UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

"DESARROLLO E IMPLEMENTACION DE UN SISTEMA DE EVALUACION BASADO EN EL 6809"

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO MECANICO ELECTRICISTA

P R E S E N T A N:

ARMANDO GARCIA GONZALEZ
DAVID TREJO MARTINEZ
ERNESTO GAMIÑO CRUZ

Director de Tesis: M. en I. Juan Carlos Roa B.

MEXICO, D. F. 1993.

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

Indice

Introducción

El gran auge que la electrónica ha mostrado en los últimos 30 años, abre un sin número de horizontes quizá nunca antes imaginados. Hasta hace 15 años la idea de adquirir un computador personal con grandes capacidades de procesamiento, memoria, velocidad y además de bajo costo, era poco menos que imposible. Más difícil aún era el pensar que este computador personal, pudiera usarse en el control automático de procesos industriales y así facilitar esta pesada tarea.

Sin embargo, hoy en día todo esto es posible realizarlo.

En México, se cuanta actualmente con la tecnología, los conocimientos y personal necesario para su realización.

Hay en el mercado mexicano una gran demanda de automatizar sus plantas industriales, modernizar sus procesos y liberar de las

tareas rutinarias o peligrosas al personal humano.

También en forma paralela a la idea anterior, existen en las universidades de México, la necesidad de formar ingenieros, que sean capases de crear estos sistemas, que cuenten en las aulas con las herramientas y la tecnología que les permita enfrentar estos retos.

Para dar respuesta a estas dos necesidades existentes se creó el sistema de evaluación y control con el microprocesador 6809 (SEC6809).

Por un lado nos permite realizar programas en lenguaje ensamblador del microprocesador, correrlos y utilizar sus periféricos, así como visualizar sus instrucciones de entrada y salida. Deseamos ir más allá de los *displays* y pantallas de cristal líquido, y se plantea la alternativa de visualizar sus instrucciones en una pantalla de T.V., lo que se consiguió al diseñar e implementar junto con la tarjeta principal (MPU), una tarjeta controladora de video. También en la entrada de datos, no quisimos limitarnos a un teclado de membrana o hexadecimal y se opto por un teclado para PC's.

Con lo anterior conseguimos un sistema de evaluación del MPU 6809, de fácil manejo y ambientación amigable, que puede ser utilizado, en las escuelas de enseñanza superior en los cursos de introducción a los microprocesadores, en donde se desee presentar los de la familia de Motorola.

Una ventaja adicional a la selección de este MPU, consiste en que nos permite fácilmente ir más allá, y adentrarnos al estudio de los microcontroladores, que para el caso de la familia 68HCXX, guarda una gran similitud.

Por otro lado, y en respuesta a la necesidad planteada, se diseñaron los siguientes módulos adicionales: control de velocidad de motores de AC, control de velocidad de motores de DC y medición y control del factor de potencia.

En el Capítulo I se realiza un estudio del MPU 6809, haciendo énfasis en su arquitectura interna y sus nemónicos.

En el Capítulo II se detallan los componentes auxiliares (periféricos) los cuales son necesarios para la implementación y correcto funcionamiento del sistema.

En el Capítulo III se presentan los aspectos de construcción del *hardware* necesario para el sistema de evaluación y de los módulos de adquisición de datos y control, el desarrollo del *software*, es decir, el sistema monitor y los programas de control, así como también el sistema de autoprueba y fallas.

CAPITULO

I

MICROPROCESADOR

MC 6809

I

Microprocesador 6809

I.a Arquitectura Interna

El microprocesador (MPU) 6809 fabricado con tecnología HCMOS es miembro de la familia M6800 de Motorola, y es en la actualidad el más poderoso microprocesador de 8 bits, ya que permite el manejo interno de 16 bits, soportando además modernas técnicas de programación, como la programación modular, entre otras cualidades. Es considerado por muchos como el MPU intermedio entre los microprocesadores de 8 y 16 bits, además de tener un set de instrucciones más lógico y comprensible en cuanto a sus nemónicos. Esta tercera generación adicional a la familia M6800, posee mejoras de arquitectura, las cuales incluyen: registros adicionales, set de

instrucciones extendido, y el más completo set de modos de direccionamiento que ningún otro MPU de 8 bits posea.

La figura siguiente nos muestra la arquitectura interna del 6809:

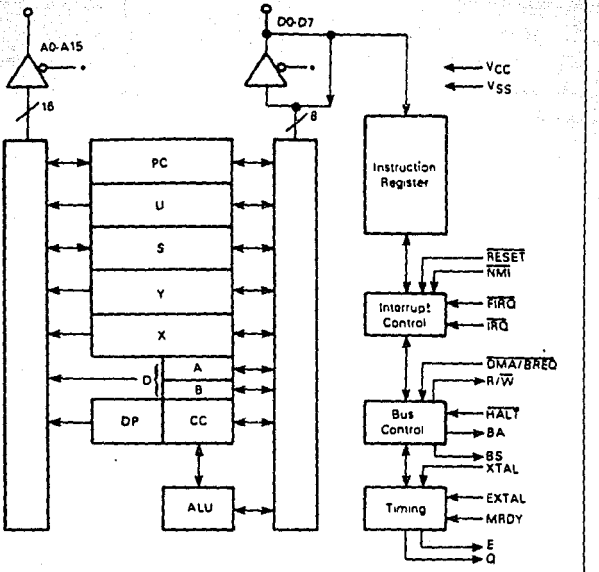


Diagrama de Bloques

Algunas de las características del 6809 se pueden agrupar en:

Compatibilidad con el MC6800

En hardware - Interfase con todos los periféricos del M6800

En software - Compatibilidad de instrucciones en código fuente y modos de direccionamiento

Características de Arquitectura

* Dos registros índice de 16 bit (S y U)
* Dos registros apuntadores indexados de 16 bits (X ,Y)
* Dos acumuladores de 8 bits (A,B), que pueden unirse para formar un acumulador de 16 bits (D)
* Un registro de página directa para dar direccionamiento directo a memoria

Características de Hardware

* Salidas de reloj para sincronización con dispositivos externos
* Entradas de control para los buffers del bus interno del MPU
* Salidas de control para casos de multiproceso
* Escritura rápida en memorias dinámicas
* Rápido direccionamiento válido en memorias lentas
* Oscilador interno (Frecuencia del cristal=4xE)
* Señal DMA/BREQ que permite operaciones DMA (Acceso Directo a Memoria)
* Señal de MRDY que permite extender el tiempo de acceso de datos para el uso de memorias lentas.

Características de Software

* Direccionamiento directo a cualquier lugar del mapa de memoria
* Contador de programa relativo
* Direccionamiento indexado expandido: 0, 5, 8 o 16 bits de offset constante, 8 o 16 bits de offset del acumulador
* Multiplicación de 8 x 8 bits no signados.
* Aritmética de 16 bits.

El 6809 tiene características de software y hardware que lo hacen el MPU ideal para aplicaciones estandares de control apto para

trabajar en un ambiente de multiproceso. Posee salidas de reloj que le permiten la sincronización con periféricos, sistemas y otros MPU's. El 6809 al igual que muchos otros MPU's fabricados por Motorola poseen un mapa de memoria unificado, es decir, no existe en ellos mapa de decodificación para puertos, lo que permite mayor potencia en el manejo de información proveniente de los puertos, ya que las instrucciones válidas para memoria son también empleadas para los puertos.

Descripción de los módulos del 6809

El **ALU** (Unidad Aritmética y Lógica) desempeña las operaciones lógicas y aritméticas. Unos registros especiales llamados acumuladores, conectados a la salida del ALU, son los encargados de almacenar los operandos y resultados de dichas operaciones. Estos acumuladores (A y B) son de propósito general,ya que son usados tanto para el manejo de información como de datos. Ciertas instrucciones unen a los registros A y B para formar un acumulador de 16 bits que es referido como el registro D y es formado tomando al registro A como el byte más significativo.

El **Registro de Código de Condición (CC)** define el estado del procesador en cualquier momento. Su contenido puede ser probado por instrucciones especiales. Cada bit de éste registro posee un significado determinado. El formato del registro de código de condición es el siguiente:

El bit 0 (C) es la bandera de carry, usualmente es el carry proveniente del ALU, aunque también es usado para representar al "borrow" para sustracciones en instrucciones como: CMP, NEG, SUB,

etc., y es el complemento del ALU binario.

El bit 1 (V) es la bandera de overflow (sobreflujo) y es puesta en uno por una operación que provoque un sobreflujo aritmético en complemento a dos. Este sobreflujo es detectado en operaciones en donde el carry proviene del bit más significativo (MSB) del ALU.

El bit 2 (Z) es la bandera de cero y es puesta en nivel alto si el resultado de la operación anterior fue identificada como cero.

El bit 3 (N) es la bandera de negativo y contiene el valor exacto del MSB generado en la operación previa. Cuando el resultado es negativo en complemento a dos el nivel de N es alto. El bit 4 (I) es el bit de máscara o mascarilla de IRQ (petición de interrupción). El MPU no reconoce interrupciones provenientes de la terminal IRQ si el bit I está en nivel alto. Las señales NMI, FIRQ, IRQ, RESET y SWI ponen a el bit I en nivel alto, mientras que las operaciones SWI2 y SWI3 no lo afectan.

El bit 5 (H) es el bit de medio acarreo, y es usado para indicar un carry proveniente del bit 3 del ALU y sólo se da en operaciones de adición de 8 bits (ADC o ADD). H es usado por la instrucción DAA para desempeñar operaciones de ajuste a decimal en BCD. El estado de ésta bandera es indeterminado en operaciones de resta.

El bit 6 (F) es la máscara de FIRQ (petición de interrupción rápida). El MPU no reconoce interrupciones de FIRQ si el nivel de éste bit es alto. Las señales NMI, FIRQ, SWI1 y RESET ponen al bit F en uno, y IRQ, SWI2 y SWI3 no lo afectan.

El bit 7 (E) es la bandera de entero, y cuando está en nivel alto indica que el estado completo del MPU, llamado estado de máquina (al contenido de todos los registros), es almacenado en el stack como reflejo del estado anterior (PC y CC). El bit E del registro de código de condición es almacenado en el stack y es empleado en el retorno de interrupciones (RTI) para determinar la extensión de una descarga del stack, por lo que el bit E representa una acción pasiva.

Registros de Indice (X,Y) Los registros de índice son usados en el modo de direccionamiento indexado. La dirección de 16 bit contenida en éste registro interviene en el cálculo de la dirección efectiva. Esta dirección puede ser tomada directamente del dato contenido en dicho registro o puede ser modificada por una

constante opcional o bien por un registro de offset. Durante algún modo indexado, el contenido de éste registro es incrementado o decrementado para apuntar a la siguiente dirección de la tabla de datos. Los cuatro registros apuntadores (X, Y, U, S) pueden ser usados como registros de índice o apuntadores.

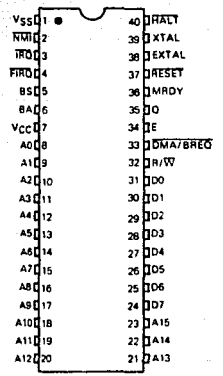
Apuntadores de Stack U, S (stack pointer) El apuntador de stack S es usado automáticamente por el procesador durante el llamado a subrutinas e interrupciones. El apuntador U es el llamado de usuario, ya que es controlado exclusivamente por el programador y permite argumentos para pasar a subrutinas o regresar de éstas con facilidad. Ambos registros permiten al 6809 el uso eficiente del stack, y un gran encadenamiento que nos permite soportar lenguajes de alto nivel así como una programación modular.

El **Contador de Programa (PC)** es usado por el procesador para apuntar a la dirección de la siguiente instrucción a ser ejecutada por el micro. La dirección relativa es una condición que permite al contador de programa ser usado como un registro de índice en algunas situaciones.

El **Registro de Página Directa DP** (direct page register) del MPU sirve de enlace en el modo de direccionamiento directo. Su contenido se refleja en la parte alta del bus de direcciones (A_8-A_{15}) durante la ejecución de una instrucción de direccionamiento directo.

Como los distintos módulos requieren de un medio de comunicación entre ellos para el intercambio de datos, comandos, etc; el procesador realiza todas las transferencias de información a través de 3 buses internos. Estos buses son el de **Control**, el de **Datos** y el de **Direcciones**.

La colocación de las terminales (patigrama) en el 6809 se muestran enseguida:



Disposición de Terminales

I.b Modos Direccionamiento y conjunto de instrucciones

Ib.1 Conjunto de Instrucciones.

Una de las características esenciales de los MPU consiste en realizar tareas que les son indicadas a través de instrucciones dadas por el programador; pero ¿ cuáles son los procesos que lleva a cabo el MPU 6809 para poder interpretar y ejecutar dichas instrucciones? ¿Cuántos tipos de instrucciones existen, y en que consiste cada grupo en particular?

Primeramente explicaremos cómo logra el 6809 "interpretar" las instrucciones para posteriormente poderlas ejecutar. A éste proceso se le conoce como ciclo de búsqueda y ejecución.

Ciclo de Búsqueda y Ejecución

Todo MPU para poder operar requiere de elementos periféricos a él, entre los que encontramos memorias RAM, ROM, generador de reloj, puertos paralelo, serie, etc. Y es precisamente en la memoria donde el programador almacena las instrucciones y los datos. Asumiremos que el contador de programa es válido en la dirección que presenta, contamos pues con 16 bits de direcciones que nos indican la dirección de la siguiente instrucción que será buscada en la memoria.

Para realizar éste ciclo, todos los procesadores proceden en tres pasos que son:

1) búsqueda de la siguiente instrucción
2) decodificación de la siguiente instrucción
3) ejecución de la instrucción

Búsqueda

Primeramente el contenido del PC es depositado en el bus de direcciones, pero al salir, también es incrementado, en 1, y éste valor es guardado en el PC, como la siguiente instrucción a buscar.

El valor del PC llega a la memoria, y en particular al

decodificador de instrucciones. Al mismo tiempo se recibe la señal de escritura RD enviada por el MPU. Unos pocos nanosegundos después, la memoria deposita los ocho bits de datos correspondientes a la dirección especificada, en el bus de datos. El MPU lee entonces el bus de datos y deposita el contenido dentro del registro específico IR, o registro de instrucciones. El ciclo de búsqueda se a completado. Los ocho bits de la instrucción están ahora en el registro interno especial del MPU llamado IR que no es accesible para el programador.

Decodificación y Ejecución

La instrucción contenida en el IR es decodificada por la unidad central del MPU y genera la secuencia correcta de señales tanto internas como externas para la ejecución de la instrucción especificada. Esta decodificación depende del tipo de instrucción que hallamos enviado. A continuación viene la ejecución, donde su rapidez depende del modo como localice la dirección de los operandos. De aquí la importancia de conocer y dominar los diferentes modos de direccionamiento.

Algunas instrucciones las ejecuta internamente el MPU, otras son búsquedas o depósito de datos, dentro de la memoria. La búsqueda y ejecución de una ejecución depende de varios ciclos de reloj.

Clases de Instrucciones

En realidad no existe una clasificación estándar de las instrucciones, sin embargo, para su estudio podemos distinguir seis categorías que son:

1) transferencia de datos
2) procesamiento de datos
3) apuntador de datos
4) prueba y bifurcación
5) entrada/salida
6) control

Antes de analizar cada grupo de instrucciones se muestra a continuación tabla de notación de los modos de direccionamiento ya que estos símbolos son empleados en los ejemplos

( )	= Apuntador de datos (8-bits) o para encerrar direcciones (16-bits)
EA	= La dirección efectiva; un apuntador dentro de la memoria creado como resultado de un modo de direccionamiento
M	= (EA) = El dato en el espacio de dirección (de memoria) apuntado para la dirección efectiva
MI	= Direccionamiento inmediato en memoria; el dato sigue inmediatamente al último byte del opcode
dd	= Offset de 8-bits (o una distancia relativa hacia una etiqueta que se evalúa con 8-bits
DDDD	= Offset de 16-bits (o una distancia relativa a una etiqueta)
P	= Inmediato, directo, indexado, extendido
Q	= Acumulador, directo, indexado, extendido
YYYY	= Offset en el rango de $-64K \leq YYYY \leq 64K$
ZZ	= Cualquier registro indexable (IX, IY, SP o US)
XX	= Valor de 8-bits en hexadecimal
*	= El PC comienza en la instrucción presente
*'	= Comienza en la instrucción siguiente
IN	= Sólo direccionamiento indexado
#	= Direccionamiento Inmediato con los bytes siguientes
$	= El valor siguiente es hexadecimal
%	= El valor siguiente es binario
<	= Antes de indexar; forza un byte de offset (para conocer la referencia siguiente), ó antes de la dirección absoluta; forza al direccionamiento directo
>	= Antes de la dirección absoluta forza al direccionamiento extendido
,	= Símbolo de indexamiento
[ ]	= Indirecto

Transferencia de Datos

Las instrucciones de transferencia de datos, los transfieren entre registros, entre registros y memoria, y entre un registro y un dispositivo de entrada/salida.

Algunos registros poseen instrucciones de transferencia especializada pudiendo ser usadas para organizar los datos, por ejemplo PUSH y PULL son operaciones para el manejo del stack.

Podemos clasificar las instrucciones de transferencia de datos del 6809 en tres categorías:

a) transferencia de 8 bits
b) transferencia de 16 bits
c) operaciones con el stack

A continuación examinaremos cada categoría.

Transferencia de 8 Bits de Datos

Las instrucciones de transferencia de datos se usan para cargar y transferir 8 bits de datos entre la memoria y los dos acumuladores

```
LDA $156A  ; Carga el acumulador A con el contenido de
             la localidad de memoria $156A
STB $158B  ; Almacena en la localidad de memoria $158B
             lo que hay en B
```

es decir copian el contenido de una localidad de memoria a alguno de los acumuladores

```
TFR A,DP   ; Transfiere el contenido de A al
             registro DP
```

La instrucción de intercambio, "intercambia" el contenido de dos registros.

```
EXG A,B    ; Copia el contenido de A en B y el de
             B en A
```

Transferencia de Datos de 16 Bits

Podemos usar las mismas instrucciones de carga y almacenamiento que fueron usadas por la transferencia de datos de 8 bits. Contamos con 5 registros de 16 bits D,X,Y,U, y S que podemos almacenarlos en memoria y/o cargarles un valor proveniente de la misma.

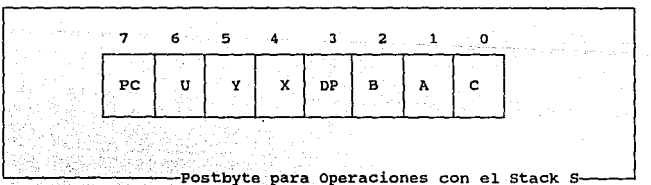
La instrucción TFR nos sirve para transferir un registro de 16 bits a cualquier otro registro que sea también de 16 bits, incluyendo el PC.

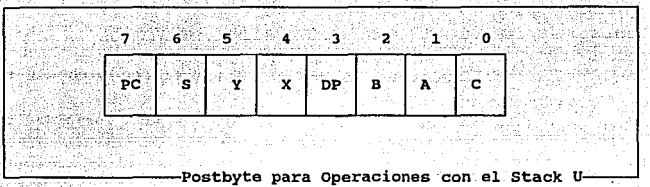
La instrucción EXG nos es útil para intercambiar el contenido entre dos registros cualesquiera de 16 bits, también incluyendo el PC. Es importante notar que por la transferencia de un nuevo valor dentro del PC, o por el intercambio del PC con otros registros podemos provocar que la ejecución del programa continúe en la localización de dirección que tiene el nuevo valor del PC.

Operaciones con el Stack

Las operaciones con el stack mueven datos entre la parte alta del stack y los registros. EL 6809 tiene dos instrucciones de stack: PUSH y PULL, así como dos apuntadores de stack; el apuntador de stack de hardware, S, y el apuntador de stack de usuario, U.

Los registros que serán introducidos dentro del stack son indicados en el byte que sigue inmediatamente al código de operación (opcode) de la instrucción del stack. Cada bit en éste estado llamado el postbyte, indica un registro. Cuando un bit es activado, éste registro es usado en la operación del stack. La figura siguiente nos puede aclarar ésta idea:





Las dos instrucciones para el apuntador del stack S, son PHS y PULS, mientras que PSHU y PULU lo son para el apuntador de stack U.

Siempre que un registro de 8 bits es introducido en el stack, el apuntador de stack es decrementado en uno, pero si el registro es de 16 bits será decrementado en dos.

Las instrucciones de PULL son iguales al PUSH, excepto que el apuntador de stack será incrementado.

Procesamientos de Datos

Las instrucciones de procesamiento de datos modifican los datos dentro de la computadora. Estas instrucciones podemos agruparlas en 4 categorías:

a) operaciones aritméticas
b) operaciones lógicas
c) rotar y operaciones de corrimiento
d) manejo de bits

Operaciones Aritméticas

El 6809 provee de tres operaciones aritméticas: suma, resta, y multiplicación.

La suma tiene dos tipos de instrucciones que son: con acarreo ADC y sin él ADD.

Similarmente la resta cuenta con dos tipos de instrucciones: con acarreo SBC y sin él SUB.

EL 6809 cuenta además con tres instrucciones especiales:

DAA COM NEG

La suma decimal se realiza con la instrucción DAA, que es útil también para implementar operaciones BCD -normalmente sumas y restas- .

COM calcula el complemento a dos en el acumulador o en alguna dirección de memoria,

NEG obtiene el negado en el acumulador o en alguna localidad de memoria (dentro del formato de complemento a dos).

Todas éstas instrucciones operan con 8 bits de datos. Las operaciones con 16 bits de datos son mas restringidas: sólo ADD y SUB están disponibles en el acumulador D.

Finalmente, también se cuenta con instrucciones de incremento y decremento; éstas operan en el acumulador y la memoria bajo un formato de 8 bits de datos. Podemos incrementar o decrementar los registros indexados y los apuntadores de stack en un formato de 16-bits, con el modo de direccionamiento de auto incremento/decremento.

En general, todas las operaciones aritméticas modifican alguna de las banderas del registro de código de condición. Sin embargo las instrucciones INC y DEC, que operan con el acumulador de 8-bits y con la localización de memoria, no modifican el bit de acarreo C. Esto significa que si incrementamos o decrementamos rebasando el valor de 255, el bit C, en el registro del código de condición, CC, no sufrirá cambios.

Si es necesario detectar un valor que cambia de positivo a negativo o viceversa, podemos probar los bits N y V.

También, es importante notar que ADD y ADC siempre afectan todas las banderas de los códigos. Esto no significa que todas éstas serán necesariamente diferentes después de la ejecución; sin embargo pueden serlo.

Operaciones Lógicas

El 6809 proporciona tres operaciones lógicas, AND, OR(inclusiva) y EOR (exclusiva) así como la suma y comparación de instrucciones CMP. Las operaciones lógicas operan con 8 bits de datos, y la instrucción CMP opera con 8 o 16 bits de datos. A continuación examinaremos éstas instrucciones.

AND Esta operación lógica se caracteriza por la tabla de verdad siguiente.

AND	0	1
0	0	0
1	0	1

Si una de las entradas es cero, el resultado será cero. Esta característica llamada enmascaramiento, es usada para colocar un cero en algún lugar requerido de la palabra.

```
LDA     10101010
ANDA    %11110000
```

El resultado de éste programa será el valor 10100000 y se encontrará en el acumulador. El símbolo % es usado para indicar que se trata de un valor binario.

OR La instrucción OR genera la operación OR inclusiva y se caracteriza por la siguiente tabla de verdad.

OR	1	0
1	1	1
0	1	0

La lógica OR se caracteriza por que si uno de los operandos es 1, el resultado es siempre 1. Resulta entonces obvio el uso de la OR, que consiste en habilitar cualquier bit de la palabra con 1.

```
LDA    10101010
ORA    %00001111
```

El valor final en el acumulador será 10101111.

EOR Genera la " OR exclusiva ", donde el resultado es 1 si y solo si, uno de los operandos es igual a 1. Si ambos operandos son iguales a 1 el resultado es 0.

EOR	0	1
0	0	1
1	1	0

Podemos usar la OR-exclusiva para comparaciones; si cualquier bit es diferente, entonces la OR-exclusiva de las dos palabras no será cero. También podemos usarla para el complemento de una palabra, para ello hacemos en una palabra que todos los bits sean 1's.

```
LDA   10101010
EOR   %11111111
```

El valor final del acumulador es 01010101. Podemos verificar que se trata del complemento del número original.

Otra ventaja de la instrucción EOR es que podemos cambiar cualquier bit del acumulador, o mantenerlo después de ejecutar EOR.

Operaciones de Rotar y Corrimiento

Corrimiento

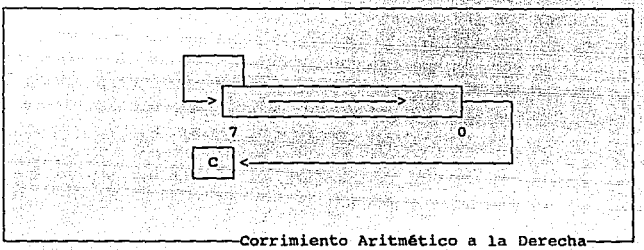
Es necesario que diferenciemos bien entre las operaciones de corrimiento y rotación. En una operación de corrimiento, los bits del registro son recorridos a la izquierda o derecha por una posición de bit. El bit que sale del registro entra al bit de acarreo C, y el bit que entra es cero.

Sin embargo existe una excepción en el corrimiento aritmético a la derecha. Cuando deseamos realizar operaciones con números negativos en el formato del complemento a dos, el bit más a la izquierda es el bit de signo. En el caso de los números negativos es 1.

Cuando dividimos un número negativo por 2, para el corrimiento a la derecha, el bit de signo deberá mantenerse negativo, i.e., el bit más a la izquierda debe ser 1. Esto puede hacerse

automáticamente con la instrucción ASR (corrimiento aritmético a la derecha). Con ésta instrucción, el bit entrante a la izquierda es igual al bit de signo. Este es 0 si el bit más a la izquierda era 0, y 1 si el bit más a la izquierda era 1.

La figura siguiente ilustra ésta situación :

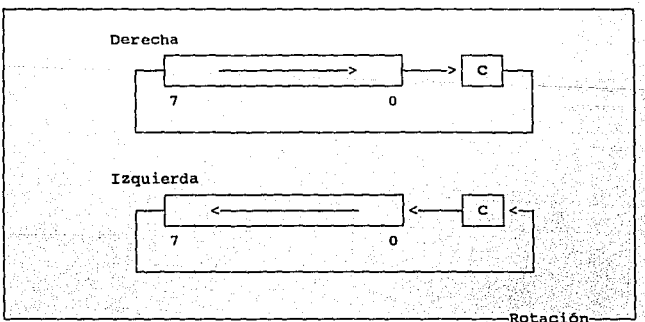


Corrimiento Aritmético a la Derecha

Rotación

Una rotación difiere de un corrimiento en que el bit que entra al registro, proviene del bit de acarreo,por lo que puede ser considerada como una operación de 9-bits.

La siguiente figura ilustra la rotación:



Rotación

Manejo de Bits

Podemos usar también dos instrucciones especiales para operar sobre el registro de código de condición (CC) : ANDCC y ORCC. Estas dos instrucciones permiten operaciones lógicas especificadas sobre el registro de código de condición, usando el byte inmediato a la instrucción, como la máscara. Por éste camino, los bits en CC pueden ser borrados o activados. Sólo en el modo de direccionamiento inmediato están disponibles éstas instrucciones.

Finalmente, la instrucción de prueba de bit, BIT, prueba el código de condición del resultado de la operación AND entre el acumulador y una localidad de memoria de 8-bits. En la instrucción de prueba de bit, jamás es cambiado el acumulador ni la localización de memoria. La operación AND cambia el CC, pero no los bits que están siendo probados.

Instrucciones de Apuntador de datos

Las instrucciones de carga de dirección efectiva (LEA) es una instrucción del apuntador de datos del 6809. Esta carga cuatro registros de direcciones: X, Y , S, y U. Las cuatro formas de ésta instrucción son: LEAX, LEAY , LEAS y LEAU. Cada registro de dirección está cargado de otro (o del mismo) registro de direcciones. Al mismo tiempo,un número especificado en la instrucción o de uno de los acumuladores A, B o D, es sumado al registro destino.

La instrucción LEA carga la dirección, no el dato apuntado por el registro de dirección.

Podemos definir fácilmente bloques de datos relativos a otras direcciones durante la ejecución del programa, si usamos la instrucción LEA, como se muestra a continuación:

INSTRUCCIÓN	OPERACIÓN	COMENTARIOS
LEA 10,X	X+10 → X	; Suma 10 a X
LEA 500,X	X+500 → X	; Suma 500 a X
LEAY A,Y	Y+A → Y	; Suma el acumulador con Y
LEAY -10,U	U-10 → U	; Resta 10 de U
LEAS -10,S	S-10 → S	; Usado para reservar área sobre el stack
LEAS 10,S	S+10 → S	; Usado para limpiar stack
LEAX 5,S	S+5 → X	; Transfiere así como suma

Manejo de la Instrucción Lea

Operaciones de Prueba y Bifurcación del 6809

Instrucciones de Prueba

Las instrucciones de prueba permiten verificar los bits y registrar su condición para valores de 0 y 1, y para combinaciones de éstos valores. Las banderas nos permiten llevar su registro dentro del registro de código de condición.

Las instrucciones del 6809 permiten realizar las siguientes pruebas:

1) combinaciones de bit
2) la posición de un sólo bit en una palabra
3) el valor de un registro comparado con el valor de la localización en memoria (> , < , =)

Generalmente las instrucciones del MPU están limitadas a probar bits individuales del registro de banderas; en comparación con otros procesadores el 6809 ofrece mejores facilidades de prueba. C es el bit de acarreo, V es sobreflujo, Z es cero y N es negativo. Los bits 4, 6 y 7 son usados con interrupciones. El código H es

usado para operaciones aritméticas BCD y no puede ser probado directamente. Los otros cuatro códigos (C, V, Z y N) pueden ser probados con instrucciones de bifurcación condicional. A continuación describiremos el papel de cada bit.

Acarreo (C)

En el caso de todos los MPU, y del 6809 en particular, el bit de acarreo realiza una doble función, la primera consiste en indicar si en la suma o resta se generó un acarreo; y en la segunda, es usado como un noveno bit en el caso de operaciones de corrimiento y rotación. Usando éste sólo bit que desempeña ambas funciones, se facilitan algunas operaciones, como la división.

Todas las operaciones de corrimiento y rotación usan el bit de acarreo y éste estará habilitado o deshabilitado, dependiendo del valor del bit que está entrando o saliendo de la palabra.

En el caso de instrucciones lógicas, podemos usar ANDCC y ORCC para habilitar o deshabilitar directamente el bit de acarreo. Las instrucciones que afectan el bit de acarreo son:

ADD	SUBC	ASL	LSR	CLR	NEG
ADC	ANDCC	ASR	ROL	CMP	DAA
SUB	ORCC	LSL	ROR	COM	MUL

También algunas instrucciones de transferencia de datos e instrucciones de control, incluyendo :

PULS PULU TFR EXG RTI CWAI

afectan el bit C, y todos los demás bits del código de condición, por lo que son cargados en el registro de código de condición CC.

Sobreflujo (V)

La bandera de sobreflujo detecta si, durante una suma ó resta, el signo del resultado fue "accidentalmente" cambiado, debido al sobreflujo del resultado dentro del bit de signo, ya que usando una representación de 8 bits, el mayor número positivo y el menor número negativo en complemento a dos que se puede generar son +127 y -128 respectivamente.

La condición del código de bit V es afectado por :

ADC	CMP	LSL	SBC
ADD	DEC	NEG	SUB
	ASL	INC	ROL

Las siguientes instrucciones siempre deshabilitan el bit V :

AND	CLR	LD	OR	COM
EOR	SEX	ST	BIT	TST

Mientras que para la instrucción DAA su estado es indefinido.

Bit de Medio Acarreo (H)

La bandera de medio acarreo indica un posible acarreo del bit 3 al bit 4 durante una operación de suma. En particular, es usada internamente por la instrucción de ajuste de suma decimal (DAA), en respuesta al ajuste del resultado; si éste ultimo es un valor correcto.

La bandera del medio-acarreo es habilitada durante una suma de 8-bits, cuando hay un acarreo del bit 3 al bit 4; siendo deshabilitada cuando no existe acarreo. Una suma de 16-bits no afecta el bit H. Las instrucciones de 8-bits ADD y ADC si afectan el bit H. Las instrucciones ASL, ASR, NEG, SBC y la forma de 8-bits de las instrucciones CMP y SUB mantienen el bit H sin afectar.

Cero (Z)

La bandera Z del código de condición nos permite usar las instrucciones de comparación indicadas en el set.

Si una operación arroja un resultado de cero, ó en el caso de una transferencia de datos, o si la palabra de 16-bits es cero, el bit cero se habilita a 1. En cualquier otro caso Z es deshabilitado a cero.

Las siguientes instrucciones condicionan el valor del bit Z

ADC	ADD	AND	OR	ASI	ASR	BIT	CMP	COM
DAO	DEC	EOR	INC	LD	NEG	ST	TST	LEAX
LEAY	LSL	LSR	MUL	SEX	ROL	ROR	SBC	SUB

La instrucción CLR siempre habilita el bit Z.

Negativo (N)

Este bit del código refleja el valor del bit más significativo de un resultado, o de un byte (o de un dato de 16-bits) mientras está siendo transferido. En notación de complemento a dos, el bit más significativo representa el signo: 0 indica un número positivo, y 1 indica un número negativo. Es por esto que el bit 7 (o bit 15, para números de 16-bits) es llamado el bit negativo.

En todos los MPU, el bit de signo juega un papel importante cuando se está comunicando con dispositivos de entrada/salida, por lo que usualmente es el bit más conveniente de prueba cuando examinamos el estado de un dispositivo de entrada/salida, el registro de estado lee automáticamente la condición del bit negativo, que es habilitado por el valor del bit 7 del registro de estado y puede ser continuamente probado por programa. Esta es la causa de que el registro de estado de muchos chips de entrada/salida, conectados al sistema del MPU, tienen el indicador más importante (usualmente ready/not ready) en el bit de la posición 7.

Las siguientes instrucciones afectan el bit negativo:

ADC	ASL	COM	INC	LSL	ROR
ADD	ASR	DAA	LD	SEX	SBC
AND	BIT	DEC	ST	NEG	SUB
	OR	CMP	EOR	TST	ROL

Las instrucciones CLR y LSR siempre "limpian" el bit N.

Instrucciones de Bifurcación

Una instrucción de bifurcación provoca una "bifurcación" a la dirección especificada en el programa. Estas cambian el flujo normal de la ejecución del programa de un modo secuencial dentro de él, aún segmento diferente del programa que es repentinamente ejecutado. Las bifurcaciones pueden ser condicionales o incondicionales. Una bifurcación incondicional es aquella donde el bifurcamiento ocurre a una dirección específica, sin tomar en cuenta cualquier otra condición. Una bifurcación condicional es aquella donde el bifurcamiento ocurre a una dirección específica sólo si una o más condiciones son cumplidas. Estas son el tipo de

instrucciones de salto usadas para tomar decisiones basadas sobre los resultados calculados.

Para describir las instrucciones de la bifurcación condicional es necesario entender la función que desempeña el registro de código de condición, por lo que todas las decisiones de bifurcación están basadas sobre éstos bits de condición.

A continuación examinaremos en mas detalle, las instrucciones de bifurcación que ofrece el 6809.

Los dos tipos de instrucciones de bifurcación que posee el 6809 son:

a) bifurcaciones de control del programa principal
L) bifurcaciones especiales: usadas para saltar hacia y desde una subrutina (JSR, BSR y RTS)

Como resultado de cualquier instrucción de bifurcación, el contador del programa (PC) es recargado con una nueva dirección y la ejecución usual del programa se toma desde éste punto. La potencia completa de las instrucciones de bifurcación sólo pueden entenderse en el contexto de varios modos de direccionamiento provistos por el MPU.

En el caso de una bifurcación condicional, uno o más de los cuatro bits del código de condición Z, C, U y N, pueden ser probados para el valor de 0 o 1.

Las abreviaciones correspondientes para los bits individuales son:

```
BCC = bifurca si el acarreo está limpio          ( C = 0)
BCS = bifurca si el acarreo está habilitado      ( C = 1)
BEQ = bifurca si es igual a cero                 ( Z = 1)
BMI = bifurca si es negativo                     ( N = 1)
BPL = bifurca si es positivo                     ( N = 0)
BVC = bifurca si el sobreflujo está limpio       ( U = 0)
BUS = bifurca si el sobreflujo está habilitado   ( U = 1)
```

Estas son algunas instrucciones de bifurcación, que prueban la combinación del código de condición de bits.

Frecuentemente son usadas después de una instrucción de comparación CMP.

Las siguientes son las abreviaciones para las instrucciones de bifurcación condicional.

BGE = bifurca si es mayor o igual a
BGT = bifurca si es mayor que (saltos cortos)
BHI = bifurca si es mayor que (saltos largos
BLE = bifurca si es menor que o igual
BLS = bifurca si es igual
BLT = bifurca si es menor que

Existen dos instrucciones de bifurcación que tienen el mismo código de operación que otras dos instrucciones, estas son:

BHS = bifurca si es mayor o igual BCC
BLO = bifurca si es el mas bajo BCS

La instrucción de bifurcación incondicional es BRA, mientras BRN es la instrucción de "nunca bifurcación". Realmente ésta última es una operación nula. Las operaciones de bifurcación condicional son una ventaja del 6809, pues generalmente, la mayoría de los MPU de 8 bits no las poseen.

La bifurcación corta puede accesar a cualquier instrucción en el rango de +129 a -129 bytes relativos al primer byte de la instrucción de la bifurcación. Por otro lado, la bifurcación larga tiene instrucciones con direcciones de 16 bits y al sumarse al PC, el bifurcamiento puede ser a cualquiera de las 65,436 localidades de memoria del 6809.

Este tipo de instrucciones evita la necesidad de bifurcaciones por una instrucción de salto (JMP). Es importante tener presente que los códigos de operación para bifurcaciones largas y cortas son diferentes.

En el caso de rutinas de interrupciones, contamos con la instrucción RTI, que es especial para retorno.

Un tipo más de bifurcación especializada está disponible: la instrucción de interrupción de software SWI. Esta instrucción guarda todos los registros en el stack de hardware S y entonces

ejecuta un salto para buscar un nuevo PC de una de las tres direcciones de las ultimas localidades de memoria. Las tres posibles localizaciones para el par de bytes con el cuál se forma el nuevo PC son: FFFA:FFFB, FFF4:FFF5 y FFF2:FFF3. SWI en una instrucción muy útil pués permite salvar el estado de maquina intacto. Frecuentemente es usada para saltar a programas especiales en los cuales comienzan y complementan otros programas.

Instrucciones de Entrada/Salida

Estas instrucciones nos permiten el manejo de dispositivos de entrada/salida (I/O). En la práctica muchos MPU usan memoria mapeada de I/O, y como los dispositivos de I/O son conectados al bus de direcciones en el mismo camino que los chips de memoria, estos también deben ser direccionados.

Podemos direccionar dispositivos de I/O por alguno de dos caminos: con localización de memoria (usando las instrucciones descritas previamente), o por el uso específico de instrucciones de I/O. El 6809 no tiene instrucciones especiales de I/O. Normalmente las instrucciones de direccionamiento de memoria usan tres bytes: una para el código de operación y dos para la dirección. Esto provoca que las instrucciones se ejecuten de manera lenta, pués se necesitan tres accesos de memoria. Sin embargo, si usamos el modo de direccionamiento especial "directo por página", donde la dirección es formada por el registro directo de página y un byte en la instrucción, entonces la instrucción para accesar un dispositivo de I/O requerirá sólo dos bytes de duración, siendo así más rápida su ejecución.

Instrucciones de Control en el 6809

Las instrucciones de control proporcionan una sincronización a las señales y pueden suspender o interrumpir un programa. Estas instrucciones modifican el modo de operación del MPU y manipulan su estado interno de información. El 6809 ofrece tres instrucciones de control :

NOP SYNC CWAI

La instrucción NOP es una instrucción de no operación que dura dos ciclos de reloj y es usada para introducir deliberadamente retardos, (2 ciclos = 2 μs con un cristal de 4 MHz) o para llenar

de intervalos un programa durante la fase de depuración.

Cuando una instrucción SYNC es ejecutada el MPU introduce estados de sincronía, detiene el procesamiento de instrucciones y espera una interrupción.

Cuando la interrupción ocurre, los estados de sincronización son borrados y el procesamiento continua. Si la interrupción está habilitada, y tarda tres ciclos o más, el procesador deberá realizar la rutina de interrupción. Si la interrupción es mascarable o si es menor de tres ciclos de duración, el MPU simplemente continua con la siguiente instrucción, sacando del stack los registros. Mientras existan estados de sincronización, los buses de datos y direcciones estarán en alta impedancia.

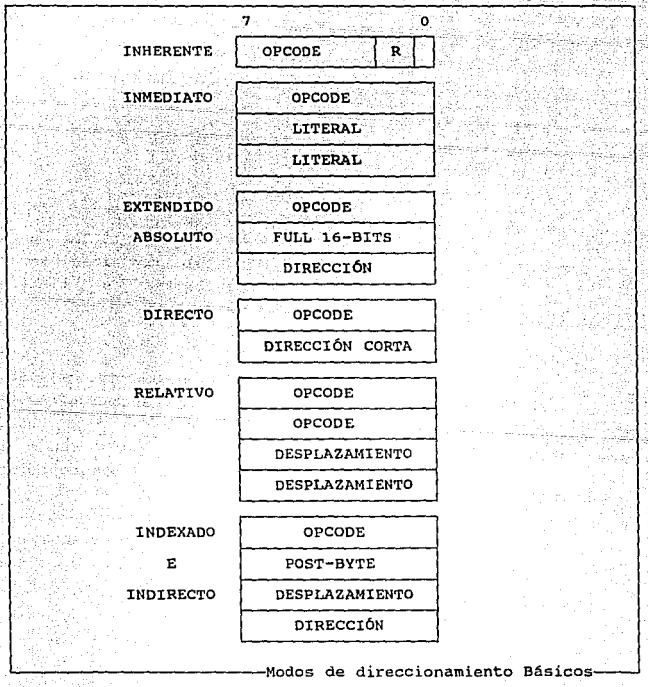
Finalmente, la última instrucción de control CWAI (borra y espera una interrupción), es para limpiar los banderas del código de condición y esperar una interrupción. Aquí los buses de datos y direcciones no son colocados en estado de alta impedancia.

Ib.2 Modos de Direccionamiento

Posibles Modos de Direccionamiento

Direccionar se refiere a especificar dentro de una instrucción la localización del operando y cuál instrucción será el operador.

A continuación examinaremos los seis modos básicos de direccionamiento.



Modos de direccionamiento Básicos

Modo de Direccionamiento Inherente

Las instrucciones que operan exclusivamente con registros usan normalmente el direccionamiento inherente. El código de operación de la instrucción posee toda la información necesaria, pués podemos observar que no se incluye información en el campo del operador.

Algunas de las instrucciones que usan éste direccionamiento son:

ABX	WSI	LSRA	RORB
ASLA	DAA	LSRB	RTI
ASLB	DECA	MUL	RTS
ASRA	DECB	NEGA	SEX
ASRB	EXG	NEGB	SWI
CLRA	INCA	NOP	SYNC
CLRB	INCB	ROLA	TFR
COMA	LSLA	ROLB	TSTA
COMB	LSLB	RORB	TSTB

Instrucciones como MUL, requieren más de dos ciclos para ejecutarse. Otras instrucciones, como TFR y EXG, necesitan mas de un byte. El direccionamiento inherente también es llamado direccionamiento de registro.

Direccionamiento Inmediato

En el direccionamiento inmediato, la dirección efectiva del dato está localizada inmediatamente después del código de operación. El 6809 usa valores de 8 y 16 bits, dependiendo del tamaño del argumento especificado por el código de operación. Desde luego el direccionamiento inmediato implica que el dato es un valor conocido o que el programa ya lo ha creado.

En el estándar del lenguaje ensamblador del 6809 señalamos que se trata de un direccionamiento inmediato si colocamos delante del operador el símbolo *#*. Las siguientes son algunas instrucciones que usan el direccionamiento inmediato:

```
LDA  # n          un  byte
LDX  # nn         dos bytes
```

Direccionamiento Extendido (o Absoluto)

Direccionamiento Extendido Directo

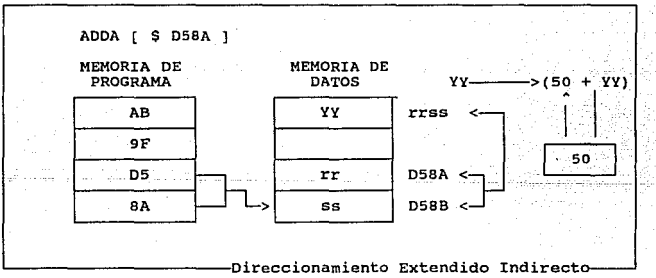
La dirección de 16 bits del operando sigue al código de operación. Este tipo de direccionamiento necesita, por lo tanto, tres bytes de instrucción. Este modo frecuentemente es usado para entradas y salidas, en donde desde la dirección de memoria se asigna el dispositivo de I/O.

```
LDA  > $0100
JMP  > $1234
```

donde los dos números hexadecimales representan direcciones de 16 bits de datos o instrucciones.

Direccionamiento Extendido Indirecto

En este modo la dirección efectiva es localizada dentro de la dirección de los dos bytes del programa siguientes al código de operación. Es decir, la instrucción dice al procesador dónde encuentra la dirección, no que valor tiene ésta.



Direccionamiento Directo

En el direccionamiento directo el código de operación es seguido por una dirección de 8 bits. La ventaja de ésta es que sólo requiere dos bytes. Algunos MPU's, tienen la desventaja de que el límite de direccionamiento dentro de éste modo va de la dirección 0 a la 255. El 6809 no tiene dichas limitaciones pués es posible direccionar cualquier byte en memoria por el uso del

direccionamiento directo y el manejo del registro directo de página DP.

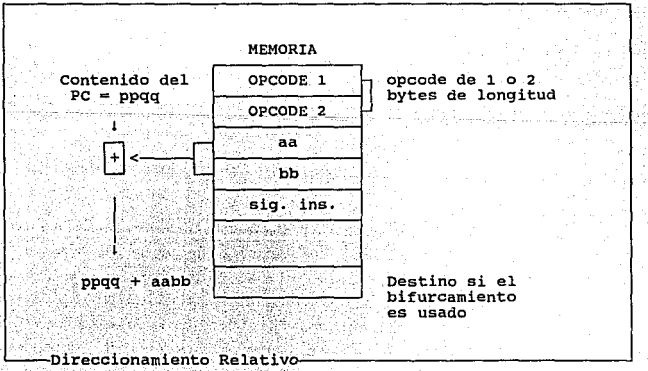
Cuando el direccionamiento directo es usado, el byte bajo de la dirección es el que sigue inmediatamente al código de operación (opcode), y el byte alto es el contenido del registro DP. Con cambios apropiados de éste registro, cualquier página en memoria puede ser direccionada. Cuando el registro DP contiene cero, el 6809 en el modo de direccionamiento directo opera de la misma manera que otros MPU's.

Direccionamiento Relativo

El direccionamiento relativo consiste en sumar un número signado al contenido del PC. Cuando se usa éste modo conjuntamente con las instrucciones de bifurcación, la suma se vuelve el nuevo contenido del PC y el bifurcamiento es cargado; si no el PC avanza a la siguiente instrucción.

Por ejemplo los bytes siguientes al opcode de bifurcamiento son tratados como un offset que es sumado al PC.

Toda la memoria puede ser alcanzada con el direccionamiento relativo largo, el offset, es decir la constante signada, puede tener ±7 bits o ±15 bits de longitud.



Direccionamiento Relativo

Para conocer el tiempo que tomará esta instrucción se debe proceder con precaución.

Si la prueba se verifica o si falla, esto es, si hay o no bifurcación, todas las instrucciones de bifurcación corta requieren de tres ciclos.

Sin embargo, la instrucción de bifurcación larga requiere cinco ciclos cuando la prueba no se cumple, y seis cuando se cumple y es realizada. Un valor promedio es usado frecuentemente para la duración de una bifurcación larga.

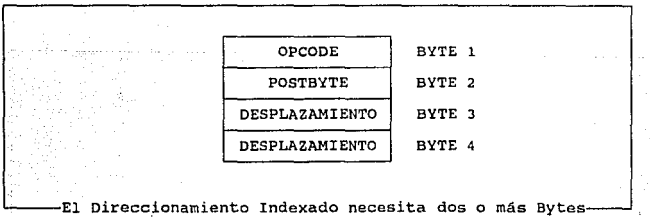
Este problema del tiempo no se aplica a todas las instrucciones de bifurcación larga, LBRA, es una instrucción que no prueba ninguna condición, y siempre tarda cinco ciclos.

Nota: la regla para diferenciar la instrucción de salto absoluto de la bifurcación relativa, consiste en que la instrucción de salto es etiquetada como JMP.

Direccionamiento Indexado

En todos los direccionamientos indexados uno de los registros apuntadores (X,Y,S,U y algunas ocasiones PC), es usado en el cálculo de la dirección efectiva (EA) del operando que será usado por la instrucción. El byte siguiente de la instrucción, especifica el tipo básico y la variación del modo de direccionamiento así como el apuntador de registro que será usado.

La estructura de una instrucción indexada aparece en la siguiente figura:



La tabla siguiente muestra el formato permitido para el post-byte.

Asignación de Bits

Registro de bit del Postbyte								DIRECCIONAMIENTO
7	6	5	4	3	2	1	0	INDEXADO
0	R	R	X	X	X	X	X	EA=,R ± 4 Bits offset
1	R	R	0	0	0	0	0	,R +
1	R	R	1	0	0	0	1	,R ++
1	R	R	0	0	0	1	0	,- R
1	R	R	1	0	0	1	1	,-- R
1	R	R	1	0	1	0	0	EA = ,R ± offset
1	R	R	1	0	1	0	1	EA = ,R ± ACCB offset
1	R	R	1	0	1	1	0	EA = ,R ± ACCA offset
1	R	R	1	1	0	0	0	EA = ,R ± 7 bits offset
1	R	R	1	1	0	0	1	EA=,R ± 15 bits offset
1	R	R	1	1	0	1	1	EA = ,R ± D offset
1	X	X	1	1	1	0	0	EA=,PC ± 7 bits offset
1	X	X	1	1	1	0	1	EA=,PC ± 15 bits offset
1	R	R	1	1	1	1	1	EA = ,Dirección

3 2 1

1) Campo del modo de Direccionamiento

2) Campo Indirecto
Bit de signo cuando B7 = 0

3) Campo de Registro
00:R = X
01:R = Y
10:R = U
11:R = D
X = no importa

Registro del Postbyte en el Direccionamiento Indexado

Direccionamiento Indexado con Cero-offset

En el modo indexado de cero-offset, un registro de dirección contiene la dirección efectiva del dato que será usado por la instrucción. Este es el modo indexado más rápido, por lo que el desplazamiento no es necesario. La instrucción es de dos bytes de longitud.

Esta opción permite seleccionar un auto incremento/decremento para uno o dos bits. Cuando en éste modo se selecciona el apuntador de registro, allí se encuentra la dirección efectiva del dato que será usado por la instrucción.

```
LDA    0,Y
LDB     ,U
```

Indexado con Offset-Constante

Cuando éste modo de direccionamiento es usado, el complemento a dos del offset y el contenido de uno de los registros apuntadores son sumados para formar la dirección efectiva (EA) del operando. Los registros de dirección se mantienen inicialmente sin cambio por la suma.

Tenemos tres tamaños de offset disponibles que son:

1) ± 4 bits (-16 a +15)
2) ± 7 bits (-128 a +127)
3) ± 15 bits (-32768 a +32767)

El offset constante de, ± 4 bits, usa el bit 4 del postbyte del opcode como un bit de signo, y los bits 0 al 3 como un offset constante. Esta instrucción es la menor, contando con dos bytes.

En el offset constante de, ± 7 bits, se designa con el byte siguiente al postbyte del opcode, un offset en complemento a dos. Esta instrucción tiene como mínimo tres bytes,

opcode + postbyte + offset.

Finalmente el offset de ± 15 bits, especifica en los dos bytes siguientes al postbyte del opcode, un offset en complemento a dos, ésta instrucción usa un mínimo de 4 bytes,

opcode + postbyte + dos bytes de offset.

Las siguientes instrucciones son algunos ejemplos:

```
LDA  33,X          LDY  -2,S          LDX  400,Y
```

Como programadores normalmente no nos preocuparemos acerca del offset, pués el ensamblador se encargará de tomarlo dentro del contador.

Existe también un tipo de direccionamiento indexado con offset constante de tipo indirecto como se muestra a continuación:

```
LDA [,X]
LDB [0,Y]
LDX [64000,S]
```

Direccionamiento Indexado con Acumulador-Offset

Cuando ésta opción es seleccionada, se designan los registros A,B o D como un offset en complemento a dos. La instrucción tiene un mínimo de dos bytes. Sin embargo,en todos los casos el offset es temporalmente sumado al contenido del registro apuntador que haya sido seleccionado para formar la dirección efectiva (EA).

Este modo es similar al indexado con offset constante excepto que el valor del complemento a dos de uno de los acumuladores (A,B o D) y el contenido de uno de los registros apuntadores (X,Y,S o U) son sumados para formar la EA.

El apuntador resultante es usado para recuperar otro apuntador de la memoria.

```
LDA A,X
LDB B,Y
LDA D,U
```

Indexado con Auto Incremento/Decremento

Si un registro de dirección está apuntando al comienzo (o fin) de una tabla de datos, el modo de auto incremento/decremento proporciona un método eficiente para accesar elementos sucesivos.

En el modo de auto incremento el registro de dirección es primeramente usado como la dirección efectiva que buscará el operando. Entonces el registro de direcciones es incrementado en uno o dos antes de que la siguiente instrucción sea buscada. Esto permite escalar a través de una tabla, de las direcciones bajas a las altas. El registro de direcciones es incrementado por uno si se

usa un dato de 8-bits, y por dos si el dato usado es de 16 bits.

Un + después del registro indica un modo de auto incremento, y que el dato es de 8-bits, mientras que ++ indica uno de 16-bits.

```
LDA ,X+
STD ,Y++
```

Esto indica que es el programador quien decide si es necesario un incremento simple o doble.

El modo de auto decremento es el contrario al modo de autoincremento

Cuando el modo de auto decremento es usado, el registro de dirección es decrementado por uno o dos, antes de que éste sea usado para buscar el operando, mientras que en el modo de auto incremento, el incremento es colocado después que el operando es buscado.

Con el auto decremento, una tabla puede ser accesada de las direcciones altas a las direcciones bajas.

Un " - " delante del nombre del registro indica el modo de auto decremento.

```
LDB  ,-Y
LDX  ,--S
```

En el segundo ejemplo, " -- " significa que S es decrementado por dos antes que sea usado como la dirección efectiva del operando.

El pre-decremento y post-incremento natural de éstos modos permite que sean usados para realizar stacks con los registros X y Y, como también con U y S, es decir para crear stacks en software.

Direccionamiento Indexado Indirecto

El modo de direccionamiento indexado indirecto es una combinación de los modos de direccionamiento indirecto e indexado.

En éste modo la dirección efectiva del operando está contenida en la localidad de memoria formada por el contenido de un registro de dirección, más cualquier offset. La característica de indirecto es indicada por el encerramiento del operando especificado por paréntesis cuadrados, [ ].

En el siguiente ejemplo el acumulador A es cargado indirectamente usando una dirección efectiva calculada del registro X y un offset:

```
$0100   LDX   #$F000   carga inmediata de X
$0103   LDA   [$10,X]  EA es ahora $F010
$F010   $F1            $F150 es ahora la nueva EA
$F150   $AA
```

Luego de la ejecución, A contiene el dato $AA, y X contiene $F000.

Existen dos casos que no pueden ser usados para direccionamiento indexado indirecto. Estos son:

a) auto incremento/decremento para uno
b) auto incremento/decremento con offset constante de 4 bits

En el primer caso si un modo de auto incremento o decremento se comienza a usar, la dirección que es de dos bytes de longitud, será apuntada por el registro indexado. Un incremento o decremento en uno no es posible, porque el registro indexado estará apuntando al byte bajo de la dirección y no tendrá la dirección completa del operando.

El set completo de instrucciones del 6809 es presentado en el apéndice de hojas técnicas.

I.c Interrupciones.

Las interrupciones son entradas que el MPU examina como parte de cada ciclo de instrucción, y provocan que el MPU reaccione a eventos asíncronos más eficientemente que el empleo de dispositivos muestreadores. El uso de interrupciones generalmente provoca que se empleé más hardware, pero las interrupciones logran una respuesta más rápida y directa.

¿ Cuándo se emplean interrupciones ?

Las interrupciones se pueden emplear cuando se requiera activar una alarma, desconectar la energía eléctrica o periféricos que estén listos para aceptar o mandar datos y deben recibir la atención inmediata del MPU.

Características de los Sistemas de Interrupción

La implementación de un sistema de interrupciones tiene muchas variaciones.

Las preguntas que sobre un sistema en particular se hacen son:

1) ¿ Cuántas entradas de interrupción se tienen ?
2) ¿ Cuándo el MPU responde a una interrupción ?
3) ¿ Cómo determina el MPU la fuente de la interrupción, si el número de fuentes excede el número de entradas de interrupción?
4) ¿ Puede el MPU determinar si una interrupción es o no importante ?
5) ¿ Cómo y cuándo se habilita o se deshabilita un sistema de interrupciones ?

Existen muchas respuestas a estas preguntas. El objetivo de todas las implementaciones, son hacer que el MPU responda rápidamente a las interrupciones, y posteriormente regrese a su actividad normal.

El número de entradas de interrupción que tenga el circuito integrado, determinan la cantidad de respuestas que el MPU puede producir. Con ayuda de hardware y software adicionales, cada entrada puede producir una respuesta interna diferente.

Desafortunadamente casi todos los microprocesadores tienen una

cantidad pequeña de ellas (una o dos regularmente). La respuesta correcta del MPU a una interrupción, es la de transferir el control a una rutina de interrupción y salvar el contenido del contador de programa, así como los registros.

El MPU ejecuta un salto a una instrucción que inicializa el servicio a la rutina de interrupción. Esta acción salva la dirección de regreso y transfiere el control a la rutina de servicio de interrupción, la cantidad de hardware requerida para lograr esta respuesta varía considerablemente. Muchos MPU generan internamente esta instrucción y dirección. El MPU puede solamente generar una diferente instrucción o dirección para cada entrada de interrupción.

Búsqueda (polling) del Vector

Si el número de dispositivos que pueden interrumpir exceden al número de entradas de interrupción, el MPU necesitará de hardware o software extra para poder localizar la fuente que interrumpió.

En este caso, el software puede tener una rutina de búsqueda que checa el dispositivo que pudo realizar la interrupción.

El empleo de la búsqueda provoca que el MPU reconozca cuando menos a un dispositivo activo. Una solución alternativa es la de emplear hardware externo que provee de una sola entrada (o vector) para cada fuente. Las dos alternativas pueden mezclarse; el vector puede identificar grupos de entradas donde el MPU pueda reconocer una en particular con la búsqueda.

Prioridad

Un sistema de interrupciones que puede diferenciar entre una interrupción que es importante de otra que no lo es, se le conoce como sistema prioritario de interrupciones.

Hardware interno puede proveer de niveles prioritarios a las entradas de interrupción. Con hardware externo se puede proveer de más niveles. El hardware externo no debe permitir que la interrupción alcance al MPU a menos que la prioridad de está sea muy alta.

Un sistema prioritario de interrupciones necesita manejar de manera especial a las interrupciones de baja prioridad, ya que éstas pueden ser ignoradas por largos períodos de tiempo.

Habilitar y Deshabilitar

Mucho sistemas de interrupción pueden habilitarse y deshabilitarse. En efecto, muchos MPU automáticamente deshabilitan las interrupciones cuando ocurre un RESET (ya que la rutina de encendido debe de inicializar el sistema) y no aceptar ninguna interrupción. Una interrupción puede no ser desactivada (llamada interrupción no enmascarada) cuyo empleo es el de advertir de una falla, y obviamente su importancia está sobre muchas otras actividades

Desventajas de las Interrupciones

Las ventajas de las interrupciones son muchas, pero existen desventajas, estas son:

1) Los sistemas de interrupción requieren de una gran cantidad de hardware.
2) Las interrupciones requieren datos que son transferidos por medio del MPU. Esto no es una ventaja en cuanto a velocidad, como lo es en el DMA.
3) Las interrupciones ocurren en cualquier momento, y su localización puede ser difícil, los errores pueden ocurrir esporádicamente.
4) Las interrupciones involucran a muchos registros,los cuales deben ser salvados y casi todas las fuentes de interrupciones son encontradas por medio del muestreo

Sistemas de Interrupciones del 6809

La respuesta interna del MPU 6809 a una interrupción es moderadamente compleja. El sistema de interrupciones consiste de:

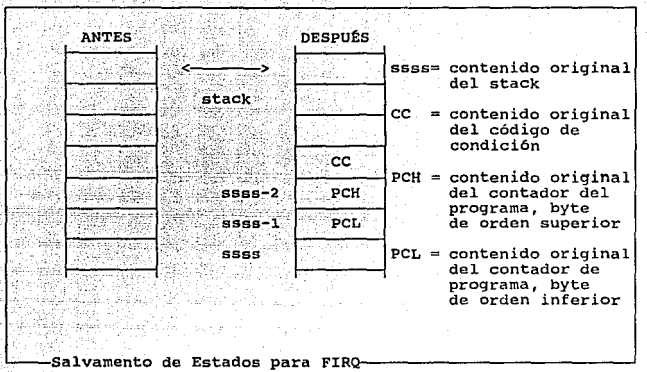
1) tres interrupciones que se activan con un nivel bajo, las dos primeras son interrupciones enmascarables IRQ y FIRQ. La tercera es no enmascarable (NMI).
2) Se desactivan bits para cada una de las interrupciones enmascaradas (IRQ Y FIRQ) si una de las interrupciones está deshabilitada. El bit de IRQ (o bandera I) es el bit 4 del código de condición; el bit de FIRQ (o bandera F) es

el bit 6 del código de condición. La bandera E (bit 7 del código de condición) distingue entre una interrupción FIRQ de otra interrupción. Como es de esperarse el procesador desactiva ambos bits I y F, cuando ocurre una interrupción. Una vez determinada la rutina de interrupción el programa debe inicializar el sistema, para poder atender nuevas interrupciones.

Respuesta del 6809 a las Interrupciones

El 6809 revisa el estado en que se encuentra el sistema de interrupciones, al final de cada instrucción. Si una de las entradas de interrupción del 6809 está habilitada, la respuesta es:

1) El MPU deshabilita la interrupción mascarable IRQ esto es, activa el bit 4 (la bandera I) del código de condición, si la entrada que se activó es FIRQ o NMI, el MPU deshabilita FIRQ, esto es, activa el bit 6 (la bandera F) del código de condición.
2) Si el MPU no está ejecutando CWAY o SYNC desactiva la bandera E en respuesta a FIRQ o la activa en caso de que ejecute CWAI o SYNC.
3) El MPU salva al contador de programa, y el estado del código de condición si la entrada es FIRQ o todos los registros empleados o cualquier otra entrada o instrucción que se esté ejecutando en el stack, la siguiente figura muestra el orden en que los estados se salvan después de que se reconoce la interrupción FIRQ.



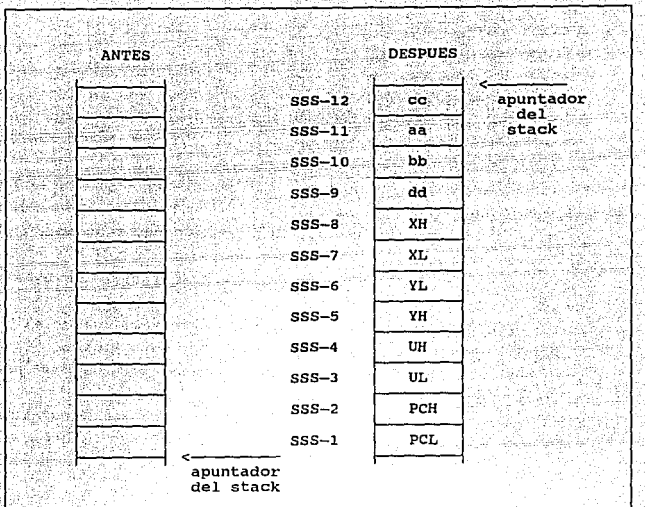
Salvamento de Estados para FIRQ

4) El MPU direcciona a una localidad específica de la memoria, y carga el contenido de esa dirección dentro del contador de programa. En la siguiente tabla se listan las localidades asignadas para varias entradas y para las instrucciones SWI.

MAPA DE MEMORIA PARA LOCALIZACIÓN POR VECTOR :		DESCRIPCIÓN DEL VECTOR DE INTERRUPCIÓN:
MSBS	LSBS	
FFFE	FFFF	RESET
FFFC	FFFD	NMI
FFFA	FFFB	SWI
FFF8	FFF9	IRQ
FFF6	FFF7	FIRQ
FFF4	FFF5	SW12
FFF2	FFF3	SW13
FFF0	FFF1	Reservada

Localidades para varias Entradas y SWI

La siguiente figura muestra el orden en que son salvados los estados después de otras entradas de interrupción, la bandera E distingue entre dos alternativas: 1 si todos los estados han sido salvados y 0 si sólo el límite inferior de los estados han sido salvados.



SSS= Contenido original del apuntador de stack.
cc = Contenido del estado del código de condición para el bit 7, lo activa a 1.
aa = Contenido del acumulador A.
bb = Contenido del acumulador B.
dd = Contenido de la dirección de la página de los registros.
XH = Registro índice X byte superior.
XL = Registro índice X byte inferior.
YH = Registro índice Y byte superior.
YL = Registro índice Y byte inferior.
UH = Apuntador del stack U byte superior.
UL = Apuntador del stack U byte inferior.
PCH= Contenido original del contador de programa byte superior.
PCL= Contenido original del contador de programa byte inferior.

Salvamentos para otras Entradas de Interrupción

Características Especiales

Características del sistema de interrupciones para el 6809

1) El 6809 salva automáticamente los estados del procesador en el stack. El contador de programa es siempre salvado, si la interrupción se ha realizado, al igual que el registro de códigos de condición incluyendo las banderas de interrupciones enmascaradas y las banderas de las interrupciones rápidas enmascaradas.
2) La solicitud rápida de interrupción no solamente provee de una segunda interrupción, sino que permite salvar el contador de programa y el registro de códigos de condición; reduciendo así el tiempo de respuesta a 9 ciclos de reloj.
   Un ciclo de reloj es necesario para transferir cada byte al stack.
3) El 6809 provee de señales de hardware externas, usando el bus disponible y el bus de líneas de estado, para indicar que es aceptada una interrupción. Estas líneas son usadas para activar algún dispositivo externo.
4) El 6809 no contiene internamente una manera para determinar el origen de una interrupción, cuando existen muchas fuentes, las cuales sobrepasen sus entradas de interrupción.

Otro tipo de Interrupciones

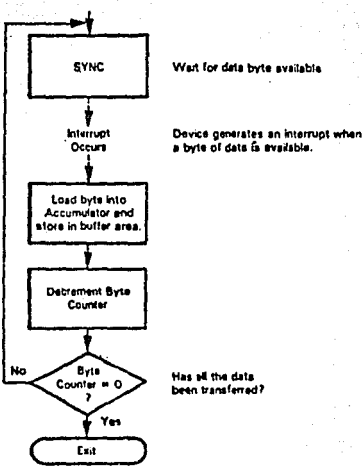
Las siguientes instrucciones especiales sirven para manejar el sistema de interrupciones del 6809

ANDCC # % 11101111 ó CLI Limpia el bit 4 del código de condición y de esta manera permite manejar una interrupción de IRQ.
ANDCC # % 10111111 ó CFL Limpia el bit 6 del código de condición y de esta manera permite el acceso a una interrupción rápida.
ANDCC # % 10101111 ó CLIF Habilita ambas interrupciones.
ORCC # % 00010000 ó SEI Habilita el bit 4 del código de condición y de ésta forma incapacita las interrupciones normales.
ORCC # % 01000000 ó SEF Activa el bit 6 del código de condición deshabilitando la interrupción rápida
ORCC # % 01010000 ó SEIF Deshabilita ambas interrupciones a la vez.

CWAI (deshabilita y espera una interrupción) A través de una lógica de compuertas "AND", provee un bit de dato junto con el código de condición, usualmente habilita las interrupciones normales y rápidas, y salva todos los registros empleados en el stack, y espera la llegada de una interrupción, la respuesta a ésta es rápida (9 ciclos de reloj) desde que los registros han sido salvados. Si una instrucción CWAI ha sido ejecutada y una interrupción rápida ocurre, el MPU comienza la rutina de la interrupción con todos los registros salvados (con la bandera E activa en el stack).

SWI (interrupción por software) Activa la bandera E, salva todos los registros en el stack y deshabilita las interrupciones normales y rápidas. Las instrucciones de SWI producen casi exactamente la misma respuesta que una señal de interrupción (de aquí el nombre). La única diferencia es la localidad de la cual el MPU obtiene el nuevo valor del contador de programa. Las instrucciones SWI son útiles para depurar programas, y para regresar el control al monitor o al sistema operativo mientras que simultáneamente salva el estado actual en el stack. La instrucción SWI es también conocida como trampa, ya que se puede usar para atrapar el MPU en rutinas especiales en caso de errores de hardware o algún otro evento poco usual. SWI se utiliza comúnmente en monitores y sistemas operativos para transferir el control del usuario al sistema; el SWI2 está supuestamente disponible para fines del usuario y entonces no debe ser usado en software de sistemas cerrados (packaged).

SYNC (sincroniza un evento externo) Causa que el procesador no ejecute instrucciones. El MPU simplemente espera una interrupción, si es mascarada o menor de tres ciclos de reloj, el MPU continúa con la siguiente instrucción del programa principal o realiza una rutina de servicio a una interrupción. SYNC es extremadamente rápido (de una alta prioridad) simplemente no emplea ningún vector ni programa. Obviamente el uso de SYNC es de aproximadamente de un ciclo. El MPU no salva el estado presente ni identifica la fuente. La siguiente figura muestra el empleo de la instrucción SYNC.



Uso de SYNC

Interrupción Rápida

La interrupción rápida es una interrupción enmascarada de alta prioridad, en respuesta a ésta, el MPU limpia la bandera E, salva el contador de programa y el código de condición en el stack (si se asume que no se está ejecutando una instrucción WAI). Así, obtiene un nuevo valor del contador de programa a partir de las direcciones de memoria FFF6 y FFF7. Las diferencias entre las interrupciones normales y rápidas son mínimas desde el punto de vista del programador.

Interrupciones No Enmascaradas

Las interrupciones no enmascaradas tienen una entrada de nivel sensitivo. El procesador por lo tanto sólo reacciona en el límite de un pulso en esta línea, y el pulso no interrumpirá su propia rutina de servicio.

Las interrupciones no enmascaradas son útiles para aplicaciones que deben responder a una pérdida de energía (usualmente salvando los datos en una memoria de bajo poder o cambiando a una batería de soporte). Se encuentran aplicaciones típicas en equipos de comunicación que deben retener códigos y mensajes parcialmente recibidos y equipos de prueba que requieren seguir la ruta de pruebas parcialmente completas.

RTI (regreso después de una interrupción) Restaura los registros que estaban almacenados en el stack cuando se termina una rutina de interrupción. Si la bandera E se encuentra deshabilitada, al ser recuperada, RTI recupera solamente los estados del registro de código de condición y el contador de programa. De ésta forma RTI es similar a RTS, sólo que RTI restaura otros registros así como el contador de programa.

Interrupciones del 6820 PIA

La mayoría de los sistemas de interrupción del 6809 involucran una PIA 6820. Cada puerto de la PIA tiene las siguientes características para usar con interrupciones.

1) Una salida del interruptor de baja velocidad.
2) Bits habilitadores de la interrupción (el bit 0 del registro de control para la línea de control 1 , el bit 3 para la línea de control 2 si ésta es una entrada)
3) Bits del estado de interrupción (el bit 7 del registro de control para la línea de control 1, el bit 6 para la línea de control 2).

Los bits 1 (línea de control 1) y 4 (línea de control 2) determinan si es que se levanta el nivel (transición alto/bajo) en la línea de control y se causa una interrupción.

1) El PIA tiene bits que habilitan la interrupción mientras que el microprocesador tiene banderas de enmascaramiento de interrupción. Esto es, los bits del PIA deben ser "1" para permitir interrupciones, mientras que las banderas del MPU deben ser " 0 " para tener el mismo efecto.
2) RESET Limpia el registro de control de la PIA y entonces deshabilita todas las interrupciones. Aún si las salidas de interrupción están unidas a NMI en el MPU, no ocurrirán interrupciones hasta que los bits habilitadores de la PIA estén activados.
3) El MPU puede verificar los bits 6 y 7 de registro de control para ver si un PIA tiene una interrupción pendiente.
4) La PIA recordará una interrupción que ocurra mientras que las interrupciones están inhabilitadas y dará una salida tan pronto como el bit habilitador esté listo.

Interrupciones del 6850 ACIA

El ACIA 6850 también puede producir interrupciones. Se deben hacer notar las siguientes características del ACIA 6850 en sistemas basados en interrupciones.

1) La interrupción de transmisión, significa que el ACIA está listo para dar datos y estará habilitado sólo si el registro de control tiene el bit 6 = 0 y el bit 5 = 1
2) La interrupción de recepción indica que el ACIA ha recibido nuevos datos y estará capacitado sólo si en el registro de control el bit 7 = 1
3) El reset maestro no afecta a los bits que habilitan la interrupción
4) La ocurrencia de cada interrupción prepara el bit 7 del registro de estado. Cada lectura de datos del ACIA ó escritura de datos sobre el ACIA limpia el bit 7

Sistema del 6809 de Búsqueda de Interrupciones

La mayoría de los sistemas de interrupciones del 6809 deben buscar cada PIA y ACIA para determinar cual causó una interrupción. El método de búsqueda es:

1) Verifica cada PIA examinando los bits 6 y 7 del registro de control

```
        LDA     PIACRA      ¿ está listo el bit 7 ?
        BPL     INTRP1      si, ha ocurrido la interrupción 1
        ASLA                ¿ está listo el bit 6?
        BMI     INTRP2      ha ocurrido la interrupción 2
```

2) Checar cada ACIA examinando el bit 7 del registro de estado

```
        LDA     ACIASR      ¿ está listo el bit 7 ?
        BPL     NXTCHK      no, no hay interrupciones en este
                            ACIA
        LSRA                si, ¿ el bit 0 está listo ?
        BCS     RCCUINT     si, se recibió una interrupción
        BRA     TXINT       no, debe haber sido una
                            interrupción del transmisor.
```

El bit 7 del ACIA del registro de estado indica que ha ocurrido una interrupción en la recepción o transmisión. El bit 0 se habilitará si hay una interrupción en el receptor y el bit 1 se habilitará si existe interrupción en el transmisor, la interrupción debe ser una o la otra, así nuestro programa asume que la interrupción es en el transmisor si no la encuentra en el receptor.

Las características importantes del sistema de búsqueda de interrupción del 6809 son:

1) El orden en el cuál los bits de estado son examinados determina la prioridad de las interrupciones. Obviamente el MPU no irá más allá si se encuentra una interrupción activa así ignora la actividad de fuentes posteriores de la secuencia del programa principal.
   Las prioridades son fáciles de establecer, solamente seleccionando el orden en que se van a examinar, pero difíciles para cambiar este orden.
2) La rutina de servicio debe de limpiar una interrupción de la PIA leyendo el registro de datos correspondiente, aún si el puerto se está usando para salidas o si no es necesaria la transferencia de datos. De otra manera, la interrupción se

activará.El programa puede usar el TST (prueba cero o menos) para leer el registro de datos de la PIA sin cambiar su contenido o los contenidos del registro de usuario.

Desventaja de las Interrupciones por Búsqueda

Las rutinas de búsqueda son adecuadas si el número de fuentes es pequeño y la frecuencia de interrupciones es grande.

Las rutinas de búsqueda son lentas y difíciles por las siguientes razones:

1) El número promedio de operaciones de búsqueda se incrementa linealmente con el número de entradas de interrupción. En promedio por supuesto, una rutina de búsqueda tendrá que examinar la mitad de entradas antes de encontrar una activa. Se puede reducir el número promedio de búsqueda de alguna manera, checando las entradas más frecuentes en primer lugar.
2) En las asignaciones PIA y ACIA rara vez son consecutivas o con la misma periodicidad, por lo tanto, se necesitan instrucciones separadas para examinar cada entrada. Las rutinas de búsqueda son entonces difíciles de extender. Se pueden utilizar tablas de asignación I/O, con una vía de acceso a los modos de asignación indicados.
3) Las interrupciones que se buscan primero pueden esconder a las que se buscan después, a menos que se varíe el orden de búsqueda. Sin embargo, variar el orden de búsqueda es difícil ya que las asignaciones no son consecutivas.

Sistema de Interrupciones por Vector del 6809

El problema de búsqueda en sistemas basados en el 6809 es típicamente resuelto por métodos especiales, con una aplicación particular o MPU, el controlador de prioridad de interrupciones del 6828 provee de un sistema vectorial de interrupciones de 8 niveles basado en entradas de interrupciones regulares. Este aditamento simplemente reconoce las asignaciones FFF8 y FFF9 (ver tabla de localidades para varias entradas y SWI) cuando aparecen en el bus de asignaciones y las reemplaza por uno de los 8 vectores, se puede utilizar también el hardware especial el cuál reconoce la señal de interrupción dada por el microprocesador 6809.

Comunicaciones entre el Programa Principal y las Rutinas de Servicio

Uno de los principales problemas al escribir programas para sistemas basados en interrupciones es el de proveer de comunicación entre el programa principal y las rutinas de servicio. Los criterios para los métodos de comunicación son:

1) No deben interferir con la ejecución normal del programa principal.
2) No deben depender del modo de operación del programa principal. Por ejemplo, no deben depender de la inactividad del programa principal (ejecutando CWAI o SYNC) o asumir que ciertos registros están siempre disponibles.
3) Deben estar bien definidos y ser capaz de manipular varias cantidades de datos.
4) No deben requerir la acción instantánea del programa principal.

Mientras más paciente es el sistema más fácil será su desarrollo y mantenimiento. La idea es hacer a las rutinas de servicio del programa principal transparentes entre si. Esto permite al programador cambiar una sin afectar a la otra. También ayuda a limitar los errores en una u otra.

Reconocimiento de Software (Software Handshake)

Un acercamiento simple para la comunicación es un reconocimiento de software. El proveedor de datos (la rutina de interrupción para entradas o el programa principal para salidas) dispone de una bandera para indicar que hay nuevos datos disponibles. El receptor de datos puede examinar la bandera y puede limpiarla después de transferir o aceptar los datos. El receptor puede a su vez disponer de otra bandera (un reconocimiento) para indicar que los datos más recientes fueron procesados y se pueden enviar más.

¿ En dónde se pueden poner las banderas y datos ? Una forma sencilla es utilizar un espacio en la memoria para cada bandera y para los datos. La localización puede ser una asignación específica en la memoria o una asignación en el stack del hardware que ha sido preparada para dicho propósito. El programa principal y las rutinas

de servicio se pueden comunicar a través de tales espacios, de manera similar a como el procesador se comunica con los dispositivos de I/O a través de los puertos.

Interrupciones Amortiguadas (Bufeered Interrups)

En el método señalado arriba se asume el manejo de I/O en una base byte a byte. El procesador debe proveer cada salida de byte por separado y debe manipular cada entrada de byte aparte.

Es claro que todas las operaciones deben proceder a una tasa que garantice ser lo suficientemente rápida para evitar la pérdida de datos, como en el I/O normal, se pueden relajar las restricciones de tiempo usando amortiguadores (buffers). Con éste método la rutina de servicio transfiere los datos I/O de un buffer y actualiza el señalador del buffer para la siguiente operación. El único tiempo que le debe importar al programa es cuando los buffer de entrada están llenos o cuando los buffer de salida están vacíos. En las rutinas de servicio actúan como implementos I/O que tienen su propia memoria local en donde los datos pueden ser almacenados temporalmente. A esta aproximación se le conoce como interrupciones amortiguadas.

Doble Amortiguamiento (Double Buffering)

De hecho se puede extender este método. Se puede tener un buffer para la rutina de servicio y otro para el programa principal. Aún el llenado o vaciado de un buffer no crea problemas mientras el otro esté disponible. Las identidades de cada buffer pueden ser intercambiadas cuando la rutina de servicio a llenado o vaciado su buffer. Este método se conoce como doble amortiguamiento. Permite la interrupción I/O en casi total independencia del programa principal.

Habilitando y Deshabilitando las Interrupciones

Otro problema al escribir programas para sistemas basados en interrupciones es decidir cuando habilitar o inhabilitar interrupciones.

Cuando Inhabilitar Interrupciones

1) Durante la inicialización del sistema de interrupciones. Esto puede involucrar en llenar los valores iniciales en señaladores (pointers), banderas y contadores, o determinando un orden inicial para búsqueda u otras operaciones. Se debe recordar que el RESET deshabilita automáticamente las interrupciones del MPU y PIA así que la rutina de inicio del sistema tendrá que habilitarlos de manera explícita.

2) Durante la utilización o servicio de una interrupción, si la interrupción no es habilitada por lo menos hasta que se termine esta, la computadora entra en un loop interminable con la interrupción. Continuamente interrumpiendo su propia rutina. Se debe recordar que el microprocesador deshabilita automáticamente las interrupciones regulares como parte de su respuesta normal. Se debe notar también que una interrupción NMI no interviene con su propia rutina de servicio, ya que la entrada es limite sensitivo (edge-sensitive) más que un nivel sensitivo (level-sensitive).

3) Durante operaciones que ocurren en tiempos real (tales como retrasos en loops I/O sincrónicos de alta velocidad) pueden producir resultados erróneos si son interrumpidas.

Cuando Habilitar Interrupciones

Se deben habilitar interrupciones lo más pronto posible cuando éstas puedan ocurrir. De otra manera el sistema podría perder una interrupción, ya sea una entrada o no dar la salida adecuada.

Inicializando el Sistema de Interrupciones

El orden normal en el cual se inicializa un sistema 6809 basado en interrupciones es el siguiente (comenzando desde RESET).

1) inicializar todos los parámetros del sistema.
2) habilitar interrupciones de cada PIA y ACIA.
3) habilitar interrupciones de MPU eliminando las banderas enmascaradas.

Interrupciones PIA

Si se desea inhabilitar una interrupción en particular, se puede hacer independientemente de otras interrupciones eliminando la bandera de habilitación de interrupción para un puerto específico. Podemos hacer esto sin afectar otros bits del registro de control usando operaciones lógicas.

1) Inhabilitando una interrupción PIA
-línea de control 1

```
        LDA     PIACR
        ANDA    #%11111110    inhabilitar interrupción de la
                              línea de control 1
        STA     PIACR
```

ó si se sabe que la interrupción está actualmente habilitada

```
        DEC     PIACR         inhabilitar interrupción de la
                              línea de control 2
```

-línea de control 2

```
        LDA     PIACR
        ANDA    #%11110111    inhabilitar interrupción de la
                              línea de control 2
        STA     PIACR
```

2) habilitando una interrupción PIA
-línea de control 1

```
        LDA     PIACR
        ORA     #%00000001     habilitar interrupción de la
                               línea de control 1
        STA     PIACR
```

ó si se sabe que esta interrupción está actualmente inhabilitada

```
        INC     PIACR         habilitar interrupción de la
                              línea de control 1
```

```
-línea de control 2

        LDA     PIACR
        ORA     #%00001000    habilitar interrupción de la
                              línea de control 2
        STA     PIACR
```

Las instrucciones INC y DEC toman ventaja del hecho de que el bit 0 es el habilitador de interrupción para la línea de control 1. Sin embargo estas instrucciones pueden afectar el registro de control del PIA por completo si son deliberadamente ejecutados cuando el habilitador de interrupciones se encuentra ya en el estado deseado.

Salvando y Almacenando Estados de Interrupción

Un problema relacionado es el almacenar el estado original del sistema de interrupción después de realizar las operaciones que se requieren para inhabilitar la interrupción.

La solución es simple: Salvar y almacenar el registro de código de condición que contiene los bits de la interrupción mascarable. Podemos salvar el registro después de deshabilitar las interrupciones con la instrucción PSHS CC; podemos restablecer el registro luego de regresar el control al programa principal ó realizando operaciones que puedan interrumpirse con la instrucción PULS CC.

Cambiando los Valores en el Stack

El microprocesador 6809 automáticamente salva todos o algunos de sus registros en respuesta a una interrupción, la instrucción RTI al final de una rutina de servicio restaura dichos registros. La mayoría de las rutinas de servicio dejan los registros en el stack para promover generalidad y simplicidad. Sin embargo, los programadores ocasionalmente encuentran necesario alterar algunos de los registros, las razones típicas son las de forzar a regresar a una asignación (address) diferente ó para inhabilitar la entrada del sistema de interrupción. En estos casos el programador debe saber como encontrar los registros en el hardware de stack.

La tabla siguiente contiene los offset-indexados que se requieren

para tener acceso a los registros en el caso de que el procesador haya salvado el estado completo.

REGISTROS	
Código de condición	00
Acumulador A	01
Acumulador B	02
Registro directo de página	03
Byte superior del índice del registro X	04
Byte inferior del índice del registro X	05
Byte superior del índice del registro Y	06
Byte inferior del índice del registro Y	07
Byte superior del stack del usuario U	08
Byte inferior del stack del usuario U	09
Byte superior del contador de programa	0A
Byte inferior del contador de programa	0B

Offsets Indexados

En la tabla siguiente se muestran los offset indexados, en respuesta a FIRQ, que se requieren cuando el procesador a salvado sólo el contador de programa y el código de condición.

REGISTROS	
Código de condición	01
Byte superior del contador de programa	02
Byte inferior del contador de programa	03

Offsets Indexados en Respuesta a FIRQ

Las rutinas típicas al usar los offset en ésta tabla son:

1) Cambiando la asignación de dirección a EEXIT (una rutina de error de salida)

```
LDX     #EEXIT  asignación de regreso = error de
                salida
STX     $0A,S
```

2) Disminuyendo la asignación de regreso en 1 (usada en caso de que una interrupción o instrucción SWI ha sido usada para reemplazar una instrucción actual del programa para depurar o para propósito de prueba).

```
        TST     $0B,S       ¿Son LSB's de asignación de
                            regreso a cero?
        BNE     DECLSB      Si, reduce el MSB's en 1
        DEC     $0A,S
DECLSB  DEC     $0B,S       Reducir LSB's de asignación de
                            regreso en 1
```

Solo si los LSB's de las asignaciones de regreso son cero es necesario disminuir los MSB's para producir una disminución correcta de los 16 bits.

3) Inhabilitando la interrupción rápida poniendo el bit de interrupción enmascarada (bit 4 del código de condición).

```
        LDA     ,S
        ORA   # % 00010000   inhabilitar interrupción regular
        STA     ,S
```

4) Inhabilitando la interrupción rápida, activando el bit de interrupción rápida enmascarada (bit 6 del código de condición)

```
LDA     ,S
ORA   # % 01000000   inhabilitar interrupción rápida
STA     ,S
```

Interrupción Sobrepasada (Interrupt Overhead)

Obviamente se debe ser extremadamente cuidadoso cuando se alteren los valores del stack, ya que dichos cambios pueden haber tenido efectos no previstos en el programa principal.

Al responder a una interrupción se hacen necesario ciclos demás "overhead", ya que el MPU puede buscar un nuevo valor del contador de programa y salvar registros en el stack. Si se utiliza un procesador de tiempo, se puede determinar la cantidad de sobrepaso involucrado en servir a interrupciones a partir de los requerimientos de tiempo de la siguiente tabla.

OPERACION	NUMERO DE CICLOS DE R E L O J
Respuesta normal a IRQ o NM	21
Respuesta normal a FIRQ	12
Respuesta a cualquier interrupción mientras se ejecuta CWAI	9
Escape hacia SYNC si una interrupción es deshabilitada	1
Ejecución de CWAI	20
Ejecución de RTI con la bandera E	15
Ejecución de RTI sin la bandera E	16
Ejecución de SWI	19
Ejecución de SWI2 o SWI3	20
Ejecución de SYNC	2

Requerimientos de Tiempo

Tiempo Real de Reloj en una Interrupción

Reloj de tiempo real. Este tipo de relojes simplemente dan una serie regular de pulsos. El intervalo entre los pulsos puede ser usado como una referencia de tiempo y las interrupciones pueden ser

usadas en cualquier múltiplo del intervalo básico de tiempo. Se debe hacer notar que los cambios involucrados en determinar la frecuencia del tiempo real de reloj (por ejemplo 10 kHz) permitan la creación de un amplio intervalo de tiempo de gran precisión. Por otro lado, el sobrepaso involucrado en conteo de las interrupciones de estos relojes, puede ser considerable, y el conteo excederá la capacidad de un sólo registro de 8 bits o una localidad de memoria. La elección de una frecuencia depende de la precisión requerida en la aplicación. El reloj por supuesto forma parte del hardware, un contador puede hacer pulsos de alta frecuencia e interrumpir al procesador sólo ocasionalmente, el programa tendrá que leer al contador para medir el tiempo con gran precisión. Un problema es sincronizar las operaciones con el reloj real. Esto tendrá algún efecto en la precisión del intervalo de sí el MPU comienza la medición al azar durante algún período, más que al comienzo. Algunas formas de sincronizar las operaciones son:

1) Iniciar juntos MPU y reloj. RESET o interrupciones de inicio pueden inicializar el reloj y el MPU.
2) Permitir el comienzo del MPU y detener el reloj bajo el control del programa.
3) Usar un reloj de alta frecuencia para que los errores en los períodos sean pequeños.
4) Alinear el reloj (esperando una interrupción) antes de comenzar la medición.

Una interrupción del reloj real tiene muy alta prioridad, ya que su precisión puede ser afectada por cualquier retraso en las interrupciones. La práctica usual es hacer que el reloj real tenga la más alta prioridad en interrupciones excepto en casos de fallas en la energía de alimentación. La rutina de servicio de reloj es muy corta para que no interfiera en otras actividades del MPU.

I d.- Descripción de las Funciones de cada una de las Terminales.

El 6809 es un circuito integrado de 40 alfileres (pines), se explicara ahora la función de cada una de sus terminales.

Las señales del 6809 se pueden dividir dentro de cuatro grupos.

Señales de Reloj

Son el primer grupo las señales de reloj (E, Q, XTAL Y EXTAL), las que sincronizan el proceso del MPU.

Entradas de Cristal XTAL y EXTAL (Terminales 39 y 38 respectivamente) Estas entradas son usadas para conectar el oscilador interno a un cristal externo.

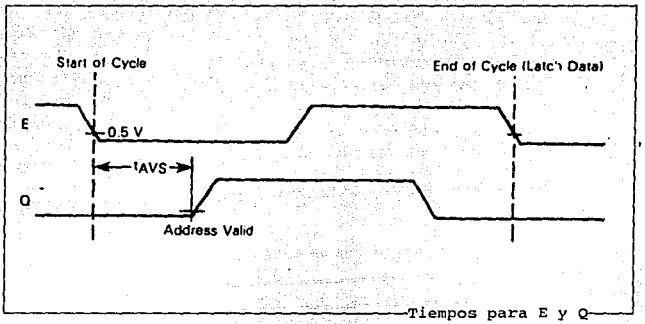
Alternativamente la terminal EXTAL puede usarse como entrada de niveles TTL de un generador de reloj externo, esto se logra conectando a XTAL a tierra.

Se debe de recordar que el cristal o el generador de reloj conectado a estas terminales es cuatro veces la frecuencia de operación del bus.

Salidas de Reloj E y Q (Terminal 34 y 35 respectivamente) E y Q son las señales de reloj requeridas por los dispositivos periféricos compatibles con el MPU para su funcionamiento, en especial, para su sincronización. La frecuencia de operación del reloj deseada deberá ser de un cuarto de la frecuencia del cristal con el que se alimente, la frecuencia de éste debe ser mayor a 0.1 MHz y menor o igual a 4 MHz.

La señal Q estara adelantada con respecto a E, esto es, una transición en Q (Q por cuadratura) debe ser seguida de una transición en E (E por habilitación) después de un retraso mínimo, estas señales son empleadas para indicar que las señales en el bus de datos y direcciones son válidas.

Una dirección del MPU es válida un tiempo t_{AQ} después de un frente de onda alto de Q, y un dato es tomado (latcheado) por el bus en el frente de onda bajo de E. Las características de señales de reloj se muestran a continuación:



Tiempos para E y Q

Control del Bus

El segundo grupo lo forman las señales de control del bus (DMA/BREQ, MRDY, BS Y BA).

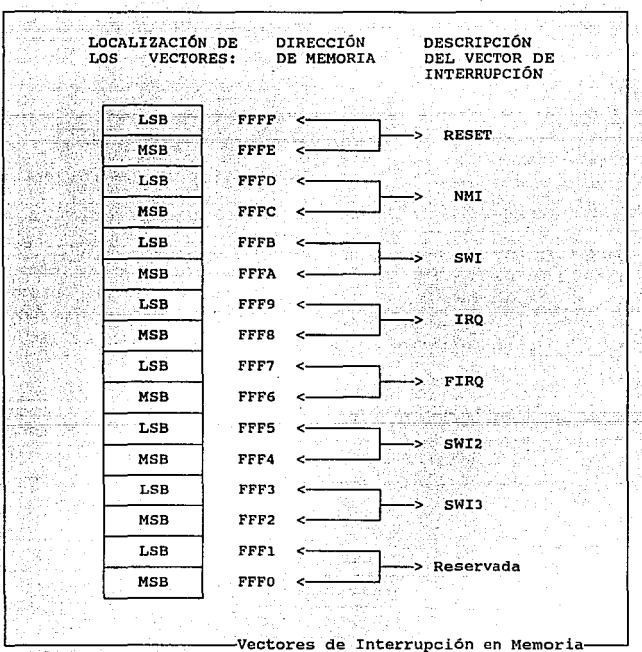
Habilitación del Bus [BA] (Terminal 6 activa alta) Es la indicadora de un control interno, el cual obliga a los buses MOS del MPU a ponerse en un estado de alta impedancia. Cuando BA retorna al nivel bajo, transcurre un ciclo muerto hasta que el MPU toma el bus. No puede ser declarada una señal BA si TSC esta activada.

Señal BS [BS] (Terminal 5 activa alta) Cuando se decodifica junto con BA representa el estado del MPU (válido en un frente de onda bajo de Q). El estado del micro se define según la siguiente tabla.

Estado del MPU		Definición del estado del MPU
BA	BS	
0	0	Normal (Corriendo)
0	1	Reconocimiento de interrupción o reset
1	0	Reconocimiento de sincronía
1	1	Reconocimiento de HALT o Petición del Bus

El reconocimiento de interrupción es indicado en los ciclos de inicialización de un vector de búsqueda de hardware (RESET, NMI FIRQ, IRQ, SWI, SWI2, SWI3). Esta señal habilita la decodificación de las cuatro líneas más bajas de direcciones, y puede proporcionar al usuario una indicación del nivel de interrupción que se esta atendiendo.

Los vectores de direcciones de las atenciones a interrupciones se muestran en la siguiente tabla:



Vectores de Interrupción en Memoria

El reconocimiento de la sincronía es indicada cuando el MPU está esperando por una sincronización externa de una línea de interrupción.

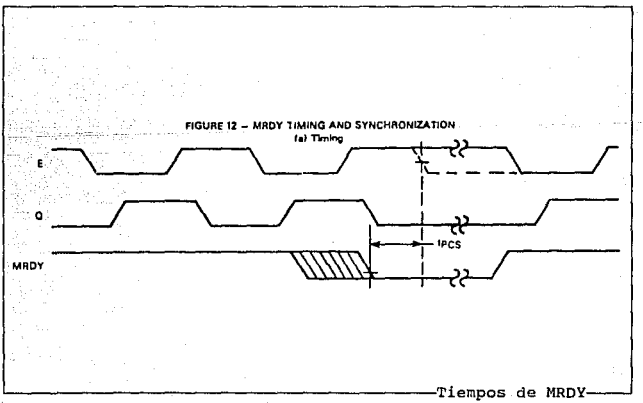
El reconocimiento de "halt" o espera, es indicado cuando el MPU está en una condición de espera (halt), en tanto que la Petición del Bus se indica cuando se ha realizado una transferencia del control del bus.

Señal de Control MRDY (Terminal 36 activa baja) Ésta señal de control permite el ensanchamiento, o mejor dicho, el alargamiento de las señales Q y E para extender el tiempo de acceso de datos.

E y Q operan normalmente mientras MRDY es alta. Cuando MRDY es nivel bajo, E y Q se alargan en múltiplos íntegros de cuartos (1/4) de ciclo de bus, estos permiten interface con memorias lentas.

Durante accesos no validos a memoria, la señal MRDY no tiene ningún efecto sobre Q y E, esto impide que el MPU aminore el proceso durante accesos no validos.

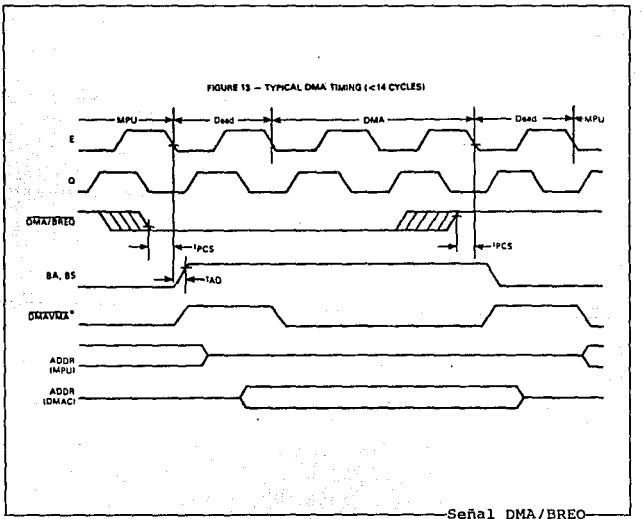
El siguiente diagrama muestra el comportamiento de MRDY:



Tiempos de MRDY

Acceso Directo a Memoria [DMA/BREQ] (Terminal 33 activa baja) La entrada DMA/BREQ provee un método de suspender la ejecución y adquirir el bus del MPU para otro uso, los usos típicos son funciones DMA y el refrescamiento de memorias dinámicas.

Un nivel bajo en esta terminal detendrá la ejecución de la instrucción al final del actual ciclo a menos que el MPU necesite de un pre-recuperado para auto-refrescamiento. El MPU reconoce la señal proveniente de esta terminal poniendo las terminales BS y BA en nivel alto. El dispositivo solicitado no podrá poseer el bus por más de 15 ciclos de éste, antes que el MPU recobre el bus para un auto-refrescamiento. El auto-refrescamiento requiere un ciclo de bus con el flanco de subida y bajada de un ciclo muerto. Un ejemplo del uso se muestra en la siguiente figura y en seguida el comportamiento de un auto-refrescamiento.

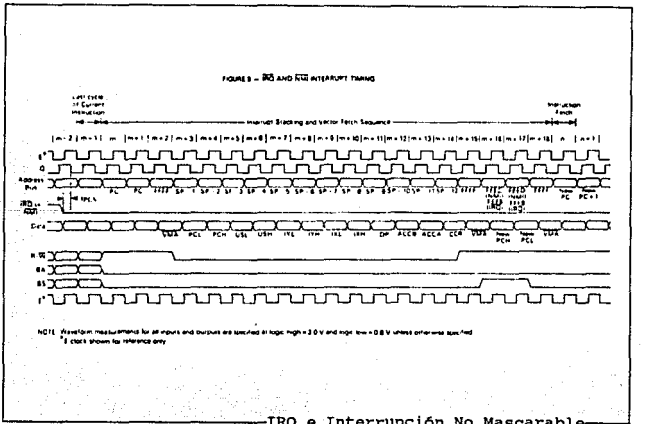


Señal DMA/BREQ

Control del MPU

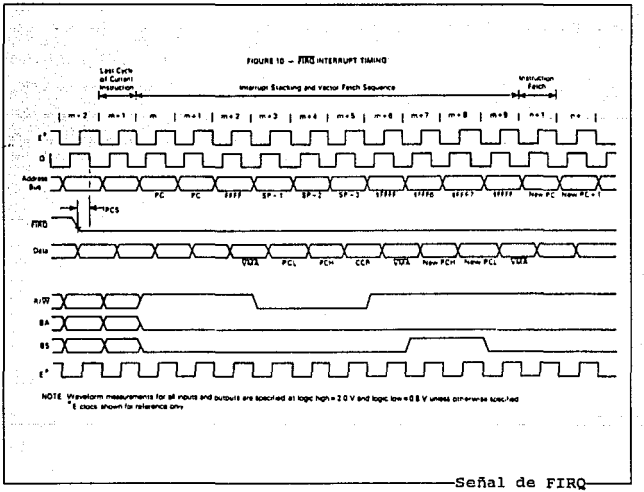
El siguiente grupo lo forman las señales de control del MPU (NMI, RESET, IRQ, FIRQ, y HALT). Todas estas señales de entrada interfieren en el estado o secuencia interna del MPU, cada señal realiza una función específica, con lo que tenemos:

Interrupción No Mascarable [NMI] (Terminal 2 activa baja) Si se presenta en esta terminal una transición negativa, es decir, un flanco a bajo, provoca la generación de la secuencia de atención a dicha interrupción (subrutina de atención a interrupción). Una interrupción no mascarable no puede ser deshabilitada y posee además una alta prioridad por arriba de FIRQ, IRQ e interrupciones de software. Durante el reconocimiento de la interrupción el estado total del MPU es almacenado en el stack, esto es, que todos los registros internos del micro son cargados al stack. Después del RESET se reconocerá una señal baja en NMI hasta la primera carga del programa del stack pointer (S). Esta señal se muestra en la gráfica.



IRQ e Interrupción No Mascarable

Petición Rápida de Interrupción [FIRQ] (Terminal 4 activa baja) Un nivel bajo en esta línea inicializa la secuencia de atención a dicha interrupción, dicha secuencia consiste en llamar a la subrutina que atiende a FIRQ, obligando que el bit de máscara (F) del registro de código de condición (CC) sea limpiado. Esta secuencia tiene mayor prioridad que IRQ y es más rápida en el sentido de que sólo almacena en el stack el contenido de CC y el del contador de programa (PC). Esta señal se observa en el diagrama de tiempos siguiente.

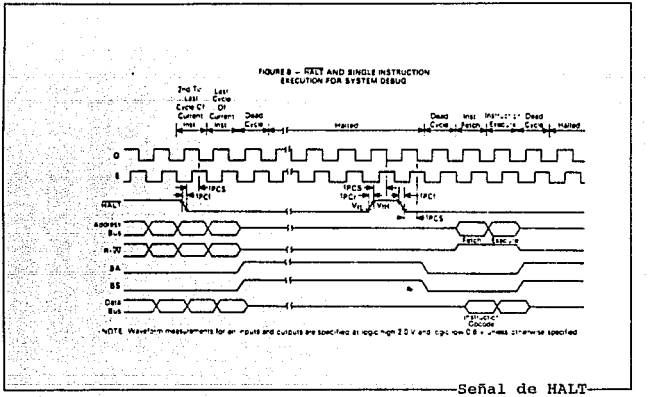


Señal de FIRQ

Señal de HALT (Terminal 40 activa baja) La presentación de un nivel bajo en esta entrada provoca un alto en la ejecución del programa en curso al finalizar la presente instrucción, y lo mantiene en este estado (detenido) indefinidamente, esto es sin la pérdida de datos.

Cuando se detiene la señal de BA es puesta en un nivel alto indicando que los buses están en alta impedancia, mientras que la señal BS está también en alto indicando un estado de Halt (alto) en el MPU. Mientras que el microprocesador permanece detenido por la afirmación de la señal de "halt" o por la instrucción de "halt", no responde al proceso externo en tiempo real de las peticiones de interrupciones, por lo que son almacenadas para ser atendidas posteriormente.

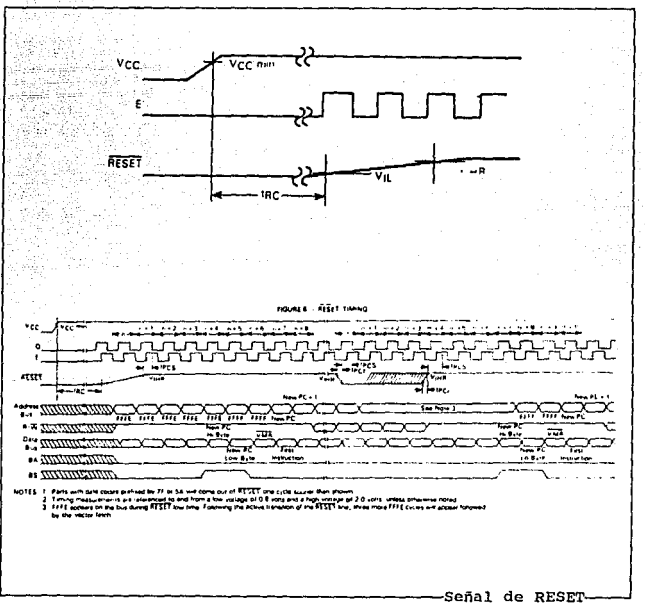
El diagrama de tiempos se muestra a continuación:



Señal de HALT

Petición de Interrupción [IRQ] (Terminal 3 activa baja) Un nivel bajo en esta línea provoca la generación de la secuencia de atención. En dicha secuencia el bit de máscara (I) del registro CC es limpiado; debido a que una interrupción IRQ almacena en el stack el estado total del microprocesador es mucho más lenta que FIRQ, teniendo menor prioridad que ésta. Tanto FIRQ como IRQ tienen que limpiar, en su rutina de servicio, la fuente de interrupción antes de hacer un regreso de interrupción (RTI). El diagrama de tiempos se muestra con el de NMI.

Señal de RESET (Terminal 37 activa baja) Esta conectada a una compuerta Schmitt-trigger, un nivel bajo en dicha entrada por más de un ciclo reestablece o reinicia al MPU. El vector de reset es cargado de las direcciones $FFFE_H$ y $FFFF_H$ cuando un reconocimiento de interrupción es válido (BA*BS=1). Durante el encendido del sistema, la línea de reset puede ser tomada como baja hasta que la señal de reloj se estabiliza. Debido a que el 6809 posee en esta línea una compuerta Schmitt-trigger con umbral de voltaje alto, mayor que los estandares de los periféricos, se asegura que éstos salgan del estado de RESET antes que el MPU. El diagrama de tiempos de la señal de RESET se muestra en seguida:

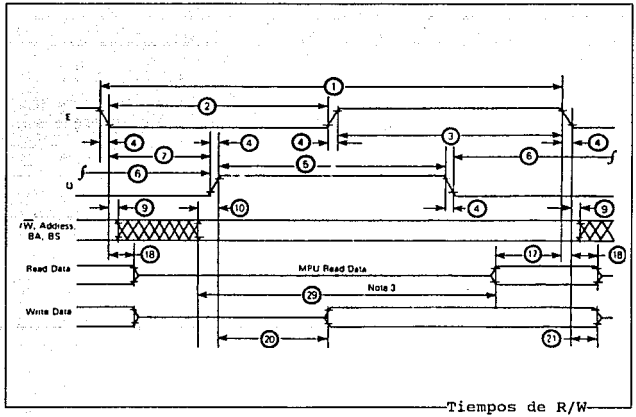


Señal de RESET

Señal de Lectura/Escritura y Buses de Datos y Direcciones

El último grupo lo forma la señal R/W y los buses.

Señal de Lectura/Escritura [R/W] (Terminal 32) Indica la dirección del dato a transferir dentro del bus de datos. Un nivel bajo indica que el MPU esta escribiendo un dato a un dispositivo externo, mientras que un nivel alto indica que se toma un dato del bus. La salida R/W es puesta en alta impedancia cuando BA es alta o cuando TSC es declarada. El comportamiento de esta señal se muestra en el siguiente diagrama:



Tiempos de R/W

Bus de Datos [D_0-D_7] (terminales 31 a 24), estas terminales son líneas bidireccionales, y son las encargadas de proporcionar la comunicación del MPU con el sistema bidireccional del bus de datos. Como las líneas de direcciones pueden manejar una compuerta Schottky o cuatro LS.

Bus de Direcciones [A_0-A_{15}] (terminales 8 a 23), que consta de 16 terminales unidireccionales, las cuales son usadas para direccionar información del MPU a los periféricos conectados al bus de direcciones, dicha información puede ser de entrada o salida. Cuando el MPU no utiliza el bus para transferir un dato, estas direccionan a $FFFF_H$, R/W=1 y BS=0, esto es lo que se denomina un acceso no válido (Dummy access). Todas las líneas del bus de direcciones son puestas en alta impedancia cuando BA es alto. Cada línea puede manejar una compuerta TTL tipo Schottky o cuatro LS.

Otras de las líneas que posee el 6809 son las de V_{CC} (terminal 7) y V_{DD} (terminal 1) éstas son empleadas para la fuente de alimentación. V_{DD} es tierra o 0 V, mientras que V_{CC} es 5 V ± 5%.

Para un adecuado funcionamiento del MPU, todas las señales deben de cubrir ciertos requisitos de tiempos, ya sea de duración o de tiempo de retraso, algunos de estos requisitos, que aparecen en algunos ciclos de tiempo mostrados anteriormente, se encuentran especificados en la siguiente tabla:

BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

Ident. Number	Characteristics	Symbol	MC6809		MC68A09		MC68B09		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time (See Note 5)	t_{cyc}	1.0	10	0.667	10	0.5	10	µs
2	Pulse Width, E Low	PW_{EL}	430	5000	280	5000	210	5000	ns
3	Pulse Width, E High	PW_{EH}	450	15600	280	15700	220	15700	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
5	Pulse Width, Q High	PW_{QH}	430	5000	280	5000	210	5000	ns
6	Pulse Width, Q Low	PW_{QL}	450	15600	280	15700	220	15700	ns
7	Delay Time, E to Q Rise	t_{AVS}	200	250	130	166	80	125	ns
9	Address Hold Time* (See Note 4)	t_{AH}	20	–	20	–	20	–	ns
10	BA, BS, R/W, and Address Valid Time to Q Rise	t_{AQ}	50	–	25	–	15	–	ns
17	Read Data Setup Time	t_{DSR}	80	–	60	–	40	–	ns
18	Read Data Hold Time*	t_{DHR}	10	–	10	–	10	–	ns
20	Data Delay Time from Q	t_{DDQ}	–	200	–	140	–	110	ns
21	Write Data Hold Time*	t_{DHW}	30	–	30	–	30	–	ns
29	Usable Access Time (See Note 3)	t_{ACC}	695	–	440	–	330	–	ns
	Processor Control Setup Time (MRDY, Interrupts, DMA/BREQ, HALT, RESET) (Figures 6, 8, 9, 10, 12, and 13)	t_{PCS}	200	–	140	–	110	–	ns
	Crystal Oscillator Start Time (Figures 6 and 7)	t_{RC}	–	100	–	100	–	100	ms
	Processor Control Rise and Fall Time (Figures 6 and 8)	t_{PCr}, t_{PCf}	–	100	–	100	–	100	ns

Características de Tiempos

De la anterior tabla se puede observar que un pulso de RESET, HALT, FIRQ, IRQ, etc, debe de durar mínimo 200 ns para ser reconocido.

El tiempo de un ciclo de reloj tanto para la señal de Q como de E debe ser de 1 μs mínimo y con ½ ciclo de trabajo.

El tiempo de acceso en la lectura es de 695 ns como mínimo, con un tiempo de captura de datos de 80 ns adicionales, es decir, se toma 695 ns, en mantener la dirección y solamente 80 ns en tomar el dato de la localidad direccionada. El tiempo de escritura tiene un retraso máximo de 200 ns después del flanco de subida de la señal Q para que se de el tiempo de escritura (durante este retraso se direcciona la localidad en la que se almacenará el dato) el cuál termina mínimo 30 ns después de concluido el ciclo de E.

I.e Consideraciones Eléctricas para el Buen Manejo del 6809

Este dispositivo cuenta con circuitería para protección de altos voltajes estáticos que lo puedan dañar en las entradas. Sin embargo se recomienda tomar precauciones y cuidar que sólo llegen los niveles máximos de voltaje permitidos; sobre todo cuando el circuito está en un estado de alta impedancia.

Características	Símbolo	Valor	Un
Alimentación de Voltaje	Vcc	-0.3 a +7.0	V
Entrada de Voltaje	Vin	-0.3 a +7.0	V
Rango de Temperatura de Operación	T	T a T L H	
MC6809, MC68A09, MC68B09		0 a +70	C
MC6809C, MC68A09C, MC68B09C		-40 a +84	
Rango de temperatura de Almacenamiento	T stg	-55 a +150	C

Rangos Máximos

De la tabla anterior podemos observar lo siguiente:

a) La alimentación de voltaje al MPU debe ser dentro del rango de -0.3 a +7.0 siendo el indicado +5 V. Cuando demos un reset o al ocurrir una falla en la alimentación de voltaje, este puede alcanzar como valor mínimo -0.3 V. Para cumplir con esta especificación, así como con la alimentación de los demás chips, nuestra fuente ya sea diseñada por nosotros o adquirida en el mercado, deberá cumplir estos parámetros, como también una protección para que el máximo voltaje que alimente al MPU no rebase +7 V.

b) El rango mínimo de la temperatura de operación es de 0°C y el máximo de 70° C. Al realizar una aplicación en la industria deberemos cuidar que el sistema opere en estos rangos de temperatura. En caso de no ser así, por ejemplo, si la T_H es mayor a 70° C, podemos usar un ventilador para descender la temperatura de operación al circular el aire más próximo.

Características	Símbolo	Valor	Unidad
Resistencia Térmica			
Cerámico		50	
Cerdip	θ_{jA}	60	C / W
Plástico		100	

Características Térmicas

En la tabla anterior podemos leer el valor de θ_{jA} que junto con las ecuaciones siguientes, nos permiten calcular para diferentes temperaturas ambientales (TA) la potencia disipada P_D .

La temperatura promedio de la juntura del chip T_j en °C puede obtenerse de:

$$T_j = T_A + (P_D \times \theta_{jA}) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

donde:

T_A = Temperatura ambiente °C

θ_{jA} = Resistencia térmica del encapsulado entre la juntura y el ambiente °C / W .

$P_D = P_{INT} + P_{PORT}$

$P_{INT} = I_{CC} \times V_{CC}$ Watts, Potencia Interna del Chip

P_{PORT} = Watts, determinada por el usuario

Para muchas aplicaciones $P_{ORT} < P_{INT}$ y puede no ser tomado en cuenta. Una relación aproximada entre P_D y T_J es:

$$P_D = K + (T_J + 273°C) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2)$$

Resolviendo las ecuaciones (1) y (2) para K tenemos:

$$K = P_D \times (T_A + 273°C) + \theta_{JA} \times P_D^2 \ldots\ldots\ldots\ldots\ldots \quad (3)$$

Donde K es una constante particular de cada chip y puede ser determinada por la ecuación (3) para medidas de P_D (en equilibrio) y una T_A conocida.

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	- 0.3 to + 7.0	V
Input Voltage	V_{in}	- 0.3 to + 7.0	V
Operating Temperature Range MC6809, MC68A09, MC68B09 MC6809C, MC68A09C, MC68B09C	T_A	T_L to T_H 0 to + 70 - 40 to + 85	°C
Storage Temperature Range	T_{stg}	- 55 to + 150	°C

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Ceramic Cerdip Plastic	θ_{JA}	 50 60 100	°C/W

Características Eléctricas de DC

En la tabla anterior podemos observar lo siguiente:

Los voltajes para el reloj en su entrada Q son compatibles con la lógica TTL, y deben ser de $V_{IHMIN} = V_{SS} + 2.0$ [V] y $V_{IHMAX} = V_{CC}$; y en la entrada E, niveles compatibles con la tecnología MOS, los valores de voltaje deben ser de $V_{IHC} = V_{CC} - 0.75$ [V] a $V_{IHCMAX} = V_{CC} + 0.3$ [V] ,mientras que para la señal de RESET tenemos $V_{IHRMIN} = V_{SS} + 4.0$ y $V_{IHRMAX} = V_{CC}$.

Para los niveles de entrada bajo en Q tenemos $V_{ILQMIN} = V_{SS} - 0.3$ y $V_{ILQMAX} = V_{SS} + 0.6$ [V]. Si $V_{SS} = 0$ [V] el mayor voltaje de entrada bajo es 0.6 [V]. Para la entrada E, el rango es de $V_{ILCMIN} = V_{SS} - 0.3$ a $V_{ILCMAX} = V_{SS} + 0.4$. Si $V_{SS} = 0$, $V_{ILCMIN} = -0.3$ y $V_{ILCMAX} = 0.4$, y para el RESET, $V_{ILRMIN} = V_{SS} - 0.3$ y $V_{ILRMAX} = V_{SS} + 0.8$ Si $V_{SS} = 0$, $V_{ILMAX} = 0.8$ por lo tanto este deberá ser el voltaje que al activar el circuito de RESET nos proporcione.

La corriente de entrada para el rango de $V_{IN} = 0$ a 5.25 [V] y $V_{CC} = V_{MAX}$ en Q y RESET será $I_{IN} = 2.5$ μA, y para la entrada E tenemos $I_{IN} = 100$ μA. Estas corrientes deben ser proporcionadas por los circuitos de RESET y de reloj.

Salidas de voltaje alto de DC, para $V_{SS} = 0$

datos	: $D_0 - D_7$	; $V_{OH} = 2.4$
direcciones	: $A_0 - A_{15}$ R/W	; $V_{OH} = 2.4$
control	: BA, BS, LIC, AVMA, SUSY	; $V_{OH} = 2.4$

estos son los niveles de salida alto, en valor mínimo que pueden ser considerados como válidos.

Salida de voltaje bajo de DC, $V_{OL} = 0.5$ [V]

En conclusión podemos decir que los voltajes que se hayan entre 0.5 y 2.4 [V] estarán en un estado lógico indefinido.

La disipación interna de potencia medida con T_A = 0°C, es de 1 W. Con este valor obtenemos la solución de las ecuaciones anteriores para cualquier valor T_A.

La frecuencia de operación del 6809 es de 2 MHz lo cuál nos determina el valor del CK y del cristal a usar.

En el estado de alta impedancia, la corriente de entrada I_{TSI} para los datos D_0 - D_7, las direcciones A_0-A_{15} y R/W es de 100 μA como máximo.

CAPITULO

II

PERIFERICOS PARA EL

MC 6809

II

Periféricos para el MC-6809

II.a Características y Selección de Memorias y Buffers.

Selección de memorias y buffers

Existen dos problemas para el diseño de rutinas de entrada salida (I/O): una es como está la interfase entre los periféricos hacia la computadora para la transferencia de datos, estados, y señales de control: la otra es como las direcciones de los dispositivos de I/O pueden ser seleccionados (uno en particular) para la transferencia de datos.

En teoría, la transferencia de datos hacia o de cualquier dispositivo de I/O es similar a la transferencia de datos que se realizan en una memoria, en efecto se puede considerar a la memoria

como otro dispositivo de I/O. La memoria tiene grandes ventajas las razones son:

1) Las operaciones son realizadas a la misma velocidad que el procesador
2) Usa el mismo tipo de señales que el procesador, los circuitos que necesita para hacer la interfase con el MPU son los mismos que manejan la memoria, y también con los mismos niveles de transición
3) No se requiere una formato especial para las señales de control de los pulsos de lectura/escritura.
4) Automáticamente guarda los datos.
5) La longitud de palabra es la usada por el microprocesador.

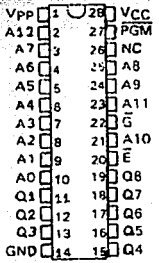
ROM (2764)

En los sistemas con microprocesador se necesita una gran cantidad de memoria que guarde información, la cual no se pierda cuando la energía del sistema deba quitarse. Este tipo de memoria es llamada Memoria de solo lectura (ROM). La información en la ROM puede leerse pero no alterarse. La ROM es usada en los sistemas porque permite al MPU inicializar a todos los periféricos cuando se enciende el sistema.

La secuencia de eventos para leer datos de una ROM es la siguiente:

Cuando una dirección activa a una ROM, proveniente del MPU, provoca que la memoria localice un dato el cual se leerá, como existen miles de datos almacenados, las líneas de dirección seleccionan solo un dato por cada línea de salida. Entonces el MPU espera una cantidad finita de tiempo, llamada tiempo de acceso, de esta manera permite a la memoria decodificar la dirección para esta entrada, y dar salida a el dato. El chip select se activa para habilitar los datos de salida, hacia el bus de datos. En este momento los datos provenientes de la ROM que están presentes en el bus de datos, y en las líneas de entrada del MPU, son leídos por el microprocesador. Finalmente el MPU carga los datos dentro de sus registros internos. El chip select (CS) o habilitador del integrado se deshabilita para remover los datos de la ROM que están presentes en el bus de datos.

El diagrama de tiempos de lectura/escritura se da en los siguientes cuadros:



Patigrama del ROM 2764

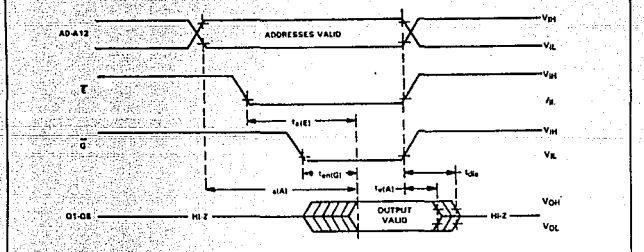


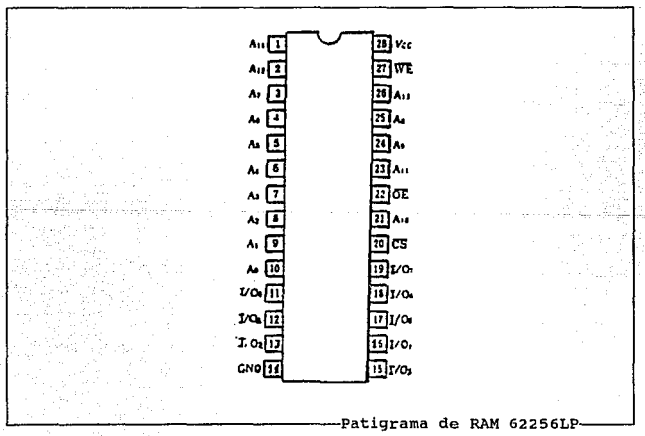
Diagrama de lectura del ROM

RAM (62256LP)

Secuencia de eventos para una lectura de la RAM

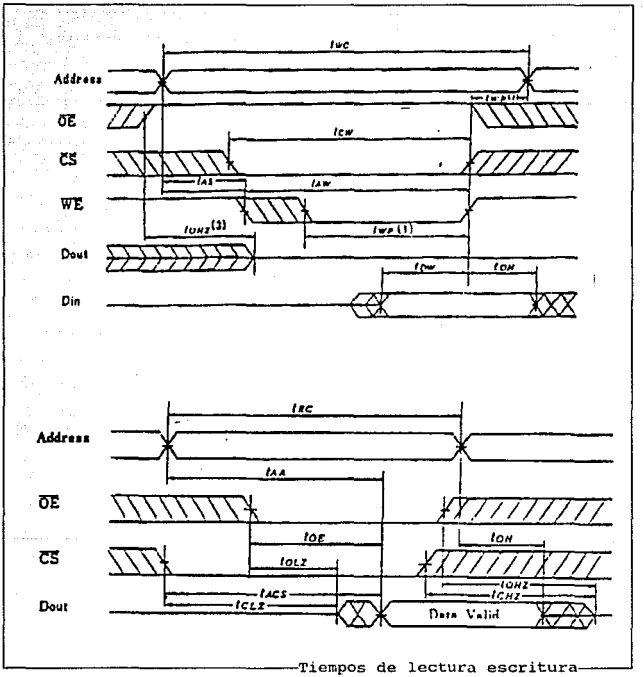
1) Primero, las líneas de dirección llegan a la entrada de la memoria. En este momento la localización del dato que se va a leer, es por medio de un decodificador que tiene la RAM.
2) Las líneas de control de R/W debe de estar en la posición de lectura.
3) El sistema debe de esperar un cierto tiempo para la lectura, llamado tiempo de acceso de lectura, que permiten a los circuitos internos de la memoria decodificar la dirección y seleccionar el dato que se leerá.
4) Después del tiempo de espera, el dato está en las líneas de salida de la memoria, y en este momento el microprocesador puede leerlo. Si el microprocesador lee el dato demasiado rápido, esto es, si no espera para el tiempo de acceso de lectura, datos erróneos pueden ser transferidos por la memoria.

El patigrama de la memoria elegida es el siguiente.



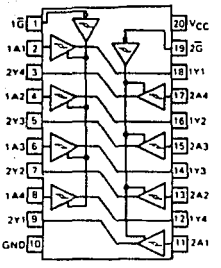
Patigrama de RAM 62256LP

Los tiempos de lectura/escritura de la RAM se dan a continuación:



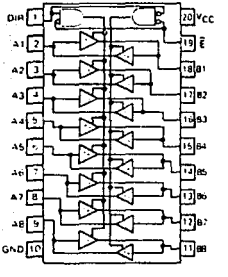
Tiempos de lectura escritura

Los buffers escogidos son, el 74LS244 para las direcciones, con una capacidad de corriente máxima de 13 [mA], su diagrama de conexiones es el siguiente:



74LS244 (bus de direcciones)

y el 74LS245 empleado para los datos.



74LS245 (bus de datos)

Tiempos de lectura y escritura del 6809

El microprocesador cuenta con ocho líneas de datos bidireccionales (D_0-D_7), y con 16 líneas para el bus de direcciones (A_0-A_{15}).

La corriente suministrada en las salidas de datos y de direcciones del MPU es baja para capacitar adecuadamente el sistema de buses del sistema, lo cual hace necesario el empleo de buffers, estos buffers son capases de suministrar mayor corriente, además deben ser de tres estados debido a que el MPU puede en algún momento ser deshabilitado por un dispositivo externo y dejar libres los buses de datos y de direcciones.

Los tiempos de acceso que emplea el 6809 para transmitir y recibir datos son los que se muestran en la siguiente gráfica de tiempos.

En dicha gráfica se pude observar la correspondencia de tiempos de lectura escritura del 6809 con respecto a las memorias elegidas.

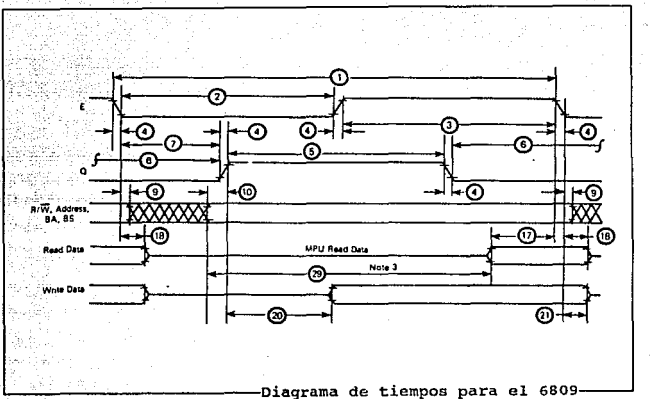


Diagrama de tiempos para el 6809

II. b Características y Funcionamiento del ACIA MC-6850.

Introducción

El MC6850 ACIA (Asyncronos communocations interfase adapter) es un adaptador para interfase de comunicación asíncrona, el cual provee el formato de datos y control para una interfase de comunicación de información serial a un sistema de bus organizado como el del microprocesador MC6800.

La interfase o conexión del bus del 6850 incluye: líneas de selección, de habilitación, de lectura/escritura, interrupción y una interfase lógica de bus que permite sobretransferir datos en un bus de 8 bits bidireccional. El dato en paralelo del sistema, es transmitido o recibido en serie por el ACIA mediante la interfaz de datos asíncrona, con un formato propio y un chequeo de errores. La configuración funcional del ACIA es programada vía el bus de datos durante la etapa de inicialización del sistema. En este registro de control programable se provee la posibilidad de variar la longitud de la palabra de transmisión, la división de los ciclos de reloj, el control de transmisión y recepción y el control de interrupciones, como se explicará más adelante. Para interacción con periféricos o modems, el ACIA posee tres líneas de control. Estas permiten al ACIA una conexión directa con el MC6860L modem digital de 0-600 bps. Un diagrama de bloques representativo de la organización interna del ACIA se muestra en la siguiente figura.

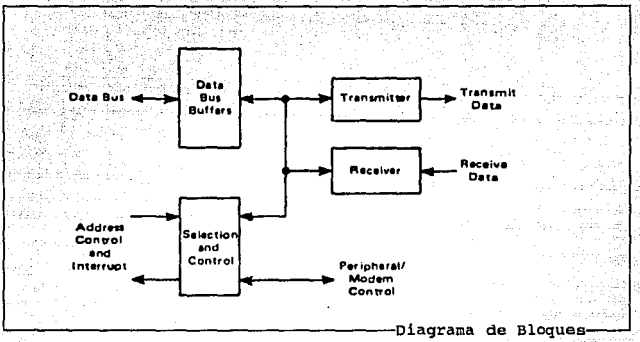
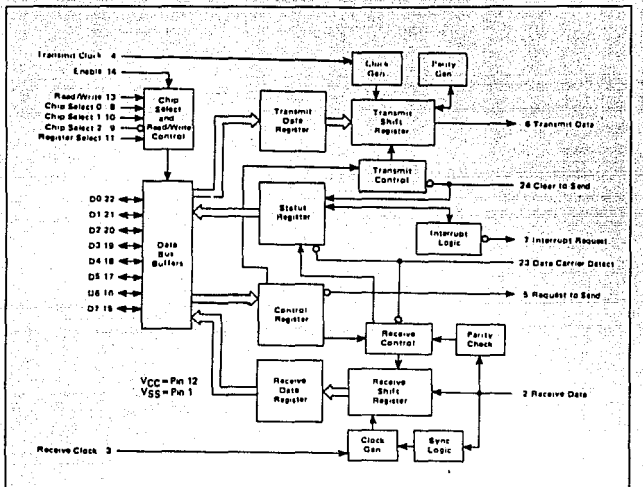


Diagrama de Bloques

Entre las características del MC6850 se encuentran:

* Transmisión de 8 o 9 bits.
* La opción de paridad par o non.
* Paridad, desborde y chequeo de error en el formato.
* Registro de control programable.
* La opción de modos de reloj ÷1, ÷16 y ÷64.
* Transmisión arriba de 1 Mbps.
* Suspensión de un bit de inicio falso.
* Funciones de control para periféricos o modems.
* Doble almacenaje.
* Operación con 1 ó 2 bits de parada.

Un diagrama más completo de la organización interna del ACIA se muestra en la siguiente figura, la cual corresponde a un diagrama de bloques expandido.



Organización Interna del ACIA

Este dispositivo contiene circuiteria para protección en entradas frente a daños debidos a altos voltajes estáticos y/o campos eléctricos; sin embargo es aconsejable tomar las precauciones necesarias para evitar la aplicación de cualquier alto voltaje a este circuito. Una formalidad en la operación es: si no son usadas algunas señales de entrada del dispositivo, el conectarlas a un nivel lógico de voltaje apropiado (p.e. V_{ss} o V_{cc}).

Operación del dispositivo.

En la interfase del bus, el ACIA presenta dos localidades de memoria direccionables. Internamente, son cuatro registros: dos registros de sólo-lectura y dos de sólo-escritura. Los registros de sólo-lectura son: el de Estados y el de Recepción de Dato; los registros de sólo-escritura son: el de Control y el de Transmisión

de Dato. La interfaz consiste de líneas de entrada y salida de datos en serie con relojes independientes, y tres líneas de control para periféricos o modems.

Registros del ACIA.

El ACIA como se mencionó anteriormente cuenta con 4 registros internos. El diagrama de bloques expandido del ACIA muestra estos registros, éstos son usados para: el estado, el control, la recepción y la transmisión de datos. El contenido de los registros es mostrado en la siguiente tabla.

Data Bus Line Number	Buffer Address RS • R/W Transmit Data Register (Write Only)	RS • R/W Receive Data Register (Read Only)	RS • R/W Control Register (Write Only)	RS • R/W Status Register (Read Only)
0	Data Bit 0*	Data Bit 0	Counter Divide Select 1 (CR0)	Receive Data Register Full (RDRF)
1	Data Bit 1	Data Bit 1	Counter Divide Select 2 (CR1)	Transmit Data Register Empty (TDRE)
2	Data Bit 2	Data Bit 2	Word Select 1 (CR2)	Data Carrier Detect (DCD)
3	Data Bit 3	Data Bit 3	Word Select 2 (CR3)	Clear to Send (CTS)
4	Data Bit 4	Data Bit 4	Word Select 3 (CR4)	Framing Error (FE)
5	Data Bit 5	Data Bit 5	Transmit Control 1 (CR5)	Receiver Overrun (OVRN)
6	Data Bit 6	Data Bit 6	Transmit Control 2 (CR6)	Parity Error (PE)
7	Data Bit 7***	Data Bit 7**	Receive Interrupt Enable (CR7)	Interrupt Request (IRQ)

* Leading bit = LSB = Bit 0
** Data bit will be zero in 7 bit plus parity modes
*** Data bit is "don't care" in 7 bit plus parity modes

Registros del ACIA

Registro de Transmisión de Dato (TDR).

Los datos son escritos en el registro de transmisión de dato durante una transición negativa de la habilitación (E) cuando ha sido direccionado el ACIA con RS alto y R/W bajo. La escritura del dato en el registro provoca que el bit de registro de transmisión de dato vacío en el registro de estado se vaya a nivel bajo, con lo que el dato puede ser transmitido. Si la transmisión es ociosa y un carácter inválido es transmitido entonces la transferencia puede

tomar el lugar (en tiempo) dentro de un bit. Si un carácter empieza a transmitirse, el nuevo carácter seguirá cuando se concluya el anterior. La transferencia del dato provoca que el bit de TDRE indique vacío.

Registro de Recepción de Dato (RDR).

El dato proveniente del receptor "deserializador" (un registro de corrimiento) es transferido automáticamente al registro de recepción de dato (RDR) vacío al recibir un carácter completo. Este evento provoca que el bit de registro de recepción de dato lleno (RDRF) en el buffer de estado vaya a alto (estado lleno). El dato puede entonces leerse a través del bus mediante un direccionamiento al ACIA y la selección del registro de recepción de dato con RS y R/W en estado alto cuando el ACIA esta habilitado. El ciclo de lectura es no destructivo causando el limpiado del bit RDRF aunque el dato es retenido en el RDR, es decir, después de ser leído el dato, la información recibida se mantiene vigente. Se transfiere automáticamente el dato del registro de corrimiento de recepción después que se lleno al registro de dato en donde se almacena y el RDR mantiene válido el contenido con el estado prevaleciente.

Registro de Control.

El registro de control del ACIA consiste de 8 bits en buffers de sólo escritura, estos son seleccionados cuando RS y R/W son bajos. Este registro controla la función de recepción, transmisión, habilitación de interrupciones, y la salida de control petición de transmisión para modems/periféricos. A continuación se detallará la función de los bits de control.

Bits de Selección del Contador Divisor de Reloj (CR0 y CR1) Estos bits determinan la relación de división del reloj utilizada entre la sección de transmisión y recepción del ACIA. Adicionalmente, estos bits son usados para proveer el reinicio maestro del ACIA, el cual limpia el registro de estados (excepto para condiciones externas de CTS y DCD) e inicializa tanto la recepción como la transmisión. El reinicio maestro no afecta a los otros bits del registro de control.

Nota: después de un encendido o una falla/reencendido estos bits deben ser cargados con unos para reiniciar el ACIA.

Después del reinicio la relación de división del reloj puede ser seleccionada. Estos bits de selección de contador proveen las siguientes relaciones de divisiones de reloj:

CR1	CR0	FUNCIÓN
0	0	+1
0	1	+16
1	0	+64
1	1	REINICIO MAESTRO

Selección de División

Bits de Selección del Formato de la Palabra (CR2,CR3 y CR4) Estos bits son usados para seleccionar la longitud de la palabra, paridad, y el número de los bits de parada, tanto para la transmisión como la recepción. El formato de decodificación es como se muestra en la siguiente tabla.

CR4	CR3	CR2	FUNCIÓN
0	0	0	7 BITS+PARIDAD PAR+2 BITS DE PARADA
0	0	1	7 BITS+PARIDAD NON+2 BITS DE PARADA
0	1	0	7 BITS+PARIDAD PAR+1 BIT DE PARADA
0	1	1	7 BITS+PARIDAD NON+1 BIT DE PARADA
1	0	0	8 BITS + 2 BITS DE PARADA
1	0	1	8 BITS + 1 BIT DE PARADA
1	1	0	8 BITS+PARIDAD PAR+1 BIT DE PARADA
1	1	1	8 BITS+PARIDAD NON+1 BIT DE PARADA

Selección de Formato

Bits de Control de Transmisión (CR5 y CR6) Dos bits de control de transmisión son provistos para el control de: la interrupción para la condición de registro de transmisión de dato vacío, la salida de petición de transmisión (RTS) y la transmisión de un nivel o carácter de ruptura (break). Para ello es empleado el siguiente formato de decodificación:

CR6	CR5	FUNCIÓN
0	0	RTS=BAJO, INTERRUP. EN TRANSMISIÓN DESHABILITADA
0	1	RTS=BAJO, INTERRUP. EN TRANSMISIÓN HABILITADA
1	0	RTS=ALTO, INTERRUP. EN TRANSMISIÓN DESHABILITADA
1	1	RTS=BAJO, TRANSMITE UN NIVEL DE RUPTURA POR SALIDA DE TRANSMISIÓN, INTER. EN TRANSMISIÓN DESHABILITADA

Control de Transmisión

Bit de Interrupción en Recepción (CR7) Las siguientes interrupciones pueden ser habilitadas con un nivel alto en la posición del bit 7 del registro de control (CR7); registro de recepción de dato lleno, desborde o una transición de bajo a alto en la línea de detector de portadora de dato (DCD).

Registro de Estado.

La información del estado del ACIA es proporcionada al MPU por la lectura de su registro de estado. Este registro de sólo lectura es direccionado cuando RS es bajo y R/W es alto. La información almacenada en este registro indica el estado del registro de transmisión de dato, error lógico y el estado de las entradas del ACIA provenientes del periférico o modem. Todo esto bajo la siguiente organización:

Bit 0 Registro de recepción de dato lleno [RDRF] (Recive Data Register Full) El bit de registro de recepción de dato lleno indica que el dato recibido a sido transferido al registro recepción de dato. RDRF es limpiado después de una lectura del MPU al registro de recepción de dato o por un reinicio maestro. El estado de registro limpio o vacío indica que el contenido del mismo no es prevaleciente ó actual. La señal de entrada detector de portadora de dato estando en estado alto también causa que RDRF indique vacío.

Bit 1 Registro de transmisión de dato vacío [TDRE] (Transmit Data Register Empty) El bit de registro de trasmisión de dato vacío es puesto en alto indicando que el contenido del registro de

transmisión de dato ha sido transferido y el nuevo dato a transferir puede ser cargado. El estado bajo en este bit indica que el registro está lleno y la transmisión del nuevo carácter no ha empezado desde la escritura del último dato.

Bit 2 Detector de portadora de dato [DCD] (Data Carrier Detect) El bit de detector de portadora de dato estará en alto cuando la entrada DCD proveniente del modem/periférico esté en nivel alto, indicando que la portadora no esta presente. Este bit al cambiar al nivel alto, causa la generación de una petición de interrupción, siempre y cuando la habilitación de interrupción en recepción este declarada (habilitada).

Este bit se mantiene en alto después de que la entrada de DCD retorna a bajo, hasta que es limpiado por la primera lectura del registro de estado y el registro de dato o hasta que ocurre un reinicio maestro. Si la entrada de DCD se mantiene en alto después de las lecturas del registro de estado y el de dato o después de ocurrir un reinicio maestro, la interrupción es limpiada y el bit del estado de DCD se mantendrá en alto, y seguirá el comportamiento de la entrada de DCD.

Bit 3 Listo para transmitir [CTS] (Clear-to-Send) El bit 3 del registro de estado indica la condición de la entrada listo para transmitir proveniente de un modem ó periférico. Un nivel bajo en CTS indica que el modem se encuentra en condiciones para entablar una transmisión. En estado alto, el bit de registro de transmisión de dato vacío es inhibido y un reinicio maestro no afecta el estado del bit listo para transmitir.

Bit 4 Error de disposición ó estructura [FE] (Framing Error) El error de disposición ó estructura indica que el carácter recibido tiene su estructura inapropiada, por ejemplo, ya sea por un bit de inicio y uno de parada, estos son detectados por la falta del primer bit de parada. Este error indica que hay fallas de sincronización, una transmisión defectuosa ó una condición de ruptura.

La bandera de error de estructura es puesta en alto o reiniciada durante el tiempo de la transferencia del dato recibido. Por esto,

este error es indicador de que el tiempo presente desde el principio al fin asociado a este carácter esta disponible.

Bit 5 Desborde en recepción [OVRN] (Reciber Overrun) Desborde es una bandera de error la cual indica que uno ó más caracteres en el flujo de datos se han perdido. Esto es, un número de caracteres fueron recibidos, pero no leídos del registro de recepción de dato (RDR) antes de que el subsecuente carácter fuese recibido y transferido. La condición de desborde se da a la mitad del último bit del segundo carácter recibido sucesivamente sin la ocurrencia de una lectura del RDR.

El desborde no va a ocurrir o marcar en el registro de estado mientras que el carácter anterior al desborde sea leído. El bit RDRF se mantiene confirmado hasta que el desborde es reiniciado.

La sincronización de recepción de caracteres es mantenida en la condición de desborde. La indicación de desborde es reiniciada después de hacer una lectura del registro de recepción de dato o bien por un reinicio maestro.

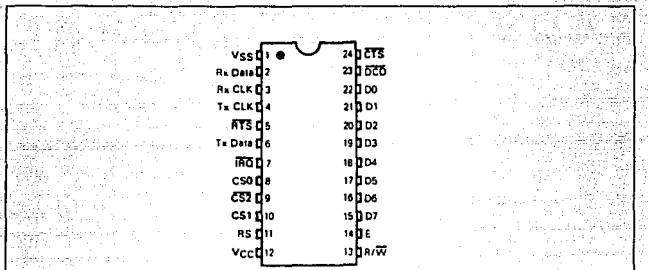
Bit 6 Error en paridad [PE] (Parity Error) La bandera de error en paridad indica que el número de "unos" del carácter leído no concuerda con la paridad par o non preseleccionada. La indicación de error en la paridad esta presente mientras el carácter se mantiene en el RDR.

Si no hay paridad seleccionada, tanto la salida del generador de paridad de transmisión y el resultado del chequeo de paridad de recepción son deshabilitados.

Bit 7 Petición de Interrupción [IRQ] (Interrupt Request) El bit de IRQ indica el estado de la salida de IRQ. Cualquier condición de interrupción, con su respectiva habilitación, puede indicarse en este bit de estado. En cualquier tiempo si la salida IRQ es baja el bit de IRQ será alto indicando la interrupción ó el estado de petición de servicio. IRQ es limpiado por una operación de lectura al registro de recepción de dato ó por una operación de escritura en el registro de transmisión.

Asignación de Terminales.

El 6850 es un circuito de 24 terminales, cada una con una función específica, su disposición de terminales se muestra en el siguiente diagrama:



Disposición de Terminales

A continuación se hará una breve explicación de la función que realiza cada terminal. Estas terminales se pueden dividir en cuatro grupos, estos son:

Funciones de Entrada/Salida.

Señales del ACIA para interfaz con el MPU. La interfaz consta de un bus de datos bidireccional de 8 bits, una línea de lectura/escritura, una línea de petición de interrupción, tres líneas de selección de chip, una línea de selección de registro interno y una línea de habilitación. Estas señales permiten al MPU tener un control completo sobre el ACIA.

Habilitador del ACIA [E](terminal 14) La terminal E, señal de habilitación, es una entrada de alta impedancia compatible con TTL, ésta habilita a los buffers del bus de datos de entrada/salida y al reloj de datos tanto el de entrada como el de salida del ACIA. Esta señal normalmente es derivada del reloj ϕ_2 del MC6800 o del reloj E del MC6809.

Lectura/Escritura [R/W](terminal 13) La línea de lectura/escritura es una entrada de alta impedancia compatible con TTL y es usada para el control de la dirección del flujo de datos a través de la interfaz del bus de datos de entrada/salida del ACIA.

Cuando R/W es alto (el MPU realiza un ciclo de lectura), los manejadores de salida del ACIA son encendidos y un registro interno seleccionado es leído.

Cuando ésta es baja, los manejadores de salida del ACIA son abiertos y el MPU escribe en un registro seleccionado; por lo tanto, la señal de R/W es empleada para seleccionar registros de sólo-escritura o sólo-lectura dentro del ACIA.

Petición de Interrupción [IRQ] (terminal 7 activa baja) Petición de interrupción es una salida con drenador abierto (open-drain, no posee pullup interno), es activa baja y compatible con TTL, ésta se emplea para interrumpir al MPU. La salida IRQ permanece baja mientras que la causa de la interrupción esta presente y la correspondiente habilitación de interrupción del ACIA este activada. El bit de estado de IRQ cuando es alto, indica que la salida correspondiente está en un estado activado o declarado.

Las interrupciones son el resultado de algunas condiciones en la inicialización de las secciones de transmisión y recepción del ACIA. La sección de transmisión causa una interrupción cuando se da la condición de seleccionar la habilitación de interrupción en transmisión (CR5*CR6), y el bit estado del registro de transmisión de dato vacío (Transmit Data Register Empty) [TDRE] es alto. El bit del registro de estado TDRE indica la condición actual del registro de transmisión de dato, excepto cuando lo inhibe listo para transmitir (Clear-to-Send) [CTS] en estado alto o el ACIA se mantiene en una condición de RESET. La interrupción es limpiada por la escritura de un dato dentro del registro de transmisión de dato. La interrupción de transmisión es mascarable por deshabilitación vía CR5 ó CR6 ó por la pérdida de CTS, el cual inhibe el bit de estado de TDRE. La sección de recepción causa una interrupción cuando es activada la habilitación de interrupción de recepción (CR7) y el bit del estado de registro de recepción de dato lleno (Receive Data Register Full) [RDRF] está en estado alto, ha ocurrido un desborde, o el detector de la portadora de dato (Data

Carrier Detect) [DCD] es puesta en alto. Una interrupción proveniente del bit estado del RDRF debida al desborde ó pérdida de portadora puede ser limpiada por una lectura del dato ó por un reinicio del ACIA. Las interrupciones de la recepción pueden ser mascaradas por el reinicio del bit de habilitación de interrupción en la recepción.

Selección del ACIA [CS0, CS1 y CS2] (terminales 8, 10 y 9) Estas tres líneas de entrada de alta impedancia compatible con TTL son empleadas para direccionar al ACIA. El ACIA es direccionado cuando CS0 y CS1 están en alto y CS2 es bajo. Las transferencias de datos hacia el ACIA o proveniente de él son desempeñadas bajo el control de las señales de habilitación (E), lectura/escritura (R/W) y selección de registro.

Selección de Registro [RS] (terminal 11) La línea de selección de registro es una entrada de alta impedancia compatible con TTL. Un nivel alto es usado para seleccionar el registro de Transmisión/Recepción de dato y un nivel bajo para el registro de Control/Estado. La línea de la señal de R/W es empleada en conjunto con la de selección de registro para direccionar registros de sólo-lectura o sólo-escritura en cada pareja de registros.

Bus de datos bidireccional [D_0-D_7] (terminales 22 a 15) Las líneas bidireccionales de datos (D0-D7) permite la transferencia de datos entre el ACIA y el MPU. Las salidas del bus de datos son manejadas por dispositivos tres-estados, éstos las mantienen en estado de alta impedancia (off), excepto cuando el MPU desempeña en el ACIA operaciones de lectura.

Entradas de Reloj.

Entradas de alta impedancia compatibles con TTL son provistas para el reloj de transmisión y recepción de datos por separado. Las frecuencias del reloj pueden ser porciones de 1, 16 ó 64 veces la frecuencia del reloj de alimentación.

Reloj de Recepción [RxCLK] (terminal 3) La entrada del reloj de recepción es usada para sincronizar la recepción del dato. En el modo +1 el dato y el reloj pueden ser sincronizados externamente. El receptor toma muestras del dato en cuanto ocurre una transición a positivo del reloj.

Reloj de Transmisión [TxCLK] (terminal 4) La entrada del reloj de trasmisión es usada para marcar o cronometrar la transmisión de dato. La transmisión del dato inicia cuando ocurre una transición a negativo o nivel bajo del reloj.

Líneas de Entrada/Salida Serial.

Recepción de Dato [RxData] (terminal 2) La línea de recepción de dato es una entrada de alta impedancia compatible con TTL, a través de la cual el dato es recibido en serie con un formato. La sincronización con un reloj para la detección del dato es realizada internamente cuando la porción usada es de 16 ó 64 veces la frecuencia de alimentación.

Transmisión de Dato [TxData] (terminal 6) La línea de transmisión de dato transfiere el dato en serie a un modem u otro periférico con un formato preseleccionado.

Control de Periféricos/Modem.

El ACIA incluye varias funciones que permiten un control limitado sobre un periférico o modem. Las funciones incluidas son: petición para trasmitir, listo para transmitir y detector de la portadora de dato.

Listo para Transmitir [CTS] (terminal 24) Esta terminal (entrada de alta impedancia compatible con TTL), permite el control automático de fin de transmisión de una comunicación ligada por vía del modem, ya que nos indica el inicio y fin de una recepción de información. Listo para transmitir se activa bajo para deshabilitar el bit del estado de registro de transmisión de dato vacío (TDRE).

Petición para Transmitir [RTS] (terminal 5) La salida de petición para transmitir habilita al MPU el control sobre periféricos/modems vía el bus de datos. La salida RTS corresponde al estado de los bits CR5 y CR6 dentro del registro de control. Cuando CR6=0 o ambos CR5 y CR6=1, la salida RTS es baja (estado activo). Esta salida puede ser usada también por terminal de datos lista (Data Terminal Ready)[DTR].

Detector de la Portadora de Dato [DCD] (terminal 23) Esta entrada de alta impedancia compatible con TTL, permite el control automático como en el fin de recepción, de una comunicación ligada por medio de una salida (detector de portadora de dato) de un modem. La entrada DCD inhibe e inicializa la sección de recepción de el ACIA cuando es alta. Una transición de bajo a alto del detector de portadora inicia una interrupción a el MPU, la cual indica que ocurre una pérdida de portadora, siempre y cuando el bit de habilitación de interrupción en recepción está activado. El RxCLK debe correr para una operación correcta de DCD.

Funcionamiento del ACIA

A continuación se mencionarán algunas características funcionales del ACIA.

Encendido/Reinicio Maestro.

El reinicio maestro (bits CR0 y CR1 del registro de control) debe ser colocado durante la inicialización del sistema para asegurar la condición de reinicio y preparar la programación del ACIA para su configuración funcional cuando la comunicación de él es requerida. Durante el 1" reinicio maestro las salidas IRQ y RTS son amarradas a un nivel alto. En todos los demás reinicios maestros, la salida RTS puede ser programada alta o baja cuando IRQ permanece fija en alto. Los bits de control CR5 y CR6 también deben ser programados para definir el estado de RTS siempre que es utilizado el reinicio maestro. El reinicio maestro cumple la función del reset que tienen en otros dispositivos.

El ACIA también contiene un encendido (power on) de reinicio lógico, al detectar en la línea de alimentación una transición de

encendido, amarra al chip en un estado de reinicio para prevenir, por la inicialización, errores en las terminales de transmisión debidos a transiciones, de este circuito depende el limpiado de transiciones de encendido de alimentación.

El reinicio de encendido es realizado por medio del bus en la programación del reinicio maestro, el cual puede ser aplicado previo a la operación del ACIA.

Después del reinicio maestro, el registro de control programable puede ser iniciado para un número de opciones, entre las cuales están: la variación de la razón de división de los ciclos de reloj, la variación de la longitud de la palabra de transmisión, uno o dos bits de parada, el tipo de paridad (par o non o sin paridad), etc.

Ejemplos de Operación del ACIA

A continuación se dará un ejemplo de como se realizaría una transmisión y recepción de datos empleando el ACIA.

Recepción.

Un dato es recibido proveniente de un periférico/modem por medio de la entrada de recepción de datos. La sincronización en la razón a un ciclo del reloj entre uno es proveniente de un reloj externo, mientras la división entre 16 y 64 son provistas por una sincronización interna.

El bit de sincronización en los modos de división entre 16 y 64 es iniciado cuando se detectan 8 ó 32 muestras bajas en la línea de recepción respectivamente. La capacidad de suspender un bit de inicio falso asegura que un medio bit de un bit de inicio es recibido antes que el reloj interno sea sincronizado al tiempo del bit.

Cuando un carácter es recibido la paridad (par o non) puede ser ó no checada y la indicación de error puede ser habilitada a lo largo del registro de estado con: error de formato, error de desborde y registro de recepción de datos lleno.

En una secuencia típica de recepción, el registro de estado es leído para determinar si un carácter es recibido por el periférico. Si el registro de recepción de datos esta lleno, cuando un comando de lectura de dato es recibido proveniente del MPU, el carácter es

colocado en el bus de 8 bits del ACIA. Cuando la paridad ha sido seleccionada para una palabra de 7 bits (7 bits más paridad), el receptor despoja al carácter del bit de paridad (D7=0), así sólo el dato es transferido al MPU, ésta característica reduce la programación del mismo.

El registro de estado puede ser leído continuamente para determinar cuando otro carácter esta habilitado en el registro de recepción de dato. La recepción es también doblemente almacenada por lo que un carácter puede ser leído proveniente del registro de recepción de datos mientras otro carácter es recibido en el registro de corrimiento. La anterior secuencia es continuada hasta que todos los caracteres son recibidos.

Transmisión.

Una secuencia típica de transmisión consiste en leer el registro de estado, ya sea como el resultado de una interrupción ó en el turno del ACIA en una secuencia de poleo.

Un carácter puede ser escrito en el registro de transmisión de dato si el resultado de la operación de lectura del estado indica que el registro de transmisión esta vacío. Este carácter es transferido a un registro de corrimiento donde es transmitido por la salida de transmisión de dato en forma seriada, precedido por un bit de inicio y seguido por uno ó dos bits de parada. Internamente la paridad (par o non) puede ser añadida opcionalmente a el carácter agregándola entre el último bit del dato y el primer bit de parada. Después que el primer carácter es escrito en el registro de transmisión de dato, el registro de estado puede ser leído nuevamente, para checar la condición de registro de transmisión de dato vacío y el actual estado del periférico.

Si el registro de transmisión esta vacío, otro carácter puede ser cargado para una transmisión continua aunque el anterior carácter este en proceso de transmisión (debido al doble almacenamiento).

El segundo carácter es automáticamente transferido al registro de corrimiento cuando el anterior a transmitir es completado.

Esta secuencia puede ser continuada hasta que todos los caracteres que se desean hayan sido transmitidos.

Consideraciones Eléctricas para el buen Manejo del ACIA

El ACIA como todo dispositivo posee una serie de características de funcionamiento, las cuales se deben de cumplir para el buen desempeño del mismo. Entre las características eléctricas recomendadas por el fabricante, se encuentran las mostradas en la siguiente tabla:

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage		V_{IH}	$V_{SS}+2.0$	–	V_{CC}	V
Input Low Voltage		V_{IL}	$V_{SS}-0.3$	–	$V_{SS}+0.8$	V
Input Leakage Current (V_{in} = 0 to 5.25 V)	R/W, CS0, CS1, CS2, Enable, RS, RxD, RxC, CTS, DCD	I_{in}	–	1.0	2.5	μA
Hi-Z (Off State) Input Current (V_{in} = 0.4 to 2.4 V)	D0-D7	I_{TSI}	–	2.0	10	μA
Output High Voltage (I_{Load} = −205 μA, Enable Pulse Width < 25 μs)	D0-D7	V_{OH}	$V_{SS}+2.4$	–	–	V
(I_{Load} = −100 μA, Enable Pulse Width < 25 μs)	TxData, RTS		$V_{SS}+2.4$	–	–	
Output Low Voltage (I_{Load} = 1.6 mA, Enable Pulse Width < 25 μs)		V_{OL}	–	–	$V_{SS}+0.4$	V
Output Leakage Current (Off State) (V_{OH} = 2.4 V)	IRQ	I_{LOH}	–	1.0	10	μA
Internal Power Dissipation (Measured at T_A = 0°C)		P_{INT}	–	300	525*	mW
Input Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	D0-D7	C_{in}	–	10	12.5	pF
	E, TxCLK, RxCLK, R/W, RS, RxData, CS0, CS1, CS2, CTS, DCD		–	7.0	7.5	
Output Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	RTS, TxData	C_{out}	–	–	10	pF
	IRQ		–	–	5.0	

*For temperatures less than T_A = 0°C, P_{INT} maximum will increase

Características Eléctricas

Consideraciones de Tiempos

Por otra parte el ACIA posee características en su funcionamiento las cuales deben de considerarse en el diseño y operación de sistemas de comunicación que lo empleen. Dichas características se relacionan con el tiempo de la seriación de datos en la transmisión y recepción, las señales de sincronización, interrupciones, etc. y con los tiempos de los ciclos del bus. Estas características se muestran a continuación:

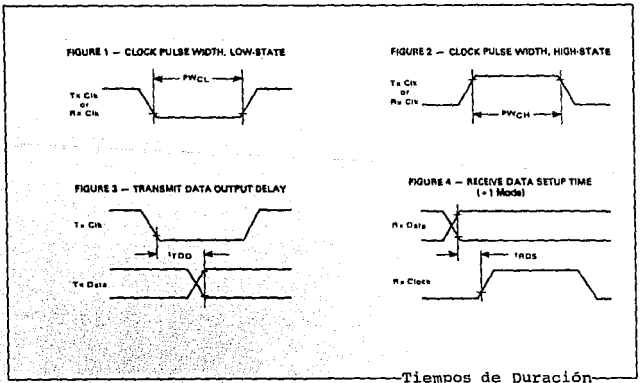
BUS TIMING CHARACTERISTICS (See Notes 1 and 2 and Figure 7)

Ident. Number	Characteristic	Symbol	MC6850		MC68A50		MC68B50		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	µs
2	Pulse Width, E Low	PW_{EL}	430	9500	280	9500	210	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9500	ns
4	Clock Rise and Fall Time	$t_{r,\ tf}$	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	50*	20	50*	20	50*	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Input Data Setup Time	t_{DSW}	195	–	80	–	60	–	ns

*The data bus output buffers are no longer sourcing or sinking current by t_{DHR}max (High Impedance)

Características de Tiempos de Bus

En la anterior tabla se enlistaron las características de algunas señales empleadas en el ACIA. En la siguiente figura se enumeran los tiempos de duración de las mismas señales y otras más requeridas para el ACIA.



Tiempos de Duración

II.c Características y Funcionamiento del PIA MC-6821

Para el correcto conocimiento del PIA MC6821 analizaremos los siguientes puntos:

a) introducción
b) diagrama de bloques
c) asignación de pines
d) funcionamiento de las principales señales del PIA
e) programación y ejemplo
f) diagrama de tiempos

Introducción

Este dispositivo es capaz de interconectar al MPU con dispositivos periféricos a través de un par de puertos bidireccionales de 8 bits y cuatro líneas de control.

La configuración inicial del PIA es programada por el MPU durante la etapa de inicialización. Cada línea del puerto puede ser programada de manera independiente para actuar como entrada o como salida ; y cada una de las cuatro líneas de control/interrupción puede ser programada para varios modos de operación ofreciendo una gran flexibilidad en muchas operaciones de interfaz.

Diagrama de Bloques

En el diagrama siguiente podemos observar que el PIA cuenta con:

1) un bus de 8-bits de datos para comunicación con el MPU
2) dos buses de 8-bits para comunicarse con los periféricos
3) dos registros programables de control
4) dos registros programables de dirección de datos
5) cuatro líneas de entrada de interrupciones controladas individualmente

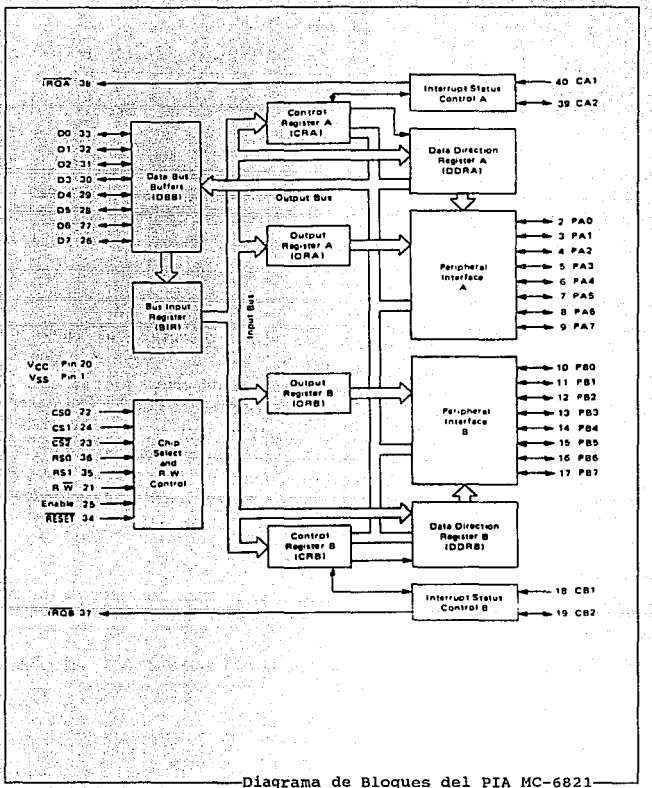
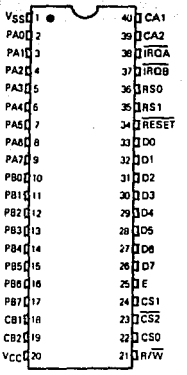


Diagrama de Bloques del PIA MC-6821

Asignación de Pines

El PIA 6821 presenta la siguiente disposición en sus pines:



Disposición de Pines en el 6821

Funcionamiento de las Principales Señales del PIA

Para el estudio de las principales señales del PIA 6821 las dividiremos en los siguientes puntos

1) señales de interfaz del PIA con el MPU
2) líneas de interfaz a periféricos del PIA
3) control interno

Señales de Interfaz del PIA con el MPU

El PIA se conecta al MPU 6809 a través de un bus de 8 bits de datos bidireccionales, tres líneas de selección de chip, dos de selección de registro, dos de petición de interrupciones, una de lectura/escritura, una de selección, habilitación y una de reset. Estas líneas las describiremos a continuación.

Datos bidireccionales (D0 - D7) Las líneas de datos bidireccionales sirven para transferir datos entre el MPU y el PIA. Los manejadores del bus de datos son dispositivos de tres estados que permanecen en estado de alta impedancia, excepto cuando el MPU desempeña en el PIA una operación de lectura. La línea de lectura/escritura está en estado de lectura (alto) cuando el PIA es seleccionado para una operación de lectura.

Habilitador (E) El pulso habilitador E es la única señal de tiempo que es alimentada a el PIA. Todas las otras señales de tiempo son referenciadas a los flancos de subida y de bajada del pulso E.

Lectura / Escritura (R/W) Esta señal es generada por el MPU para controlar el sentido de la transferencia de datos en el respectivo bus. Un estado bajo en la línea R/W del PIA habilita los buffers de entrada y los datos son transferidos del MPU a el PIA, una vez que la señal E del dispositivo fue seleccionada. Un nivel alto en R/W habilita el PIA para enviar los datos al MPU a través del bus de datos. Las salidas buffer del PIA son activadas cuando la propia dirección y el pulso E está presente.

Selectores de Chip (CS0, CS1, CS2) Estas tres señales de entrada son usadas para seleccionar el PIA. CS0 y CS1 deben estar altas y CS2 debe estar baja para la selección del dispositivo, de esta manera, la transferencia de los datos, se desarrolla bajo el control de la señal E y R/W. Las líneas de selección de chip deben permanecer estables durante la duración del pulso E. El dispositivo es deseleccionado cuando cualquiera de las líneas selectoras de chip (CS0, CS1, CS2) están en estado inactivo.

Reset La línea de RESET es utilizada para restablecer todos los bits de los registros en el PIA usando una lógica cero. Esta línea también puede ser usada como un inicializador maestro durante la operación del sistema.

Selectores de Registro (RS0, RS1) Las dos líneas de selección de registro nos permiten escoger algunos de varios registros dentro del PIA . Estas dos líneas son usadas, conjuntamente con el control

de registros internos para seleccionar un registro particular que será escrito o leído.

Las líneas de registro y selección de chip deberán estar en estado estable durante la duración del pulso E, mientras se realiza el ciclo de lectura o escritura.

Petición de Interrupción (IRQA y IRQB) Al estar en niveles bajos las líneas de petición de interrupción IRQA e IRQB pueden interrumpir inmediatamente al MPU o a través de circuiteria de priorización de interrupciones. Estas líneas son "drain abierto" lo que permite que las líneas de petición de interrupción puedan estar unidas en una configuración OR alambrada.

Cada línea de petición de interrupción tiene dos bits internos de bandera de interrupción que pueden provocar que la línea de petición de interrupción vaya a un estado bajo. Cada bit de bandera es asociado con una línea de interrupción de algún periférico en particular. El PIA cuenta también con cuatro bits habilitadores de interrupción que pueden usarse para inhibir una interrupción de algún dispositivo periférico.

El servicio a las interrupciones por el MPU debe estar acompañado por una rutina de software que, basada en una priorización lea secuencialmente y pruebe los dos registros de control en cada PIA para los bits de bandera que estén activos.

Las banderas de interrupción son borradas (llevadas a cero) como resultado de una operación de lectura de datos periféricos del registro de datos correspondiente. Luego de ser borrado, el bit de bandera de interrupción no puede ser habilitado para activarse, hasta ser deseleccionado el PIA durante un pulso E. Este pulso es usado para la condición de las líneas de control de interrupción (CA1, CA2, CB1, CB2).

Cuando éstas líneas son usadas como entradas de interrupción el último pulso E debe ocurrir del borde inactivo hacia el borde activo de la señal de entrada de interrupción en la red de sensado de borde. Si la bandera de interrupción a sido habilitada y la red de sensado de borde a sido condicionada apropiadamente, la bandera de interrupción será habilitada en la siguiente transición activa de la entrada de interrupción del pin.

Líneas para Interfaz de Periféricos en el PIA

El PIA proporciona dos puertos de 8-bits bidireccionales y cuatro líneas de control/interrupción para interconectarse con dispositivos periféricos

Puerto A (PA0 - PA7) Cada una de las líneas del puerto puede ser programada para funcionar como entrada o salida. Para las líneas que serán salidas deberá colocarse un "1" en el bit correspondiente de Registro de Dirección de Datos (DDR) y un "0" para que dicha línea se active como entrada.

Durante una operación de lectura del puerto por el MPU, los datos en éste último, actúan como entradas directas en las correspondientes líneas del Bus de Datos del MPU.

En el modo de entrada, la resistencia interna de PULL-UP en dichas líneas permite tener un máximo de 1.5 cargas en el nivel TTL estandar.

El dato en el registro de salida A (DRA), aparecerá sobre las líneas del puerto que fueron programadas para ser salidas. Un "1" escrito dentro del registro provocará un "nivel alto" en su correspondiente línea de dato, mientras un "0" resultará en un bajo. El registro de salida de datos A puede ser leído por una operación de "lectura de datos periféricos en A" hecha por el MPU, cuando las líneas correspondientes fueron programadas como salidas. Este dato será leído directamente si el voltaje en las líneas del dato periférico es mayor que 2.0 v para una lógica positiva de salida y menor de 0.8 para una lógica negativa.

Puerto B (PB0 - PB7) Las líneas de datos periféricos en la sección B del PIA pueden ser programadas para actuar cada una como entradas o salidas en forma similar a PA0 - PA7.

Cuentan también con la propiedad de tres estados colocándose en estado de alta impedancia cuando las líneas del puerto son usadas como entradas.

En suma, todos los datos sobre las líneas del puerto PB0 - PB7 serán leídos apropiadamente en las líneas programadas como salidas si los voltajes son arriba de 2 [V] para un nivel "alto" y abajo de 0.8 [V] para un nivel "bajo". Las salidas en éstas líneas son compatibles con el estandar TTL y pueden usarse como una fuente de

hasta 1 [mA] a 1.5 [V] para manejar directamente la base de un transistor de switcheo.

Entrada de Interrupción (CA1 - CB1) Las líneas periféricas de entrada CA1 y CB1 son líneas de "sólo entrada" que activan las banderas de interrupción del registro de control. La transición activa para éstas señales es programada también por los dos registros de control.

Control Periférico (CA2) La línea de control periférico CA2 puede programarse para funcionar como una interrupción de entrada o como una salida para control de algún periférico. Si es una salida, esta línea es compatible con el estandar TTL; ya que tiene una resistencia interna de pull-up a la entrada, permitiendo 1.5 cargas de TTL.

Control Periférico (CB2) La línea de control periférico CB2 también puede ser programada para actuar como entrada de interrupción o control de salida de algún periférico. Si es entrada, esta línea estará en alta impedancia siendo compatible con el estandard TTL. Si está como salida, también es compatible con TTL y puede usarse como una fuente de 1 [mA] a 1.5 [V] y con ella manejar directamente la base de un transistor de switcheo.

Esta línea es programada por medio del registro de control B.

Controles Internos

Inicialización

Un RESET tiene el efecto de colocar en ceros todos los registros del PIA. Esta señal habilitará a PA0-PA7, PB0-PB7, CA2 y CB2 como entradas y todas las interrupciones serán deshabilitadas. El PIA debe ser configurado durante el programa de restablecimiento que sigue al RESET.

Existen 6 localidades dentro del PIA que están accesibles al bus de datos del MPU :

a) dos registros periféricos ORA(B)
b) dos registros de dirección de datos DDR A(B)
c) dos registros de control CRA(B)

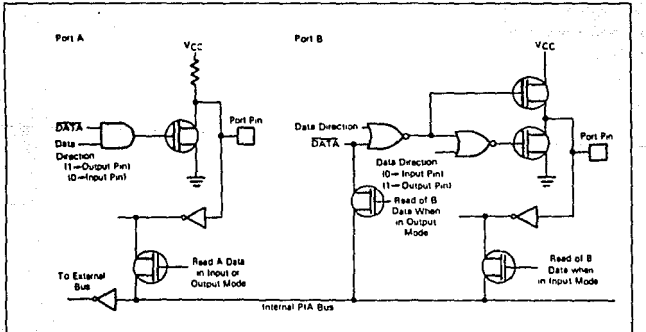
La selección de dichos registros está controlada por las entradas RS0 y RS1 junto con el bit dos en el registro de control. En la figura siguiente se muestra ésta situación.

		Registro de Control de Bit		
RS1	RS0	CRA-1	CRB-2	Localidad Seleccionada
0	0	1	X	Registro Periférico A
0	0	0	X	Registro de Dirección de Dato A
0	1	X	X	Registro de Control A
1	0	X	1	Registro Periférico B
1	0	X	0	Registro de Dirección de Dato B
1	1	X	X	Registro de Control B

Direccionamiento Interno

Características de Hardware de los Puertos A - B

Como se muestra en la figura siguiente el PIA 6821 cuenta con dos Puertos de I/O con características diferentes.



Circuitos equivalentes de los Puertos A y B

El lado A está diseñado para manejar normalmente lógica CMOS con el margen de 30% a 70% del margen del nivel máximo, e incorpora un dispositivo interno de pull-up que permite ser igualmente conectado en el modo de entrada. Es por esto que el lado A maneja más corriente en el modo de entrada que el puerto B.

En contraste el lado B usa un buffer normal de tres estados NMOS. Aquí se puede manejar cargas extras como Darlington, sin ningún problema. Cuando el PIA sale del RESET el puerto A representa entradas con resistores pull-up, mientras que el lado B (también en modo de entrada) estará flotando entre "alto" o "bajo", dependiendo del encendido de la carga que allí se encuentre conectada.

Registro de Control (CRA y CRB) Los dos registros de control permiten al MPU la operación de las cuatro líneas de control de periféricos CA1, CA2, CB1, CB2.

En resumen, éstas líneas capacitan al MPU para habilitar las líneas de interrupción y monitorear el estado de las banderas de interrupción. Los bits 0 al 5 de los dos registros de control pueden escribirse o leerse por el MPU cuando las líneas de selección de chip apropiadas y las señales de selección de registro son aplicadas. Los bits 6 y 7 de ambos registros sólo pueden ser leídos y serán modificados por la aparición de interrupciones externas sobre la líneas de control CA1, CA2, CB1 ó CB2 .

El formato de la palabra de control es mostrado en la siguiente figura :

b7	b6	b5	b4	b3	b2	b1	b0
IRQA (B)1 Bandera	IRQA (B)2 Bandera	CA2 (CB2) Control			DDR Acceso	CA1 (B1) Control	

- b0 -

CA1 (CB2) Solicitud de interrupción habilitada/deshabilitada

b0 = 0 : Deshabilita la interrupción del MPU IRQA (B) para una transición activa (CB1) *

b0 = 1 : Habilita la interrupción del MPU IRQA (B) para una transición activa de CA1(CB1)

* : IRQA(B) ocurrirá en la siguiente (MPU generated) transición positiva de b0 si ocurrió una transición activa mientras fue deshabilitada la interrupción

- b1 -

Determina la transición CA1 (CB1) activa para pruebas sobre la bandera de interrupciones IRQA (B)1 ---- bit 7

b1 = 0 : IRQA (B)1 habilitado para transición alto-bajo en CA1 (CB1)

b1 = 1 : IRQA (B)1 habilitado para una transición bajo-alto en CA1 (CB1)

- b2 -

Determina si el Registro de Dirección de Dato y el Registro de Salida es direccionado.

b2 = 0 : Selecciona el registro de dirección de datos

b2 = 1 : Selecciona el registro de salida

- b3 b4 b5 -

I) CA2 (CB2) establecidas como entradas para b5 = 0

b5 b4 b3

0 CA2 (CB2) Petición de interrupción habilitada/deshabilitada

b3 = 0 : deshabilita IRQA (B) interrumpiendo al MPU por una transición activa CA2 (CB2)*

b3 = 1 : habilita la interrupción IRQA(B) en el MPU por una transición activa CA2 (CB2)

* : IRQA(B) ocurrirá en la siguiente (MPU generated) transición positiva de b3 si ocurrió una transición activa mientras la interrupción fue deshabilitada

Determina la transición activa CA2 (CB2) para la habilitación de bandera de interrupción IRQA(B) ---- bit b6

b4 = 0 : IRQA(B)2 habilitado para transición alto-bajo en CA2 (CB2)

b4 = 1 : IRQA(B)2 habilitado para transición bajo-alto en CA2(CB2)

II) CA2 (CB2) Establecido como salida por b5=1 (Note que la operación de CA2 y CB2 como funciones de salida no son idénticas)

b5 b4 b3

1 0 -> CA2 [NEG]

b3 = 0 : lee STROBE con el Restablecimiento de CA1 CA2 va abajo en la primera transición alto-bajo de E siguiendo a la lectura del MPU del Registro de Salida A; Regresa al estado alto para la siguiente transición activa CA1, como especifica para el bit 1

b3 = 1 : lee STROBE con restablecimiento de E CA2 pasa bajo en la primera transición alto-bajo de E, siguiendo una lectura del MPU del registro de salida A; regresa a alto para la siguiente transición alto-bajo de E durante una deselección.

-> CB2

b3 = 0 : escribe STROBE con restablecimiento de CB1, y CB2 va a un estado bajo en la primera transición bajo-alto siguiendo a una escritura del MPU dentro del registro de salida B; regresando a alto para la siguiente transición activa CB1 como específica para el bit 1. CRB-b7 debe primeramente borrarse para una lectura de dato

b3 = 1 : escribe STROBE con restablecimiento de E CB2 va a un estado "bajo" en la primer transición bajo-alto de E siguiendo una escritura del MPU dentro del registro de salida B; regresando a alto para la siguiente transición bajo-alto de E, siguiendo un pulso de E que ocurre mientras la parte fue deseleccionada

b5 b4 b3
1 1 └> Set/Reset CA2 (CB2)
CA2 (CB2) pasa a un nivel "bajo" cuando el MPU escribe dentro del registro de control y b3 = 0
CA2 (CB2) pasa a ser alto con una escritura dentro del registro de control y b3 = 1

- b6 -

IRQA(B)2 Bandera de interrupción (bit 6)
Cuando CA2(CB2) es una entrada, IRQ(B) va a una transición activa alta CA2(CB2) borrado automáticamente por una lectura del MPU del registro de salida A(B). Pudiendo también ser borrado por un reset de hardware CA2(CB2) establecida como salida (b5 = 1); IRQA(B)2=0 no afectada por transiciones de CA2(CB2)

- b7 -

IRQA(IRQB) Bandera de interrupción (bit 7)
Va a alta con una transición de CA1(CB1); es borrada automáticamente por una lectura del MPU del registro de salida A(B). Pudiendo también ser borrada por un reset de hardware

Bits de Control en la Dirección de Acceso (CRA-2 y CRB-2) El bit 2 en cada registro de control (CRA y CRB), determina la selección del registro periférico de salida o el correspondiente registro de dirección de datos (DDR), cuando la señal apropiada del registro de selección fue aplicada a RS0 y RS1. Un "1" en el bit-2 permite accesos al registro interfaz de periféricos, mientras un "0" provoca que el DDR sea direccionado.

Bandera de Interrupción (CRA-6, CRA-7, CRB-6 y CRB-7) Los cuatro bits de las banderas de interrupción son habilitados por transiciones activas de señales en las cuatro interrupciones y las líneas de control de periféricos estando programadas como entradas. Estos bits no pueden habilitarse directamente del bus de datos del MPU y son deshabilitados indirectamente por una operación de lectura de un dato periférico sobre la sección apropiada.

Manejo de las Líneas de Control Periférico CA2 y CB2 (CRA-3, CRA-4, CRA-5, CRB-3, CRB-4 y CRB-5) Los bits 3, 4 y 5 de los dos registros de control son usados para manejar las líneas de control periférico CA2 y CB2. Estos bits determinan si las líneas de control serán interrupciones de entrada o señales de control de salida. Si el bit CRA-5 (CRB-5) es bajo, CA2(CB2) es una línea de entrada de interrupción similar a CA1(CB1). Cuando CRA-5 (CRB-5) es alta, CA2(CB2) queda como una señal de salida que puede usarse para el control de la transferencia de datos periféricos. En el modo de salida CA2 y CB2 tienen ligeramente diferentes características de carga.

Control de las Líneas de Entrada de Interrupción (CRA-0, CRB-0 CRA-1 y CRB-1) Los dos bits de menos significación en el registro de control son usados para controlar las líneas de entrada de interrupción CA1 y CB1. Los bits CRA-0 y CRB-0 sirven para habilitar las señales de interrupción del MPU IRQA e IRQB, respectivamente. Los bits CRA-1 y CRB-1 determinan la transición activa de las señales de entrada de interrupción, CA1 y CB1.

Modo de Operación de las Líneas de Control CA1 y CA2

La transferencia de datos entre el PIA y algún dispositivo externo en ocasiones requiere de una razón de tiempo para tomar el dato de su lugar; por lo tanto las líneas de control serán necesarias entre el PIA y el dispositivo externo cuando a sido leído el dato.

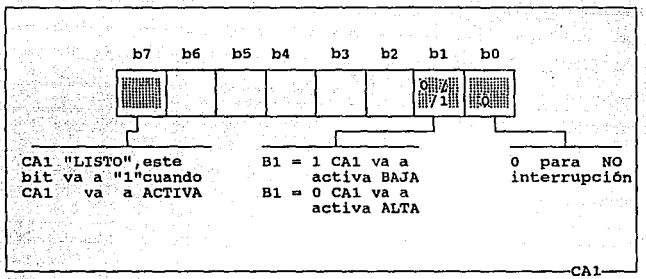
Para el Pto. A tenemos dos líneas de control CA1 (entrada hacia el PIA) y CA2 (entrada o salida).

CA1 puede informar al PIA, actuando como receptor de datos, que el dato está disponible cuando este dato es leído por el PIA.

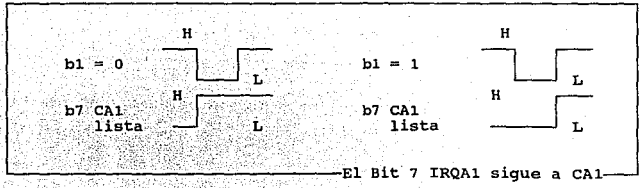
CA2 puede informar al dispositivo externo que el dato ha sido leído. Por lo tanto otro byte debe ser colocado sobre la línea de datos. CB1 y CB2 pueden desempeñar de manera similar la misma función para el Pto.B.

Ambos, CA1 y CA2 son controlados por bits específicos de el Registro de Control A como se muestra a continuación.

CA1 Control



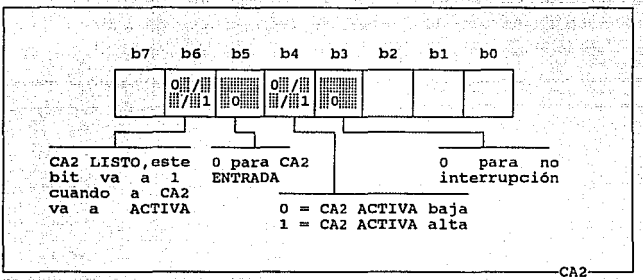
Los tres bits asociados con CA1 son mostrados arriba. No estamos usando interrupciones en este momento por lo tanto b0 = 0. b1 determina si CA1 habilitará el b7 "LISTO" cuando CA1 va abajo (b1=0) o a alto (b1=1). EL bit 7 CA1 "LISTO", también llamado IRQA1 en la literatura de Motorola, indica cuando va al estado 1 que CA1 esta ACTIVA.



El b7 "LISTO" del PIA será automáticamente borrado cuando el dato es leído del Data Buffer, por ejemplo al realizar una LDA A PIABFA. Este bit es de sólo lectura, es decir no puede ser habilitado o borrado por una instrucción de escritura.

CA2 Control

Cuando el b5 del registro de control es igual con 1, CA2 actúa como una línea de entrada similar para CA1.



b5 = 0 para entrada y b4 y b3 realizan lo mismo que b1 y b0 para CA1.

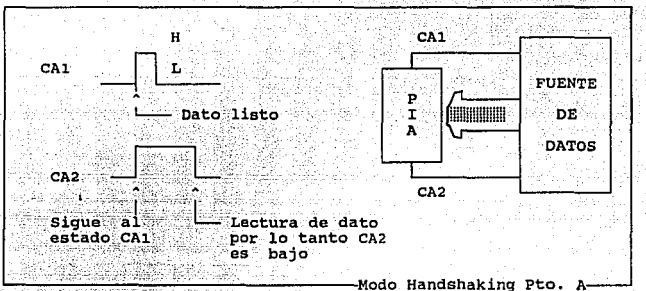
Modos de Operación de CA2 en Salida

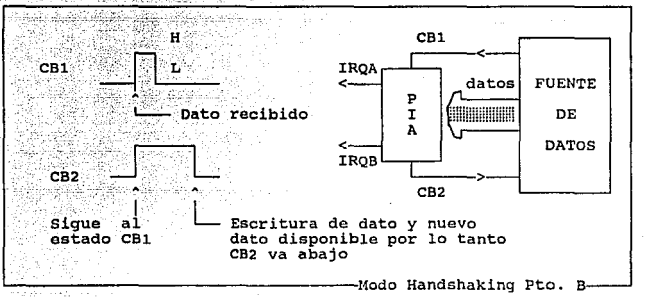
Existen aun dos modos de operación de CA2 actuando como salida, es decir b5 = 1, que son el modo "HANDSHAKING" y el modo "STROBE". HANDSHAKING

Modo "HANDSAKING"

CA2 puede usarse como una línea de control de salida en una modo "HANDSHAKING" cuando b5 = 1 y los bits b4 y b3 = 0. En este modo el Pto. A actúa como un receptor de datos. CA2 irá a alto automáticamente cuando CA1 va a ACTIVA e irá abajo cuando el buffer de datos A es leído.

Cuando CA2 va abajo el dispositivo externo sabrá que el nuevo dato puede colocarse en las líneas de datos.





En forma abreviada podemos decir que para el Pto. A :

Fuente de Datos ——>	activa CA1	; El dato puede ser leído
PIA ——————>	activa CA2 alto	; Bandera de enterado
MPU ——————>	lee el dato	; El dato es leído
CA2 ——————>	BAJO al ser	; OK. El dato ha sido leído, puedes enviar otro

El modo "handshaking" permite una óptima operación en el flujo de los datos.

Modo "STROBE"

Este último modo está disponible cuando b5 = 1, b4 = 0 y b3 = 1, y es similar al modo visto anteriormente donde CA2 va abajo cuando el dato es leído (LDA A PIABFA) dentro del Data Buffer (Pto. A). Difiere en que CA2 regresa automáticamente al estado "1" varios microsegundos (una instrucción) después. Similarmente en el Pto. B, CB2 va ABAJO cuando una operación de escritura (STA A PIABFB) se ejecuta y regresa al estado "1" automáticamente.

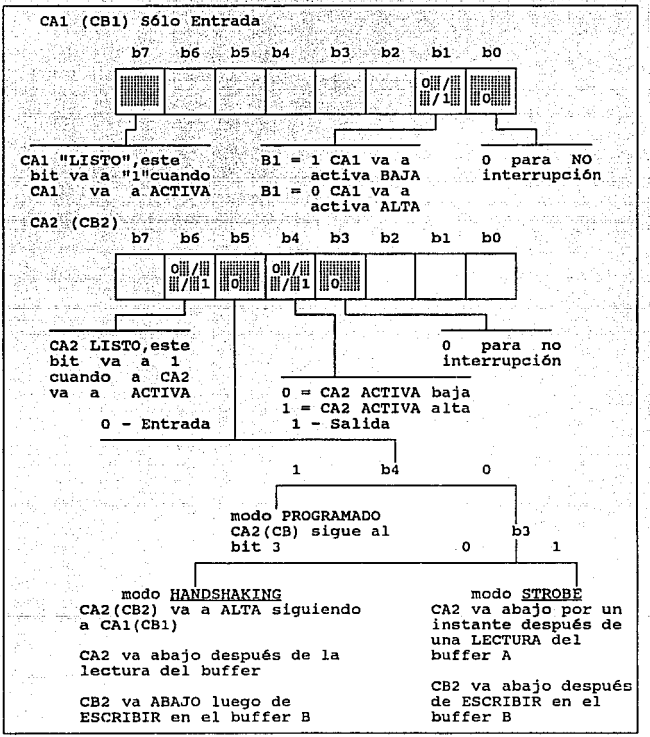

CA1 (CB1) Sólo Entrada
b7 b6 b5 b4 b3 b2 b1 b0
0 / 1
0
CA1 "LISTO",este bit va a "1"cuando CA1 va a ACTIVA
B1 = 1 CA1 va a activa BAJA
B1 = 0 CA1 va a activa ALTA
0 para NO interrupción
CA2 (CB2)
b7 b6 b5 b4 b3 b2 b1 b0
0 / 1
0
0 / 1
0
CA2 LISTO,este bit va a 1 cuando a CA2 va a ACTIVA
0 = CA2 ACTIVA baja
1 = CA2 ACTIVA alta
0 para no interrupción
0 - Entrada
1 - Salida
1
b4
0
modo PROGRAMADO
CA2(CB) sigue al bit 3
b3
0
1
modo HANDSHAKING
CA2(CB2) va a ALTA siguiendo a CA1(CB1)
CA2 va abajo después de la lectura del buffer
CB2 va ABAJO luego de ESCRIBIR en el buffer B
modo STROBE
CA2 va abajo por un instante después de una LECTURA del buffer A
CB2 va abajo después de ESCRIBIR en el buffer B

Diagrama de Tiempos

En la siguiente figura se muestra el diagrama de tiempos del PIA MC6821 para los ciclos de lectura y escritura.

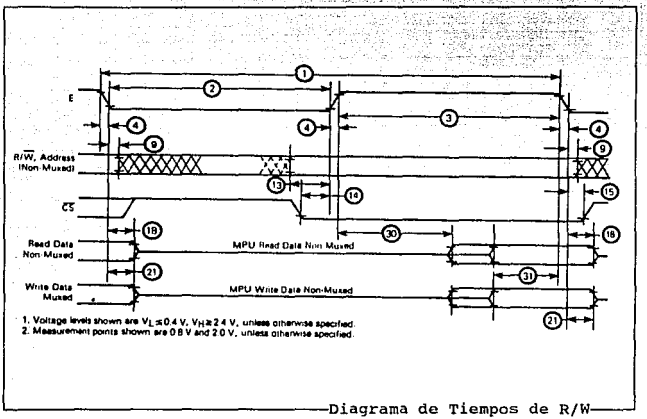


Diagrama de Tiempos de R/W

En el diagrama anterior podemos observar que primeramente debe darse un cambio en E de bajo a alto, habilitarse CS como baja y que pasados el tiempo que resulta de sumar 14 con 30, es decir 170 ns, podemos habilitar READ/DATA o WRITE/DATA y operar con los datos confiablemente. Para hacerlos tendremos el tiempo denominado como 31 que es de 60 ns. Estos resultados fueron extraídos de la siguiente tabla.

BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

Ident. Number	Characteristic	Symbol	MC6821		MC68A21		MC68B21		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	µs
2	Pulse Width, E Low	PW_{EL}	430	–	280	–	210	–	ns
3	Pulse Width, E High	PW_{EH}	450	–	280	–	220	–	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	-	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	50*	20	50*	20	50*	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

*The data bus output buffers are no longer sourcing or sinking current by t_{DH}(R)max (High Impedance)

Tabla de Equivalencias de Tiempos

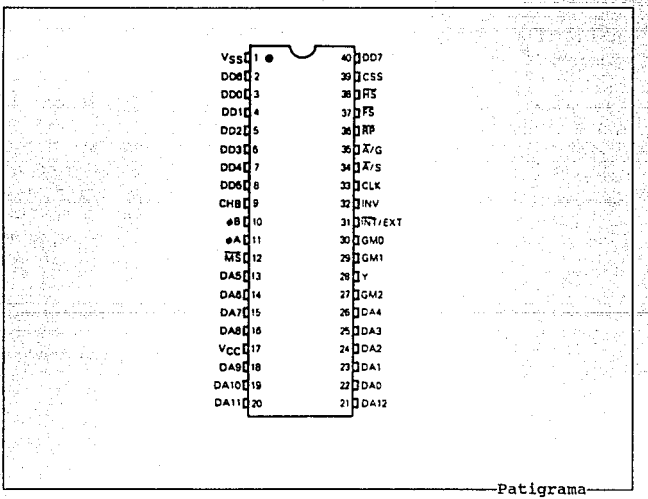
En el apéndice de hojas de datos técnicos podemos observar más información sobre otros tiempos de importancia.

II.d Generador de Video (VDG) MC-6847 y Modulador MC-1372

Para el despliegue al exterior de los procesos que realice el sistema, se empleará el generador de video VDG MC-6847Y y el modulador MC-1372 de la familia de Motorola, el cual modifica las señales del VDG para poder ser captadas por un televisor.

Generador de Video

La arquitectura externa del MC6847 se da en el siguiente diagrama.



Patigrama

El generador de video (VDG) provee de la interfase para poder visualizar en un receptor de televisión, blanco y negro o color, los datos que se manejen en una computadora.

El generador de video lee un dato de la memoria y produce una señal de video, la cual generará caracteres alfanuméricos o gráficas, la señal de video puede ser modulada, para transmitirse por el canal 3 o 4, y usando el circuito integrado MC1372.

Las características de el MC6847 son:

* Compatible con el MPU 6809
* Genera 4 diferentes formas alfanuméricas, dos tipos de semigráficas, y 8 tipos de gráficas.
* Alfanuméricos, con posibilidad de 32 caracteres por línea y 16 líneas.
* Soporta alfanuméricos en video inverso.
* Ofrece gráficos con las siguientes posibilidades, 64x64, 128x64, 128x96, 128x192 o 256x192.
* Compatible con circuitos TTL.

El MC6847 generador de video, provee de una interfase simple, para mostrar información digital hacia un monitor de color o blanco y negro. La transmisión de señales de televisión en Norte y Sudamérica y Japón conforman el Comité de Sistema Nacional de Televisión. Este sistema está basado en la repetición de la pantalla (campo) de 60 veces por segundo, son 525 líneas entrelazadas para la estructura de la pantalla, o la mitad de este número por campo. El MC6847 emite un campo de 262 líneas, 60 veces por segundo.

Descripción de señales

A continuación una esplicación de les terminales, empezando por los buses de datos y direcciones.

Líneas de direcciones para el despliegue [DA0-DA12] (terminales 13 a 16 y 18 a 26) Trece líneas de direcciones son usadas por el VDG para buscar y mandar a la pantalla el contenido de la memoria. La dirección de inicio que es seleccionada por la memoria está localizada en la esquina superior izquierda de la pantalla. Así como en las televisiones el barrido se hace de izquierda a derecha y de arriba a abajo, el VDG incrementa las direcciones de la memoria RAM, estas pueden tener alta impedancia, cuando MS

(terminal 12) va hacia nivel bajo. A_0-A_3 cambian cuando se activa el área de la pantalla. A_4 cambia cuando se transmiten alfanuméricos, semigráficos y modos CG2, CG3, CG6, RG6. A_5-A_{12} no cambian cuando se activa el área de la pantalla, pero en lugar de esto, ocurren cambios en las direcciones durante el tiempo de borrado, y cercanos a la frontera de este.

Datos de entrada [DD0-DD7] (terminales 2 a 8 y 40) Las 8 líneas de datos son usadas para datos de entrada provenientes de la RAM o una ROM generadora de caracteres, con el fin de ser procesados por el VDG. El dato es entonces interpretado y transformado en luminancia (Y) y salidas de croma (ΦA y ΦB).

Salidas de video [ΦA, ΦB, Y, CHB]

Estas cuatro salidas analógicas son usadas para transferir luminancia y color a un receptor de televisión, por medio del modulador de RF MC1372, estas se explican en seguida.

Luminancia [Y] (terminal 28) Son 6 niveles analógicos de salida que contienen la sincronía, líneas de retroceso, y cuatro niveles de luminancia para el video.

ΦA (terminal 11) Estos tres niveles de salida analógicos se usan en combinación con ΦB e Y para lograr uno de ocho colores posibles.

ΦB Estos cuatro niveles analógicos de salida logran en combinación con ΦA e Y uno de ocho colores. Adicionalmente, un nivel analógico es usado para especificar el tiempo que va a estar presente un color.

Croma [CHB] (terminal 9) Esta terminal es una salida analógica y provee de una referencia de DC correspondiente al valor de ΦA y ΦB. CHB es usado para garantizar el buen acoplamiento al minimizar las variaciones existentes entre el MC6847 y el MC1372, cuando está en nivel bajo, desactiva registros dentro del circuito.

Sincronización de las salidas (FS, HS, RP)

Estas salidas son dirigidas a circuitos externos a el VDG, y que dan tiempos internos de referencia para los siguientes estados del VDG.

Sincronía de campo [FS] (terminal 37) La transición de nivel alto a nivel bajo de la salida FS coincide con el fin del área activa en la pantalla. Durante este tiempo, el MPU puede tener total acceso hacia la RAM sin causar "parpadeos" no deseados en la pantalla. El cambio de nivel bajo a alto coincide con el inicio del pulso de sincronía vertical.

Sincronía horizontal [HS] (terminal 38) Los HS coinciden con el pulso de sincronía horizontal, provenientes del televisor y recibidos por el VDG. La transición de nivel alto a bajo de la salida HS, coincide con el estado principal del pulso de sincronía horizontal, y la transición de bajo a alto coincide con el tiempo de retroceso.

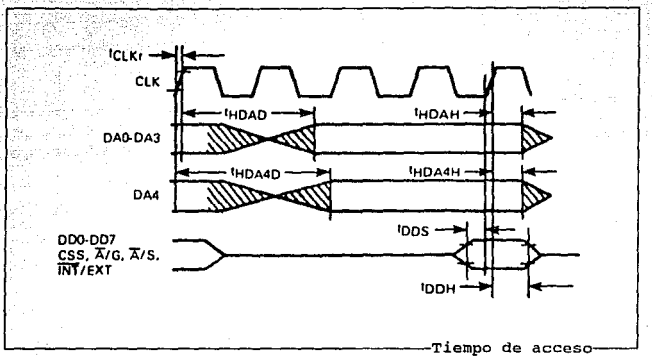
Línea de preencendido [RP, Row Preset] (terminal 36) Cuando es deseado, cualquier carácter generado por la ROM puede ser empleado por el VDG. De esta manera, un contador externo de cuatro bits puede suministrar el incremento a la línea de direcciones. El reloj del contador es realizado por la señal HS y este es desactivado por la señal de RP. Los pulsos de RP ocurren en todos los caracteres alfanuméricos y en el modo de semigráficos. No ocurren estos pulsos de salida para el modo de gráficos.

Sincronización de los datos de entrada (MS, CLK).

Control de tres estados [MS] (terminal 12) Es una entrada que, cuando está en nivel bajo, forza a las líneas de dirección del VDG a un estado de alta impedancia.

Reloj [CLK] (terminal 33) La entrada de reloj del VDG requerida es de 3.5795 MHz, de una señal cuadrada. El ciclo de trabajo de este reloj debe de estar entre el 45 y 55%, la siguiente figura indica

las características del tiempo de acceso de los datos para cada pulso de reloj.



Líneas de entrada para modos de control [A/G, A/S, INT/EXT, GM0, GM1, GM2, CSS, INV] (terminales 35, 34, 31, 30, 29, 27, 39 y 32 respectivamente) Ocho entradas son usadas para controlar el modo de operación del VDG. A/S, INT/EXT, CSS, e INV, pueden hacer cambios de carácter a carácter básicamente.La terminal CSS es usado para seleccionar entre dos colores posibles cuando el VDG está en modo alfanumérico y puede activar entre dos colores que estarán activos cuando el VDG está en modo de semigráficos o gráficos. Se tienen dos diferentes tipos de acceso a la memoria para estos modos, con ciclos de acceso uno corto y otro largo. Una diferencia entre estos tiempos de acceso son, para el corto es el cambio de un ciclo completo de reloj (3.58 MHz)de el correspondiente acceso normal de tiempo (largo) Los accesos de tiempo corto son leídos en la memoria dos veces que si se usaran los tiempos de acceso largos.

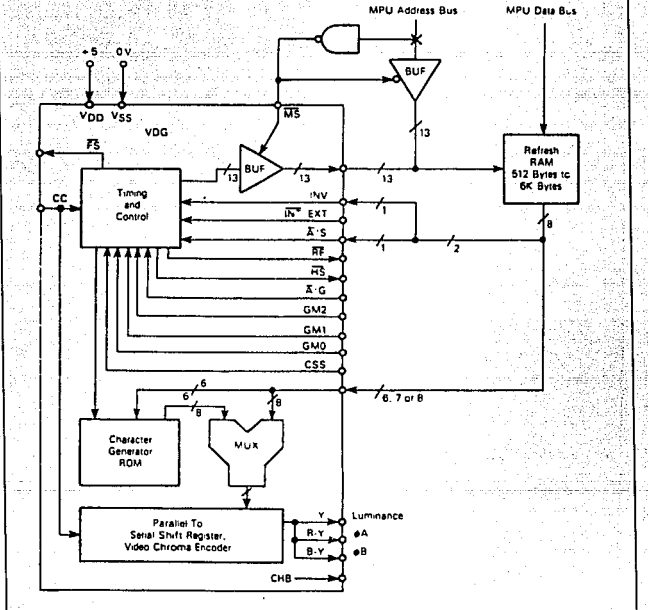
En la siguiente tabla se ilustran los diferentes modos que pueden obtenerse usando estas líneas de control.

A/G	A/S	INT/EXT	INV	GM2	GM1	GM0	Alpha/Graphic Mode Select	# of Colors
0	0	0	0	X	X	X	Internal Alphanumerics	2
0	0	0	1	X	X	X	Internal Alphanumerics Inverted	
0	0	1	0	X	X	X	External Alphanumerics	
0	0	1	1	X	X	X	External Alphanumerics Inverted	
0	1	0	X	X	X	X	Semigraphics 4 (SG4)	8
0	1	1	X	X	X	X	Semigraphics 6 (SG6)	8
1	X	X	X	0	0	0	64 x 64 Color Graphics One (CG1)	4
1	X	X	X	0	0	1	128 x 64 Resolution Graphics One (RG1)	2
1	X	X	X	0	1	0	128 x 64 Color Graphics Two (CG2)	4
1	X	X	X	0	1	1	128 x 96 Resolution Graphics Two (RG2)	2
1	X	X	X	1	0	0	128 x 96 Color Graphics Three (CG3)	4
1	X	X	X	1	0	1	128 x 192 Resolution Graphics Three (RG3)	2
1	X	X	X	1	1	0	128 x 192 Color Graphics Six (CG6)	4
1	X	X	X	1	1	1	256 x 192 Resolution Graphics Six (RG6)	2

Líneas para modos de control

Operación del VDG

El diagrama simplificado del VDG se muestra en la siguiente figura.



Arquitectura del VDG

El generador externo de 3.58 Mhz que dan la gama de colores y que sirve de reloj para manejar el VDG. Como puede observarse en la figura que se muestra a continuación.

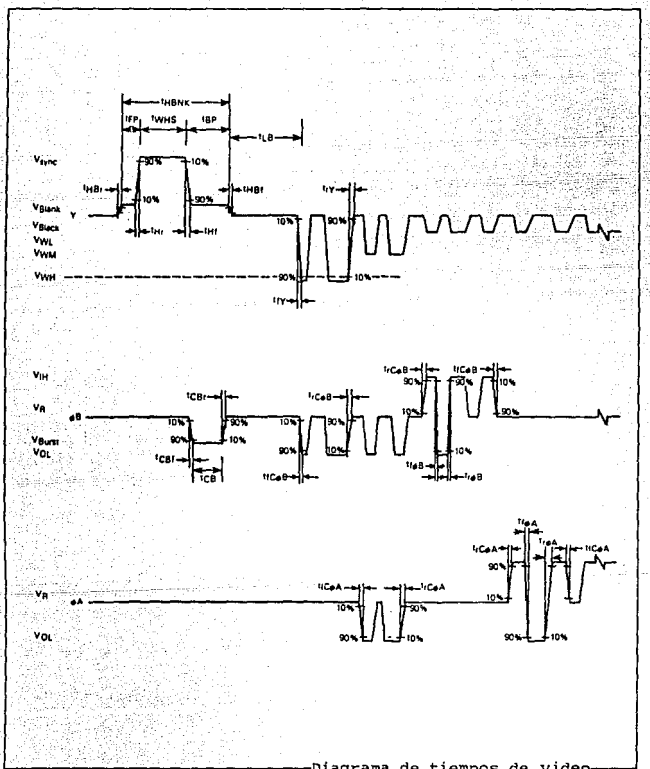


Diagrama de tiempos de video

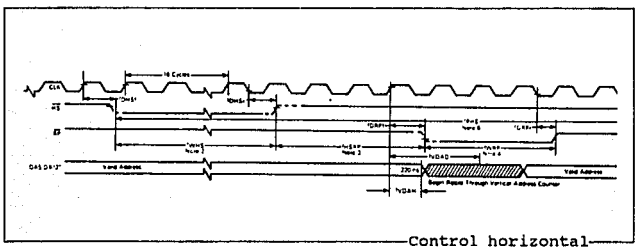
Notese que el espacio horizontal entre tiempos de borrado es de 193.1 períodos de reloj (53.95 ms.). La compensación de la orilla izquierda de la pantalla es de 283 períodos y la otra parte es de 128 períodos (37.75 ms). De las 242 líneas verticales de la pantalla entre tiempos de retroceso, 192 son utilizadas para los caracteres. La compensación de la parte superior de la pantalla es de 25 líneas. Bajo el impedimento del reloj, el elemento menor que puede ser transmitido por el VDG es la mitad del ciclo de reloj que es de 3.58 MHz para explorar una línea.

Manejo de las direcciones de la memoria

Los controladores de las memorias normalmente manejan el "refrescamiento" de las direcciones del video dentro del registro de memoria para posiblemente ser desplegado por el CRT, cuando MS se activa en bajo por un dispositivo externo, las salidas van hacia un estado de alta impedancia que provoquen o no la interrupción del MPU para mandar direcciones existentes en la memoria, el MPU puede manipular datos directamente de la memoria.

Tiempo de video y control

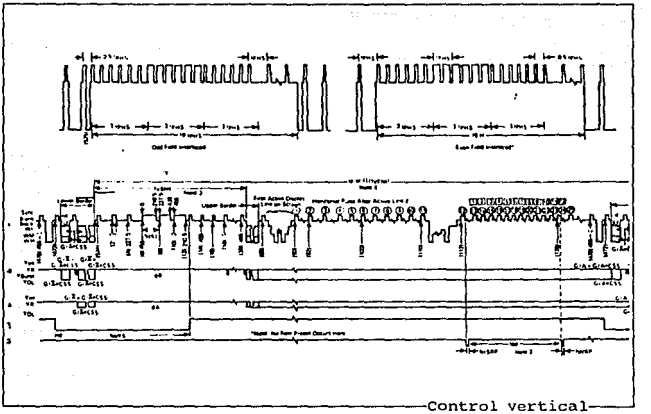
Este subsistema de el VDG incluye el modo de sincronización en la generación de la decodificación, asociada a una línea lógica del contador, y emplea una frecuencia de 3.58 MHz para generar la información horizontal y vertical, en el video y croma se emplea para generar los colores e información de video. El tiempo horizontal para el VDG se muestra en la gráfica de tiempos siguiente.



Control horizontal

Diez y medio ciclo de los 3.58 MHz son transmitidos sobre la parte de cada período horizontal de retroceso, el color es suprimido durante la sincronía vertical y a intervalos regulares. Tonalidades de color pueden ser suprimidas en gráficos de gran densidad que empleen dos colores, este manejo puede dar interesantes efectos de arcoiris sobre la pantalla.

Tiempos para la sección vertical del VDG se muestran a continuación.



Control vertical

Un nuevo trazo vertical, es iniciado por la señal de luminancia correspondiente a el nivel de borrado. Este período vertical de borrado comienza con tres líneas de pulsos de compensación seguidos de tres líneas verticales de sincronía de forma de diente de sierra seguidos de otras tres líneas de pulsos de compensación. Una parte del período de borrado vertical contiene los pulsos de sincronía horizontal, las líneas de compensación y los dientes de sierra son de una frecuencia de media línea. Nótese la diferencia de espacio entre el último pulso de sincronía horizontal y el primer pulso de compensación en el plano y campo impar. La mitad de la línea es diferente entre campos y cuando se hace el entrelazo de estos,

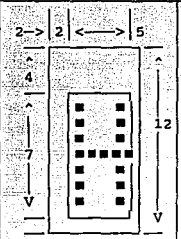
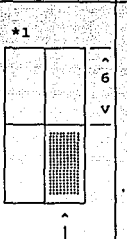
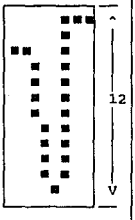
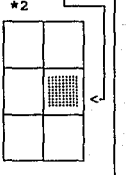
formarán la pantalla.

La frecuencia de 3.58 MHz para el color es usada para incrementar a el contador.

Modos de despliegue en la pantalla.

Se tienen dos maneras de lograr esto en el VDG llamados modos mayores. El modo mayor 1 contiene 4 alfanuméricos y dos modos limitados de gráficos, el modo mayor 2 contiene 8 modos de gráficos, de estos, 4 pueden emplear completamente los colores, los otros 4 tienen restricciones en los colores. Los modos de selección en modo mayor 2 junto con la resolución y la cantidad de memoria necesaria se dan en las siguiente tabla.

La forma como se componen cada elemento en modo mayor 1 en la modalidad de alfanumérico o semigráfico, al igual que los colores disponibles y la cantidad de memoria requerida se muestran a continuación:

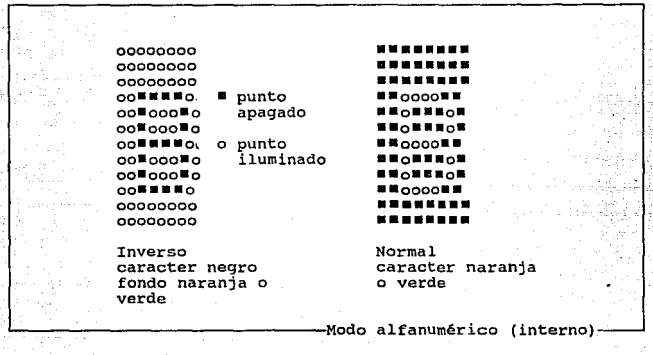
Modo	Memoria	Elemento	Colo-res	Elemento	Colores
Alfanumé-ricos (inter-nos) Semigrá-ficos 4 (*1)	512x8		2		8
Alfanumé-ricos (exter-nos) Semigrá-ficos 6 (*2)			2		4

Modo mayor 1

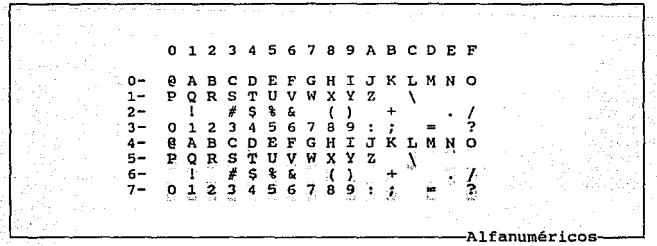
Modo	Memoria	Colores	Resolución
64x64 Color	1kx8	4	Matriz de 64x64 elementos
128x64	1kx8	2	Matriz de 128 de ancho por 64
128x64 Color	2kx8	4	elementos de alto
128x96	1.5kx8	2	Matriz de 128 de ancho por 96
128x96 Color	3kx8	4	elementos de alto
128x192	3kx8	2	128 elementos de ancho por
128x192 Color	6kx8	4	192 de alto
256x192	6kx8	2	256 elementos de ancho por 192 de alto

Modo mayor 2

En el modo mayor 1 la ventana en donde se despliega el alfanumérico está dividida en 32 columnas y 16 renglones, los cuales requieren de 512 bytes de memoria, cada elemento del carácter tiene 8 medios del período por 12 líneas desplegadas, como se muestra en seguida, el área fuera del alfanumérico es negra.

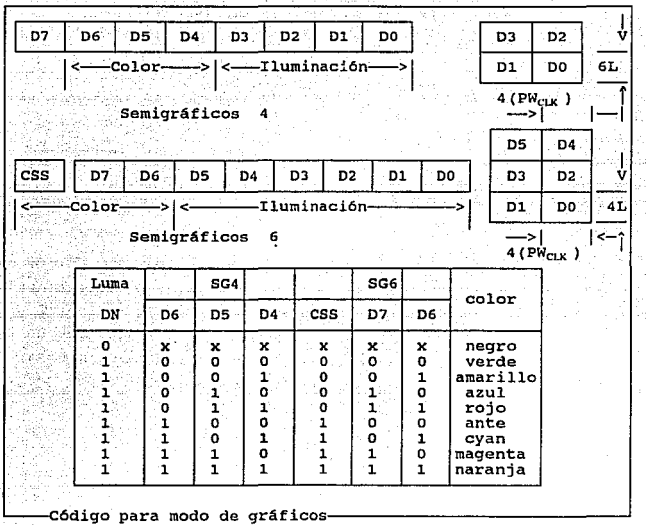


El VDG construye los caracteres, contenidos en la ROM, estos son 64 caracteres ASCII en un formato de 5X7,como se observa en la siguiente figura



El caracter (de 5x7) está posicionado dentro del caracter elemental de 8x12, dos columnas a la derecha y tres renglones abajo. 6 bits de los 8 bits de la palabra de dato son usados por el generador interno de caracteres ASCII. Los dos bits restantes pueden ser usados para implementar el video inverso, encendido o cambio de color, o la sección externa del generador ROM de caracteres, que puede seleccionar sobre un caracter u otro. Si se desea desplegar letras minúsculas, caracteres especiales, o gráficas limitadas, un ROM externo puede ser usado. Si tal ROM externo es usado, todos los elementos del cuadro de 8x12 o pixels, en los elementos del caracter pueden ser utilizados, los caracteres pueden ser; tanto verdes en un fondo verde obscuro o anaranjadas en un fondo anaranjado obscuro, dependiendo del estado del pin de CSS, invirtiéndolo puede ser usado para desplegar caracteres en un fondo brillante.

Los dos modos de gráficas limitadas son, semigráficas 4 y semigráficas 6. En semigráficas 4, el bloque de caracteres de puntos (dot) está dividido por dentro de 4 pixels (cada pixel tiene 4 y medio registros, por seis de líneas de reconocimiento o registro, los cuatro bits inferiores (DD0-DD3) de cada bits de datos, selecciona uno de los 16 posibles modelos de iluminación mientras que los siguientes 3 bits (DD4-DD6) determinan el color de los elementos iluminados, el bit más significativo no es usado. La siguiente figura muestra el color y los modelos posibles a seleccionar.



Luma		SG4			SG6		color
DN	D6	D5	D4	CSS	D7	D6	
0	x	x	x	x	x	x	negro
1	0	0	0	0	0	0	verde
1	0	0	1	0	0	1	amarillo
1	0	1	0	0	1	0	azul
1	0	1	1	0	1	1	rojo
1	1	0	0	1	0	0	ante
1	1	0	1	1	0	1	cyan
1	1	1	0	1	1	0	magenta
1	1	1	1	1	1	1	naranja

En las semigráficas 6, el bloque de caracteres de puntos de 8x12 está dividido dentro de seis pixels, cada cuatro renglones y medio por cuatro líneas de reconocimiento. Con los seis bits menos significativos de cada byte de los datos que llegan, se selecciona uno de los 64 modelos posibles de iluminación, mientras que el CSS de entrada y los bits de mayor orden (DD6-DD7) determinan el color de los elementos que estarán iluminados

La pantalla de despliegue en el modo mayor 2 (gráficas completas) tiene un formato menos riguroso que en el modo principal 1. Los elementos que son desplegados varían de una línea de reconocimiento o registro a tres líneas de reconocimiento en altura. La longitud del elemento desplegado es de 8 o 16 medios períodos de amplitud, cada elemento desplegado está dividido entre 4 u 8 pixeles. Como los modos semigráficos se seleccionan los datos que serán

iluminados, cuando está en alto el pixel es iluminado con el color elegido por "CSS" (color set select), cuando está en bajo el pixel es negro.

En modos de colores completos, pares de bits de datos eligen uno de los cuatro colores definido por el pin del CSS, dependiendo del estado de CSS, el área de afuera de la ventana de despliegue puede ser verde o ante. Los formatos de despliegue y selección de colores para este modo mayor están en la figura rotulada con "Modo alfanumérico (interno)".

Modo de gráficas de colores uno (CG1) de 64x64

El modo de gráficas de colores de 64x64 genera una matriz de despliegue de 64 elementos de ancho por 64 elementos de alto, cada elemento podría ser uno de los cuatro colores. Se requiere una memoria de despliegue de 1kx8. Para desplegar el contenido de la RAM es necesario tener acceso a ella 16 veces por cada línea horizontal. Cada pixel es igual a dos y medio ciclos de reloj por tres líneas de reconocimiento.

Modo de gráficas de resolución uno, de 128x64 (RG1)

El modo de gráficos de 128x64 una matriz de 128 elementos de amplitud por 64 elementos de alto. Cada elemento puede estar encendido o apagado, sin embargo, el despliegue completo puede ser de uno de los dos colores, seleccionado por el pin de CSS. Se requiere una memoria de despliegue de 1kx8. A la memoria RAM de despliegue se le tiene acceso 16 veces por línea horizontal, cada pixel requiere de 2.5 ciclos de reloj, por las tres líneas de reconocimiento.

Modo de gráficas de colores dos (CG2) de 128x64

El modo de gráficas de colores de 128x64 genera una matriz de despliegue de 128 elementos de amplitud por 64 elementos de alto. Cada elemento puede ser uno de los cuatro colores. Se requiere una memoria de despliegue de 2kx8. A la RAM de despliegue se le tiene acceso 32 veces por línea horizontal. Cada pixel requiere de 2.5 ciclos de reloj por las 3 líneas de reconocimiento.

Modo de gráficas de resolución 2 (RG2) de 128x96

El modo de gráficas de 128x96, genera una matriz de despliegue de 128 elementos de ancho por 96 elementos de alto, cada elemento puede estar encendido o apagado, sin embargo, el despliegue completo podría ser uno de los dos colores seleccionados por el CSS. Se requiere una memoria de despliegue de 1.5kx8. La RAM de despliegue se le tiene acceso 16 veces por cada línea horizontal. Cada pixel requiere de 2.5 ciclos de reloj por 2 líneas de reconocimiento.

Modo de gráficas de colores 3 (CG3) de 128x96

El modo de gráficas de colores de 128x96, genera un despliegue de 128 elementos de ancho por 96 elementos de alto cada elemento puede ser uno de los cuatro colores que pueden ser seleccionados por el CSS. Se requiere una memoria de despliegue de 3kx8. La RAM de despliegue se le tiene acceso 32 veces por cada línea horizontal. Cada pixel tiene 2.5 ciclos de reloj, además de dos líneas de reconocimiento.

Modo de gráficas de resolución 3 (RG3) de 128x192

El modo de gráficas de 128x192, genera una matriz de 128 elementos de ancho por 192 elementos de alto, cada elemento puede estar encendido (ON) o apagado (OFF), y el elemento ON puede ser uno de los 2 colores seleccionado por CSS. Se requiere una memoria de despliegue de 3kx8. A la RAM de despliegue se le requiere 16 veces por línea horizontal. Cada pixel requiere de 2.5 ciclos de reloj por 1 línea de reconocimiento.

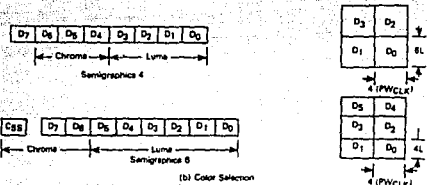
Modo de gráficas de colores 6 (CG6) 128x192

El modo de gráficas de colores de 128x192, genera un despliegue de 128 elementos de ancho por 192 de alto, cada elemento puede ser uno de los cuatro colores posibles. Se requiere de una memoria de despliegue de 6kx8. A la RAM de despliegue se le tiene acceso 32 veces por línea horizontal, se requieren de 2.5 ciclos de reloj para cada pixel y 1 línea de reconocimiento.

Modo de gráficas de resolución 6 (RG6) 256X192

El modo de gráficas de 256x192 genera un despliegue de 256 elementos de ancho por 192 elementos de alto. Cada elemento puede estar en ON (encendido) o OFF (apagado), y el elemento ON podría ser uno de los dos colores seleccionados por CSS. Se requiere una memoria de despliegue de 6kx8. A la RAM de despliegue se tiene acceso 32 veces por línea horizontal. para cada pixel se tienen 2.5 por cada línea de reconocimiento.

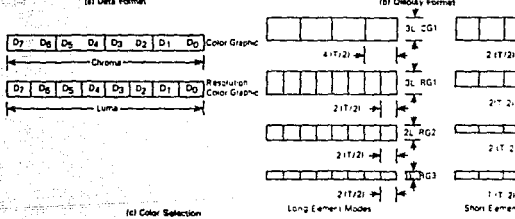
Para la selección del modo de gráfico CG o RG se da la siguiente tabla.



(b) Color Selection

Luma DN	SG4 D6	SG4 D5	SG4 D4	CSS	SG6 D7	SG6 D6	Color
0	X	X	X	X	X	X	Black
1	0	0	0	0	0	0	Green
1	0	0	1	0	0	1	Yellow
1	0	1	0	0	1	0	Blue
1	0	1	1	0	1	1	Red
1	1	0	0	1	0	0	Buff
1	1	0	1	1	0	1	Cyan
1	1	1	0	1	1	0	Magenta
1	1	1	1	1	1	1	Orange

FIGURE 22 – GRAPHIC MODE ENCODING



(c) Color Selection

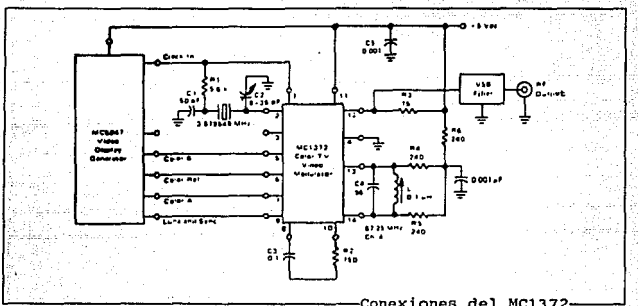
CSS	Border	Resolution Color Mode D_N	Resolution Color Mode Color	Color Mode D_{N-1}	Color Mode D_N	Color Mode Color
0	Green	0	Black	0	0	Green
0	Green	1	Green	0	1	Yellow
0	Green	1	Green	1	0	Blue
0	Green	1	Green	1	1	Red
1	Buff	0	Black	0	0	Buff
1	Buff	1	Buff	0	1	Cyan
1	Buff	1	Buff	1	0	Magenta
1	Buff	1	Buff	1	1	Orange

Selección del Modo Gráfico

Circuito Modulador de Video MC-1372

Descripción de operación.

El diagrama externo del modulador de video se da a continuación:



Conexiones del MC1372

En seguida se dará el funcionamiento de las terminales del modulador MC1372.

Oscilador de entrada (terminal 2 de entrada) Oscilador de el subportador del color para la realimentación de la entrada, señal proveniente del reloj de salida, es externamente cambiada de fase y una componente de AC es acoplada en este pin.

Referencia de color (terminal 6 de entrada) El voltaje de DC acoplado a esta terminal establece el voltaje de referencia, al cual las entradas del color A y B son comparadas.

Modulador de chroma (terminal 8 de salida) Salida de baja impedancia (emisor seguidor) proporciona la suma vectorial de los moduladores de chroma A y B.

Reloj de salida (terminal 1 de salida) Proporciona una onda cuadrada con frecuencia igual a la del oscilador de la subportadora

de crominancia, esta salida es capaz de manejar una carga LS-TTL.

Ajuste del ciclo de trabajo (terminal 3 de entrada) Un voltaje de DC acoplado a este pin ajusta el ciclo de trabajo de la señal de salida del reloj, si se deja desconectado, el ciclo de trabajo es de aproximadamente del 50%.

Entrada B de color (terminal 5 de entrada) Un voltaje de DC de entrada acoplado al modulador de chroma B, cuya fase la dirige el modulador A por aproximadamente 100°, la amplitud y polaridad de la salida del modulador corresponden a la diferencia de voltaje entre esta terminal y el voltaje del color de referencia del pin 6.

Entrada A de color (terminal 7 de entrada) Entrada de DC acoplada a el modulador A de chroma, y retrasa la fase de B por aproximadamente 100°, la amplitud y la polaridad de la salida del modulador corresponden a la diferencia de voltaje entre este pin y el voltaje de referencia del pin 6.

Luminancia (terminal 9 de entrada) Entrada para el modulador de RF. Este pin acepta una componente de DC acoplada a la luminancia y señal de sincronía. La amplitud de la señal de RF de salida, se incrementa con un voltaje positivo aplicado a esta entrada, y el voltaje promedio resulta ser igual a cero (una modulación de 100%), una señal con cambio positivo de la sincronía debe ser usada.

Crominancia (terminal 10 de entrada) Entrada para el modulador de RF.este pin acepta un voltaje de AC acoplado a la crominancia, proporcionado por la salida del modulador de chroma (pin 8), la señal es atenuada por un divisor de resistencias interno antes de que sea aplicado a al modulador de RF.

Modulador de RF (terminal 12 de salida) La salida modulada proviene de dos colectores, la impedancia de salida y la ganancia pueden ser seleccionadas por medio de la elección de una resistencia conectada entre este pin y una fuente de DC.

Tierra (terminal 4)

V_{cc} (terminal 11)

RF [tanque] (terminal 13 y 14 de entrada circuito tanque) Un circuito sintonizado conectado entre estos pin determina la frecuencia del oscilador de RF. El circuito sintonizado debe proporcionar una baja resistencia para el voltaje de DC. Aplicando un voltaje de referencia entre estos pins da como resultado una banda base para el video en la salida del modulador.

En la siguiente grafica se puede observar un diagrama a bloques del modulador.

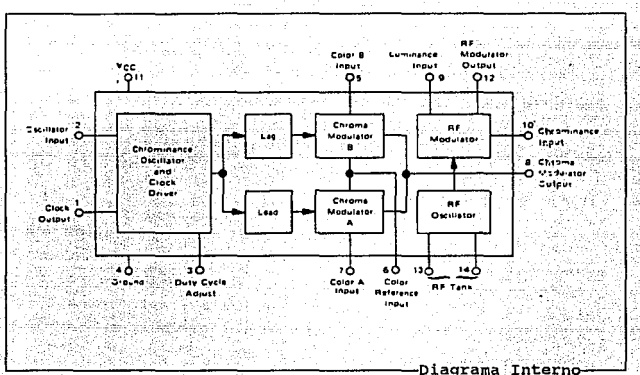


Diagrama Interno

II. e Características y Funcionamiento del PTM MC-6840.

Introducción

El MC6840 PTM (Programer Timer Module) Provee un sistema variable de intervalos de tiempo, que puede ser usado como sistema generador de interrupciones, generador de señales de salida y puede ser empleado para medir frecuencia, conteo de eventos, medición de intervalos y tareas similares.

Un diagrama a bloques del PTM se muestra a continuación:

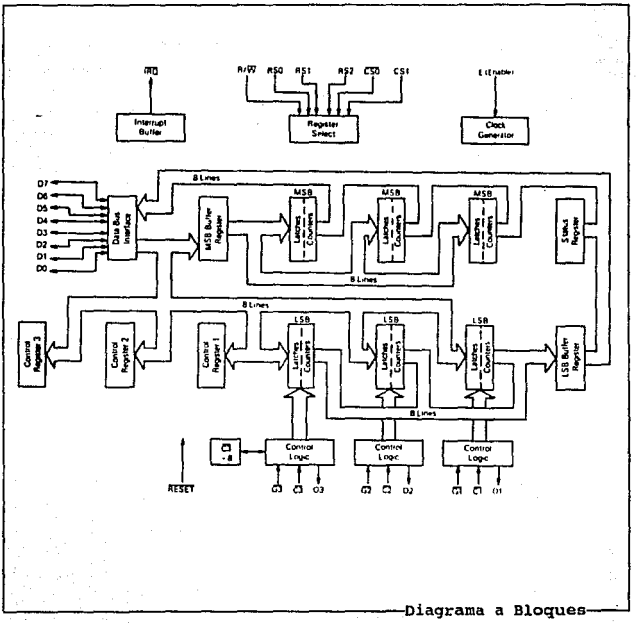


Diagrama a Bloques

El PTM posee tres contadores binarios de dieciséis bits, con sus respectivos registros de control y un registro de estado y es capaz de realizar tareas como:

Medición de Frecuencia.
Conteo de Eventos.
Medición de Intervalos.
Generador de Onda Cuadrada.
Generar Señales de Retraso.
Generar Pulsos de Duración Controlada.
Generar Pulsos de Duración Modulada.

Los contadores tienen registros de almacenamiento, entradas de reloj independientes y el circuito de comparación y habilitación necesarios para implementar las funciones antes mencionadas.

Registros del PTM

El PTM ocupa ocho localidades del mapa de direcciones del MPU, estas ocho localidades junto con el bit cero (CR20) del registro de control dos (CR2) y la línea de lectura/escritura (R/W), son empleados para direccionar todos los registros internos del PTM.

Dos de estos registros son compartidos por los tres contadores, y pueden seleccionarse con diferentes direcciones, lo que diferencia su función es la secuencia con que sean empleados, estos son: El registro del byte mas significativo (MSB) de los contadores en escritura y el registro del byte menos significativo (LSB) en la lectura.

El MSB en el mapa de direcciones de escritura siempre va seguido por un registro de contador, al igual que en la lectura el LSB va precedido por otro, como se mostró en la tabla de registros del PTM. Esto se debe a que si se realiza una lectura o escritura de 16 bits en el PTM, esta corresponda a una lectura o carga de un contador.

Por ejemplo para una escritura al contador tres, primero se escribe al MSB y posteriormente al registro de contador tres, de esta forma se cargan los 16 bits del contador tres.

Para facilitar el empleo de estos registros su direccionamiento se repite, no es que sean varios registros, es uno mismo.

Registro de Control.

El PTM posee tres registros de control, uno para cada contador, estos registros de 8 bits tienen casi las mismas funciones, diferenciados solamente por su correspondiente bit cero, este bit tiene una función diferente dependiendo del contador. Se explicará la función que cada bit desempeña, independientemente del contador que se trate a excepción del bit cero.

Bit 0 (CRX0) Tiene las siguientes funciones dependiendo del contador al que pertenezca:

CR10 Bit de reinicio interno. Cuando el bit menos significativo del registro de control uno (CR1) se encuentra en estado bajo, todos los contadores operan de acuerdo a los restantes bits de los registros de control. Escribiendo un uno en esta localidad, los contadores son reiniciados con el contenido de los respectivos latches de los temporizadores, las entradas de reloj son deshabilitadas, y las salidas de los contadores y las banderas de interrupciones son reiniciadas.

CR20 Bit de selección de registro. De acuerdo al estado de este bit se podrá seleccionar al registro de control uno o tres. Debido a que los registros de control uno y tres ocupan la misma localidad, el bit menos significativo del registro de control dos se emplea para decodificar a uno u otro. En un estado bajo, se tendrá acceso al registro de control tres (CR3) y con un estado alto se direccionará al registro de control uno (CR1). Cuando se ha reiniciado al MPU, por medio de la señal de RESET, todos los bits de los registros de control son limpiados (puestos en cero lógico), con esto se tiene acceso al registro de control tres. Una forma recomendable de programación de inicio es: programar al contador tres, al contador dos y finalizar con el contador uno, con esta secuencia solo se modifica el bit CR20 una sola vez.

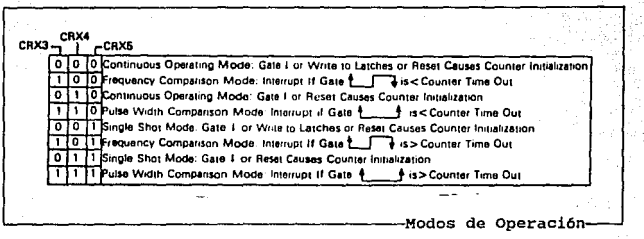
CR30 Bit de control de reloj. El contador tres como se mencionó, cuenta con un divisor entre ocho a la entrada del reloj, y con este bit se puede seleccionar. En estado bajo el contador funciona normalmente, es decir las señales de reloj pasan directamente al contador. Cuando CR30 esta en alto, la señal de reloj pasa por el divisor entre ocho antes de llegar al contador.

Bit 1 (CRX1) Origen del Reloj. El estado de este bit indica la procedencia de la señal de reloj del contador, interna o externa. Si se tiene un cero lógico, la fuente de la señal de reloj es externa, proveniente de la línea CX. En estado alto, el contador emplea la señal de habilitación (E) como entrada de reloj.

Bit 2 (CRX2) Control del Modo de Conteo. Este bit selecciona la forma en que será tratada la información contenida en los latches del correspondiente temporizador. Si se manejará como una palabra de 16 bits o como dos bytes de 8 bits. En el modo de una palabra de 16 bits, un cero lógico en este bit, el contador será decrementado hasta cero, N+1 veces el período de la señal de reloj, donde N es definido como un número de 16 bits en el contador. Con el CRX2 en alto, el contador se decrementará (L+1)*(M+1) veces el período de la señal de reloj, donde L y M se refieren, respectivamente, al byte menos significativo (LSB) y al byte mas significativo (MSB) contenidos en el temporizador.

Como se manejen los contadores es importante cuando se desea generar señales, ya que de acuerdo a esto, estará dada la forma de dicha señal.

Bit 3, 4 y 5 (CRX3, CRX4 y CRX5) Control de Interrupción y Modo de Operación. La función que realiza cada uno de estos bits, se explicará mas adelante en la sección de Modo de Operación de los Temporizadores. Una síntesis de las funciones que se realizan se muestra en la siguiente tabla:



Bit 6 (CRX6) Habilitación de Interrupción. El estado de este bit habilita o inhibe la generación de interrupciones. En estado bajo de CRX6 es empleado como máscara para las interrupciones, mientras que en uno lógico, la generación se habilita.

La causas que generan interrupciones son: finalización de la cuenta de los temporizadores en los modos de generación de señales(contadores en cero), tanto en el continuo como en el disparo simple, o incumplimiento de una condición en el modo de comparación, tanto de frecuencia como de ancho de pulso, estos modos de operación se explican más adelante.

Bit 7 (CRX7) Habilitación de la Salida del Contador. Este bit se emplea para enmascarar la respectiva salida del contador OX. En estado bajo, la terminal OX, permanecerá en alto y deshabilitada. Cuando CRX7 tiene un uno lógico, se habilita la salida para la generación de señales en el modo de operación simple o continuo.

Registro de Estado.

El PTM tiene un registro de estado de solo lectura, el cual contiene cuatro banderas de interrupción. Los restantes cuatro bits del registro no son usados y permanecen en cero cuando son leidos. Los bits 0, 1 y 2 son asignados a los temporizadores 1, 2 y 3 respectivamente como los bits de las banderas individuales, mientras que el bit 7 es la bandera compuesta de interrupción CIF (Composite Interrupt Flag). Esta bandera será activada si alguna de las banderas individuales se activo, esto será, si y sólo si, el

bit 6 de su respectivo registro de control esta en uno lógico y se ha presentado alguna causa de generación de interrupción. La condición para que se active CIF puede ser expresada por:

$$INT = I1*CR16 + I2*CR26 + I3*CR36$$

donde:

INT = Bandera compuesta de interrupción CIF (bit 7)
I1 = Bandera de interrupción del contador 1 (bit 0)
I2 = Bandera de interrupción del contador 2 (bit 1)
I3 = Bandera de interrupción del contador 3 (bit 2)

Una interrupción es limpiada por una condición de reinicio, la cual puede provocarse por: línea de RESET=0 o Bit de Reinicio Interno (CR10)=1. Al igual puede ser limpiada por medio de una lectura al temporizador que la origino, previendo que antes de ésta, se realizó una lectura al registro de estado. Esta condición de lectura del registro de estado-lectura del temporizador, es diseñada para prever perdidas de interrupciones, las cuales son generadas despues de que se leyo el registro de estado, pero antes de leerse el temporizador.

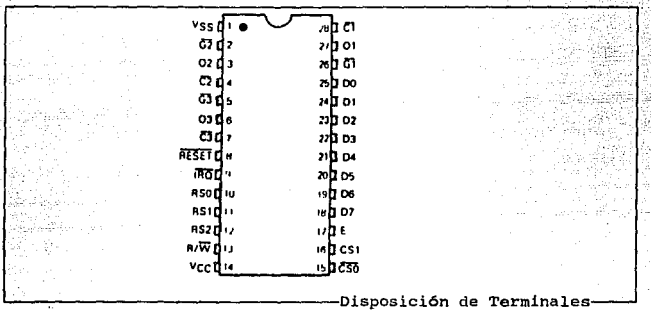
Una interrupción individual también puede ser limpiada por una escritura a los latches del temporizador, o por una secuencia de inicialización del mismo, previendo que tanto la escritura como la inicialización afecten al temporizador correspondiente a la bandera de interrupción individual.

Inicialización del Temporizador.

Una inicialización es producida cuando el contenido en los latches de los temporizadores es transferida al contador. Esta transferencia puede producirse por una escritura proveniente del MPU, la activación de la señal de entrada (GX) o un reinicio; según se haya programado en el correspondiente registro de control. Un reinicio puede ser producido por: línea de RESET=0, Bit de Reinicio Interno (CR10)=1 o reinicio del contador; este último ocurre, cuando una transición negativa (flanco de bajada) del reloj de entrada al contador, es reconocida despues que el contador ha alcanzado un estado de vacio (contenido del contador cero). En este caso, el dato es transferido de los latches al contador.

Asignación de Terminales.

El 6840 es un circuito de 24 terminales, cada una de ellas con una función específica, la asignación de terminales en el circuito integrado se muestra en el siguente cuadro:



Disposición de Terminales

A continuación se hará una breve explicación de la función que realiza cada terminal. Para ello, estas terminales se pueden dividir en dos grupos, de acuerdo a la función que realizan, estos grupos son interfaz de entrada/salida e interfaz de control.

Interfaz de Entrada/Salida.

La interfaz entre el PTM y el MPU consta de un bus de datos bidireccional, una línea de lectura/escritura, una línea de petición de interrupción, dos líneas de selección de chip, tres líneas de selección de registros internos, una línea de habilitación y una línea de inicialización.

Habilitador del PTM [E] (terminal 17) La terminal E, señal de habilitación, es una entrada de alta impedancia compatible con TTL, ésta habilita a los buffers del bus de datos de entrada/salida para sincronizar la transferencia de información con el MPU y puede funcionar a la vez como entrada de reloj para los contadores.

Lectura/Escritura [R/W] (terminal 13) La línea de lectura/escritura es usada para el control de la dirección del flujo de datos a través de la interfaz del bus de datos de entrada/salida del PTM.

Cuando R/W es alto (el MPU realiza un ciclo de lectura), los manejadores de salida del PTM son encendidos y un registro interno seleccionado es leído.

Cuando ésta es baja, los manejadores de salida del PTM son abiertos y el MPU escribe en un registro seleccionado; por lo tanto, como la señal de R/W es empleada para seleccionar registros de sólo-escritura o sólo-lectura dentro del PTM, no es posible realizar operaciones de lectura modificación y escritura con dichos registros.

Petición de Interrupción [IRQ] (terminal 9 activa baja) Petición de interrupción es una salida con drenador abierto (open-drain, no posee pullup interno), ésta se emplea para interrumpir al MPU. La salida IRQ permanece baja mientras que la causa de la interrupción esta presente y la correspondiente habilitación de interrupción, bit 7 CIF (Composite Interrupt Flag) del Registro de Estado este activado.

Selección del PTM [CS0 y CS1] (terminales 15 y 16) Estas dos líneas son empleadas para direccionar al PTM. El PTM es direccionado cuando CS1 está en alto y CS0 es bajo. Las transferencias de datos hacia el PTM o provenientes de él son desempeñadas bajo el control de las señales de habilitación (E), lectura/escritura (R/W) y selección de registro.

Selección de Registro [RS0, RS1 y RS2] (terminales 10, 11 y 12) Las líneas de selección son usadas en conjunto con la de R/W para seleccionar un registro interno, contador y latches como se muestra en la siguiente tabla.

Register Select Inputs			Operations	
RS2	RS1	RS0	R/W = 0	R/W = 1
0	0	0	CR20 = 0 Write Control Register #3 CR20 = 1 Write Control Register #1	No Operation
0	0	1	Write Control Register #2	Read Status Register
0	1	0	Write MSB Buffer Register	Read Timer #1 Counter
0	1	1	Write Timer #1 Latches	Read LSB Buffer Register
1	0	0	Write MSB Buffer Register	Read Timer #2 Counter
1	0	1	Write Timer #2 Latches	Read LSB Buffer Register
1	1	0	Write MSB Buffer Register	Read Timer #3 Counter
1	1	1	Write Timer #3 Latches	Read LSB Buffer Register

Selección de Registros

Bus de Datos Bidireccional [D_0-D_7] (terminales 25 a 18) Las líneas bidireccionales de datos (D0-D7) permiten la transferencia de datos entre el PTM y el MPU. Las salidas del bus de datos son manejadas por dispositivos tres-estados, éstos las mantienen en estado de alta impedancia (off), excepto cuando el MPU desempeña en el PTM operaciones de lectura.

Inicialización [Reset] (terminal 8) La línea de RESET necesita dos ciclos de la señal de E para aceptar una señal de reinició y es procesada hasta el tercer ciclo, si el RESET es asíncrono se requiere de otro ciclo adicional. El RESET provoca:

- Todos los registros (latches) de los contadores son puestos en la cuenta máxima.
- Todos los bits de los registros de control son limpiados, excepto de CR10 el cual es activado (Internal Reset Bit).
- Todos los contadores son puestos al contenido de los registros (latches).
- Todas las salidas de los contadores son restablecidas y todos los relojes de los contadores son deshabilitados.
- Todos los bits del Registro de Estado son limpiadas (banderas de interrupción).

Interfaz de Control.

Cada uno de los tres contadores internos del PTM posee tres líneas de control, cada una con determinada función: una entrada para reloj externo, una entrada de control, y una salida del contador. Las entradas son de alta impedancia, compatibles con TTL y la salida es capaz de soportar dos cargas TTL.

Entradas de Reloj [C1, C2 y C3] (terminales) Las terminales de entrada C1, C2 y C3 aceptan señales con niveles de voltaje TTL asíncronos, los cuales decrementan los contadores 1, 2 y 3 respectivamente. Los niveles alto y bajo del reloj externo deben de ser estables cuando menos el tiempo de duración de la señal de habilitación E (un período del reloj del sistema), mas la suma de los tiempos de actualización (*setup*) y retén (*hold*) del reloj de entrada.

La entrada del reloj externo es tomada por los pulsos de la señal de habilitación, para que un pulso de reloj sea sincronizado y procesado se requieren de tres períodos de la señal de habilitación. En el cuarto pulso de la señal de habilitación el contador es decrementado. Esto no afecta la frecuencia del reloj externo, solamente provoca un retraso entre la transición del reloj y el reconocimiento de la misma por parte del PTM.

El contador tres presenta un caso especial, ya que cuenta a la entrada del reloj con un divisor entre 8, el cual puede ser seleccionado mediante la programación adecuada de su registro de control. El divisor entre ocho consta de un contador asíncrono, y en el caso de ser seleccionado, los tiempos de actualización y de retén no son aplicados; por lo que siempre y cuando se respete un tiempo mínimo de duración del pulso de entrada todas las transiciones del reloj serán reconocidas. Sin embargo para garantizar que el pulso de reloj sea reconocido en el presente ciclo de habilitación, es necesario que presente un cierto tiempo de sincronización entre la transición del reloj y el flanco de caída de la señal de habilitación, en caso contrario, al desconocerse el tiempo de sincronización, es posible que la transición de la terminal C3 no sea procesada sino hasta el siguiente pulso de habilitación.

Entradas de Control [G1, G2 y G3] (terminales) Las entradas G1, G2 y G3 aceptan señales compatibles con TTL asíncronas, las cuales son usadas para el disparo o inicio/reinicio de las funciones de los contadores 1, 2 y 3 respectivamente. Las señales son registradas por la señal de habilitación, en forma similar a las del reloj. En forma más detallada, estas entradas se emplean para la medición de intervalos de tiempo, frecuencia y para el disparo de inicio en la generación de señales.

Estas entradas de los temporizadores afectan directamente el funcionamiento de los contadores de 16 bits, por lo que la operación del G3 es independiente del divisor entre ocho.

Salidas de Contador [O1, O2 y O3] (terminales) Las terminales de salida de los contadores O1, O2 y O3 son capaces de manejar dos cargas TTL y producir una señal a la salida en cualquiera de las dos modalidades que posee, disparo sencillo o modo continuo. La forma de onda se define de acuerdo a la programación realizada en el registro de control correspondiente, seleccionando un contador sencillo de 16 bits o bien dos contadores de 8. Un contador sencillo de 16 bits produce una señal cuadrada tanto en modo de disparo sencillo como en el modo continuo. Seleccionando dos contadores de 8 bits, se pueden generar señales con ciclo de trabajo variable en ambos modos de operación. Un bit de cada registro de control (bit 7) es empleado para habilitar su correspondiente salida, en caso de estar desactivado el bit (0), la salida permanecerá baja a pesar del modo de operación. Si el bit es limpiado estando en operación y en nivel alto la señal de salida, esta ira a bajo en el primer pulso de habilitación seguido a una escritura al registro de control.

Modos de Operación de los Temporizadores.

El PTM ha sido diseñado para trabajar en una amplia variedad de aplicaciones. Estas son ejecutadas mediante el uso de tres bits de cada registro de control (CRX3, CRX4 y CRX5) que definen diferentes modos de operación de los temporizadores. Estos modos son divididos en: Generación de Señales y Comparación de Señales. Como se muestra

en la siguiente tabla:

Registro de Control			Modo de Operación	
CRX3	CRX4	CRX5		
0	*	0	Continuo	Generación
0	*	1	Disparo Simple	
1	0	*	Frecuencia	Comparación
1	1	*	Ancho de Pulso	

* Define la seleción de una funcion adicional.

Uno de los modos de operación de generación de señales, el continuo, es empleado para producir señales periódicas, las cuales pueden tener el ciclo de trabajo simétrico o variable. El otro modo, disparo simple, un pulso único con ancho programable es generado.

Los modos de comparación, incluyen el de frecuencia y ancho de pulso, los cuales son empleados para medir pulsos cíclicos o únicos respectivamente.

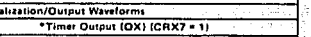
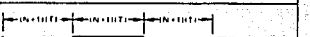
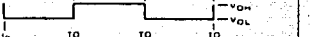
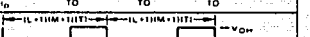
Generación Continua.

El modo de generación continuo producirá una señal ininterrumpida con un período proporcional al numero almacenado en uno de los latches de los temporizadores. Cualquiera de los temporizadores, puede programarse para operar en este modo (con los bits 3 y 5 del registro de control igual a cero).

Este modo de operación, como se mencionó, tiene dos variantes: generación de la señal con ciclo de trabajo simétrico o con ciclo variable. Esto se realiza según el modo de conteo que se haya seleccionado para el contador (contador de 16 bit o dos contadores de 8 bits), por medio del CRX2. En el modo de contador simple de 16 bits, si se toma a N como un numero de 16 bits, con el cual se carga el contador, la señal producida tendra un ciclo de trabajo del 50%, cuya duración será (N+1)*T, donde T es el periodo de la señal de reloj, es decir, la señal tendra un período de 2*(N+1)*T.

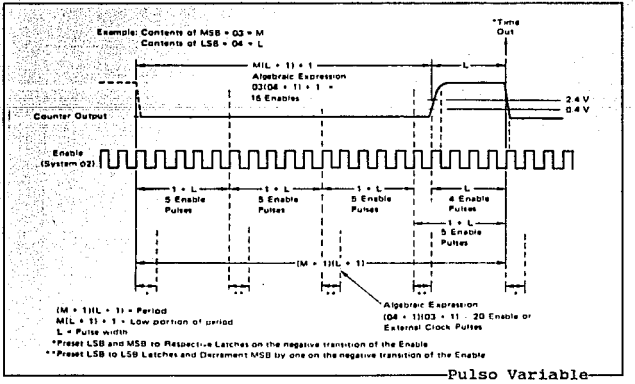
En el modo de dos contadores de 8 bits, si se tiene que con M y L se carga el contador, donde M es el contenido del byte más

significativo, y L es el menos significativo; la señal tendrá un pulso cuyo ancho estará dado por L*T y un período de (L+1)*(M+1)*T, es decir un ciclo de trabajo igual a 1/[2*M+M/L+1]. Esto se expresa en la siguiente tabla:

Synthesis Modes		CONTINUOUS MODE (CRX3 = 0, CRX5 = 0)	
Control Register		Initialization/Output Waveforms	
CRX2	CRX4	Counter Initialization	*Timer Output (OX) (CRX7 = 1)
0	0	$\overline{G}_i$+W+R	
0	1	$\overline{G}_i$+R	
1	0	$\overline{G}_i$+W+R	
1	1	$\overline{G}_i$+R	

Modo de Operación Continuo

Un ejemplo de aplicación se muestra a continuación.



Pulso Variable

Con este último modo se pueden producir señales con mudulación de ancho de pulso.

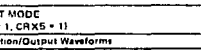
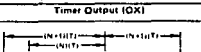
Generación Disparo Simple.

El modo de generación simple producirá una señal discontinua con un ancho de pulso proporcional al número almacenado en uno de los latches de los temporizadores.

En este modo de operación también se tienen dos variantes; ambas son muy semejantes, como se vera, la diferencia se estipula que en un modo se puede alcanzar un ancho de pulso mayor que en el otro, y que el pulso se genera con un retraso igual al período de la señal de reloj, invariablemente del valor con que se carge, mientras que en el otro modo, se puede programar el numero de ciclos de retraso que se deseen, aun cuando no se pueden alcanzar los tiempos que el anterior.

Con el contador simple de 16 bits, la duración del pulso sera de N*T, este pulso se generará con un retraso de un período de la señal de reloj.

En el modo de dos contadores de 8 bits, el pulso tendrá una duración de L*T, y un retraso dado por [(2*M*L)+M]*T. Esto se expresa en la siguiente tabla:

Synthesis Modes		SINGLE-SHOT MODE (CRX3 = 0, CRX7 = 1, CRX5 = 1)	
Control Register		Initialization/Output Waveforms	
CRX2	CRX4	Counter Initialization	Timer Output (OX)
0	0	$\overline{G}_i$+W+R	
0	1	$\overline{G}_i$+R	
1	0	$\overline{G}_i$+W+R	
1	1	$\overline{G}_i$+R	

Modo de Operación Disparo Simple

Comparación de Frecuencia y Ancho de Pulso.

Estos dos modos operan en forma muy semejante, se cargan los contadores con un valor X, se compara el tiempo que transcurre desde que se ha dado, ya sea el período de la señal o el ancho del pulso bajo; de acuerdo al estado de CRX4; con el tiempo en que el contador finalizo su cuenta; es posible que se genere una interrupción si este último es mayor o menor con respecto al otro, según se haya programado el bit CRX5. Para ello se emplean las líneas de control (GX) como entradas para las señales a comparar.

En ambos modos los contadores son cargados con un número X (donde X es igual a (N+1) en el modo de contador simple de 16 bits o igual a (M+1)*(L+1) en el modo de dos contadores de 8 bits).
La siguiente tabla muestra ambos modos y sus variantes.

CRX3 = 1			
CRX4	CRX5	Application	Condition for Setting Individual Interrupt Flag
0	0	Frequency Comparison	Interrupt Generated if Gate Input Period (1/F) is less than Counter Time Out (TO)
0	1	Frequency Comparison	Interrupt Generated if Gate Input Period (1/F) is greater than Counter Time Out (TO)
1	0	Pulse Width Comparison	Interrupt Generated if Gate Input "Down Time" is less than Counter Time Out (TO)
1	1	Pulse Width Comparison	Interrupt Generated if Gate Input "Down Time" is greater than Counter Time Out (TO)

Modo de Comparación de Señales

En el modo de comparación de frecuencia, los contadores se habilitan cuando se presenta una transición de alto a bajo en la entrada de control, con esto se empiezan a decrementar a cada período de la señal de reloj. Si CRX5 esta en bajo una interrupción será generada si el período de la señal de entrada (la siguiente transición de alto a bajo) es menor que el tiempo de la carga de los contadores, en caso contrario, los contadores se deshabilitarán al llegar a cero, y se repetira la acción de comparación con la siguiente transición. Si CRX5 esta en alto, la interrupción se generará si los contadores llegan a cero antes de que se presente la transición de alto a bajo de la señal de entrada.

En el modo de comparación de ancho de pulso, los contadores

también se habilitan cuando se presenta una transición de alto a bajo en la entrada de control. Si CRX5 esta en bajo una interrupción será generada si el tiempo de estado bajo de la entrada (la transición de bajo a alto) es menor que el tiempo de la carga de los contadores, en caso contrario, los contadores se deshabilitarán al llegar a cero, y se repetira la acción de comparación con la siguiente transición. Si CRX5 esta en alto, la interrupción se generará si los contadores llegan a cero antes de que se presente la transición de bajo a alto de la señal de entrada.

En el caso de que el pulso a comparar sea en estado alto, será necesario que se invierta la señal de entrada antes de llegar al PTM.

Consideraciones Eléctricas para el buen Manejo del PTM.

El PTM como todo dispositivo posee una serie de características de funcionamiento, las cuales deben ser respetadas para una adecuada operación. Las características más importantes se muestran en la siguiente tabla:

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage		V_{IH}	$V_{SS}+2.0$	–	V_{CC}	V
Input Low Voltage		V_{IL}	$V_{SS}-0.3$	–	$V_{SS}+0.8$	V
Input Leakage Current (V_{in} = 0 to 5.25 V)		I_{in}	–	1.0	2.5	μA
Hi-Z (Off) State Input Current (V_{in} = 0.5 to 2.4 V)	D0-D7	I_{TSI}	–	2.0	10	μA
Output High Voltage (I_{Load} = −205 μA) (I_{Load} = −200 μA)	D0-D7 Other Outputs	V_{OH}	$V_{SS}+2.4$ $V_{SS}+2.4$	– –	– –	V
Output Low Voltage (I_{Load} = 1.6 mA) (I_{Load} = 3.2 mA)	$\overline{IRQ}$, D0-D7 Q1-Q3	V_{OL}	– –	– –	$V_{SS}+0.4$ $V_{SS}+0.4$	V
Output Leakage Current (Off) State (V_{OH} = 2.4 V)	$\overline{IRQ}$	I_{LOH}	–	1.0	10	μA
Internal Power Dissipation (Measured at $T_A = T_L$)		P_{INT}	–	470	700	mW
Input Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	D0-D7 All Others	C_{in}	– –	– –	12.5 7.5	pF
Output Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	$\overline{IRQ}$ Q1, Q2, Q3	C_{out}	– –	– –	5.0 10	pF

Características Eléctricas

Consideraciones de Tiempos.

Por otra parte el PTM tiene características de funcionemiento, las cuales deben de considerarse para el diseño de la aplicación. Estas características se relacionan con los tiempos de las señales de entrada y salida del PTM. En la siguiente tabla se muestran algunas importantes que se deben de tener en cuenta.

BUS TIMING CHARACTERISTICS (See Notes 1, 2, and 3)

Ident. Number	Characteristic	Symbol	MC6840		MC68A40		MC68B40		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	9500	280	9500	210	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9500	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	50*	20	50*	20	50*	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Peripheral Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Peripheral Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

*The data bus output buffers are no longer sourcing or sinking current by t_{DHR} max (High Impedance)

NOTES:
1. Not all signals are applicable to every Part.
2. Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified.
3. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.

Tiempos de Señales

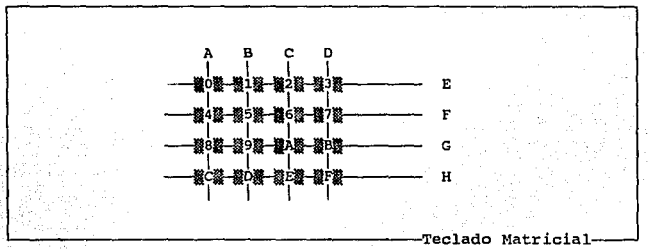
II.f Principio de Operación y Características de los Teclados para "PC".

Uno de los medios más comunes que se emplean para comunicarse del exterior a un sistema digital es un teclado. A continuación explicaremos el funcionamiento de éste.

Principio de Operación

El teclado se basa en el cierre y/o apertura de interruptores (teclas), los cuales están dispuestos en un arreglo especifico, dicho arreglo puede ser matricial o uno a uno. El disponer las teclas en arreglos es para poder definir que tecla fue presionada dentro del conjunto de teclas, esto se realiza mediante la decodificación de cada tecla, es decir, cada tecla posee un código único, este código se le impone según su localización dentro del arreglo.

En el arreglo matricial, las teclas forman una especie de red, en donde cada tecla ocupa una posición bien definida dentro de esa red, al ser presionada una tecla, es posible conocer, el renglón y la columna que ocupa, y así poder determinar que tecla fue la que se presionó, este tipo de disposición es empleado para disminuir el número de líneas empleadas para la decodificación de cada tecla, por ejemplo, si se tienen 16 teclas, se necesitarían sólo 8 líneas como se observa en la figura siguiente.



Esto se optimiza cuando el número de teclas es grande. El arreglo uno a uno es aquel que designa una línea de decodificación para cada una de las teclas, es decir, cada tecla posee una línea exclusiva, cuando es presionada, sólo por dicha línea se enviará el código, como se intuye para un teclado de 16 teclas se requiere de 17 líneas de decodificación (16 para las teclas y una común).

La mayoría de los teclados para PC's emplean el arreglo matricial.

El principio de operación empleado para el reconocimiento de teclas presionadas, puede ser: capacitivo, mediante la variación del espesor de un elemento dieléctrico; de membrana, empleando elementos conductores en un membrana plástica; de mica, empleando elementos conductores colocados sobre la mica; en base a contactos, etc. De estos principios el más empleado es el de contactos.

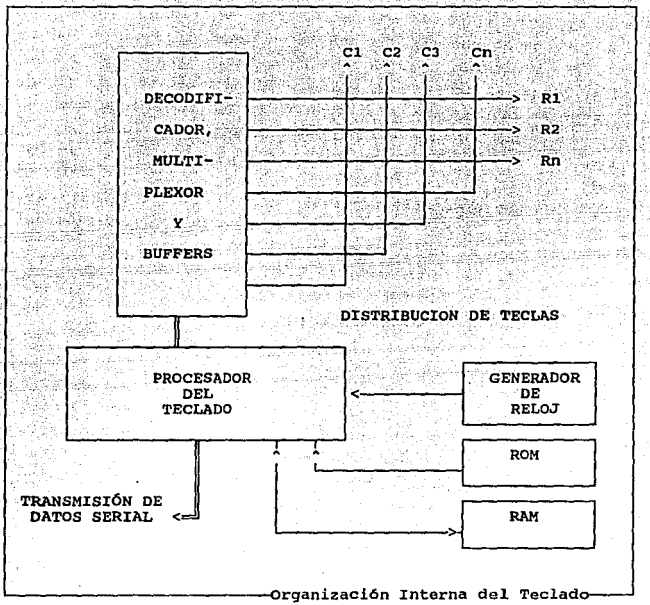
Elementos que Constituyen un Teclado

Un teclado esta constituido generalmente por un microprocesador de uso específico, el cual contiene la mayoría de las veces dentro del mismo circuito, un puerto serie, una memoria RAM, una memoria ROM, y un puerto paralelo. Todo esto con el único objetivo de estar explorando y verificando el estado de las teclas, obtener el código de la tecla y poder transmitir dicho código.

Operación del Teclado

Una forma simple de describir al teclado, es verlo como una unidad desmontable de 84 teclas, el cual transmite datos a una velocidad de 1200 bauds.

Una aproximación de la organización interna de un teclado se muestra en la siguiente gráfica.



Organización Interna del Teclado

Dentro de sus características se encuentran:

Interfase

El teclado usa un interfaz de comunicación serial bidireccional compuesta de 5 líneas de conexión, está es empleada para enviar señales entre el usuario (por teclado) y el sistema.

Secuencia de "Código de Exploración de Tecla"

El teclado es capaz de detectar cualquier tecla que es presionada, y al momento su código de exploración podrá ser enviado por la interfaz en secuencia correcta, independientemente del número de teclas que se mantengan presionadas. Algunos códigos de

tecla mientras la interfaz está deshabilitada, (cuando la Key Lock está encendida) pueden perderse. Los códigos de teclas son almacenados sólo cuando el teclado no está siendo atendido por el sistema.

Buffer del Teclado

El teclado tiene un buffer de 16 caracteres first-in-first-out (FIFO), donde se almacenan los códigos hasta que la interfaz este lista para recibirlos.

Una condición de sobreflujo del buffer ocurrirá si más de 16 códigos son almacenados en el buffer antes que el primer dato de tecla sea enviado. El decimoséptimo código es reemplazado con el código de sobreflujo 00_H (la posición decimoséptima es reservada para el código de sobreflujo). Si más teclas son presionadas antes que el sistema reconozca una salida del teclado, el dato se perderá. Cuando el teclado es reconocido para enviar datos, el carácter del buffer es enviado en operación normal, y el nuevo dato de entrada puede detectarse y enviarse.

Teclas

Todas las teclas son clasificadas como "make/break", lo que significa que cuando una tecla es presionada, el teclado envía un código make para cada tecla. Cuando la tecla es soltada, el código break es enviado (el código break para cada una de las teclas es el código make precedido por $F0_H$).

Todas las teclas son de escritura automática (typematic). Cuando una tecla es presionada y mantenida así, el teclado envía continuamente el código make para esta tecla hasta ser soltada.

La velocidad de transmisión a la cual es enviado el código make es conocido como velocidad de transmisión de escritura automática (typematic rate), esto se explicará más adelante. Cuando dos o más teclas son mantenidas presionadas, sólo la última tecla presionada repite el typematic rate. La operación typematic es detenida cuando la última tecla presionada es soltada, incluso cuando otra tecla es mantenida abajo. Cuando una tecla es presionada y mantenida así cuando la comunicación está deshabilitada, sólo el primer código make es almacenado en el buffer, esto previene el sobreflujo del buffer como resultado de la acción typematic.

Ejecución de Funciones y Tiempo de Encendido

La secuencia de operación que sigue el teclado desde el momento en que se alimenta y una vez iniciada su operación normal, en caso de un correcto funcionamiento se muestra en los siguientes diagramas, éste funcionamiento puede ser alterado según lo que el teclado reciba como comando y el estado de la comunicación con el sistema y sobre todo el estado funcional del teclado.

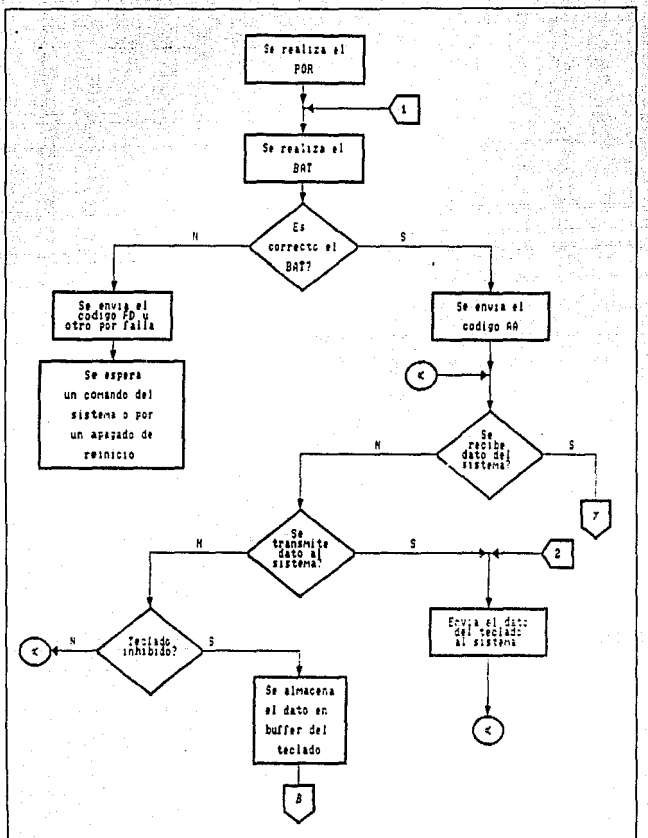


Diagrama de Operación del Teclado

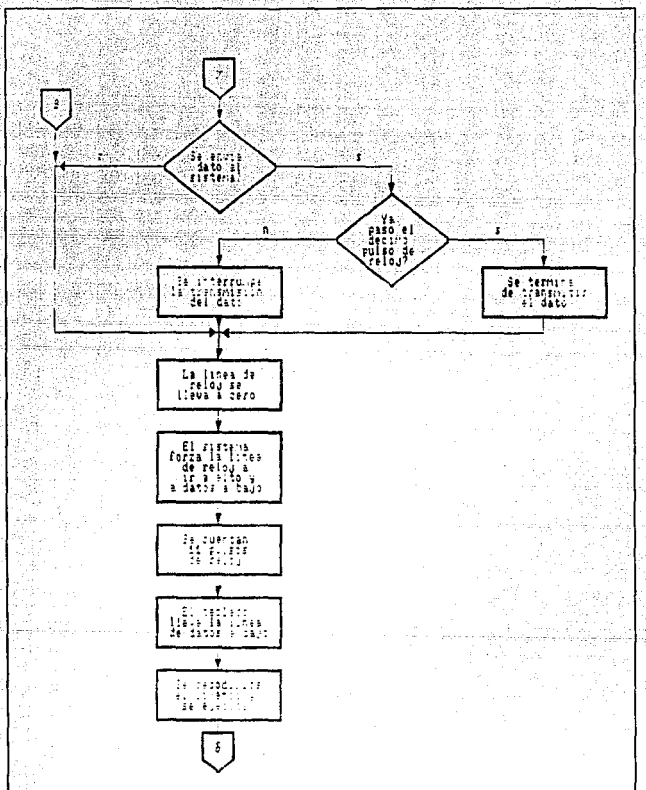


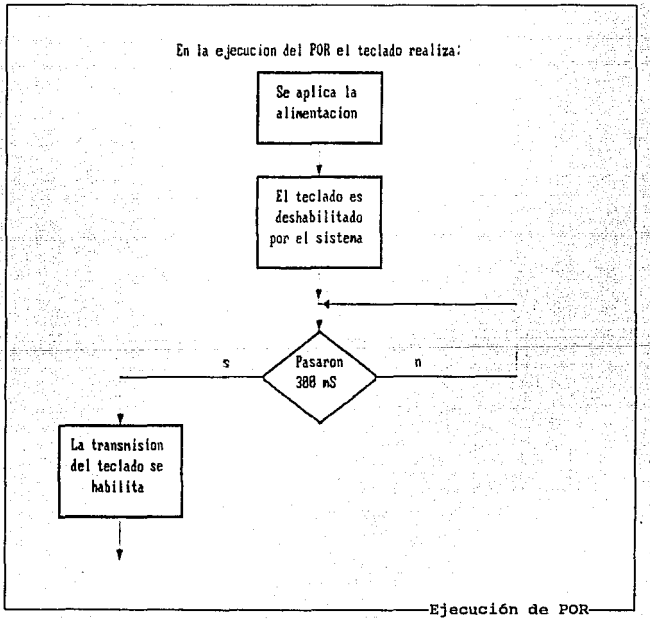
Diagrama de Operación del Teclado

Reinicio de Encendido

La lógica del teclado genera un POR (Power-On-Reset) cuando se le aplica la alimentación. El POR dura un mínimo de 300 ms y un máximo de 9 s.

Nota: El teclado puede emitir una entrada falsa durante los primeros 200 ms después de que +5 V_{DC} es estabilizado al 90% del nivel. Debido a esto, la interfaz del teclado debe ser deshabilitada en este período.

El POR se muestra en el siguiente diagrama:

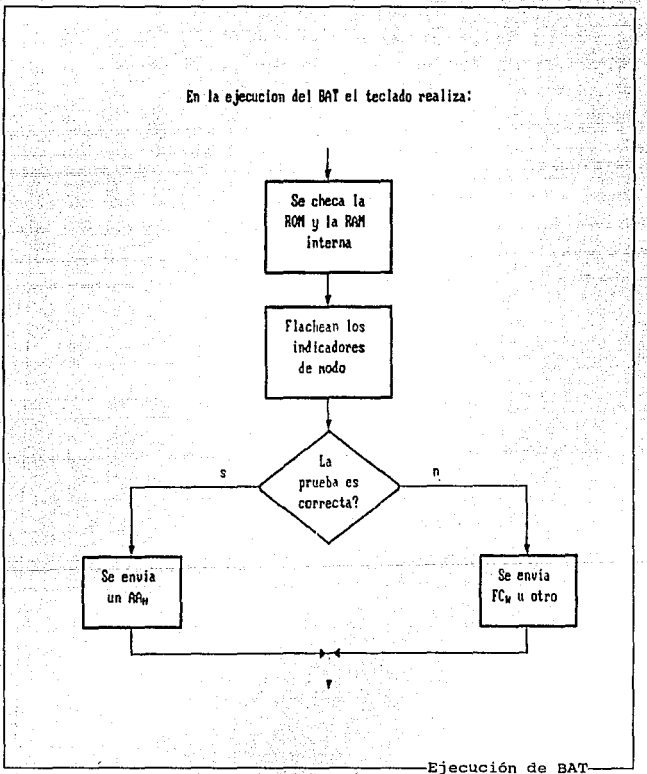


Ejecución de POR

Prueba de Seguridad Básica

Inmediatamente seguido de POR, el teclado ejecuta una prueba de seguridad básica BAT (Basic Assurance Test).

El BAT se muestra a continuación:



Ejecución de BAT

Esta prueba consiste de un chequeo de toda la ROM, un bit -estaca (stuck-bit) y una prueba de direccionamiento de toda la RAM del MPU del teclado. Los indicadores de modo (tres diodos emisores de luz - LED's- en la parte superior derecha del teclado) son encendidos y apagados como resultado de su operación.

La ejecución del BAT puede tomar de 600 a 900 mS, esto es en adición al tiempo requerido por POR. El BAT también puede iniciarse por un comando de RESET.

Después de BAT y cuando la interfase de comunicación esta habilitada (líneas de reloj y datos son puestas en nivel alto), el teclado envía un código completo por la interfaz, cualquiera de los dos: AA_H para la realización satisfactoria ó FC_H (ó cualquier otro código) para una falla. Si el sistema emite un comando de RESET, el teclado envía el código de realizado el BAT nuevamente, es decir, la prueba se vuelve a ejecutar. Por otra parte, el teclado pone las teclas en typematic y make/break.

Comandos Provenientes del Sistema

Los comandos descritos a continuación pueden ser enviados al teclado en cualquier instante y el teclado responderá dentro de un lapso de 20 mS.

Nota: Los comandos enviados por el sistema tienen diferente significado cuando son enviados por el teclado.

RESET (FF_H)

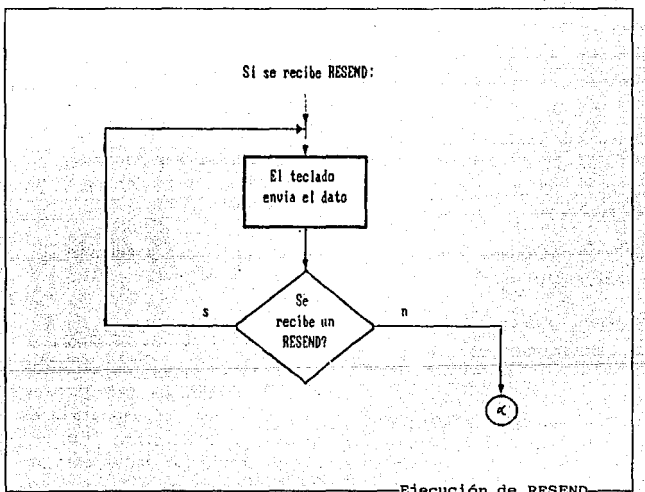
El sistema envía el comando de RESET para iniciar una programa de reinicio y una autoprueba interna del teclado. El teclado reconoce el comando con una señal de reconocimiento "acknowledge" (ACK) y como resultado el sistema acepta el ACK antes de ser ejecutado el comando. El sistema acepta la señal de ACK junto con el levantamiento de las líneas de reloj y datos por un mínimo de 500μS. El teclado es deshabilitado desde la recepción del comando de RESET hasta que el ACK es aceptado o hasta que otro comando anule al anterior.

Enseguida de aceptar el ACK, el teclado comienza la operación de reinicio, que es similar al reinicio de encendido. El teclado

limpia las salidas del buffer y pone los valores por default para typematic y para la velocidad de retraso.

RESEND (FE_H)

El sistema puede enviar este comando cuando es detectado un error en cualquier transmisión del teclado. Sólo puede enviarse después de una transmisión y antes de que el sistema habilite la interfase y permita la siguiente salida del teclado. En la recepción de RESEND, el teclado envía la salida anterior nuevamente, a menos que la salida previa fuese RESEND. En este caso, el teclado reenvía el último byte antes del comando de RESEND.



No Operación (NOP) (de $F7_H$ a FD_H y de EF_H a $F2_H$)

Este comando es reservado y efectivamente es una no operación ó NOP. El sistema no usa este código. Si se envía, el teclado reconocerá el comando y continuará en el anterior estado de exploración, es decir, ninguna otra operación ocurrirá.

Poner por Default (SET DEFAULT) ($F6_H$)

El comando de poner por default reinicia todas las condiciones de estado por default del encendido. El teclado responde con ACK, limpia el buffer de salida, pone las condiciones por default y continua con la exploración (esto si el teclado fue previamente habilitado).

Poner por default y deshabilita ($F5_H$)

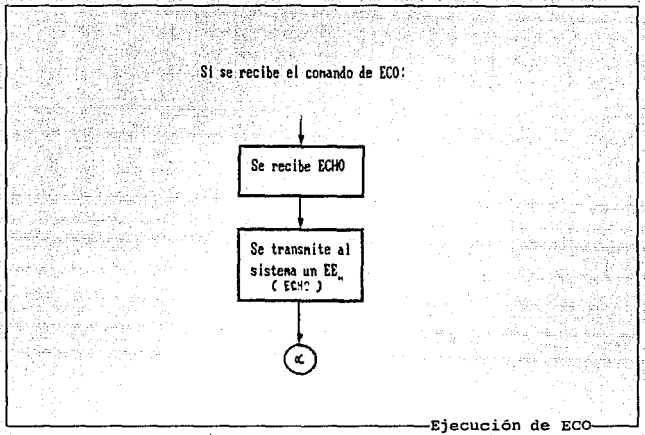
Este comando es similar a poner por default, excepto que el teclado detiene el barrido (exploración) y espera la siguiente instrucción.

Habilitación ($F4_H$)

En la recepción de este comando, el teclado responde con un ACK, limpia el buffer de salida y comienza la exploración.

Eco (ECHO) (EE_H)

Eco es un diagnostico de ayuda. Cuando el teclado recibe este comando, es emitido un EE_H como respuesta y continua explorando si el teclado fue previamente habilitado.



Ejecución de ECO

Puesta/Reinicio de Indicadores de Modo (ED_H)

El teclado ofrece tres indicadores de modo que son accesibles al sistema. El teclado activa o desactiva cualquiera de estos modos cuando recibe un comando válido del sistema. Estos pueden activar o desactivar cualquier combinación. El sistema recuerda el estado previo de un indicador de modo, de tal forma que si es puesto, no cambiará cuando una secuencia de comando es emitida para modificar el estado de otro indicador.

El comando tiene el siguiente formato:

COMANDO	OPCIÓN

Un comando puesta/reinicio de indicador de modo consiste de 2 bytes. El primero es el byte de comando y tiene el siguiente aspecto:

11101101_B ED_H

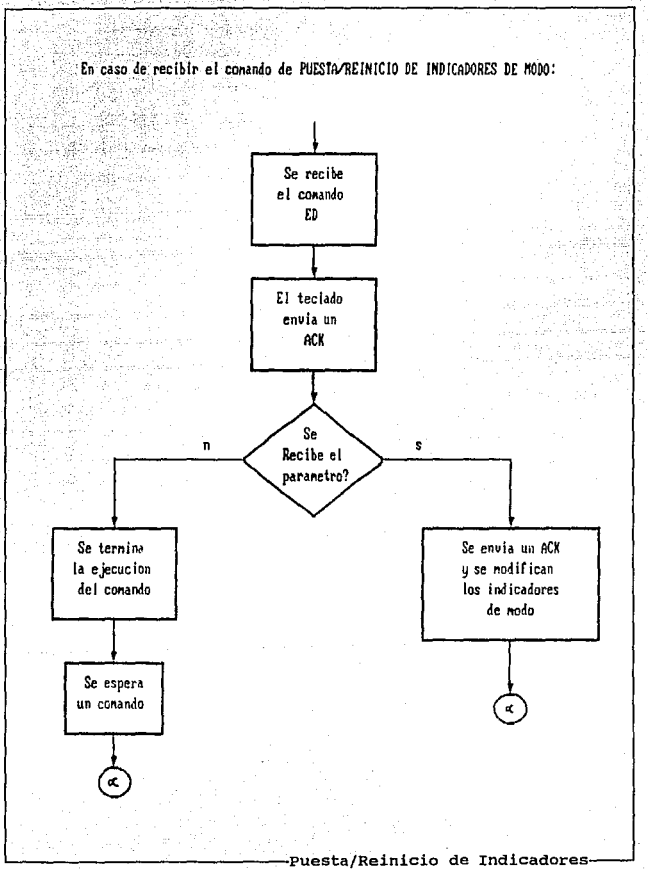
El segundo byte es de opciones. Este es una lista de indicadores que son actuados por él. El formato del byte de opción es:

Bit 7	Reservado	Bit 3	Reservado
Bit 6	Reservado	Bit 2	Indicador de Caps Lock
Bit 5	Reservado	Bit 1	Indicador de Numeric Lock
Bit 4	Reservado	Bit 0	Indicador de Scroll Lock

Opciones de Indicadores de Modo

Nota: El bit 7 es el bit más significativo y el bit 0 el menos significativo.

El Teclado responderá al comando de puesta/reinicio de indicador de modo con un ACK, detendrá la exploración, y aguardará por el byte de opción, nuevamente responderá al byte de opción con un ACK, pondrá los indicadores de modo y continuará la exploración si el teclado fue previamente habilitado. Si otro comando es recibido en lugar del byte de opción, la ejecución de la función de puesta/reinicio de indicadores de modo es suspendida sin cambios en los estados de los indicadores, y el nuevo comando es procesado, y la exploración es restaurada.


En caso de recibir el comando de PUESTA/REINICIO DE INDICADORES DE MODO:
Se recibe el comando ED
El teclado envia un ACK
Se Recibe el parametro?
n
s
Se termina la ejecucion del comando
Se espera un comando
Se envia un ACK y se modifican los indicadores de modo
Puesta/Reinicio de Indicadores

Asignando la Velocidad/Retraso de Typematic ($F3_H$)

El sistema emite este comando seguido por un parámetro, para cambiar la velocidad/retraso de Typematic. La velocidad y retraso de typematic son determinados por el valor del byte que sigue al comando. Los bits 6 y 5 sirven para el parámetro de retraso y los bits 4, 3, 2, 1 y 0 (el bit menos significativo) son los parámetros para la velocidad. El bit 7, el más significativo, es siempre cero.

La siguiente tabla muestra las posibles velocidades:

Bit Rate		Bit Rate	
00000	30.0	10000	7.5
00001	26.7	10001	6.7
00010	24.0	10010	6.0
00011	21.8	10011	5.5
00100	20.0	10100	5.0
00101	18.5	10101	4.6
00110	17.1	10110	4.3
00111	16.0	10111	4.0
01000	15.0	11000	3.7
01001	13.3	11001	3.3
01010	12.0	11010	3.0
01011	10.9	11011	2.7
01100	10.0	11100	2.5
01101	9.2	11101	2.3
01110	8.6	11110	2.1
01111	8.0	11111	2.0

Velocidades de Typematic

El retraso es igual en complemento a 1 en valor binario de los bits 6 y 5 multiplicados por 250 mS ± 20%. El período (intervalo de una typematic a la siguiente) es determinado por la siguiente ecuación:

$$Período=(8+A)2^{B}*0.00417 \text{ seg}$$

Donde: A = valor binario de los bits 2, 1 y 0
B = valor binario de los bits 4 y 3

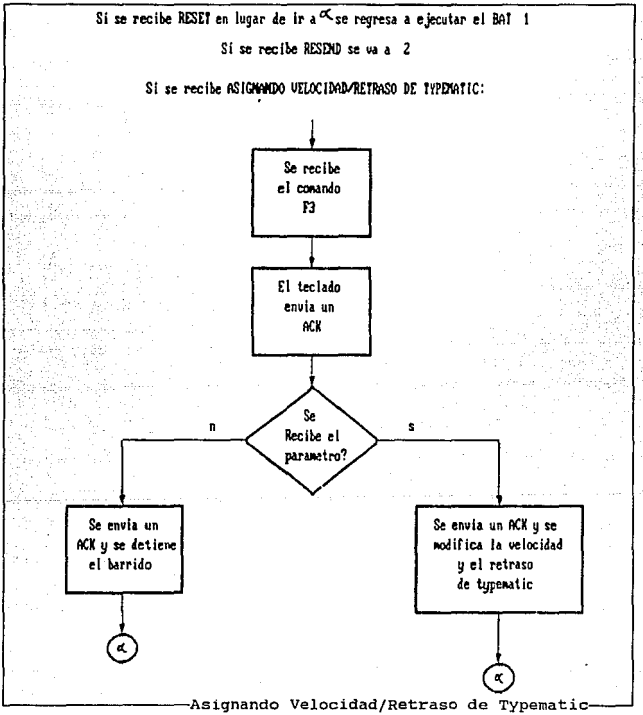
La velocidad de typematic (códigos make por segundo) es 1/período. El teclado responde a este comando con ACK, detiene la exploración, y espera el parámetro de velocidad. El teclado responde al parámetro de velocidad con otro ACK, cambia la velocidad y el retraso, y continua la exploración (si el teclado fue previamente habilitado).

Si un comando es recibido en lugar del parámetro de velocidad, la función de asignar la velocidad de typematic termina sin cambios al

rango existente, y el nuevo comando es procesado. Sin embargo, el teclado no continua explorando, a menos que la instrucción haya sido un comando de habilitación.

La velocidad por default del teclado es la siguiente:

Velocidad de typematic = 10 caracteres por segundo y
Retraso = 500 mS ± 20%



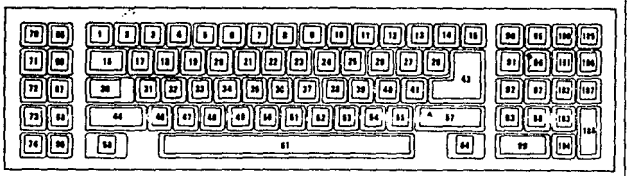
Asignando Velocidad/Retraso de Typematic

Salidas del Teclado

Códigos de Exploración de Tecla

Cada tecla tiene asignado un único código de exploración (make) de 8 bits, que es enviado al sistema cuando la tecla es presionada. Cada tecla envía el código break cuando es soltada; el código break consiste de 2 bytes , tanto el primero como el segundo son idénticos e iguales al código exploración make para dicha tecla.

El código de exploración Typematic es el mismo que el código make. La siguiente tabla muestra la disposición de las teclas en un teclado y el código de exploración meke correspondiente.



Disposición de un Teclado

Key Positions and Their Make Codes

1--DE	18--1D	36--33	55--4A	90--76
2--16	19--24	37--3B	56--51	91--6C
3--1E	20--2D	38--42	57--59	92--6B
4--26	21--2C	39--4B	58--11	93--69
5--25	22--35	40--4C	60--19	94--77
6--2E	23--3C	41--52	61--29	98--75
7--36,	24--43	43--5A	64--58	97--73
8--3D	25--44	44--12	65--D8	98--72
9--3E	26--4D	46--1A	66--DC	99--70
10--46	27--54	47--22	67--0B	100--7E
11--45	28--5B	48--21	68--0A	101--7D
12--4E	30--14	49--2A	69--09	102--74
13--55	31--1C	50--32	70--05	103-7A
14--5D	32--1B	51--31	71--04	104--71
15--66	33--23	52--3A	72--D3	105--84
16--0D	34--2B	53--41	73--83	106--7C
17--15	35--34	54--49	74--01	107--7B

Códigos Make

Códigos de Comandos al Sistema

Los códigos de estos comandos son enviados por el teclado y tienen diferente significado cuando son emitidos por el sistema.

Resend (FE_H)

El teclado emite un comando de reenvio enseguida de recibir una entrada inválida o cualquier entrada con paridad incorrecta. Si el sistema no envía nada al teclado la respuesta no es requerida.

ACK (FA_H)

El teclado envía un ACK como respuesta a cualquier entrada válida, excepto para los comandos de ECHO y RESEND. Si el teclado es interrumpido cuando envía un ACK, éste descartará el ACK y aceptará y responderá al nuevo comando.

Sobreflujo (00_H)

Un carácter de sobre flujo es puesto en la posición 17 del buffer del teclado, por encima del último código si el buffer llega a estar lleno. El código es enviado al sistema como un sobreflujo cuando alcanza la parte alta del buffer.

Falla de Diagnóstico (FD_H)

La prueba periódica del teclado sensa el sistema del teclado y envía un código de falla de diagnostico si éste detecta cualquier problema. Si una falla ocurre durante el BAT, el teclado detiene la exploración y aguarda por un comando del sistema o por un apagado para reiniciar. Si la falla es reportada después de habilitar la exploración, ésta es continuada.

Prefijo de Código de Ruptura o Break ($F0_H$)

Este código es enviado como el primer byte de una secuencia de 2 bytes, el cual indica la liberación (que fue soltada) de una tecla.

Código de Completado el BAT (AA_H)

Enseguida de una terminación satisfactoria del BAT, el teclado envía AA_H. FC_H (ó cualquier otro código) realiza el microprocesador del teclado cuando se dio una falla.

Respuesta Eco (EE_H)

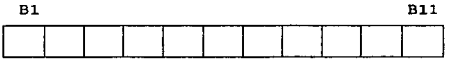
Este código es enviado en respuesta a un comando eco (ECHO) del sistema.

Señales de Reloj y de Datos

El teclado y el sistema se comunican a través de las líneas de reloj y datos. La fuente de cada una de éstas es un dispositivo open-colector en el teclado, los cuales permiten que tanto el sistema como el teclado forcen una línea a nivel negativo. Cuando no hay comunicación, tanto la línea de reloj y datos están en nivel positivo.

Una transmisión de datos del y hacia el teclado consiste de un flujo de datos de 11 bits, los cuales son enviados en serie por la línea de datos. La siguiente figura muestra la estructura de los datos.

BIT	FUNCIÓN	BIT	FUNCIÓN
1	0 bit de inicio	7	Bit de dato 5
2	Bit de dato 0 (LSB)	8	Bit de dato 6
3	Bit de dato 1	9	Bit de dato 7 (MSB)
4	Bit de dato 2	10	Bit cero ó
5	Bit de dato 3		Bit de paridad (NON)
6	Bit de dato 4	11	Bit de parada



Formato de los Datos

El bit de paridad es 1 ó 0, pero los bits de datos altos de las dos palabras (código make y break) más el bit de paridad siempre deben ser un número non.

Cuando el sistema envía datos al teclado, la línea de datos es forzada a nivel negativo y permite a la línea de reloj ir a nivel positivo. Cuando el teclado envía datos al sistema o los recibe de éste, es generada la señal de reloj al tiempo de dato. El sistema

puede prevenir al teclado para un envío de datos por un force de la línea de reloj a nivel negativo, la línea de datos puede ir a nivel positivo ó negativo durante este tiempo.

Durante el BAT, el teclado permite a la línea de reloj y datos ir a nivel positivo.

Salida de Datos del Teclado

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

Si tanto el reloj y datos son altos, el teclado envía el bit 0 de inicio, los 8 bit de datos, cero si se trata del byte del código make o el bit de paridad non (1 o 0) si se trata del código break y el bit de parada. Para el bit de paridad se consideran los dos códigos, es decir, se toma en cuenta el número de unos del byte del código make y del byte del código break. Los datos son válidos antes del corte o flanco de caída y después del flanco de subida de cada pulso de reloj. Durante la transmisión, el teclado checa la línea de reloj para un posible nivel positivo al final de cada 60 mS. Si el sistema baja la línea de reloj de un nivel positivo después de que el teclado inicio el envío de un dato, una condición conocida como contención de línea (line contention) ocurre, y el teclado detiene el envío de datos. Si la contención de línea ocurre antes del flanco de subida del décimo ciclo de reloj (bit de paridad), el buffer del teclado retorna las líneas de reloj y datos a nivel positivo. Si la contención no ocurre por el décimo ciclo de reloj, el teclado termina la transmisión.

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

Entrada de Datos al Teclado

Cuando el sistema esta listo para enviar datos al teclado, primero verifica si éste se encuentra transmitiendo datos. Si el teclado esta enviando datos pero no ha llegado al décimo ciclo de

reloj, el sistema puede anular la salida del teclado por medio de forzar la línea de reloj a nivel negativo. Si la transmisión del teclado paso del décimo ciclo de reloj, el sistema debe recibir la transmisión. Si el teclado no esta enviando datos o si el sistema decide anular la salida del teclado, lleva a la línea de reloj a nivel negativo por más de 60 mS mientras prepara la transmisión. Cuando el sistema esta listo para enviar el bit de inicio (la línea de datos será baja), permitiendo a la línea de reloj ir a nivel positivo. El teclado checa la línea de reloj en intervalos de no menos de 60 mS. Si la petición de envío es detectada, el teclado cuenta 11 bits. Después del décimo bit, el teclado forza la línea de datos a bajo y cuenta un bit más (el bit de parada). Esta acción señala al sistema que el teclado ha recibido el dato.

En la recepción de esta señal, el sistema regresa a un estado de listo, en el cual puede aceptar la salida del teclado o puede inhibirlo hasta estar listo.

Cada comando del sistema o transmisión de datos al teclado requiere de una respuesta del mismo, para que el sistema pueda enviar la siguiente salida. El teclado responderá dentro de un lapso de 20 mS a menos que el sistema prevenga la salida del teclado. Si la respuesta del teclado es inválida o con error de paridad, el sistema envía el comando ó dato nuevamente. Un comando de RESEND puede no ser enviado en este caso.

Especificaciones

Las especificaciones del teclado se refieren a sus dimensiones y proporciones, éstas son:

Longitudes:

Largo	540 mm
Ancho	100 mm
Alto	225 mm

Peso:

2.8 Kg

Conector del teclado

El cable del teclado que lo conecta al sistema es a través de un conector DIN de 5 terminales. La siguiente figura enlista las terminales de conexión y sus respectiva señal.

PIN DEL CONECTOR	NOMBRE DE LA SEÑAL
1	CLOCK
2	DATOS
3	NO SE USA
5	GROUND
4	+5 V_{DC}

Secuencia de Operación

Un ejemplo de la operación del teclado seria la siguiente: Cuando el sistema se enciende, el teclado recibe la alimentación por medio del conector DIN provocando la ejecución del POR. Al finalizar el POR, se inicia la ejecución del BAT; dependiendo del resultado de BAT el teclado envía al sistema el código de finalizado y como se llevo a cabo (bien ó mal) si el sistema recibió un código de error, puede hacerlo notificar al usuario para que se hagan las correcciones necesarias, en caso contrario, se puede estar esperando ya sea por el envío de datos (si una tecla fue presionada) ó por una recepción de comandos del sistema.

En el caso de envío de datos al sistema, al ser presionada una tecla, el procesador del teclado, lo detecta y genera su código make, para enviar este código verifica si la línea de datos es positiva (alta), en caso afirmativo, se envía el primer bit (0 de inicio), los bits de datos, un bit cero y el de alto posteriormente el código break (el bit de inicio 0, los 8 bits de datos el bit de paridad tomando ambos códigos y el bit de parada). Si la tecla

continua presionada, se continua enviando su código typematic hasta que es soltada. En caso de que la línea de datos sea negativa (baja), el procesador almacena el código make en el buffer una sola vez, a pesar de que la tecla presionada, posteriormente revisa la línea de reloj, si ésta es baja, espera a que el sistema lo habilite, en caso de ser alta, el teclado se alista para recibir un comando del sistema.

En el caso de envíos del sistema al teclado, el sistema verifica si el teclado no esta enviando datos, en caso de hacerlo ve en que bit de transmisión va, en caso de no haber llegado al décimo, puede interrumpir la transmisión o esperar a que termine; si ya paso el décimo bit, el sistema de esperar a que termine. Cuando el sistema quiere transmitir al teclado, este pone la línea de datos en bajo y la de reloj en alto, el sistema entonces empieza a transmitir, el teclado debe ahora contar 11 ciclos de reloj, una vez contados éstos, pone la línea de datos en bajo, como respuesta al mensaje.

Esta secuencia se repite continuamente.

CAPITULO

III

DISEÑO Y CONSTRUCCION

III

Diseño y Construcción

III a.- Desarrollo y Pruebas del Sistema Monitor.

Introducción.

Para que el sistema funcionara adecuadamente se realizaron dos programas, el primero de ellos, llamado *Programa de evaluación*, tiene como finalidad el ayudar al usuario a crear y probar sus programas en lenguaje ensamblador o en código de máquina del 6809, con la disposición de todos los recursos del sistema; el segundo programa es el llamado *programa principal*. Dentro de las funciones que realiza están: las mediciones de voltaje, corriente y demás variables necesarias para realizar su función de control, inicialización del sistema, etc. Ambos programas desarrollan su

función de distinta manera, mientras el programa de evaluación necesita que se le den los comandos adecuados semejante en terminales al modo línea de operación, el programa de control es manejado por medio de opciones (menús), las cuales son desplegadas en forma consecutiva, con base a la opción anteriormente solicitada.

Ambos programas son independientes uno del otro y estos se encuentran grabados en dos memorias EPROM por separado. Cuando se desee trabajar el sistema únicamente como módulo de evaluación o como módulo de control, solo se necesitará colocar la memoria EPROM adecuada en la tarjeta de central de proceso ya sea la que contenga el programa de evaluación o el programa de control respectivamente. Cuando se quiera utilizar el sistema con ambas opciones las dos memorias deben incluirse en el sistema, la memoria del programa principal deberá de estar en la tarjeta central, mientras que el programa de evaluación deberá colocarse en la tarjeta de adquisición de datos y control. Los programas se almacenaron en memorias separadas para poder ejecutar una u otra independientemente y el espacio que ocupan cada uno de los programas.

El funcionamiento de estos programas se detalla a continuación.

Programa de Evaluación.

El programa de evaluación es empleado para depurar programas de aplicación para el 6809, se puede verificar la operación de un programa ya sea en el modo de ejecución paso a paso y/o por medio de la ubicación de puntos de ruptura. El programa cuenta con varias opciones como el ensamblador/desensamblador, apuntador a memoria, colocación de puntos de ruptura, etc.

Gracias al desensamblador incluido en el programa de evaluación, es posible que el programa a verificar pueda ser examinado en nemónicos. Este mismo puede ser modificado en su funcionamiento si se emplea el ensamblador, el cual acepta nemónicos de instrucciones, y el cual puede calcular saltos y evaluar expresiones. Este programa, como se mencionó anteriormente funciona con base a comandos.

Sumario de Comandos.

La siguiente tabla sintetiza los comandos disponibles dentro del sistema de evaluación, así como su sintaxis.

C <reg>	Modifica los registros del MPU.
D [<exp>,<exp>]	Despliegue del contenido de un bloque de memoria.
F	Despliegue de registros/llamado de funciones.
G [<exp>,...]	Ejecución del programa del usuario con puntos de ruptura.
I [<exp>]	Despliegue/ensamble de instrucciones.
M [<exp>]	Despliegue del contenido de memoria.
P [<exp>,<exp>]	Despliegue del programa objeto.
Q	Salir del programa de evaluación.
R	Despliegue del contenido de registros.
S	Ejecución del programa del usuario paso a paso.

Donde:

<reg> es cualquier registro del 6809 y

<exp> es una expresión para definir un valor.

Comandos.

Cuando el apuntador (*prompt*) "F>" es desplegado en el televisor, el programa de evaluación esta listo para recibir un comando proveniente del teclado. Estos comandos consisten de una letra, normalmente seguida de un parámetro opcional. Mientras los parámetros son insertados, el usuario tiene las siguientes opciones: puede corregir errores presionando la tecla de retroceso (*back space*), cancelar el comando presionando cancelación (*esc*) o ejecutar el comando presionando regreso de carro (*return o enter*).

El parámetro del comando consiste de una o más expresiones usadas normalmente para especificar direcciones, márgenes u operandos. La expresión más simple y más común es la de una constante en hexadecimal.

```
FFFE o 8D0A o 6C o F
```

Se pueden introducir expresiones más complejas, como suma restas, etc, de estas se discutirá más adelante.

Algunos comandos requieren que se presione la tecla de enter para iniciar su ejecución, en otros, la ejecución es inmediata.

Descripción de los comandos.

C Cambio en el contenido de registro.

Cuando el usuario desee modificar operandos, datos, direcciones, etc puede hacer uso de este comando. Cuando el apuntador F> es desplegado, se puede modificar el contenido de cualquier registro interno del 6809; para ello, se inserta el carácter correspondiente al registro deseado enseguida del comando C. Los caracteres correspondientes a cada registro son:

C	Registro de condición de código.
A	Registro acumulador A.
B	Registro acumulador B.
D	Registro acumulador D (A y B).
Z	Registro apuntador de página directa.
X	Registro indexado X.
Y	Registro indexado Y.
P	Registro contador de programa.
U	Registro apuntador de usuario.
S	Registro apuntador de sistema.

Se despliega el contenido actual en hexadecimal del registro y se puede introducir una expresión para especificar el valor a reemplazar. Por ejemplo, para cambiar el contenido del registro acumulador D por $2100:

```
F> C D=FFFF 2100
```

Cabe hacer notar que cuando se introduce un valor inválido, ya sea directamente o por medio de una expresión, el registró no es alterado en su contenido.

D Despliegue de un bloque de memoria.

El comando D permite desplegar el contenido de un bloque de memoria en hexadecimal y el código ASCII correspondiente a dicho valor, esto es siempre y cuando el código ASCII pueda ser desplegado por el VDG, en caso contrario, se sustituirá por "." Esto es empleado para visualizar y verificar el contenido de arreglos o caracteres almacenados, los cuales serán empleados por el programa o bien como resultados de operaciones efectuadas por el programa.

El comando D acepta un par de expresiones, las cuales especifican las direcciones de inicio y final del bloque a desplegar. Si el comando es invocado sin parámetros, el rango empleado en el último comando P o D es usado por omisión.

```
F> D 8000,8007
8000  61 62 63 64  ABCD
8004  00 00 00 00  ....
F>
```

F Llamado a función.

Algunas veces se desea que dentro del seguimiento de un programa paso a paso, la ejecución de una subrutina (llamado a función) se realice como si se tratara de una instrucción simple, para ello, el comando F permite la ejecución de un BSR, LBSR o JSR y la subrutina asociada como una sola instrucción. El salto a la subrutina a ser ejecutada debe ser especificado por el contenido del registro P, si P no apunta a un BSR o JSR, el comando F se comporta como si se tratara del comando S. Por ejemplo si el programa a seguir y el estado del MPU son:

```
S=77FF FHI...C A=21 B=00 Z=00
X=FFFF Y=FFFF U=FFFF
P=1004 BSR 217C
```

Obsérvese cual es el contenido de los registros acumuladores, de los registros apuntadores y de los registros de índice, así como la instrucción a la cual se esta apuntando por medio de P. Cuando es ejecutado el comando F:

```
F> F
S=77FF F.IN.VC A=12 B=24 Z=00
X=C324 Y=C23F U=77FD
P=1006 LDA #03
```

Se observa que en el desplegado siguiente varios registros han cambiado su contenido y el contador de programa P direcciona a la instrucción siguiente, reflejando así la ejecución de la subrutina como si fuese una instrucción común.

G Ejecución del programa del usuario.

El comando G cede el control total del sistema al programa del usuario. El programa de evaluación recupera y restaura el contenido de los registros dentro del programa del usuario y comienza el seguimiento de éste. Opcionalmente, se permite el insertar puntos de ruptura para regresar el control al programa de evaluación. Estos puntos de ruptura deben ser asignados a las direcciones de inicio de instrucciones, en las cuales se desea detener la ejecución, en caso de no encontrar dicha dirección, el programa del usuario tendrá control completo de los recursos del sistema. Por ejemplo, para iniciar la ejecución del programa con BSR 217C:

```
F> G
```

Alternativamente, si se desea detener la ejecución del programa si se encuentra una instrucción en las direcciones $2185 o $2119:

```
F> G 2185, 219C
```

Si el punto de ruptura es encontrado, el programa de usuario es suspendido, el programa de evaluación toma el control y el contenido final de los registros es desplegado y almacenando para una próxima recuperación. Cabe aclarar que los puntos de ruptura solo se pueden introducir al inicio de la dirección correspondiente a la instrucción y solo dentro de las direcciones de memoria RAM.

I Despliegue/cambio de instrucciones.

El comando I despliega o cambia el contenido del programa de usuario, esto gracias al desensamblador/ensamblador. El usuario especifica la dirección de inicio y el programa de evaluación despliega las instrucciones sucesivas en nemónicos. En cada despliegue el usuario puede seleccionar una de tres opciones:

1. Introducir una nueva instrucción.
2. Dejar la instrucción sin cambio.
3. Terminar la ejecución del comando.

Introduciendo una instrucción en nemónicos y presionando enter se reemplaza la instrucción anterior, presionando solo enter la

instrucción no se modifica y para finalizar la ejecución del comando se presiona la tecla de ESC. En el ejemplo siguiente el comando I es empleado para crear una pequeña subrutina en la localidad $2000:

```
F> I 2000
2000  NEG <00  LDX #FFFF
2003  NEG <00  LEAX -1,X
2005  NEG <00  BNE 2003
2007  NEG <00  RTO
? 2007  NEG <00  RTS
2008  NEG <00
F>
```

En el ejemplo, un valor simple en hexadecimal es empleado para especificar el destino de un salto o el margen de un índice, pero pueden emplearse expresiones más complejas, de estas se hablara más adelante. El programa de evaluación detecta un error durante la adquisición de la instrucción (2007), en este caso, la instrucción anterior es desplegada nuevamente, indicando que hay un error a la entrada (RTO en lugar de RTS).

La mayoría de los nemónicos del 6809 son desplegados por el programa de evaluación a excepción de ORCC y ANDCC, los cuales son desplegados como SEC y CLC respectivamente. Por ejemplo, si se desea realizar una operación ORCC #01 o ANDCC #FE se tiene en su lugar SECC ....C y CLCC ....C . El nemónico CWAI es aceptado pero el operando es desplegado como una lista de bits. Si no hay expresión a continuación del comando I, el programa de evaluación toma por omisión la dirección apuntada por el registro P.

M Despliegue/cambio en memoria.

El comando M despliega y cambia el contenido de áreas de memoria. El usuario especifica la dirección de inicio del bloque de memoria a ser examinado. El programa de evaluación despliega el contenido de los sucesivos bytes de memoria, brindándole al usuario una de tres opciones:

1. Introducir un valor nuevo.
2. Dejar el valor sin cambio.
3. Terminar la ejecución del comando.

Introduciendo una expresión y presionando enter se reemplaza el valor, presionando solo enter, se deja sin modificar el dato y para finalizar la ejecución se presiona la tecla de ESC.

En el ejemplo siguiente, el comando M limpia los 8 bytes de la dirección $8000 a la $8007

```
F> M 8000
8000     FA      0
8001     FF      0
8002     FA      0
8003     FF      0
8004     56      0
8005     34      0
8006     FF      0
8007     FF      0
8008     FE
```

Si no hay una expresión a continuación del comando M, el programa asume la dirección apuntada por el registro S.

A pesar de que el comando M puede también ser usado para modificar instrucciones del programa de usuario, el comando I, descrito anteriormente, es mucho más apropiado para tal propósito.

P Despliegue de programa en S1/S9.

El comando P permite desplegar el programa en el formato de código objeto MIKBUG™ S1/S9. Este formato esta formado en primer lugar por los caracteres "S1", seguido por el numero de bytes a desplegar, la dirección correspondiente y al final por la suma de bytes (checksum). Esta opción es con la finalidad de poder checar un programa con paquetes que trabajen con este formato.

```
F> P 8000,801F
S107800061626364FE
S10780040000000074
S10780100000000068
S10780140000000064
```

Si el rango de direcciones no es especificado, el rango del último comando D o P es empleado por omisión.

Q Salir del programa de evaluación.

El comando Q permite regresar al programa principal de control en caso de que exista. De esta forma se termina de ejecutar el programa de evaluación.

R Despliegue del contenido de los registros.

El comando R se empleará cuando el usuario desee revisar el contenido de los registros del MPU conforme a la ejecución de su programa. Por supuesto, los valores desplegados no corresponden a los valores reales de los registros internos del 6809, puesto que éste se encuentra ocupado ejecutando el programa de evaluación.

```
F> R
S=BF92 FHINZV. A=FF B=FF Z=00
X=FFFF Y=FFFF U=FFFF
P=1000 LDS #77FF
```

En el ejemplo se obtiene que el apuntador S contiene $BF92, los registros acumuladores A y B contienen $FF, etc. El registro de código de condición es desplegado como una lista de bits, en la cual cada bit activado (1) es mostrado por su nombre. Un bit desactivado (0) es mostrado por "." En el ejemplo todos los bits a excepción del bit "c" están activados. También la instrucción por ejecutar es mostrada en forma de nemónicos.

S Ejecución paso a paso del programa de usuario.

Este comando permite ejecutar un programa del usuario instrucción por instrucción, es decir, sólo una instrucción a la vez, y así poder examinar el resultado de ésta con detenimiento. El comando S ejecuta la instrucción especificada por el registro P, además del despliegue del contenido de los registros, esto es con la finalidad de poder verificar los cambios que sufren éstos. Por ejemplo si se presiona la tecla S la instrucción por ejecutar del ejemplo anterior es LDS #77FF:

```
F> S
S=77FF FHI...C A=21 B=00 Z=00
X=FFFF Y=FFFF U=FFFF
P=1004 BSR 217C
```

Aquí, S=77FF refleja el resultado directo de la ejecución de la instrucción, FHI...C y P=1004 son también efectos colaterales de la misma.

Evaluación de expresiones.

En los ejemplos anteriores, se utilizó una simple constante en hexadecimal para especificar dirección, margen, operando, u otros. Siempre que el programa de evaluación espere un valor numérico como dato adicional al comando, una expresión puede ser asignada. Una expresión es formada en base a operandos constantes combinada con operadores matemáticos. La operación es realizada usando aritmética de 16 bits no signada, e ignorando el sobreflujo que se pudiera generar como resultado de ésta. Dos tipos de constantes son reconocidas para dicho caso:

Hexadecimal- contiene los dígitos 0-9 o A-F.

Carácter - Se inicia con un apóstrofe seguido inmediatamente por un carácter ASCII de despliegue.

Cinco operaciones son reconocidas:

+	adición de 16 bits.
-	substracción de 16 bits.
*	multiplicación de 16 bits.
/	división de 16 bits (cociente).
%	división de 16 bits (residuo).

La evaluación de la expresión es estrictamente de izquierda a derecha sin operadores precedentes. Esto es: 3+2*5 evaluado es $19 (25 en decimal) y no $0d (13 en decimal). Además no se permiten espacios dentro de la expresión entre operandos y operadores.

```
LDB #'X      carga B con el código ASCII de X.
LDA #'H-40   carga A con el código ASCII de backspace.
LEAX 3*6,X   incrementa X con 66 en decimal
```

Vectores de interrupción.

El programa de evaluación permite una completa flexibilidad dentro del sistema y con los dispositivos externos, y una de las funciones que hay en la mayoría de los sistemas es la interacción con interrupciones. Para tal propósito, dos bytes son reservados para cada uno de los vectores de atención a interrupción en la parte alta de la memoria RAM; justo arriba del área de las variables del programa de evaluación. Estas direcciones son:

Interrupción	Dirección
SWI3	5FF2
SWI2	5FF4
FIRQ	5FF6
IRQ	5FF8
SWI	5FFA
NMI	5FFC
RESET	5FFE

Instrucciones del 6809.

Nemónicos.

Los siguientes nemónicos son desplegados y aceptados por el ensamblador/desensamblador del programa de evaluación.

abx	asrb	bne	cmpd	eora	lbhi	ldb	mul	rolb	sts
adca	bcc	bpl	cmps	eorb	lble	ldd	neg	ror	stu
adcb	bcs	bra	cmpu	exg	lbls	lds	nega	rora	stx
adda	beq	brn	cmpx	inc	lblt	ldu	negb	rorb	sty
addb	bge	bsr	cmpy	inca	lbmi	ldx	nop	rti	suba
addd	bgt	bvc	com	incb	lbne	ldy	ora	rts	subb
anda	bhi	bvs	coma	jmp	lbpl	leas	orb	sbca	subd
andb	bita	clcc	comb	jsr	lbra	leau	pshs	sbcb	swi
asl	bitb	clr	cwai	lbcc	lbrn	leax	pshu	secc	swi2
asla	ble	clra	daa	lbcs	lbsr	leay	puls	sex	swi3
aslb	bls	clrb	dec	lbeq	lbvc	lsr	pulu	sta	sync
asr	blt	cmpa	deca	lbge	lbvs	lsra	rol	stb	tfr
asra	bmi	cmpb	decb	lbgt	lda	lsrb	rola	std	tst

Nemónicos

Formato del operando.

Para especificar la dirección del operando se tiene la siguiente sintaxis para los diferentes modos de direccionamiento del 6809:

Modo de direccionamiento	Sintaxis	Ejemplo
Extendido	<expresión>	LDA 5f21
Directo	<<expresión>>	LDA <5f
Indirecto	[<expresión>,<registro>]	LDA [2,Y]
Inmediato	#<expresión>	LDA #'G
Indexado	<expresión>,<registro> 0,<registro>++ o...	LDA -4,Y LDB 0,X++

Tres instrucciones aceptan una lista de bits como operando:

```
SECC C,V,F     activa los bits de carry, sobreflujo y FIRQ.
CLCC C,V,F     limpia los bits de carry, sobreflujo y FIRQ.
CWAI F,I       limpia las interrupciones enmascarables.
```

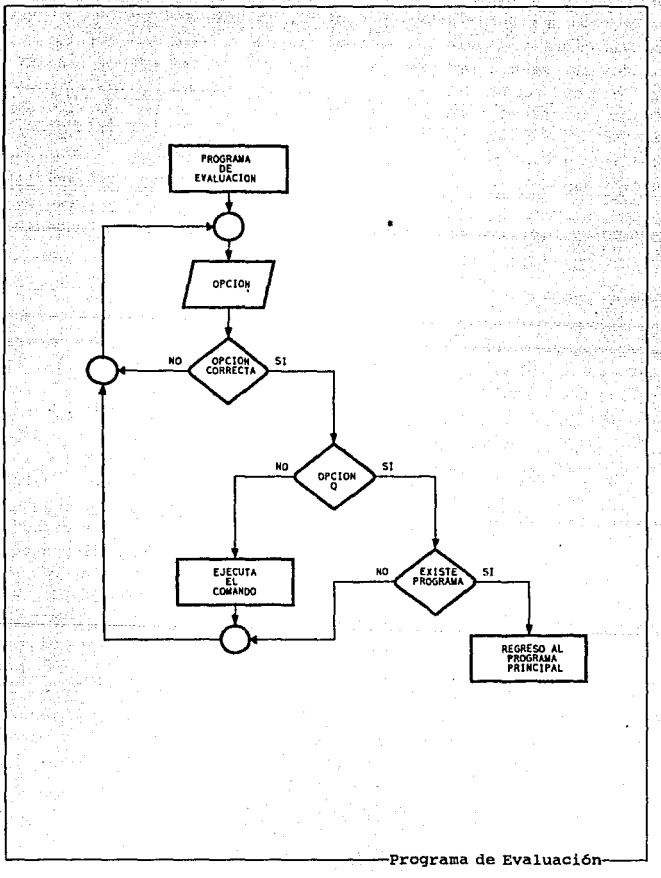
Cuatro instrucciones aceptan una lista de registros:

```
PSHU A,B,X     almacena A,B y X en stack S.
PULS P         realiza la función de RTS.
```

Dos instrucciones aceptan una pareja de registros:

```
TFR AB         transfiere el contenido da A a B.
EXG X,Y        intercambia el contenido de X e Y.
```

El siguiente diagrama de flujo muestra la forma en que se va ejecutando el programa según el código que halla sido introducido.



Programa de Evaluación

Programa Principal.

El programa principal o de control, llamado así por ser el que mantiene a las variables a controlar (en este caso el f.p. y velocidad) dentro de un rango de operación. El programa principal es el encargado de realizar diversas acciones, tales como: Inicialización del sistema, la autoprueba, despliegue de errores, etc. En el caso de detectarse errores que alteran el funcionamiento del sistema de forma considerable, el sistema permanecerá en un estado de bloqueo, del cual no saldrá hasta haberse corregido la falla y reiniciado el sistema. Este programa como se mencionó anteriormente, opera en base a la sucesión de menús, en los cuales se despliegan las posibles acciones a ejecutar.

Autoprueba.

La primera función que realiza el sistema es la ejecución del programa principal. Dentro de éste, una de sus rutinas es el autoprueba, el cual realiza la inicialización del ACIA, el PIA, el teclado, verifica la cantidad de RAM existente, verifica la existencia del programa de evaluación, de la tarjeta de video, etc. Terminando la autoprueba, el sistema despliega el resultado de ésta y el estado del sistema. El despliegue lo realiza de acuerdo a las condiciones de operación que obtuvo de la autoprueba, estas condiciones alteran también la forma en que se llevará a cabo el despliegue del primer menú. Si durante la autoprueba no se detectó la existencia del programa de evaluación, o el de cualquier otro, el menú principal será saltado, siendo el menú de control, el primero en desplegarse. Del como se realiza esta autoprueba se hablará con detenimiento más adelante.

Menú Principal.

El menú principal es desplegado por el programa, cuando se detecto en la autoprueba la existencia de un programa opcional; éste cuenta con dos posibles opciones, las cuales son:

1. Ejecución del programa de control.
2. Ejecución del programa de evaluación.

Si se selecciono la opción 2 se iniciará la ejecución del programa de evaluación, el cual se explico anteriormente. En el caso de seleccionar la opción 1, se desplegará el menú de control.

Menú de control.

El menú de control despliega las siguientes opciones:

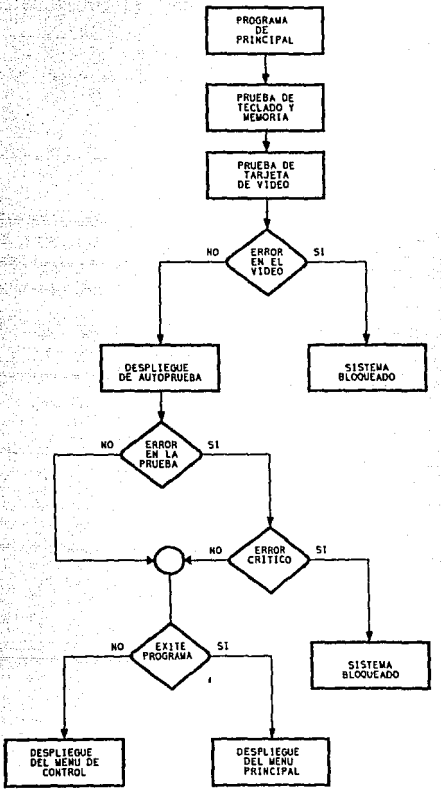
1. Despliegue de mediciones.
2. Control de reactivos.
3. Control de motores.

En la opción del despliegue de mediciones, se presentan los valores correspondientes al voltaje de línea y corriente de línea en RMS, frecuencia y factor de potencia.

En la opción de control de reactivos se presenta un menú en el cual se habilita el monitoreo del factor de potencia, el usuario selecciona el factor de potencia al cual desea que opere el sistema.

En la opción de control de motores se encuentran las opciones de arranque, paro y selección de velocidad de operación de los motores, tanto para DC como AC.

Para volver de un menú al anterior o cancelar alguna operación solo se requiere presionar la tecla ESC. Se muestra enseguida un diagrama de flujo del programa.



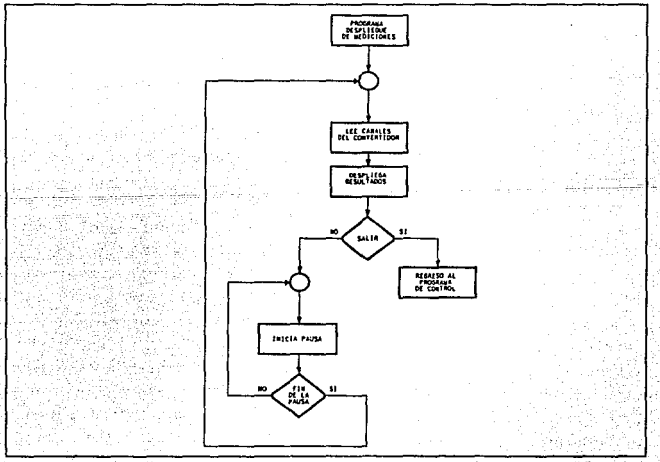
Programa Principal

Programa de Despliegue de mediciones.

Este programa realiza las lecturas del convertidor analógico/digital, de los canales asignados al trasductor de voltaje, corriente y frecuencia, además de leer el canal conectado al circuito de medición de defasamiento. Dentro de este programa se llama a una subrutina, la cual, teniendo como entrada el voltaje proporcional al ángulo que existe entre el voltaje y la corriente, nos entrega el valor correspondiente a dicho ángulo, y si es de adelanto o de atraso.

Estas lecturas son desplegadas en la pantalla en forma de lista y actualizadas cada 0.5 segundos aproximadamente, no hay más comandos dentro de esta opción, ni se ejecuta otra acción, para salir de esta pantalla, solo se debe presionar la tecla de "esc", cualquier otra no es atendida por el programa.

En siguiente diagrama de flujo corresponde a dicho programa.



Programa de Control de Reactivos.

Este programa consta de cuarto partes, estas son:

a) Menú principal.
b) Control manual.
c) Control automático.
d) Corrección del f.p.

A continuación se explican cada uno de los incisos anteriores.

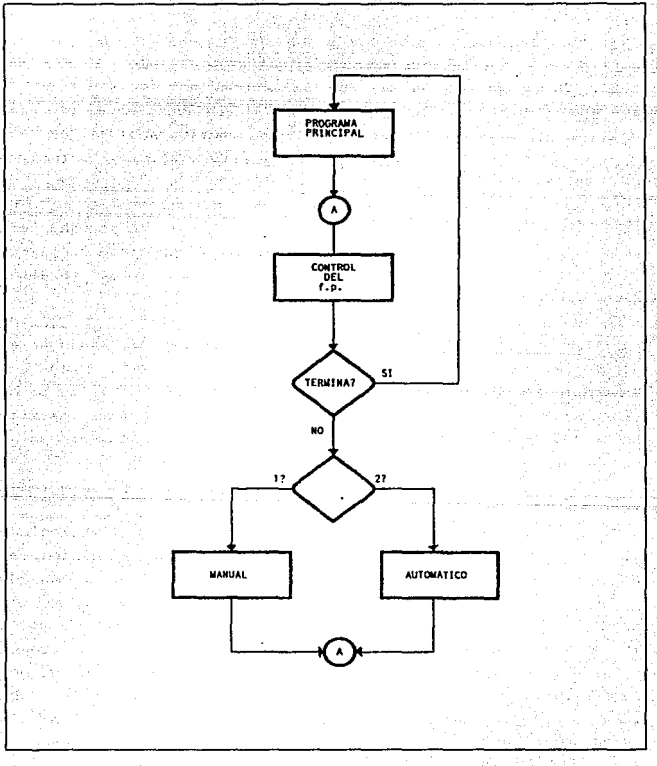
a) El menú principal permite el acceso a los dos tipos de control, es decir el manual y el automático.

b) Con este control es posible corregir el f.p. al conectar y/o desconectar los capacitores manualmente, esto es, se indica el número de los cuatro capacitores existentes, si este está conectado se desconectará, en caso contrario se activará.

c) Este control define el f.p. mínimo al cual debe operar el sistema eléctrico, este es definido por el usuario. (al inicializarse el sistema de corrección y evaluación, al f.p. se le asigna el valor de 0.85). Los capacitores serán conectados y/o desconectados automáticamente para cumplir con el f.p. mínimo establecido.

d) Esta parte del programa funciona cuando se a entrado al control automático es decir se encarga de conectar y desconectar los capacitores (la corrección del f.p. realizada por este programa no se llevará a cabo si se eligió el control manual), mandando señales de error cuando no se puede obtener el f.p. definido por el usuario, o cuando se eligió el control manual y el f.p. del sistema está por debajo del que se estableció. (si no se estableció ninguno entonces está por debajo de 0.85), estos mensajes aparecen en cualquier momento sin importar si se está en los menús del f.p. o no. También se encarga de calcular las potencias aparente y real y desplegarlas en las tres pantallas existentes. Como es necesario

checar continuamente las condiciones del f.p. este programa funciona mediante interrupciones, que se dan aproximadamente cada segundo. Un diagrama de flujo de dicho programa se muestra a continuación.



Programa de Control de Motores.

El programa de control de motores ejecuta dos acciones, el control de un motor de DC de 24v y el control de un motor de AC de 127v.

Seleccionando la opción de motor de DC, en el programa se despliegan las características nominales del mismo, además del estado de operación en que se encuentra. El motor de DC, a diferencia del de AC, puede manejarse independientemente del controlador, debido a un interruptor que configura dicha tarjeta. El programa verifica el estado de dicho interruptor, y conforme a esté habilita o no la posibilidad de modificar el estado de operación del motor. Dicho estado se puede modificar en sentido de giro, velocidad y en encendido o parado. Todo esto es manejado con opciones que se despliegan consecutivamente. El programa despliega además la velocidad a la que gira el motor, en caso de estar encendido.

Para el caso del motor de AC, de manera similar al de DC, el programa despliega las características nominales del motor y el estado de operación en que se encuentra. Para este motor, el controlador tiene completo control sobre el mismo, es decir, no se puede manejar independientemente como el de DC, el programa permite que se modifique el estado de operación del motor únicamente en su velocidad, la cual es monitoreada y desplegada.

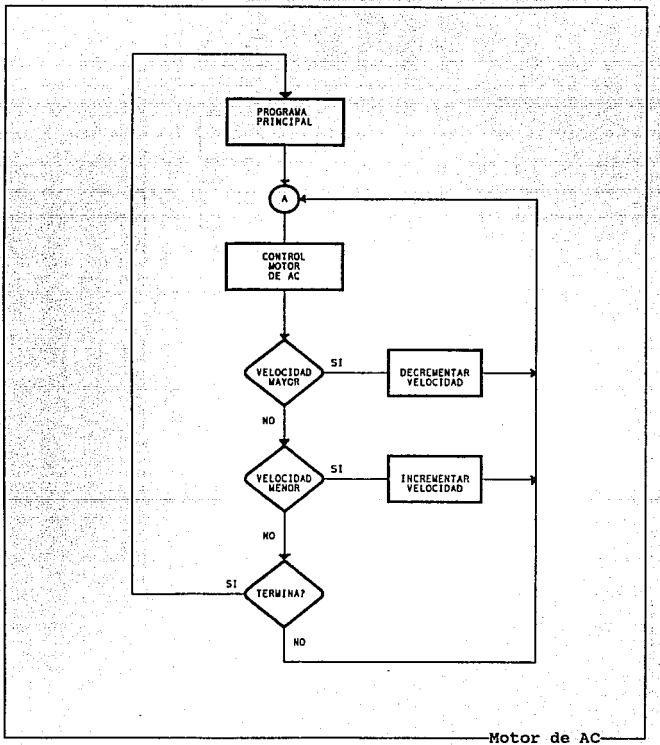
Programa que controla el motor de A.C.

Este programa es un submenú que permite controlar la velocidad del motor, su diagrama de flujo está en la página siguiente, al entrar por primera vez en esta rutina el motor está apagado, cuando se le indican las revoluciones a las cuales debe de girar, la tarjeta de control lo enciende, y le transmite una palabra de 8 bits que corresponde a la velocidad deseada, (esto se comprueba mediante el sensor magnético colocado en el rotor del motor) si por efectos de carga el motor girara a menor velocidad de la que debe trabajar, el programa se encarga de mandar un aumento a el voltaje que se aplica al motor, y compensa con esto el efecto de la carga, si esta carga se quita, ahora el voltaje se disminuye y se logran

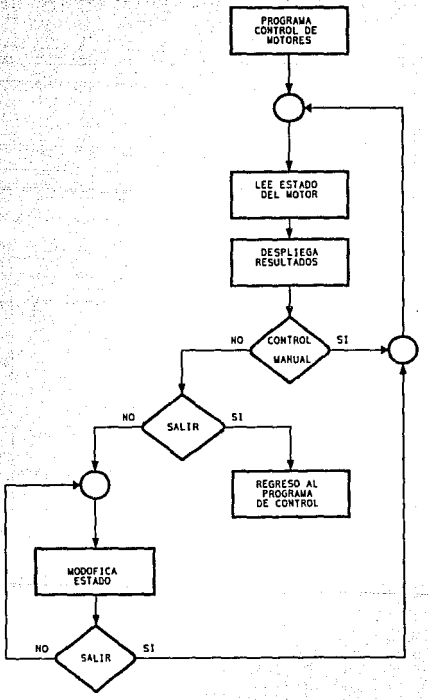
las revoluciones a las que debe girar.

Solo es necesario oprimir la tecla ESC para volver al programa principal.

Los siguientes diagramas de flujo muestran como operan dichos programas.



Motor de AC


PROGRAMA CONTROL DE MOTORES
LEE ESTADO DEL MOTOR
DESPLIEGA RESULTADOS
CONTROL MANUAL
NO
SI
SALIR
NO
SI
REGRESO AL PROGRAMA DE CONTROL
MODOFICA ESTADO
SALIR
NO
SI

Motor de DC

III b.- Hardware Construcción y Evaluación.

Primeramente de acuerdo a nuestras necesidades, hablando de los dispositivos que serían empleados en el sistema, y a la forma en que resolveríamos el problema de diseño se elaboró el siguiente mapa de decodificación:

Direción	Tamaño	Dispositivo:
0000_H $5FFF_H$	24K	RAM1
6000_H $7FFF_H$	8K	RAM/EPROM
8000_H $9FFF_H$	8K	RAM2
$A000_H$ $DFFF_H$	16K	Periféricos (IOS)
E000 $FFFF_H$	8K	EPROM

La función que desempeña cada uno de los dispositivos es:

RAM1 Almacena los programas que el usuario desea verificar, así como las variables y algunos datos que el programa principal o el de evaluación necesitan. Esta memoria también es empleada para el manejo del stack o pila.

RAM/EPROM Estas direcciones pueden ser empleadas por la memoria RAM1 y tener la misma función que ésta, o bien, ser empleadas para el direccionamiento del programa de evaluación o algún programa auxiliar contenido en una EPROM de 2K.

RAM2 Estas direcciones son empleadas para la memoria RAM de video, es decir, se emplea para almacenar los caracteres, que serán desplegados por el VDG en el televisor.

IOS Es empleado para direccionar los dispositivos periféricos del sistema. Cada dispositivo auxiliar ocupa un espacio dentro de este bloque, a continuación se da la ubicación de estos dispositivos:

Dispositivo:	Direccion:	Función:
PIA		
	$A000_H$	Registro periférico y de direcciones A.
	$A001_H$	Registro de control A.
	$A002_H$	Registro periférico y de direcciones B.
	$A003_H$	Registro de control B.
ACIA		
	$A004_H$	Registro de control y estado.
	$A005_H$	Registro de recepción y transmisión.
VDG		
	$A008_H$	Deshabilitación del VDG.
	$A009_H$	Habilitación del VDG.
	$A00A_H$	Deshabilita señal de sincronización.
	$A00B_H$	Habilita señal de sincronización.
	$A00C_H$	Limpia señal de sincronización.
	$A00D_H$	Palabra de estado.
PTM		
	$A010_H$	Registro de control 3 o 1
	$A011_H$	Registro de estado y control 2
	$A012_H$	Contador 1 y buffer del MSB
	$A013_H$	Contador 1 y buffer del LSB
	$A014_H$	Contador 2 y buffer del MSB
	$A015_H$	Contador 2 y buffer del LSB
	$A016_H$	Contador 3 y buffer del MSB
	$A017_H$	Contador 3 y buffer del LSB
ADC		
	$A018_H$	Medición de Frecuencia.
	$A019_H$	Medición de Voltaje.
	$A01A_H$	Medición de Corriente.
	$A01B_H$	Medición del Defasamiento.
	$A01C_H$	Medición de Velocidad.
	$A01D_H$	
	$A01E_H$	
	$A01F_H$	
DAC		
	$A020_H$	Control del Motor de DC.
	$A022_H$	
Control		
	$A024_H$	Paro del Motor de DC.
	$A026_H$	Arranque del Motor de DC.

EPROM Este espacio es ocupado por el programa de evaluación o por el programa principal.

Distribución de Componentes.

Para obtener un sistema modular, el desarrollo del prototipo se realizo en tarjetas bajo el estandar STD-BUS, tomando como base para la división de componentes, las funciones que realizaban cada uno de los elementos que integran el sistema de tal forma el sistema se conformó con tres tarjetas bajo dicho estandar y tres tarjetas auxiliares. Las tarjetas en STD-BUS son:

Tarjeta central de proceso
Tarjeta de despliegue
Tarjeta de adquisición de datos, control y programa adicional.

Dentro de cada una de estas tarjetas, se busco de igual forma, realizar módulos con funciones especificas, aislamiento, decodificación, etc

Las tarjetas periféricas auxiliares son:

Tarjeta de control de velocidad para motor de AC.
Tarjeta de control de velocidad para motor de DC.
Tarjeta de control de la conmutación del banco de capacitores.

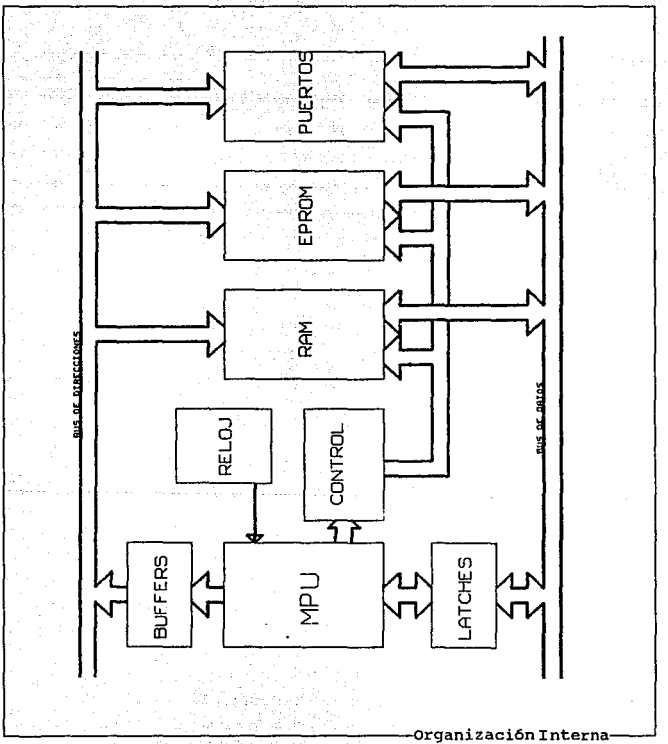
Tarjeta Central de Proceso.

La tarjeta principal esta integrada por el MPU, memoria RAM, memoria EPROM principal, puerto paralelo, puerto serie y el generador de reloj.

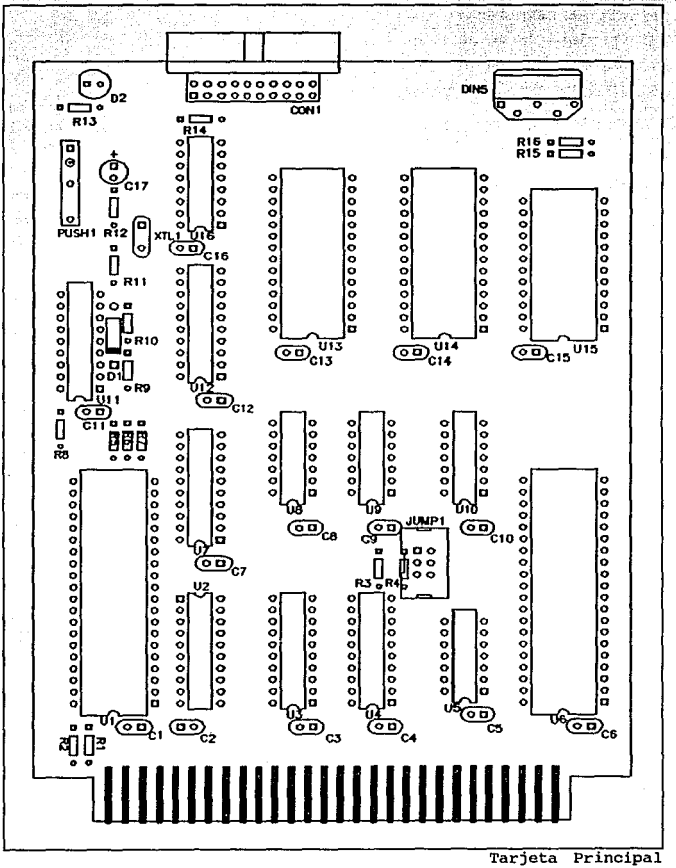
Esta tarjeta es la encargada de realizar todas las operaciones lógicas, numéricas, de control y toma de decisiones, mandando y recibiendo información de las otras dos tarjetas, además de ser la encargada de recibir los comando y datos y enviar palabras de control al mundo exterior, específicamente los datos y comandos recibidos del mundo exterior son provenientes del usuario, éstos son recibidos por medio de la interfaz del teclado y el puerto serie, mientras que las palabras de control son dirigidas a la tarjeta de control de la conmutación del banco de capacitores, con esta palabra se seleccionan los capacitores que entrarán en

operación. Mediante esta conmutación se controlará el factor de potencia y los reactivos que haya en las líneas de transmisión.

Un diagrama esquemático de dicha tarjeta se muestra a continuación, así como la ubicación de los componentes dentro de la tarjeta.



Organización Interna



Tarjeta Principal

Tarjeta de Despliegue

La tarjeta de despliegue es la encargada de realizar la interfaz del sistema al mundo exterior para el despliegue de procesos, resultados, verificación de comandos, etc, que son de utilidad al usuario.

La tarjeta se realizó con base al VDG y cuenta con memoria RAM (memoria de despliegue) circuitos de decodificación, control, desacoplamiento y el modulador.

Los circuitos de decodificación se encargan de identificar y reconocer llamados del MPU al VDG para su habilitación deshabilitación y programación o accesos del MPU a la memoria de despliegue para su actualización o refrescamiento.

Los circuitos de control se encargan de realizar la programación del VDG, habilitar o deshabilitar la señal de sincronización del VDG con el MPU, la cual identifica cuando el VDG ha terminado de hacer un barrido de pantalla y detiene el funcionamiento del mismo; hacer el direccionamiento de la memoria de despliegue para la actualización de información del MPU, así como activar o desactivar los dispositivos de acoplamiento de la tarjeta al canal de comunicación.

Los circuitos de desacoplamiento son buffers los cuales solo entran en función cuando el MPU hace accesos a la memoria de despliegue, en estado normal de operación del VDG, estos se encargan de desacoplar el bus de direcciones y datos de la tarjeta de despliegue al bus del canal (peine), y así evitar conflictos con el bus del MPU cuando no es llamado.

El modulador realiza el acondicionamiento de las señales de luminancia y color que el VDG genera para ser captadas por el televisor, ajustando por medio de un circuito de oscilación el canal en el cual se desea recibir la señal.

Cabe aquí mencionar como opera esta tarjeta. Primero la memoria de despliegue esta dividida en dos bloques funcionales de acuerdo al modo de operación del VDG estos son:

Dirección del Bloque:	Modo de operación:
8000_H-$97FF_H$ 9800_H-$99FF_H$	Gráficos de Color o Resolución Alfanuméricos

La tarjeta puede ser programada para que trabaje en todos los modos de operación del VDG excepto el modo de semigráficos seis, que debido a la implementación no es posible de tener acceso a ésta.

Los comandos así como las características de cada modo se dan a continuación:

Modo de Operación:	Palabra:	Características:	
		Carácter/Fondo	Borde
Alfanumérico	00-07	Verde/Negro	Negro
	08-0F	Negro/Verde	Negro
	20-27	Naranja/Negro	Negro
	28-2F	Negro/Naranja	Negro
Semigráficos	10-1F	Todos/Negro	Negro
SG4	30-3F	Todos/Negro	Negro
Gráfico		Colores	
CG1	80	Todos/Verde	Verde
	A0	Todos/Ante	Ante
RG1	81	Negro/Verde	Verde
	A1	Negro/Ante	Ante
CG2	82	Todos/Verde	Verde
	A2	Todos/Ante	Ante
RG2	83	Negro/Verde	Verde
	A3	Negro/Ante	Ante
CG3	84	Todos/Verde	Verde
	A4	Todos/Ante	Ante
RG3	85	Negro/Verde	Verde
	A5	Negro/Ante	Ante
CG6	86	Todos/Verde	Verde
	A6	Todos/Ante	Ante
RG6	87	Negro/Verde	Verde
	A7	Negro/Ante	Ante

En los modos CG* se tienen todos los colores (verde, amarillo, azul, rojo, ante, cyan, magenta y naranja).
En los modos RG* solo se tiene el negro y el color del borde.

Al encender el sistema el VDG se encuentra deshabilitado, para deshabilitarlo en las siguientes ocaciones, se debe habilitar la señal de sincronia, la cual, una vez que el VDG ha concluido un campo es activada, deteniendo el funcionamiento del VDG e indicando a la vez que dicha acción se ha realizado por medio de la señal de sincronización al MPU (él cual debe de esperar la respuesta en un estado de espera de la señal de sincrinización mediante la instrucción SYNC).

La habilitación o deshabilitación de cualquira de las funciones de la tarjeta de despliegue se realizan con base a direccionamientos *dummy*, es decir, solo es necesario que el MPU presente en su bus de direcciones las localidades que representan las funciones, para que estas sean activadas o no.

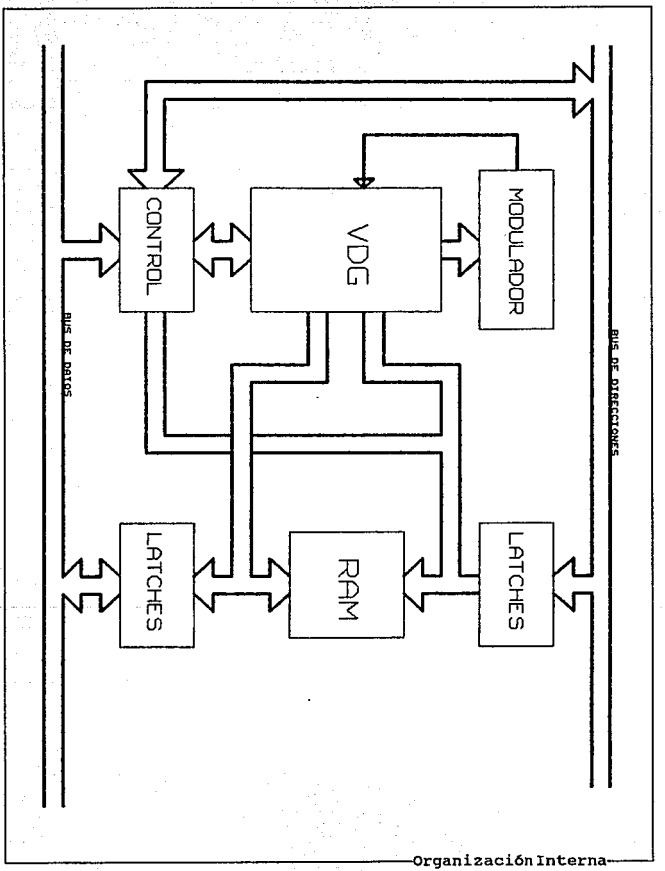
Por ejemplo: si se desea deshabilitar al VDG se tiene que debe de realizar una lectura o escritura a la localidad $A008_H$ para que la acción se cumpla, el dato almacenado o leido según la intrucción empleada no tiene significado.

Sólo cuando el VDG esta deshabilitado se puede programar su modo de operación o bien tener acceso a la memoria RAM de despliegue para verificación o modificación.

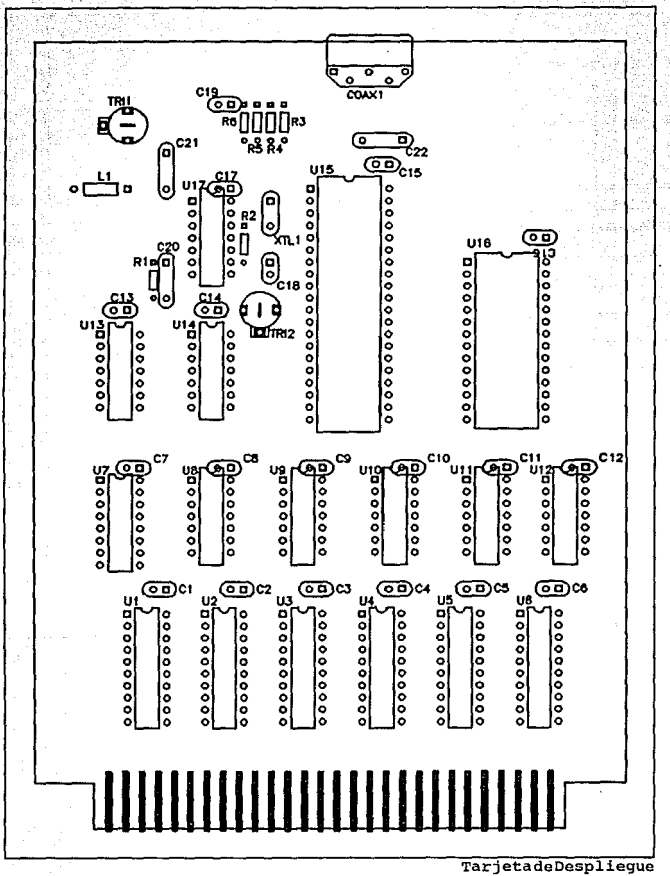
Los pasos a seguir para dicha acción son:

- Se habilita la señal de sincronia.
- Se espera por la señal de sincronización.
- Se deshabilita al VDG.
- Se almacena la palabra de control o bien se modifica el contenido de la RAM.
- Se habilita al VDG.

Un diagrama esquemático de la organización de la tarjeta, así como de la disposición de los componentes dentro de la misma se muestran a continuación.



Organización Interna



TarjetadeDespliegue

Tarjeta de Adquisición de Datos, Control y Programa Adicional.

La tarjeta esta compuesta por dispositivos auxiliares al sistema como memoria EPROM adicional y el PTM (contador de eventos y temporizador); así como los dispositivos de adquisición de datos como son los convertidores Analógico-Digital, los cuales están conectados a los convertidores de corriente a voltaje, voltaje a voltaje y frecuencia a voltaje; y los dispositivos de control que son los convertidores Digital-Analógico y circuitos opto-acoplados.

Como se dijo esta tarjeta tiene sus circuitos de control y de decodificación, cuya función es idéntica a la que desempeñan en la tarjeta del VDG.

Explicaremos a continuación en forma simplificada la función que desempeñan los demás elementos que conforman esta tarjeta.

Memoria adicional. Este espacio es ocupado por el programa de evaluación cuando se quiere añadir esta opción al programa principal.

PTM (Programer Timer Module). Es empleado para interrumpir al MPU cada cierto período de tiempo, esto es empleado para el monitoreo del factor de potencia en forma automática dentro de cierto rango de operación especificado por el usuario o el rango por omisión, además de tener la opción de maniobrar cualquiera de los dos contadores restantes como contadores de eventos ya sean internos (señal de reloj E) o externos, generar interrupciones periódicas, generar señales cuadradas de salida, etc.

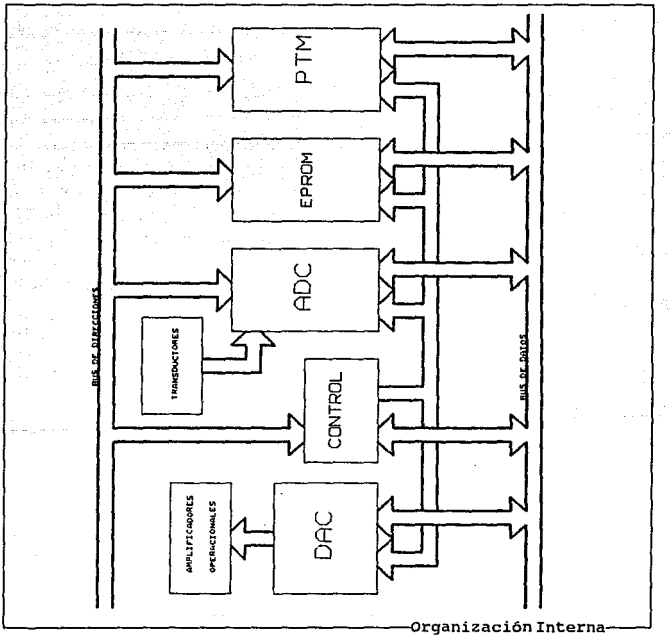
Los dispositivos de adquisición se encargan de acondicionar las señales provenientes de los transductores, para el convertidor analógico a digital y así adecuarla para ser manejada por el MPU, estas señales y sus transductores son:

Señal	Transductor
Voltaje	Transformador de voltaje
Corriente	Amperímetro de gancho (transformador de corriente)
Velocidad	Interruptores de efecto hall

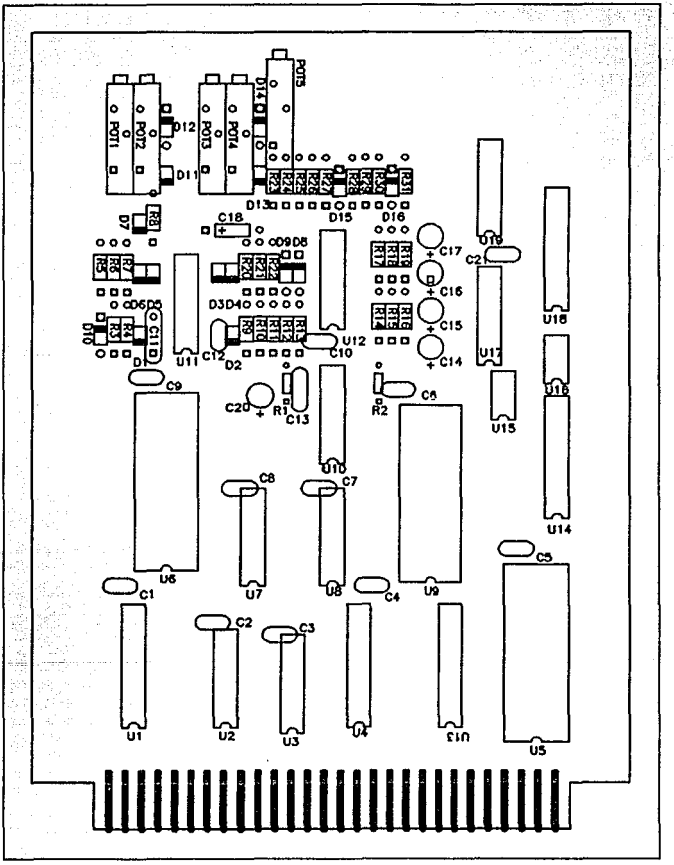
Los dispositivos de control son convertidores digital a analógico que proporcionan niveles de voltaje adecuados a transductores para el control de motores. Esta tarjeta tiene a su vez conectadas las dos tarjetas auxiliares para el control de velocidad de motores una para el motor de DC y la otra para el de AC.

Todos los circuitos empleados junto con sus respectivos transductores se explicaran más adelante.

Un diagrama de bloques de la organización de la tarjeta y de la distribución de componentes se muestra a continuación.

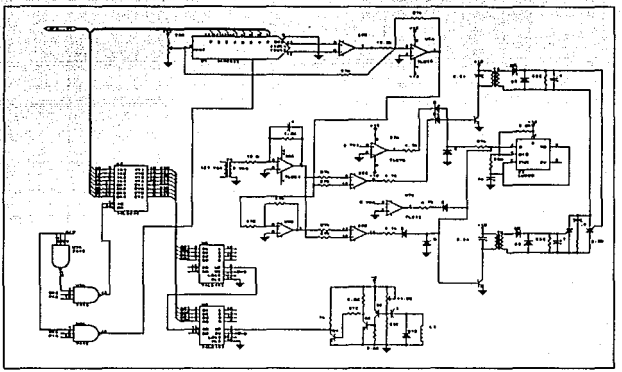


Organización Interna


POT1
POT2
POT3
POT4
POT5
D12
D11
D13
D15
D16
C18
C17
C16
C15
C14
C2
D7
D9
D8
D6
D3
D4
D5
D1
D2
C12
C10
C13
C20
R1
R2
C9
C8
C7
C6
C5
C4
C1
C3
U1
U2
U3
U4
U5
U6
U7
U8
U9
U10
U11
U12
U13
U14
U15
U16
U17
U18
U19

Tarjeta de Control de Motor de AC.

El circuito empleado para controlar la velocidad en el motor de corriente alterna es el que se muestra a continuación.



Existen dos tipos de control para dispositivos que emplean voltaje de C.A. estos son:

1.- Control encendido-apagado.
2.- Control de fase.

El control encendido - apagado, es empleado como "switch" para conectar o desconectar la carga a la fuente.

El control de fase es empleado como "switch" para conectar la carga a la fuente en solo una parte de cada ciclo de la fuente, con esto se logra variar el voltaje eficaz que recibirá el motor, y lográndose así controlar su velocidad.

El último control fue el que se empleó para controlar al motor.

El circuito empleado para el encendido de los tyristores es un

control cosenoidal, que permite tener un incremento lineal de potencia, para cualquier incremento del ángulo de disparo del scr, esto y la explicación del circuito se detallan en el índice.

Un sensor magnético es el que se empleó para detectar las revoluciones a las que gira el motor, cada pulso obtenido se emplea como reloj a dos contadores, los cuales después de un intervalo nos dicen cual es la velocidad del motor, con esto se logra verificar que la velocidad del motor es la que se seleccionó.

Tanto la información que se transmite al motor como la que se recibe son por medio de un puerto paralelo.

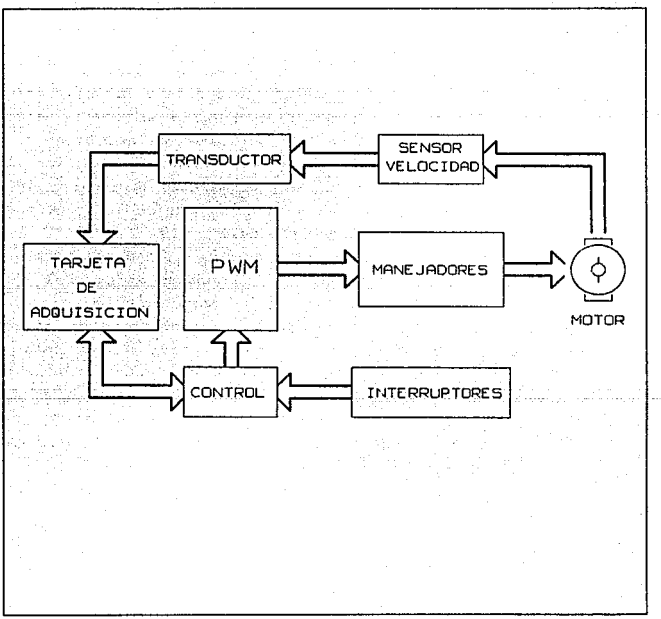
Tarjeta de Control para Motor de DC.

Esta tarjeta se encarga de manejar al motor de DC, de acuerdo a la configuración de un interruptor ubicado en la misma; la función de este interruptor es la de seleccionar uno de dos posibles modos para el manejo del motor y el sentido de giro en uno de estos modos. Esto es, el usuario tiene la posibilidad de seleccionar si el motor será controlado por medio del controlador, con la posibilidad de modificar la velocidad y el sentido de giro; o bien, un operador tendrá dicho control con la limitante de la velocidad, la cual esta fijada a la velocidad nominal.

Cuando se tiene seleccionado el modo de operación automático, el controlador es el encargado de enviar las señales de control a la tarjeta manejadora. Estas señales de control son niveles de voltaje, los cuales indican la velocidad y el sentido en que girará el motor, esta tarjeta se encarga además de recibir las señales provenientes del transductor de velocidad y adecuarla para el convertidor analógico-digital de la tarjeta de adquisición de datos.

El motor se controla con base de un convertidor DC-DC (también conocido como *Chopper* o Troceador). El convertidor DC-DC es un dispositivo que por medio de interruptores, en este caso transistores de potencia, permite regular el voltaje promedio de DC que se aplica al motor. Existen varios tipos de convertidores, el utilizado por nosotros es del tipo D, también llamado puente H o convertidor bidireccional, el cual lleva a cabo la conmutación de los transistores de una misma rama, esto brinda la posibilidad de variar el sentido de giro del motor además de su velocidad. Los

transistores del puente son manejados por medio de un modulador de ancho de pulso (PWM) (*Pulse Width Modulator*), el cual genera un tren de pulsos de frecuencia fija, cuyo ciclo de trabajo varia de acuerdo con una señal de modulación. Esta señal de modulación es la que proporciona la tarjeta de control y es la encargada de indicar el sentido y la velocidad de giro del motor. Con una modulación menor de 50% del ciclo de trabajo, el motor gira en un sentido, si la modulación es mayor del 50% el motor gira en sentido contrario, y con una modulación igual al 50% el motor se detiene. Un diagrama en bloques de la tarjeta se muestra a continuación:



III c.- Sistemas de Autoprueba y Fallas.

El sistema cuenta con la rutina de autoprueba, la cual es ejecutada en el encendido y en cada reinicio del mismo, la función de esta subrutina, como se mencionó anteriormente, es la de inicializar los dispositivos periféricos, esto son:

Puerto Serial Sincróno (ACIA).
Puerto Paralelo (PIA).
Generador del Despliegue de Video.
Dispositivos de adquisición de datos.
Dispositivos de salida.

Además de verificar el estado de la memoria RAM disponible del sistema, tanto de la tarjeta principal como la memoria empleada para el despliegue. Verifica además si parte de las direcciones destinadas a memoria RAM están siendo empleadas por la memoria EPROM del programa de evaluación.

Las fallas que se pueden detectar como anteriormente se menciono son:

Falla en la interfaz con el teclado.
Falla en la memoria RAM principal.
Falla en la tarjeta de video.

Dependiendo de la falla que el sistema haya detectado puede continuar o no su funcionamiento. Con mayor detalle se explicarán las formas en que son indicadas las fallas localizadas por el sistema:

Falla en la Interfaz con el Teclado. Este falla puede deberse a múltiples razones, como son:

- Mal funcionamiento del teclado. Cuando el sistema es encendido el teclado ejecuta su autoprueba (BAT), dentro de esta es localizada una falla y el teclado lo notifica enviando un código de error en la ejecución del BAT.

- Mal funcionamiento del ACIA. Cuando el dispositivo periférico que se encarga de realizar la interfaz con el teclado esta en mal estado y no funciona adecuadamente, el sistema puede estar recibiendo de esté datos y comandos erróneos.
- Mala conexión entre el teclado y el ACIA. Cuando el conector no esta bien conectado.
- La combinación de cualquiera de los casos anteriores.

Esto se puede corregir si la comunicación del ACIA y el teclado se hiciera bidireccional, y no en un solo sentido, claro esta que esto complica tanto el software como el hardware de manera considerable.

El sistema notifica al usuario de esta falla durante el despliegue, en el televisor, de los resultados de la autoprueba, detiene cualquier ejecución y espera a que la falla se corrija y se reinicie el sistema; además de encender un led indicador de esta falla.

Falla en la Memoria RAM Principal. Cuando la autoprueba a detectado un error en la memoria RAM de la tarjeta principal el sistema lo notifica al usuario durante el despliegue de resultados mediante la asignación del numero de localidades de RAM que se encuentran disponibles, el usuario puede verificar rápidamente de acuerdo a la configuración que se seleccionó si hay o no falla en RAM.

Falla en la Tarjeta de Video. La autoprueba puede localizar fallas en la tarjeta de despliegue, estas fallas pueden ser ocasionadas por:

- Falla en la memoria de despliegue. El sistema de evaluación encuentra que la memoria RAM empleada para el despliegue no funciona adecuadamente o no hay.
- Falla en la programación del VDG. El sistema al inicializar el VDG verifica que no trabaja acorde al modo de operación que se le asigno.
- Falta de la tarjeta de despliegue. Que la tarjeta no este en el módulo o no este bien conectada.

El sistema notifica al usuario por medio de el encendido de un led indicador de esta falla, además de detener la ejecución de cualquier operación.

Tanto la falla en la interfaz del teclado como en la tarjeta de despliegue detienen la ejecución del programa y cualquier acción de control, es decir, el sistema queda parado (en estado de "halt"); dicho estado es notificado al usuario por medio del encendido de un led indicador de detenido.

Falla en las señales de entrada. Ahora bien para la detección de fallas en la adquisición de señales de la línea, se cuenta con la rutina que monitorea el factor de potencia dentro de un rango establecido con anterioridad (si no se da un f.p. inicial, el sistema toma el valor de .85), esta rutina es ejecutada por medio de interrupciones periódicas, las cuales son ejecutadas por el PTM, cuando ha llegado a esta rutina, se verifica el estado de sistema y de algunas señales de importancia, por ejemplo niveles de voltaje, corriente en la línea así como de su frecuencia, esto con el propósito de detectar posibles fallas en la línea de transmisión o cortos circuitos, también se verifica la velocidad de motores y la alimentación que poseen para la detección de sobrecargas o un mal funcionamiento.

En el caso de localizar alguna falla el sistema responde con el comando de control adecuado como el cierre o apertura de interruptores, para cortar el suministro de corriente o encender alguna alarma. Además de enviar un mensaje de advertencia al usuario por medio del televisor.

III d.- Sistemas de Blindaje y Aislamiento entre Etapas.

Aislamiento Entre Etapas.

Para aislar las diferentes etapas del manejo e intercambio de información entre las tarjetas, se usaron buffers 74LS244 y 74LS245, este ultimo en el caso del bus de datos bidireccional, para que sean estos dispositivos quienes suministren la corriente necesaria y no sobrecargar a los circuitos que controlan o "mandan" dicha información.

Estos buffers fueron colocados en los siguientes sitios:

- A la salida del Microprocesador, en particular en el bus de direcciones así como en el de datos.
- A la entrada de la tarjeta de video en el bus de direcciones y de datos.
- A la entrada de la tarjeta de interfaces en el bus de datos.

Blindaje

Al trabajar con señales de RF se debe tener cuidado que las señales que se reciben del medio no afecten a nuestra información así como también el ruido generado por nosotros.

Es por esto que al circuito modulador de RF le fue colocado una caja de Faraday, y todos los gabinetes usados son de metal, siendo el del modulador el que también cuenta con tapa de metal.

También se utilizó cable coaxial y conector RCA.

Además de estos aislamientos se emplearon otros como:

- Optoacopladores para aislar la etapa de control y la de potencia en la tarjeta controladora del motor de DC.
- Transformadores para aislar la etapa de control y la de potencia en la tarjeta controladora del motor de AC.
- Optoacopladores para aislar la etapa de control y la de potencia en la tarjeta de control de la conmutación del banco de capacitores.
- Amplificadores operacionales en configuración segidor de voltaje para aislar algunas partes en la circuteria analógica como en el convertidor de frecuancia a voltaje.

III e.- Interfaces para Adquisición de Datos.

Medición de voltaje, corriente, fase, f.p. y frecuencia.

Uno de los puntos principales de esta tesis consiste en realizar las mediciones de voltaje, corriente, fase y frecuencia, estos son esenciales, si se desea conocer las condiciones de potencia a las que está trabajando cualquier sistema eléctrico de potencia, al realizar las mediciones anteriores se puede calcular:

S potencia aparente.
P potencia real.
Q potencia reactiva.
f.p. factor de potencia.
cos θ
sen θ

Mediante las siguientes ecuaciones para un sistema monofásico:

$$S = VI = \sqrt{P^2 + Q^2}$$

$$P = S\,SEN\,(\theta) = V\,I\,COS\,(\theta) = \sqrt{S^2 - Q^2}$$

$$Q = S\,SEN(\theta) = VI\,SEN(\theta) = \sqrt{S^2 - P^2}$$

Nuestro sistema que está basado en el MPU 6809 requiere una interfase para comunicarse con el sistema eléctrico de potencia, es decir que convierta señales analógicas a digitales, para lograr esto se empleó el convertidor ADC0809, el cual tiene las siguientes características:

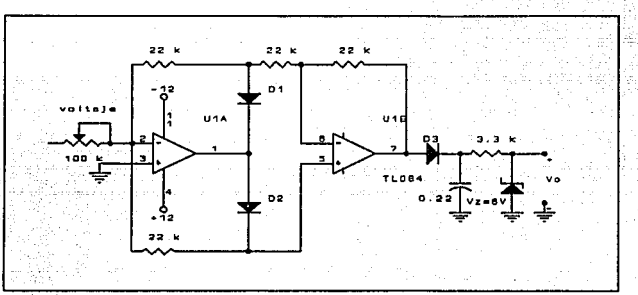
El ADC6809 es un dispositivo CMOS, con 8 canales multiplexados, sobre un convertidor analógico-digital de 8 bits (A/D), y un control lógico compatible con microprocesadores. Los 8 canales multiplexados pueden ser controlados a través de un decodificador de 3 bits, el cual es activado con el bus de direcciones para poder seleccionar una de las ocho entradas analógicas, conectadas directamente con un comparador, la conversión se realiza empleando el método de aproximaciones sucesivas, una entrada de 3 estados, un "switched capacitor", un muestreador, y registro que mantiene los 8 bits obtenidos. Los métodos de comparación y conversión empleados eliminan la posibilidad de omitir códigos, así como la necesidad de ajustar a cero o a escala máxima. El ADC0809 tiene 8 entradas analógicas, es decir ocupa 8 localidades en el mapa de memoria;

A008 - A00F

La decodificación puede observarse en el mapa de memoria. así como su diagrama de tiempos presentado en este capítulo.

Medición de voltaje y corriente.

Para la medición de voltaje y corriente se empleo un solo circuito, este se encarga de rectificar la señal alterna (de voltaje o de corriente) y dar su equivalente en voltaje de D.C. para lograr esto se empleó el circuito siguiente

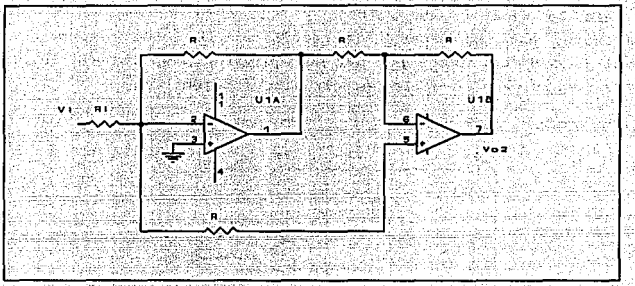


El circuito anterior es un rectificador de onda completa que nos permite rectificar las señales de corriente y de voltaje para poder ser usada en el MPU. Este circuito debe analizarse en dos partes:

a) cuando $V_i > 0$

b) cuando $V_i < 0$

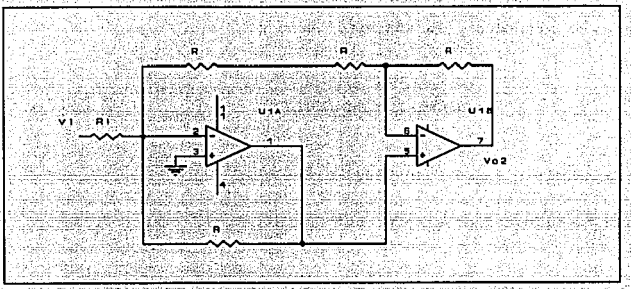
A) Si V_i es positiva el diodo D_1 conduce debido a que el AO es un inversor, y D_2 no conduce, con estas consideraciones, se puede obtener un circuito equivalente:



y su salida esta dada por:

$$V_{oa} = \frac{R}{R_1} V_i$$

b) Si V_i es negativa el diodo D_2 conduce y D_1 no conduce, su circuito equivalente es:



y su salida es:

$$V_{ab} = -\frac{R}{R_1} V_i$$

Con la suma de ambas respuestas se tiene la salida total.

$$V_o = \begin{cases} \frac{R}{R_1} V_i & para \quad V_i > 0 \\ -\frac{R}{R_1} V_i & para \quad V_i < 0 \end{cases}$$

f.p.

El coseno del ángulo de faseθ entre el voltaje y la corriente se conoce como el *factor de potencia* (f.p.). Un circuito predominantemente inductivo tiene un f.p. en atraso y un circuito predominantemente capacitivo tiene un f.p. en adelanto, es decir, f.p. en atraso y f.p. en adelanto, respectivamente, indican cuando la corriente está atrasando o adelantando el voltaje aplicado.

A la cantidad cosθ se le llama f.p. y es la razón entre la potencia promedio y la aparente; esto es,

$$f.p. = \frac{potencia\ promedio}{potencia\ aparente} = \cos(\theta) = \frac{P}{V_o I_o}$$

El ángulo θ se denomina ángulo del f.p..

La impedancia de la carga es:

$$Z = \frac{V}{I} = \frac{V\angle\phi_1}{I\angle\phi_2} = \frac{V}{I}\angle(\phi_1-\phi_2) = \frac{V}{I}\angle\theta$$

de modo que el ángulo de la carga es el ángulo del f.p.. La forma general de una impedancia de carga es:

$$Z = R + jX$$

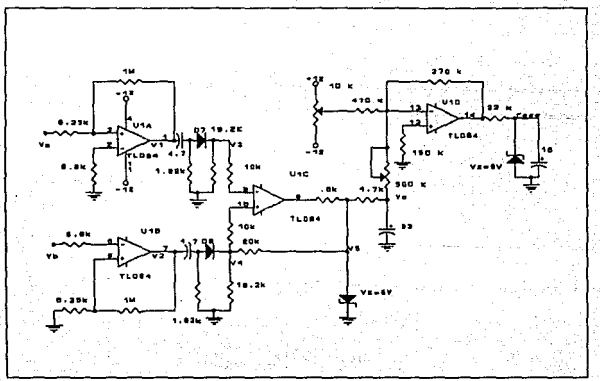
Si X > 0, la carga es inductiva, si X < 0 se dice que la carga es capacitiva.

Una mejora en el f.p. global del sistema para una determinada carga dada, da por resultado.

1) liberar una capacidad de alimentación que permite cargas adicionales.
2) Reducir la caída de tensión de línea y mejorar la regulación de tensión global del conjunto.
3) Aumentar el rendimiento global del sistema (línea equipo)
4) Costo de funcionamiento menores.

Medición del defasamiento entre el voltaje y la corriente

Debido a que en los sistemas eléctricos de potencia es de gran importancia medir el defasamiento entre el voltaje y la corriente, ya que una vez obtenido el defazamiento entre al voltaje y la corriente, se puede conocer el f.p. al cual se esta trabajando, así como las potencias que se generan en todo sistema eléctrico estas son; S (potencia aparente), P (potencia real), Q (potencia reactiva), el circuito mostrado a continuación logra darnos el equivalente en voltaje de D.C. del defasamiento



De las salidas de los amplificadores operacionales A_1 y A_2 se obtienen unas señales cuadradas debido a que trabajan como comparadores.

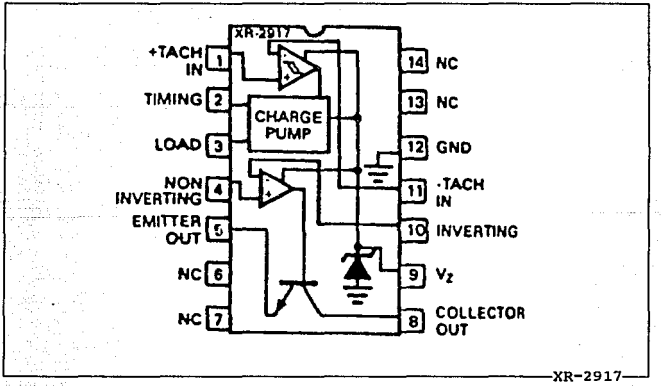
Debido a los diodos que se tiene y a la configuración resistencia capacitor provocan una caída exponencial a las entradas de A_3.

A_3 es un flip flop analógico por lo tanto solo da dos estados lógicos "0" y "1", dependiendo de lo que reciba a las entradas.

A la salida del amplificador operacional A_3 la señal cuadrada que proporciona tiene un ciclo de trabajo variable, este depende del defasamiento existente entre el voltaje y la corriente, es decir tiene un nivel de D.C. variable y proporcional al defasamiento el cual se mantiene estable por medio del capacitor a la salida y del amplificador A_4, para una mejor comprensión de este circuito refiérase al apéndice.

Medición de frecuencia

Para la medición de frecuencia se empleó un convertidor de frecuencia voltaje XR-2917.



Este consiste de: un comparador a la entrada con 40 [mV] de histéresis "charged-pump", diodo zener, una salida de un amplificador operacional y un transistor, tiene una buena linealidad y alta corriente de salida.

El voltaje de salida es una función del voltaje regulado por el zener (V_z), una resistencia (R_1) y un capacitor (C_1), los cuales son conectados a la "charged-pump" y a la entrada de frecuencia (f_i), la reducción del rizo es lograda por medio de un capacitor (C_2). El transistor de salida puede manejar una corriente máxima de carga de 40 [mA] y V_{CEmax} de 28 [V].

El XR-2917 convierte una entrada de frecuencia a una salida de voltaje proporcional, las entradas diferenciales proveen de histéresis para un excelente rechazo del ruido y la capacidad de poner los niveles de conmutación a las entradas de los comparadores, las entradas no deben tener voltajes menores a cero sin antes poner resistencias a las entradas.

La salida del comparador está alimentando a la "charged-pump" donde la corriente es enviada hacia el capacitor C_1 que realiza la función de tiempo de carga, la misma corriente es transmitida a la resistencia de carga R_1, donde el capacitor C_2 realiza el filtrado, donde es usado para integrar los pulsos y proveer un voltaje proporcional a través de la resistencia de carga. El voltaje a través de la resistencia de carga está en función de: la fuente de voltaje, frecuencia de entrada, tiempo de carga del capacitor y de la resistencia de carga.

$$V_{R1} = V_z \, f_1 \, C_1 \, R_1$$

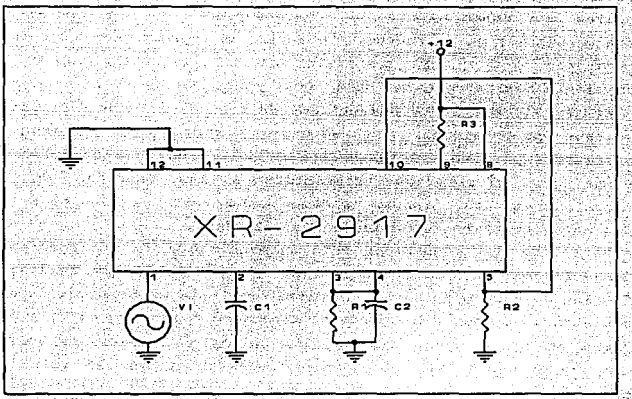
El tamaño del capacitor que realiza la función de integración (C_2) es dependiente únicamente de los requerimientos de tiempo de respuesta y rizo a la salida.

La salida del amplificador operacional y del transistor son usados para manejar las corrientes a la salida; finalmente la ecuación de conversión es:

$$V_0 = V_z \, f_i \, C_1 \, R_1 \, K$$

K es típicamente 1.

A continuación se muestra el circuito empleado.



Las ecuaciones de diseño son:

$$V_{outmax} = R_1 \, I_3$$

Donde I_3 = 170 [μA] si Vcc = 12 [V]

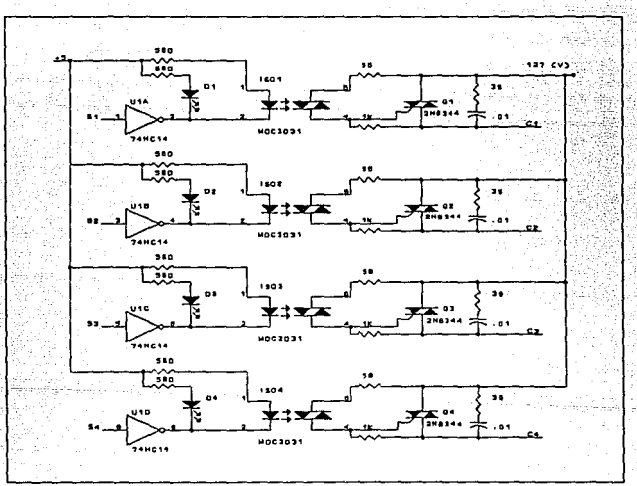
$$f_{in\,max} = \frac{I_2}{C_1 \, V_z}$$

El voltaje de rizo está dado por:

$$V_r = \frac{V_z}{Z} + \frac{C_1}{C_2}\left(1 - \frac{V_z\, f_{in}\, C_1}{I_3}\right) [pp]$$

Corrección del f.p.

Una vez conocido el ángulo de desfasamiento entre la corriente con respecto al voltaje, y que este no este entre el rango permitido (mayor de 85) será necesario corregirlo, esto se logra mediante diferentes métodos como lo son, motores síncronos, capacitores, etc., nosotros empleamos 4 capacitores para corregirlo, y se usó el siguiente circuito.



Cuando se transmite un 1 (proveniente del PIA)se encienda un LED que indica el capacitor que se a encendido, esto es, se a encendido un optotriac, que a su vez enciende un TRIAC de potencia

que conecta a los capacitores con la línea, pero es conveniente hacerlo en el cruce por cero de la señal, el optotriac empleado tiene un detector de cruce por cero, esto se aprecia en la hoja de datos del MOC311, en donde se observa que el optotriac trabaja cuando recibe un pulso que coincide en el cruce por cero de la señal de voltaje.

No

Existe

Pagina

Conclusiones

Del Sistema de Evaluación y Control del MPU 6809 (SEC 6809) podemos concluir lo siguiente:

a) Como sistema de evaluación presenta una gran flexibilidad por permitir al usuario escribir sus programas a evaluar directamente en lenguaje ensamblador y accesar a cualquier parte del sistema.

b) En su aplicación como controlador industrial el SEC 6809 nos permite obtener lecturas de frecuencia, voltaje y corriente y calcular el factor de potencia así como también el valor de los reactivos presentes, ajustándolos al valor prestablecido. Como una alternativa para un mejor calculo del factor de

potencia se propone el análisis de los armónicos, en lugar de la utilización del ángulo de defasamiento como el aquí empleado.

c) El desempeño de la parte controladora de velocidad de motores (DC y AC) presenta varios inconvenientes. Sin embargo, para los fines demostrativos del SEC 6809, son suficientes.

d) En su elaboración se utilizó el MPU 6809, y sus periféricos de manera discreta. Actualmente todo esto podemos encontrarlo en un solo circuito integrado, por ejemplo el MCU 68HC11. Ofreciéndonos también una reducción considerable de espacio.

e) Para el manejo del despliegue de la información se utilizó el VDG 6847, que permite el manejo de caracteres alfanuméricos, semigráficos y gráficos. En el modo gráfico, se obtiene la mejor resolución, pero para un manejo eficiente y adecuado, se requiere de un mayor manejo de software.

f) En relación a la interfaz del teclado para PC implementada en el SEC 6809 se utilizó el 6850 (ACIA, puerto serie), se propone la sustitución de dicho circuito por el empleado en las PC'c.

g) Este sistema fue diseñado para trabajar en sistemas monofásicos, con la posibilidad de extenderse a trifásicos, con un incremento de hardware.

h) El prototipo esta formado por tarjetas, las cuales cumplen funciones determinadas, con la posibilidad de probar cada una de ellas en forma independiente, a la vez que permite el sustituirlas por prototipos mejorados.

i) En México se cuenta con la capacidad técnica para desarrollar sistemas de buena calidad.

APENDICE

A

CALCULOS

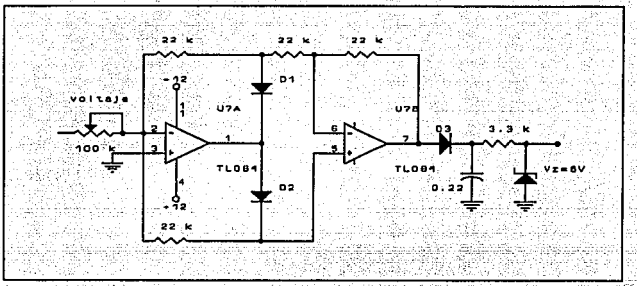
Apéndice A

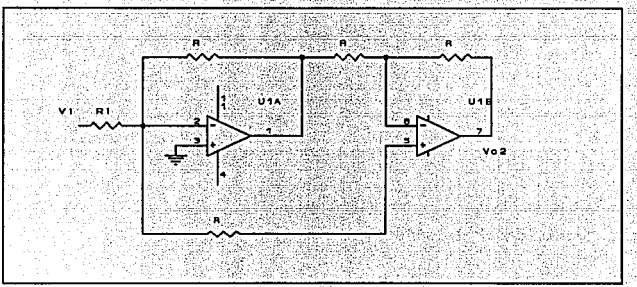
A.1 Rectificador de onda completa.

El siguiente circuito con amplificadores operacionales, es un rectificador de onda completa, cuyo análisis se presenta a continuación:

a) Cuando $v_i > 0$ se tiene el siguiente circuito equivalente;

Las ecuaciones de corriente en los nodos A y B considerando a los amplificadores operacionales ideales son:





$$\sum i_A = 0 \quad ; \quad \frac{V_{o1}}{R} + \frac{V_i}{R_1} \quad \ldots . (1)$$

$$\sum i_B = 0 \quad ; \quad \frac{V_{o1}}{R} + \frac{V_o}{R} \quad \ldots . (2)$$

de 1;

$$V_{o1} = -\frac{R}{R_1} V_i$$

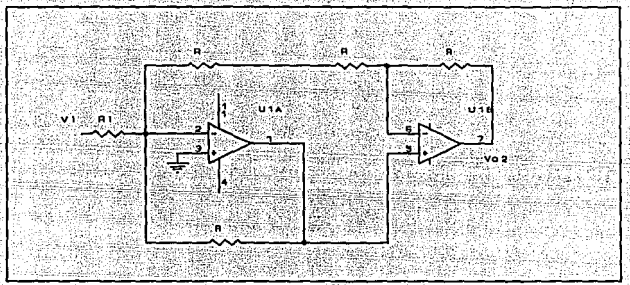
de 2;

$$V_o = -V_{o1}$$

por lo tanto;

$$V_o = \frac{R}{R_1} V_i$$

b) Cuando $V_i < 0$, el circuito equivalente es:



$$\sum i_D = 0 \quad ; \quad \frac{V_o - V_{-2}}{R} - \frac{V_{-2}}{2R} = 0 \quad \ldots . (3)$$

$$\sum i_C = 0 \quad ; \quad \frac{V_i}{R_1} + \frac{V_{-2}}{2R} + \frac{V_{o1}}{R} = 0 \quad \ldots . (4)$$

$$V_{o1} = V_{+2} = V_{-2} \quad \ldots . (5)$$

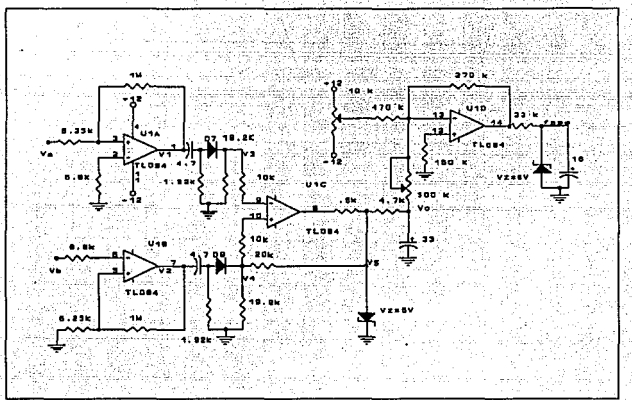
de 3;

$$V_o = \frac{3}{2} V_{-2} \quad \ldots . (3')$$

sustituyendo (3') y (5) en (4) y despejando V_o;

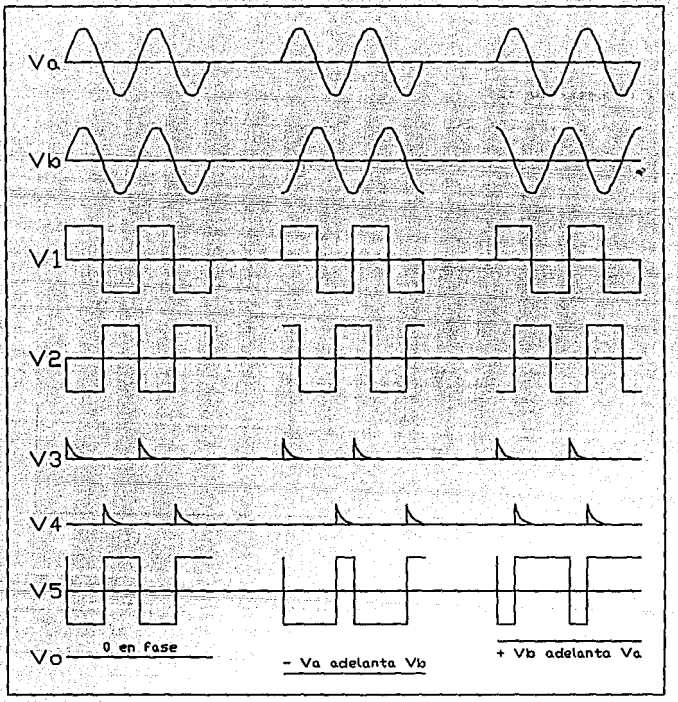
$$V_o = -\frac{R}{R_1} V_1$$

A.2 Detector de fase



Su funcionamiento es:

Los amplificadores A_1, A_2 son comparadores ya que; A_1 tiene una configuración inversora solo que la entrada esta en la terminal no inversora (da una condición inestable) y A_2 una configuración no inversora pero la entrada esta en al terminal inversora, esto provoca que A_1 y A_2 estén siempre saturados (en +Vcc o -Vcc), gráficamente se pueden ver las salidas de A_1 y A_2 (en el circuito son los voltajes V_1 y V_2, y sus gráficas se pueden observar en la página siguiente) cuando a la entrada se tiene una señal senoidal, Va y Vb.



Debido a los diodos la señal que ira hacia A_3 será la que se muestra en las gráficas, para los voltajes V_3 y V_4, para que V_3 ocurra, debe de haber una transición de nivel bajo a nivel alto de V_1, lo mismo sucede para V_4, para la transición de V_2.

Un momento antes que se de el pulso, es decir A_1 está saturado en $-V_{cc}$, C_1 tiene la siguiente condición de voltaje (su voltaje es igual a $-V_{cc}$) y su gráfica de voltaje en el capacitor cuando se da la transición es la que se da en la hoja de gráficas, y que corresponden a los voltajes V_3 y V_4.

Para evitar que el capacitor se descargue lentamente e interfiera con la llegada del siguiente pulso (o del pulso de la entrada V_+ si el pulso en ese momento es en la entrada V_-) este debe cumplir lo siguiente:

$$R_7\ C_1 = R_8\ C_2 < \frac{1}{2f_c} = \frac{1}{2}\ T$$

Para que R_9 y R_{10} (que provocan una condición de 0 [V] a las entradas de A_3 cuando el pulso no existe) no provoquen una descarga demasiado rápida de los capacitores C_1 y C_2, cuando los diodos D_1 y D_2 conduzcan, es decir la corriente por R_9 y R_{10} debe ser insignificante con respecto a R_7 y R_8, una buena condición para lograr esto es que se cumpla:

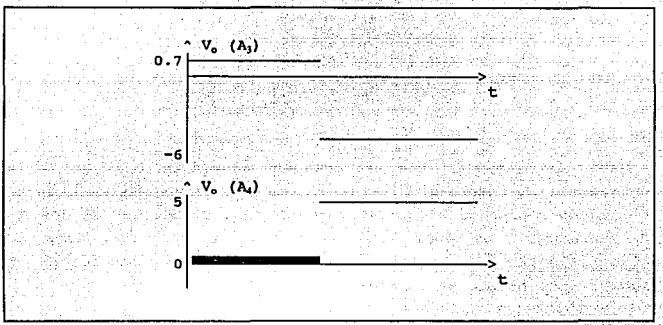
$$R_7 < 0.1\ R_9$$

A_3 es un flip-flop analógico es decir cambia de +Vcc a -Vcc dependiendo de la señal que le sea aplicada a las entradas, esto puede observarse en la hoja de gráficas (V_3 y V_4), cuando llega la señal de V_3 la salida V_5 va hacia nivel bajo, y cuando se da la señal V_4, V_5 va hacia nivel alto.

Este flip-flop tiene un nivel de histéresis, las resistencias R_{10} , R_{13} y el diodo zener establecen los niveles de histéresis, estos niveles se dan con la siguiente relación:

$$V = \frac{R_{10}\, V_z}{R_{10} + R_{13}} \quad [V]$$

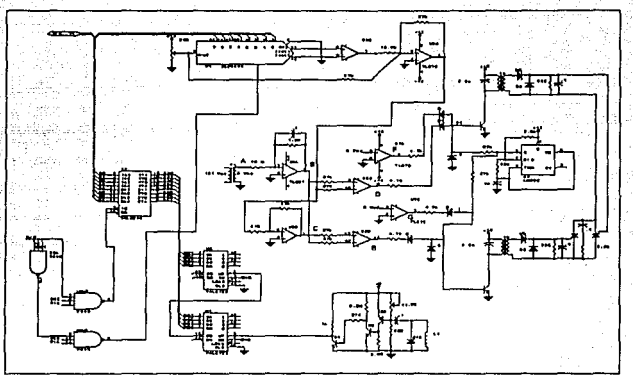
A_4 nos permite ajustar la salida de A_3 a un nivel de voltaje de 0 a +5 [V], esto debe hacerse ya que a la salida de A_3 se tiene un rango de -0.7 a +6 [V], esto se observa en las gráficas siguientes:



Esto se logra al aumentar la ganancia del A_4 (pot 3) y al sumarle un voltaje (pot 4).

A.3 Control de Velocidad del Motor de A.C.

El circuito empleado es el siguiente:



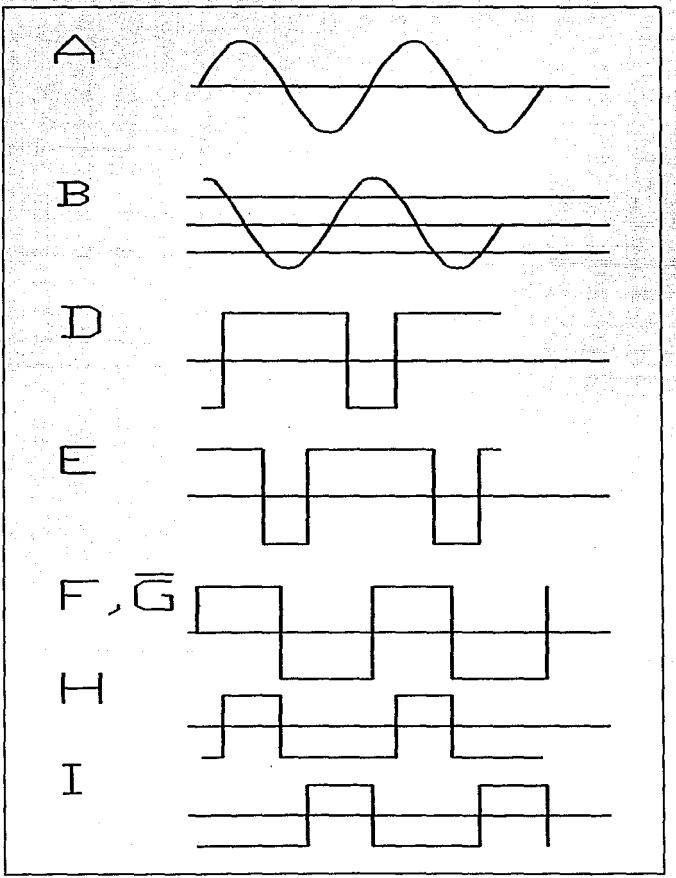
Como se mencionó anteriormente el control de encendido de los scr, es conocido como control cosenoidal, los amplificadores que realizan esta función son los A.O. del U6A al U6D.

U6A es un integrador, y su salida está dada por:

$$V_o = -\frac{1}{R_1 C}\int V_i \, dt$$

V_i es una senoidal por lo tanto su repuesta es un coseno, con ganancia de $1/R_1C$, para una mejor comprensión de su funcionamiento observese las gráficas de la siguiente página.

El voltaje que sirve de referencia (véase la gráfica B) e indica el momento en que los scr se enciendan lo transmite el DAC0830 y que corresponde a la palabra lógica transmitida por el


A
B
D
E
F, $\bar{G}$
H
I

MPU, para la velocidad que el usuario requiera que tenga el motor, U6B invierte el voltaje de referencia (uno para cada scr), D y E, dan un "1" cuando el scr deba dispararse, pero este nivel alto dura más del medio ciclo, por eso es necesario limitar a el nivel alto únicamente para su correspondiente medio ciclo, esto se logra con los A.O. U6F y U6G, gráfica F,G, para dar la salida deseada H e I.

El circuito de disparo de los scr lo forman, un oscilador, transformador excitado por un transistor, este a su vez es controlado por los A.O., en el lado del secundario, el voltaje se rectifica (para proteger la compuerta del scr contra voltajes en inversa que lo dañarían), y es aplicado a la compuerta del scr.

El circuito detector de frecuencia lo forman tres transistores, que se encargan de amplificar el pulso proveniente del sensor magnético, este pulso sirve como reloj para dos contadores, estos reflejan la frecuencia a la que trabaja el motor, y será transmitido al MPU por medio del puerto paralelo.

A.4 Control de Velocidad del Motor de D.C.

El motor empleado en el controlador es un motor con imán permanente. Una ventaja de utilizar este tipo de excitación es su relativa facilidad de fabricación, una mayor eficiencia con respecto a los motores que emplean devanados, y además se eliminan cables de conexión. Además con este tipo excitación, se tiene un control de velocidad a través de la tensión de inducido, con una relación prácticamente lineal entre estos dos parámetros.

$$T = K_T \phi I_a$$

Donde:

$$K_T = \frac{PZ}{4a}$$

Donde:

P es el número de polos.

Z es el número de conductores activos, el doble de espiras por bobina.

a es el número de ramas, las bobinas en paralelo entre las escobillas de polaridad opuesta.

Este control de velocidad basa su funcionamiento en la ecuación:

$$\omega_m = \frac{V_a - I_a R_a}{K_a \phi}$$

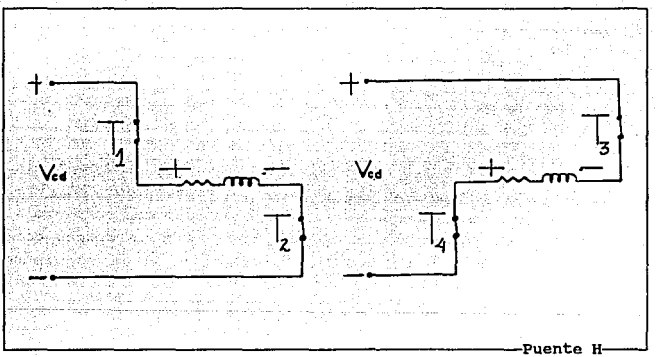
Existen tres tipos de regulación de velocidad:

a) Regulación reostática: Manteniendo V_a y ϕ constantes, puede disminuirse la velocidad, aumentando la resistencia de armadura. Este tipo de regulación es ineficiente, ya que consume energía en el reóstato.

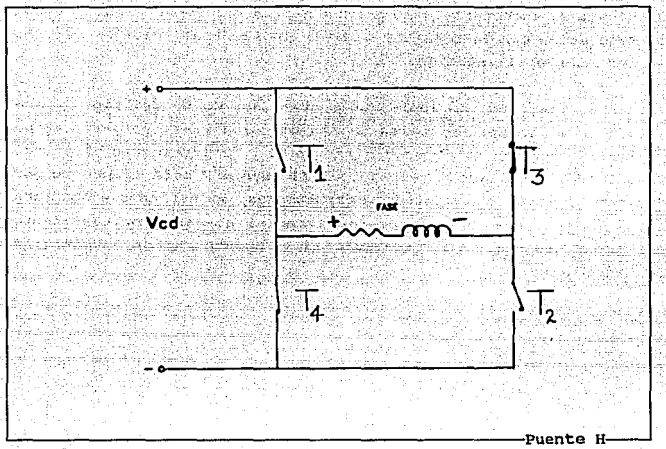
b) Regulación por flujo: Después del arranque se inserta un reóstato para poder reducir el flujo en el circuito del inductor. Esta regulación es buena, excepto por el hecho de que cuando se

disminuya ϕ, aumentará I_a, lo que puede producir sobrecalentamiento.
c) Regulación por la tensión: Se basa en variar el voltaje aplicado al inductor o al inducido.

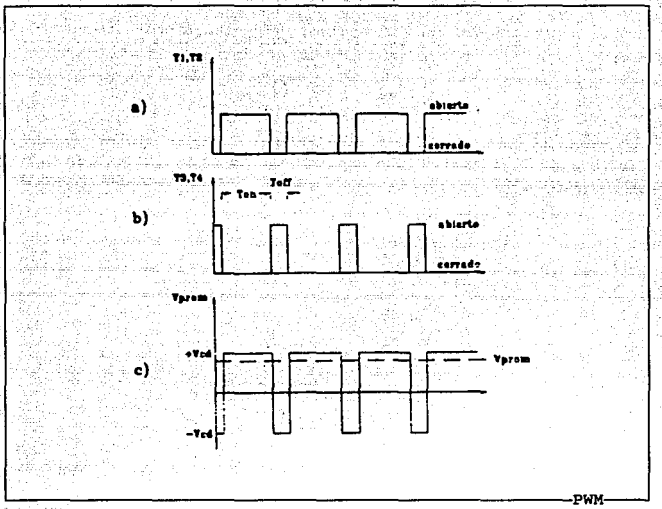
Como se mencionó, se empleo un convertidor CD-CD del tipo D para alimentar al motor. Los pares de conmutadores T1-T2 y T3-T4, abren y cierran de forma alternada; mientras T1 y T2 permanecen abiertos, T3 y T4 están cerrados y viceversa. El puente H está compuesto por dos ramas, y cada rama por dos conmutadores. Se define que una rama es la trayectoria cerrada que se forma al actuar simultáneamente dos conmutadores. La tipología del puente H se muestra a continuación:



En la siguiente figura se puede ver que una rama esta definida por T1 y T2; la otra rama la forma T3 y T4. Se observa también que cuando conduce T1 y T2, tomando como referencia la polaridad de la fase, la fuente (V_{cd}) se aplica con polaridad positiva, mientras que para la otra rama, se induce con polaridad negativa.



Si se conmuta con frecuencia fija la apertura y cierre de las ramas, el voltaje promedio que entrega el convertidor (V_{prom}) se regula por la razón de tiempo que una rama conduce con respecto al tiempo de conducción de la otra. A esta técnica se le conoce como modulación de ancho de pulso. Esto se observa en la siguiente gráfica:



Si la velocidad de conmutación es alta con respecto de la constante de tiempo eléctrica de la bobina de la fase del motor, ésta funcionará de acuerdo al voltaje promedio de la señal (V_{prom}). El voltaje promedio alimentado al motor por el puente H se define como:

$$V_{prom} = \frac{T_{on} - T_{off}}{T_{on} + T_{off}} V_{cd}$$

Y si se define como ciclo de trabajo (d) a :

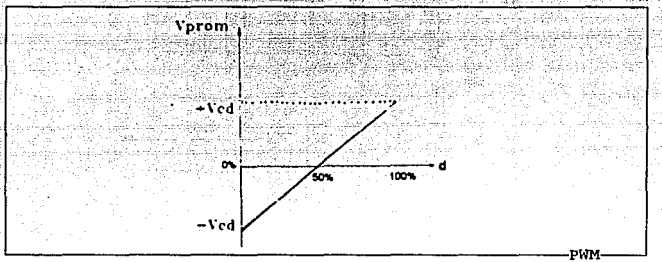
$$d = \frac{T_{on}}{T}$$

donde:

τ es el período del tren de pulsos que comanda el puente. Combinando las ecuaciones anteriores, el voltaje promedio esta dado por:

$$V_{prom} = (2d - 1)\, V_{cd}$$

El comportamiento de dicho funcionamiento se muestra en la figura siguiente:



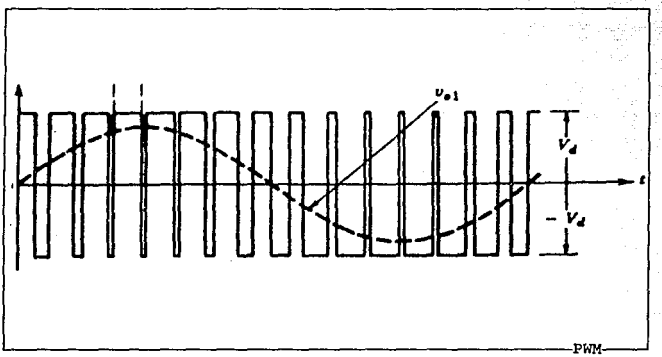
En la anterior gráfica se muestra que la variación del voltaje promedio es proporcional al ciclo de trabajo (d), variando este ciclo se podrá controlar la velocidad y el sentido de giro del motor.

Se determino que los transistores conmutarán a 20 kHz, por ser una frecuencia fuera del rango audible para el ser humano y por que el período de dicha conmutación es mucho menor a la constante eléctrica del motor. Esta constante (τ_e) esta definida por:

$$\tau_a = \frac{L}{R}$$

La conmutación de los transistores se realizó por medio de un PWM. La función de éste es generar un tren de pulsos de frecuencia fija (20 kHz) cuyo ciclo de trabajo varía en función del voltaje de una señal de entrada (moduladora).

Tradicionalmente, una señal modulada en ancho de pulso es obtenida a través de un comparador de voltaje que tiene a su entrada una señal moduladora (V_c) y una señal diente de sierra, cuando el nivel de voltaje del diente de sierra es menor que la señal moduladora, se obtiene un nivel lógico alto a la salida, y cuando el diente de sierra es mayor, se obtiene un cero lógico como se observa en la siguiente gráfica:

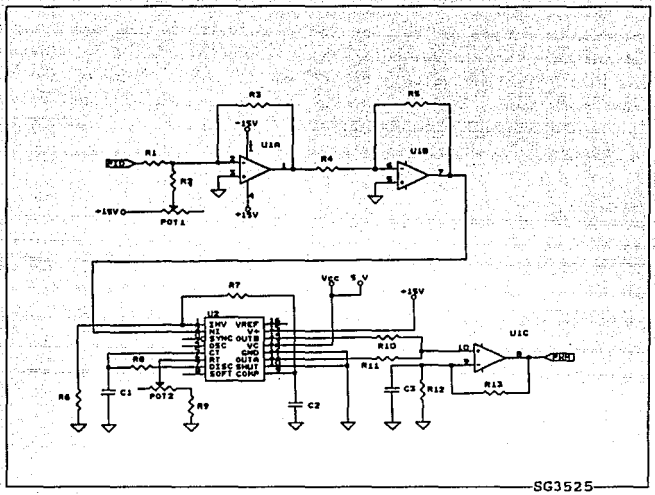


Así, el tren de pulsos es de la misma frecuencia que el diente de sierra y el ancho de pulso o ciclo de trabajo (d) estará dado por:

$$d = \frac{V_c}{V_m}$$

donde V_m es el voltaje máximo del diente de sierra.

Para la implementación física del PWM se empleó el circuito integrado SG3525 de motorola con la siguiente configuración:



SG3525

Ya que la señal diente de sierra generada internamente por el SG3525, tiene un valor pico a pico de 3.1v y un nivel de CD de 1.2v, y para que la relación entre el voltaje de salida del amplificador y el voltaje de comando sea lineal, es decir, se debe lograr que cuando el voltaje de entrada al PWM sea de 0v, se tenga un ciclo de trabajo del 50%, el cual corresponde a un voltaje promedio cero a la salida del convertidor CD-CD, la señal de entrada debe acondicionarce a dicho nivel de DC, de modo que pueda compararse correctamente. El acondicionamiento de la señal se lleva a efecto en dos amplificadores operacionales, el primero de ellos, un sumador inversor, atenúa la señal de entrada para que quede contenida dentro del voltaje pico a pico de la señal diente de sierra, además de sumarle un nivel de DC. El segundo operacional tiene como finalidad el recuperar la polaridad original y desacoplar al PWM.

En el circuito integrado se usa el amplificador de error interno en una configuración no inversora de ganancia unitaria, esto se logra al conectar la terminal 9 (COMP) a la entrada inversora del amplificador (terminal 1) por medio de una resistencia de 33kΩ y conectando la entrada inversora a tierra por medio de una resistencia de 1MΩ. Para poder variar el ciclo de trabajo más allá del 50%, se suman las salidas A (terminal 11) y B (terminal 14) por medio de un sumador no inversor, y se elimina el tiempo muerto entre ellas, conectando una resistencia de 1Ω entre las terminales 5 y 7 (DISC y C_t respectivamente), el valor de esta resistencia se encuentra de acuerdo a la gráfica R Vs. tiempo muerto. La frecuencia de la señal del PWM se fija con un capacitor y una resistencia conectados entre C_t (terminal 5) y R_t (terminal 6) a tierra, la resistencia se forma por la conexión en serie de una resistencia de valor fijo y un potenciómetro que sirve de ajuste fino. La compensación de frecuencia se hace con un capacitor de 1nF conectado entre COMP (terminal 9) y tierra. Como no se utiliza la protección contra sobrevoltaje del integrado, la terminal 10 (SHUT) se deshabilita conectándola a tierra.

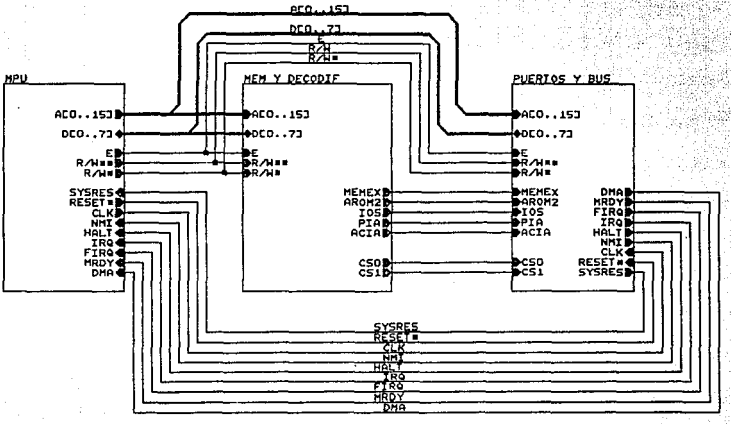
APENDICE

B

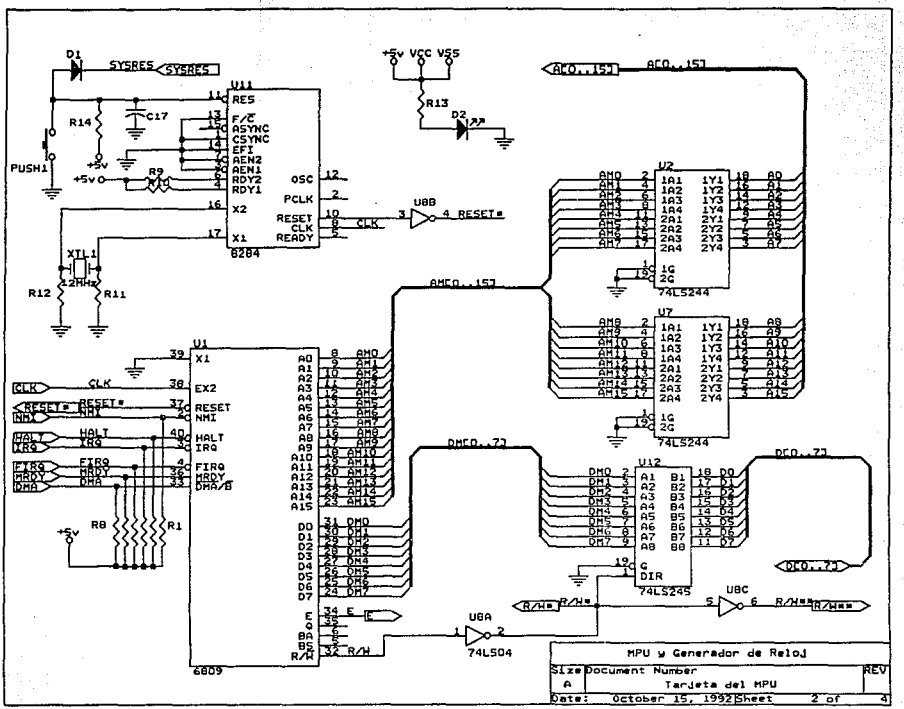
DIAGRAMAS ELECTRICOS

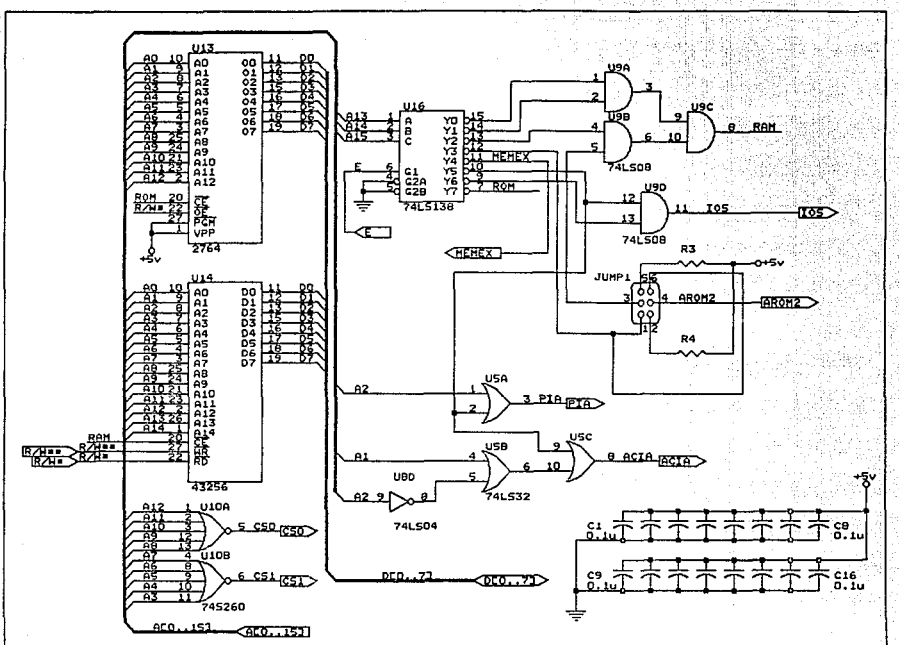
Apéndice B

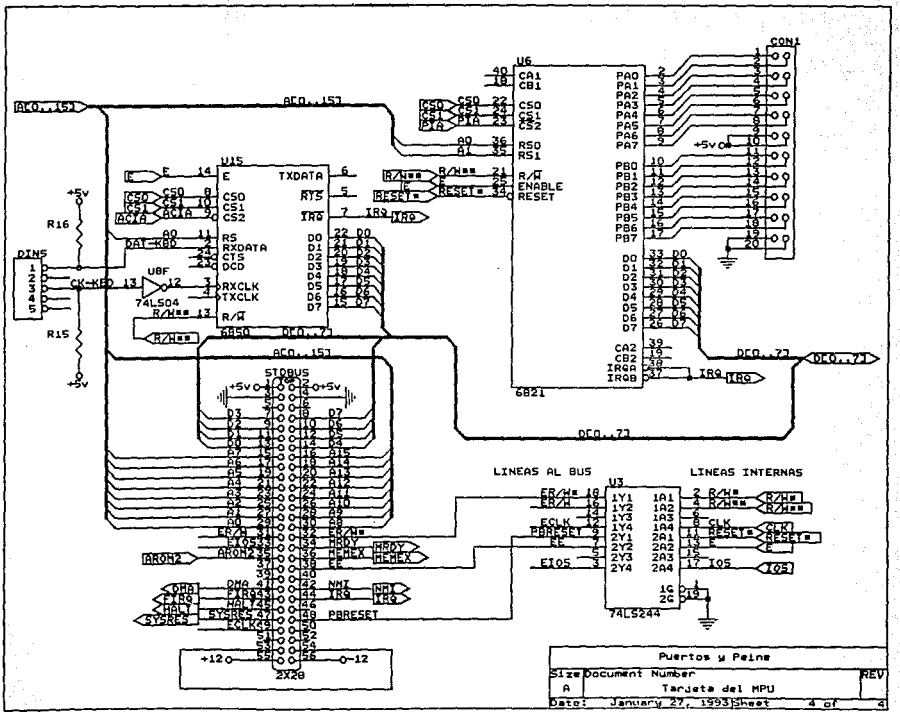
B. Diagramas Eléctricos.

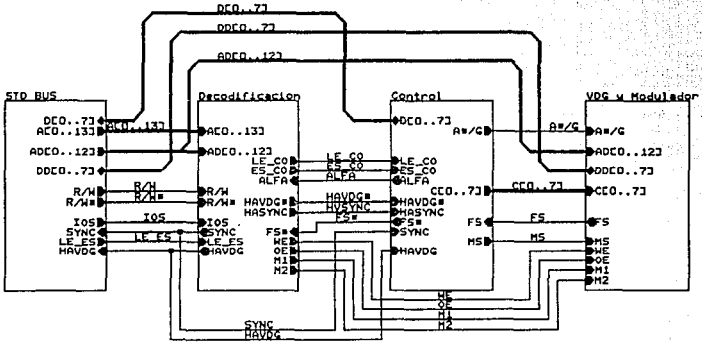


Organizacion		
Size	Document Number	REV
A	Tarjeta del MPU	
Date: October 15, 1992	Sheet 1 of 4	

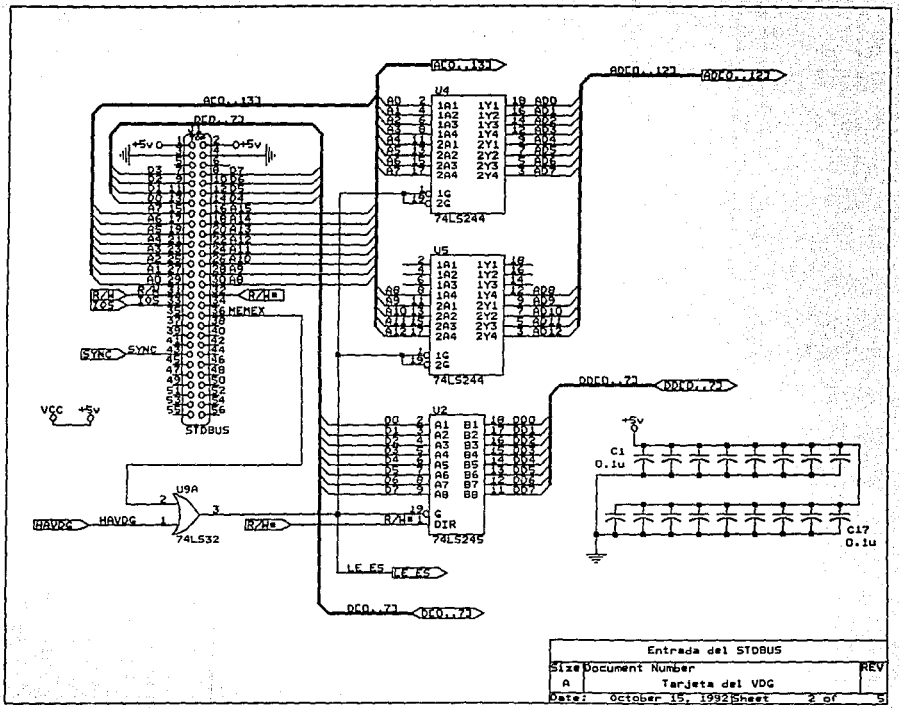

D1
SYSRES
SYSRES
U11
RES
F/C
ASYNC
CSYNC
EFI
AEN2
AEN1
RDY2
RDY1
OSC
PCLK
RESET
CLK
READY
X2
X1
8284
R14
C17
PUSH1
+5v
R9
XTL1
12MHz
R12
R11
+5v VCC VSS
R13
D2
U8B
CLK
4 RESET*
AMD0..15
U1
X1
EX2
RESET
NMI
HALT
IRQ
FIRQ
MRDY
DMA
DMA/B
AD0
AD1
AD2
AD3
AD4
AD5
AD6
AD7
AD8
AD9
AD10
AD11
AD12
AD13
AD14
AD15
DD0
DD1
DD2
DD3
DD4
DD5
DD6
DD7
E
Q
BA
BS
R/W
6809
R8
R1
+5v
U2
74LS244
U7
74LS244
DMD0..7
U12
74LS245
DIR
DD0..7
R/W
R/W*
U8A
74LS04
U8C
R/W**
R/W***
MPU y Generador de Reloj
Size
A
Document Number
Tarjeta del MPU
REV
Date: October 15, 1992
Sheet 2 of 4

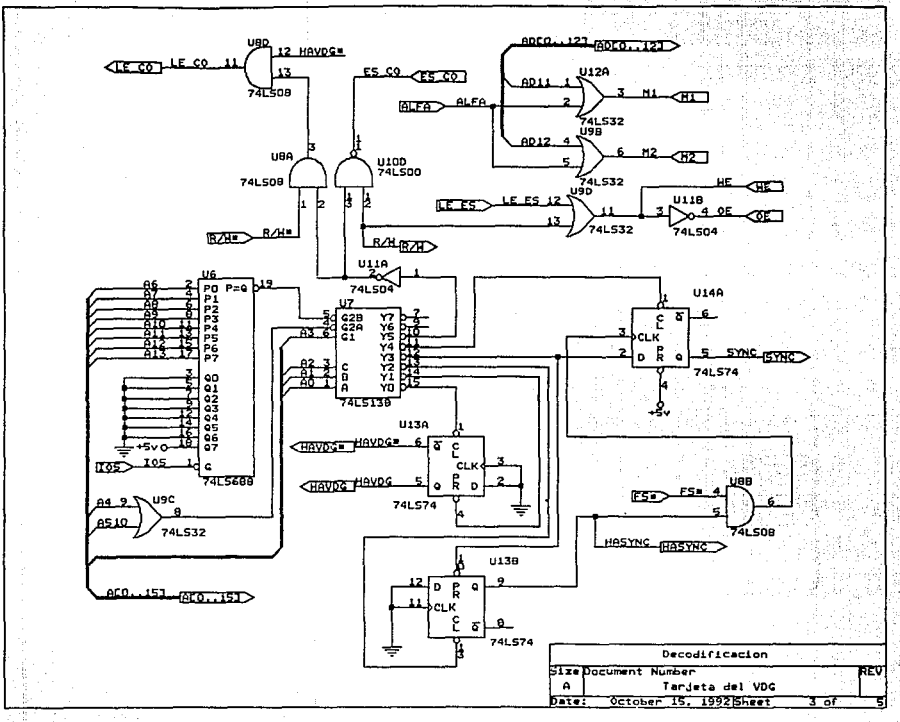

U13
2764
U14
43256
U16
74LS138
U9A
U9B
U9C
74LS08
U9D
74LS08
U5A
U5B
U5C
74LS32
U8D
74LS04
U10A
U10B
74S260
R3
R4
JUMP1
+5v
RAM
R/W
MEMEX
ROM
AROM2
PIA
ACIA
IO5
C50
C51
C1
C8
C9
C16
0.1u
DFO..7]
DCO..7]
AFO..15]
ACO..15]
Decodificacion, EPROM y RAM
Size
A
Document Number
Tarjeta del MPU
REV
Date: January 27, 1993
Sheet 3 of 4

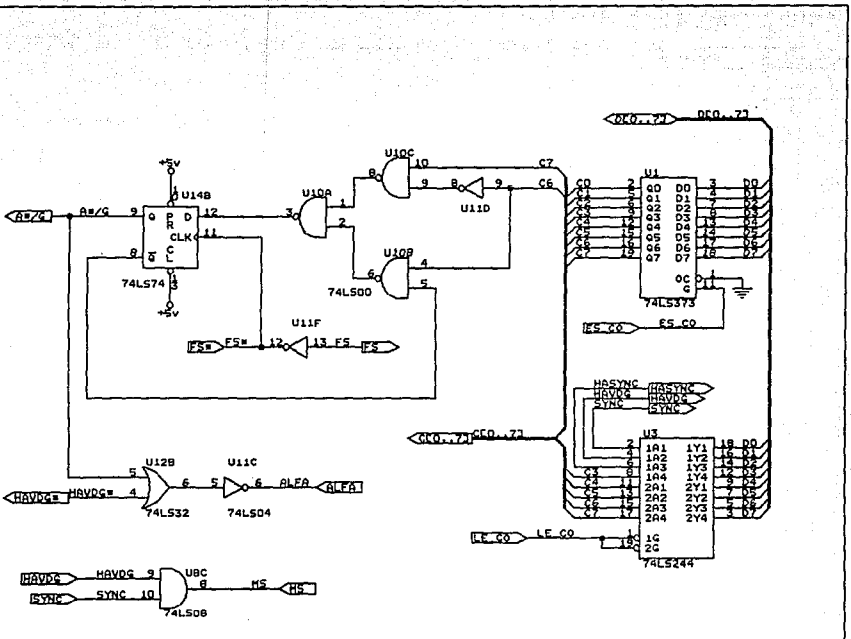

U6
6821
U15
6850
U8F
74LS04
DIN5
R16
R15
CON1
STDBUS
2X20
U3
74LS244
LINEAS AL BUS
LINEAS INTERNAS
Puertos y Peine
Tarjeta del MPU
Size A
Document Number
REV
Date: January 27, 1993
Sheet 4 of 4



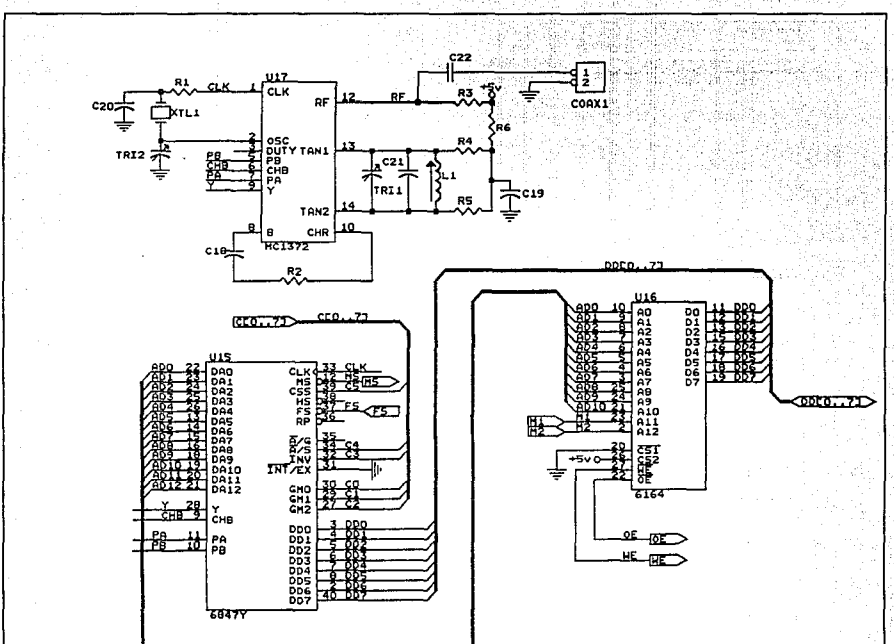
Organizacion		
Size: A	Document Number: Tarjeta del VDG	REV:
Date: October 15, 1992	Sheet 1 of 5	

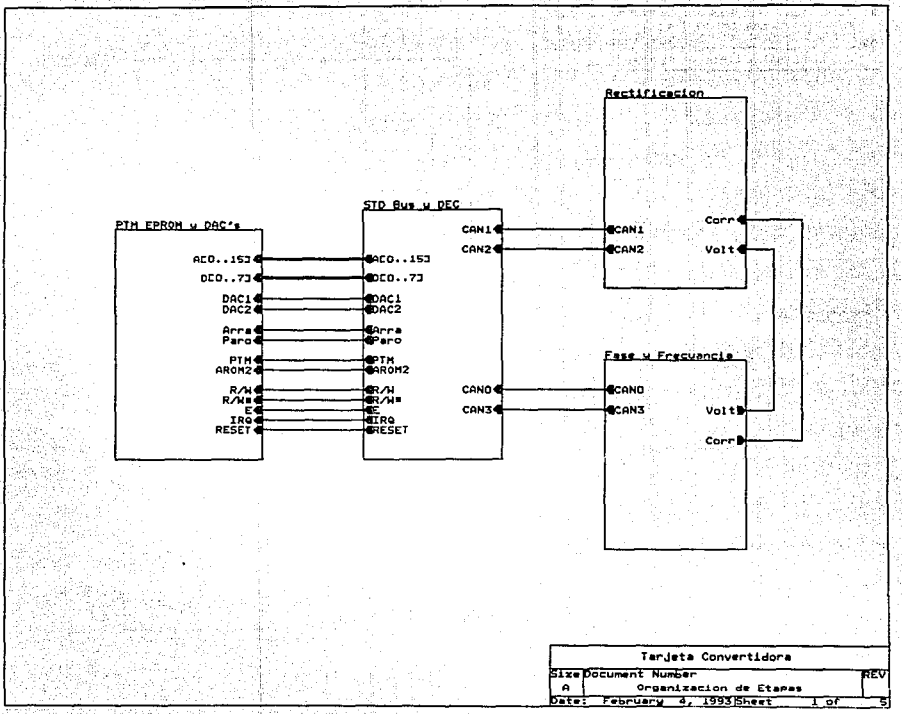

ADO..13
ADO..13
ADO..7
ADO..7
ADO..12
ADO..12
U4
74LS244
U5
74LS244
MEMEX
STDBUS
SYNC
SYNC
R/W
R/We
R/We
VCC
+5v
DDO..7
DDO..7
U2
74LS245
DIR
U9A
74LS32
HAVDG
HAVDG
LE ES
LE ES
+5v
C1
0.1u
C17
0.1u
Entrada del STDBUS
Size
Document Number
REV
A
Tarjeta del VDG
Date: October 15, 1982
Sheet 2 of 5

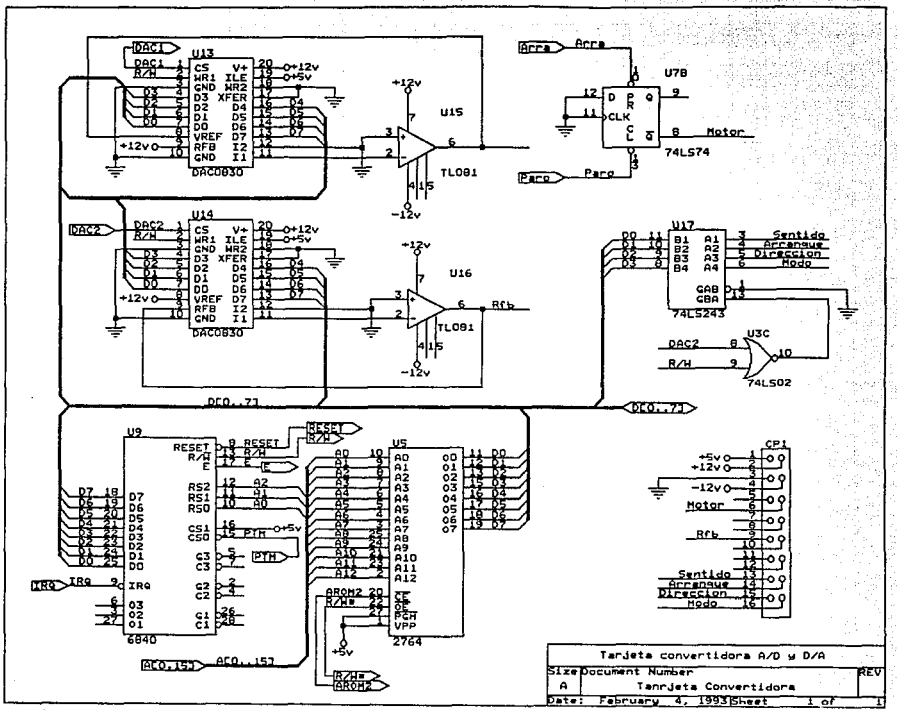

U8D
74LS08
LE C0
HAVDG=
ES C0
U8A
74LS08
U10D
74LS00
ALFA
U12A
74LS32
AD11
M1
AD12
M2
U9B
74LS32
WE
U9D
74LS32
LE ES
U11B
74LS04
OE
R/W=
R/W
U11A
74LS04
U6
P=Q
U7
74LS138
U14A
74LS74
CLK
SYNC=
SYNC
+5v
I05
74LS688
U13A
74LS74
HAVDG
U9C
74LS32
A4
A5
U13B
74LS74
FS=
U8B
74LS08
HASYNC
AC0..15
Decodificacion
Size
Document Number
Tarjeta del VDG
REV
A
Date: October 15, 1992
Sheet 3 of 5

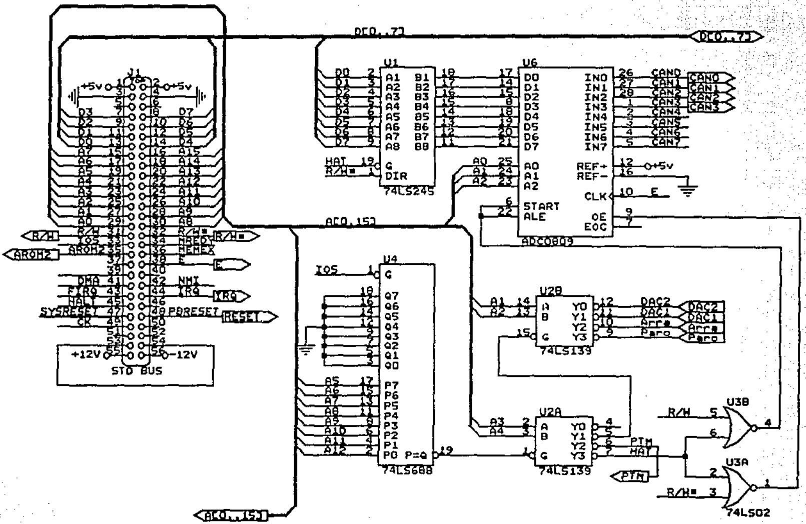


Control y Programacion		
Size: A	Document Number: Tarjeta del VDG	REV
Date: October 15, 1992	Sheet 4 of 5	

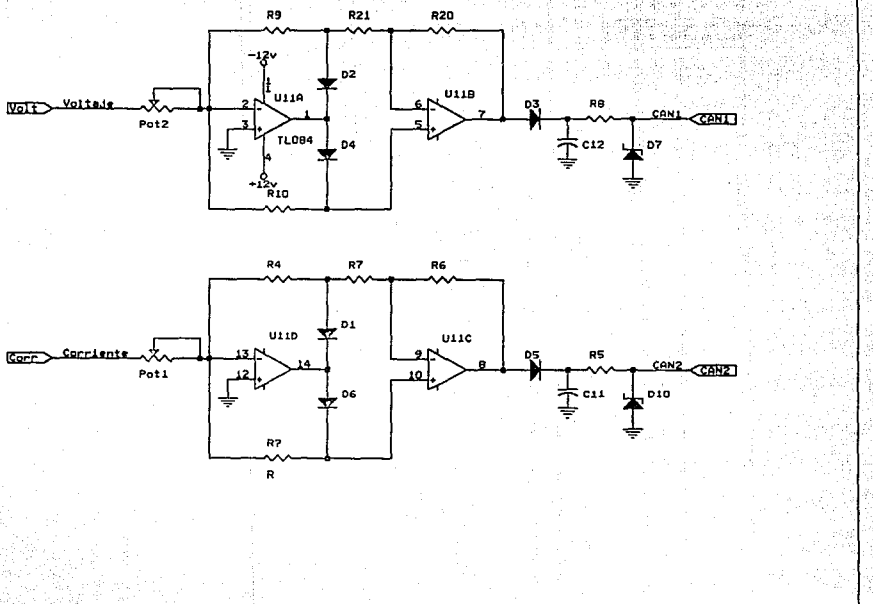

U17
CLK
RF
OSC
DUTY TAN1
PB
CHB
PA
Y
TAN2
CHR
HCI372
XTL1
TRI2
TRI1
L1
C20
C21
C22
C19
C18
R1
R2
R3
R4
R5
R6
+5v
COAX1
U15
6847Y
U16
6164
CKO..7]
CFO..7]
DDFO..7]
ADFO..12]
ADFO..12]
VDG, Memoria de Video y Modulador
Size Document Number
A
Tarjeta del VDG
REV
Date: October 15, 1992 Sheet 5 of 5

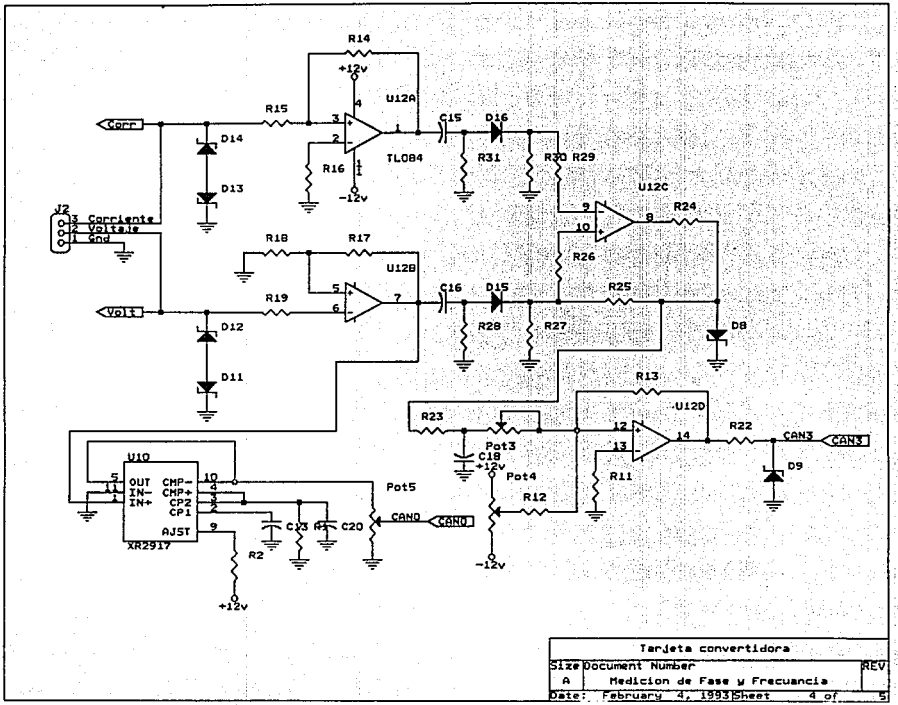

Rectificacion
STD Bus w DEC
PTM EPROM w DAC's
CAN1
CAN2
Corr
Volt
AD0..15
DD0..7
DAC1
DAC2
Arra
Paro
PTM
AROM2
R/W
E
IRQ
RESET
Fase w Frecuencia
CAN0
CAN3
Volt
Corr
Tarjeta Convertidora
Size
A
Document Number
Organizacion de Etapas
REV
Date: February 4, 1993
Sheet 1 of 5

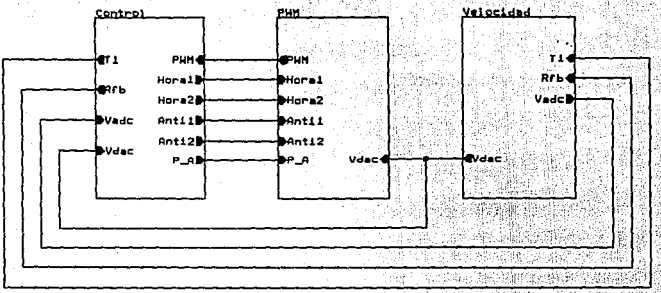

DAC1
U13
DAC0830
+12v
+5v
-12v
U15
TL081
Arra
U7B
74LS74
Motor
Para
DAC2
U14
U16
Rfb
U17
74LS243
Sentido
Arranque
Direccion
Modo
U3C
74LS02
R/W
DB0..7
U9
6840
RESET
E
IRQ
U5
2764
AROM2
CP1
Tarjeta convertidora A/D y D/A
Tarjeta Convertidora
Size
Document Number
REV
Date: February 4, 1993
Sheet 1 of 1



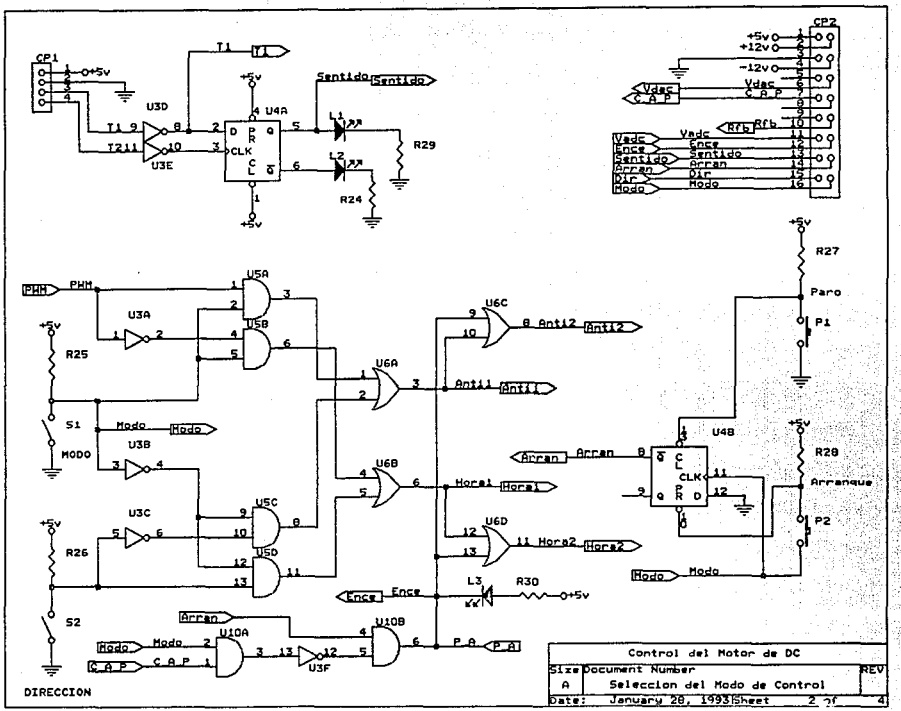
Tarjeta Convertidora		
Size	Document Number	REV
A	STD Bus, Decodificacion y ADC	
Date: February 4, 1993	Sheet 2 of 5	

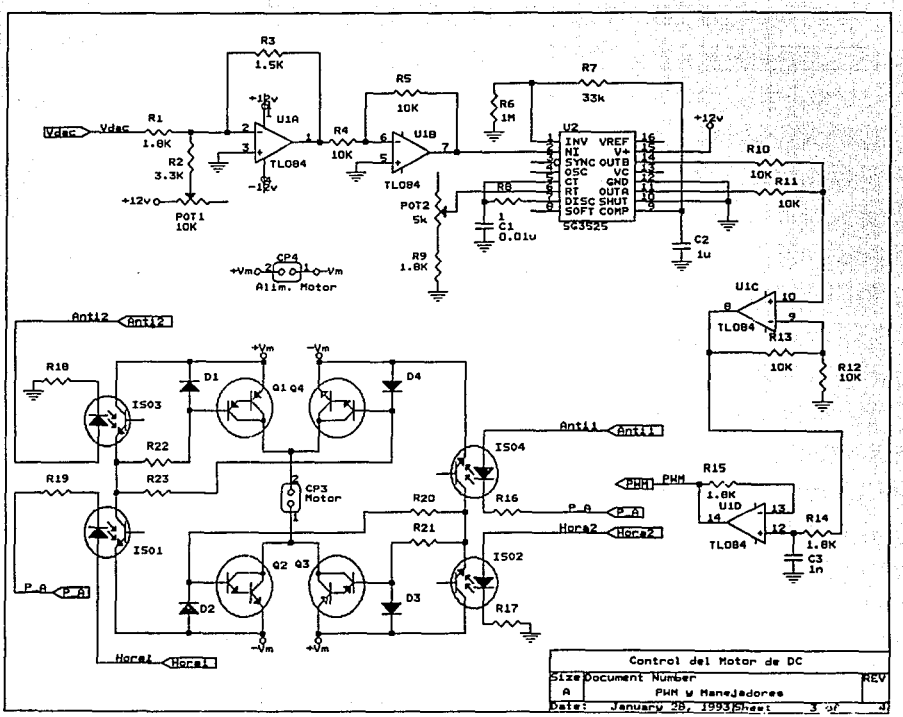

R9
R21
R20
-12v
D2
U11A
U11B
D3
R8
CAN1
CAN1
Volt
Voltaje
Pot2
2
3
1
TL084
4
6
5
7
C12
D7
D4
+12v
R10
R4
R7
R6
U11D
D1
U11C
Corr
Corriente
Pot1
13
12
14
9
10
8
D5
R5
CAN2
CAN2
C11
D10
D6
R7
R
Tarjeta convertidora
Size
Document Number
REV
A
Medicion de Voltaje y Corriente
Date: February 4, 1993
Sheet 3 of 5

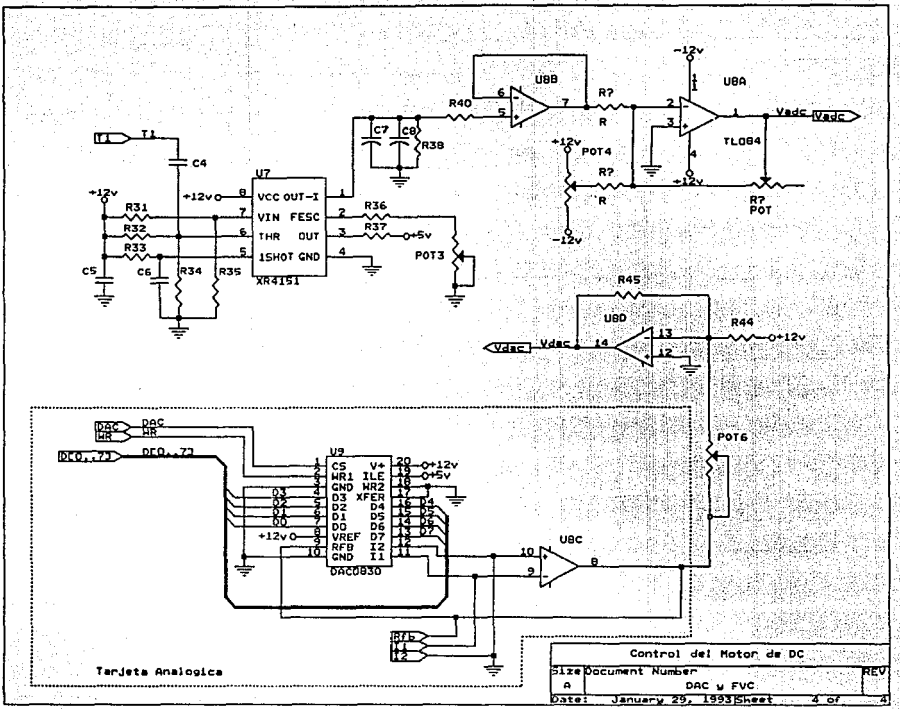

R14
+12v
U12A
R15
Corr
D14
D13
C15
D16
R31
R30
R29
R16
TL084
-12v
U12C
R24
J2
3 Corriente
2 Voltaje
1 Gnd
R18
R17
U12B
R26
R19
Volt
D12
D11
C16
D15
R28
R27
R25
D8
R13
U12D
R23
Pot3
C18
+12v
R22
CAN3
D9
U10
OUT
IN-
IN+
CMP-
CMP+
CP2
CP1
AJST
XR2917
Pot5
Pot4
R11
R12
CAN0
C19
C20
R2
-12v
+12v
Tarjeta convertidora
Size
A
Document Number
Medicion de Fase y Frecuencia
REV
Date: February 4, 1993
Sheet 4 of 5

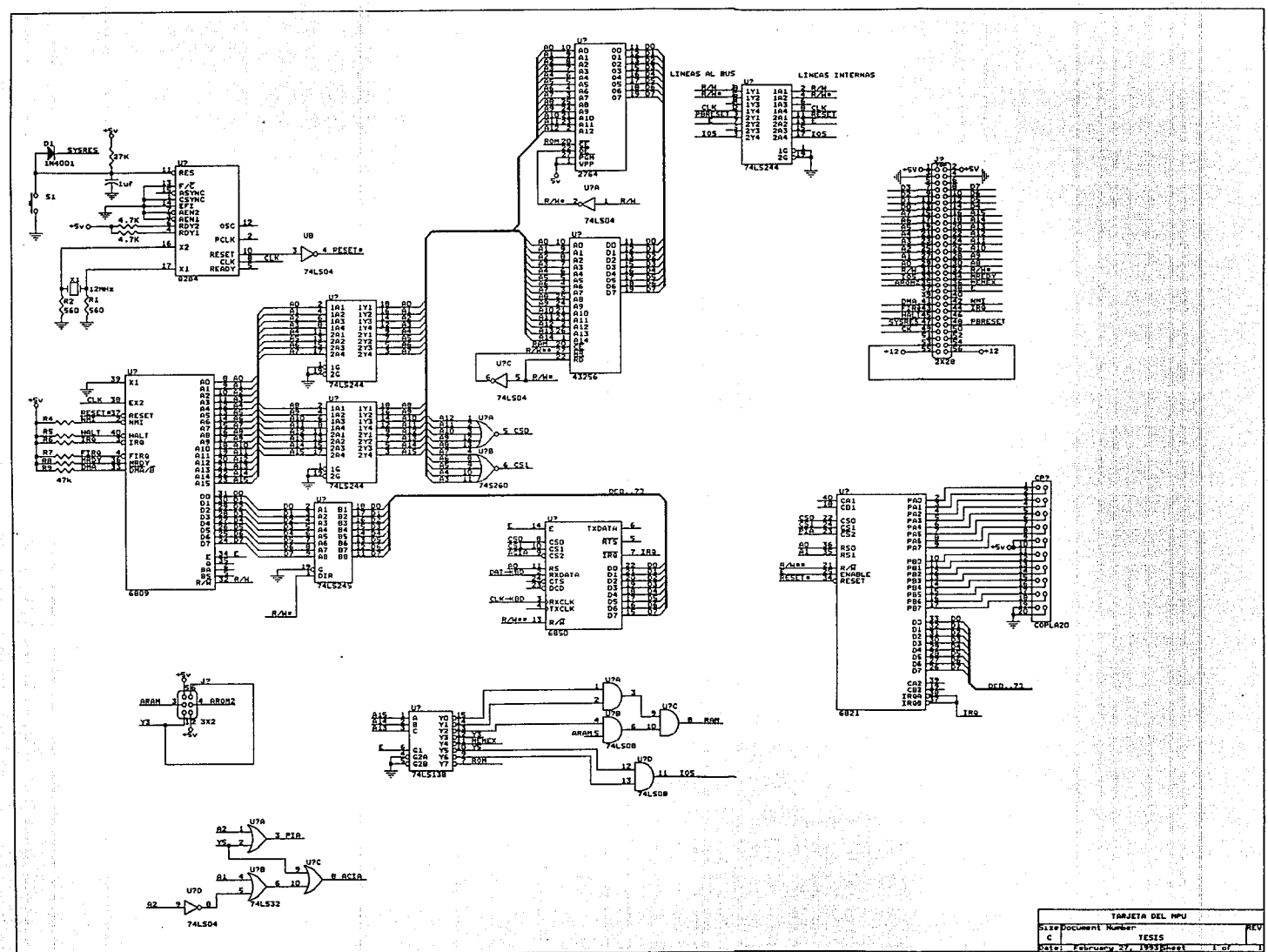


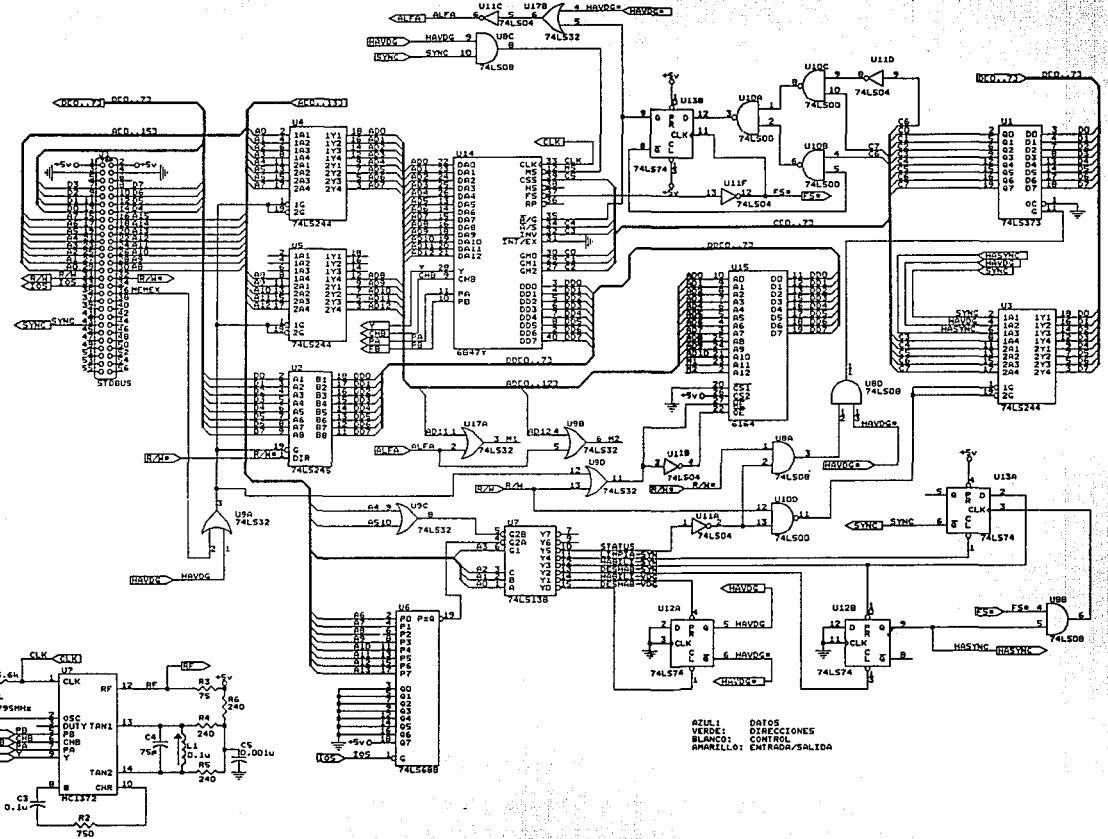
Control del Motor de DC		
Size: A	Document Number: Organizacion del Circuito	REV:
Date: January 28, 1993	Sheet:	of

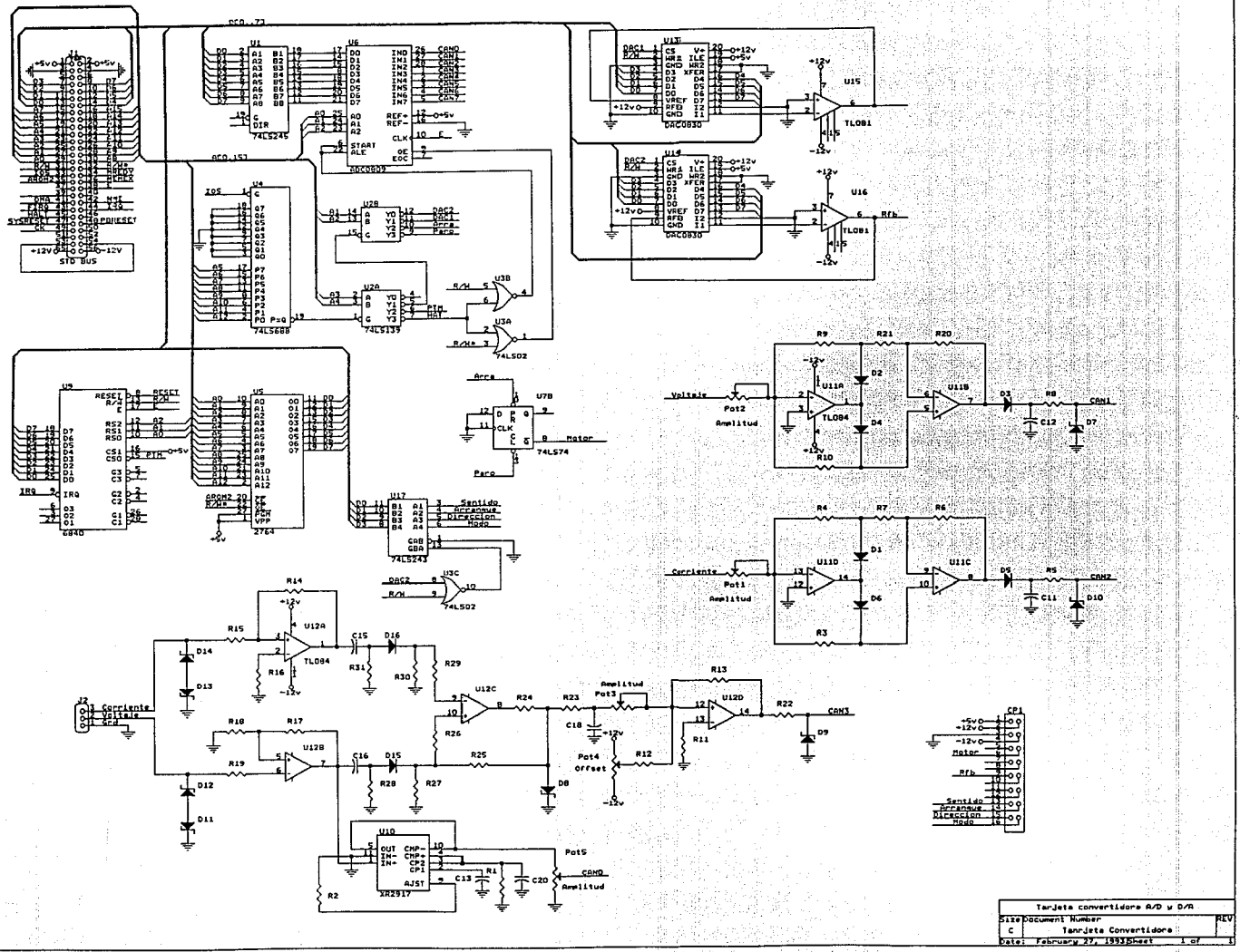

CP1
CP2
U3D
U3E
U4A
U3A
U3B
U3C
U5A
U5B
U5C
U5D
U6A
U6B
U6C
U6D
U4B
U10A
U10B
U3F
R24
R25
R26
R27
R28
R29
R30
L1
L2
L3
S1
S2
P1
P2
PHM
Modo
Sentido
Vdac
Rfb
Ence
Arran
Dir
Paro
Arranque
Anti1
Anti2
Hora1
Hora2
P A
DIRECCION
Control del Motor de DC
Seleccion del Modo de Control
January 28, 1993
Sheet 2 of 4

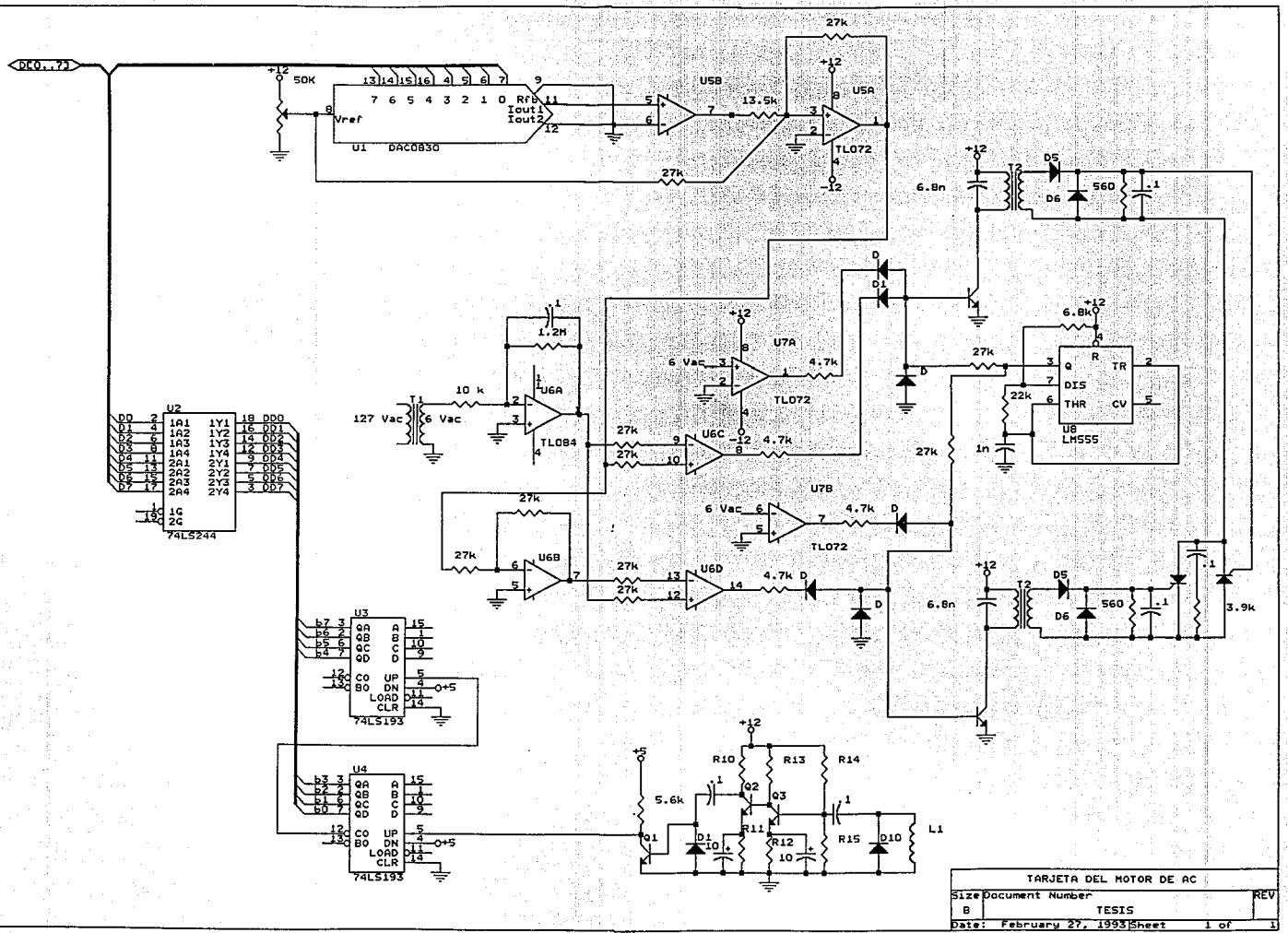

R3
1.5K
R5
10K
R7
33k
R6
1M
+12v
R1
1.8K
R2
3.3K
U1A
TL084
R4
10K
U1B
TL084
U2
INV
NI
SYNC
OSC
CT
RT
DISC
SOFT COMP
VREF
V+
OUTB
VC
GND
OUTA
SHUT
SG3525
R10
10K
R11
10K
R8
POT2
5k
C1
0.01u
C2
1u
POT1
10K
R9
1.8K
CP4
Alim. Motor
+Vm
-Vm
U1C
TL084
R13
10K
R12
10K
Anti2
R18
D1
Q1
Q4
D4
IS03
R22
R19
R23
CP3
Motor
IS01
Anti1
IS04
PWM
R15
1.8K
U1D
TL084
R14
1.8K
C3
1n
R20
R16
P_A
Hora2
R21
Q2
Q3
IS02
D2
D3
R17
Hora1
Control del Motor de DC
Size
A
Document Number
PWM y Manejadores
REV
Date: January 28, 1993
Sheet 3 of 4


U8B
U8A
TL084
-12v
+12v
R40
R38
C7
C8
C4
C5
C6
R31
R32
R33
R34
R35
R36
R37
R45
R44
U7
XR4151
VCC OUT-I
VIN FESC
THR OUT
1SHOT GND
POT3
POT4
POT6
U8D
U8C
U9
DAC0830
CS V+
WR1 ILE
GND WR2
D3 XFER
D2 D4
D1 D5
D0 D6
VREF D7
RFB I2
GND I1
DAC WR
DFO..7
Vdac
Tarjeta Analogica
Control del Motor de DC
Size Document Number REV
A DAC y FVC
Date: January 29, 1993 Sheet 4 of 4


LINEAS AL BUS
LINEAS INTERNAS
SYSRES
1N4001
37K
1uf
S1
4.7K
4.7K
8284
12MHz
560
560
74LS04
2764
74LS244
74LS245
74LS260
43256
6809
6850
6821
74LS32
74LS138
74LS08
2X20
COPLA20
TARJETA DEL MPU
TESIS
February 27, 1993


AZUL: DATOS
VERDE: DIRECCIONES
BLANCO: CONTROL
AMARILLO: ENTRADA/SALIDA
TARJETA DEL VDC
TESIS
February 27, 1993


Tarjeta convertidora A/D y D/A
Size C
Document Number
Tarjeta Convertidora
REV
Date: February 27, 1993
Sheet 1 of 1
STD BUS
74LS245
DAC0809
DAC0830
74LS688
74LS139
74LS02
74LS74
74LS243
6840
2764
TL084
TL081
XR2917
Voltaje
Amplitud
Pot2
Corriente
Pot1
Pot3
Pot4
Offset
Pot5
Sentido
Arranque
Direccion
Modo
Motor
Rfb
CAN1
CAN2
CAN3


DC0..7?
+12
50K
Vref
U1 DAC0830
Rfb
Iout1
Iout2
USB
USA
TL072
13.5k
27k
-12
6.8n
D5
D6
560
.1
D1
1.2M
10 k
U6A
TL084
127 Vac
6 Vac
U7A
4.7k
U6C
U7B
U6B
U6D
6.8k
22k
1n
U8
LM555
R
TR
DIS
THR
CV
3.9k
U2
74LS244
1A1
1A2
1A3
1A4
2A1
2A2
2A3
2A4
1Y1
1Y2
1Y3
1Y4
2Y1
2Y2
2Y3
2Y4
1G
2G
U3
U4
74LS193
QA
QB
QC
QD
CO
BO
UP
DN
LOAD
CLR
+5
R10
R13
R14
5.6k
Q1
Q2
Q3
R11
R12
R15
D10
L1
TARJETA DEL MOTOR DE AC
Size
B
Document Number
TESIS
REV
Date: February 27, 1993
Sheet 1 of 1

APENDICE

C

PROGRAMAS

Apéndice C

C.1 Programa de evaluación.

```
                    *
                    *          Programa de Evaluación
                    *
                    *       sis_eva.asm            920310
                    *
                    *  Programa que desempeña las funciones de
                    *  despliegue, decodificación y control del
                    *  sistema de evaluación.

                        *       inicio del programa
0000                            org  $0000
                        *  inicialización de los apuntadores y las
                        *  pilas (staks)
0000 BE 5F E0           fan     LDX  $5FE0
0003 8C E1 55                   CMPX #$E155
0006 26 10                      BNE  dif
0008 30 1F              loop    LEAX -1,X
```

```
000A 26 FC                      BNE  loop
000C BF 5F E0                   STX  $5FE0
000F 10 FE 5F DA                LDS  $5FDA
0013 8E E1 93                   LDX  #$E193
0016 1E 15                      EXG  X,PC
0018 CE 5F 77           dif     LDU  #$5F77
001B 4F                         CLRA
001C A7 43                      STA  3,U
001E 43                         COMA
001F A7 C4                      STA  ,U
0021 BE 5F EE                   LDX  $5FEE
0024 AF 4A                      STX  10,U
0026 32 C8 63                   LEAS 99,U
0029 17 10 59                   LBSR pr_as
002C 8D 26                      BSR  apun
002E 8D 53              opc     BSR  prom
0030 26 FC                      BNE  opc
0032 AD 84                      JSR  ,X
0034 20 F8                      BRA  opc
                        *
                        *       rutina del despliege del encabezado
                        *
0054                            org  $0054

0054 34 40              apun    PSHS U
0056 33 8C DD                   LEAU -35,PC
0059 17 0F A9                   LBSR alfa
005C 35 C0                      PULS PC,U
                        *
                        *       rutina de despliege del promp
                        *
0083                            org  $0083

0083 34 40              prom    PSHS U
0085 33 8C D6           rep     LEAU -42,PC
0088 17 0F 7A                   LBSR alfa
008B 17 0F EB           vuel    LBSR beta
008E 84 5F                      ANDA #$5F
0090 81 0D                      CMPA #$0D
0092 27 F1                      BEQ  rep
0094 33 8C CD                   LEAU -51,PC
0097 A1 C4              ver     CMPA ,U
0099 33 43                      LEAU 3,U
009B 22 FA                      BHI  ver
009D 26 EC                      BNE  vuel
009F 17 0F BE                   LBSR gama
00A2 86 20                      LDA  #$20
00A4 17 0F B9                   LBSR gama
00A7 5F                         CLRB
00A8 4F                         CLRA
00A9 A3 5E                      SUBD -2,U
00AB 30 8C D5                   LEAX -43,PC
00AE 30 8B                      LEAX D,X
00B0 5F                         CLRB
00B1 4F                         CLRA
00B2 35 C0                      PULS PC,U
```

```
                        *
                        *       rutina de la opción  r
                        *       despliege de los registros
                        *
01A8                            org $01a8

01A8 86 0D              op_r    LDA  #$0D
01AA 17 10 4D                   LBSR vdg
01AD 30 4C                      LEAX 12,U
01AF AF 5E                      STX  -2,U
01B1 30 5E                      LEAX -2,U
01B3 31 8C 1A                   LEAY 26,PC
01B6 16 11 49                   LBRA corta
01B9 86 0D                      LDA  #$0D
01BB 17 10 3C                   LBSR vdg
01BE 86 50                      LDA  #$50
01C0 17 10 37                   LBSR vdg
01C3 AE 84                      LDX  ,X
01C5 C6 3D                      LDB  #$3D
01C7 1C FC                      ANDCC #$FC
01C9 12                         NOP
01CA 12                         NOP

                        *
                        *       Rutina de escritura/lectura
                        *
1005                            org  $1005

1005 34 02              alfa    PSHS A
1007 A6 C0              cont    LDA  ,U+
1009 27 04                      BEQ  lis
100B 8D 53                      BSR  gama
100D 20 F8                      BRA  cont
100F 35 82              lis     PULS PC,A
                        *
                        *       rutina de atención al ACIA
                        *
1060                            org  $1060

1060 34 02              gama    PSHS A
1062 B6 A0 04           polea   LDA  $A004
1065 44                         LSRA
1066 24 08                      BCC  escri
1068 8D 0F              lee     BSR  beta
106A 81 13                      CMPA #$13
106C 27 FA                      BEQ  lee
106E 20 F2                      BRA  polea
1070 35 02              escri   PULS A              ;lee el dato a
1072 16 01 85                   LBRA vdg            ;desplegar
1075 12                 vue1    NOP                 ;conversion de
1076 12                         NOP                 ;ASCII a código
1077 12                         NOP                 ;del VDG
1078 39                         RTS
1079 16 02 08           beta    LBRA tecl
107C 86 02              vue2    LDA  #$02
107E B7 5F F6                   STA  $5FF6
```

```
1081 B6 5F F1                    LDA  $5FF1
1084 39                          RTS
1085 CC FF 14          pr_as     LDD  #$FF14
1088 B7 A0 04                    STA  $A004
108B F7 A0 04                    STB  $A004
108E 39                          RTS
                       *
                       *         rutina de la opcion q
                       *         abandonar el programa
10A0                             org  $10a0

10A0 34 40                       PSHS U
10A2 33 8C 00                    LEAU ,PC
10A5 11 83 F0 A5                 CMPU #$F0A5
10A9 27 12                       BEQ  no_hay
10AB 33 8C 1D                    LEAU 29,PC
10AE 17 FF 54                    LBSR alfa
10B1 17 FF C5                    LBSR beta
10B4 81 53                       CMPA #$53
10B6 26 0B                       BNE  arrep
10B8 8E E1 1E                    LDX  #$E11E
10BB 1E 15                       EXG  X,PC
10BD 33 8C 30          no_hay    LEAU 48,PC
10C0 17 FF 42                    LBSR alfa
10C3 33 8C 46          arrep     LEAU 70,PC
10C6 17 FF 3C                    LBSR alfa
10C9 35 C0                       PULS PC,U
                       *
                       *         rutina de inicializacion del VDG
                       *
1130                             org  $1130

1130 B6 A0 08                    LDA  $A008    ;deshabilita al VDG
1133 10 8E 98 00                 LDY  #$9800   ;ram para el modo
1137 10 BF 9B 00                 STY  $9B00    ;alfa-número,
113B 7F 9B 02                    CLR  $9B02   ;apuntador de renglon
113E 8E 02 00                    LDX  #$200    ;y columna del VDG
1141 86 20                       LDA  #$20
1143 A7 A0             cls       STA  ,Y+
1145 30 82                       LEAX ,-X
1147 26 FA                       BNE  cls
1149 86 00                        LDA  #$00     ;palabra de control
114B B7 A0 0D                    STA  $A00D    ;para el vdg
114E B6 A0 09                    LDA  $A009    ;habilita al vdg
1151 7F A0 03                    CLR  $A003    ;programacion del pia
1154 7F A0 04                    CLR  $A004    ;puerto a como salidas
1157 86 04                       LDA  #$04     ;puerto b como
1159 B7 A0 03                    STA  $A003    ;entradas
115C 7F A0 01                    CLR  $A001
115F 86 FF                       LDA  #$FF
1161 B7 A0 00                    STA  $A000
1164 86 04                       LDA  #$04
1166 B7 A0 01                    STA  $A001
1169 7F 5F F2                    CLR  $5FF2    ;banderas del teclado
116C 7F 5F F3                    CLR  $5FF3
116F 86 02                       LDA  #$02
```

```
1171 B7 5F F6            STA  $5FF6
1174 B6 5F FF            LDA  $5FFF
1177 81 AA               CMPA #$AA
1179 27 08               BEQ  por
117B 86 AA               LDA  #$AA
117D B7 5F FF            STA  $5FFF
1180 16 EE 7D            LBRA fan
1183 7A 5F F6      por   DEC  $5FF6
1186 16 EE 77            LBRA fan
                   *
                   *     rutina de conversion de codigo ascii
                   *      a codigo del vdg
                   *
11FA                     org  $11fa

11FA 34 02         vdg   PSHS A          ;guardamos el valor
11FC B7 A0 08            STA  $A008     ;ASCII, deshabitita al
11FF 10 BF 9B 04         STY  $9B04      ;VDG, guardamos los
1203 F7 9B 03            STB  $9B03      ;registros empleados
1206 10 BE 9B 00         LDY  $9B00      ;carga el renglon
120A F6 9B 02            LDB  $9B02      ;carga la columna
120D 81 0A               CMPA #$0A      ;espacio a la derecha
120F 26 03               BNE  espa
1211 86 20               LDA  #$20
1213 12                  NOP
1214 81 0D         espa  CMPA #$0D      ;comparamos con carry
1216 26 05               BNE  as_vd      ;return
1218 12                  NOP
1219 86 61               LDA  #$61
121B 20 19               BRA  de_vd
121D 81 08          as_vd  CMPA #$08       ;compara con back
121F 26 08               BNE  back       ;space
1221 12                  NOP
1222 5A                  DECB
1223 86 20               LDA  #$20
1225 A7 A5               STA  B,Y
1227 20 18               BRA  erase
1229 12            back  NOP
122A 81 61                CMPA #$61       ;preguntamos si es
122C 2D 02               BLT  may       ;minúscula
122E 80 20               SUBA #$20
1230 81 40          may    CMPA #$40       ;preguntamos si es
1232 2D 02               BLT  de_vd     ;mayúscula
1234 80 40               SUBA #$40
1236 81 61         de_vd  CMPA #$61      ;pregunta por carry
1238 27 19               BEQ  in_re     ;return
123A A7 A5               STA  B,Y       ;almacena en la ram
123C 5C                  INCB           ;incrementa columna
123D C1 20               CMPB #$20      ;son 32 caracteres?
123F 2C 12               BGE  in_re
1241 F7 9B 02       erase  STB  $9B02     ;guarda la columna
1244 F6 9B 03       ter   LDB  $9B03      ;recuperamos
1247 10 BE 9B 04          LDY  $9B04      ;valores
124B B7 A0 09             STA  $A009      ;habilita la vdg
124E 35 02                PULS A          ;recupera registro
1250 16 FE 22             LBRA vue1
```

```
1253 7F 9B 02          in_re  CLR  $9B02     ;inicia columna
1256 31 A8 20                 LEAY $20,Y     ;cambio de renglon
1259 10 8C 9A 00              CMPY #$9A00    ;ya son 16 renglones
125D 2C 06                    BGE  re_pa
125F 10 BF 9B 00       almy   STY  $9B00      ;almacena el nuevo
1263 20 DF                    BRA  ter       ;renglón
1265 10 8E 98 00       re_pa  LDY  #$9800    ;recorre los datos de
1269 E6 A8 20          re_da  LDB  $20,Y      ;pantalla un renglon
126C E7 A0                    STB  ,Y+       ;arriba
126E 10 8C 99 E0              CMPY #$99E0
1272 2D F5                    BLT  re_da
1274 5F                       CLRB
1275 34 02                    PSHS A
1277 86 20                    LDA  #$20
1279 A7 A5             llen   STA  B,Y
127B 5C                       INCB
127C C1 20                    CMPB #$20
127E 2D F9                    BLT  llen
1280 35 02                    PULS A
1282 20 DB                    BRA  almy
1284 B6 A0 04          tecl   LDA  $A004
1287 84 01                    ANDA #$01
1289 27 F9                    BEQ  tecl
128B B6 A0 05                 LDA  $A005
128E 7A 5F F6                 DEC  $5FF6
1291 26 F1                    BNE  tecl
1293 81 75                    CMPA #$75      ;pregunta si la tecla
1295 26 0C                     BNE  conv      ;fue caps lock
1297 86 FF                    LDA  #$FF
1299 B8 5F F3                 EORA $5FF3
129C B7 5F F3                 STA  $5FF3
129F 86 13                    LDA  #$13
12A1 20 59                    BRA  fin
12A3 81 8B             conv   CMPA #$8B
12A5 26 0C                    BNE  sig
12A7 86 FF                    LDA  #$FF
12A9 B8 5F F2                 EORA $5FF2     ;pregunta si la tecla
12AC B7 5F F2                  STA  $5FF2     ;fue num luck
12AF 86 13                    LDA  #$13
12B1 20 49                    BRA  fin
12B3 10 BF 5F F4       sig    STY  $5FF4
12B7 F7 5F F1                 STB  $5FF1
12BA 31 8C 63                 LEAY  $63,PC
12BD 12                       NOP
12BE F6 5F F2                 LDB  $5FF2      ;pregunta por la
12C1 27 0A                    BEQ  nor1       ;bandera de num luck
12C3 81 8F                    CMPA #$8F
12C5 2D 06                    BLT  nor1
12C7 81 A7                    CMPA #$A7
12C9 2E 02                    BGT  nor1
12CB 8B 3C                    ADDA #$3C
12CD F6 5F F3          nor1   LDB  $5FF3      ;pregunta por la
12D0 27 1E                    BEQ  nor2      ;bandera de caps lock
12D2 81 05                    CMPA #$05
12D4 2D 1A                    BLT  nor2
12D6 81 1B                    CMPA #$1B
```

```
12D8 2E 02                    BGT  con1
12DA 8B A4                    ADDA #$A4
12DC 81 4F              con1  CMPA #$4F
12DE 2D 10                    BLT  nor2
12E0 81 51                    CMPA #$51
12E2 2E 02                    BGT  con2
12E4 8B 72                    ADDA #$72
12E6 81 67              con2  CMPA #$67
12E8 2D 06                    BLT  nor2
12EA 81 6B                    CMPA #$6B
12EC 2E 02                    BGT  nor2
12EE 8B 5E                    ADDA #$5E
12F0 80 03              nor2  SUBA #$03
12F2 44                       LSRA
12F3 A6 A6                    LDA  A,Y
12F5 10 BE 5F F4              LDY  $5FF4
12F9 F6 5F F1                 LDB  $5FF1
12FC B7 5F F1           fin   STA  $5FF1
12FF 16 FD 7A                 LBRA vue2
1302 17 EE EA           corta LBSR $01EF
1305 17 EE D6                 LBSR $01DE
1308 17 EE EA                 LBSR $01F5
130B 17 EE E7                 LBSR $01F5
130E 17 EE E4                 LBSR $01F5
1311 86 0D                    LDA  #$0D
1313 17 FE E4                 LBSR vdg
1316 17 EE D6                 LBSR $01EF
1319 17 EE D3                 LBSR $01EF
131C 17 EE D0                 LBSR $01EF
131F 16 EE 97                 LBRA op_r
```

C.2 Programa de control

```
                        *
                        *     Programa  Principal  de  Control
                        *     sis_con.asm            920509
                        *
                        *      Auto prueba de inicio del sistema
                        *
0000                            org  $0000

0000 10 CE 5F BF                  lds  #$5fbf     ;inicializamos el
0004 CE 5F DF                     ldu  #$5fdf     ;el stack
0007 7F A0 01                     clr  $a001      ;inicializacion del
000A 86 FF                        lda  #$ff       ;puerto A del PIA
000C B7 A0 00                     sta  $a000      ;como salidas
000F 86 04                        lda  #$04
0011 B7 A0 01                     sta  $a001
0014 CC FF 14                     ldd  #$ff14     ;inicializacion del
0017 B7 A0 04                     sta  $a004      ;ACIA para el
001A F7 A0 04                     stb  $a004      ;teclado
001D B6 5F FF                     lda  $5fff      ;se verifica el
0020 81 AA                        cmpa #$aa       ;funcionamiento del
0022 27 1D                        beq  bat        ;teclado
0024 81 FC                        cmpa #$fc
0026 27 19                        beq  bat
0028 8E FF FF                     ldx  #$ffff
002B B6 A0 04               tecla  lda  $a004      ;para el caso del
002E 84 01                        anda #$01       ;POR se espera la
0030 26 06                       bne  codi       ;finalización del bat
0032 30 82                        leax ,-x        ;del teclado
0034 26 F5                        bne  tecla
0036 20 07                        bra  malf
0038 B6 A0 05               codi   lda  $a005      ;el teclado funciona
003B 81 AA                        cmpa #$aa
003D 27 02                        beq  bat
003F 86 FC                  malf   lda  #$fc       ;el teclado o el ACIA
0041 B7 5F FF                bat   sta  $5fff      ;estan mal o no hay
0044 B7 A0 08                      sta  $a008
0047 7F 5F FE                     clr  $5ffe      ;banderas para la rom
004A 7F 5F FD                     clr  $5ffd      ;adicional y ram del
004D 8E 00 00                     ldx  #$0000     ;VDG
0050 8C A0 00               ram    cmpx #$a000     ;se inicia la prueba
0053 27 46                        beq  tram       ;le RAM
0055 E6 84                        ldb  ,x
0057 86 AA                        lda  #$aa
0059 A7 84                        sta  ,x
005B A1 84                        cmpa ,x
005D 26 0B                        bne  fram
005F 43                           coma
0060 A7 84                        sta  ,x
0062 A1 84                        cmpa ,x
0064 26 04                        bne  fram
0066 E7 80                        stb  ,x+
0068 20 E6                        bra  ram
006A E7 84                  fram   stb  ,x         ;se encontro el final
006C 8C 60 00                     cmpx #$6000     ;de un bloque de ram
```

```
006F 27 07                beq  vrom      ;verificamos si hay
0071 8C 80 00             cmpx #$8000    ;rom y ram de video
0074 27 20                beq  rvdg
0076 20 23                bra  tram
0078 10 8E 60 00    vrom  ldy  #$6000    ;se verifica si
007C E6 A0                ldb  ,y+       ;hay una rom
007E C1 BE                cmpb #$be
0080 26 10                bne  norom
0082 E6 A0                ldb  ,y+
0084 C1 5F                cmpb #$5f
0086 26 0A                bne  norom
0088 86 AA                lda  #$aa      ;encuentra el
008A B7 5F FE       nrom  sta  $5ffe     ;programa de
008D 8E 80 00             ldx  #$8000    ;evaluación
0090 20 BE                bra  ram
0092 86 FC          norom lda  #$fc      ;no hay rom o no
0094 20 F4                bra  nrom      ;es el esperado
0096 86 FC          rvdg  lda  #$fc
0098 B7 5F FD             sta  $5ffd     ;no hay ram de video
009B BF 5F FB       tram  stx  $5ffb
009E B6 5F FD             lda  $5ffd
00A1 5F                   clrb
00A2 81 00                cmpa #$00
00A4 26 5F                bne  halt
00A6 10 8E 80 00    desp  ldy  #$8000    ;prueba de video
00AA 8E 1F FF             ldx  #$1fff    ;despliege de varios
00AD 86 20                lda  #$20      ;modos de operacion
00AF A7 A0          panta sta  ,y+
00B1 30 82                leax ,-x
00B3 26 FA                bne  panta
00B5 30 8C 11             leax $11,pc
00B8 A6 80          pvdg  lda  ,x+
00BA B7 A0 08             sta  $a008
00BD B7 A0 0D             sta  $a00d
00C0 B7 A0 09             sta  $a009
00C3 81 00                cmpa #$00
00C5 26 F1                bne  pvdg
00C7 20 16                bra  desple


00DF                      org $00df

00DF 7F 5F F2       desple clr  $5ff2    ;inicializamos las
00E2 7F 5F F3              clr  $5ff3    ;banderas del teclado
00E5 7F 5F F6              clr  $5ff6
00E8 7F A0 03                clr  $a003     ;programacion del
00EB 7F A0 04              clr  $a004    ;puerto b del pia
00EE 86 04                 lda  #$04     ;como entradas
00F0 B7 A0 03              sta  $a003
00F3 5F                    clrb
00F4 86 01                 lda  #$01
00F6 B7 5F F6              sta  $5ff6
00F9 17 00 C4              lbsr repo
00FC B6 5F FF              lda  $5fff
00FF 81 FC                 cmpa #$fc
0101 26 09                 bne  ready
```

```
0103 C6 01                      ldb  #$01
0105 CA 80              halt    orb  #$80      ;sistema detenido por
0107 F7 A0 00                   stb  $a000     ;falla
010A 3C FF                      cwai #$ff
                        *
                        * despliege del menu principal
                        *
010C CC 00 04           ready   ldd  #$04
010F 10 8E 9A 00                ldy  #$9a00
0113 B7 A0 08                   sta  $a008
0116 A7 A0              limp    sta  ,y+       ;limpia las
0118 5A                         decb           ;localidades de los
0119 26 FB                      bne  limp      ;datos de control
011B B7 A0 09                   sta  $a009
011E 17 0F 0D           inicio  lbsr provdg    ;programacion del vdg
0121 8E 00 00                   ldx  #$00      ;en el sistema de
0124 10 8E 9A 00                ldy  #$9a00    ;control
0128 C6 04                      ldb  #$04
012A B7 A0 08                   sta  $a008
012D 16 05 E0                   lbra trasp     ;se recuperan los
0130 12                         nop            ;datos de control
0131 12                         nop
0132 12                         nop
0133 12                         nop
0134 B7 A0 09           retor   sta  $a009
0137 B6 5F FE           reini   lda  $5ffe
013A 81 00                      cmpa #$00
013C 27 1F                      beq  menco     ;hay programa?
013E 30 8C 1F                   leax $1f,pc
0141 10 8E 98 46                ldy  #$9846
0145 17 01 78                   lbsr escr      ;escribe el
0148 31 A8 48                   leay $48,y     ;encabezado
014B 17 01 72                   lbsr escr      ;escribe la primera
014E 31 A8 49                   leay $49,y     ;opción
0151 17 01 6C                   lbsr escr      ;escribe la segunda
0154 31 A8 47                   leay $47,y     ;opción
0157 17 01 66                   lbsr escr      ;escribe opcion
015A 16 02 23                   lbra menu
015D 16 02 63           menco   lbra fant
                        *
                        *    rutina del despliege de la autoprueba
                        *
01C0                            org $01c0

01C0 30 8C 73           repo    leax $73,pc
01C3 10 8E 98 44                ldy  #$9844
01C7 17 00 F6                   lbsr escr      ;escribe el
01CA 31 A8 46                   leay $46,y     ;encabezado
01CD 17 00 F0                   lbsr escr      ;escribe ram
01D0 FC 5F FB                   ldd  $5ffb     ;disponible
01D3 17 01 94                   lbsr ajust
01D6 17 00 FC                   lbsr he_de     ;el total de
01D9 C6 05                      ldb  #$05      ;direcciones, se
01DB 37 02              digit   pulu a         ;transforma a decimal
01DD 8B 30                      adda #$30
01DF 17 01 7F                   lbsr esca
```

```
01E2 5A                 blan    decb
01E3 26 F6                      bne  digit
01E5 17 00 D8                   lbsr escr       ;escribe bytes
01E8 31 24                      leay $4,y
01EA B6 5F FE                   lda  $5ffe
01ED 81 00                      cmpa #$00
01EF 27 0E                      beq  omit
01F1 17 00 CC                   lbsr escr       ;esribe programa
01F4 81 FC                      cmpa #$fc       ;auxiliar
01F6 27 02                      beq  desc
01F8 20 02                      bra  pros
01FA 30 0B              desc    leax $0b,x
01FC 17 00 C1           pros    lbsr escr
01FF 30 8C 46           omit    leax $46,pc
0202 30 88 46                   leax $46,x
0205 B6 5F FF                   lda  $5fff
0208 10 8E 99 81                ldy  #$9981
020C 81 AA                      cmpa #$aa
020E 27 08                      beq  paus
0210 30 88 14                   leax $14,x
0213 17 00 AA                    lbsr escr      ;escribe falla de
0216 5F                         clrb            ;teclado
0217 39                         rts
0218 17 00 A5            paus    lbsr escr      ;escribe presione
021B C6 05                      ldb  #$05       ;una tecla
021D 8E FF FF           loop    ldx  #$ffff
0220 B6 A0 04           rupt    lda  $a004
0223 84 01                      anda #$01
0225 26 08                      bne  resp
0227 30 82                      leax ,-x
0229 26 F5                      bne  rupt
022B 5A                         decb
022C 26 EF                      bne  loop
022E 39                         rts
022F B6 A0 05           resp    lda  $a005
0232 7C 5F F6                   inc  $5ff6
0235 39                         rts
                        *
                        *       subrutina de escritura a pantalla
                        *
02C0                         org $02c0

02C0 34 02              escr    pshs a
02C2 B7 A0 08                   sta  $a008      ;deshabilita al vdg
02C5 A6 80               es1     lda  ,x+        ;lee el dato de la
02C7 81 FF                     cmpa #$ff       ;la tabla verifica si
02C9 27 04                      beq  pun        ;es el ultimo
02CB A7 A0                      sta  ,y+        ;carácter
02CD 20 F6                      bra  es1
02CF B7 A0 09           pun     sta  $a009      ;habilita al vdg
02D2 35 02                      puls a
02D4 39                         rts
                        *
                        *     subrutina de conversion de hex a dec
                        *
02D5 34 30              he_de   pshs x,y
```

```
02D7 1E 02                    exg  d,y
02D9 8E 00 05                 ldx  #$05
02DC CC 00 0A           nudig ldd  #$0a
02DF 8D 0A                    bsr  div16
02E1 1E 02                    exg  d,y
02E3 36 04                    pshu b
02E5 30 82                    leax ,-x
02E7 26 F3                    bne  nudig
02E9 35 B0                    puls x,y,pc
                        *
                        *     subrutina de division de 16/16 bits
                        *
02EB 34 10              div16 pshs x          ;d=divisor     d=res
02ED 10 BF 5F F7              sty  $5ff7      ;y=dividendo   y=cos
02F1 10 8E 00 00              ldy  #$00
02F5 10 B3 5F F7              cmpd $5ff7      ;y/d
02F9 25 10                    blo  real       ;verifica si d>y
02FB 27 05                    beq  igual      ;verifica si d=y
02FD FC 5F F7                 ldd  $5ff7      ;d>y
0300 35 90                    puls x,pc
0302 10 8E 00 01        igual ldy  #$01       ;d=y
0306 CC 00 00                 ldd  #$00
0309 35 90                    puls x,pc
030B 8E 00 0F           real  ldx  #$0f
030E 8D 4A              shift bsr  corri      ;ajustamos el divisor
0310 1E 02                    exg  d,y
0312 30 82                    leax ,-x
0314 26 F8                    bne  shift
0316 FD 5F F9                 std  $5ff9
0319 FC 5F F7                 ldd  $5ff7
031C 10 BF 5F F7              sty  $5ff7
0320 8E 00 10                 ldx  #$10
0323 10 8E 00 00              ldy  #$00
0327 1E 02                    exg  d,y
0329 12                 div   nop             ;restas y

032A 10 B3 5F F7              cmpd $5ff7      ;corrimientos
032E 2D 24                    blt  otra
0330 1E 02                    exg  d,y
0332 16 04 4B                 lbra par1
0335 1E 02              par2  exg  d,y
0337 F2 5F F8                 sbcb $5ff8
033A B2 5F F7                 sbca $5ff7
033D 1E 02                    exg  d,y
033F 8D 19                    bsr  corri
0341 1E 02                    exg  d,y
0343 C3 00 01                 addd #$01
0346 1E 02                    exg  d,y
0348 C9 00                    adcb #$00
034A 89 00                    adca #$00
034C 30 82              decre leax ,-x
034E 26 D9                    bne  div
0350 1E 02                    exg  d,y
0352 35 90                    puls x,pc
0354 1E 02              otra  exg  d,y
0356 8D 02              par3  bsr  corri
```

```
0358 20 F2                      bra  decre
                                *
                                *     subrutina de corrimientos de 32 bits
                                *
035A 58                         corri  lslb
035B 49                                rola
035C 1E 02                             exg  d,y
035E 59                                rolb
035F 49                                rola
0360 39                                rts
                                *
                                *     subrutina de escritura de caracter
                                *
0361 B7 A0 08                   esca   sta  $a008
0364 A7 A0                             sta  ,y+
0366 B7 A0 09                          sta  $a009
0369 39                                rts
                                *
                                *     subrutina de total de ram
                                *
036A 10 83 A0 00                ajust  cmpd #$a000
036E 26 0F                             bne  fin
0370 B6 5F FE                          lda  $5ffe
0373 81 00                             cmpa #$00
0375 27 05                             beq  total
0377 CC 7F FF                          ldd  #$7fff
037A 20 03                             bra  fin
037C CC 9F FF                   total  ldd  #$9fff
037F 39                         fin    rts
                                *
                                *     lectura de la opcion del menu
                                *          principal
                                *
0380 5F                         menu   clrb
0381 17 0C 9E                          lbsr beta
0384 81 31                             cmpa #$31      ;programa de control
0386 27 06                             beq  vali
0388 81 32                             cmpa #$32      ;prog de evaluacion
038A 26 F4                             bne  menu
038C C6 FF                             ldb  #$ff
038E 8D D1                      vali   bsr  esca
0390 17 0C 8F                          lbsr beta
0393 81 0D                             cmpa #$0d
0395 27 0A                             beq  ejec
0397 31 A2                             leay ,-y
0399 86 20                             lda  #$20
039B 8D C4                             bsr  esca
039D 31 A2                             leay ,-y
039F 20 DF                             bra  menu
03A1 C1 00                      ejec   cmpb #$00
03A3 27 1E                             beq  fant
                                *
                                *     preparacion para ejecutar el
                                *     programa de evaluacion
                                *
03A5 8E 00 00                          ldx  #$00
```

```
03A8 10 8E 9A 00          ldy  #$9a00
03AC C6 04                ldb  #$04
03AE B7 A0 08             sta  $a008
03B1 A6 80          restau lda  ,x+
03B3 A7 A0                sta  ,y+
03B5 5A                   decb
03B6 26 F9                bne  restau
03B8 B7 A0 09             sta  $a009
03BB 12                   nop
03BC 12                   nop
03BD 12                   nop
03BE 8E 71 30             ldx  #$7130
03C1 1E 15                exg  x,pc
                    *
                    *     menu del programa de control
                    *
03C3 8D 66          fant  bsr  cls
03C5 10 8E 98 45          ldy  #$9845
03C9 30 8C 78             leax $78,pc
03CC 17 FE F1             lbsr escr      ;escribe encabezado
03CF 31 A8 47             leay $47,y
03D2 17 FE EB             lbsr escr      ;escribe primera
03D5 31 A8 24             leay $24,y     ;opción
03D8 17 FE E5             lbsr escr      ;escribe segunda
03DB 31 A8 2D             leay $2d,y     ;opción
03DE 17 FE DF             lbsr escr      ;escribe tercera
03E1 31 A8 4B             leay $4b,y     ;opción
03E4 17 FE D9             lbsr escr      ;pide opcion
03E7 5F             leo   clrb
03E8 17 0C 37             lbsr beta
03EB 81 1B                cmpa #$1b
03ED 27 37                beq  reto
03EF 81 31                cmpa #$31
03F1 27 12                beq  eje
03F3 5C                   incb
03F4 5C                   incb
03F5 81 32                cmpa #$32
03F7 27 0C                beq  eje
03F9 5C                   incb
03FA 5C                   incb
03FB 81 33                cmpa #$33
03FD 27 06                beq  eje
03FF 12                   nop
0400 12                   nop
0401 12                   nop
0402 12                   nop
0403 20 E2                bra  leo
0405 17 FF 59       eje   lbsr esca
0408 17 0C 17             lbsr beta
040B 81 0D                cmpa #$0d
040D 27 0B                beq  ruta
040F 31 A2                leay ,-y
0411 86 20                lda  #$20
0413 17 FF 4B             lbsr esca
0416 31 A2                leay ,-y
0418 20 CD                bra  leo
```

```
041A 10 AE 85          ruta   ldy  b,x
041D 12                       nop              ;opción  direccion
041E 12                       nop              ;   1      04c0
041F 12                       nop              ;   2
0420 12                       nop              ;   3
0421 12                       nop
0422 AD A4                    jsr  ,y
0424 20 9D                    bra  fant
0426 8D 03             reto   bsr  cls
0428 16 FD 0C                 lbra reini
                       *
                       *      subrutina que limpia la pantalla
                       *
042B 34 32             cls    pshs a,x,y
042D 86 20                    lda  #$20
042F 8E 02 00                 ldx  #$200
0432 10 8E 98 00              ldy  #$9800
0436 B7 A0 08                 sta  $a008
0439 A7 A0             clsalf sta  ,y+
043B 30 82                    leax ,-x
043D 26 FA                    bne  clsalf
043F B7 A0 09                 sta  $a009
0442 35 B2                    puls a,x,y,pc

06E0                          org $06e0

06E0 17 FB F2                 lbsr he_de
06E3 C6 05                    ldb  #$05
06E5 37 02             cero   pulu a
06E7 81 00                    cmpa #$00
06E9 26 0C                    bne  vali
06EB 8B 20                    adda #$20
06ED 17 FC 71                 lbsr esca
06F0 5A                       decb
06F1 C1 01                    cmpb #$01
06F3 26 F0                    bne  cero
06F5 37 02             d1     pulu a
06F7 8B 30             vali   adda #$30
06F9 17 FC 65                 lbsr esca
06FC 5A                       decb
06FD 26 F6                    bne  d1
06FF 17 FB BE                 lbsr escr
0702 39                       rts

0703 86 5A                    lda  #$5a
0705 B7 40 19                 sta  $4019
0708 CC 03 52                 ldd  #$0352
070B FD 40 08                 std  $4008
070E 1A 10                    orcc #$10
0710 CC FF FF                 ldd  #$ffff
0713 FD A0 16                 std  $a016
0716 CC C3 01                 ldd  #$c301
0719 FD A0 10                 std  $a010
071C 5A                       decb
071D FD A0 10                 std  $a010
                       *
```

```
                        *       subrutina de restauracion e
                        *         iniciaizacion de stack
                        *
0710                            org  $0710

0710 A6 A0              trasp   lda  ,y+
0712 A7 80                      sta  ,x+
0714 5A                         decb
0715 26 F9                      bne  trasp
0717 10 CE 5F BF                lds  #$5fbf
071B CE 5F DF                   ldu  #$5fdf
071E 16 FA 13                   lbra retor

                        *
                        *       modifica la division
                        *
0780                            org  $0780

0780 26 08              par1    bne  resta
0782 10 B3 5F F9                cmpd $5ff9
0786 10 25 FB CC                lblo par3
078A B3 5F F9           resta   subd $5ff9
078D 16 FB A5                   lbra par2
                        *
                        *       subrutina de escritura de textos
                        *
1000                            org  $1000

1000 34 02              alfa    PSHS A
1002 A6 C0              cont    LDA  ,U+
1004 27 04                      BEQ  lis
1006 8D 04                      BSR  gama
1008 20 F8                      BRA  cont
100A 35 82              lis     PULS PC,A
                        *
                        *       rutina de atencion al acia
                        *
100C 34 02              gama    PSHS A
100E B6 A0 04           polea   LDA  $A004
1011 44                         LSRA
1012 24 08                      BCC  escri
1014 8D 0C              lee     BSR  beta
1016 81 13                      CMPA #$13
1018 27 FA                      BEQ  lee
101A 20 F2                      BRA  polea
101C 35 02              escri   PULS A          ;lee el dato a
101E 17 00 32                 LBSR vdg        ;desplegar conversion
1021 39                         RTS             ;ascii a vdg
1022 17 00 A0           beta    LBSR tecl       ;lectura de tecla
1025 86 02                      LDA  #$02
1027 B7 5F F6                   STA  $5FF6
102A B6 5F F1                   LDA  $5FF1
102D 39                         RTS
                        *
                        *       rutina de inicializacion del vdg
                        *
```

```
102E 34 32          provdg pshs a,x,y
1030 B6 A0 08              LDA  $A008     ;deshabilita al vdg
1033 10 8E 98 00           LDY  #$9800    ;ram para modo
1037 10 BF 9B 00           STY  $9B00     ;alfa-numerico
103B 7F 9B 02             CLR  $9B02    ;apuntador de renglon
103E 86 20                 lda  #$20      ;y columna del vdg
1040 8E 02 00              ldx  #$200
1043 A7 A0          cl     sta  ,y+
1045 30 82                 leax ,-x
1047 26 FA                 bne  cl
1049 86 00                 LDA  #$00      ;palabra de control
104B B7 A0 0D              STA  $A00D     ;para el vdg
104E B6 A0 09              LDA  $A009     ;habilita al vdg
1051 35 B2                 puls a,x,y,pc
                    *
                    *      empieza la rutina de conversion de
                    *      codigo ascii  a codigo del vdg
                    *
1053 34 02          vdg    PSHS A         ;guardamos el valor
1055 B7 A0 08              STA  $A008     ;ASCII, deshabilita
1058 10 BF 9B 04           STY  $9B04     ;al VDG, guardamos

105C F7 9B 03              STB  $9B03     ;registros empleados
105F 10 BE 9B 00           LDY  $9B00     ;carga el valor del
1063 F6 9B 02             LDB  $9B02    ;renglon y la columna
1066 81 0D                CMPA #$0D     ;comparamos con carry
1068 26 04                 BNE  as_vd     ;return
106A 86 61                 LDA  #$61
106C 20 0B                 BRA  de_vd
106E 81 08           as_vd   CMPA #$08      ;compara con back
1070 26 07                 BNE  de_vd     ;space
1072 5A                    DECB
1073 86 20                 LDA  #$20
1075 A7 A5                 STA  B,Y
1077 20 0B                 BRA  erase
1079 81 61          de_vd  CMPA #$61      ;pregunta por carry
107B 27 17                 BEQ  in_re     ;return
107D A7 A5                 STA  B,Y       ;almacena en la ram
107F 5C                    INCB           ;incrementa columna
1080 C1 20                 CMPB #$20      ;son 32 caracteres?
1082 2C 10                 BGE  in_re
1084 F7 9B 02       erase  STB  $9B02      ;guarda la columna
1087 F6 9B 03       ter    LDB  $9B03     ;recuperamos valores
108A 10 BE 9B 04           LDY  $9B04
108E B7 A0 09              STA  $A009     ;habilita la vdg
1091 35 02                 PULS A         ;recupera registro
1093 39                    rts
1094 7F 9B 02       in_re  CLR  $9B02     ;inicia columna
1097 31 A8 20              LEAY $20,Y     ;cambio de renglon
109A 10 8C 9A 00           CMPY #$9A00    ;ya son 16 renglones
109E 2C 06                 BGE  re_pa
10A0 10 BF 9B 00    almy   STY  $9B00     ;almacena el nuevo
10A4 20 E1                 BRA  ter       ;renglón
10A6 10 8E 98 00    re_pa  LDY  #$9800    ;recorre los datos de
10AA E6 A8 20       re_da  LDB  $20,Y     ;la pantalla un
10AD E7 A0                 STB  ,Y+       ;renglón arriba
```

```
10AF 10 8C 99 E0          CMPY #$99E0
10B3 2D F5                BLT  re_da
10B5 5F                   CLRB
10B6 34 02                PSHS A
10B8 86 20                LDA  #$20
10BA A7 A5          llen  STA  B,Y
10BC 5C                   INCB
10BD C1 20                CMPB #$20
10BF 2D F9                BLT  llen
10C1 35 02                PULS A
10C3 20 DB                BRA  almy
                    *
                    *     subrutina de atencion al teclado
                    *
10C5 B6 A0 04       tecl  LDA  $A004
10C8 84 01                ANDA #$01
10CA 27 F9                BEQ  tecl
10CC B6 A0 05             LDA  $A005
10CF 7A 5F F6             DEC  $5FF6
10D2 26 F1                BNE  tecl
10D4 81 75                CMPA #$75      ;pregunta si la
10D6 26 0C                BNE  conv      ;tecla fue caps
10D8 86 FF                LDA  #$FF      ;locks
10DA B8 5F F3             EORA $5FF3
10DD B7 5F F3             STA  $5FF3
10E0 86 13                LDA  #$13
10E2 20 58                BRA  final
10E4 81 8B          conv  CMPA #$8B
10E6 26 0C                BNE  sig
10E8 86 FF                LDA  #$FF
10EA B8 5F F2             EORA $5FF2     ;pregunta si la
10ED B7 5F F2             STA  $5FF2     ;tecla fue num luck
10F0 86 13                LDA  #$13
10F2 20 48                BRA  final
10F4 10 BF 5F F4    sig   STY  $5FF4
10F8 F7 5F F1             STB  $5FF1
10FB 31 8C 42             LEAY $42,PC
10FE F6 5F F2             LDB  $5FF2     ;pregunta por la
1101 27 0A                BEQ  nor1      ;bandera de num luck
1103 81 8F                CMPA #$8F
1105 2D 06                BLT  nor1
1107 81 A7                CMPA #$A7
1109 2E 02                BGT  nor1
110B 8B 3C                ADDA #$3C
110D F6 5F F3       nor1  LDB  $5FF3     ;pregunta por la
1110 27 1E                BEQ  nor2      ;bandera de caps lock
1112 81 05                CMPA #$05
1114 2D 1A                BLT  nor2
1116 81 1B                CMPA #$1B
1118 2E 02                BGT  con1
111A 8B A4                ADDA #$A4
111C 81 4F          con1  CMPA #$4F
111E 2D 10                BLT  nor2
1120 81 51                CMPA #$51
1122 2E 02                BGT  con2
1124 8B 72                ADDA #$72
```

```
1126 81 67              con2   CMPA #$67
1128 2D 06                     BLT  nor2
112A 81 6B                     CMPA #$6B
112C 2E 02                     BGT  nor2
112E 8B 5E                     ADDA #$5E
1130 80 03              nor2   SUBA #$03
1132 44                        LSRA
1133 A6 A6                     LDA  A,Y
1135 10 BE 5F F4               LDY  $5FF4
1139 F6 5F F1                  LDB  $5FF1
113C B7 5F F1           final  STA  $5FF1
113F 39                        rts

                        *
                        *  subrutina de despliegue de motor de dc
                        *  me_mot.asm              921111
                        *
11B1                           org     $11b1

11B1 17 F2 77           com    lbsr cls
11B4 10 8E 98 28               ldy  #$9828
11B8 30 8D 01 72               leax $172,pc
11BC 17 F1 01                  lbsr escr       ;escribe encabezado
11BF 31 A8 4C                  leay $4c,y
11C2 17 F0 FB                  lbsr escr       ;escribe
11C5 31 29                     leay $09,y      ;características
11C7 17 F0 F6                  lbsr escr       ;escribe voltaje
11CA 31 A8 14                  leay $14,y
11CD 17 F0 F0                  lbsr escr       ;escribe velocidad
11D0 31 A8 2B                  leay $2b,y
11D3 17 F0 EA                  lbsr escr       ;escribe reporte
11D6 31 2E                     leay $0e,y
11D8 17 F0 E5                  lbsr escr       ;escribe modo
11DB 31 2E                     leay $0e,y
11DD 17 F0 E0                  lbsr escr       ;escribe estado
11E0 31 A8 EC                  leay -$14,y
11E3 B6 A0 20           lesta  lda  edo
11E6 36 02                     pshu a
11E8 85 01                     bita #$01
11EA 26 09                     bne  auto
11EC 17 F0 D1                  lbsr escr       ;escribe manual
11EF 30 07                     leax $07,x
11F1 31 2F                     leay $0f,y
11F3 20 07                     bra  c1
11F5 30 07              auto   leax $07,x
11F7 17 F0 C6                  lbsr escr       ;escribe auto
11FA 31 2F                     leay $0f,y
11FC 85 04              c1     bita #$04
11FE 26 09                     bne  alto
1200 17 F0 BD                  lbsr escr       ;escribe encendido
1203 30 0A                     leax $0a,x
1205 31 2F                     leay $0f,y
1207 20 07                     bra  c2
1209 30 0A              alto   leax $0a,x
120B 17 F0 B2                  lbsr escr       ;escribe parado
120E 31 2F                     leay $0f,y
```

```
1210 84 05          c2      anda #$05
1212 81 05                  cmpa #$05
1214 27 4E                  beq  omit
1216 17 F0 A7               lbsr escr       ;escribe sentido
1219 37 02                  pulu a
121B 36 02                  pshu a
121D 84 0F                  anda #$0f
121F 27 23                  beq  anti
1221 81 02                  cmpa #$02
1223 10 27 00 DE            lbeq erro
1227 81 08                  cmpa #$08
1229 10 27 00 D8            lbeq erro
122D 84 0E                  anda #$0e
122F 81 04                  cmpa #$04
1231 27 11                  beq  anti
1233 81 0C                  cmpa #$0c
1235 27 0D                  beq  anti
1237 84 0C                  anda #$0c
1239 27 09                  beq  anti
123B 30 04                  leax $04,x
123D 17 F0 80               lbsr escr       ;escribe horario
1240 31 25                  leay $05,y
1242 20 05                  bra  c3
1244 17 F0 79       anti    lbsr escr       ;escribe antihorario
1247 31 21                  leay $01,y
1249 37 02          c3      pulu a
124B 36 02                  pshu a
124D 85 04                  bita #$04
124F 26 13                  bne  omit
1251 17 F0 6C               lbsr escr       ;escribe velocidad
1254 17 02 5C               lbsr leve
1257 17 F4 86               lbsr decim      ;escribe el valor
125A 37 02                  pulu a          ;escribe RPM
125C 36 02                  pshu a
125E 85 01                  bita #$01
1260 27 38                  beq  esc
1262 20 27                  bra  opti
1264 10 8E 99 60    omit    ldy  #$9960     ;limpia en  caso de
1268 8E 00 3F               ldx  #$3f       ;no estar encendido
126B C6 20                  ldb  #$20       ;la velocidad  y el
126D 85 01                  bita #$01       ;sentido en el modo
126F 26 06                  bne  pros       ;manual
1271 31 A8 20               leay $20,y
1274 30 88 20               leax $20,x
1277 F7 A0 08       pros    stb  $a008
127A E7 A0          lim1    stb  ,y+
127C 30 82                  leax ,-x
127E 26 FA                  bne  lim1
1280 F7 A0 09               stb  $a009
1283 85 01                  bita #$01
1285 27 13                  beq  esc
1287 30 8D 01 70            leax $170,pc
128B 10 8E 99 C1    opti    ldy  #$99c1
128F 17 F0 2E               lbsr escr       ;escribe deseas
1292 31 A8 11               leay $11,y
1295 17 F0 28               lbsr escr       ;escribe su operacion
```

```
1298 20 15                    bra  poleo
129A 10 8E 99 C1      esc     ldy  #$99c1     ;limpia en caso de
129E 8E 00 35                 ldx  #$35       ;modificar el modo
12A1 C6 20                    ldb  #$20       ;deseas modificar
12A3 F7 A0 08                 stb  $a008
12A6 E7 A0            lim2    stb  ,y+
12A8 30 82                    leax -x
12AA 26 FA                    bne  lim2
12AC F7 A0 09                 stb  $a009
12AF 8E FF FF         poleo   ldx  #$ffff     ;espera la tecla de
12B2 37 04                    pulu b
12B4 B6 A0 04          loop    lda  $a004       ;esc en cualquier
12B7 84 01                    anda #$01       ;modo y la s o la n
12B9 26 0F                    bne  tec        ;en el modo auto
12BB 30 82                    leax ,-x        ;durante un tiempo
12BD 26 F5                     bne  loop       ;sino refresca los
12BF 30 8D 00 E5              leax $e5,pc     ;datos, apunta a
12C3 10 8E 99 37              ldy  #$9937     ;manual
12C7 16 FF 19                 lbra lesta
12CA 17 FD 55         tec     lbsr beta       ;se detecto una tecla
12CD 81 1B                    cmpa #$1b
12CF 27 23                    beq  reg
12D1 C5 01                    bitb #$01
12D3 27 DF                    beq  loop
12D5 81 53                    cmpa #$53       ;opción s
12D7 26 1C                    bne  no
12D9 17 F0 85                 lbsr esca
12DC 17 FD 43                 lbsr beta
12DF 81 0D                    cmpa #$0d
12E1 27 0B                    beq  modif
12E3 31 A2            reto    leay ,-y
12E5 86 20                    lda  #$20
12E7 17 F0 77                 lbsr esca
12EA 31 A2                    leay ,-y
12EC 20 C6                    bra  loop
12EE 17 01 DD         modif   lbsr cambia
12F1 16 FE BD                 lbra com
12F4 39               reg     rts             ;opción esc
12F5 81 4E            no      cmpa #$4e       ;opción n
12F7 26 BB                    bne  loop
12F9 17 F0 65                 lbsr esca
12FC 17 FD 23                 lbsr beta
12FF 81 0D                    cmpa #$0d
1301 27 F1                    beq  reg
1303 20 DE                    bra  reto
1305 10 8E 99 23      erro    ldy  #$9923
1309 30 8D 01 13              leax $113,pc
130D 17 EF B0                 lbsr escr
1310 31 26                    leay $06,y
1312 17 EF AB                 lbsr escr
1315 31 25                    leay $05,y
1317 17 EF A6                 lbsr escr
131A 31 23                    leay $03,y
131C 17 EF A1                 lbsr escr
131F 31 A8 2B                 leay $2b,y
1322 17 EF 9B                 lbsr escr
```

```
1325 31 23                      leay $03,y
1327 17 EF 96                   lbsr escr
132A 17 FC F5                   lbsr beta
132D 39                         rts

                        *
                        *  subrutina de despliegue de motor de dc
                        *  me_mot2.asm              921117
                        *
14B3                            org $14b3

14B3 C6 03              leve    ldb  #$03       ;subrutina de lectura
14B5 36 20                      pshu y          ;de la velocidad
14B7 B7 A0 1C           c4      sta  vel
14BA 10 8E 01 00                ldy  #$100
14BE 31 A2              c5      leay ,-y
14C0 26 FC                      bne  c5
14C2 B6 A0 1C                   lda  vel
14C5 5A                         decb
14C6 26 EF                      bne  c4
14C8 37 20                      pulu y
14CA C6 54                      ldb  #$54       ;ajuste de la
14CC 3D                         mul             ;velocidad
14CD 39                         rts
                        *
                        *       subrutina de modificacion de
                        *            operación
                        *
14CE 17 EF 5A           cambia lbsr cls
14D1 10 8E 98 28                ldy  #$9828
14D5 30 8D 01 40                leax $140,pc
14D9 17 ED E4                   lbsr escr       ;escribe encabezado
14DC 31 A8 2F                   leay $2f,y
14DF 17 ED DE                   lbsr escr       ;escribe 1
14E2 31 A8 17                   leay $17,y
14E5 17 ED D8                   lbsr escr       ;escribe a
14E8 31 A8 16                   leay $16,y
14EB 17 ED D2                   lbsr escr       ;escribe b
14EE 31 A8 29                   leay $29,y
14F1 17 ED CC                   lbsr escr       ;escribe 2
14F4 31 A8 32                   leay $32,y
14F7 17 ED C6                   lbsr escr       ; escribe 3
14FA 31 A8 20                   leay $20,y
14FD 17 ED C0                   lbsr escr       ;escribe a
1500 31 A8 14                   leay $14,y
1503 17 ED BA                   lbsr escr       ;escribe b
1506 B6 A0 20                   lda  edo
1509 85 04                      bita #$04       ;verifica se esta
150B 26 19                      bne  stop       ;encendido
150D 10 8E 98 8B                ldy  #$988b
1511 85 08                     bita #$08        ;indica en el sentido
1513 26 03                      bne  nof        ;esta girando
1515 31 A8 20                   leay $20,y
1518 86 2A              nof     lda  #$2a
151A 17 EE 44                   lbsr esca
151D 10 8E 99 4B                ldy  #$994b     ;indica que esta
```

```
1521 17 EE 3D            lbsr esca      ;encendido
1524 20 0B               bra  repite
1526 86 2A         stop  lda  #$2a      ;indica que esta
1528 10 8E 99 6B         ldy  #$996b    ;parado
152C 17 EE 32            lbsr esca
152F 20 17               bra  paro
1531 B6 A0 20      repite lda  edo
1534 85 01               bita #$01
1536 27 63               beq  manual    ;verifica si se
1538 85 04               bita #$04      ;modificó el modo de
153A 26 0C               bne  paro      ;operación
153C 10 8E 98 F2         ldy  #$98f2
1540 17 FF 70            lbsr leve
1543 17 F1 9A           lbsr decim     ;escribe la velocidad
1546 30 1C               leax -$04,x
1548 10 8E 99 A2   paro  ldy  #$99a2
154C 30 04               leax $04,x
154E 17 ED 6F      rep   lbsr escr      ;escribe modificar
1551 36 10               pshu x
1553 8E FF FF            ldx  #$ffff
1556 F6 A0 04      lazo  ldb  $a004
1559 C4 01               andb #$01
155B 26 0B               bne  tecla
155D 30 82               leax ,-x
155F 26 F5               bne  lazo
1561 37 10               pulu x
1563 30 88 E8            leax -$18,x
1566 20 C9               bra  repite
1568 17 FA B7      tecla lbsr beta      ;lee la opcion a
156B 81 1B               cmpa #$1b      ;a ejecutar 1 a 3
156D 27 41               beq  regre
156F F6 A0 20            ldb  edo
1572 C5 04               bitb #$04
1574 26 1F               bne  mayor
1576 81 31               cmpa #$31
1578 2D DC               blt  lazo
157A 81 33         normal cmpa #$33
157C 2E D8               bgt  lazo
157E 17 ED E0            lbsr esca
1581 1F 89               tfr  a,b
1583 17 FA 9C            lbsr beta
1586 81 0D               cmpa #$0d
1588 27 29               beq  corr
158A 31 A2               leay ,-y
158C 86 20               lda  #$20
158E 17 ED D0            lbsr esca
1591 31 A2               leay ,-y
1593 20 C1               bra  lazo
1595 81 33         mayor cmpa #$33
1597 2D BD               blt  lazo
1599 20 DF               bra  normal
159B 36 10        manual pshu x        ;se indica que se
159D 30 8D 01 0A         leax $10a,pc   ;modificó el modo de
15A1 10 8E 99 81         ldy  #$9981    ;operación espera una
15A5 17 ED 18            lbsr escr      ;tecla para continuar
15A8 31 23               leay $03,y
```

```
15AA 17 ED 13              lbsr escr
15AD 17 FA 72              lbsr beta
15B0 37 10          regre  pulu x          ;finaliza la opcion
15B2 39                    rts
15B3 17 01 A2       corr   lbsr liren
15B6 37 10                 pulu x
15B8 C1 32                 cmpb #$32
15BA 27 54                 beq  move       ;modifica velocidad
15BC 17 ED 01              lbsr escr       ;escribe opción
15BF 36 04          tecl   pshu b          ;deseada
15C1 17 FA 5E              lbsr beta
15C4 81 1B                 cmpa #$1b
15C6 27 29                 beq  reto2
15C8 81 41                 cmpa #$41
15CA 2D 25                 blt  reto2
15CC 81 42                 cmpa #$42
15CE 2E 21                 bgt  reto2
15D0 17 ED 8E              lbsr esca
15D3 1F 89                 tfr  a,b
15D5 17 FA 4A              lbsr beta
15D8 81 0D                 cmpa #$0d
15DA 27 24                 beq  eje
15DC 81 1B                 cmpa #$1b
15DE 27 11                 beq  reto2
15E0 81 08                 cmpa #$08
15E2 26 0D                 bne  reto2
15E4 31 A2                 leay ,-y
15E6 86 20                 lda  #$20
15E8 17 ED 76              lbsr esca
15EB 31 A2                 leay ,-y
15ED 37 04                 pulu b
15EF 20 CE                 bra  tecl
15F1 36 10          reto2  pshu x
15F3 17 01 62              lbsr liren
15F6 37 10                 pulu x
15F8 30 88 D9              leax -$27,x
15FB 37 04                 pulu b
15FD 16 FF 31       nu     lbra repite
1600 37 02          eje    pulu a
1602 36 10                 pshu x
1604 17 00 E3              lbsr real
1607 37 10                 pulu x
1609 30 11                 leax -$0f,x
160B 30 88 E8       nue    leax -$18,x
160E 20 ED                 bra  nu
1610 36 10          move   pshu x
1612 17 02 61              lbsr cave
1615 37 10                 pulu x
1617 20 F2                 bra  nue

16EA                       org $16ea

16EA 36 10          real   pshu  x
16EC 81 31                 cmpa #$31
16EE 10 27 01 46           lbeq sen        ;modifica sentido
16F2 C1 41                 cmpb #$41
```

```
16F4 27 59                    beq  arra      ;enciende el motor?
16F6 B6 A0 20                 lda  edo
16F9 84 04                    anda #$04
16FB 10 26 00 6D              lbne fin
16FF 86 20                    lda  #$20
1701 10 8E 98 8B              ldy  #$988b
1705 17 EC 59                 lbsr esca
1708 31 A8 1F                 leay $1f,y
170B 17 EC 53                 lbsr esca
170E 31 A8 44                 leay $44,y
1711 C6 0A                    ldb  #$0a
1713 17 EC 4B          live   lbsr esca      ;limpia las
1716 5A                       decb            ;localidades de la
1717 26 FA                    bne  live       ;velocidad
1719 10 8E 99 4B              ldy  #$994b
171D 17 EC 41                 lbsr esca
1720 86 2A                    lda  #$2a
1722 31 A8 1F                 leay $1f,y
1725 17 EC 39                 lbsr esca
1728 B6 5F F2                 lda  dac
172B 81 80             stop1  cmpa #$80      ;detiene el motor
172D 27 12                    beq  quie       ;gradualmente
172F 25 0D                    blo  incre
1731 4A                       deca
1732 B7 A0 20          alm    sta  edo
1735 8E 00 10                 ldx  #$10
1738 30 82             c7     leax ,-x
173A 26 FC                    bne  c7
173C 20 ED                    bra  stop1
173E 4C                incre  inca
173F 20 F1                    bra  alm
1741 B7 5F F2          quie   sta  dac
1744 B7 A0 24                 sta  $a024
1747 7F 5F F0                 clr  velus
174A 7F 5F F1                 clr  velus+1
174D 20 1D                    bra  fin
174F B6 A0 20          arra   lda  edo       ;enciende el motor
1752 84 04                    anda #$04
1754 27 16                    beq  fin
1756 20 5C                    bra  salto
1758 10 8E 99 A0       liren  ldy  #$99a0    ;limpia el renglon
175C 86 20                    lda  #$20       ;de comandos
175E 8E 00 1F                 ldx  #$1f
1761 17 EB FD          borr   lbsr esca
1764 30 82                    leax ,-x
1766 26 F9                    bne  borr
1768 31 A8 E2                 leay -$1e,y
176B 39                       rts
176C 8D EA             fin    bsr  liren     ;finaliza la opcion
176E 39                       rts

17B4                          org $17b4

17B4 8D A2             salto  bsr  liren
17B6 30 8C B7                 leax -$49,pc    ;apunta a sentido?
17B9 17 EB 04                 lbsr escr
```

```
17BC 17 00 EE            lbsr lot       ;polea el teclado
17BF C1 1B               cmpb #$1b      ;para  la  opcion
17C1 27 A9               beq  fin       ;A, B o esc
17C3 31 30               leay -$10,y
17C5 36 04               pshu b
17C7 17 EA F6      otto  lbsr escr      ;escribe velocidad?
17CA 17 01 4F            lbsr reciv     ;polea el teclado
17CD C1 1B               cmpb #$1b      ;para 5 digitos o
17CF 26 04               bne  s1        ;esc
17D1 37 04               pulu b
17D3 20 97               bra  fin
17D5 36 20         s1    pshu y
17D7 17 F1 CB            lbsr calcula   ;convirte e d a h
17DA 37 20               pulu y
17DC 10 83 08 34         cmpd #$0834
17E0 2E 4B               bgt  error
17E2 FD 5F F0            std  velus
17E5 8E 00 04            ldx  #$04
17E8 44            rot   lsra
17E9 56                  rorb
17EA 30 82               leax ,-x
17EC 26 FA               bne  rot
17EE 37 02               pulu a
17F0 81 41               cmpa #$41
17F2 26 27               bne  hor
17F4 86 FF               lda  #$ff
17F6 8D 08               bsr  arr
17F8 8D 13         m1    bsr  pau
17FA 4A                  deca
17FB 5A                  decb
17FC 2E FA               bgt  m1
17FE 20 24               bra  ala
1800 B7 A0 20      arr   sta  $a020
1803 17 00 D8            lbsr actua
1806 C4 7E               andb #$7e
1808 5C                  incb
1809 B7 A0 26            sta  $a026
180C 39                  rts
180D 34 10         pau   pshs x
180F 8E 01 00            ldx  #$100
1812 B7 A0 20            sta  $a020
1815 30 82         ret   leax ,-x
1817 26 FC               bne  ret
1819 35 90               puls x,pc
181B 4F            hor   clra
181C 8D E2               bsr  arr
181E 8D ED         m2    bsr  pau
1820 4C                  inca
1821 5A                  decb
1822 2C FA               bge  m2
1824 B7 5F F2      ala   sta  dac
1827 17 01 73            lbsr ajvel
182A 16 FF 3F            lbra fin

182D 31 A8 EE      error leay -$12,y
1830 17 EA 8D            lbsr escr
```

```
1833 17 F7 EC          nec     lbsr beta
1836 81 0D                     cmpa #$0d
1838 26 F9                     bne  nec
183A 17 FF 1B                  lbsr liren
183D 30 88 CE                  leax -$32,x
1840 20 85                     bra  otto
1842 B6 5F F2          sen     lda  dac
1845 C1 41                     cmpb #$41
1847 26 1A                     bne  hor2
1849 81 80                     cmpa #$80
184B 10 22 FF 1D               lbhi fin
184F 73 5F F2                  com  dac
1852 4C                rep1    inca
1853 8D B8                     bsr  pau
1855 B1 5F F2                  cmpa dac
1858 26 F8                     bne  rep1
185A 17 00 96          te      lbsr actua2
185D 17 01 3D                  lbsr ajvel
1860 16 FF 09                  lbra fin
1863 81 80             hor2    cmpa #$80
1865 10 25 FF 03               lblo fin
1869 73 5F F2                  com  dac
186C 4A                rep2    deca
186D 8D 9E                     bsr  pau
186F B1 5F F2                  cmpa dac
1872 26 F8                     bne  rep2
1874 20 E4                     bra  te
1876 30 8D FF 06       cave    leax -$fa,pc
187A 17 EA 43                  lbsr escr
187D 17 00 9C                  lbsr reciv
1880 C1 1B                     cmpb #$1b
1882 27 13                     beq  fin2
1884 36 20                     pshu y
1886 17 F1 1C                  lbsr calcula
1889 37 20                     pulu y
188B 10 83 08 34               cmpd #$0834
188F 2E 0A                     bgt  error2
1891 FD 5F F0                  std  velus
1894 17 01 06                  lbsr ajvel
1897 17 FE BE          fin2    lbsr liren
189A 39                        rts
189B 31 A8 EE          error2  leay -$12,y
189E 17 EA 1F                  lbsr escr
18A1 17 F7 7E          nec2    lbsr beta
18A4 81 0D                     cmpa #$0d
18A6 26 F9                     bne  nec2
18A8 17 FE AD                  lbsr liren
18AB 20 C9                     bra  cave
18AD 17 F7 72          lot     lbsr beta
18B0 81 1B                     cmpa #$1b
18B2 27 27                     beq  ffin
18B4 81 41                     cmpa #$41
18B6 2D F5                     blt  lot
18B8 81 42                     cmpa #$42
18BA 2E F1                     bgt  lot
18BC 17 EA A2                  lbsr esca
```

```
18BF 1F 89              tfr  a,b
18C1 17 F7 5E       re  lbsr beta
18C4 81 0D              cmpa #$0d
18C6 27 15              beq  tter
18C8 81 1B              cmpa #$1b
18CA 27 0F              beq  ffin
18CC 81 08              cmpa #$08
18CE 26 F1              bne  re
18D0 31 A2              leay ,-y
18D2 86 20              lda  #$20
18D4 17 EA 8A           lbsr esca
18D7 31 A2              leay ,-y
18D9 20 D2              bra  lot
18DB 1F 89          ffin tfr  a,b
18DD 39             tter rts
18DE 36 22          actua pshu y,a
18E0 10 8E 99 4B        ldy  #$994b
18E4 86 2A              lda  #$2a
18E6 17 EA 78           lbsr esca
18E9 86 20              lda  #$20
18EB 31 A8 1F           leay $1f,y
18EE 17 EA 70           lbsr esca
18F1 37 22              pulu y,a
18F3 36 22          actua2 pshu y,a
18F5 10 8E 98 8B        ldy  #$988b
18F9 81 80              cmpa #$80
18FB 24 0F              bhs  hor3
18FD 86 20              lda  #$20
18FF 17 EA 5F           lbsr esca
1902 31 A8 1F           leay $1f,y
1905 86 2A              lda  #$2a
1907 17 EA 57           lbsr esca
190A 20 0D              bra  fac
190C 86 2A          hor3 lda  #$2a
190E 17 EA 50           lbsr esca
1911 31 A8 1F           leay $1f,y
1914 86 20              lda  #$20
1916 17 EA 48           lbsr esca
1919 37 22          fac  pulu y,a
191B 39                 rts
191C 36 10          reciv pshu x
191E 36 20              pshu y
1920 17 F0 CC           lbsr limpia
1923 8E 40 31           ldx  #$4031
1926 C6 05              ldb  #$05
1928 37 20              pulu y
192A 36 20              pshu y
192C 8D 32          otdi bsr  ledi
192E 81 1B              cmpa #$1b
1930 27 1F              beq  escapa
1932 5A                 decb
1933 27 23              beq  ovflo
1935 81 0D              cmpa #$0d
1937 27 1A              beq  final
1939 36 02              pshu a
193B A6 1D              lda  -$3,x
```

```
193D A7 1C                      sta  -$4,x
193F A6 1E                      lda  -$2,x
1941 A7 1D                      sta  -$3,x
1943 A6 1F                      lda  -$1,x
1945 A7 1E                      sta  -$2,x
1947 A6 84                      lda  ,x
1949 A7 1F                      sta  -$1,x
194B 37 02                      pulu a
194D A7 84                      sta  ,x
194F 20 DB                      bra  otdi
1951 C6 1B              escapa  ldb  #$1b
1953 37 20              final   pulu y
1955 37 10                      pulu x
1957 39                         rts
1958 CC 33 30           ovflo   ldd  #$3330
195B FD 40 2D                   std  $402d
195E 20 F3                      bra  final
1960 17 F6 BF           ledi    lbsr beta
1963 81 1B                      cmpa #$1b
1965 27 13                      beq  fffin
1967 81 0D                      cmpa #$0d
1969 27 0F                      beq  fffin
196B 81 08                      cmpa #$08
196D 27 0C                      beq  bs
196F 81 39                      cmpa #$39
1971 2E ED                      bgt  ledi
1973 81 30                      cmpa #$30
1975 2D E9                      blt  ledi
1977 17 E9 E7                   lbsr esca
197A 39                 fffin   rts
197B 31 A2              bs      leay ,-y
197D 86 20                      lda  #$20
197F 17 E9 DF                   lbsr esca
1982 31 A2                      leay ,-y
1984 5C                         incb
1985 96 00                      lda
1987 20 D7                      bra  ledi
1989 A6 1F                      lda  -$1,x
198B A7 84                      sta  ,x
198D A6 1E                      lda  -$2,x
198F A7 1F                      sta  -$1,x
1991 A6 1D                      lda  -$3,x
1993 A7 1E                      sta  -$2,x
1995 A6 1C                      lda  -$4,x
1997 A7 1D                      sta  -$3,x
1999 86 30                      lda  #$30
199B A7 1C                      sta  -$4,x
199D 12                 ajvel   nop
199E 39                         rts

199D                        org $199d

199D FC 5F FB           ajvel   ldd velus
19A0 26 04                      bne  difi
19A2 B7 A0 24                   sta  $a024
19A5 39                         rts
```

```
19A6 36 30          difi   pshu x,y
19A8 7F 5F F0              clr  bande
19AB 17 FB 05              lbsr leve
19AE B3 5F FB              subd velus
19B1 2E 09                 bgt  mayor
19B3 27 1A                 beq  fin
19B5 7A 5F F0              dec  bande
19B8 53             menor  comb
19B9 43                    coma
19BA C9 00                 adcb #$00
19BC 1F 02          mayor  tfr  d,y
19BE B6 5F FD              lda  dac
19C1 81 80                 cmpa #$80
19C3 25 57                 blo  cero
19C5 5F                    clrb
19C6 F1 5F F0              cmpb bande
19C9 27 2F                 beq  decre
19CB 81 FF                 cmpa #$ff
19CD 26 03                 bne  incre
19CF 37 30          fin    pulu x,y
19D1 39                    rts
19D2 17 00 85       incre  lbsr suma
19D5 B3 5F FB              subd velus
19D8 2E 04                 bgt  mayor2
19DA 27 F3                 beq  fin
19DC 20 DA                 bra  menor
19DE BE 5F FB       mayor2 ldx  velus
19E1 10 BF 5F FB           sty  velus
19E5 10 B3 5F FB           cmpd velus
19E9 2F 0A                 ble  final
19EB B6 5F FD       decre2 lda  dac
19EE 4A                    deca
19EF B7 5F FD              sta  dac
19F2 B7 A0 20              sta  edo
19F5 BF 5F FB       final  stx  velus
19F8 20 D5                 bra  fin
19FA 8D 6B          decre  bsr  resta
19FC B3 5F FB              subd velus
19FF 2E BB                 bgt  mayor
1A01 27 CC                 beq  fin
1A03 BE 5F FB              ldx  velus
1A06 10 BF 5F FB           sty  velus
1A0A 10 B3 5F FB           cmpd velus
1A0E 2F E5                 ble  final
1A10 B6 5F FD       incre2 lda  dac
1A13 4C                    inca
1A14 B7 5F FD              sta  dac
1A17 B7 A0 20              sta  edo
1A1A 20 D9                 bra  final
1A1C 5F             cero   clrb
1A1D F1 5F F0              cmpb bande
1A20 27 1E                 beq  incre3
1A22 81 00                 cmpa #$00
1A24 27 A9                 beq  fin
1A26 8D 3F                 bsr  resta
1A28 B3 5F FB              subd velus
```

```
1A2B 2E 04              bgt  mayor3
1A2D 27 A0              beq  fin
1A2F 20 87              bra  menor
1A31 BE 5F FB     mayor3 ldx  velus
1A34 10 BF 5F FB        sty  velus
1A38 10 B3 5F FB        cmpd velus
1A3C 2F B7              ble  final
1A3E 20 D0              bra  incre2
1A40 8D 18        incre3 bsr  suma
1A42 B3 5F FB           subd velus
1A45 10 2E FF 73        lbgt mayor
1A49 27 84              beq  fin
1A4B BE 5F FB           ldx  velus
1A4E 10 BF 5F FB        sty  velus
1A52 10 B3 5F FB        cmpd velus
1A56 2F 9D              ble  final
1A58 20 91              bra  decre2
1A5A 4C           suma  inca
1A5B B7 5F FD     lomismo sta  dac
1A5E B7 A0 20           sta  edo
1A61 8D 07              bsr  retardo
1A63 17 FA 4D           lbsr leve
1A66 39                 rts
1A67 4A           resta deca
1A68 20 F1              bra lomismo
1A6A 8E 0F FF     retardo ldx  #$fff
1A6D 30 82        pausa leax ,-x
1A6F 26 FC              bne  pausa
1A71 39                 rts
                  *
                  *     POTENCIA.ASM
                  *                  22-9-92
                  *
                  *  PROGRAMA QUE CONTROLA EL BANCO DE
                  *  CAPACITORES PARA PODER MODIFICAR
                  *  EL f.p.
                  *
E790                    org $e790

E790 34 36              pshs y,x,b,a    ;menu principal
E792 BD E4 2B     ini   jsr cls
E795 8E EE 50           ldx #i          ;borrado de la
E798 BD EA 3E           jsr despli      ;pantalla
E79B 8E EE B1           ldx #ii         ;falta
E79E BD EA 3E           jsr despli
E7A1 BD EA C0           jsr fpsp
E7A4 86 55              lda #$55
E7A6 B1 40 1A           cmpa $401a
E7A9 27 0B              beq ini2
E7AB B7 40 1A           sta $401a
E7AE 86 00              lda #$00
E7B0 B7 40 16           sta $4016
E7B3 B7 A0 00           sta piabfa
E7B6 BD F0 22     ini2  jsr beta        ;poleo del teclado
E7B9 81 1B              cmpa #$1b
E7BB 27 41              beq seg3
```

```
E7BD 81 31              cmpa #$31
E7BF 27 04              beq ter
E7C1 81 32              cmpa #$32
E7C3 26 F1              bne ini2
E7C5 B7 A0 08     ter   sta $a008
E7C8 B7 99 B7           sta $99b7
E7CB B7 A0 09           sta $a009
E7CE B7 40 14           sta $4014
E7D1 BD F0 22     cuar  jsr beta
E7D4 81 08              cmpa #$08       ;codigo de backspace
E7D6 26 11              bne quin
E7D8 86 20              lda #$20
E7DA B7 A0 08           sta $a008
E7DD B7 99 B7           sta $99b7
E7E0 B7 A0 09           sta $a009
E7E3 20 D1              bra ini2
E7E5 81 1B              cmpa #$1b
E7E7 27 15              beq seg3
E7E9 81 0D        quin  cmpa #$0d       ;codigo de enter
E7EB 26 E4              bne cuar
E7ED B6 40 14           lda $4014
E7F0 81 31              cmpa  #$31      ;¿es 1?
E7F2 26 05              bne prim
E7F4 BD E8 0C           jsr uno         ;primera pantalla
E7F7 20 99              bra ini
E7F9 BD E8 71     prim  jsr dos         ;¿es 2?
E7FC 20 94              bra ini         ;segunda pantalla
E7FE 86 AA        seg3  lda #$aa
E800 B1 40 19           cmpa $4019
E803 26 05              bne seg4
E805 86 5A              lda #$5a
E807 B7 40 19           sta $4019
E80A 35 B6        seg4  puls pc,y,x,b,a
                  *
                  * PRIMERA PANTALLA (uno)
                  *

E80C 86 55        uno   lda #$55        ;indica deshabilitar
E80E B7 40 19           sta $4019       ;interrupciones
E811 BD E4 2B           jsr cls
E814 8E EE C7           ldx #iii
E817 BD EA 3E           jsr despli
E81A 8E EE B1           ldx #ii
E81D BD EA 3E           jsr despli
E820 BD EA 90           jsr cone
E823 BD EA 55           jsr des
E826 BD EA C0           jsr fpsp
E829 BD F0 22     comi  jsr beta
E82C 81 1B              cmpa #$1b
E82E 26 01              bne sigue
E830 39                 rts
E831 80 34        sigue suba #$34
E833 2E F4              bgt comi
E835 8B 04              adda #$4
E837 2F F0              ble comi
                  *
```

```
                        * ANALISIS DEL DATO
                        *
E839 B7 40 18                   sta $4018
E83C C6 08                      ldb #$08
E83E 58                 cont    lslb
E83F 4A                         deca
E840 26 FC                      bne cont
E842 F7 40 17                   stb $4017
E845 F4 40 16                   andb $4016
E848 26 0E                      bne desa
E84A F6 40 17                   ldb $4017       ;no esta habilitado
E84D FA 40 16                   orb $4016
E850 F7 40 16                   stb $4016
E853 F7 A0 00                   stb piabfa
E856 20 0A                      bra desa2
E858 53                 desa    comb
E859 F4 40 16                   andb $4016
E85C F7 40 16                   stb $4016
E85F F7 A0 00                   stb piabfa
E862 F6 40 18           desa2   ldb $4018
E865 8E EF 8E                   ldx #iv
E868 58                         lslb
E869 10 AE 85                   ldy b,x
E86C BD EA 6A                   jsr corri
E86F 20 B8                      bra comi

                        *
                        * SEGUNDA PANTALLA
                        *

E871 BD E4 2B           dos     jsr cls
E874 8E EF 3B                   ldx #v
E877 BD EA 3E                   jsr despli
E87A 8E EE B1                   ldx #ii
E87D BD EA 3E                   jsr despli
E880 BD EA 90                   jsr cone
E883 BD EA 55                   jsr des         ;despliegue de
E886 BD E9 FD                  jsr valor       ;pantalla, escribe el
E889 BD EA C0                   jsr fpsp         ;f.p.
E88C 86 AA                      lda #$aa
E88E B7 40 19                   sta $4019
E891 16 00 8A                   lbra segundo3

E894 BD F0 22           se      jsr beta

E897 81 1B                      cmpa #$1b
E899 26 01                      bne seg2
E89B 39                         rts

E89C 81 08              seg2    cmpa #$08
E89E 26 65                      bne segun
E8A0 8E 40 29                   ldx #$4029
E8A3 BC 40 23                   cmpx $4023
E8A6 27 5D                      beq segun
E8A8 30 01                      leax +1,x
```

```
E8AA BC 40 23                   cmpx $4023
E8AD 26 13                      bne segund
E8AF C6 30                      ldb #$30
E8B1 F7 40 20                   stb $4020
E8B4 5C                         incb
E8B5 F7 40 21                   stb $4021
E8B8 C6 2E                      ldb #$2e
E8BA F7 40 22                   stb $4022
E8BD B7 40 28                   sta $4028
E8C0 20 26                      bra segundo
E8C2 30 01              segund  leax +1,x
E8C4 BC 40 23                   cmpx $4023
E8C7 26 13                      bne segundo10
E8C9 C6 FF                      ldb #$ff
E8CB F7 40 20                   stb $4020
E8CE F6 40 29                   ldb $4029
E8D1 C1 31                      cmpb #$31
E8D3 27 13                      beq segundo
E8D5 C6 2E                      ldb #$2e
E8D7 F7 40 22                   stb $4022
E8DA 20 0C                      bra segundo
E8DC 30 01              segundo10 leax +1,x
E8DE BC 40 23                   cmpx $4023
E8E1 27 05                      beq segundo
E8E3 C6 30                      ldb #$30
E8E5 F7 40 20                   stb $4020
E8E8 BE 40 23           segundo ldx $4023
E8EB 10 BE 40 25                ldy $4025
E8EF C6 20                      ldb #$20
E8F1 B7 A0 08                   sta $a008
E8F4 E7 82                      stb ,-x
E8F6 E7 A2                      stb ,-y
E8F8 B7 A0 09                   sta $a009
E8FB BF 40 23                   stx $4023
E8FE 10 BF 40 25                sty $4025
E902 7C 40 27                   inc $4027
E905 81 0D              segun   cmpa #$0d
E907 26 2A                      bne segundo1
E909 C6 31                      ldb #$31
E90B F1 40 29                   cmpb $4029
E90E 27 08                      beq segundo2
E910 8E 40 2C                   ldx #$402c
E913 BC 40 23                   cmpx $4023
E916 2E 06                      bgt segundo3
E918 BD EC 39           segundo2  jsr cambia
E91B BD E9 FD                   jsr valor
E91E 86 20              segundo3  lda #$20
E920 BD EC 19                   jsr pone
E923 B7 A0 08                   sta $a008
E926 A7 80              segundo4  sta ,x+
E928 A7 A0                      sta ,y+
E92A 5A                         decb
E92B 26 F9                      bne segundo4
E92D B7 A0 09                   sta $a009
E930 16 FF 61                   lbra se
E933 B1 40 22           segundo1  cmpa $4022
```

```
E936 26 40              bne segundo6
E938 8E 40 2A           ldx #$402a
E93B BC 40 23           cmpx $4023
E93E 27 1C              beq segundo7
E940 7C 40 24           inc $4024
E943 7C 40 26           inc $4026
E946 C6 FF              ldb #$ff
E948 F7 40 28           stb $4028
E94B 7A 40 27           dec $4027
E94E C6 30              ldb #$30
E950 F7 A0 08           stb $a008
E953 F7 99 D7           stb $99d7
E956 F7 A0 09           stb $a009
E959 F7 40 29           stb $4029
E95C BD EC 00         segundo7  jsr manejo
E95F F6 40 29           ldb $4029
E962 C1 31              cmpb #$31
E964 27 0A              beq segundo12
E966 C6 30              ldb #$30
E968 F7 40 20           stb $4020
E96B C6 39              ldb #$39
E96D F7 40 21           stb $4021
E970 C6 FF            segundo12 ldb #$ff
E972 F7 40 22           stb $4022
E975 7A 40 27           dec $4027
E978 B1 40 20         segundo6  cmpa $4020
E97B 10 25 FF 15        lblo se
E97F B1 40 21           cmpa $4021
E982 10 22 FF 0E        lbhi se
E986 C6 FF              ldb #$ff
E988 F1 40 28           cmpb $4028
E98B 27 06              beq segundo8
E98D F7 40 28           stb $4028
E990 F7 40 20           stb $4020
E993 BD EC 00         segundo8  jsr manejo
E996 7A 40 27           dec $4027
E999 10 26 FE F7        lbne se
E99D C6 FF              ldb #$ff
E99F F7 40 20           stb $4020
E9A2 16 FE EF           lbra se

                      *
                      * SUBRUTINAS
                      *
                      * CAMBIA HEXA A DECIMAL (calcula)
                      *

9A5 10 8E 40 2D       calcula ldy #$402d
E9A9 C6 05              ldb #$5
E9AB A6 A4            cal     lda ,y
E9AD 80 30              suba #$30
E9AF A7 A0              sta ,y+
E9B1 5A                 decb
E9B2 26 F7              bne cal
E9B4 E6 3E              ldb -2,y
E9B6 86 0A              lda #$a
```

```
E9B8 3D                         mul
E9B9 6F 3E                      clr -2,y
E9BB E3 3E                      addd -2,y
E9BD ED 3E                      std -2,y
E9BF E6 3D                      ldb -3,y
E9C1 86 64                      lda #$64
E9C3 3D                         mul
E9C4 E3 3E                      addd -2,y
E9C6 ED 3E                      std -2,y
E9C8 E6 3C                      ldb -4,y
E9CA 86 E8                      lda #$e8
E9CC 3D                         mul
E9CD E3 3E                      addd -2,y
E9CF ED 3E                      std -2,y
E9D1 E6 3C                      ldb -4,y
E9D3 86 03                      lda #$3
E9D5 3D                         mul
E9D6 EB 3E                      addb -2,y
E9D8 E7 3E                      stb -2,y
E9DA E6 3B                      ldb -5,y
E9DC 86 10                      lda #$10
E9DE 3D                         mul
E9DF E3 3E                      addd -2,y
E9E1 ED 3E                      std -2,y
E9E3 E6 3B                      ldb -5,y
E9E5 86 27                      lda #$27
E9E7 3D                         mul
E9E8 1E 89                      exg a,b
E9EA E3 3E                      addd -2,y
E9EC ED 3E                      std -2,y
E9EE 39                         rts

                        *
                        * LIMPIA REGISTRO DE DECIMAL A HEXA
                        *

E9EF 10 8E 40 2D        limpia  ldy #$402d
E9F3 C6 05                      ldb #$5
E9F5 86 30                      lda #$30
E9F7 A7 A0              lim     sta ,y+
E9F9 5A                         decb
E9FA 26 FB                      bne lim
E9FC 39                         rts

                        *
                        * CAMBIA DE DECIMAL A HEXA EL VALOR DEL
                        * F.P.     (valor)
                        *

E9FD BD EB 3D           valor   jsr fp
EA00 BD E9 EF                   jsr limpia
EA03 BD EA 0D                   jsr conver
EA06 BD E9 A5                   jsr calcula
EA09 FD 40 08                   std $4008
EA0C 39                         rts
```

```
                        *
                        * PREPARA EL VALOR MINIMO DEL F.P. DADO
                        * EN DECIMAL PARA CONVERTIRLO EN HEXA
                        *  (conver)
                        *
EA0D 8E 40 05           conver ldx #$4005
EA10 10 8E 40 2E               ldy #$402e
EA14 A6 80                     lda ,x+
EA16 A7 A0                     sta ,y+
EA18 81 31                     cmpa #$31
EA1A 27 0E                     beq a1
EA1C A6 80                     lda ,x+
EA1E A7 A0                     sta ,y+
EA20 A6 80                     lda ,x+
EA22 81 20                     cmpa #$20
EA24 26 02                     bne a2
EA26 8B 10                     adda #$10
EA28 A7 A0              a2     sta ,y+
EA2A 39                 a1     rts

                        *
                        * LEE VOLTAJE Y CORRIENTE (lecturas)
                        *
EA2B BD E7 22           lecturas  jsr ajvo    ;lee voltaje
EA2E FD 40 01                  std $4001      ;y corriente
EA31 BD E7 40                  jsr ajco       ; ademas calcula
EA34 FD 40 0A                  std $400a      ;las potencias
EA37 BD EB F1                  jsr aparente
EA3A BD EB 7B                  jsr real
EA3D 39                        rts

                        *
                        * DESPLIEGUE DE PANTALLA (despli)
                        *
EA3E A6 80              despli lda ,x+
EA40 B7 40 00                  sta $4000
EA43 BD EA 4C           inic2  jsr asigna
EA46 7A 40 00                  dec $4000
EA49 26 F8                     bne inic2
EA4B 39                        rts

                        *
                        * ASIGNA DIRECCIONES DE DATOS (asigna)
                        *
EA4C E6 80              asigna ldb ,x+
EA4E 10 AE 81                  ldy ,x++
EA51 BD EB 2F                  jsr mues
EA54 39                        rts

                        *
                        * DESPLIEGA "DES" PARA COMPLETAR
                        *   "DESCONECTADO" (des)
```

```
                        *
EA55 C6 10              des     ldb #$10
EA57 F7 40 17                   stb $4017
EA5A C6 04                      ldb #$4
EA5C 10 8E 99 08                ldy #$9908
EA60 BD EA 6A           comp    jsr corri
EA63 31 A8 20                   leay $20,y
EA66 5A                         decb
EA67 26 F7                      bne comp
EA69 39                         rts

                        *
                        * DESPLIEGA O NO "DES" (corri)
                        *

EA6A B7 A0 08           corri   sta $a008
EA6D B6 40 17                   lda $4017
EA70 78 40 17                   lsl $4017
EA73 B4 40 16                   anda $4016
EA76 27 08                      beq corri1
EA78 86 20                      lda #$20
EA7A A7 A4                      sta ,y
EA7C A7 21                      sta +1,y
EA7E 20 0A                      bra corri2
EA80 86 04              corri1  lda #$4
EA82 A7 A4                      sta ,y
EA84 86 05                      lda #$5
EA86 A7 21                      sta +1,y
EA88 86 13                      lda #$13
EA8A A7 22              corri2  sta +2,y
EA8C B7 A0 09                   sta $a009
EA8F 39                         rts

                        *
                        * DESPLIEGA CONECTADO    (cone)
                        *

EA90 C6 04              cone    ldb #$04
EA92 F7 A0 08                   stb $a008
EA95 F7 40 14                   stb $4014
EA98 10 8E 99 06                ldy #$9906
EA9C C6 31                      ldb #$31
EA9E 86 09              conec   lda #$09
EAA0 B7 40 15                   sta $4015
EAA3 E7 A4                      stb ,y
EAA5 5C                         incb
EAA6 8E EF 32                   ldx  #vi
EAA9 31 25                      leay +5,y
EAAB A6 80              tado    lda ,x+
EAAD A7 A0                      sta ,y+
EAAF 7A 40 15                   dec $4015
EAB2 26 F7                      bne tado
EAB4 31 A8 12                   leay $12,y
EAB7 7A 40 14                   dec $4014
EABA 26 E2                      bne conec
```

```
EABC F7 A0 09            stb $a009
EABF 39                  rts

                         *
                         * DESPLIEGA EL F.P., S y P
                         *

EAC0 FC 40 03            fpsp   ldd $4003
EAC3 BD E2 D5                   jsr he_de
EAC6 10 8E 98 74                ldy #$9874
EACA BD E6 34                   jsr escfp
EACD FC 40 1C                   ldd $401c
EAD0 BD E2 D5                   jsr he_de
EAD3 10 8E 98 94                ldy #$9894
EAD7 BD EA F4                   jsr saca
EADA 8E EF A6                   ldx #ix
EADD BD E2 C0                   jsr escr
EAE0 FC 40 1E                   ldd $401e
EAE3 BD E2 D5                   jsr he_de
EAE6 10 8E 98 B4                ldy #$98b4
EAEA BD EA F4                   jsr saca
EAED 8E EF AC                   ldx #x
EAF0 BD E2 C0                   jsr escr
EAF3 39                         rts

                         *
                         * ESCRIBE EN VALOR DECIMAL LAS POTENCIAS
                         *       (saca)
                         *

EAF4 B7 A0 08            saca   sta $a008
EAF7 86 20                      lda #$20
EAF9 C6 0D                      ldb #$0d
EAFB A7 A5               sac1   sta b,y
EAFD 5A                         decb
EAFE 2C FB                      bge sac1
EB00 86 05                      lda #$05
EB02 37 04               sa     pulu b
EB04 4A                         deca
EB05 81 02                      cmpa #$02
EB07 27 13                      beq sac2
EB09 C1 00                      cmpb #$00
EB0B 27 F5                      beq sa
EB0D CB 30                      addb #$30
EB0F E7 A0                      stb ,y+
EB11 37 04               sac    pulu b
EB13 CB 30                      addb #$30
EB15 E7 A0                      stb ,y+
EB17 4A                         deca
EB18 81 02                      cmpa #$02
EB1A 26 F5                      bne sac
EB1C C6 2E               sac2   ldb #$2e
EB1E E7 A0                      stb ,y+
EB20 37 04               sac3   pulu b
EB22 CB 30                      addb #$30
EB24 E7 A0                      stb ,y+
```

```
EB26 4A                    deca
EB27 26 F7                 bne sac3
EB29 B7 A0 09              sta $a009
EB2C 31 21                 leay 1,y
EB2E 39                    rts

                           *
                           * DESPLIEGA EN LA PANTALLA (despli)
                           *        (mues)
                           *

EB2F B7 A0 08              mues   sta $a008
EB32 A6 80                 muest  lda ,x+
EB34 A7 A0                        sta ,y+
EB36 5A                           decb
EB37 26 F9                        bne muest
EB39 B7 A0 09                     sta $a009
EB3C 39                           rts

                           *
                           * ESCRIBE EL MINIMO F.P. EN LA T.V.
                           *          (fp)
                           *

EB3D B7 A0 08              fp     sta $a008
EB40 86 AA                        lda #$aa
EB42 8E 40 05                     ldx #$4005
EB45 10 8E 99 B8                  ldy #$99b8
EB49 B1 40 1B                     cmpa $401b
EB4C 27 19                        beq tres
EB4E B7 40 1B                     sta $401b
EB51 86 30                        lda #$30
EB53 A7 80                        sta ,x+
EB55 A7 A0                        sta ,y+
EB57 86 2E                        lda #$2e
EB59 A7 A0                        sta ,y+
EB5B 86 38                        lda #$38
EB5D A7 80                        sta ,x+
EB5F A7 A0                        sta ,y+
EB61 86 35                        lda #$35
EB63 A7 80                        sta ,x+
EB65 20 0E                        bra cuatro
EB67 A6 80                 tres   lda ,x+
EB69 A7 A0                        sta ,y+
EB6B 86 2E                        lda #$2e
EB6D A7 A0                        sta ,y+
EB6F A6 80                        lda ,x+
EB71 A7 A0                        sta ,y+
EB73 A6 80                        lda ,x+
EB75 A7 A0                 cuatro sta ,y+
EB77 B7 A0 09                     sta $a009
EB7A 39                           rts

                           *
                           * CALCULA LA POTENCIA  REAL
                           *
```

```
EB7B BD EB 8B          real   jsr poten
EB7E 1E 02                    exg d,y
EB80 FC 40 0A                 ldd $400a
EB83 BD EB D1                 jsr multi
EB86 10 BF 40 1E              sty $401e
EB8A 39                       rts

                       *
                       * CALCULOS PARA OBTENER LA POTENCIA REAL
                       *     (poten)
                       *

EB8B CC 00 00          poten  ldd #$00
EB8E FD 40 11                 std $4011
EB91 CC 00 64                 ldd #$64
EB94 10 BE 40 03              ldy $4003
EB98 BD E2 EB                 jsr div16
EB9B 10 BF 40 0C              sty $400c
EB9F 10 BE 40 01              ldy $4001
EBA3 BD EB D1                 jsr multi
EBA6 CC 00 0A                 ldd #$0a
EBA9 BD E2 EB                 jsr div16
EBAC FD 40 11                 std $4011
EBAF 1E 20                    exg y,d
EBB1 F7 40 13                 stb $4013
EBB4 10 BE 40 01              ldy $4001
EBB8 CC 00 0A                 ldd #$0a
EBBB BD E2 EB                 jsr div16
EBBE B6 40 0C                 lda $400c
EBC1 FB 40 13                 addb $4013
EBC4 3D                       mul
EBC5 1E 02                    exg d,y
EBC7 CC 00 64                 ldd #$64
EBCA BD E2 EB                 jsr div16
EBCD F3 40 11                 addd $4011
EBD0 39                       rts

                       *
                       * MULTIPLICACION ENTRE 8 Y 16 BITS
                       *        (multi)
                       *

EBD1 10 BF 40 37       multi  sty $4037
EBD5 F7 40 10                 stb $4010
EBD8 B6 40 38                 lda $4038
EBDB 3D                       mul
EBDC FD 40 0E                 std $400e
EBDF F6 40 10                 ldb $4010
EBE2 B6 40 37                 lda $4037
EBE5 3D                       mul
EBE6 FB 40 0E                 addb $400e
EBE9 F7 40 0E                 stb $400e
EBEC 10 BE 40 0E              ldy $400e
EBF0 39                       rts
```

```
                        *
                        * CALCULA LA POTENCIA APARENTE
                        *
EBF1 10 BE 40 01        aparente  ldy $4001
EBF5 FC 40 0A                     ldd $400a
EBF8 BD EB D1                     jsr multi
EBFB 10 BF 40 1C                  sty $401c
EBFF 39                           rts

                        *
                        * MANEJO (manejo)
                        *
EC00 BE 40 23           manejo ldx $4023
EC03 10 BE 40 25               ldy $4025
EC07 A7 80                     sta ,x+
EC09 B7 A0 08                  sta $a008
EC0C A7 A0                     sta ,y+
EC0E B7 A0 09                  sta $a009
EC11 BF 40 23                  stx $4023
EC14 10 BF 40 25               sty $4025
EC18 39                        rts

                        *
                        * INICIALIZA LOCALIDADES DE DATOS (pone)
                        *
EC19 10 8E 40 20        pone   ldy #$4020
EC1D C6 30                     ldb #$30
EC1F E7 A0                     stb ,y+
EC21 5C                        incb
EC22 E7 A0                     stb ,y+
EC24 C6 2E                     ldb #$2e
EC26 E7 A0                     stb ,y+
EC28 8E 40 29                  ldx #$4029
EC2B AF A1                     stx ,y++
EC2D 8E 99 D7                  ldx #$99d7
EC30 AF A1                     stx ,y++
EC32 C6 04                     ldb #$4
EC34 E7 A0                     stb ,y+
EC36 E7 A0                     stb ,y+
EC38 39                        rts

                        *
                        * CAMBIA DATOS PARA DEL f.p. (cambia)
                        *
EC39 8E 40 29           cambia ldx #$4029
EC3C 10 8E 40 05               ldy #$4005
EC40 E6 81                     ldb ,x++
EC42 E7 A0                     stb ,y+
EC44 E6 80                     ldb ,x+
EC46 E7 A0                     stb ,y+
EC48 E6 80                     ldb ,x+
EC4A E7 A0                     stb ,y+
```

```
EC4C 39                         rts
                                *
                                * RUTINA DE INTERRUPCION PARA EL F.P.
                                * ACTIVA O NO LOS CAPACITORES
                                *

EC4D 86 01                int   lda #$01        ;deshabilita al
EC4F B7 A0 10                   sta $a010       ;ptm
EC52 B6 A0 16                   lda $a016       ;limpia al ptm
EC55 86 55                      lda #$55
EC57 B1 40 19                   cmpa $4019
EC5A 26 21                      bne sale1
EC5C BD E6 00                   jsr lefp
EC5F FD 40 03                   std $4003
EC62 10 B3 40 08                cmpd $4008
EC66 25 0F                      blo sale5
EC68 B7 A0 08                   sta $a008
EC6B B6 98 1B                   lda $981b
EC6E B7 A0 09                   sta $a009
EC71 81 20                      cmpa #$20
EC73 10 27 00 74                lbeq todos
EC77 BD ED 01             sale5 jsr aviso2
EC7A 16 00 6E                   lbra todos
EC7D 86 AA                sale1 lda #$aa
EC7F B1 40 19                   cmpa $4019
EC82 27 07                      beq inte
EC84 86 5A                      lda #$5a
EC86 B1 40 19                   cmpa $4019
EC89 26 70                      bne sale
EC8B BD E6 00             inte  jsr lefp
EC8E 10 B3 40 03                cmpd $4003
EC92 27 09                      beq inte2
EC94 FD 40 03                   std $4003
EC97 10 B3 40 08                cmpd $4008
EC9B 2B 12                      bmi todo
EC9D BD E7 22             inte2 jsr ajvo
ECA0 F1 40 01                   cmpb $4001
ECA3 26 46                      bne todos
ECA5 BD E7 40                   jsr ajco
ECA8 F1 40 0A                   cmpb $400a
ECAB 27 4E                      beq sale
ECAD 20 3C                      bra todos
ECAF 86 AA                todo  lda #$aa
ECB1 B1 40 36                   cmpa $4036
ECB4 27 05                      beq tod
ECB6 B7 40 36                   sta $4036
ECB9 20 30                      bra todos
ECBB F6 40 16             tod   ldb $4016
ECBE B6 5F F4                   lda ang
ECC1 26 0A                      bne todo1
ECC3 C1 F0                      cmpb #$f0
ECC5 26 02                      bne todo2
ECC7 20 08                      bra todos5
ECC9 CB 10                todo2 addb #$10
ECCB 20 18                      bra todo3
ECCD C1 00                todo1 cmpb #$00
```

```
ECCF 27 0D              beq sale6
ECD1 B7 A0 08       todos5 sta $a008
ECD4 B6 98 02              lda $9802
ECD7 B7 A0 09              sta $a009
ECDA 81 20                 cmpa #$20
ECDC 27 05                 beq todo4
ECDE BD ED 20       sale6  jsr aviso
ECE1 20 08                 bra todos
ECE3 C0 10          todo4  subb #$10
ECE5 F7 40 16       todo3  stb $4016
ECE8 F7 A0 00              stb piabfa
ECEB BD EA 2B       todos  jsr lecturas
ECEE B6 40 19              lda $4019
ECF1 81 5A                 cmpa #$5a
ECF3 27 06                 beq sale
ECF5 BD EA C0              jsr fpsp
ECF8 BD EA 55              jsr des
ECFB 86 00          sale   lda #$00       ;habilita al
ECFD B7 A0 10              sta $a010      ;ptm
ED00 3B                    rti

                    *
                    * SUBRUTINA DE AVISO DE CAMBIO DE F.P.
                    *          (aviso2)
                    *

ED01 B7 A0 08       aviso2 sta $a008
ED04 10 8E 98 1B           ldy #$981b
ED08 86 20                 lda #$20
ED0A C6 04                 ldb #$04
ED0C A1 A4                 cmpa ,y
ED0E 26 07                 bne avi
ED10 8E EE B5              ldx #vii
ED13 BD EB 2F              jsr mues
ED16 39                    rts
ED17 A7 A0          avi    sta ,y+
ED19 5A                    decb
ED1A 26 FB                 bne avi
ED1C B7 A0 09              sta $a009
ED1F 39                    rts

                    *
                    * SUBRUTINA QUE INDICA QUE NO ES POSIBLE
                    * OBTENER EL F.P. DESEADO
                    *      (aviso)
                    *

ED20 B7 A0 08       aviso  sta $a008
ED23 10 8E 98 02           ldy #$9802
ED27 86 20                 lda #$20
ED29 A1 A4                 cmpa ,y
ED2B 26 0F                 bne av
ED2D 8E EE B5              ldx #vii
ED30 C6 04                 ldb #$04
ED32 BD EB 2F              jsr mues
ED35 8E EF 98              ldx #viii
```

```
ED38 BD EA 3E                jsr despli
ED3B 39                      rts
ED3C C6 0F             av    ldb #$0f
ED3E A7 A0             av1   sta ,y+
ED40 5A                      decb
ED41 26 FB                   bne av1
ED43 B7 A0 09                sta $a009
ED46 39                      rts
```

APENDICE

D

HOJAS TECNICAS

Apéndice D

D. Hojas Técnicas.

MOTOROLA

MC6809

8-BIT MICROPROCESSING UNIT

The MC6809 is a revolutionary high performance 8-bit microprocessor which supports modern programming techniques such as position independence, reentrancy, and modular programming.

This third-generation addition to the M6800 Family has major architectural improvements which include additional registers, instructions, and addressing modes.

The basic instructions of any computer are greatly enhanced by the presence of powerful addressing modes. The MC6809 has the most complete set of addressing modes available on any 8-bit microprocessor today.

The MC6809 has hardware and software features which make it an ideal processor for higher level language execution or standard controller applications.

M6800 COMPATIBLE

- Hardware — Interfaces with All M6800 Peripherals
- Software — Upward Source Code Compatible Instruction Set and Addressing Modes

ARCHITECTURAL FEATURES

- Two 16-Bit Index Registers
- Two 16-Bit Indexable Stack Pointers
- Two 8-Bit Accumulators can be Concatenated to Form One 16-Bit Accumulator
- Direct Page Register Allows Direct Addressing Throughout Memory

HARDWARE FEATURES

- On-Chip Oscillator (Crystal Frequency = 4 x E)
- DMA/BREQ Allows DMA Operation on Memory Refresh
- Fast Interrupt Request Input Stacks Only Condition Code Register and Program Counter
- MRDY Input Extends Data Access Times for Use with Slow Memory
- Interrupt Acknowledge Output Allows Vectoring by Devices
- Sync Acknowledge Output Allows for Synchronization to External Event
- Single Bus Cycle RESET
- Single 5-Volt Supply Operation
- NMI Inhibited After RESET Until After First Load of Stack Pointer
- Early Address Valid Allows Use with Slower Memories
- Early Write Data for Dynamic Memories

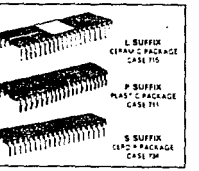
SOFTWARE FEATURES

- 10 Addressing Modes
  - 6800 Upward Compatible Addressing Modes
  - Direct Addressing Anywhere in Memory Map
  - Long Relative Branches
  - Program Counter Relative
  - True Indirect Addressing
  - Expanded Indexed Addressing
    - 0-, 5-, 8-, or 16-Bit Constant Offsets
    - 8- or 16-Bit Accumulator Offsets
    - Auto Increment-Decrement by 1 or 2
- Improved Stack Manipulation
- 1464 Instructions with Unique Addressing Modes
- 8 x 8 Unsigned Multiply
- 16-Bit Arithmetic
- Transfer/Exchange All Registers
- Push/Pull Any Registers or Any Set of Registers
- Load Effective Address

HMOS

(HIGH DENSITY N-CHANNEL, SILICON-GATE)

8-BIT MICROPROCESSING UNIT

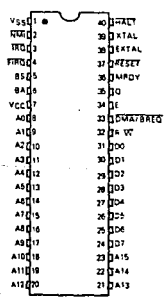


L SUFFIX
CERAMIC PACKAGE
CASE 715

P SUFFIX
PLASTIC PACKAGE
CASE 711

S SUFFIX
CERDIP PACKAGE
CASE 734

PIN ASSIGNMENT



MOTOROLA

MC6821

MOS

(N-CHANNEL, SILICON-GATE, DEPLETION LOAD)

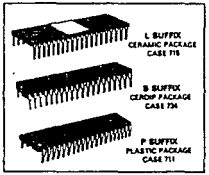
PERIPHERAL INTERFACE ADAPTER

PERIPHERAL INTERFACE ADAPTER (PIA)

The MC6821 Peripheral Interface Adapter provides the universal means of interfacing peripheral equipment to the M6800 family of microprocessors. This device is capable of interfacing the MPU to peripherals through two 8-bit bidirectional peripheral data buses and four control lines. No external logic is required for interfacing to most peripheral devices.

The functional configuration of the PIA is programmed by the MPU during system initialization. Each of the peripheral data lines can be programmed to act as an input or output, and each of the four control/interrupt lines may be programmed for one of several control modes. This allows a high degree of flexibility in the overall operation of the interface.

- 8-Bit Bidirectional Data Bus for Communication with the MPU
- Two Bidirectional 8-Bit Buses for Interface to Peripherals
- Two Programmable Control Registers
- Two Programmable Data Direction Registers
- Four Individually-Controlled Interrupt Input Lines, Two Usable as Peripheral Control Outputs
- Handshake Control Logic for Input and Output Peripheral Operation
- High-Impedance Three-State and Direct Transistor Drive Peripheral Lines
- Program Controlled Interrupt and Interrupt Disable Capability
- CMOS Drive Capability on Side A Peripheral Lines
- Two TTL Drive Capability on All A and B Side Buffers
- TTL-Compatible
- Static Operation



L BUFFIX
CERAMIC PACKAGE
CASE 715

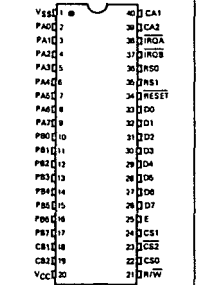
S BUFFIX
CERDIP PACKAGE
CASE 734

P BUFFIX
PLASTIC PACKAGE
CASE 711

ORDERING INFORMATION

Package Type	Frequency (MHz)	Temperature	Order Number
Ceramic L Suffix	1.0	0°C to 70°C	MC6821L
	1.0	-40°C to 85°C	MC6821CL
	1.5	0°C to 70°C	MC68A21L
	1.5	-40°C to 85°C	MC68A21CL
	2.0	0°C to 70°C	MC68B21L
Cerdip S Suffix	1.0	0°C to 70°C	MC6821S
	1.0	-40°C to 85°C	MC6821CS
	1.5	0°C to 70°C	MC68A21S
	1.5	-40°C to 85°C	MC68A21CS
	2.0	0°C to 70°C	MC68B21S
Plastic P Suffix	1.0	0°C to 70°C	MC6821P
	1.0	-40°C to 85°C	MC6821CP
	1.5	0°C to 70°C	MC68A21P
	1.5	-40°C to 85°C	MC68A21CP
	2.0	0°C to 70°C	MC68B21P

PIN ASSIGNMENT



MOTOROLA

MC6840

PROGRAMMABLE TIMER MODULE (PTM)

The MC6840 is a programmable subsystem component of the M6800 family designed to provide variable system time intervals.

The MC6840 has three 16-bit binary counters, three corresponding control registers, and a status register. These counters are under software control and may be used to cause system interrupts and/or generate output signals. The MC6840 may be utilized for such tasks as frequency measurements, event counting, interval measuring, and similar tasks. The device may be used for square wave generation, gated delay signals, single pulses of controlled duration, and pulse width modulation as well as system interrupts.

- Operates from a Single 5 Volt Power Supply
- Fully TTL Compatible
- Single System Clock Required (Enable)
- Selectable Prescaler on Timer 3 Capable of 4 MHz for the MC6840, 8 MHz for the MC68A40 and 8 MHz for the MC68B40
- Programmable Interrupts ($\overline{IRQ}$) Output to MPU
- Readable Down Counter Indicates Counts to Go Until Time-Out
- Selectable Gating for Frequency or Pulse-Width Comparison
- $\overline{RESET}$ Input
- Three Asynchronous External Clock and Gate/Trigger Inputs Internally Synchronized
- Three Maskable Outputs

ORDERING INFORMATION

Package Type	Frequency	Temperature Range	Order Number
Ceramic L Suffix	1.0 MHz	0°C to 70°C	MC6840L
	1.0 MHz	−40°C to +85°C	MC68A40CL
	1.5 MHz	0°C to 70°C	MC68A40L
	1.5 MHz	−40°C to +85°C	MC68A40CL
	2.0 MHz	0°C to 70°C	MC68B40L
Plastic P Suffix	1.0 MHz	0°C to 70°C	MC6840P
	1.0 MHz	−40°C to +85°C	MC68A40CP
	1.5 MHz	0°C to 70°C	MC68A40P
	1.5 MHz	−40°C to +85°C	MC68A40CP
	2.0 MHz	0°C to 70°C	MC68B40P
Cerdip S Suffix	1.0 MHz	0°C to 70°C	MC6840S
	1.0 MHz	−40°C to +85°C	MC68A40CS
	1.5 MHz	0°C to 70°C	MC68A40S
	1.5 MHz	−40°C to +85°C	MC68A40CS
	2.0 MHz	0°C to 70°C	MC68B40S

MOS

N-CHANNEL, SILICON-GATE DEPLETION LOAD

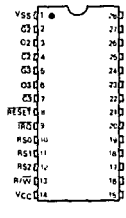
PROGRAMMABLE TIMER

L SUFFIX
CERAMIC PACKAGE
CASE 716

P SUFFIX
PLASTIC PACKAGE
CASE 710

S SUFFIX
CERDIP PACKAGE
CASE 733

PIN ASSIGNMENT



MOTOROLA

MC6847
Non-Interlace
MC6847Y
Interlace

MOS
(N-CHANNEL, SILICON-GATE)

VIDEO DISPLAY GENERATOR

MC6847/MC6847Y VIDEO DISPLAY GENERATOR (VDG)

The video display generator (VDG) provides a means of interfacing the M6800 microprocessor family (or similar products) to a standard color or black and white NTSC television receiver. Applications of the VDG include video games, process control displays, home computers, education, communications, and graphics applications.

The VDG reads data from memory and produces a video signal which will allow the generation of alphanumeric or graphic displays. The generated video signal may be modulated to either channel 3 or 4 by using the compatible MC1372 (TV chroma and video modulator). This modulated signal is suitable for reception by a standard unmodified television receiver. A typical TV game is shown in Figure 1.

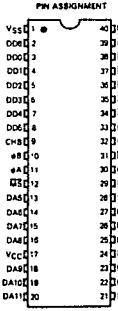
- Compatible with the M6800 Family, the M68000 Family, and Other Microprocessor Families
- Generates Four Different Alphanumeric Display Modes, Two Semigraphic Modes, and Eight Graphic Display Modes
- The Alphanumeric Modes Display 32 Characters Per Line by 16 Lines Using Either the Internal ROM or an External Character Generator
- Alphanumeric and Semigraphic Modes May Be Mixed on a Character-by-Character Basis
- Alphanumeric Modes Support Selectable Inverse on a Character-by-Character Basis
- Internal ROM May Be Mask Programmed with a Custom Pattern
- Full Graphic Modes Offer 64 × 64, 128 × 64, 128 × 96, 128 × 192, or 256 × 192 Densities
- Full Graphic Modes Use One of Two 4-Color Sets or One of Two 2-Color Sets
- Compatible with the MC1372 and MC1373 Modulators Via Y, R-Y (eA), and B-Y (eB) Interface
- Compatible with the MC6883 (74LS783) Synchronous-Address Multiplexer
- Available in Either an Interface (NTSC Standard) or Non-Interface Version

L SUFFIX
CERAMIC PACKAGE
CASE 716

P SUFFIX
PLASTIC PACKAGE
CASE 711

S SUFFIX
CERDIP PACKAGE
CASE 734

PIN ASSIGNMENT



MOTOROLA

MC6850

ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER (ACIA)

The MC6850 Asynchronous Communications Interface Adapter provides the data formatting and control to interface serial asynchronous data communications information to bus organized systems such as the MC6800 Microprocessing Unit.

The bus interface of the MC6850 includes select, enable, read/write, interrupt and bus interface logic to allow data transfer over an 8-bit bidirectional data bus. The parallel data of the bus system is serially transmitted and received by the asynchronous data interface, with proper formatting and error checking. The functional configuration of the ACIA is programmed via the data bus during system initialization. A programmable Control Register provides variable word lengths, clock division ratios, transmit control, receive control, and interrupt control. For peripheral or modem operation, three control lines are provided. These lines allow the ACIA to interface directly with the MC6860L 0-600 bps digital modem.

- 8- and 9-Bit Transmission
- Optional Even and Odd Parity
- Parity, Overrun and Framing Error Checking
- Programmable Control Register
- Optional ÷1, ÷16, and ÷64 Clock Modes
- Up to 1.0 Mbps Transmission
- False Start Bit Deletion
- Peripheral/Modem Control Functions
- Double Buffered
- One- or Two-Stop Bit Operation

MOS
(N-CHANNEL, SILICON-GATE)

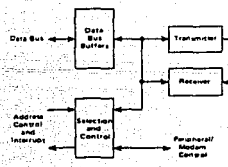
ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER

S SUFFIX
CERDIP PACKAGE
CASE 623

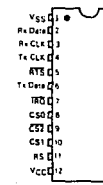
P SUFFIX
PLASTIC PACKAGE
CASE 709

L SUFFIX
CERAMIC PACKAGE
CASE 716

MC6850 ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER BLOCK DIAGRAM



PIN ASSIGNMENT



MOTOROLA

MC1372

COLOR TV VIDEO MODULATOR

. . . an integrated circuit used to generate an RF TV signal from baseband color-difference and luminance signals.

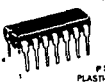
The MC1372 contains a chroma subcarrier oscillator, a lead and lag network, a quasi-quadrature suppressed carrier DSB chroma modulator, an RF oscillator and modulator, and an LSTTL compatible clock driver with adjustable duty cycle.

The MC1372 is a companion part to the MC6847 Video Display Generator, providing and accepting the correct dc interconnection levels. This device may also be used as a general purpose modulator with a variety of video signal generating devices such as video games, test equipment, video tape recorders, etc.

- Single 5.0 Vdc Supply Operation for NMOS and TTL Compatibility
- Minimal External Components
- Compatible with MC6847 Video Display Generator
- Sound Carrier Addition Capability
- Modulates Channel 3 or 4 Carrier with Encoded Video Signal
- Low Power Dissipation
- Linear Chroma Modulators for High Versatility
- Composite Video Signal Generation Capability
- Ground Referenced Video Prevents Overmodulation

COLOR TV VIDEO MODULATOR CIRCUIT

SILICON MONOLITHIC INTEGRATED CIRCUIT



P SUFFIX
PLASTIC PACKAGE
CASE 646-05

Pin Connections

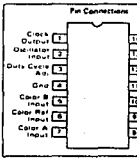
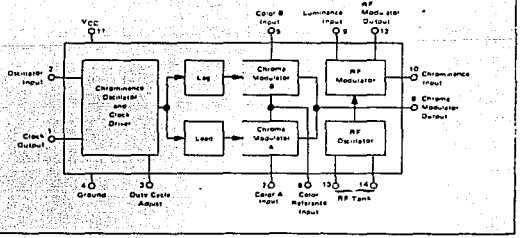


FIGURE 1 – BLOCK DIAGRAM



MOTOROLA

SN54LS245
SN74LS245

OCTAL BUS TRANSCEIVER

LOW POWER SCHOTTKY

DESCRIPTION — The SN54LS/74LS245 is an Octal Bus Transmitter/Receiver designed for 8-line asynchronous 2-way data communication between data buses. Direction Input (DIR) controls transmission of Data from bus A to bus B or bus B to bus A depending upon its logic level. The Enable input (Ē) can be used to isolate the buses.

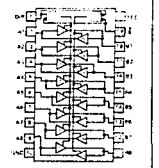
- HYSTERESIS INPUTS TO IMPROVE NOISE IMMUNITY
- 2-WAY ASYNCHRONOUS DATA BUS COMMUNICATION
- INPUT DIODES LIMIT HIGH-SPEED TERMINATION EFFECTS

TRUTH TABLE

INPUTS		OUTPUT
Ē	DIR	
L	L	Bus B Data to Bus A
L	H	Bus A Data to Bus B
H	X	Isolation

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

LOGIC AND CONNECTION DIAGRAM
DIP (TOP VIEW)



J Suffix — Case 732-03 (Ceramic)
N Suffix — Case 738-01 (Plastic)

5

MOTOROLA

SN54LS/74LS240
SN54LS/74LS241
SN54LS/74LS244

OCTAL BUFFER/LINE DRIVER WITH 3-STATE OUTPUTS

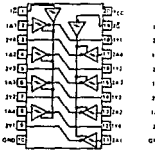
LOW POWER SCHOTTKY

DESCRIPTION — The SN54LS/74LS240, 241 and 244 are Octal Buffers and Line Drivers designed to be employed as memory address drivers, clock drivers and bus-oriented transmitters/receivers which provide improved PC board density.

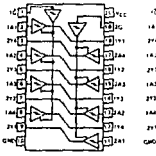
- HYSTERESIS AT INPUTS TO IMPROVE NOISE MARGINS
- 3-STATE OUTPUTS DRIVE BUS LINES OR BUFFER MEMORY ADDRESS REGISTERS
- INPUT CLAMP DIODES LIMIT HIGH-SPEED TERMINATION EFFECTS

LOGIC AND CONNECTION DIAGRAMS DIP (TOP VIEW)

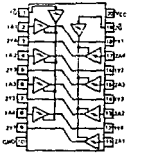
SN54LS/74LS240



SN54LS/74LS241



SN54LS/74LS244



TRUTH TABLES

SN54LS/74LS240

INPUTS		OUTPUT
$\overline{1G}$, $\overline{2G}$	D	
L	L	H
L	H	L
H	X	(Z)

SN54LS/74LS244

INPUTS		OUTPUT
$\overline{1G}$, $\overline{2G}$	D	
L	L	L
L	H	H
H	X	(Z)

SN54LS/74LS241

INPUTS		OUTPUT
$\overline{1G}$	D	
L	L	L
L	H	H
H	X	(Z)

INPUTS		OUTPUT
2G	D	
H	L	L
H	H	H
L	X	(Z)

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial
Z = HIGH Impedance

J Suffix — Case 732-03 (Ceramic)
N Suffix — Case 738-01 (Plastic)

MOTOROLA

4N38
4N38A

OPTICAL COUPLERS WITH NPN TRANSISTOR OUTPUT

. . . gallium-arsenide LED optically coupled to a silicon photo-transistor. Designed for applications requiring electrical isolation, high breakdown voltage and low leakage such as teletypewriter interfacing, telephone line pulsing and driving high-voltage relays.

- High Isolation Voltage — V_{ISO} = 7500 V (Min)
- High Collector Emitter Breakdown Voltage — $V_{(BR)CEO}$ = 80 V (Min)
- Economical Dual-in-Line Package
- 4N38A UL Recognized, File Number E54915

OPTO COUPLER/ISOLATOR

TRANSISTOR OUTPUT

*MAXIMUM RATINGS (T_A = 25°C unless otherwise noted)

Rating	Symbol	Value	Unit
INFRARED EMITTING DIODE MAXIMUM RATINGS			
Reverse Voltage	V_R	3.0	Volts
Forward Current — Continuous	I_F	80	mA
Forward Current — Peak (Pulse Width = 300 μs, 2.0% Duty Cycle)	I_F	3.0	Amp
Total Device Dissipation @ T_A = 25°C; Negligible Power in Transistor; Derate above 25°C	P_D	150 2.0	mW mW/°C
PHOTOTRANSISTOR MAXIMUM RATINGS			
Collector Emitter Voltage	V_{CEO}	80	Volts
Emitter Collector Voltage	V_{ECO}	7.0	Volts
Collector Base Voltage	V_{CBO}	80	Volts
Total Device Dissipation @ T_A = 25°C; Negligible Power in Diode; Derate above 25°C	P_D	150 2.0	mW mW/°C
TOTAL DEVICE RATINGS			
Total Device Dissipation @ T_A = 25°C; Equal Power Dissipation in Each Element; Derate above 25°C	P_D	250 3.3	mW mW/°C
Junction Temperature Range	T_J	-55 to +100	°C
Storage Temperature Range	T_{STG}	-55 to +150	°C
Soldering Temperature (10 s)		260	°C

*Indicates JEDEC Registered Data

FIGURE 1 – MAXIMUM POWER DISSIPATION

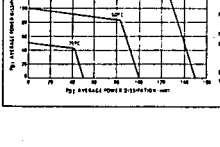


Figure 1 is based upon using the following formula:

$T_{J1} = T_A + R_{\theta JA}(P_{D1} + K_\theta P_{D2})$

where

T_{J1} = Junction Temperature (100°C)

T_A = Ambient Temperature

$R_{\theta JA}$ = Junction to Ambient Thermal Resistance (500°C/W)

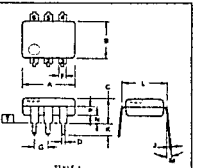
P_{D1} = Power Dissipation in One Chip

P_{D2} = Power Dissipation in Other Chip

K_θ = Thermal Coupling Coefficient (20%)

Example

With P_{D1} = 80 mW in the LED @ T_A = 50°C, the transistor P_D (P_{D2}) must be less than 50 mW.



STYLE 1
PIN 1 ANODE
2 CATHODE
3 NC
4 EMITTER
5 COLLECTOR
6 BASE

NOTES
1. DIMENSIONS A AND B ARE DATUMS.
2. T IS SEATING PLANE.
3. POSITIONAL TOLERANCES FOR LEADS:
4. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
5. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	9.15	9.65	0.360	0.380
B	6.10	6.60	0.240	0.260
C	3.94	5.08	0.155	0.200
D	0.41	0.51	0.016	0.020
F	1.07	1.23	0.042	0.048
G	2.54 BSC		0.100 BSC	
J	0.19	0.36	0.008	0.014
K	2.92	3.81	0.100	0.150
L	7.62 BSC		0.300 BSC	
M	0°	15°	0°	15°
N	0.76	1.14	0.030	0.045
P	7.37	7.87	0.290	0.310

CASE 730A-01

MOTOROLA

MOC3040
MOC3041

ZERO VOLTAGE CROSSING OPTICALLY ISOLATED TRIAC DRIVERS

These devices consist of gallium-arsenide infrared-emitting diodes optically coupled to monolithic silicon detectors performing the functions of Zero Voltage Crossing bilateral triac drivers.

They are designed for use with a triac in the interface of logic systems to equipment powered from 220 Vac lines, such as solid-state relays, industrial controls, motors, solenoids and consumer appliances, etc.

- Simplifies Logic Control of 220 Vac Power
- Zero Voltage Crossing
- High Breakdown Voltage: V_{DRM} = 400 V Min
- High Isolation Voltage: V_{ISO} = 7500 Vac (Min)
- Small, Economical, 6-Pin DIP Package
- Same Pin Configuration as MOC3020/3021
- UL Recognized, File No. E54915
- dv/dt of 100 V/µs Typ

OPTO COUPLER / ISOLATOR ZERO CROSSING TRIAC DRIVER

400 VOLTS

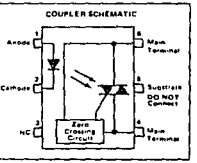
CASE 730A-01

MAXIMUM RATINGS (T_A = 25°C unless otherwise noted)

Rating	Symbol	Value	Unit
INFRARED EMITTING DIODE MAXIMUM RATINGS			
Reverse Voltage	V_R	6.0	Volts
Forward Current — Continuous	I_F	50	mA
Total Power Dissipation @ T_A = 25°C Negligible Power in Output Driver Derate above 25°C	P_D	120 1.33	mW mW/°C
OUTPUT DRIVER MAXIMUM RATINGS			
Off-State Output Terminal Voltage	V_{DRM}	400	Volts
On-State RMS Current T_A = 25°C (Full Cycle, 50 to 60 Hz) T_A = 70°C	$I_{T(RMS)}$	100 50	mA mA
Peak Nonrepetitive Surge Current (PW = 10 ms)	I_{TSM}	1.2	A
Total Power Dissipation @ T_A = 25°C Derate above 25°C	P_D	300 4.0	mW mW/°C
TOTAL DEVICE MAXIMUM RATINGS			
Isolation Surge Voltage (1) (Peak ac Voltage, 60 Hz, 5 Second Duration)	V_{ISO}	7500	Vac
Total Power Dissipation @ T_A = 25°C Derate above 25°C	P_D	330 4.4	mW mW/°C
Junction Temperature Range	T_J	-40 to +100	°C
Ambient Operating Temperature Range	T_A	-40 to +70	°C
Storage Temperature Range	T_{stg}	-40 to +150	°C
Soldering Temperature (10 s)	-	260	°C

(1) Isolation Surge Voltage, V_{ISO}, is an internal device dielectric breakdown rating.

COUPLER SCHEMATIC



MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

NPN	PNP
TIP120	TIP125
TIP121	TIP126
TIP122	TIP127

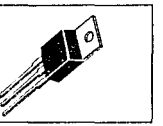
PLASTIC MEDIUM-POWER COMPLEMENTARY SILICON TRANSISTORS

... designed for general-purpose amplifier and low-speed switching applications.

- High DC Current Gain — h_{FE} = 2500 (Typ) @ I_C = 4.0 Adc
- Collector-Emitter Sustaining Voltage — @ 100 mAdc
  $V_{CEO(sus)}$ = 60 Vdc (Min) — TIP120, TIP125
  = 80 Vdc (Min) — TIP121, TIP126
  = 100 Vdc (Min) — TIP122, TIP127
- Low Collector-Emitter Saturation Voltage —
  $V_{CE(sat)}$ = 2.0 Vdc (Max) @ I_C = 3.0 Adc
  = 4.0 Vdc (Max) @ I_C = 5.0 Adc
- Monolithic Construction with Built-In Base-Emitter Shunt Resistors
- TO-220AB Compact Package
- TO-66 Leadform Also Available

DARLINGTON 5 AMPERE COMPLEMENTARY SILICON POWER TRANSISTORS

60-80-100 VOLTS
65 WATTS



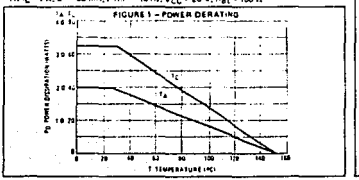
*MAXIMUM RATINGS

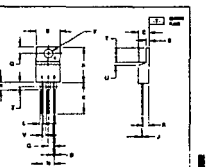
Rating	Symbol	TIP120, TIP125	TIP121, TIP126	TIP122, TIP127	Unit
Collector-Emitter Voltage	V_{CEO}	60	80	100	Vdc
Collector-Base Voltage	V_{CB}	60	80	100	Vdc
Emitter-Base Voltage	V_{EB}	5.0			Vdc
Collector Current — Continuous	I_C	5.0			Adc
Peak		8.0			
Base Current	I_B	120			mAdc
Total Power Dissipation @ T_C = 25°C	P_D	65			Watts
Derate above 25°C		0.52			W/°C
Total Power Dissipation @ T_A = 25°C	P_D	2.0			Watts
Derate above 25°C		0.016			W/°C
Unclamped Inductive Load Energy (1)	E	50			mJ
Operating and Storage Junction Temperature Range	T_J, T_{stg}	−65 to +150			°C

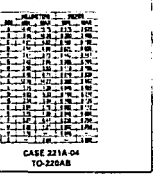
THERMAL CHARACTERISTICS

Characteristics	Symbol	Max	Unit
Thermal Resistance, Junction to Case	$R_{\theta JC}$	1.92	°C/W
Thermal Resistance, Junction to Ambient	$R_{\theta JA}$	62.5	°C/W

(1) I_C = 1 A, L = 30 mH, P.R.F. = 10 Hz, V_{CC} = 20 V, $I_{B(pk)}$ = 100 mA.







CASE 221A-04
TO-220AB

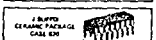
MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

SG1525A/SG1527A
SG2525A/SG2527A
SG3525A/SG3527A

PULSE WIDTH MODULATOR CONTROL CIRCUITS

SILICON MONOLITHIC INTEGRATED CIRCUITS

J SUFFIX
CERAMIC PACKAGE
CASE 620



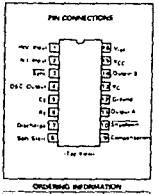
N SUFFIX
PLASTIC PACKAGE
CASE 648

PULSE WIDTH MODULATOR CONTROL CIRCUITS

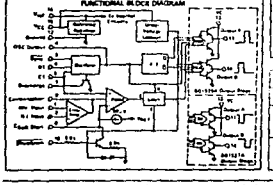
The SG1525A/1527A series of pulse width modulator control circuits offer improved performance and lower external parts count when implemented by combining all types of switching power supplies. The on-chip +5.1 volt reference is trimmed to ± 1% and the error amplifier has an input common mode voltage range that includes the reference voltage, thus eliminating the need for external divider resistors. A sync input to the oscillator enables multiple units to be slaved or a single unit to be synchronized to an external system clock. A single resistor connected between the CT and Discharge pins provides a wide range of dead time adjustment. These devices also feature built-in soft-start circuitry, requiring only an external timing capacitor. A shutdown pin controls both the soft-start circuitry and the output stages, providing instantaneous turn-off through the PWM latch with pulse-by-pulse shutdown, as well as soft-start recycle with longer shutdown commands. This under-voltage lockout inhibits the output and the charging of the soft-start capacitor when VCC is below nominal. The output stages are totem-pole design capable of sourcing and sinking in excess of 200 mA. The output stage of the SG1525A series features NOR logic, resulting in a low output for an off-state, while the SG1527A series utilizes OR logic, which gives a high output when off. The SG1525A is available in military, industrial and commercial temperature ranges.

- 8.0 to 35 Volt Operation
- 5.1 Volt ± 1.0% Trimmed Reference
- 100 Hz to 400 kHz Oscillator Range
- Separate Oscillator Sync Pin
- Adjustable Dead Time Control
- Input Undervoltage Lockout
- Latching PWM to Prevent Multiple Pulses
- Pulse-by-Pulse Shutdown
- Dual Source/Sink Outputs: ± 400 mA Peak

PIN CONNECTIONS



FUNCTIONAL BLOCK DIAGRAM



ORDERING INFORMATION

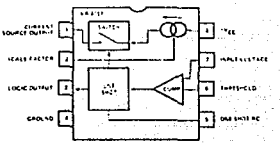
Device	Temperature Range	Package
SG1525AJ SG1527AJ	−55 to +150°C	Ceramic DIP Ceramic DIP
SG2525AJ SG2525AN SG2527AJ SG2527AN	−25 to +85°C	Ceramic DIP Plastic DIP Ceramic DIP Plastic DIP
SG3525AJ SG3525AN SG3527AJ SG3527AN	0 to +70°C	Ceramic DIP Plastic DIP Ceramic DIP Plastic DIP

Voltage-to-Frequency Converter

GENERAL DESCRIPTION

The XR-4151 is a device designed to provide a simple, low-cost method for converting a DC voltage into a proportional pulse repetition frequency. It is also capable of converting an input frequency into a proportional output voltage. The XR-4151 is useful in a wide range of applications including A/D and D/A conversion and data transmission

FUNCTIONAL BLOCK DIAGRAM



FEATURES

Single Supply Operation (+8V to −22V)
Pulse Output Compatible With All Logic Forms
Programmable Scale Factor (K)
Linearity ±0.05% Typical–Precision Mode
Temperature Stability ± 100% ppm/°C Typical
High Noise Rejection
Inherent Monotonicity
Easily Transmittable Output
Simple Full Scale Trim
Single-Ended Input, Referenced to Ground
Also Provides Frequency-to-Voltage Conversion
Direct Replacement for RC/RV/RM-4151

ORDERING INFORMATION

Part Number	Package	Operating Temperature
XR-4151P	Plastic	−40°C to +85°C
XR-4151CP	Plastic	0°C to +70°C

8

APPLICATIONS

Voltage-to-Frequency Conversion
A/D and D/A Conversion
Data Transmission
Frequency-to-Voltage Conversion
Transducer Interface
System Isolation

ABSOLUTE MAXIMUM RATINGS

Power Supply	22V
Output Sink Current	20 mA
Internal Power Dissipation	500 mW
Input Voltage	−0.2V to +V_{CC}
Output Short Circuit to Ground	Continuous

SYSTEM DESCRIPTION

The XR-4151 is a precision voltage to frequency convertor featuring 0.05% conversion linearity, high noise rejection, monotonicity, and single supply operation from 8V to 22V. An RC network on Pin 5 sets the maximum full scale frequency. Input voltage on Pin 7 is compared with the voltage on Pin 6 (which is generally controlled by the current source output, Pin 1). Frequency output is proportioned to the voltage on Pin 7. The current source is controlled by the resistance on Pin 2 (nominally 14kΩ) with I = 1.9 V/R. The output is an open collector at Pin 3.

Frequency-to-Voltage Converter

GENERAL DESCRIPTION

The XR-2917 Frequency-to-Voltage Converter is a high accuracy converter consisting of input comparator with 40 mV hysteresis, charge pump, Zener regulator, and out put op amp and transistor. Designed for tachometer and motor control applications, it features excellent linearity and high current output.

Output voltage is a simple function of the Zener regulator voltage (V_Z), a resistor (R_1) and capacitor (C_1) which are connected to the charge pump, and the input frequency (f_{in}). Ripple reduction is implemented by addition of one capacitor (C_2) which is used to achieve frequency doubling. The output transistor can swing to ground, sink a load current of 40 mA, and offers a maximum V_{CE} of 28 V. Stable and accurate frequency to voltage or current conversion is ensured by the on-chip Zener regulator which is connected across the power leads. The Zener may be used with any supply voltage (up to 28 V) when a suitable resistor is connected between the Zener and the supply.

The XR-2917 may be operated with a ground referenced input or differential tachometer input with uncommitted op amp inputs. The ground referenced configuration is most basic, allowing the realization of single speed, frequency switching, and buffered frequency-to-voltage or current conversion applications. Differential input configurations allow the tachometer to be floated, while uncommitted op amp inputs free the op amp for implementation of active filter conditioning of the tachometer output.

The XR-2917, available in a 14 Pin DIP, operates from a single power supply of up to 28 V.

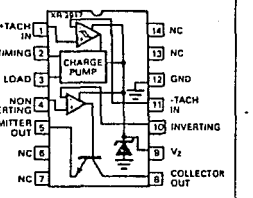
FEATURES

- Design Simplicity: $V_{OUT} = f_{in} \times V_Z \times R_1 \times C_1$
- Frequency Doubling to Decrease Output Ripple
- On Chip Zener for Functional Stability
- Excellent Linearity
- Floating Output Drive Transistor Provides 40 mA Source or Sink
- Ground Referenced Tachometer Input Which Interfaces Directly with Variable Reluctance Magnetic Pickups

ORDERING INFORMATION

Part Number	Package	Operating Temperature
XR-2917CN	Ceramic	0°C to +70°C
XR-2917CP	Plastic	0°C to +70°C

FUNCTIONAL BLOCK DIAGRAM



SYSTEM DESCRIPTION

The XR-2917 converts an input frequency to a proportional output voltage. Differential inputs provide hysteresis for excellent noise rejection and the capability of setting the comparator's input switching level. Inputs should not be taken below ground without some lead resistance.

The output of the comparator is fed into a charge pump where current is pumped through a timing capacitor (C_1). This same current is mirrored in the load resistor (R_1) where a filter capacitor (C_2) may be used to integrate current pulses and provide a proportional voltage across the load resistor. The result is a voltage across the load resistor which is a function of the supply voltage, input frequency, timing capacitor, and load resistor.

$$V_{R_1} = V_Z \times f_{in} \times C_1 \times R_1$$

The size of the integrating capacitor (C_2) is dependent only on the requirements of response time and output ripple.

The output op amp and transistor are then used to buffer the output drive capability of the part. Thus, the final conversion equation is:

$$V_O = V_Z \times f_{in} \times C_1 \times R_1 \times K$$

where K is the gain provided by the tachometer section and is typically unity.