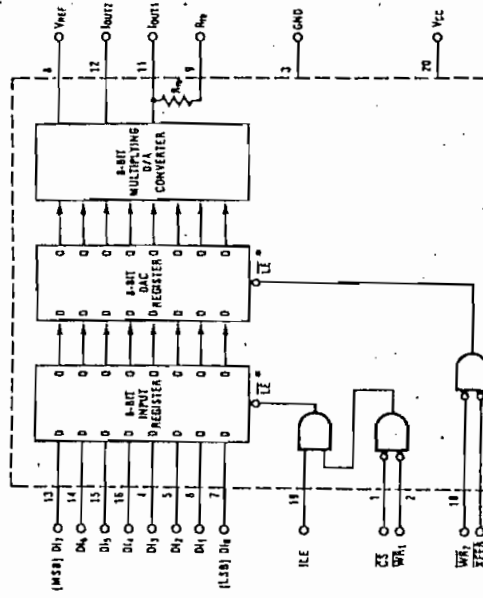
Power Supply Sensitivity: Power supply sensitivity is a measure of the effect of power supply changes on the DAC full-scale output.

Settling Time: Settling time is the time required from a code transition until the DAC output reaches within $\pm 1/2$ LSB of the final output value. Full-scale settling time requires a zero to full-scale or full-scale to zero output change.

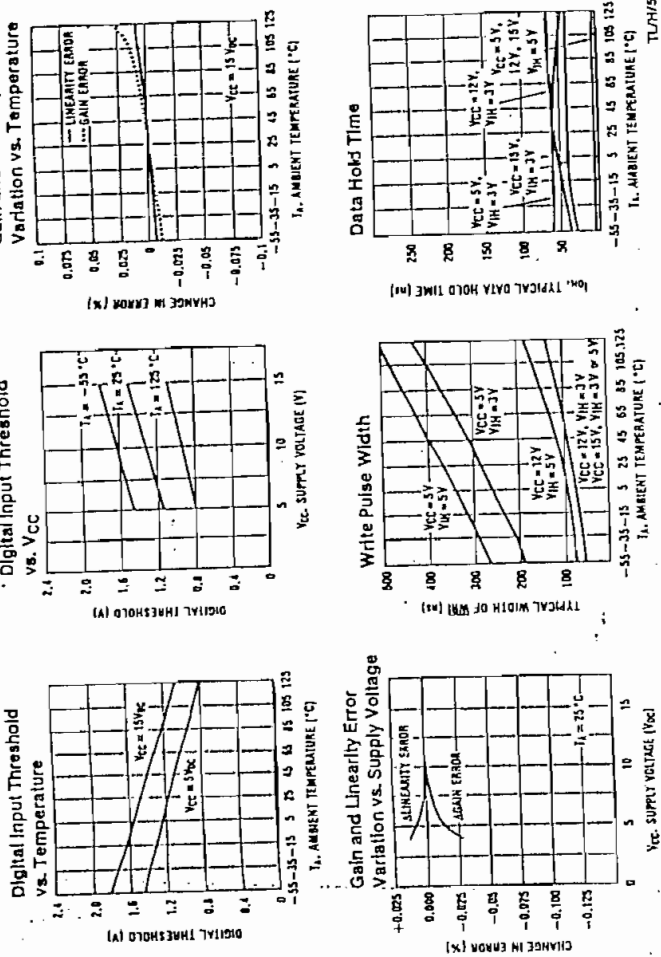
Full-Scale Error: Full scale error is a measure of the output error between an ideal DAC and the actual device output. Ideally, for the DAC0830 series, full-scale is $V_{REF} - 1LSB$. For $V_{REF} = 10V$ and unipolar operation, $V_{FULL-SCALE} = 10.00000V - 99mV = 9.961V$. Full-scale error is adjustable to zero.

Differential Nonlinearity: The difference between any two consecutive codes in the transfer curve from the theoretical 1 LSB is differential nonlinearity.

Monotonic: If the output of a DAC increases for increasing digital input code, then the DAC is monotonic. An 8-bit DAC which is monotonic to 8 bits simply means that increasing digital input codes will produce an increasing analog output.



Typical Performance Characteristics



TLU/H/5608-5

DAC0830 Series Application Hints

These DAC's are the industry's first microprocessor compatible, double-buffered 8-bit multiplying D to A converters. Double-buffering allows the utmost application flexibility from a digital control point of view. This 20-pin device is also pin for pin compatible (with one exception) with the DAC1230, a 12-bit MICRO-DAC. In the event that a system's analog output resolution and accuracy must be upgraded, substituting the DAC1230 can be easily accomplished. By tying address bit A_0 to the ILE pin, a two-byte μP write instruction (double precision) which automatically increments the address for the second byte write (starting with $A_0 = "1"$) can be used. This allows either an 8-bit or the 12-bit part to be used with no hardware or software changes. For the simplest 8-bit application, this pin should be tied to V_{CC} (also see other uses in section 1.1).

Analog signal control versatility is provided by a precision R-2R ladder network which allows full 4-quadrant multiplication of a wide range bipolar reference voltage by an applied digital word.

1.0 DIGITAL CONSIDERATIONS

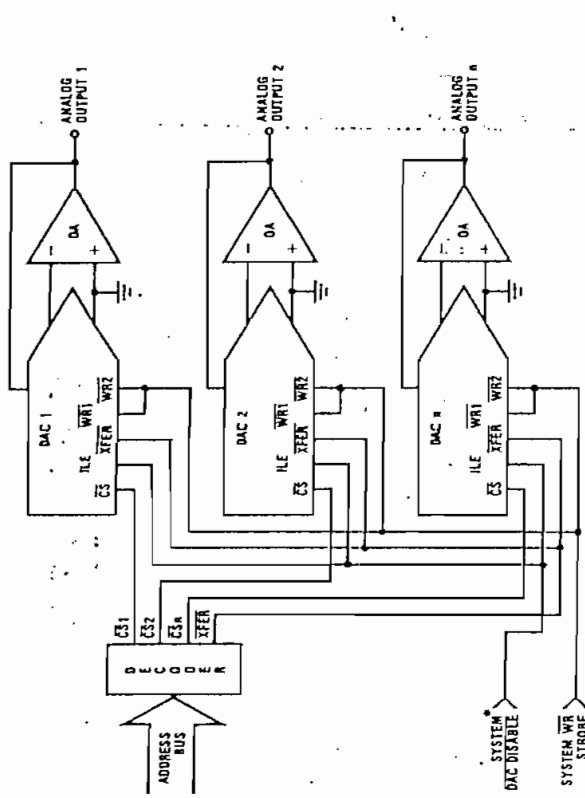
A most unique characteristic of these DAC's is that the 8-bit digital input byte is double-buffered. This means that the data must transfer through two independently controlled 8-bit latching registers before being applied to the R-2R ladder network to change the analog output. The addition of a second register allows two useful control features. First, any DAC in a system can simultaneously hold the current DAC data in a register (DAC register) and the next data word in the second register (input register) to allow fast updating of the DAC output on demand. Second, and probably more important, double-buffering allows any number of DAC's in a

system to be updated to their new analog output levels simultaneously via a common strobe signal. The timing requirements and logic level convention of the register control signals have been designed to minimize or eliminate external interfacing logic when applied to most popular microprocessors and development systems. It is easy to think of these converters as 8-bit "write-only" memory locations that provide an analog output quantity. All inputs to these DAC's meet TTL voltage level specs and can also be driven directly with high voltage CMOS logic in non-microprocessor based systems. To prevent damage to the chip from static discharge, all unused digital inputs are tied to V_{CC} or ground. If any of the digital inputs are inadvertently left floating, the DAC interprets the pin as a logic "1".

1.1 Double-Buffered Operation

Updating the analog output of these DAC's in a double-buffered manner is basically a two step or double write operation. In a microprocessor system two unique system addresses must be decoded, one for the input latch controlled by the \overline{CS} pin and a second for the DAC latch which is controlled by the \overline{XFER} line. If more than one DAC is being driven, Figure 2, the \overline{CS} line of each DAC would typically be decoded individually, but all of the converters could share a common \overline{XFER} address to allow simultaneous updating of any number of DAC's. The timing for this operation is shown, Figure 3.

It is important to note that the analog outputs that change after a simultaneous transfer are those from the DAC's whose input register had been modified prior to the \overline{XFER} command.



*TIE TO LOGIC 1 IF NOT NEEDED (SEE SEC. 1.1).

FIGURE 2. Controlling Multiple DACs

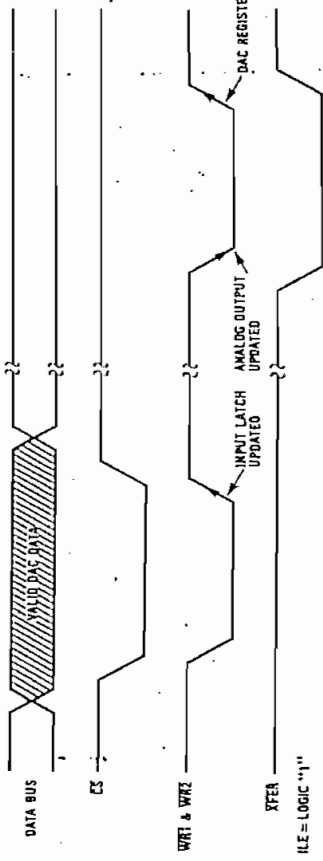


FIGURE 3

The ILE pin is an active high chip select which can be decoded from the address bus as a qualifier for the normal \overline{CS} signal generated during a write operation. This can be used to provide a higher degree of decoding unique control signals for a particular DAC, and thereby create a more efficient addressing scheme.

Another useful application of the ILE pin of each DAC in a multiple DAC system is to tie these inputs together and use this as a control line that can effectively "freeze" the outputs of all the DAC's at their present value. Pulling this line low latches the input register and prevents new data from being written to the DAC. This can be particularly useful in multiprocessing systems to allow a processor other than the

one controlling the DAC's to take over control of the data bus and control lines. If this second system were to use the same addresses as those decoded for DAC control (but for a different purpose) the ILE function would prevent the DAC's from being erroneously altered. In a "Stand-Alone" system the control signals are generated by discrete logic. In this case double-buffering can be controlled by simply taking \overline{CS} and \overline{XFER} to a logic "0", to a logic "1", and pulling $\overline{WR1}$ low to load data to the output latch. Pulling $\overline{WR2}$ low will then update the analog output. A logic "1" on either of these lines will prevent the changing of the analog output.

TLU/H/5608-4



ADC0808, ADC0809 8-Bit μ P Compatible A/D Converters With 8-Channel Multiplexer

Features

- Resolution — 8-bits
- Total unadjusted error — $\pm 1/2$ LSB and ± 1 LSB
- No missing codes
- Conversion time — 100 μ s
- Single supply — 5 V_{DC}
- Operates ratiometrically or with 5 V_{DC} or analog span adjusted voltage reference
- 8-channel multiplexer with latched control logic
- Easy interface to all microprocessors, or operates "stand alone"
- Outputs meet T_L voltage level specifications
- 0V to 5V analog input voltage range with single 5V supply
- No zero or full-scale adjust required
- Standard hermetic or molded 28-pin DIP package
- Temperature range — 40°C to +85°C or -55°C to +125°C
- Low power consumption — 15 mW
- Latched TRI-STATE[®] output

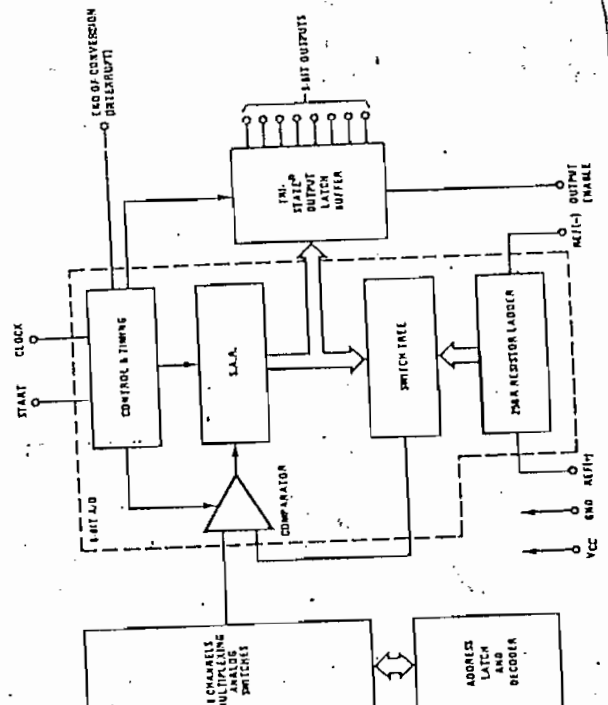
General Description

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

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

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

Block Diagram



Typical Applications (Continued)

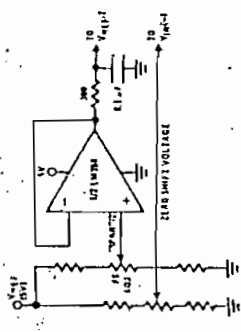


FIGURE 19. Offsetting the Zero of the ADC0801 and Performing an Input Range (Span) Adjustment

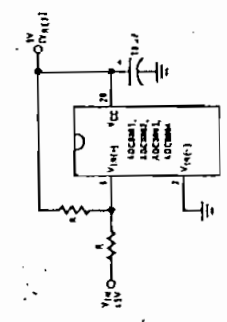


FIGURE 20. Handling ± 5 V Analog Input Range

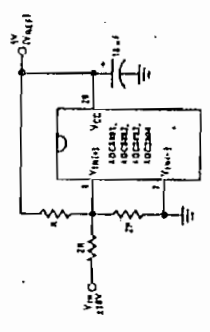


FIGURE 21. Handling ± 10 V Analog Input Range

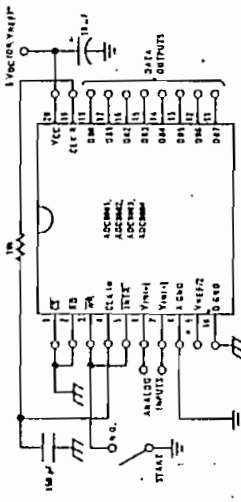


FIGURE 22. Free Running Connection

Ordering Information

TEMPERATURE RANGE	-40°C TO +85°C	-55°C TO +125°C
• 1 1/2 Bit Adjusted	ADC0801LCD	ADC0801LD
• 1 1/2 Bit Unadjusted	ADC0802LCD	ADC0802LD
• 1 1/2 Bit Adjusted	ADC0803LCD	ADC0803LD
• 1 1/2 Bit Unadjusted	ADC0804LCD	ADC0804LD
PACKAGE OUTLINE	N28A—MOLDED DIP	D208A—CAVITY DIP

Absolute Maximum Ratings (Notes 1 and 2)

Supply Voltage (V _{CC}) (Note 3)	5.5V
Voltage at Any Pin	-0.5V to (V _{CC} + 0.5V)
Excess Control Inputs	-0.5V to +1.5V
Voltage at Control Inputs (START, OE, CLOCK, ALE, ADD A, ADD B, ADD C)	-0.5V to +1.5V
Storage Temperature Range	-65°C to +150°C
Package Dissipation at T _A = 25°C	875 mW
Lead Temperature (Soldering, 10 seconds)	300°C

Operating Ratings (Notes 1 and 2)

Temperature Range (Note 1)	T _{MIN} ≤ T _A ≤ T _{MAX} -55°C ≤ T _A ≤ +125°C
ADC0808CCJ, ADC0809CCN	-40°C ≤ T _A ≤ +85°C
Range of V _{CC} (Note 1)	4.5V to 5.5V

Electrical Characteristics

Converter Specifications: V_{CC} = 5V, V_{REF(+) = V_{REF(-)} = GND, T_{MIN} ≤ T_A ≤ T_{MAX} and f_{CLK} = 640 kHz unless otherwise stated.}

Parameter	Conditions	Min	Typ	Max	Units
ADC0808 Total Unadjusted Error (Note 5)	25°C T _{MIN} to T _{MAX}		± 1/2		LSB
ADC0809 Total Unadjusted Error (Note 5)	0°C to 70°C T _{MIN} to T _{MAX}		± 3/4		LSB
Input Resistance	From Ref(+) to Ref(-)	1.0	2.5		kΩ
Analog Input Voltage Range	(Note 4) V(+) or V(-)	GND-0.10	V _{CC}	V _{CC} +0.10	V
Voltage, Top of Ladder	Measured at Ref(+)	V _{CC} /2-0.1	V _{CC}	V _{CC} +0.1	V
Voltage, Center of Ladder				V _{CC} /2±0.1	V
Voltage, Bottom of Ladder	Measured at Ref(-)	-0.1	0		V
Comparator Input Current	f _c = 640 kHz (Note 6)	-2	± 0.5	2	μA

Absolute Maximum Ratings (Notes 1 and 2)

Supply Voltage (V _{CC}) (Note 3)	5.5V
Voltage at Any Pin	-0.5V to (V _{CC} + 0.5V)
Excess Control Inputs	-0.5V to +1.5V
Voltage at Control Inputs (START, OE, CLOCK, ALE, ADD A, ADD B, ADD C)	-0.5V to +1.5V
Storage Temperature Range	-65°C to +150°C
Package Dissipation at T _A = 25°C	875 mW
Lead Temperature (Soldering, 10 seconds)	300°C

Operating Ratings (Notes 1 and 2)

Temperature Range (Note 1)	T _{MIN} ≤ T _A ≤ T _{MAX} -55°C ≤ T _A ≤ +125°C
ADC0808CCJ, ADC0809CCN	-40°C ≤ T _A ≤ +85°C
Range of V _{CC} (Note 1)	4.5V to 5.5V

Electrical Characteristics

Converter Specifications: V_{CC} = 5V, V_{REF(+) = V_{REF(-)} = GND, T_{MIN} ≤ T_A ≤ T_{MAX} and f_{CLK} = 640 kHz unless otherwise stated.}

Parameter	Conditions	Min	Typ	Max	Units
ADC0808 Total Unadjusted Error (Note 5)	25°C T _{MIN} to T _{MAX}		± 1/2		LSB
ADC0809 Total Unadjusted Error (Note 5)	0°C to 70°C T _{MIN} to T _{MAX}		± 3/4		LSB
Input Resistance	From Ref(+) to Ref(-)	1.0	2.5		kΩ
Analog Input Voltage Range	(Note 4) V(+) or V(-)	GND-0.10	V _{CC}	V _{CC} +0.10	V
Voltage, Top of Ladder	Measured at Ref(+)	V _{CC} /2-0.1	V _{CC}	V _{CC} +0.1	V
Voltage, Center of Ladder				V _{CC} /2±0.1	V
Voltage, Bottom of Ladder	Measured at Ref(-)	-0.1	0		V
Comparator Input Current	f _c = 640 kHz (Note 6)	-2	± 0.5	2	μA

Electrical Characteristics (Continued)

Digital Levels and DC Specifications: ADC0808CCJ 4.5V ≤ V_{CC} ≤ 5.5V, -55°C ≤ T_A ≤ +125°C unless otherwise noted. ADC0808CCJ, ADC0808CCN, and ADC0809CCN 4.75V ≤ V_{CC} ≤ 5.25V, -40°C ≤ T_A ≤ +85°C unless otherwise noted.

DATA OUTPUTS AND EOC (INTERRUPT)

Parameter	Conditions	Min	Typ	Max	Units
V _{OUT(1)}	Logical "1" Output Voltage	V _{CC} -0.4			V
V _{OUT(8)}	Logical "0" Output Voltage			0.45	V
V _{OUT(8)}	Logical "0" Output Voltage EOC			0.45	V
I _{OUT}	TRI-STATE Output Current			3	μA
I _{OUT}	TRI-STATE Output Current			3	μA

Electrical Characteristics

Timing Specifications: V_{CC} = V_{REF(+) = V_{REF(-)} = 5V, V_{REF(-)} = GND, I₁ = I₂ = 20 nA and T_A = 25°C unless otherwise noted.}

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t _{WS}	Minimum Start Pulse Width	(Figure 5)		100	200	ns
t _{WALE, 1}	Minimum ALE Pulse Width	(Figure 5)		100	200	ns
t ₃	Minimum Address Set-Up Time	(Figure 5)		25	50	ns
t ₄	Minimum Address Hold Time	(Figure 5)		25	50	ns
t ₀	Minimum Address Delay Time	(Figure 5)		1	2.5	μs
t _{H1} , t _{H0}	Analog MUX Delay Time From ALE	R _S = 0Ω (Figure 5)		125	250	ns
t _{H1} , t _{H0}	OE Control to 0 Logic State	C _L = 50 pF, R _L = 10k (Figure 8)		125	250	ns
t _{H1} , t _{H0}	OE Control to HIGH	C _L = 10 pF, R _L = 10k (Figure 8)		100	116	μs
t _c	Conversion Time	f ₀ = 640 kHz (Figure 5) (Note 7)	90		1280	kHz
t _c	Clock Frequency	(Figure 5)	10		8 + 2 μs	Clock Periods
t _{EOC}	EOC Delay Time	(Figure 5)	0		15	pF
C _{IN}	Input Capacitance	At Control Inputs		10	15	pF
C _{OUT}	TRI-STATE* Output Capacitance	At TRI-STATE* Outputs, (Note 12)		10	15	pF

Note 1: Absolute maximum ratings are those values beyond which the life of the device may be impaired.
 Note 2: All voltages are measured with respect to GND, unless otherwise specified.
 Note 3: A zero-volt diode exists, internally, from V_{CC} to GND and has a typical breakdown voltage of 7 V_{CC}.
 Note 4: Two on-chip diodes are tied to each analog input which will forward conduct for analog input voltages one diode drop below ground for one diode drop greater than the V_{CC} supply. The spec allows 100 mV forward bias of either diode. This means that as long as the analog V_{IN} does not exceed the supply voltage by more than 100 mV, the output code will be correct. To achieve an absolute 0 V_{CC} to 0.5 V_{CC} input voltage range will therefore require a minimum supply voltage of 4.900 V_{CC} over temperature variations, initial tolerances, and multiple errors.
 Note 5: Total unadjusted error includes offset, gain scale, linearity, and multiple errors. See Figure 3. None of these A/Ds requires a zero of full-scale adjustment. However, if an all-zero code is desired for an analog input other than 0.0V, or if a narrow full-scale span exists (for example, 0.5V to 4.5V full-scale) the reference voltage can be adjusted to achieve this. See Figure 11.
 Note 6: Comparator input current is a plus current into or out of the chopper stabilized comparator. The bias current varies directly with clock frequency and has little temperature dependence (Figure 6). See paragraph 4.0.
 Note 7: The outputs of the data register are updated one clock cycle before the rising edge of EOC.

Functional Description

Multiplexer. The device contains an 8-channel single-ended analog signal multiplexer. A particular input channel is selected by using the address decoder. Table 1 shows the input states for the address lines to select any channel. The address is latched into the decoder on the low-to-high transition of the address latch enable signal.

TABLE 1

SELECTED ANALOG CHANNEL	ADDRESS LINE		
	C	B	A
IN0	L	L	L
IN1	L	L	H
IN2	L	H	L
IN3	L	H	H
IN4	H	L	L
IN5	H	L	H
IN6	H	H	L
IN7	H	H	H

CONVERTER CHARACTERISTICS

The Converter

The heart of this single chip data acquisition system is its 8-bit analog-to-digital converter. The converter is designed

to give fast, accurate, and repeatable conversions over a wide range of temperatures. The converter is partitioned into 3 major sections: the 256R ladder network, the successive approximation register, and the comparator. The converter's digital outputs are positive true.

The 256R ladder network approach (Figure 1) was chosen over the conventional R2R ladder because of its inherent monotonicity, which guarantees no missing digital codes. Monotonicity is particularly important in closed loop feedback control systems. A non-monotonic relationship can cause oscillations that will be catastrophic for the system. Additionally, the 256R network does not cause load variations on the reference voltage.

The bottom resistor and the top resistor of the ladder network in Figure 1 are not the same value as the remainder of the network. The difference in these resistors causes the output characteristic to be symmetrical with the zero and full-scale points of the transfer curve. The first output transition occurs when the analog signal has reached $+1/2$ LSB and succeeding output transitions occur every 1 LSB later up to full-scale.

The successive approximation register (SAR) performs 8 iterations to approximate the input voltage. For any SAR type converter, n -iterations are required for an n -bit converter. Figure 2 shows a typical example of a 3-bit converter. In the ADC0808, ADC0809, the approximation technique is extended to 8 bits using the 256R network.

Functional Description (continued)

The A/D converter's successive approximation register (SAR) is reset on the positive edge of the start conversion (SC) pulse. The conversion is begun on the falling edge of the start conversion pulse. A conversion in process will be interrupted by receipt of a new start conversion pulse. Continuous conversion may be accomplished by tying the end-of-conversion (EOC) output to the SC input. If used in this mode, an external start conversion pulse should be applied after power up. End-of-conversion will go low between 0 and 8 clock pulses after the rising edge of start conversion.

The most important section of the A/D converter is the comparator. It is this section which is responsible for the ultimate accuracy of the entire converter. It is also the

comparator drift which has the greatest influence on the repeatability of the device. A chopper-stabilized comparator provides the most effective method of satisfying all the converter requirements.

The chopper-stabilized comparator converts the DC input signal into an AC signal. This signal is then fed through a high gain AC amplifier and has the DC level restored. This technique limits the drift component of the amplifier since the drift is a DC component which is not passed by the AC amplifier. This makes the entire A/D converter extremely insensitive to temperature, long term drift and input offset errors.

Figure 4 shows a typical error curve for the ADC0808 as measured using the procedures outlined in AN-179.

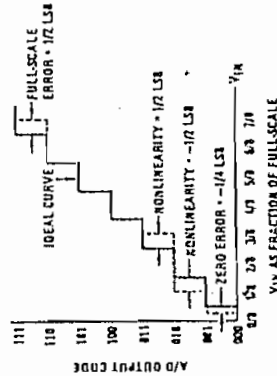


FIGURE 2. 3-Bit A/D Transfer Curve

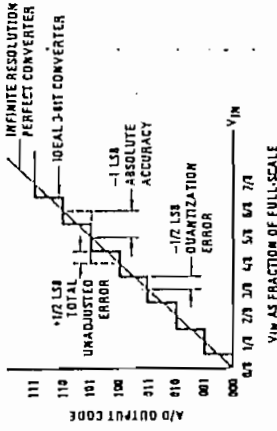


FIGURE 3. 3-Bit A/D Absolute Accuracy Curve

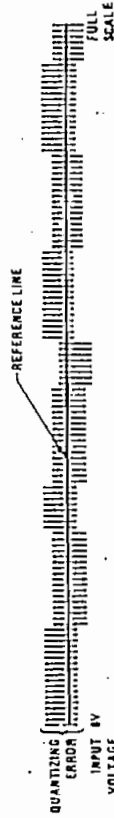


FIGURE 4. Typical Error Curve

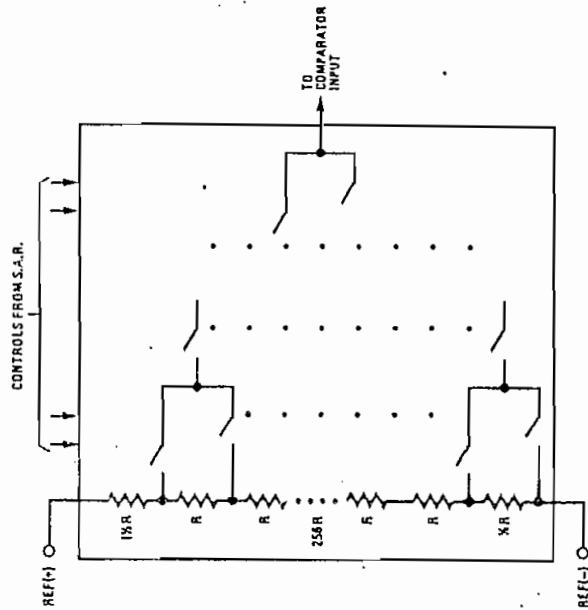
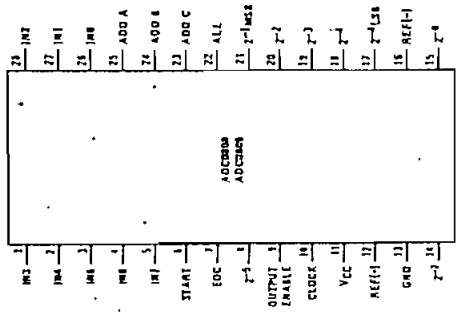


FIGURE 1. Resistor Ladder and Switch Tree

Connection Diagram

Dual-In-Line Package



TOP VIEW

Timing Diagram

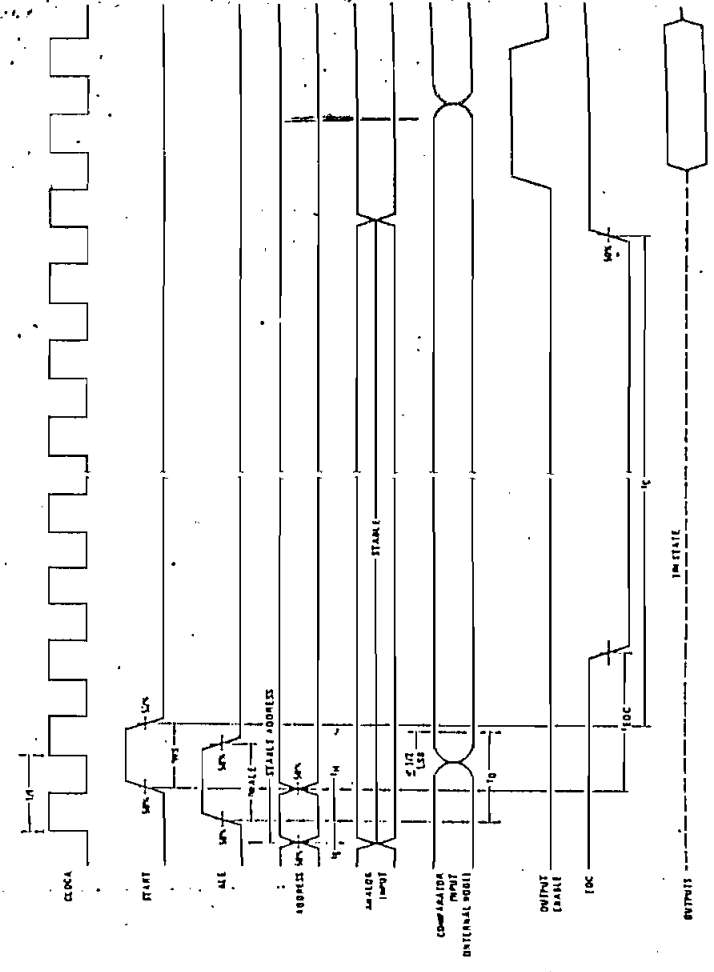


FIGURE 5

Typical Performance Characteristics

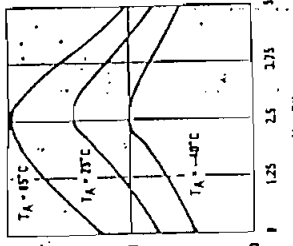
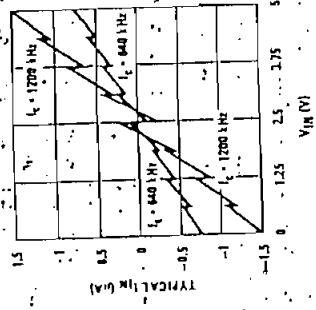


FIGURE 6. Comparator I_{iH} vs V_{iH} ($V_{CC} = V_{REF} = 5V$)

FIGURE 7. Multiplexer R_{ON} vs V_{iH} ($V_{CC} = V_{REF} = 5V$)

TRI-STATE Test Circuits and Timing Diagrams

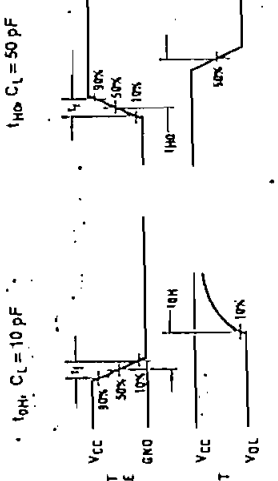
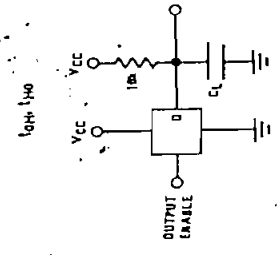
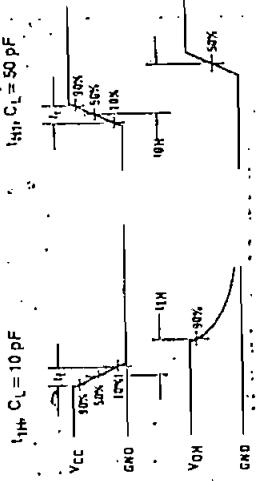
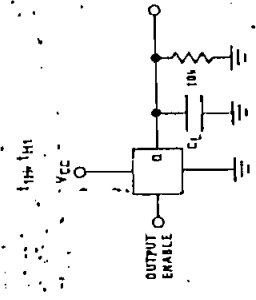


FIGURE 8