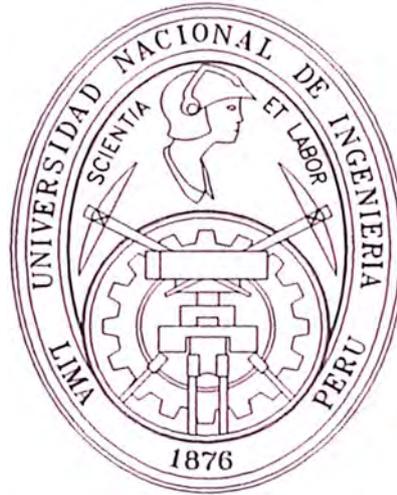


# Universidad Nacional de Ingeniería

FACULTAD DE INGENIERIA ELECTRICA Y ELECTRONICA



## "Diseño y Construcción de un Estabilizador de Voltaje Usando un Microcontrolador"

**T E S I S**

PARA OPTAR EL TITULO PROFESIONAL DE:

**INGENIERO ELECTRONICO**

PRESENTADO POR:

**JUAN MEZA ARRIETA**

P R O M O C I O N : 1 9 8 4 - I I

LIMA - PERU  
1998

**DISEÑO Y CONSTRUCCION DE UN ESTABILIZADOR  
DE VOLTAJE USANDO UN MICROCONTROLADOR**

A mis queridos padres:  
Aquilio y María por su  
apoyo material y  
espiritual.

A mis hermanos: Hibernón,  
Arturo, Natividad y Jesús  
Antonio por la ayuda y  
comprensión de siempre.

## **AGRADECIMIENTOS**

Como preámbulo a este trabajo, quiero expresar mi agradecimiento sincero a los profesores de la Universidad Nacional de Ingeniería por haber contribuído eficazmente en mi formación profesional, que con sus enseñanzas enrumbaron mis conocimientos para aplicar en la vida cotidiana.

Asimismo, mi reconocimiento y gratitud a la Facultad de Ingeniería de Computación y Sistemas de la Universidad de San Martín de Porres FICS por facilitarme en la bibliografía y el uso del Laboratorio de Electrónica.

De una manera especial mi reconocimiento y agradecimiento al Ing. Domingo Lazo Ochoa, en calidad de mi asesor por brindarme el apoyo para la culminación de la presente tesis.

## SUMARIO

En la línea de alimentación de la energía eléctrica se presentan problemas de sobrevoltaje y bajas de voltaje; estos problemas causan el deterioro y mal funcionamiento de los equipos y dispositivos electrónicos ya que están trabajando en condiciones diferentes para las cuales fueron diseñados.

En esta tesis; como solución a estos problemas se plantea el diseño de un estabilizador de voltaje usando un microcontrolador que basándose en un programa tiene como objetivo el de mantener un voltaje de salida estable para ello se realiza el sensado del voltaje de salida, voltaje de entrada y la corriente por los triacs constantemente.

Las características de diseño ha hecho posible la implementación de un equipo confiable y económico.

## INDICE

	pág.
PROLOGO .....	1
<b>CAPITULO I.</b>	
<b>ASPECTOS GENERALES SOBRE ESTABILIZADORES</b>	
1.1 Reguladores y estabilizadores de voltaje.....	4
1.2 Clases de reguladores de voltaje.....	5
1.2.1 Reguladores para corriente continua.....	5
1.2.2 Reguladores para corriente alterna.....	6
1.2.3 Reguladores para corriente continua y alterna...	8
1.3 Estabilizador de voltaje.....	8
1.3.1 Componentes de un estabilizador de voltaje standard.....	10
1.3.2 Tipos de estabilizadores de voltaje.....	11
<b>CAPITULO II.</b>	
<b>ASPECTOS GENERALES SOBRE MICROCONTROLADORES</b>	
2.1 Microcontrolador.....	14
2.2 Microcontroladores de 8 bits, características generales.....	15
2.3 Algunos microcontroladores de 8 bits.....	17
2.3.1 Sección aritmética.....	17
2.3.2 Unidad aritmética y lógica.....	17

2.3.3	Decodificador de instrucción.....	19
2.3.4	Acumulador.....	19
2.3.5	Memoria de programa.....	19
2.3.6	Memoria de datos.....	21
2.3.7	Líneas de entrada/salida.....	23
2.3.8	Entrada INT y de prueba.....	24
2.3.9	Contador de programa y pila o stack.....	24
2.3.10	Palabra de estado del programa (PSW).....	25
2.3.11	Lógica de salto condicional.....	28
2.3.12	Interrupción.....	28
2.3.13	Temporizador contador.....	29
2.3.14	Circuitos de reloj y de temporizador.....	33
2.3.15	Reseteado del microcontrolador .....	35
2.3.16	Avance paso a paso.....	36
2.3.17	Descripción de los terminales.....	39
2.3.18	Set de instrucciones.....	42
2.4	Organización de la memoria en los microcontroladores.....	52
2.4.1	Memoria de programas.....	53
2.4.2	Memoria de datos.....	53
2.4.3	Expansiones en el microcontrolador 8048.....	53
2.5	Control de periféricos, puertos de entrada/salida en los microcontroladores.....	55
2.5.1	El microcontrolador como microprocesador.....	55
2.5.2	Ampliación de las líneas de entrada/salida.....	56
2.6	Modelo de programación de microcontroladores...	56
2.7	Microcontroladores de 16 bits.....	57

**CAPITULO III****ANALISIS DE LOS PRINCIPALES PROBLEMAS DE ESTABILIZACION DE  
VOLTAJE Y SISTEMA DE SOLUCIONES**

3.1	Problemas en la línea de alimentación.....	59
3.2	Causas y efectos de los problemas de estabilización de voltaje.....	60
3.3	Solución a los problemas.....	61
3.4	Alternativas de solución a las variaciones de voltaje.....	62
3.4.1	Estabilizadores tradicionales.....	62
3.4.2	Estabilizador controlado por microcontrolador.	62
3.4.3	Análisis de las alternativas y elección de la mejor.....	63

**CAPITULO IV****DISEÑO Y CONSTRUCCION DEL SISTEMA PROPUESTO, SIMULACION Y  
PRUEBAS DE FUNCIONAMIENTO**

4.1	Características y especificaciones técnicas....	65
4.2	Diagrama de bloques.....	66
4.3	Filosofía de diseño.....	69
4.4	Diseño del autotransformador.....	70
4.5	Diseño del circuito de sensado del voltaje de salida.....	71
4.6	Diseño del circuito de sensado de la corriente por los triacs.....	75
4.7	Diseño del circuito de banco de triacs de menor peso.....	77
4.8	Diseño del circuito de banco de triacs de mayor peso.....	79

4.9	Diseño del circuito de sensado del voltaje de entrada.....	81
4.10	Diseño del circuito de control.....	82
4.11	Diseño de la fuente de alimentación.....	85
4.12	Programación del software.....	90
4.12.1	Programa en Basic para generar las combinaciones posibles que permiten la salida mas cercana a 220 voltios.....	90
4.12.2	Programa en "lenguaje C" para generar las combinaciones posibles que permiten la salida más cercana a 220 voltios.....	97
4.12.3	Diagramas de flujo para el programa de control.....	104
4.12.4	Programa en ensamblador para el sistema de control.....	111
4.12.5	Análisis del programa.....	123
4.12.6	Simulación mediante Spice.....	126
4.12.7	Implementación.....	126
4.12.8	Limitaciones del diseño.....	141
4.12.9	Presupuesto del estabilizador de voltaje usando un microcontrolador.....	142
	CONCLUSIONES Y RECOMENDACIONES.....	145
	ANEXO A	
	CIRCUITO USADO PARA SIMULACION Y RESULTADOS	
	GRAFICOS OBTENIDOS.....	148
	ANEXO B	
	DIAGRAMA DE TIEMPOS DE LAS INSTRUCCIONES.....	154

ANEXO C

JUEGO DE INSTRUCCIONES DEL 8048AH - 8748AH..... 156

ANEXO D

CARACTERÍSTICAS GENERALES DEL 8748H..... 172

ANEXO E

ESQUEMA DEL ESTABILIZADOR DE VOLTAJE USANDO UN  
MICROCONTROLADOR..... 176

BIBLIOGRAFIA..... 178

## PROLOGO

Uno de los problemas que se presenta en la línea de alimentación de la energía eléctrica son las sobretensiones y caídas de tensión; ocurriendo con mucha frecuencia las caídas de tensión o bajas de tensión, en las llamadas horas de punta en las grandes ciudades y en provincias por la deficiente generación y suministro de la energía eléctrica.

Las sobretensiones son originados por dispositivos que consumen gran cantidad de energía eléctrica conectados a la línea de alimentación que se apagan repentinamente, como efectos produciendo el calentamiento de los equipos electrónicos, las bajas de tensión se producen por la conmutación o arranque de equipos que consumen grandes cantidades de energía eléctrica y por las fallas de operación de la planta; como efectos de las bajas de tensión se pueden dañar los equipos electrónicos ya que se tiene una operación forzada.

El propósito de esta tesis es plantear una alternativa de solución a los problemas de sobretensión y bajada de tensión con una técnica no tradicional para el cual se diseña y se construye un estabilizador de voltaje usando un microcontrolador.

Indudablemente la electrónica avanza a pasos agigantados, donde la tecnología de hace algunos años atrás, hoy en día llega a ser una tecnología hasta cierto punto obsoleta o pasada, asimismo junto con esos avances tecnológicos, también aparecen dispositivos electrónicos y circuitos integrados muy eficientes y compactos.

El trabajo aquí presentado está basado precisamente en la utilización de esos componentes electrónicos, haciendo que el control sea más eficiente y económico.

Eficiente en razón de que intervienen un número reducido de dispositivos electrónicos para el diseño del estabilizador de voltaje usando un microcontrolador, el cual me permite una reducción considerable en la construcción del proyecto planteado.

El proyecto final requiere tanto de un diseño circuital (Hardware) como de una programación (Software), el diseño circuital está hecho de tal manera que solamente se han empleado un número reducido de componentes, gracias al uso del microcontrolador.

El programa final ha sido grabado en la memoria EPROM del microcontrolador.

Para la materialización o implementación del circuito de control, se ha hecho un diseño por computadora, usando el paquete llamado smartwork, con el cual se ha podido posteriormente realizar los respectivos fotolitos que me ha permitido obtener las tarjetas del circuito impreso, tanto de las fuentes como del circuito de control, mediante un proceso químico serigráfico.

El siguiente paso ha consistido en la ubicación, colocación y soldadura de todos los componentes electrónicos especificados en el diseño del estabilizador de voltaje usando un microcontrolador.

Son 4 los capítulos en los que se ha dividido la presente tesis.

En el CAPITULO I, se realiza una descripción general de los conceptos relacionados con los estabilizadores de voltaje.

En el CAPITULO II, se describe los conceptos relacionados con el microcontrolador.

En el CAPITULO III, se presenta un análisis de los principales problemas de estabilización de voltaje y las alternativas de solución al respecto.

En el CAPITULO IV, se describe detalladamente el proceso de diseño del hardware y el software necesarios para el sistema propuesto, asimismo se detalla el proceso de implementación y materialización del estabilizador de voltaje usando un microcontrolador.

Por todo lo mencionado anteriormente espero que esta tesis sirva como una ayuda para que en el futuro se realice una mayor difusión y utilización de los microcontroladores en la realización de nuevos diseños de proyectos y sistemas electrónicos.

## CAPITULO I ASPECTOS GENERALES SOBRE ESTABILIZADORES

### 1.1 Reguladores y estabilizadores de voltaje

Se llama compensador a cualquier dispositivo cuya misión sea compensar las variaciones del voltaje de las líneas de alimentación de energía eléctrica.

Dentro de la denominación general de compensadores se tiene dos clases distintas, que son:

Reguladores de voltaje y estabilizadores de voltaje.

Se llaman reguladores de voltaje a los aparatos, generalmente accionados manualmente, que compensan las variaciones de voltaje de la línea de alimentación. En principio se componen de una serie de contactos dispuestos en arco de círculo, que mediante una maneta hacen variar el número de espiras de un transformador cuando la corriente es alterna, e intercalando elementos de acumulador cuando la corriente es continua. Desde luego, la maneta es movida por una persona que debe estar atenta a las fluctuaciones del voltaje de la línea de alimentación para obrar en consecuencia. Por consiguiente, los reguladores de tensión pueden estar destinados a compensar un voltaje continuo o alterno, y además es necesario que una persona vigile para accionar el regulador.

Se denominan estabilizadores de voltaje a los dispositivos que automáticamente compensan las variaciones del voltaje de la línea de alimentación, por consiguiente no requieren intervención manual de ninguna clase, ventaja realmente importante con respecto a los reguladores.

## **1.2 Clases de reguladores de voltaje**

Considerando los procedimientos empleados para regular el voltaje utilizando métodos manuales, es decir que requieren la atención de una persona para graduar convenientemente el valor del voltaje adecuado cuando hay variaciones en la línea de alimentación.

Hay reguladores destinados exclusivamente para corriente continua, otros solo sirven para corriente alterna, y finalmente existen dispositivos que pueden utilizarse indistintamente tanto para corriente continua y alterna.

### **1.2.1 Reguladores para corriente continua**

En lugares donde se trabaja con corriente continua, para compensar las disminuciones del valor normal del voltaje se puede intercalar elementos de una batería de acumuladores destinada para tal efecto, la cual debe tener no sólo un número de elementos capaces de suplir la falta de voltaje, sino también tener la capacidad suficiente.

Para determinar un compensador de voltaje de corriente continua es necesario predeterminar la disminución máxima de voltaje que se acostumbra producir y además el consumo global de la instalación.

### **1.2.2 Reguladores para corriente alterna.**

Este tipo de reguladores pueden ser a transformador o con autotransformador, a su vez pueden estar destinados para corrientes monofásicas, bifásicas, trifásicas, hexafásicas, etc, pero los mas usados son los monofásicos y los trifásicos.

#### **1.2.2.1 Reguladores a transformador**

La ventaja indiscutible del transformador es que el circuito de alimentación, está eléctricamente aislado del secundario, lo cual en muchos aparatos es una necesidad imprescindible.

El devanado primario, conectado a la línea de alimentación es el que recibe las variaciones del voltaje el secundario con un número mayor de espiras eleva el voltaje para recuperar la variación que haya ocurrido en el voltaje de la línea de alimentación.

La relación entre ambas tensiones primaria y secundaria dependen del número de espiras de los dos devanados, como una clave de la regulación del voltaje mediante transformadores, bastará con variar el número de espiras del primario o del secundario, para obtener la gama de variaciones de voltaje que se desea; esto conduce a dos métodos que consisten en variar el número de espiras del primario, o bién hacer la variación del número de espiras en el secundario .

No obstante si la diferencia de voltajes entre primario y secundario no es elevada como ocurre en el caso de los reguladores de tensión puede emplearse cualquiera de los

dos procedimientos.

La determinación del núcleo, secciones de los conductores del primario y del secundario, sus números de espiras correspondientes, etc., constituyen el cálculo del transformador.

### 1.2.2.2 Regulador a autotransformador

Cuando es necesario elevar o bajar un voltaje se puede emplear autotransformadores, la única limitación es que el aparato de uso permita que exista una conexión directa con la línea de alimentación, los autotransformadores pueden servir para corrientes monofásicas y trifásicas lo mismo que los transformadores la casi totalidad de los reguladores de voltaje están contruidos con autotransformadores, siendo las excepciones los casos mencionados anteriormente.

Los reguladores de tensión trifásicos no difieren de los monofásicos en cuanto a su fundamento, en realidad actúan como tres reguladores monofásicos montados de tal manera que la regulación de las tres fases se realiza simultáneamente con un solo control, esto se consigue acoplado mecánicamente el control de las tres fases de tal forma las variaciones del número de espiras se hacen simultáneamente en las tres fases. Desde luego estas variaciones pueden hacerse de dos maneras distintas, espira por espira como en el caso de los reguladores monofásicos o bien por secciones con varias espiras en cada una de ellas.

### **1.2.3 Reguladores para corriente continua y alterna**

Sólo en casos muy especiales se utiliza esta clase de reguladores ya que están limitados a dos condiciones. La primera es que sólo reducen la tensión, no es posible aumentarla con este procedimiento y por lo tanto no tiene aplicación práctica para el caso general de elevar la tensión de la línea de alimentación.

En segundo lugar la potencia eléctrica que puede obtenerse es muy limitada.

### **1.3 Estabilizador de voltaje.**

Un Estabilizador de voltaje es un sistema electrónico, que tiene como función principal compensar automáticamente las variaciones del voltaje de la línea de energía eléctrica, manteniendo niveles de voltaje y condiciones que aseguren un adecuado funcionamiento de los equipos electrónicos.

Por la forma de construcción los estabilizadores pueden ser: estabilizadores de tomas y de regulación continua.

Los estabilizadores de tomas se construyen con un transformador cuyo primario o secundario se conecta a través de tomas seleccionadas por conmutadores estáticos normalmente triacs o tiristores, dependiendo de los voltajes de entrada y de la carga.

El circuito de control selecciona la toma apropiada, para asegurar una tensión de salida dentro de los márgenes establecidos, este tipo de estabilizador es apto para potencias de hasta unos 4 KVA.

Las tomas pueden estar en el primario o en el

secundario existiendo ventajas e inconvenientes, desde el punto de vista de la protección frente a ruidos es mejor disponerlas en el secundario.

Para evitar o reducir las corrientes entre tomas al conmutar de una a otra se usa la conmutación en los pasos por cero de la intensidad de corriente eléctrica.

En potencias superiores a 5 KVA es mas aconsejable utilizar un transformador compensador intercalado entre la red y el transformador de aislamiento adiciona o quita tensión al primario de este para mantener la tensión de salida dentro de márgenes.

El primario del transformador compensador se alimenta mediante la selección de una toma del primario del transformador de aislamiento, como la relación de espiras del transformador es del orden de 5/2 aproximadamente para un estabilizador de +- 15% de tensión de entrada, la intensidad que tienen que manejar los triacs se ve reducida en dicha relación respecto de la nominal de salida.

Los estabilizadores de tomas explicados pueden realizarse con autotransformadores en lugar de transformadores de aislamiento para reducir el tamaño y el costo, en contrapartida se pierde el aislamiento galvánico entre la entrada y la salida.

Estabilizadores de regulación continua, en los estabilizadores de tomas, cuando se pasa de una a otra, la tensión de salida es sometida a un incremento o a un decremento brusco de orden del doble de la precisión de

salida, así para un estabilizador de precisión de  $\pm 3\%$ , el salto de tensión es del 6% aproximadamente, algunas cargas no admiten estos saltos bruscos, suelen ser raras y entre ellas se encuentran ciertos instrumentos químicos y fotográficos, en tales casos es necesario recurrir a estabilizadores de regulación continua

### **1.3.1 Componentes de un estabilizador de voltaje estandard.**

El estabilizador de voltaje es un conjunto de circuitos electrónicos, que relacionados entre si permiten el funcionamiento eficiente y seguro de otros sistemas electrónicos.

Un estabilizador de voltaje estandard, está formado principalmente por dos etapas : Una etapa de control y una etapa de potencia.

La etapa de control, cuenta con una fuente de alimentación para el control electrónico AC-Dc no regulado, con el fin de proporcionar niveles de variación VDC, reflejo del VAC de entrada, el voltaje DC no regulado se lleva a un sistema de comparadores de pasos, para el control de mando de potencia, sirviendo como mando de disparo al banco de Triac previamente pasando por las interfaces hasta los foto acopladores.

No existe contacto eléctrico entre la etapa de control y la etapa de potencia, gracias a la característica propia del opto acoplador, que además de proporcionar aislamiento galvánico, cuenta con detectores de cruce por cero, es entonces que la conmutación de Triac a Triac se

produce exactamente cuando la tensión alterna de potencia pasa por cero anulando de esta forma ruidos por RF debido a la conmutación.

A esta configuración se le adiciona para lograr mayor eficiencia y seguridad un circuito de control de sensor veloz de red, conformado por un comparador de nivel, mediante interfaces y un flip flop retienen el dato lógico, que tiene por función detectar el VAC de entrada cuando se produce subtensiones y sobretensiones bruscas, de esta forma anula a todos los foto acopladores y por consiguiente el banco de Triacs, no existiendo VAC en la salida del estabilizador.

La etapa de potencia esta formada por los foto acopladores, el banco de Triacs, transformador de acoplamiento y las protecciones conforman el limitador de corriente para cada Triac y el fusible de protección especificando según la corriente de carga máxima.

### **1.3.2 Tipos de estabilizadores de voltaje**

Entre los tipos de estabilizadores de voltaje podemos mencionar los siguientes :

#### **1.3.2.1 Estabilizadores electromecánicos**

Este tipo de estabilizador tiene un motor controlado por un detector de estado sólido. Cuando la salida se desvía del valor nominal el control actúa de modo que el transformador auxiliar conectado en serie con la línea genere un voltaje de compensación para recuperar el voltaje nominal. Este equipo es demasiado lento para movilizar una masa inerte como es el motor, estos

actualmente no son muy comunes ni comerciales.

### **1.3.2.2 Estabilizadores de conmutación**

Según el tipo de conmutación, se puede mencionar los siguientes:

#### **1.3.2.2.1 Estabilizadores basado en relés o híbridos**

Aquí el motor es reemplazado por relés para la selección de derivación que generará la tensión de compensación.

Los estabilizadores de voltaje híbridos, constituyen uno de los estabilizadores mas antiguos, que se encuentran en el mercado su funcionamiento es de forma electromecánica es decir que sus devanados conmutan en forma electromecánica utilizando relés adecuados a la potencia necesaria.

#### **1.3.2.2.2 Estabilizadores de estado sólido.**

Los estabilizadores de voltaje de estado sólido tiene su funcionamiento en forma electrónica, utilizan un transformador con muchas salidas colocando Triacs para conmutar en cada uno de sus devanados, el sistema sensa y escoge cual de ellos deberá actuar para una condición específica. Un filtro capacitor es necesario colocar en la entrada para minimizar el ruido de alta frecuencia. El disparo de los conmutadores estáticos es mediante el sistema de cruce por cero para evitar deformaciones de la onda.

Estos son protección ideales para soportar muchas horas de trabajo continuos y capaces de ofrecer una alta precisión y confiabilidad en los niveles de voltaje.

### **1.3.2.3 Estabilizadores ferroresonantes**

Los estabilizadores de voltaje ferroresonantes, son contruidos con tecnología ferroresonante, en la cual intervienen simultaneamente diversos fenómenos distintos : Eléctricos, magnéticos, de resonancia eléctrica, frecuencia y desfasamiento, este tipo de sistemas tienen su aplicación mayormente en zonas industriales.

## CAPITULO II ASPECTOS GENERALES SOBRE MICROCONTROLADORES

### 2.1 Microcontrolador.

Un microcontrolador es un ordenador en un sólo circuito integrado, es decir se integraron junto con el microprocesador los subsistemas que anteriormente formaban unidades especializadas e independientes, pero unidas por las pistas del circuito impreso con el microprocesador.

Un microcontrolador integra los siguientes subsistemas, aunque no siempre presenta todos los que se mencionan a continuación:

- CPU.
- RAM.
- ROM o EPROM.
- Entradas / Salidas.
- Contadores y temporizadores.
- Conversores Analógico / Digital.
- Conversores Digital / Analógico.
- Gestión de interrupciones.

Actualmente un microcontrolador es algo más que el enunciado anteriormente; por su evolución constante y por la importancia en el mercado de aplicaciones que cada día tiene, pero su característica más sobresaliente es que es un sistema integrado, aspecto muy atractivo para su

aplicación en muchos campos.

## **2.2 Microcontroladores de 8 bits, Características**

### **Generales.**

Los microcontroladores de 8 bits, son el estándar para aplicaciones de mediana complejidad, por su facilidad de manejo y nivel de prestaciones.

Entre las características mas importantes tenemos:

- Capacidad de proceso de palabra de 8 bits.
- Circuito de reloj incorporado.
- Frecuencia de reloj de hasta 30 Mhz.
- Múltiples puertos de entrada/salida programables.
- Bajo consumo de energía en las versiones CHMOS.
- Alta inmunidad al ruido eléctrico.
- Ampliación del set de instrucciones con algunas muy potentes para la adquisición y tratamiento de datos, tablas, multiplicación, división, etc.
- Instrucciones lógicas y de bifurcación orientadas al proceso de señales bit a bit.
- Espacios de memoria de trabajo separados.
- Direccionamiento de 64 Kbytes de ROM interna + externa.
- Direccionamiento de 64 Kbytes de RAM externa.
- Protección de la memoria de programas ( encriptación ).
- Sistema de vigilancia del funcionamiento óptimo de la CPU.

Posibilidad de comunicación estándar con otros sistemas Tipo full - duplex.

Conversores A/D integrados en el propio chip.

- Salida de modulación de ancho de pulso (PWM) para la conversión D/A.
- 2 ó 3 temporizadores de 16 bits.  
5 ó 6 interrupciones programables con niveles de prioridad.
- Posibilidad de incorporar otras funciones bajo diseño específico ASIC (Application Specific Integrated Circuits).

Además permite que el microcontrolador sea utilizado como un microprocesador implementando la memoria RAM, EPROM y/o unidades de entrada/salida que el usuario considere mas adecuada a su aplicación.

Por si esto fuera poco, algunos fabricantes, han lanzado al mercado microcontroladores con interpretes de lenguajes de alto nivel incorporados en su memoria ROM interna, lo que permite dar respuesta a situaciones donde es preciso automatizar algún proceso de una manera fácil, por personal no demasiado especializado en el conocimiento y aplicación del lenguaje ensamblador, pero con una gran experiencia en el campo de la automatización y conocedores de un lenguaje de programación de alto nivel.

Estos microcontroladores ofrecen también :

- Herramientas de programación y autosoportadas.
- Funciones de autómata programable en un lenguaje de alto nivel.

Paralelamente al desarrollo de los microcontroladores evolucionan las herramientas de instrumentación que permiten el diseño y mantenimiento de los equipos que

llevan estos circuitos integrados.

### **2.3 Algunos microcontroladores de 8 bits.**

En esta parte, se hará una breve descripción de toda la arquitectura y características funcionales del 8048 AH en base al diagrama de bloque de la Fig.1

#### **2.3.1 Sección aritmética**

La sección aritmética del procesador contiene las funciones básicas de manipulación de datos del 8048 AH y puede ser dividido en los siguientes bloques:

- Unidad aritmética y lógica (ALU).
- Acumulador.
- Carry flag.
- Decodificador de instrucción.

En una típica operación, el dato almacenado en el acumulador es combinado en la ALU con el dato desde otra fuente sobre el bus interno (semejante así a un registro o puerto de I/O), el resultado es almacenado en el acumulador o en otro registro.

#### **2.3.2 Unidad aritmética y lógica**

La ALU acepta palabras de 8 bit de datos, desde una o dos fuentes y genera un resultado de 8 bit bajo control de la decodificación de instrucción. La ALU puede ejecutar las siguientes funciones:

- Add con o sin carry.
- AND, OR, OR exclusivo.
- Incremento / decremento.
- Complementar bit.

Rotación Izquierda, Derecha.

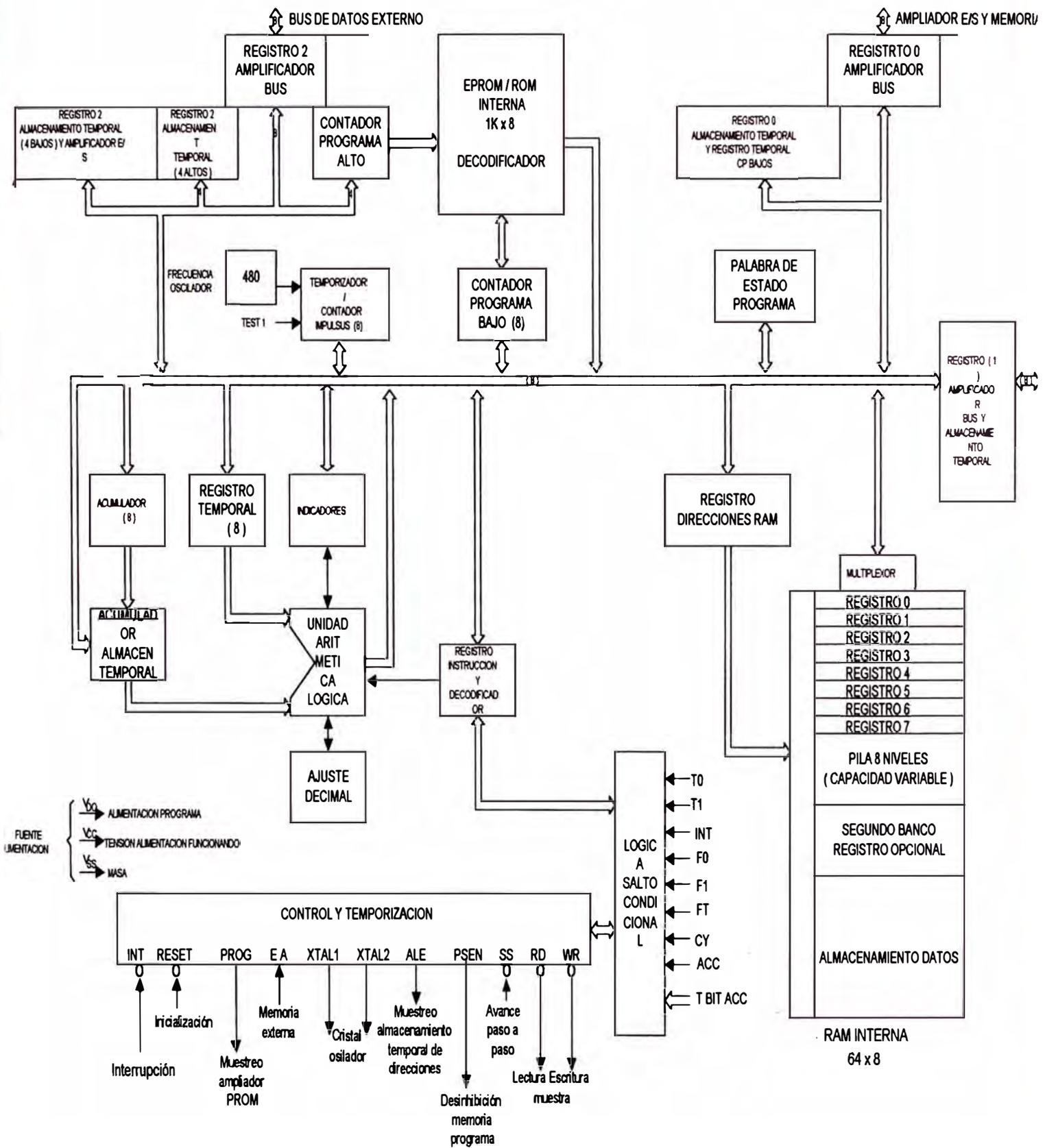


Fig. 1

Diagrama de bloques 8048 AH/8748 AH

- SWAP Nibbles (intercambiar 4 bit de un byte).
- Ajuste decimal BCD.

Si la operación realizada por la ALU resulta en un valor representado por más de 8 bit (sobrepasamiento del bit más significativo), un Carry flag es colocado en la palabra del Status Program.

### **2.3.3 Decodificador de instrucción**

La porción del código de operación (opcode) de cada una de las instrucciones de programa, es almacenado en el decodificador de Instrucción y convertido a salidas que controla las funciones de cada uno de los bloques de la Sección Aritmética. Estas líneas de control, la fuente de datos, y el registro destino así como también las funciones son realizadas en la ALU.

### **2.3.4 Acumulador**

El acumulador es el registro de datos sencillo y más importante en el procesador, es una de las fuentes de entrada hacia la ALU y frecuentemente el destino de los resultados de operaciones realizadas en la ALU.

Los datos hacia y desde puertos I/O y memoria, del mismo modo normalmente pasan a través del acumulador.

### **2.3.5 Memoria de programa**

La memoria de programa residente consiste de 1024, 2048 ó 4096 palabras de 8 bit de ancho que son direccionadas por el contador de programa, en el 8748AH y 8749AH, esta memoria es una EPROM.

Dentro de ésta memoria hay 3 direcciones o localizaciones de especial importancia como lo muestra la Fig. 2.

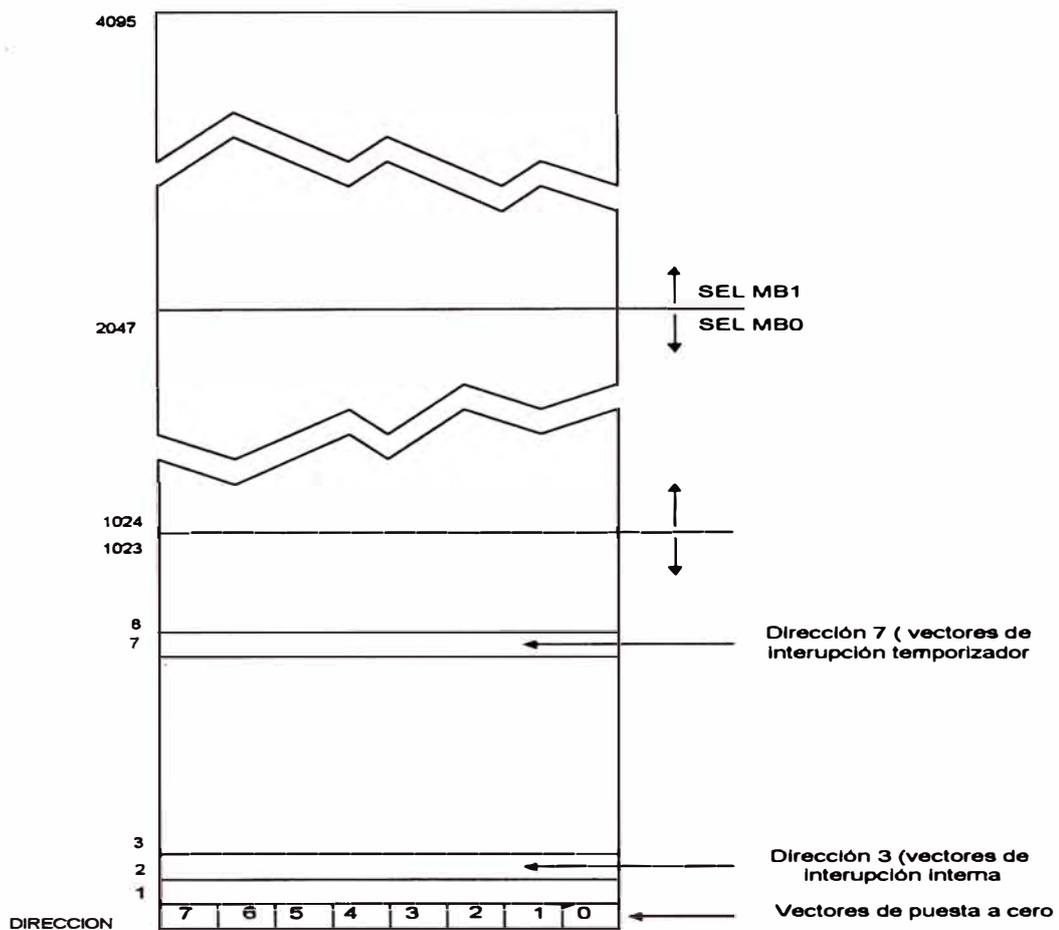


Fig. 2

### Mapa de memoria de programa

#### Localización 0

Activando la línea Reset del procesador, causa la primera instrucción a ser buscada desde la localización 0.

#### Localización 3

Activando la línea de entrada Interrupt del procesador (si Interrupt es habilitado), causa un salto hacia la subrutina de localización 3.

#### Localización 7

Una Interrupción o rebasamiento del contador/temporizador (si es habilitado), causa un salto

hacia la subrutina de localización 7, por consiguiente, la primera instrucción a ser ejecutada después de la inicialización es almacenada en localización 0, la primera palabra de una interrupción externa de servicio de subrutina es almacenada en localización 3, y la primera palabra de un servicio de rutina de contador/temporizador es almacenada en localización 7.

### **2.3.6 Memoria de datos**

La memoria de datos residente es organizada como 64, 128, o 256 palabras de 8 bit de ancho en el 8048AH, 8049AH, 8059AH respectivamente.

Todas las localizaciones son indirectamente direccionadas por medio de los registros R0 y R1 del banco de registros de la RAM, así como muestra la Fig.3.

Las primeras 8 localizaciones (0-7) del arreglo son designadas como registros de trabajo y son directamente direccionadas por varias instrucciones. Donde estos registros son más fácilmente direccionables, ellos son habitualmente usados para almacenar resultados intermedios frecuentemente accedados.

Mediante la ejecución de la instrucción de conmutación del banco de registro SEL RB, las localizaciones 24-31 de la RAM, son designadas como Registros de Trabajo en lugar de las localizaciones 0-7 y son entonces directamente direccionables.

Este segundo banco de registros de trabajo se puede utilizar como ampliación del primer banco o reservarlo para su uso durante unas subrutinas de interrupción,

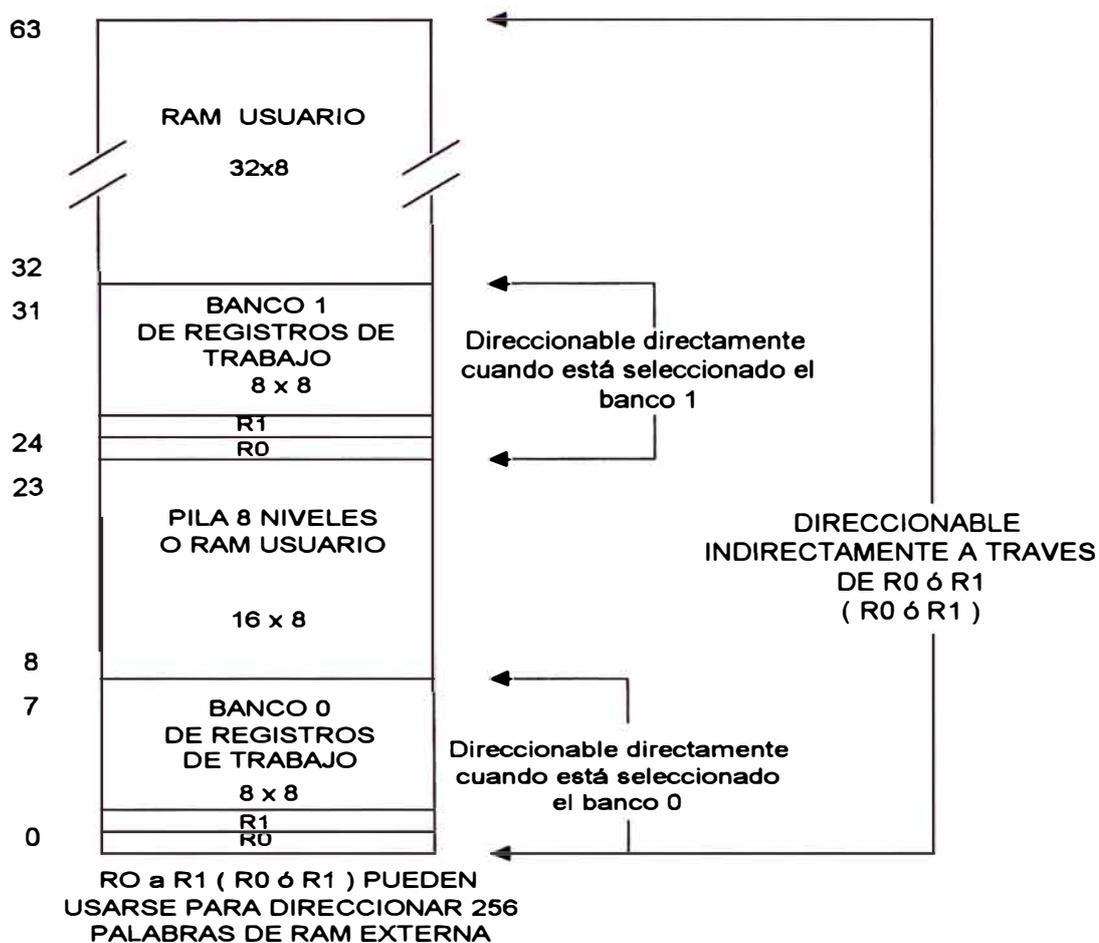


Fig. 3

## Mapa de memoria de datos

permitiendo reservar el contenido de los registros del banco 0, utilizados en el mismo programa, mediante la conmutación de banco. En caso de no ser utilizado este segundo banco, las posiciones 24 - 31 pueden ser direccionadas como aplicación general de memoria RAM.

Ya que los punteros de la RAM R0 y R1, son una parte del sistema de registros de trabajo, la conmutación de banco crea dos nuevos registros punteros ( R'0 y R'1 ), los cuales se pueden utilizar junto con R0 y R1 para facilitar el acceso al mismo tiempo a 4 zonas de trabajo

separadas dentro de la RAM.

Las localizaciones 8-23 de la RAM; son direccionadas por el puntero de pila durante la llamada a subrutinas, así como por los Registros Punteros de la RAM, R0 y R1.

Si el nivel de subrutinas anidadas es menor que 8, todos los registros de pila no son requeridos y pueden ser usados como localizaciones de RAM de propósito general.

### **2.3.7 Líneas de entrada/salida**

El 8048AH tiene 27 líneas, agrupadas en 3 registros de entrada/salida de 8 líneas cada uno, teniendo funciones de entrada, salida o registro bidireccional, y 3 entradas de examen que pueden alterar secuencias de programas cuando son examinadas por instrucciones de salto condicional.

**Puertos 1 y 2** o registro de E/S 1 y 2.

Los Puertos 1 y 2 o registros de E/S 1 y 2 constan de 8 bit cada uno y tienen ambas idénticas características.

Los datos grabados en estos registros, cuando trabajan como salidas, se almacenan estáticamente y permanecen invariables mientras no se graben otros datos.

Actuando como registros de entrada, los datos no se almacenan y deben permanecer presentes mientras se efectúa la lectura de los mismos mediante una instrucción de entrada de datos.

Las entradas son totalmente compatibles con TTL, y las salidas son capaces de exitar una carga TTL.

Las Líneas de estos 2 registros se llaman cuasibidireccionables, debido a que el circuito de salida tiene una estructura especial que permite que cada línea

pueda servir como entrada, salida y salida con almacenamiento temporal estático de los datos.

#### **Bus de datos**

El Bus de datos es también un registro de E/S de 8 bit, bidireccional con entradas y salidas comunes, con la diferencia respecto a los anteriores de no poderse mezclar las entradas y salidas, o sea, las 8 líneas deberán ser todas entradas o todas salidas.

Cuando no se escribe ni se lee, el bus de datos se encuentra en estado de alta impedancia.

#### **2.3.8 Entrada INT y de prueba**

Tres pines sirven como entradas y a la vez pueden ser examinadas mediante instrucciones de salto condicional. Estos son T0, T1 e INT, y mediante ellos se consiguen saltos de programa sin necesidad de cargar el contenido de un registro de E/S en el acumulador, además de realizar otras funciones.

#### **2.3.9 Contador de programa y pila o stack**

El contador de programa es un contador independiente, mientras tanto el contador de programa y pila es implementado usando pares de registros del arreglo de memoria de datos, solamente los bits 10, 11 o 12 del contador de programa son usados para direccionar las 1024, 2048 o 4096 palabras sobre una tarjeta de memoria de programa del 8048AH, 8049AH o 8050AH, mientras tanto el bit mas significativo puede ser usado para buscar el programa de memoria externa ver fig. 4.

El contador de programa es inicializado a cero por la

activación de la línea Reset.



Contador convencional programa

- Computo de 000 a 7FF
- Rebasamiento capacidad 7FF a 000

Fig. 4

Contador de programa

Una interrupción o llamada a una subrutina, causa que el contenido del contador de programa sea almacenado en uno de los 8 registros pares del contador de programa de pila como lo muestra la fi. 5, el par a ser usado es determinado por 3 bits del puntero de pila el cual es parte del program status word (PSW).

El fin de una subrutina, la cual es señalada, por una instrucción de retorno ( RET o RETR ), causa que el puntero de pila sea decrementado y el contenido del registro par resultante a ser transferido hacia el contador de programa.

#### **2.3.10 Palabra de estado del programa (PSW)**

Existe una serie de biestables organizados en una palabra de 8 bits, los cuales pueden cargarse desde el acumulador y se encargan de informar del estado de ciertos parámetros de un programa. Esta palabra de 8 bits

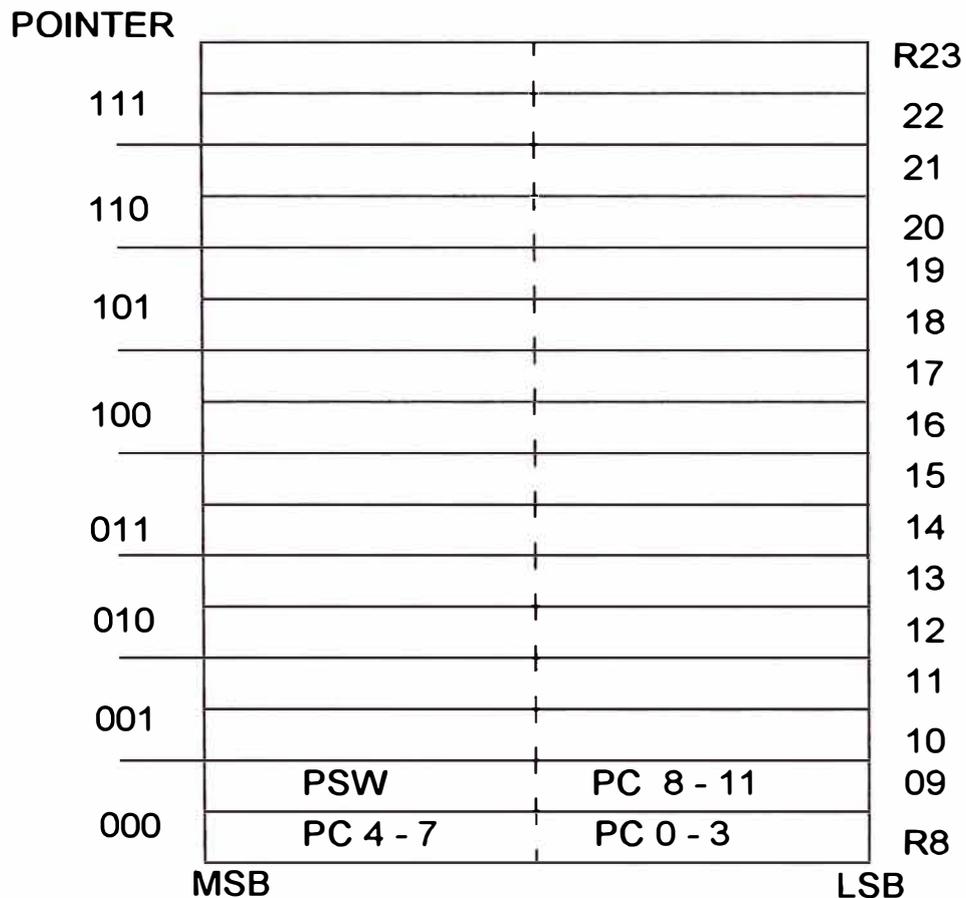
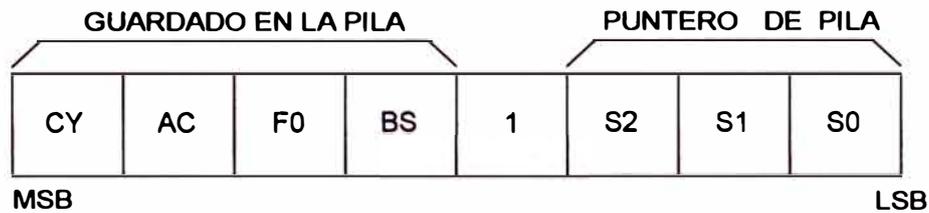


Fig. 5

## Puntero de stack

es lo que constituye la llamada palabra de estado de programa, PSW (Program Status Word). Estos biestables pueden ser leídos o grabados muy fácilmente. La fig. 6 muestra la información disponible en la palabra.

Los cuatro bits mas significativos de la PSW son almacenados en la pila del contador de programa al producirse un salto a subrutina, o vector de interrupción, y restablecidos a su estado inicial automáticamente mediante la instrucción RETR. La instrucción de retorno RET no restablece la PSW.



CY Acarreo

AC Acarreo auxiliar

FO Indicador 0

BS Selección banco registro

Fig. 6

Program status word

La función de cada uno de los bits de la PSW es la siguiente:

- Bits 0-2 bits del puntero de pila (S0, S1, S2).
- Bit 3 no se usa (se encuentra a nivel de "1" para casos de lectura de la PSW ).
- Bit 4 conmutador de banco de registros de trabajo.
- Bit 5 bit del indicador 0 ( f0 ), controlable por el usuario que puede ser borrado, complementado, o examinado mediante la instrucción de salto condicional JFO.
- Bit 6 acarreo auxiliar, AC ( Auxiliary Carry ), generado mediante una instrucción de suma ADD y utilizado por la instrucción de ajuste decimal DAA.
- Bit 7 acarreo, CY ( Carry ) que indica que una operación previa ha tenido un resultado con rebasamiento de capacidad en el acumulador.

### **2.3.11 Lógica de salto condicional**

La lógica de salto condicional en el procesador, habilita varias condiciones internas y externas hacia el procesador a ser examinadas por el programa de usuario. Una instrucción de salto condicional, puede efectuar un cambio en la secuencia, de la ejecución del programa.

### **2.3.12 Interrupción**

Al aplicar un " 0 " lógico en el terminal de entrada INT. se inicia, una secuencia de interrupción. La línea de interrupción se muestra en cada ciclo de Máquina, en el tiempo que dura la señal ALE, y cuando se detecta una interrupción se produce un salto a una subrutina de la posición 3 de la memoria de programa, en cuanto hayan sido completados todos los ciclos de la instrucción que se lleve a cabo en ese momento. Al igual que en cualquier llamada a subrutina, el contenido del contador de programa, y la palabra de estados son guardados en la pila. La posición 3 de la memoria de programa contiene normalmente un salto incondicional a una subrutina de interrupción que se encuentra en otra parte de la memoria de programa. El final de una subrutina de interrupción que se encuentra en otra parte de la memoria de programa, el final de una subrutina de interrupción se produce cuando es ejecutada la instrucción de retorno y restablecimiento del estado del programa RETR. El sistema de interrupciones consta de un único nivel, en el cual, mientras se está ejecutando alguna interrupción, una nueva petición exterior es ignorada mientras no se ejecuta la instrucción

de RETR que se encargará de desinhibir el circuito lógico de entrada de interrupción.

Esta secuencia es la misma para interrupciones internas, producidas por el rebasamiento de la capacidad del temporizador. En caso de detectar una interrupción interna y otra externa al mismo tiempo, tendrá prioridad la externa.

### **Secuencia de interrupción.**

La entrada de interrupción puede ser inhibida o desinhibida por programa mediante las instrucciones EN I y DIS I. Las interrupciones pueden ser inhibidas mediante la puesta a cero y mantenidas así hasta ser desinhibidas por programa. Una petición de interrupción debe ser retirada antes de que se ejecute la instrucción de retorno de subrutina RETR. De otra forma, el procesador volvería a reiniciar la subrutina de interrupción inmediatamente.

El terminal INT puede ser examinado mediante la instrucción de salto condicional INI. Esta instrucción puede ser usada para detectar la presencia de interrupciones pendientes de ejecución antes de que las mismas sean desinhibidas. En caso de que las interrupciones estén inhibidas, el terminal INT puede ser usado como entrada de muestreo al igual que T0 y T1.

### **2.3.13 Temporizador contador.**

El contador temporizador de 8 bits permite al microcontrolador la cuenta de eventos, o la producción de tiempos exactos de retardo, sin recargar el microcontrolador, la única diferencia entre ambas

funciones es la fuente de entrada al contador, fig. 7.

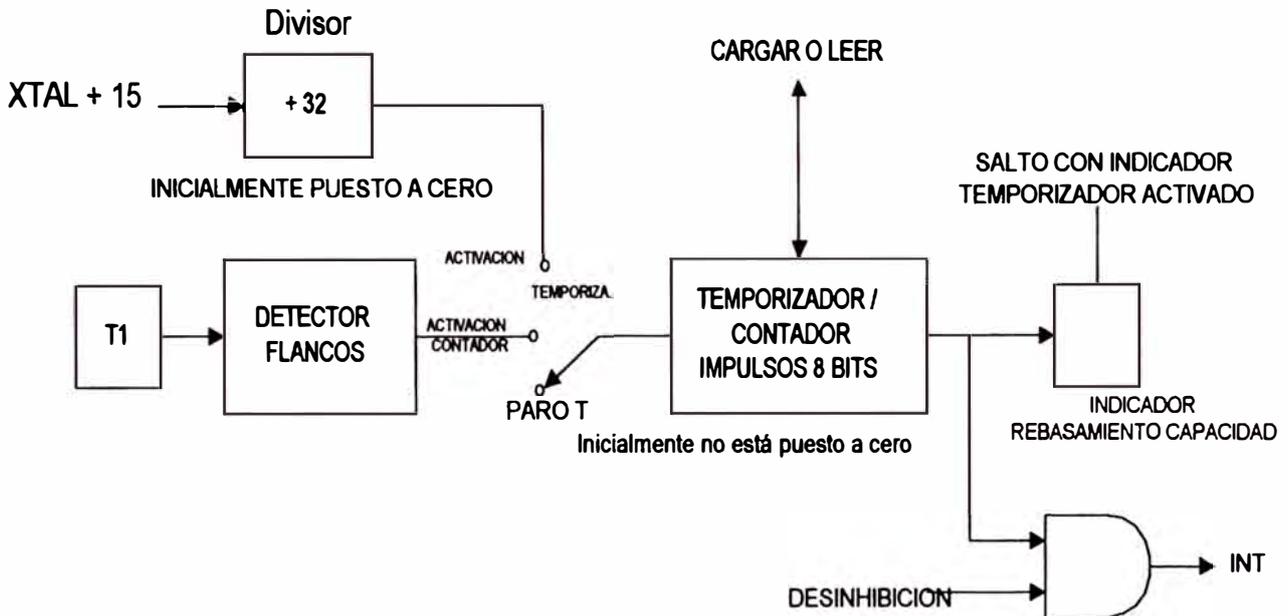


Fig. 7

### Temporizador / Contador

El contador binario de 8 bits puede ser cargado y leído mediante dos instrucciones MOV, que transfieren el contenido del acumulador al contador y viceversa. El contenido del contador no queda afectado por la puesta a cero general, siendo inicializado únicamente mediante la instrucción.

MOV T,A. El contador puede pararse mediante la puesta a cero general o la instrucción STOP TCNT, manteniéndose parado mientras no es puesto en marcha como temporizador mediante la instrucción START CNT. Una vez puesto en marcha, el contador irá incrementando su valor hasta el máximo cómputo (FF), dando señal de rebase de capacidad y continuando el cómputo a partir de cero, hasta que es

parado mediante la instrucción STOP TCNT, o la puesta a cero general.

El incremento desde el máximo valor de cómputo a cero (rebase de capacidad) activa un biestable indicador de rebase de capacidad y genera una petición de interrupción. El estado del indicador de rebase de capacidad puede examinarse mediante la instrucción de salto condicional JTF. El indicador es borrado mediante la puesta a cero general o la ejecución de la instrucción JTF. La petición de interrupción es almacenada en un biestable y seguidamente combinada con la señal de petición de interrupción externa INT, a través de una puerta OR. La interrupción del temporizador puede ser desinhibida o inhibida independientemente de la señal de interrupción externa, mediante las instrucciones EN TCNTI y DIS TCNTI.

Si está desinhibida, un rebasamiento de la capacidad en el contador causará un salto a la posición 7 de la memoria donde se encuentra almacenada la rutina de servicio del temporizador o del contador de sucesos. En caso de presentarse simultáneamente dos peticiones de interrupción, una externa y otra del temporizador, tendrá preferencia la externa y la llamada a subrutina de interrupción se hará en la posición 3 de la memoria. No obstante, la interrupción del temporizador es almacenada temporalmente, permaneciendo a la espera de que haya sido servido el dispositivo externo que pidió interrupción en el momento en que se produzca el retorno de la subrutina de interrupción del temporizador. La petición de

interrupción pendiente se borra en el momento que se salta a la posición 7 de la memoria, o puede ser suprimido mediante la instrucción DIS TCNTI.

### **Contador de sucesos**

La ejecución de la instrucción START CNT conecta el terminal de entrada T1 a la entrada del contador y la desinhibe. Cada flanco de bajada en T originará un incremento de valor en el contador. La máxima velocidad a la cual el contador puede ser incrementado es una vez por cada tres ciclos (cada 7,5 microseg. si se usa un cristal de 6 MHz); no obstante, esta no es la mínima frecuencia a la que puede funcionar. La entrada T1 necesita estar un mínimo de 500 microseg. en estado de alto después de cada transición.

### **Contador utilizado como temporizador**

La ejecución de la instrucción START T conecta un reloj interno a la entrada del contador y desinhibe el contador.

El reloj interno es el resultado de pasar los 400 kHz de ciclo de reloj básico (señal ALE) a través de un divisor por 32. El divisor se pone a cero durante la ejecución de la instrucción START T. El reloj resultante de 12,5 kHz incrementa el contador cada 80 microseg., (siempre que se encuentre conectado un cristal de 6 MHz)., se obtienen diferentes retardos, desde 80 microseg. hasta 20 ms (256 cálculos), cargando el contador con un valor (inicialización del contador) y la detección de la señal de rebase de capacidad. Se pueden obtener tiempos de retardo mayores de 20 ms mediante la acumulación de

múltiples impulsos de rebase de capacidad en un registro, controlando el proceso por programa. Para tiempos menores de 80 microseg., puede aplicarse un reloj externo a la entrada T1 haciendo operar el contador en el modo "contador de sucesos". La señal ALE dividida por 3 o más puede hacer las funciones de reloj externo. Pueden obtenerse fácilmente tiempos de retardo muy pequeños o muy grandes mediante lazos de retardo realizados por programa.

### 2.3.14 Circuitos de reloj y de temporizador

La generación de secuencias en el 8048 está totalmente resuelta dentro del mismo, a excepción de la frecuencia de referencia, que puede ser generada por cristal, por inductancias o por una fuente de reloj externa. Los circuitos de reloj y de temporización pueden dividirse en los siguientes bloques funcionales, Fig. 8.

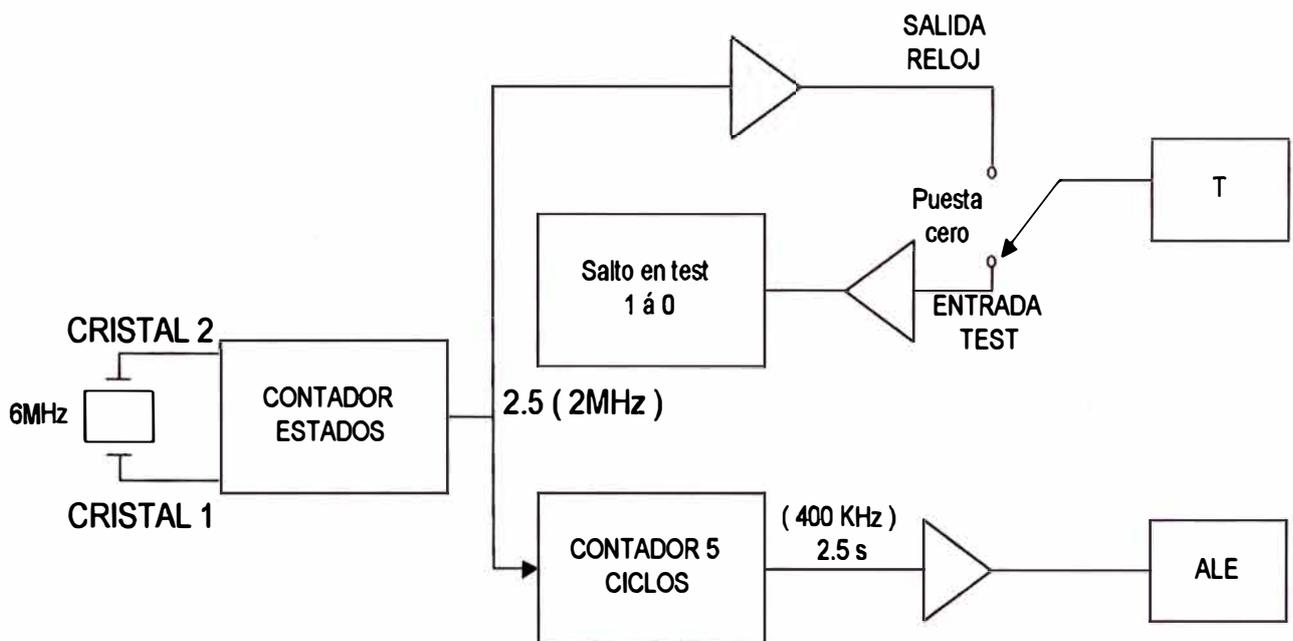


Fig. 8

Circuitos de reloj y temporización

## **Oscilador**

El oscilador consta de una serie de circuitos resonantes de alta ganancia con una escala de frecuencias que va de 1 a 6 MHz. El terminal exterior X1 es la entrada a la etapa amplificadora, siendo X2 la salida de la misma.

Un cristal o inductancia conectada entre los terminales X1 y X2 proporciona la realimentación y desplazamiento de fase necesarios para que el circuito se ponga a oscilar.

Un cristal de 5.9904 MHz permite obtener fácilmente todas las frecuencias normalizadas en comunicaciones. Si no es necesaria una gran precisión en la frecuencia de referencia, ni una velocidad alta en el procesador, puede utilizarse una inductancia en lugar de un cristal. Con una inductancia, la frecuencia del oscilador puede ser aproximadamente de 3 a 5 MHz. Para mayores velocidades de operación se utilizará un cristal. Asimismo, se puede aplicar un reloj externo en los terminales X1 y X2 como fuente de referencia.

## **Contador de estados**

La salida del oscilador se divide por tres en el contador de estados para crear un reloj que define el tiempo de los estados de la máquina (CLK). La señal CLK puede estar disponible en el terminal externo T0 mediante la ejecución de la instrucción ENTO CLK. La salida CLK por T0 puede ser inhibida mediante la puesta a cero general del procesador.

## **Contador de Ciclos**

La señal CLK es dividida posteriormente por cinco en el

contador de ciclos, proporcionando una señal de reloj que define el ciclo de máquina compuesto por cinco estados de máquina. Este reloj se denomina ALE (Address Latch Enable) y, entre otras funciones, se utiliza para desinhibir los registros de dirección de las memorias externas. Esta señal se encuentra disponible permanentemente en el terminal de salida ALE.

### 2.3.15 Reseteado del microcontrolador

Esta entrada de control permite inicializar al microcontrolador Fig. 9

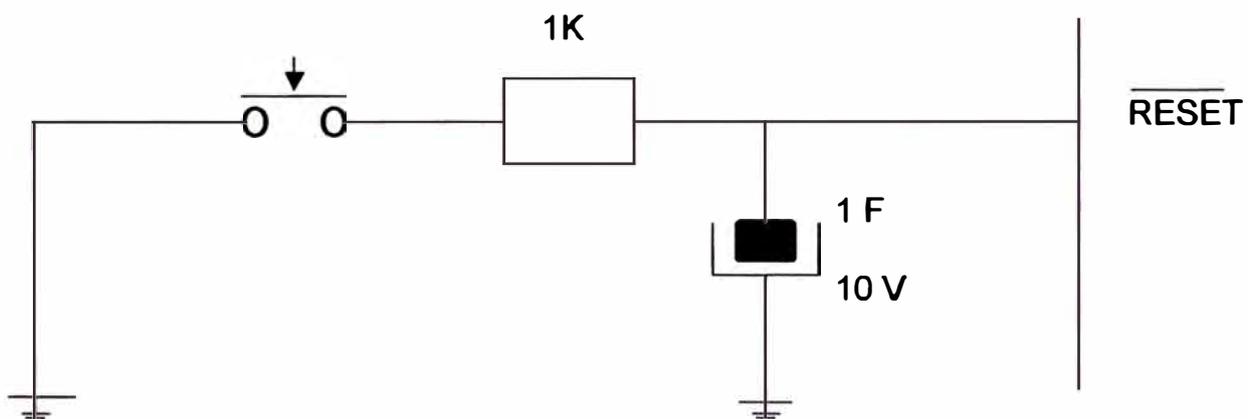


Fig. 9

Circuito de reseteo del microcontrolador

Consta de un disparador Schmitt y una resistencia conectada a positivo que, en combinación con un condensador exterior de 1  $\mu$ F, proporciona un impulso de puesta a cero interno de suficiente duración como para garantizar que todo el circuito ha sido puesto a cero. En caso de que el impulso de puesta a cero sea generado exteriormente, el terminal de puesta a cero tiene que

mantenerse a nivel de masa (0,5 V máximo) por un periodo de tiempo mínimo de 50 ms inmediatamente después de que la tensión de alimentación ha llegado a su valor nominal.

La puesta a cero realiza las siguientes funciones:

- Pone a cero el contador de programa.

Pone a cero el puntero de pila.

- Selecciona el banco de 0 de registros.

Selecciona el banco 0 de memoria.

Pone el bus en el estado de alta impedancia (excepto cuando EA=5V).

- Pone los registros de E/S 1 y 2 en el modo de entrada.

- Inhibe las interrupciones (la del temporizador y la externa).

- Para el temporizador.

- Borra el indicador del temporizador (FT).

- Borra F0 y F1.

- Inhibe la salida de reloj desde T0.

### **2.3.16 Avance paso a paso**

Esta característica proporciona al usuario la posibilidad de hacer progresar el programa instrucción a instrucción. Mientras el programa se encuentra detenido, la dirección de la siguiente instrucción a ejecutar se encuentra disponible en la barra y la mitad del registro de E/S. El usuario puede, gracias a esto, seguir el programa a través de cualquier paso de instrucción.

En la fig.10 puede verse la disposición hardware del mismo.

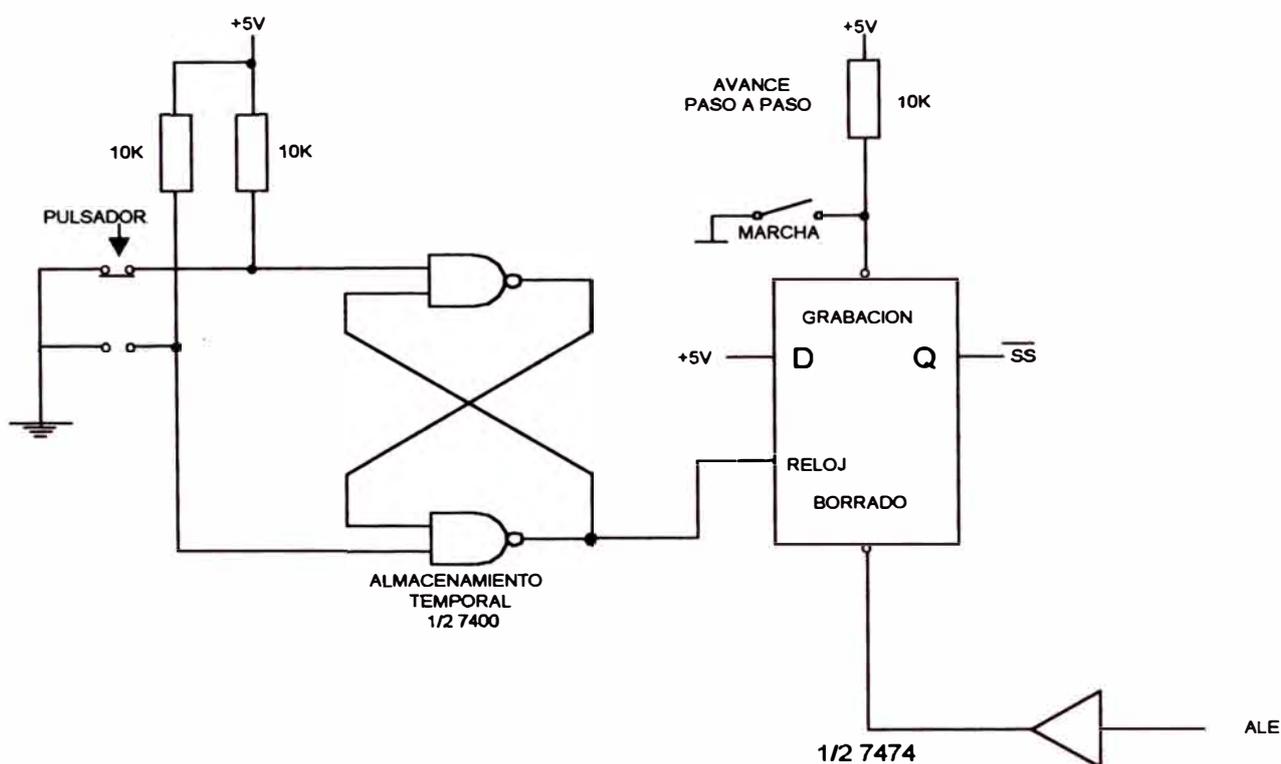


Fig. 10

Circuito de avance paso a paso

### Secuencia de funcionamiento

El 8048 en avance paso a paso trabaja en la siguiente forma:

- El microcontrolador recibe una petición de paro mediante la aplicación de un nivel bajo en SS.
- El microcontrolador responde deteniéndose en el tiempo de búsqueda de la siguiente instrucción. Si se está ejecutando una instrucción de dos ciclos cuando se recibe la orden de avance paso a paso, ambos ciclos son completados antes de parar el programa.
- El microcontrolador se da por enterado del estado de paro mediante la subida a uno de ALE. En este estado (que puede ser mantenido indefinidamente), la dirección

de la siguiente instrucción a ejecutar se encuentra presente en el bus y en la mitad del registro de E/S 2.

- Posteriormente, la entrada SS pasa a nivel uno lógico, reanudando su funcionamiento el microcontrolador para poder buscar la siguiente instrucción a ejecutar. La reanudación del funcionamiento es indicada por el microcontrolador poniendo a nivel cero la salida ALE.
- Para parar el microcontrolador en la siguiente instrucción, SS debe llevarse a cero otra vez, tan pronto como ALE pase a cero. En caso de que SS permanezca en estado alto, el microcontrolador seguirá en funcionamiento.

Para generar la señal SS se utiliza un biestable tipo D con borrado y puesta a uno. En el modo de marcha (microcontrolador funcionando) la entrada SS permanece a nivel uno ya que se mantiene la puesta a uno del biestable (tiene preferencia sobre el borrado). Para seguir avanzando paso a paso, la puesta 1 queda libre permitiendo a la señal ALE poner SS a 0 a través del borrado del biestable. El microcontrolador se encuentra ahora en el estado de parada. La siguiente instrucción es iniciada por la aplicación de un 1, por mediación del reloj en el biestable. Este 1 no aparecerá en SS a menos que ALE lo sea también, dejando libre el borrado del biestable. En el cambio de SS a 1, el microcontrolador comienza una instrucción a la vez que pone ALE a 0, borrando SS a través de la entrada de borrado y haciendo que el microcontrolador vuelva a entrar en el estado de parada.

### 2.3.17 Descripción de los terminales.

El encapsulado del 8048 y 8748 es de 40 patillas dispuestas en doble fila (DIP). A continuación explicaremos la función concreta realizada por cada una de ellas. Si no se indica lo contrario, cada entrada es compatible con dispositivos TTL y cada salida puede excitar cargas TTL.

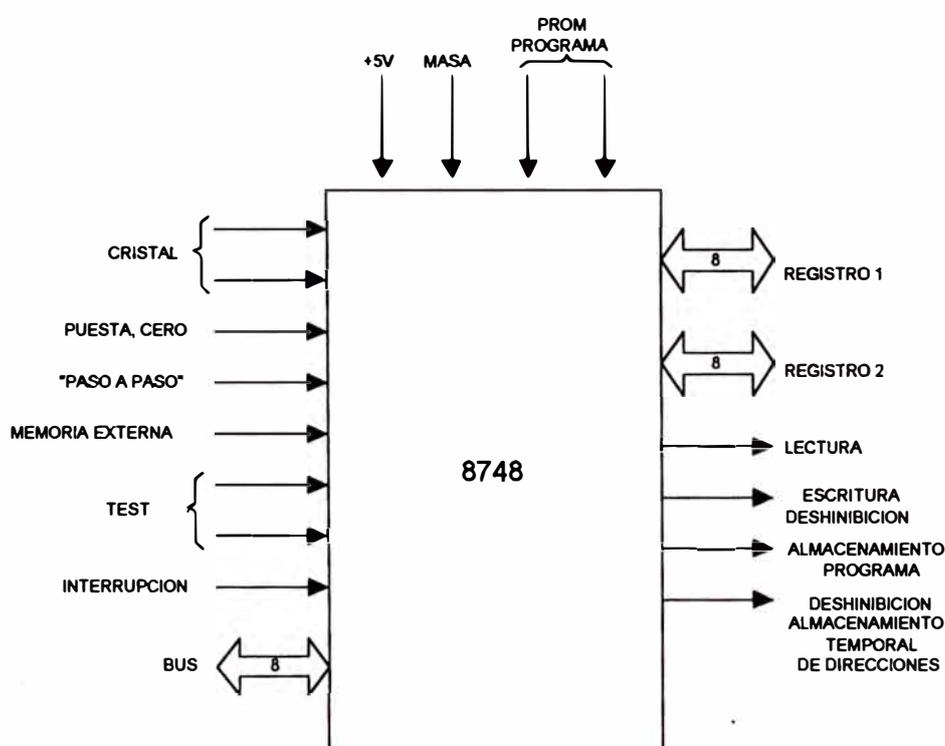


Fig. 11

Terminales del 8748

Desig.	Patilla	Función
Vss	20	Masa
VDD	26	Tensión de alimentación para programación; 25 V programando; 5V operando, tanto el ROM como EPROM.

Patilla de régimen de baja potencia en la versión 8048 ROM.

Vcc	40	Tensión de alimentación; 5 V operando y programando cuando se trata del 8748.
Prog	25	Patillas de entrada de impulsos de programación (25 V) durante las programaciones del 8748.
P10-P17	27-34	Registro de entrada/salida cuasibidireccional de 8 bits.
P20-P27	21-24 35-38	Registro de E/S cuasibidireccional de 8 bits.
D0-D7	12-19	Registro de E/S bidireccional que puede ser leído y escrito en sincronismo mediante las salidas de muestreo RD y WR. El registro de E/S puede almacenar estáticamente un información.

Contiene los 8 bits bajos del contador de programa durante un acceso a memoria externa y recibe la instrucción direccionada bajo el control de las señal PSEN.

También contiene la dirección y el dato durante la ejecución de una instrucción de almacenamiento de información en RAM externa, bajo el control de las señales ALE, RD Y WR.

T0	1	Terminal de entrada examinable mediante las instrucciones condicionales de
----	---	--

transferencia jTO y jNTO. T0 puede ser designado como salida de reloj usando la instrucción ENTO CLK, T0 se usa también durante la programación de la EPROM (8748).

T1	39	Terminal de entrada examinable mediante las instrucciones JT1 y JNT1. Asimismo, puede designarse como entrada del contador de sucesos mediante la instrucción STRT CNT.
INT	6	Entrada de interrupción. Inicia una interrupción siempre que éstas sean permitidas. Las interrupciones son inhibidas después de aplicar la puesta a 0.
RD	8	Salida de muestreo activada durante una lectura del BUS. Puede usarse para permitir la información de entrada al BUS, proveniente de un dispositivo exterior. Se utiliza como muestreo de lectura de memoria externa de datos.
RESET	4	Entrada de puesta a 0 para inicializar el $\mu$ c. También se utiliza durante la programación y verificación de la EPROM del 8748.
WR	10	Salida de muestreo, activa durante una escritura del BUS. Se usa como muestreo de escritura a memoria externa de datos.

ALE	11	<p>Permiso del registro de almacenamiento temporal de las direcciones. Esta señal se presenta durante cada ciclo y es utilizable como salida del reloj.</p> <p>El flanco de bajada de ALE introduce la dirección dentro de una memoria externa de datos o de programa.</p>
PSEN	9	<p>Permiso del almacenador del programa.</p> <p>Esta salida se presenta solo durante un acceso a memoria externa de programa.</p>
SS	5	<p>Entrada de avance paso a paso que se usa, junto a la señal ALE, para que el <math>\mu\text{c}</math> avance un solo paso en cada instrucción.</p>
EA	7	<p>Entrada de acceso externo que permite buscar el programa en la memoria externa. Es importante en simuladores y programas correctores y esencial para examen y verificación de un programa.</p>
XTAL1	2	<p>Entrada del cristal para el oscilador interno, o entrada para una fuente de frecuencia externa.</p>
XTAL2	3	<p>Entrada 2 del cristal.</p>

### 2.3.18 Set de instrucciones.

Las instrucciones tienen dos octetos como máximo y algunas, aproximadamente el 70%, un solo octeto de longitud. Además, todas las instrucciones tienen uno o dos

ciclos de tiempo de ejecución ( $2,5 \mu\text{s}$  ó  $5 \mu\text{s}$  si usamos un cristal de 6 MHz), y aproximadamente el 50% tiene un solo ciclo. Las instrucciones de doble ciclo incluyen todas las instrucciones inmediatas y las de E/S.

El microcontrolador puede efectuar operaciones aritméticas, tanto en código binario como en BCD.

### 2.3.18.1 Operaciones de transferencia de datos

Tal como se puede observar en la fig. 12.

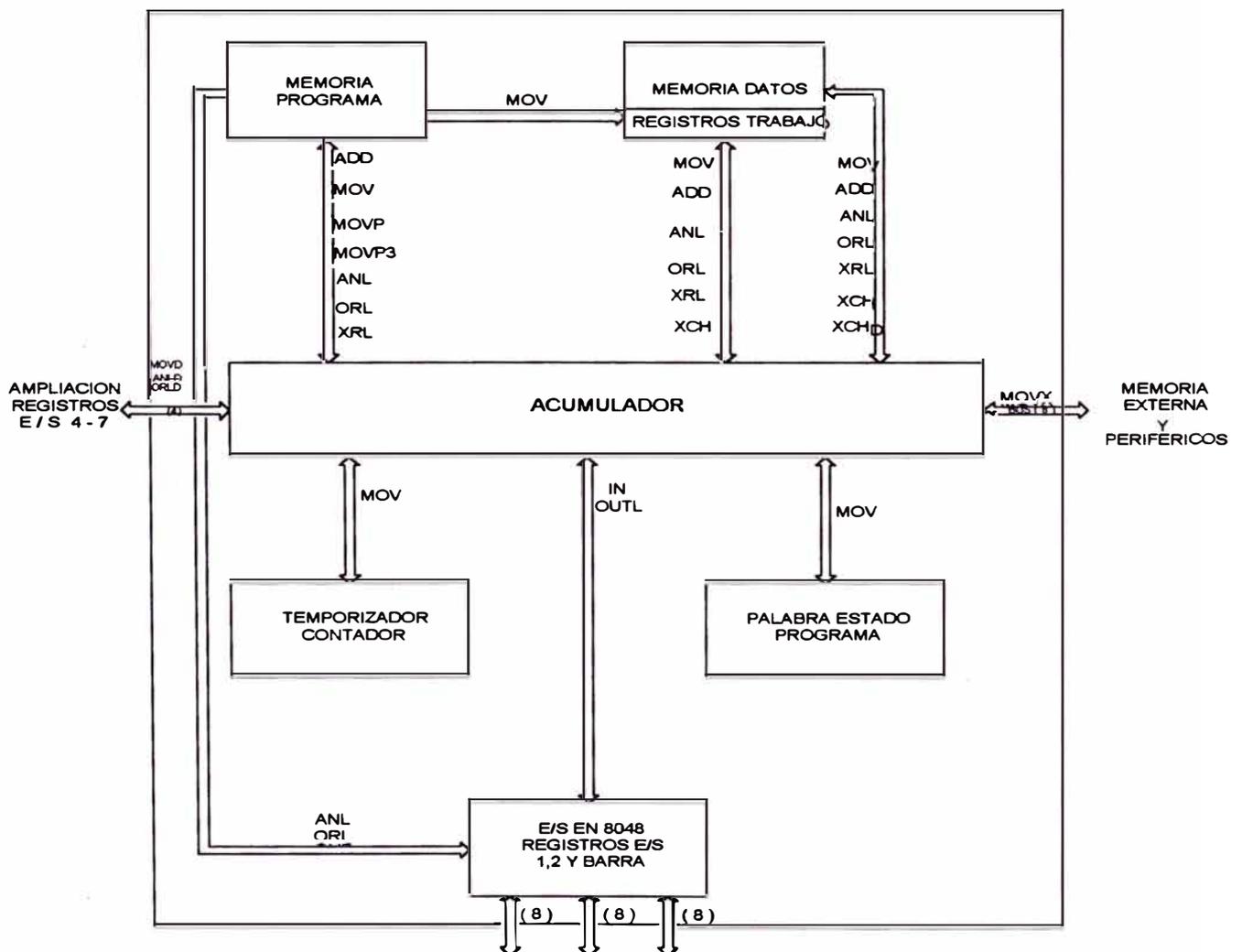


Fig. 12

Instrucciones de transferencia de datos

El acumulador de 8 bits es el punto de paso obligado para todas las transferencias de datos en el 8048.

Los datos pueden ser transferidos directamente entre los 8 registros de cada banco de registros de trabajo y el acumulador. El registro de destino es especificado por la propia instrucción. El conjunto de posiciones de memoria RAM interna está organizado como si se tratara de una memoria de datos, y es direccionada indirectamente a través del contenido de los registros R0 y R1 incluidos en los bancos de registro de trabajo.

Estos registros también son usados para direccionar indirectamente una memoria de datos externa en caso de que exista. Las transferencias hacia y desde la RAM interna requieren un solo ciclo, mientras que con la RAM externa se necesitan dos. Las constantes almacenadas en la memoria de programa pueden ser cargadas directamente en el acumulador y en los 8 registros de trabajo. Los datos pueden ser transferidos directamente entre el acumulador y el temporizador/contador interno, o entre el acumulador y la palabra de estado (PSW). Modificando PSW se puede alterar el estado de la máquina y restablecerla después de una interrupción o de una alteración del puntero de la pila, si se considera necesario.

#### **2.3.18.2 Operaciones con el acumulador**

A través del A pueden ser cursados, con o sin acarreo, datos inmediatos de memoria de registros de trabajo.

Asimismo, estos datos pueden ser operados mediante el A, por funciones AND, OR y O-exclusiva. Los datos pueden

ser transferidos hacia o desde el Acumulador, a partir de registros de trabajo, o de la memoria de datos. Los dos contenidos pueden ser cambiados en una sencilla operación.

Además, los cuatro bits menos significativos del acumulador pueden ser intercambiados con los cuatro bits menos significativos de cualquier posición en la memoria RAM de datos interna. Esta instrucción, junto con una instrucción que se encarga de permutar los 4 bits menos significativos con los 4 bits más significativos del acumulador, permite una fácil manipulación de las palabras de 4 bits, incluyendo números en código BCD. Para facilitar operaciones aritméticas en código BCD existe la instrucción de ajuste decimal. Esta instrucción es utilizada para corregir el resultado de una operación de suma binaria entre dos números en código BCD. Ejecutando el ajuste decimal en el resultado binario de una operación, se obtiene el resultado equivalente en código BCD.

Finalmente, el acumulador puede ser incrementado, borrado, complementado y desplazado 1 bit a derecha o izquierda con o sin acarreo.

Si bien en el 8048 no existen instrucciones de substracción, esta operación puede ser fácilmente realizada con tres instrucciones de un solo octeto, un solo ciclo. Un operando puede ser restado del acumulador y el resultado vuelto a introducir en el mismo mediante el complemento del acumulador, sumando el valor al acumulador y nuevo complemento del acumulador.

### **2.3.18.3 Operaciones con registros**

Se puede acceder a los registros de trabajo a través del acumulador tal como se ha explicado anteriormente, o bien pueden ser cargados de forma inmediata a partir de contenidos almacenados en la memoria de programa.

Además, pueden ser incrementados o decrementados o usados en lazos contadores utilizando el decremento y salto si la instrucción no es 0, tal como se explicará más adelante.

Toda la memoria de datos, incluyendo los registros de trabajo, es accesible mediante instrucciones con direccionamiento indirecto a través de R0 y R1.

### **2.3.18.4 Indicadores (FLAGS)**

En el 8048 existen cuatro indicadores accesibles para el usuario: acarreo, acarreo auxiliar, F0 y F1. El acarreo indica que ha habido rebasamiento de capacidad en el acumulador, mientras que el acarreo auxiliar indica rebasamiento de capacidad entre dígitos en código BCD y es utilizado en operaciones de ajuste decimal. Ambos, acarreo y acarreo auxiliar, son accesibles a través de la palabra de estado, siendo almacenados en la pila durante la ejecución de las subrutinas. F0 y F1 son indicadores de aplicación general utilizables en función de las necesidades del programa. Ambos indicadores pueden ser borrados o complementados y examinados mediante instrucciones de salto adicional. F0 es accesible también a través de la palabra de estado y puede ser almacenado en un registro junto con los indicadores de acarreo.

### 2.3.18.5 Instrucciones de bifurcación o de salto

La instrucción de salto incondicional es de dos octetos y permite saltos a cualquier posición del primer banco de 2K palabras de memoria de programa. Saltos al segundo banco de 2 K de la memoria (pueden ser directamente direccionables hasta 4 K), pueden realizarse primero mediante la ejecución de la instrucción de salto propiamente dicha. El límite de 2 K sólo puede ser superado mediante instrucciones de salto o de llamada a subrutina, es decir, el cambio de banco no se realiza hasta haber ejecutado una instrucción de salto.

Una vez seleccionado el banco de memoria, los siguientes saltos serán a dicho banco hasta que se ejecute otra instrucción de selección de banco de memoria. Una subrutina que se encuentre en el banco opuesto puede ser accedida mediante la instrucción de selección de banco de memoria seguida de otra de llamada a subrutina. Una vez completada la ejecución de la subrutina se retornará automáticamente al banco de origen y la siguiente instrucción de salto que se encuentre será de nuevo transferida al banco opuesto.

Los saltos condicionales pueden examinar las siguientes entradas y estados internos:

- Terminal de entrada T0.
- Terminal de entrada T1.
- Terminal de entrada INT.
- Acumulador a cero.
- Cualquier bit del acumulador.

- Indicador de acarreo.
- Indicador F0.
- Indicador F1.

Los saltos condicionales permiten una bifurcación a cualquier posición de una página de memoria (256 octetos) en ejecución. Las condiciones examinadas son valores instantáneos en el momento de ejecutarse un salto condicional. Por ejemplo, la instrucción de salto, si el acumulador está a cero, examina el acumulador de la misma, no un indicador de cero intermedio.

La instrucción de decremento de un registro y salto, si no es cero, combina una instrucción de decremento y otra de bifurcación para crear una instrucción muy útil en la implementación de lazos contadores.

Esta instrucción puede designar cualquiera de los 8 registros de trabajo y efectuar una bifurcación a cualquier dirección de la página que se esté ejecutando.

La instrucción de salto indirecto de un solo octeto permite acceder a cualquier posición de memoria basándose en el contenido del acumulador. El contenido del acumulador apunta a la posición de la memoria programa que contienen la dirección de salto. La dirección de salto, de 8 bits, está referida a la página de ejecución. Esta instrucción podría ser usada, por ejemplo, para acceder a cualquiera de las diferentes rutinas basadas en caracteres ASCII, los cuales son cargados en el acumulador. De esta forma, las entradas de teclas ASCII pueden ser usadas para iniciar diferentes rutinas.

### **2.3.18.6 Subrutinas**

El salto a una subrutina se ejecuta mediante la instrucción CALL. Dicha instrucción puede generar saltos incondicionales a cualquier posición de un banco de memorias de 2 K y de la misma forma saltar por encima del límite de 2 K. Dos instrucciones de retorno distintas determinan si se restablece o no el contenido de la PSW (cuatro bits mas significativos) al retornar de una subrutina.

### **2.3.18.7 Instrucciones del temporizador.**

El temporizador/contador de 8 bits puede ser cargado o leído a través del acumulador, bien cuando está parado, o bien cuando está contando. El contador puede ser activado mediante un reloj interno, realizando las funciones de un temporizador, o bien efectuar las funciones de contador de impulsos o temporizador a partir de un reloj externo aplicado a la patilla de entrada T1. La instrucción ejecutada determina que fuente de impulsos se utiliza. Una sola instrucción detiene el contador, independiente de que esté operando con una fuente de reloj interna o externa.

Además, dos instrucciones distintas permiten inhibir o desinhibir el impulso de interrupción del temporizador.

### **2.3.18.8 Instrucciones de control.**

Dos instrucciones permiten que la fuente externa de interrupciones pueda ser inhibida o desinhibida.

Inicialmente las interrupciones se encuentran desinhibidas y son automáticamente inhibidas durante la ejecución de una subrutina de interrupción, siendo otra

vez desinhibidas al finalizar la misma.

Existen cuatro instrucciones de selección de banco de memoria, dos para designar el banco de registros de trabajo efectivo a utilizar y dos para controlar los bancos de memoria de programa. Las instrucciones de cambio de registro de trabajo permiten al programador la inmediata sustitución por un segundo banco de 8 registros de trabajo del que tengan en uso en tal momento. Esto permite disponer de 16 registros de trabajo o bien puede ser usado como medio rápido para salvar el contenido de los registros cuando se produce una interrupción. El usuario tiene la opción de cambiar de banco cuando se produce una interrupción.

No obstante, si los bancos son cambiados, el de origen será automáticamente restituido durante la ejecución de una instrucción de retorno y restablecimiento del estado de PSW al final de la subrutina de interrupción.

Una instrucción especial desinhibe un reloj interno de frecuencia igual a un tercio de la del cristal, que tiene una salida por la patilla T0. Este reloj puede ser usado como aplicación general de un reloj en un sistema.

Esta instrucción será utilizada solamente al inicializar el sistema, ya que la salida de reloj puede ser inhibida únicamente mediante la aplicación de la puesta a cero del sistema.

#### **2.3.18.9 Instrucciones de entrada/salida.**

Los registros de E/S 1 y 2 son registros estáticos de E/S de 8 bits los cuales pueden ser cargados hacia o desde

el acumulador. Las salidas son almacenadas estáticamente, pero las entradas no son almacenadas y se hace necesario leerlas mientras están presentes. Además, datos inmediatos de la memoria de programa pueden ser sometidos a funciones AND u OR directamente con los registros de E/S 1 y 2, quedando el resultado en los registros de E/S. Esto permite realizar máscaras almacenadas en la memoria de programa, para seleccionar la puesta a 1 ó a 0 de bits individuales de los registros de E/S.

Los registros de E/S 1 y 2 están estructurados de forma que permitan la entrada de información por una patilla seleccionada, siempre que antes escribamos por programa un 1 en dicha patilla.

Un registro de E/S de 8 bits, llamado bus, puede ser accedido también a través del acumulador y puede tener salidas con almacenamiento temporal estático. En él se pueden realizar funciones AND y OR entre datos inmediatos de la memoria de programa directamente con sus salidas; no obstante este proceso es distinto al de los registros de E/S 1 y 2, ya que se necesita tratar los 8 bits del bus a la vez en todo momento, sean entradas o salidas.

Además de ser un registro de E/S estático, el bus puede ser usado como un registro de E/S bidireccional síncrono usando las instrucciones de transferencia externa que permiten acceder a la memoria de datos externa. Cuando estas instrucciones son ejecutadas, se genera el correspondiente impulso de lectura o escritura y se hace válido el dato únicamente durante ese tiempo. Mientras

no hay transferencia de datos al bus, éste se encuentra en estado de alta impedancia.

Los registros de E/S pueden ser expandidos externamente mediante la mitad del registro E/S constituido en la barra expansora de cuatro bits.

Los dispositivos expansores de E/S en este bus constan de cuatro de registros de E/S de cuatro bits, direccionables como registros de E/S del 4 al 7. Estos registros de E/S tienen sus propias instrucciones AND y OR así como instrucciones de transferencia para entrada o salida de datos. No obstante las instrucciones AND y OR del expansor combinan el contenido del acumulador con el registro de E/S seleccionado, mientras que los registros de E/S internos lo hacen directamente con un dato inmediato.

Dispositivos de E/S pueden también ser añadidos externamente usando el registro de bus de E/S como bus de expansión. En este caso, los registros de E/S son direccionados como si se tratara de la memoria de datos externa, encontrándose direccionado en el espacio de direcciones de la memoria de datos externa por los registros punteros R0 y R1.

#### **2.4 Organización de la memoria en los microcontroladores**

En los microcontroladores las memorias están organizadas en dos bloques que son:

- Memoria de programas y
- Memoria de datos.

### **2.4.1 Memoria de programas.**

Es una memoria de tipo ROM o EPROM, donde las capacidades varían dependiendo de la serie, tales como: en el caso del 8048 corresponde una ROM de 1024 bytes, en el caso del 8748 se tiene una EPROM de 1024 bytes, para el 8051 es de 4096 bytes, y en el 8052 es de 8192 bytes, estas memorias son internas pudiéndose extender estas capacidades mediante memorias externas.

### **2.4.2 Memoria de datos.**

Es una memoria para lectura y escritura, donde las capacidades varían dependiendo de la serie, tales como: 64 bytes para el 8048 y el 8748, 128 bytes para el 8051, estas memorias pueden ser extendidas hasta 64 K bytes de memoria de datos externa.

### **2.4.3 Expansiones en el microcontrolador 8048.**

Cuando las capacidades existentes en el microcontrolador 8048 no son suficientes, estos pueden ser directa y fácilmente ampliados en las siguientes áreas:

- Memoria de programa ( ROM o EPROM ) a 4096 bytes.
- Memoria de datos (RAM) a 320 bits.

#### **2.4.3.1 Ampliación de la memoria de programa.**

Las direcciones menores a la 1024 se direccionan internamente, generándose externamente solo la señal ALE.

A partir de la dirección 1024, el microcontrolador inicia automáticamente las direcciones de memoria de programa externa, y ocurre lo siguiente:

- Los 12 bits del PC se sacan a través del bus de datos a la mitad de la puerta 2.

- ALE nos indicará cuando es dirección, y la utilizaremos para almacenarla.
- PSEN indica que el microcontrolador trabaja con memoria externa.
- El bus de datos pasa al modo de entrada flotante y acepta el dato.

Para direccionar memoria superior a 2 K bytes, hay que ejecutar la instrucción de conmutación de barras de programa.

(SEL MB0, SEL MB1) seguido de una instrucción de salto CALL o JMP. Este dispositivo extiende el campo de instrucciones direccionables a más de 2 K bytes, y previene al usuario del desbordamiento inadvertido de esos 2 K.

#### **2.4.3.2 Ampliación de la memoria de datos.**

Empleando el circuito integrado 8155, se obtiene una expansión de 256 bytes de RAM, además de 22 líneas de E/S. Esta RAM, al igual que las puertas de E/S, precisa de las líneas RD, WR e IO/M.

Los pasos que se siguen en un ciclo de lectura/escritura son los siguientes:

- Los contenidos de R0 o R1 son sacados al bus de datos.
- ALE indica la dirección es válida, y su flanco de bajada es utilizado para guardar dicha dirección.
- Un impulso de RD (lectura) o WR (escritura) indica el tipo de proceso de datos. La salida de datos es válida durante el flanco de bajada de WR, y la entrada lo es en el RD.

- Los 8 bits del dato son ingresados o sacados a través del bus.

## **2.5 Control de periféricos, puertos de entrada/salida en los microcontroladores.**

Los puertos de E/S nos permiten la comunicación con una buena parte de los periféricos comerciales y con los que el usuario diseñe.

En el caso del microcontrolador 8048 se tiene 27 líneas de E/S agrupadas en 3 registros de E/S de 8 bits cada uno, teniendo como funciones de entrada, salida o registro bidireccional.

En el caso del 8051 y del 8052 se tiene 4 puertos bidireccionales, de 8 bits cada uno que permiten la lectura y escritura en el periférico correspondiente, las salidas están latcheadas, lo que posibilita mantener el dato indefinidamente hasta que sobrescriba la información original.

Otra característica importante es que los puertos pueden ser utilizados como buses de direcciones, datos y control, por lo que en estas circunstancias se dice que el microcontrolador trabaja como microprocesador.

### **2.5.1 El microcontrolador como microprocesador.**

Un microcontrolador puede ser utilizado como un microprocesador implementando la memoria RAM, EPROM y/o unidades de entrada/salida que el usuario considere mas adecuada a su aplicación. Naturalmente esto trae como consecuencia una reducción de su capacidad autónoma de controlar periféricos a través de sus puertos.

En el caso del microcontrolador 8051, los drivers de salida de los puertos P0 y P1 y los buffers de entrada del puerto P0 se pueden utilizar para acceder a la memoria externa del sistema. En estas condiciones, el puerto P0 se configura como salida del byte bajo del bus de direcciones ( A0 a A7 ), de un bus que consta de 16 bits ( capacidad de direccionamiento de 64 K ), multiplexado en el tiempo como bus de datos bidireccional ( D0 a D7 ). El puerto P2 se configura como salida del byte alto del bus de direcciones ( A8 a A15 ). De esta manera el microcontrolador se configura a modo de CPU de un sistema externo con unas pocas capacidades de expansión definibles por el usuario.

Naturalmente, en estas condiciones, le queda como puerto integro, para el control de los periféricos, el puerto P1 y parte del puerto P3, puesto que hay señales que se utilizan para el control del sistema (bus de control ).

### **2.5.2 Ampliación de las líneas de entrada/salida.**

Una de las formas consiste en la utilización de periféricos programables, como por ejemplo:

- 8214; codificador de prioridades de interrupción.
- 8251; interface de comunicaciones en serie.
- 8255; E/S de aplicaciones generales.
- 8279; interface para teclado y displays.
- 8253; temporizador.

### **2.6 Modelo de programación de microcontroladores.**

Se entiende por modelo de programación al conjunto de

registros accesibles por el usuario que junto al set de instrucciones y los modos de direccionamiento constituyen las herramientas imprescindibles para el programador.

En el caso del microcontrolador 8048, las instrucciones tienen 2 bytes como máximo, teniendo el siguiente grupo de instrucciones:

- Instrucciones de transferencia de datos.
- Instrucciones aritméticas.
- Instrucciones lógicas.
- Instrucciones de bifurcación o salto.
- Instrucciones del temporizador.
- Instrucciones de control.
- Instrucciones para subrutinas.
- instrucciones de entrada/salida.

En el caso del microcontrolador 8051 se tiene el siguiente grupo de instrucciones:

- Instrucciones aritméticas.
- Instrucciones lógicas.
- Instrucciones para transferencia de datos en:
  - RAM interna.
  - RAM externa.
- Instrucciones para el tratamiento de tablas.
- Instrucciones booleanas.
- Instrucciones de salto.

## **2.7 Microcontroladores de 16 bits.**

La familia MCS-96 está basada en la arquitectura del 8096 que ofrece un nivel de prestaciones avanzadas con unas evolucionadas unidades de entrada/salida y distintos

periféricos integrados en la misma pieza de silicio junto con una potente CPU de 16 bits. Estos microcontroladores suplen, en sistemas de control complejo, las soluciones que requieren múltiples chips. Para desarrollar aplicaciones en este entorno, INTEL suministra soportes como lenguajes de alto nivel, un ensamblador y hardware de emulación. Incluso entre las herramientas que ofrece el fabricante se encuentra un software que traslada programas escritos en instrucciones de los microcontroladores de 8 bits a instrucciones de microcontroladores de 16 bits.

La familia MCS-96 presenta las siguientes características:

- 232 registros archivo. Funcionalmente pueden considerarse como acumuladores.
- Arquitectura dirigida al proceso registro a registro, evitando el cuello de botella que producen las arquitecturas de un único acumulador.
- Convertidor A/D de 10 bits con la inclusión del circuito de captura y mantenimiento (S/H Sample and hold).
- 20 fuentes de interrupción.
- Salida de pulso modulado en anchura ( convertidor D/A).
- Tiempo de ejecución programable en EPROM.
- Subsistema de E/S de alta velocidad.
- Puerto serie de comunicación full duplex.
- Protección de los programas en ROM o EPROM.
- Bus configurable dinámicamente para trabajar con 8 o 16 bits.

**CAPITULO III**  
**ANÁLISIS DE LOS PRINCIPALES PROBLEMAS DE ESTABILIZACIÓN DE**  
**VOLTAJE Y SISTEMA DE SOLUCIONES.**

**3.1 Problemas en la línea de alimentación.**

Entre los problemas que se presentan en la línea de alimentación de la corriente eléctrica, se puede mencionar los siguientes:

- Sobretensiones; son voltajes altos mayores de 220 voltios, que duran mas de un ciclo.
- Bajas de tensión; Son voltajes bajos suministrados en la línea de energía eléctrica, que duran minutos y horas.

En nuestro país esto sucede con mucha frecuencia en las llamadas horas punta y en provincias en zonas con deficiente generación y suministro eléctrico.

- Caídas de tensión; Es el caso opuesto a la sobre tensión.
- Falla de línea de alimentación, apagones; Es el corte súbito del suministro eléctrico, y puede durar minutos, horas y hasta días.
- Picos; Son altos voltajes generados por elementos externos en una fracción de segundo.
- Ruidos de RF; Es un término genérico empleado para definir a varios tipos de impulsos de alta frecuencia que existen en las ondas sinusoidales normales.

Distorsiones armónicas; Son deformaciones de la onda sinusoidal.

### **3.2 Causas y efectos de los problemas de estabilización de voltaje.**

Entre las causas y efectos producidos por los problemas de estabilización de voltaje, se pueden mencionar los siguientes:

- Sobretensiones; como causas son originados por dispositivos conectados a la línea de alimentación que se apagan repentinamente y que consume gran cantidad de energía.

Como efectos; produce el calentamiento general de los equipos electrónicos, causando desperfectos.

- Bajas de tensión; originados en la planta, una carga excesiva en la planta o consumo en las horas punta.

Como efectos, pueden dañar los equipos electrónicos, ya que se tiene la operación forzada de los equipos.

- Caída de tensión; se produce como falla momentánea por efecto de operación en la planta, operaciones de conmutación o arranque de grandes motores en las fábricas.

Como efectos pueden causar daños en los equipos electrónicos, por ejemplo en las computadoras puede producir fallas en la fuente de alimentación, disminución de las velocidades en el disk drive y discos duros .

- Falla de línea de alimentación; son causados por defectos en las líneas de alimentación de la energía

eléctrica, y posibles fallas técnicas en la planta de generación de la energía eléctrica.

Como efectos produce la interrupción de la operación de los equipos electrónicos.

- Picos; causados por operaciones de conmutación, como efecto pueden también dañar a los equipos electrónicos.
- Ruidos de RF; causados por la operación de los soldadores, transmisión de radiodifusión.

Como efecto; produce un funcionamiento irregular de los equipos electrónicos, interferencia en las comunicaciones.

- Distorsiones armónicas; son producidos por las cargas no lineales conectadas a la línea de alimentación.

Como efecto, afectan a otros equipos conectados a la línea de alimentación.

### **3.3 Solución a los problemas.**

Como mecanismos de solución se puede mencionar los siguientes:

- Para los problemas de alto voltaje, bajo voltaje y caídas de voltaje como solución se plantea el uso de estabilizadores.
- Para el problema de falla de energía (apagones), como solución se plantea el uso de las UPS.
- Para el problema de los picos, se recomienda usar los supresores de pico.
- Para los problemas de RF, se recomienda el uso de los filtros como una posible solución al problema.

### **3.4 Alternativas de solución a las variaciones de voltaje**

#### **3.4.1 Estabilizadores tradicionales**

Consideremos así a los llamados estabilizadores híbridos y los de estado sólido, existentes en el mercado cuya filosofía de funcionamiento se basa en que estos equipos miden la entrada que va a una serie de comparadores, y al bajar o subir el voltaje de entrada conmuta a un relé u otro relé para hacer la compensación respectiva, en el caso de los llamados estabilizadores híbridos, y conmuta a un triac u otro triac en el caso de los llamados estabilizadores de estado sólido.

En estos casos no controlan el rango de entrada, utilizan ese rango solo para conmutar los relees de salida, pero generalmente cuando se coloca una carga a un estabilizador ocurre una variación del voltaje de salida por efecto de carga, por lo tanto este tipo de estabilizadores no controlan este problema, porque solo hacen cambios con la entrada y como la entrada no cambia el equipo no se da cuenta del problema.

#### **3.4.2 Estabilizador controlado por microcontrolador**

La filosofía de funcionamiento de este equipo consiste en mantener constante el voltaje de salida, para lo cual se toma el voltaje de salida y se realimenta al circuito de control, constituido fundamentalmente por el microcontrolador, donde dependiendo de que el voltaje de salida esta muy bajo o esta muy alto, se ejecuta una rutina de salida, cuya función es la de conmutar un par de triacs, que a su vez conmutan a un determinado tab de un

autotransformador, que tiene 8 devanados en el primario, 4 de los cuales corresponden a los tabs de menor peso y los 4 restantes corresponden a los tabs de mayor peso, de tal manera que combinando entre pares de tabs se logra la compensación respectiva para lograr el objetivo de mantener el voltaje de salida constante.

Quiere decir que este dispositivo, en comparación con los tradicionales realiza los cambios con la salida, lo cual garantiza de que cualquier variación del voltaje de salida por efecto de carga inmediatamente, será comunicado al microcontrolador para su respectiva solución, el voltaje de entrada se toma en cuenta solo para garantizar de que el estabilizador funcione adecuadamente dentro del rango de las variaciones del voltaje de entrada para el cual fue diseñado.

### **3.4.3 Análisis de las alternativas y elección de la mejor**

La primera alternativa, cuyo funcionamiento se basa de que la conmutación se hace con la entrada tiene ciertas desventajas ya que es común observar que cuando alimentamos una carga se produce variaciones del voltaje de salida por efectos de carga, y en casos de que no exista variaciones en la entrada no se producirá las correcciones respectivas.

La segunda alternativa, cuyo funcionamiento se basa de que la conmutación se realiza con la salida y además el uso de un dispositivo innovador como el microcontrolador proporcionan una ventaja en la compensación de cualquier variación del voltaje de salida.

Por lo tanto, considerando todos los aspectos presentados en los párrafos precedentes, se elegirá para el desarrollo y montaje de la segunda alternativa.

**CAPITULO IV**  
**DISEÑO Y CONSTRUCCION DEL SISTEMA PROPUESTO ,SIMULACION Y PRUEBAS DE FUNCIONAMIENTO.**

**4.1 Características y especificaciones técnicas.**

- Basado en microcontrolador.
- Totalmente estado sólido con conmutadores controlados de silicio ( Triacs ).
- Conmutación cuando la corriente pasa por cero, lo que elimina la emisión de interferencia electromagnética.
- Detección de paso por cero de voltaje de salida para sincronizar el muestreo y retención del valor del voltaje de salida lo que permite una respuesta de reacción a los cambios del voltaje de salida de medio ciclo (aproximadamente de 8.3 ms. ).
- Protección contra subvoltaje de 175 volt. Y sobrevoltaje de 260 volt. de la entrada.
- 3 indicadores luminosos leds muestran el estado del voltaje de entrada.
- Voltaje de entrada muy bajo.
- Voltaje de entrada normal.
- Voltaje de entrada muy alto.
- Reinicio del sistema en forma selectiva:  
Manual o automática:
- Manual.- Se podrá reiniciar el sistema presionando un pulsador cada vez que se desee reiniciar el

sistema luego de una parada.

Automática.- El sistema se reinicia en forma automática después de 5 segundos (temporizador) de encontrarse el voltaje de entrada en un rango aceptable de 175 volt. a 260 volt.

- El circuito de control está basado en un microcontrolador que toma las decisiones en base a las señales dadas por los dispositivos periféricos. La etapa de control se encuentra aislada de la etapa de potencia mediante fotoacopladores, lo que reduce la probabilidad de fallas y hacer mas seguro un servicio de mantenimiento y/o reparación.

Filtro de línea contra interferencia de alta frecuencia.

- El sistema es modular, el uso de conectores permite un cómodo ensamblaje y en caso de fallos una rápida sustitución del módulo defectuoso.

8 Triacs dispuesto en 4 tomas superiores y 4 tomas inferiores en el primario de un transformador, permiten obtener 16 combinaciones de voltajes, y una precisión de 1.5 % como máximo con respecto a 220 volt. o 110 volt.

- Fusible externo para la protección del equipo.
- Las cargas pueden ser capacitivas, inductivas, resistivas o reactivas (combinación de las anteriores).
- El sistema tiene un circuito de protección contra falla propia, inhabilitándose e indicando mediante el flatcheo del "led" normal.

#### **4.2 Diagrama de bloques**

Antes de entrar en detalles, es conveniente tener una

visión en conjunto por bloques ver fig.13 a continuación describiré cada bloque.

El circuito de sensado de voltaje de entrada consiste en un transformador que toma el voltaje de entrada que luego ingresa a dos niveles de comparación cuyas salidas son  $V_{max}$ . y  $V_{min}$ . con el cual se tiene el rango de variación del voltaje de entrada, sobre el cual el estabilizador funciona adecuadamente las salidas  $V_{max}$ . y  $V_{min}$ . de los comparadores van a los pines P2.6 y P2.7 del microcontrolador respectivamente.

El circuito de sensado de la corriente que pasa por los Triacs, consiste en un transformador de corriente, que toma la corriente que pasa por los Triacs y luego ingresa a un comparador cuya salida va al pin T1 del microcontrolador, esta información permite al microcontrolador activar al Triac justo en el instante preciso.

El banco de Triacs de menor peso, esta formado por 4 Triacs que conmutará a los 4 devanados de menor peso del primario del autotransformador, los gates de los triacs serán activados mediante dispositivos fotoacopladores que serán controlados por un decodificador.

El banco de triacs de mayor peso, esta formado por 4 triacs que conmutará a los 4 devanados de mayor peso del primario del autotransformador, los gates de estos Triacs serán activados mediante dispositivos fotoacopladores que serán controlados por otro decodificador.

El autotransformador, está construido con 8 tomas en el

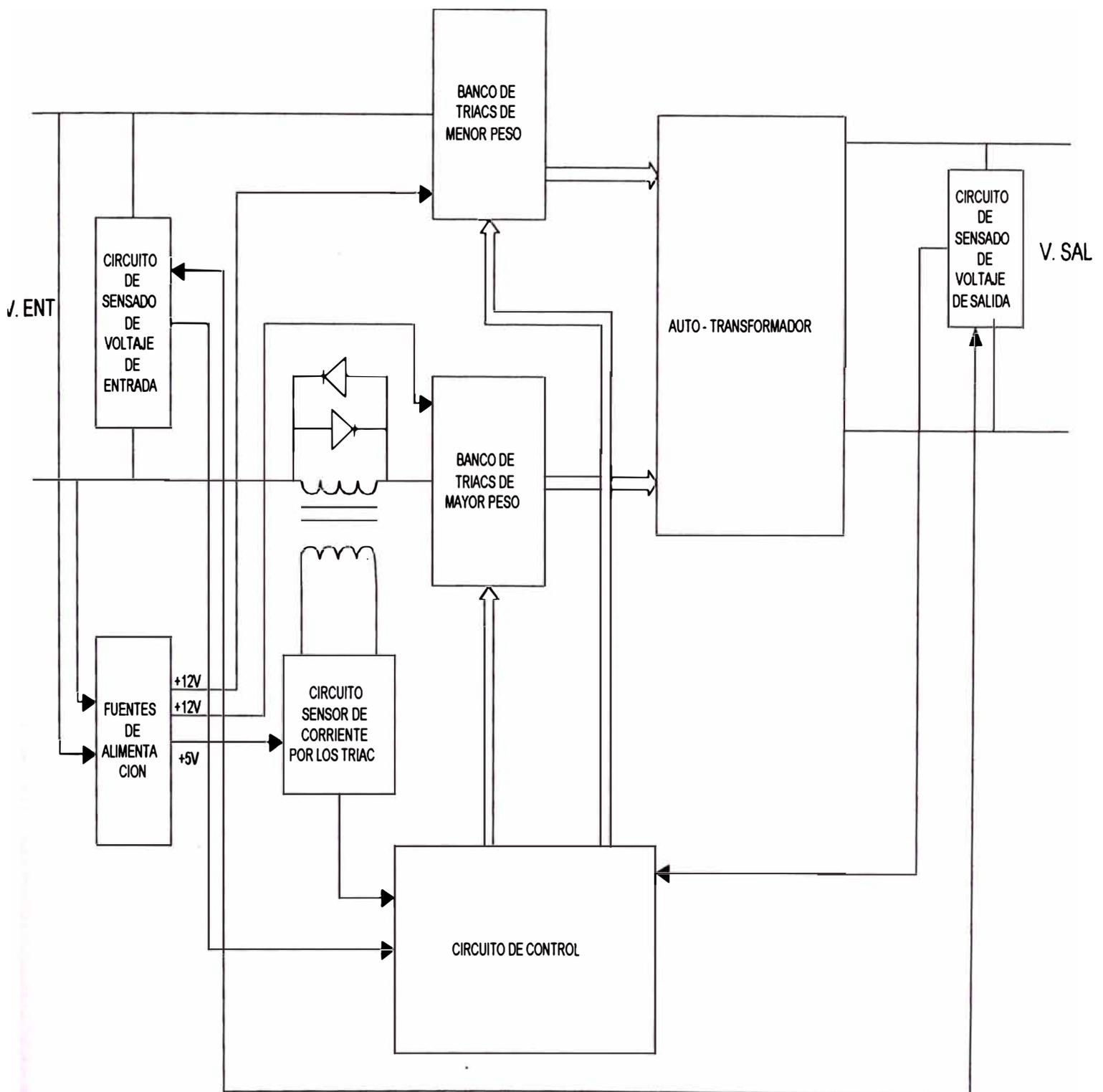


Fig. 13

Diagrama de bloques del estabilizador de voltaje usando microcontrolador

primario, distribuidas en 4 tomas de menor peso y 4 tomas de mayor peso para obtener 16 combinaciones entre estas.

El circuito de sensado del voltaje de salida, consiste en un transformador, con el cual se toman los voltajes de salida de 220 voltios y se baja a 18 voltios, que luego se rectifica e ingresa a un comparador de 2 niveles cuyas salidas  $V_h$  y  $V_l$  van a los pines P2.4 y P2.5 del microcontrolador respectivamente, esta parte constituye el bloque de realimentación del sistema.

El sistema de control esta formado por el microcontrolador que recibe la información de la señal de voltaje de salida, además la información de la corriente que pasa por los triacs, con estas dos informaciones se tienen los datos suficientes para realizar la conmutación en el instante preciso, además recibe la información del voltaje de entrada que da el rango sobre el cual va operar adecuadamente el estabilizador.

#### **4.3 Filosofía de diseño**

El voltaje de salida del sistema se realimenta al circuito de control, mediante un transformador de 220V/18V; la corriente que pasa por los triacs a través del paralelo de 2 diodos en contraposición y el transformador; entonces con éstas dos informaciones, el voltaje de salida y la corriente por los triacs se tienen los datos necesarios para hacer la conmutación en el instante preciso.

Para hacer la conmutación no tiene porque estar midiéndose el voltaje de salida y el momento de hacer

dicha conmutación es cuando la corriente por los triacs pasa por cero.

#### **4.4 Diseño del autotransformador**

Con el sensado del voltaje de entrada se tiene un margen de medición de los voltajes máximo y mínimo dentro del cual el estabilizador va funcionar adecuadamente, con este margen de entrada y 16 combinaciones.

Considerando la variación de voltaje de entrada entre 175 voltios y 255 voltios con este margen amplio y las 16 combinaciones se calcula la variación del voltaje de salida como la división de la diferencia de los valores máximo y mínimo de la entrada entre las 16 combinaciones, osea  $80/16$ , lo cual nos da un delta igual a 5 voltios, como este delta es la diferencia entre toma y toma consideremos la mitad, con lo cual el voltaje de salida lo podemos establecer en  $V_{sal}=220\pm 1/2(\text{delta})$  osea  $V_{sal}=220\pm 2.5V$ , donde delta es igual a 5 voltios, que es el margen al cual va variar el voltaje de salida.

Mediante un programa en lenguaje de alto nivel que se detalla en la sección del software se calcula la salida mas cercana a 220 voltios de un autotransformador con 8 tomas en el primario dispuestas en 4 tomas de menor peso y 4 tomas de mayor peso, con lo cual obtenemos las 16 combinaciones, con el delta del voltaje hallado se calcula las tomas de menor peso en 250V, 245V, 240V, y 235V las tomas de mayor peso se calculan considerando que tienen 4 veces el peso que la diferencia de las tomas de menor peso, resultando 0V, 20V, 40V y 60V, las tomas de mayor

peso son asignadas a  $V_0=0$  volt.  $V_1=20V$ ,  $V_2=40$  volt. y  $V_3=60$  volt. y las tomas de menor peso a  $V_4=235$  volt.,  $V_5=240$  volt.,  $V_6=245$  volt. y  $V_7=250$  volt.

Corriendo el programa mencionado anteriormente se puede determinar que combinación da el voltaje mas cercano a 220 Volt. para una variación de voltaje de entrada entre 175 volt. a 255 volt., con estas características establecidas se construye el autotransformador que tiene 8 tomas en el primario distribuidas en 4 tomas en un sector y 4 en el otro sector, para obtener 16 combinaciones, una de estas combinaciones tiene que ingresar a una ventana y el microcontrolador lo que hace es medir esto y comparar con 2 niveles.

#### **4.5 Diseño del circuito de sensado del voltaje de salida**

El voltaje de salida se baja de 220 voltios a 18 voltios mediante un transformador, se rectifica mediante los diodos D1-D4 y luego a través de la resistencia R2 en serie con un potenciómetro P1 en paralelo con un condensador C1; el voltaje del condensador C1 es tomado y va a un sistema de comparación de 2 niveles, este sistema de comparación es diseñado usando los OPAMP LM 324 y tomando en cuenta los valores de voltaje de salida mínimo y voltaje de salida máximo obtenido al correr el programa en Basic que se detalla en la sección del software, estos valores son  $V_s$  min. igual a 217.5556 voltios y  $V_s$  max. igual a 224.4 voltios.

Tomando voltaje para nivel máximo de 2.24 voltios correspondiente a 224 voltios y otro voltaje para nivel

mínimo de 2.17 voltios correspondiente 217 voltios, con los cuales se fijan estos 2 niveles de comparación, osea lo que se quiere es que la salida esté dentro de éste rango.

El condensador C1 tiene una capacidad de valor lo suficientemente pequeña para que no haya un voltaje de continua que se demore demasiado tiempo sino que debe responder a las variaciones rápidas de voltaje.

La onda rectificada va subir o bajar dependiendo del voltaje de salida, si el voltaje de entrada varía conservando las mismas tomas, entonces el voltaje de salida también varía. La señal de voltaje en el condensador, es una especie de carga y descarga pero como se toma en tiempos pequeños, actúa como un filtro pasa bajo, si la capacidad del condensador C1 es grande más atenuada será la amplitud de la señal de voltaje en el condensador lo cual va tender a ser una continua y va tardar en responder mientras que si la capacidad del condensador es más pequeña se tiene una mayor ondulación que es lo que se necesita para tener una respuesta rápida a las variaciones de voltaje.

Ajustamos mediante un potenciómetro para que la onda de voltaje del condensador se coloque dentro del rango de comparación considerado y moviendo con un variac obtenemos la salida a 220 voltios con el cual tenemos un nivel y como ya se conoce los dos niveles a comparar, entonces se generan dos ondas en las salidas de los comparadores llamemos VH y VL a estos niveles que resultan de la

comparación.

La salida VH del comparador va al P2.4 y la salida VL del otro comparador va al P2.5 que corresponde al puerto 2 del microcontrolador.

Cuando la onda de voltaje en el condensador supera el nivel máximo de la comparación, entonces en la salida VH del comparador tenemos una onda de nivel alto, cuando la onda de voltaje en el condensador baja del valor mínimo de comparación en la salida VL del otro comparador se tiene una onda de nivel alto como se ve en la fig. 24, las ondas se van alternando.

Cuando el voltaje de salida está centrado alrededor de 220 voltios, osea está dentro del rango, éstas ondas tanto el VH y el VL son iguales, solamente están desfasados.

Además el transistor Q1 es muy importante, el cual cuando el voltaje es cero está en corte y apenas pasa de cero ya se satura, quiere decir que el transistor en un lapso de tiempo muy pequeño genera pulsos llamemos pulso de paso por cero ó VP0 a este pulso, el diodo D5 cumple con la finalidad de hacer llegar una señal rectificadas pero no filtrada a la base del transistor Q1.

Las ondas VH, VL y el pulso de paso por cero, se pueden observar en la fig. 24, obtenida mediante las simulaciones realizadas con SPICE.

Cuando el voltaje de salida comienza a subir la onda de salida VH del comparador comienza a ensancharse y la onda de salida VL del otro comparador comienza a adelgazarse, si el voltaje de salida sigue subiendo llega un momento en

que invade la zona de paso por cero y aún sigue ensanchándose más, mientras que el otro cada vez se adelgaza más.

Como el microcontrolador solo lee el VH y VL en el flanco de subida del pulso de paso por cero (VP0), osea solo interesa justo en el momento de paso por cero cual de las ondas está en estado alto; si estuviera en estado alto el VH entonces el voltaje de salida está muy alto; si estuviera en estado alto el VL entonces el voltaje de salida está muy bajo, esto es muy importante porque a partir de esta información el microcontrolador se da cuenta de lo que está sucediendo y tomará la acción correctiva correspondiente.

Los pulsos son suficientemente pequeños, para que estos se ensanchen o angosten, dependiendo de la magnitud de los voltajes de salida, osea que el tiempo de carga del condensador debe ser menor que 8.3 ms., que es la duración del semi periodo, todo esto funciona como un muestreo y una retención osea que el microcontrolador muestrea el voltaje de salida en tiempos fijos, cuando el voltaje pasa por cero es donde se muestrea de como esta el voltaje de salida, sólo hay efecto en el flanco de subida del pulso de paso por cero Vp0 que va al pin T0 del microcontrolador que es el que detecta el paso por cero, por eso en el programa existe una subrutina que indique saltar a la dirección alfa si T0 es igual a uno, si la subrutina ya encuentra el pulso en alto no hace nada, espera a que baje el Vp0 pero como alfa esta en el mismo

sitio recién cuando baja pasa a la siguiente dirección, saltar a la dirección beta si T0 es igual a cero y esta esperando a que suba y justo cuando sube ya no se cumple esta condición y allí recién lee el puerto 2 por donde ingresa el valor de VL y VH hacia el acumulador del microcontrolador, el pin correspondiente al VH es el P2.4 y el pin correspondiente al VL es el P2.5 del microcontrolador como se ve en la fig. 16.

Allí se va verificar si alguno de los dos valores esta en un nivel alto, es por que el voltaje de salida se ha salido del rango, es decir al momento de cargar el acumulador entonces ya se sabe si el voltaje de salida esta muy alto o muy bajo, si cualquiera de ellos esta en alto significa que hay que hacer una conmutación, para ello existe una subrutina de corrección o sea la subrutina de salida que es la que va ha corregir cambiando los tabs del autotransformador.

#### **4.6 Diseño del circuito de sensado de la corriente por los triacs.**

Una vez conocida la información acerca de la corrección mediante la subrutina de salida, o sea como ya se sabe que se tiene que subir o bajar de tabs del autotransformador, en ese momento se debe verificar la corriente que pasa por los Triacs, para ello esta corriente es derivada a través de un transformador, una resistencia R5 y dos diodos D1 y D2 que son el 1N4848 en paralelo pero invertidos y va hacia un amplificador operacional LM393, cuya salida constituye la onda PI0 y va hacia el pin T1 del

microcontrolador.

Uno de los terminales del secundario del transformador en lugar de poner a tierra se está colocando a un voltaje de referencia de 2.5 voltios, ya que se tiene un divisor de voltaje respecto a 5 voltios que es el voltaje de alimentación como se observa en la fig. 14, la razón de colocar el par de diodos, paralelos pero invertidos es que en este tipo de amplificadores operacionales los voltajes aplicados en la entrada deben estar dentro del rango de los voltajes de polarización sino la salida del amplificador operacional se distorsiona, como este amplificador operacional esta polarizado con 5 voltios y tierra entonces los voltajes de entrada solo deben estar en el rango de 0 voltios a +5 voltios, por eso para garantizar esta condición se colocan los dos diodos en paralelo pero invertidos ya que en un momento tenemos 2.5 voltios mas 0.7 voltios correspondiente al diodo dando un resultado de 3.2 voltios, y en otro momento tendremos 2.5 voltios menos 0.7 voltios dando como resultado 1.8 voltios ambos valores ya sea 3.2 voltios y 1.8 voltios están dentro del rango del voltaje de polarización por lo tanto no habrá distorsión en la salida del amplificador operacional, si no colocamos el voltaje de referencia en uno de los terminales del secundario del transformador tendríamos 0 voltios mas 0.7 voltios en un caso y 0 voltios menos 0.7 voltios en otro caso donde uno de los valores que es el - 0.7 voltios sale fuera del rango de polarización del amplificador operacional por lo tanto se

distorsiona la salida.

El valor de las resistencias R27 y R28 para el divisor de voltaje debe ser de tal manera que no sea tan alto comparable con la impedancia de entrada del amplificador operacional; el condensador C9 de 10 micro faradios de capacidad se coloca porque en un divisor de voltaje que se toma como referencia, siempre se coloca un condensador para reducir probables perturbaciones de la fuente.

#### **4.7 Diseño del circuito de banco de triacs de menor peso**

Este circuito está formado por 4 triacs, cuya función es conmutar a un determinado tab de menor peso del primario del autotransformador con lo cual se logra la corrección a alguna anomalía detectada en el voltaje de salida.

El gate de estos triacs se activan mediante fotoacopladores, ya que según las normas la parte de potencia debe estar separado de la parte de control porque al existir esta separación tranquilamente se puede manipular la tarjeta de control lo cual permite una mayor seguridad para realizar mediciones, porque la tierra del circuito no está a un alto voltaje.

La separación de la etapa de potencia de la etapa de control también puede ser realizada mediante transformadores, el uso de un fotoacoplador o un transformador depende de la capacidad de la corriente de los triacs, los fotoacopladores van hasta unos 100 mA., como máximo, pero si el triacs es de más potencia entonces se usaría transformadores como un medio de

separación entre las etapas de potencia y de control, pero el fotoacoplador es más barato con respecto al transformador.

Cada fotoacoplador es activado por el colector de un transistor, donde además se coloca un led para indicar cual de los triac de menor peso está funcionando en un momento dado, estos leds van a permitir un mecanismo de mantenimiento del equipo, en el caso de que estén todos apagados o estén prendidos más de uno permitirá detectar fallas ya que en condiciones normales de trabajo solo debe estar funcionando un solo triac del banco del menor peso, lo cual implica estar encendido un solo led, la base de estos transistores son activados por las salidas de la U2B que constituye 1/2 de un circuito integrado TTL 74LS139 que es un doble decodificador de 2 a 4 líneas, este decodificador tiene un pin de habilitación identificado con la letra G, que es activo con cero lógico y que viene del P1.7 del puerto 1 del microcontrolador, la U2B tiene sus entradas A y B comandados por P1.0 y P1.1 del puerto 1 del microcontrolador, donde para una combinación binaria de estas entradas se activa solo una salida de las 4 salidas posibles identificadas por Y0 hasta Y3, cada una de estas salidas activan la base de los transistores que controlan a los fotoacopladores y a su vez estos a los gates de los triacs, por consiguiente al no tener mas de una salida activa en el decodificador no es posible tener 2 triacs de este banco funcionando a la vez.

La combinación de la resistencia con el condensador en

serie y en paralelo con el Triac es para limitar los cambios bruscos de tensión en los bornes del Triac, los fabricantes de Triac especifican el  $dv/dt$  máx. con lo cual indican de que la variación de voltaje en los bornes de un Triac no deben superar el  $dv/dt$  máx. sino el Triac se cruza.

La resistencia en serie, es una resistencia de valor pequeña de 0.2 Ohmios a 5 W. se coloca con la finalidad de limitar la corriente de que en el caso de alguna perturbación o transitorio se pueda disparar dos Triacs contiguos, al mismo tiempo y el Triac no se quemé, esta resistencia va hacia un determinado tab del primario del autotransformador.

Los fabricantes de Triacs también dan especificaciones de la corriente máxima, que generalmente es de unas 10 veces la corriente nominal del Triac.

#### **4.8 Diseño del circuito de banco de triacs de mayor peso**

El diseño de esta parte es similar al diseño del circuito de banco de triacs de menor peso que fue tratado con bastante detalle, el circuito de banco de triacs de mayor peso está formado también por 4 triacs, cuya función es conmutar a un determinado tab de mayor peso del primario del autotransformador, ya que la corrección de alguna anomalía detectada en el voltaje depende de la conmutación de un triac del banco de menor peso en combinación con otro triac del banco de mayor peso, esta es una de las 16 combinaciones posibles con lo cual se logra la salida más cercana a 220 voltios.

El gate de los triacs se activan mediante fotoacopladores ya que como explicamos anteriormente la etapa de potencia debe estar siempre separada de la etapa de control, cada fotoacoplador es activado a través del colector de un transistor donde además se coloca un led para indicar cuál de los triacs de mayor peso está funcionando en un momento dado, la base de estos transistores son activadas por las salidas de la U2A que viene a ser 1/2 de un circuito integrado TTL 74LS139, que es un doble decodificador de 2 a 4 líneas, cuyo pin de habilitación identificado por la letra G es común con la U2B.

La U2A tiene sus entradas A y B comandados por P1.2 y P1.3 del puerto 1 del microcontrolador, donde para una combinación binaria de estas entradas se activa una sola salida de este decodificador, que activa a la base de un transistor, cuyo colector controla a un fotoacoplador y este a su vez al gate de uno de los triacs del banco de mayor peso con lo cual se está seleccionando un determinado tab de mayor peso del primario del autotransformador; en realidad la U2A y la U2B deben funcionar a la vez ya que la conmutación hacia los tabs siempre es una del banco de menor peso con otra del banco de mayor peso, por lo tanto el microcontrolador enviará la combinación binaria adecuada a través de P1.0, P1.1 para U2B y a través de P1.2, P1.3 para U2A, con esta combinación se logra la conmutación más apropiada con la finalidad de obtener a la salida siempre un voltaje lo más

próximo posible a 220 voltios.

La combinación de la resistencia con el condensador en serie y en paralelo con el triac, además la resistencia pequeña de 0.2 Ohmios a 5 W. en serie, cumplen las mismas funciones explicadas en el diseño del circuito de banco de triacs de menor peso.

#### **4.9 Diseño del circuito de sensado del voltaje de entrada**

El voltaje de entrada se baja a 12 voltios mediante un transformador, se rectifica y luego mediante la resistencia R24 en serie con el diodo D3 y a su vez en serie con el condensador C5 en paralelo con un potenciómetro P2, la señal es tomada desde el potenciómetro P2 y va a un sistema de comparación de 2 niveles, este sistema de comparación es diseñado usando el LM324, el nivel máximo de comparación es 2.55 voltios correspondiente a 255 voltios y el nivel mínimo es 1.75 voltios, correspondiente a 175 voltios, estos valores se han tomado ya que para el diseño del sistema propuesto se parte considerando la variación del voltaje de entrada entre 175 voltios y 255 voltios, la salida de uno de los comparadores llamado Vmax. va al P2.6 del puerto 2 del microcontrolador y la salida del otro comparador llamado Vmin. va al P2.7, si el voltaje de entrada está dentro del rango de estos 2 niveles en las salidas Vmax y Vmin se tiene nivel bajo, cuando en Vmax. se detecta un nivel alto significa que el voltaje de entrada está muy alto, y cuando en Vmin. se detecta un nivel alto significa que el voltaje de entrada está muy bajo, de esta manera se le

informa al microcontrolador a cerca del voltaje de entrada para que desconecte el sistema, como un mecanismo de prevención.

Además con criterio de visualización en la salida correspondiente a  $V_{max}$ . se coloca un led de color rojo para indicar el voltaje de entrada está muy alto, y en la salida correspondiente a  $V_{min}$ . se coloca un led de color amarillo para indicar que el voltaje de entrada está muy bajo; los transistores Q12 y Q13 sirven para proporcionar la corriente necesaria para los leds indicadores, el diodo D6 es para evitar que los leds muestren la sensación de encendido cuando el voltaje de entrada está dentro del rango de los niveles establecidos.

#### **4.10 Diseño del circuito de control**

El circuito de control fundamentalmente está formado por el microcontrolador 8748, el cual posee una Eprom de 1 Kbytes, una RAM de 64 bytes, 2 puertos de entrada y salida y un Timer internamente como elementos importantes.

A través del puesto 1 se realiza el control de los siguientes elementos: P1.0 y P1.1 constituye las entradas A y B del decodificador U2B que controla al banco de triacs de menor peso, P1.2 y P1.3 son las entradas A y B del decodificador U2A que controla al banco de triacs de mayor peso, el P1.7 va a la entrada G de los decodificadores y cumple la función de habilitación.

La entrada de modo es el P1.6, normalmente ésta entrada esta abierta y sirve para realizar la prueba o testeo del estabilizador para lo cual se coloca un puente y luego

automáticamente se realizan todas las conmutaciones de las posibles combinaciones de los triacs, con lo cual se puede verificar el voltaje de salida, además se tiene leds como se mencionó anteriormente que indican cuales son los triacs que están funcionando, esta entrada de modo también permite detección de fallas, e identificar cual es el triac defectuoso, osea permite el mantenimiento del equipo.

El switch de Auto-Man, que va al pin P1.5, permite el funcionamiento automático o manual de tal manera cuando ocurre que el voltaje de entrada es muy alto o muy bajo respecto al rango de entrada establecido, el cual el estabilizador detecta y de inmediato se desconecta como una medida de seguridad y cuando el voltaje de entrada nuevamente entra dentro del rango establecido ocurre un retardo de 5 seg. y recién entrega energía esto ocurre cuando el switch Auto-Man esta en Auto, y en el caso de que el switch estuviera en Manual espera que se pulse el pulsador que va al P1.4 del microcontrolador, para recién entregar energía.

Mediante el puerto 2 se realiza el control de los siguientes elementos: a través del P2.4 y P2.5 recibe la información del estado de las salidas VH y VL del circuito de sensado del voltaje de salida, de tal manera si en VH se lee un nivel alto significa que el voltaje de salida esta muy alto, y si en VL se lee un nivel alto significa que el voltaje de salida esta muy bajo, estos valores se leen en el flanco de subida del pulso de paso por cero ,

si uno de estos valores ya sea VH o VL esta en alto entonces el microcontrolador ejecuta la rutina de salida con la cual se realiza una conmutación entre las posibles combinaciones de tal manera se da una solución a la anomalía existente en el voltaje de salida.

A través de P2.6 correspondiente a Vmax. y P2.7 correspondiente a Vmin. el microcontrolador recibe la información acerca del voltaje de entrada desde el circuito de sensado de voltaje de entrada, de tal manera que si en Vmax. o en Vmin. se encuentra un nivel alto significa que el voltaje de entrada salió del rango para el cual fue diseñado el estabilizador, es decir que el voltaje de entrada esta muy alto o muy bajo, entonces con esta información el microcontrolador desconecta al estabilizador como un mecanismo de protección.

Por el P2.3 se activa a la base del transistor Q10 que va alimentar a través del colector a un led de color verde el cual es un indicador de funcionamiento normal del estabilizador, a través de P2.2 se alimenta la base del mismo transistor, en este caso el led funciona como un led auxiliar, cuya función de este led al flatchear es indicar anomalías en el estabilizador.

La entrada T0 es a través del cual se tiene el momento preciso del flanco de subida del pulso de paso por cero generado por el transistor Q1, esta información es útil para que el microcontrolador pueda tomar decisiones durante la corrida de las rutinas, de acuerdo a los valores leídos en las entradas en VH y VL.

La entrada T1 es a través de la cual el microcontrolador toma la información correspondiente a la corriente por los triacs, esto se hace mediante el PIO del circuito de sensado de la corriente por los triacs.

Además a través de los pines 2 y 3 del microcontrolador se proporciona la señal de reloj del sistema, esta señal de reloj es generada por un cristal de 5.58 Mhz. y los condensadores C3 y C4.

#### **4.11 Diseño de la fuente de alimentación.**

La fuente de alimentación del sistema esta formada por un transformador de 220/12 voltios, con 3 devanados de 12 voltios en el secundario, 2 de estos devanados se utiliza para realizar la fuente de los 2 bancos de triacs del sistema, para estas fuentes en cada devanado se colocan un puente de diodos para la rectificación y un condensador para el filtro, una de estas fuentes cuya salida es de 12 voltios alimenta a los 4 triacs del banco de menor peso, y la otra fuente también de 12 voltios de salida alimenta a los 4 triacs del banco de mayor peso.

El tercer devanado también tiene un puente de diodos para la rectificación y un condensador para el filtro, pero además se coloca el LM7805, que es un regulador para una salida de 5 voltios, esta fuente es la que se utiliza para la polarización de todos los componentes de las tarjetas, incluyendo al propio microcontrolador.

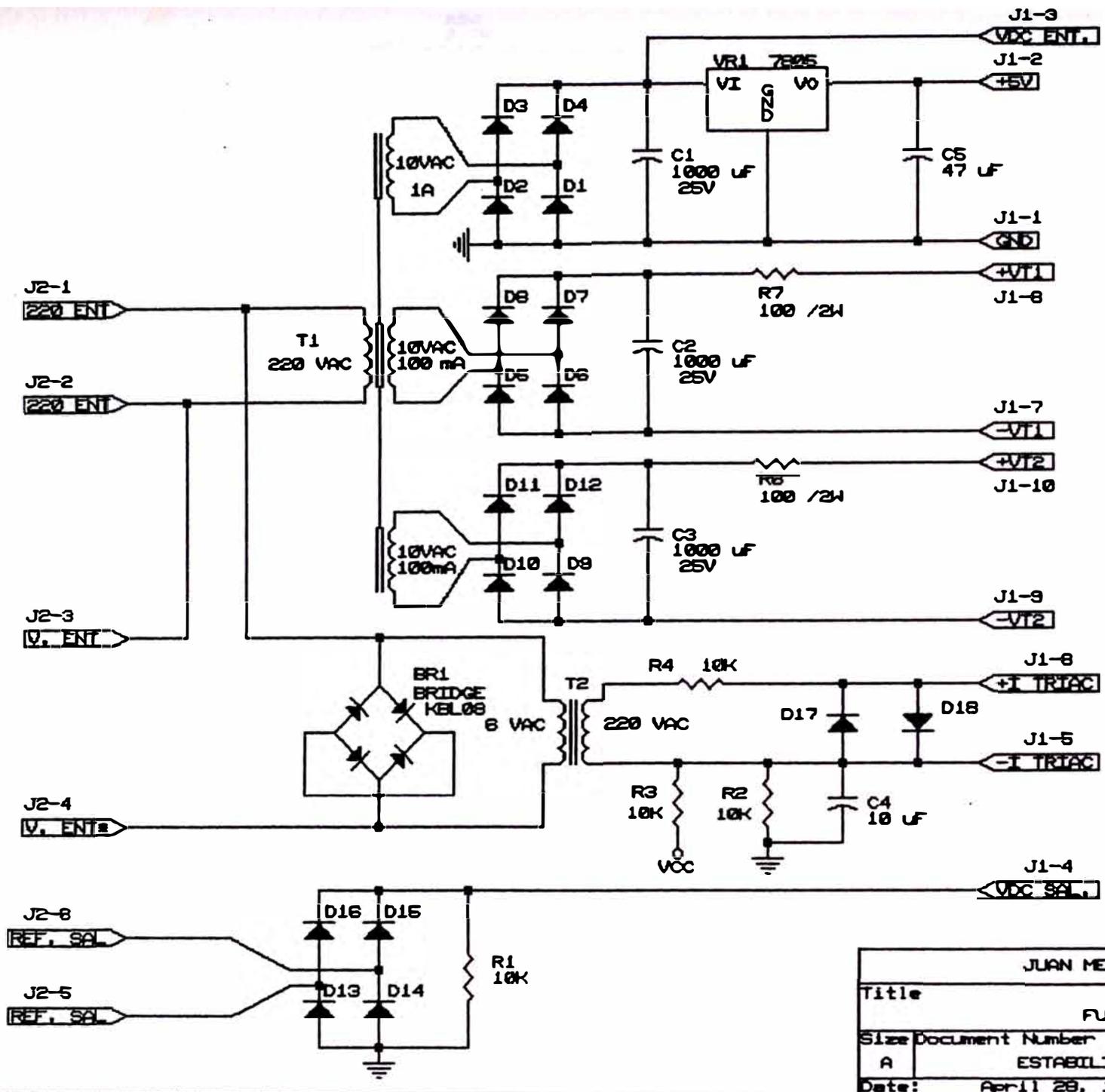


Fig. 14

JUAN MEZA ARRIETA		
Title		
FUENTES		
Size	Document Number	REV
A	ESTABILIZADOR T4X4C	01
Date:	April 28, 1998	Sheet 1 of 4

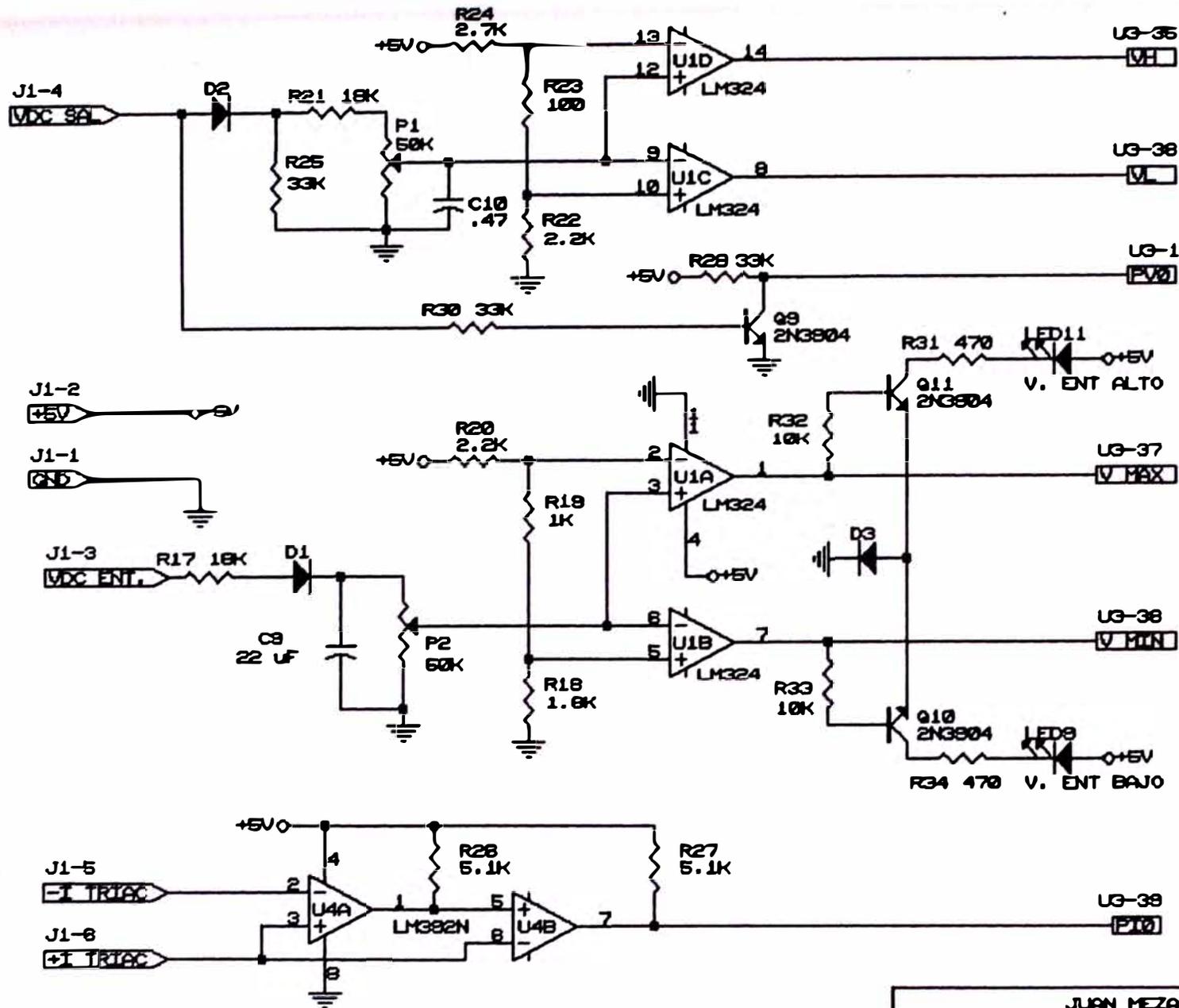


Fig. 15

JUAN MEZA ARRIETA		
Title		
ENTRADAS		
Size	Document Number	REV
A	ESTABILIZADOR T4X4C	01
Date:	April 28, 1988 Sheet 2 of 4	

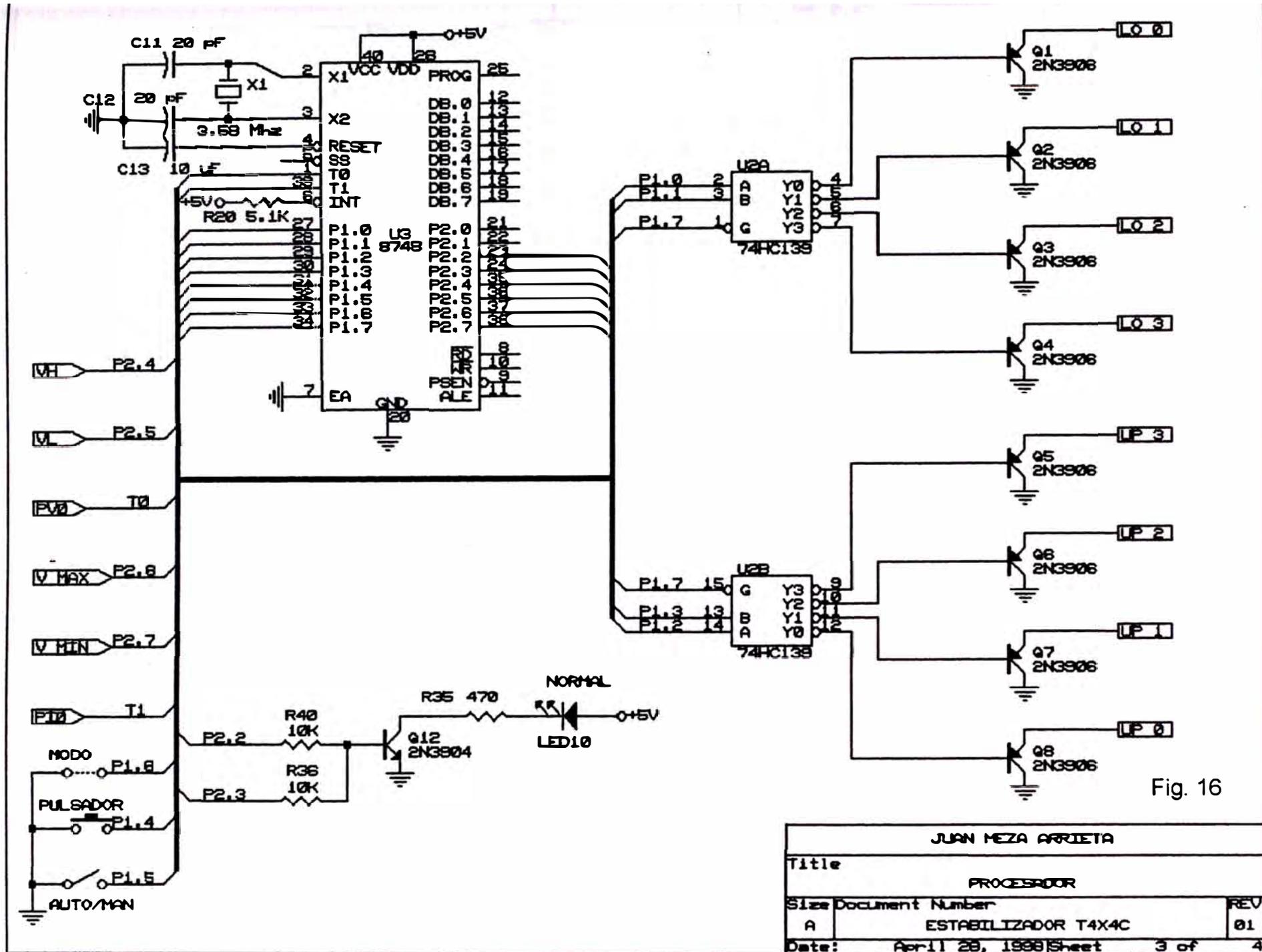


Fig. 16

JUAN MEZA ARRIETA		
Title		
PROCESADOR		
Size	Document Number	REV
A	ESTABILIZADOR T4X4C	01
Date:	April 28, 1988	Sheet 3 of 4

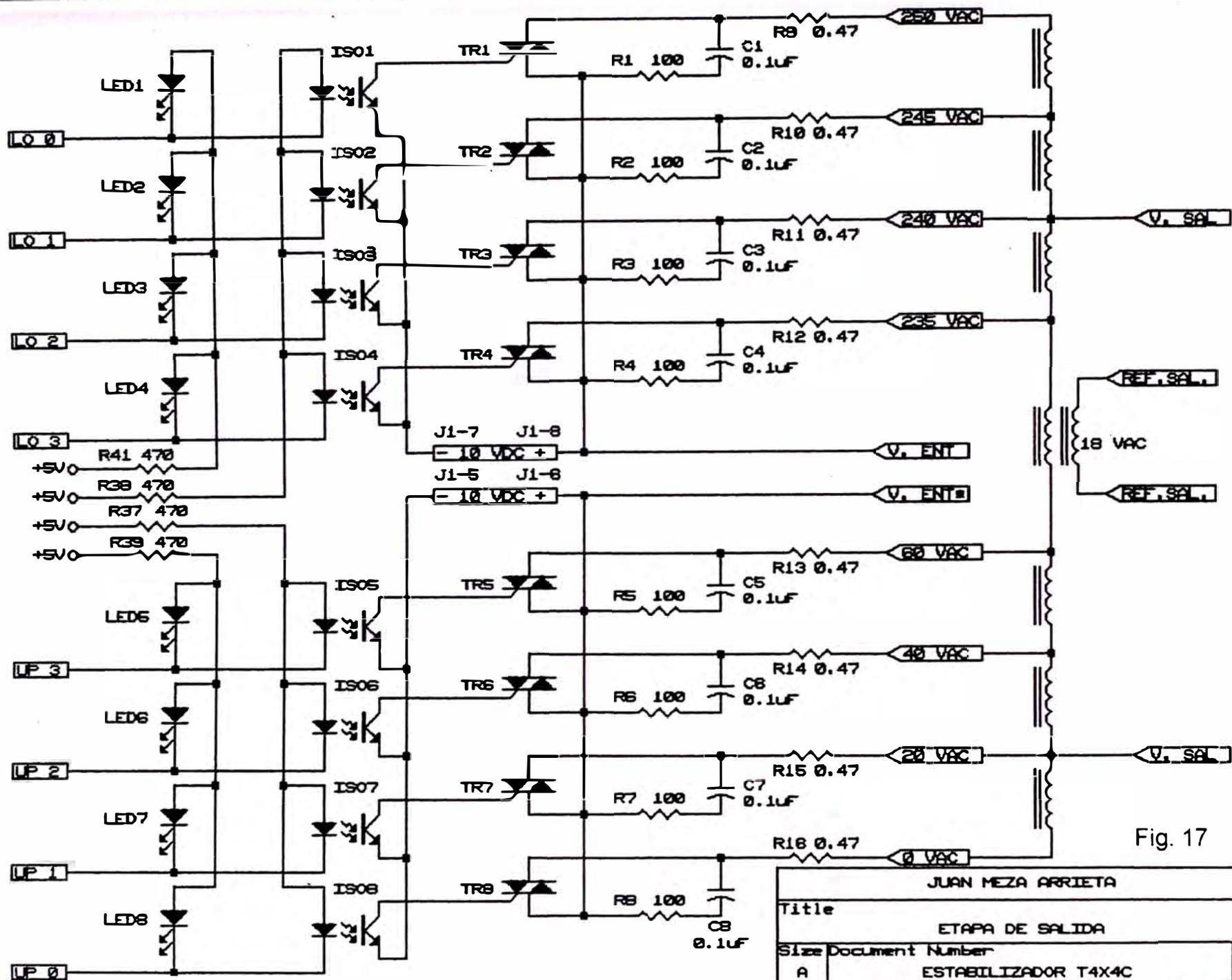


Fig. 17

JUAN MEZA ARRIETA		
Title		
ETAPA DE SALIDA		
Size	Document Number	REV
A	ESTABILIZADOR T4X4C	01
Date:	April 26, 1998	Sheet 4 of 4

## 4.12 Programación del software

### 4.12.1 Programa en Basic para generar las combinaciones posibles que permiten la salida más cercana a 220 voltios

```
10 REM "trafo"
20 CLS
30 LPRINT "  TABLA DE ENTRADA / SALIDA "
40 LPRINT "  -----"
50 LPRINT
60 LPRINT
70 LPRINT " Este es un cálculo por computador de la
          salida"
75 LPRINT " mas cercana a 220 VOLT. de un
          autotransformador "
80 LPRINT " con 8 tomas en el primario dispuestas en 4"
85 LPRINT " superiores x 4 inferiores lo que da 16
          combinaciones"
90 LPRINT " (una superior y otra inferior)"
95 LPRINT " Las tomas del autotransformador se muestran
          en la fig."
100 LPRINT " y son las que se medirían si el voltaje de"
110 LPRINT "entrada es 220 voltios, la entrada se hará
          variar desde"
120 LPRINT "175 voltios hasta 255 voltios"
140 LPRINT "El programa calcula la combinación que da la"
150 LPRINT "salida mas cerca a 220 voltios y se imprimirán
          el voltaje de"
160 LPRINT "entrada,salida y la combinación"
```

```

165 LPRINT "correspondiente. "
170 LPRINT "Estos valores teóricos serán en la práctica
      una aproximación"
175 LPRINT "del voltaje de salida del "
180 LPRINT " estabilizador T-4x4uC"
190 DIM k(16)
200 VC=220
210 DIM V(7)
220 LPRINT
230 LPRINT
240 LPRINT " TOMAS DEL TRANSFORMADOR "
250 LPRINT "      -----"
260 FOR N=0 TO 7
270 READ V(N)
280 LPRINT "      V(";N;")=";V(N)
290 NEXT N
300 FOR Q=1 TO 4 : LPRINT : NEXT Q
310 FOR H=0 TO 3
320 FOR J= 0 TO 3
330 P=4*H+J
340 K(P)=VC/(V(7-J)-V(H))
350 NEXT J
360 NEXT H
370 LPRINT "ENT (VOLT) SAL (VOLT) TOMA COMBINADA "
380 LPRINT " -----"
390 DIM VS(16)
400 GAMMA= 0
410 OMEGA= 0

```

```
420 LPRINT CHR$(27);"e";CHR$(0);CHR$(18)
430 LPRINT
440 FOR VE=175 TO 255
450 MIN=100
460 FOR J=0 TO 15
470 VS(J)=K(J)*VE
480 DELTA = 220 - VS(J)
490 IF MIN > ABS(DELTA) THEN GOSUB 660
500 NEXT J
510 PRINT VE, VS
520 LPRINT " ";VE;CHR$(9);VS;CHR$(9);H
530 VY=220 - VS
540 IF VY > GAMMA THEN GAMMA = VY
550 IF VY < OMEGA THEN OMEGA= VY
560 IF VE=190 THEN GOSUB 700
570 IF VE= 243 GOSUB 700
580 NEXT VE
590 PRINT " ";GAMMA, OMEGA
600 LPRINT
610 LPRINT "Para el rango de entrada 175 - 255 voltios "
615 LPRINT "se tiene los siguientes voltajes de salida"
620 LPRINT "mínimo y máximo : "
630 LPRINT
640 LPRINT "VS min=";220 - GAMMA;"V VS max=";220-
        OMEGA;"VOLT"
650 END
660 MIN = DELTA
670 VS = VS(J)
```

```
680 H=J
690 RETURN
700 INPUT A$
710 LPRINT
720 LPRINT
730 LPRINT
740 LPRINT "ENT (VOLT) SAL VOLT) TOMA COMB"
750 LPRINT "-----"
760 LPRINT
770 RETURN
780 DATA 0,20,40,60,235,240,245,250
```

#### TABLA DE ENTRADA/SALIDA

Este es un cálculo por computador de la salida mas cercana a 220 volt. de un autotransformador con 8 tomas en el primario dispuestas en 4 superiores x 4 inferiores lo que da 16 combinaciones (una superior y una inferior).

Las tomas del transformador se muestran en la figura y son las que se medirían si el voltaje de entrada es 220 voltios.

La entrada se hará variar desde 175 voltios hasta 255 voltios el programa calcula la combinación que da la salida mas cerca a 220 voltios y se imprimirán el voltaje de entrada, salida y la combinación correspondiente.

Estos valores teóricos serán en la práctica una aproximación del voltaje de salida del estabilizador T-4x4uC.

## TOMAS DEL TRANSFORMADOR

$$V(0) = 0$$

$$V(1) = 20$$

$$V(2) = 40$$

$$V(3) = 60$$

$$V(4) = 235$$

$$V(5) = 240$$

$$V(6) = 245$$

$$V(7) = 250$$

ENT (VOLT)	SAL (VOLT)	TOMA COMBINADA
------------	------------	----------------

175	220	15
176	221.2572	15
177	222.5143	15
178	217.5556	14
179	218.7778	14
180	220	14
181	221.2222	14
182	222.4445	14
183	217.6216	13
184	218.8108	13
185	220	13
186	221.1892	13
187	222.3784	13
188	217.6842	12
189	218.8421	12
190	220	12

ENT (VOLT)	SAL (VOLT)	TOMA COMBINADA
-----	-----	-----
191	221.1579	12
192	222.3158	12
193	217.7436	11
194	218.8718	11
195	220	11
196	221.1282	11
197	222.2564	11
198	217.8	10
199	218.9	10
200	220	10
201	221.1	10
202	222.2	10
203	217.8537	9
204	218.9268	9
205	220	9
206	221.0732	9
207	222.1464	9
208	217.9048	8
209	218.9524	8
210	220	8
211	221.0477	8
212	222.0953	8
213	217.9535	7
214	218.9768	7
215	220	7
216	221.0233	7

ENT (VOLT)	SAL (VOLT)	TOMA COMBINADA
-----	-----	-----
217	222.0465	7
218	218	6
219	219	6
220	220	6
221	221	6
222	222	6
223	218.0445	5
224	219.0222	5
225	220	5
226	220.9778	5
227	221.9556	5
228	218.087	4
229	219.0435	4
230	220	4
231	220.9565	4
232	221.9131	4
233	218.1277	3
234	219.0638	3
235	220	3
236	220.9362	3
237	221.8724	3
238	218.1667	2
239	219.0834	2
240	220	2
241	220.9167	2
242	221.8334	2

ENT (VOLT)	SAL (VOLT)	TOMA COMBINADA
-----	-----	-----
243	218.2041	1
244	219.102	1
245	220	1
246	220.898	1
247	221.7959	1
248	218.24	0
249	219.12	0
250	220	0
251	220.88	0
252	221.76	0
253	222.64	0
254	223.52	0
255	224.4	0

Para el rango de entrada 175 - 255 voltios se tiene los siguientes voltajes de salida mínimo y máximo:

VS min = 217.5556 volt. VS max = 224.4 volt.

#### **4.12.2 Programa en "lenguaje C" para generar las combinaciones posibles que permiten la salida más cercana a 220 voltios**

```
/* trafo */
float min, delta, vvs=0,vy;
int v[8]={0,20,40,60,235,240,245,250};
float vs[16],k[16], vc,gamma,omega;
main()
{
```

```

clrscr();
printf("          TABLA ENTRADA/SALIDA\n");
printf("          =====\n");
printf("\n\nEste es un cálculo por computador de la salida
mas cercana a\n");
printf("220 volt de un autotransformador con 8 tomas en el
primario\n");
printf("dispuestas en 4 superiores x 4 inferiores lo que
da 16 combi-\n")
printf("naciones (una superior y una inferior)\n");
printf("Las tomas del autotransformador se muestran en la
figura y son\n")
printf("las que se medirían si el voltaje de entrada es
220 voltios\n");
printf("El programa calcula la combinación que da la
salida más cerca\n");
printf("a 220 voltios y se imprimiran el voltaje de
entrada, salida y\n");
Printf("la combinación correspondiente\n");
printf("Estos valores teóricos serán en la práctica una
aproximación \n");
printf("del voltaje de salida del estabilizador
T-4X4uC\n");
getch();
clrscr();
vc=220 ;
printf("\n\n\n TOMAS DEL TRANSFORMADOR \n");
printf("          ===== \n");

```

```

for (n=0;n<=7;n++)
  { printf (" V(%d)= %d\n",n,v[n]);
  }
for(q=1;q<=4;q++)
{printf("\n");
}
{printf("\n");
}
for (h=0;h<=3;h++)
  {for(j=0;j<=3;j++)
    {p=4*h+j;
    k[p]=vc/(v[7-j]-v[h]);
    }
  }
printf("\n\tENT(volt)  SAL(volt)  TOMA  COMBINADA\n\n");
gamma=0;
omega=0;
for (ve=175;ve<=255;ve++)
{min=100;delay(1000);
  for (j=0;j<=15;j++)
    {vs[j]=k[j]*ve;
    delta=220 - vs[j];
    if (min>abs(delta))
      {vs[j]=k[j]*ve;
      delta=220 - vs[j];
      if (min>abs(delta))
        {gosub660();}
      }
    }
}

```

```

printf("\t %d \t\t\ %6.2f \t\t %d \n",ve,vvs,h);
vy=220 - vvs;
if (vy>gamma)      omega=vy;
if (vy<omega )    omega=vy;
if (ve==190)      gosub700();
if (ve==243)      gosub700();
}
printf("      %d      %d\n",gamma,omega);
printf("para el rango de entrada 175-255 voltios se
      tiene\n");
printf("los siguientes voltajes de salida mínimo y
      máximos \n");
printf("\n\nVS min =%d volt      vs max=%d      volt\n",
      220-gamma,
getch());
}
gosub660()
{ min=delta;
  vvs=vs[j];
  h=j;
}
gosub700()
{
  printf("\n\tENT (volt) SAL(volt) TOMA      COMBINADA\n\n");
}

```

## TOMAS DEL TRANSFORMADOR

-----

V(0) = 0  
 V(1) = 20  
 V(2) = 40  
 V(3) = 60  
 V(4) = 235  
 V(5) = 240  
 V(6) = 245  
 V(7) = 250

ENT (VOLT)	SAL (VOLT)	TOMA COMBINADA
-----	-----	-----
175	220	15
176	221.2572	15
177	222.5143	15
178	217.5556	14
179	218.7778	14
180	220	14
181	221.2222	14
182	222.4445	14
183	217.6216	13
184	218.8108	13
185	220	13
186	221.1892	13
187	222.3784	13
188	217.6842	12
189	218.8421	12

ENT (VOLT)	SAL (VOLT)	TOMA COMBINADA
-----	-----	-----
190	220	12
191	221.1579	12
192	222.3158	12
193	217.7436	11
194	218.8718	11
195	220	11
196	221.1282	11
197	222.2564	11
198	217.8	10
199	218.9	10
200	220	10
201	221.1	10
202	222.2	10
203	217.8537	9
204	218.9268	9
205	220	9
206	221.0732	9
207	222.1464	9
208	217.9048	8
209	218.9524	8
210	220	8
211	221.0477	8
212	222.0953	8
213	217.9535	7
214	218.9768	7
215	220	7

ENT (VOLT)	SAL (VOLT)	TOMA COMBINADA
-----	-----	-----
216	221.0233	7
217	222.0465	7
218	218	6
219	219	6
220	220	6
221	221	6
222	222	6
223	218.0445	5
224	219.0222	5
225	220	5
226	220.9778	5
227	221.9556	5
228	218.087	4
229	219.0435	4
230	220	4
231	220.9565	4
232	221.9131	4
233	218.1277	3
234	219.0638	3
235	220	3
236	220.9362	3
237	221.8724	3
238	218.1667	2
239	219.0834	2
240	220	2
241	220.9167	2

ENT (VOLT)	SAL (VOLT)	TOMA COMBINADA
242	221.8334	2
243	218.2041	1
244	219.102	1
245	220	1
246	220.898	1
247	221.7959	1
248	218.24	0
249	219.12	0
250	220	0
251	220.88	0
252	221.76	0
253	222.64	0
254	223.52	0
255	224.4	0

Para el rango de entrada 175 - 255 voltios se tiene los siguientes voltajes de salida mínimo y máximo:

VS min = 217.5556 volt. VS max = 224.4 volt.

#### **4.12.3 Diagramas de flujo para el programa de control**

Previamente antes de entrar a realizar el programa ensamblador en detalle se ha hecho 6 diagramas de flujo donde se presentan en forma clara la lógica del algoritmo de cada una de las subrutinas y partes, de las cuales se compone el programa de control del sistema escrito en el ensamblador del microcontrolador utilizado en el diseño del sistema propuesto. Estos diagramas de flujo se encuentran en las figuras del 18 al 23.

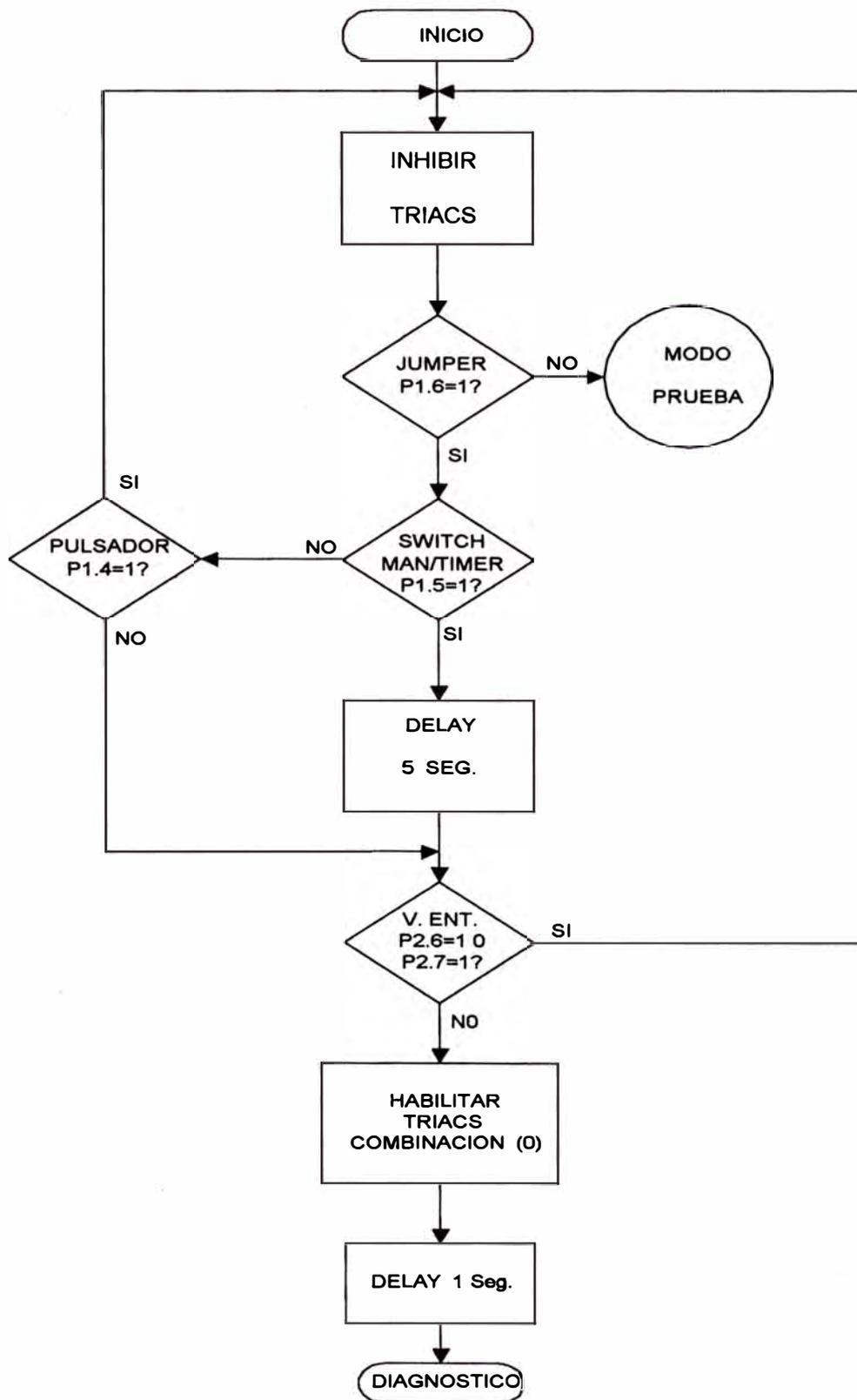


Fig. 18

Diagrama de flujo general

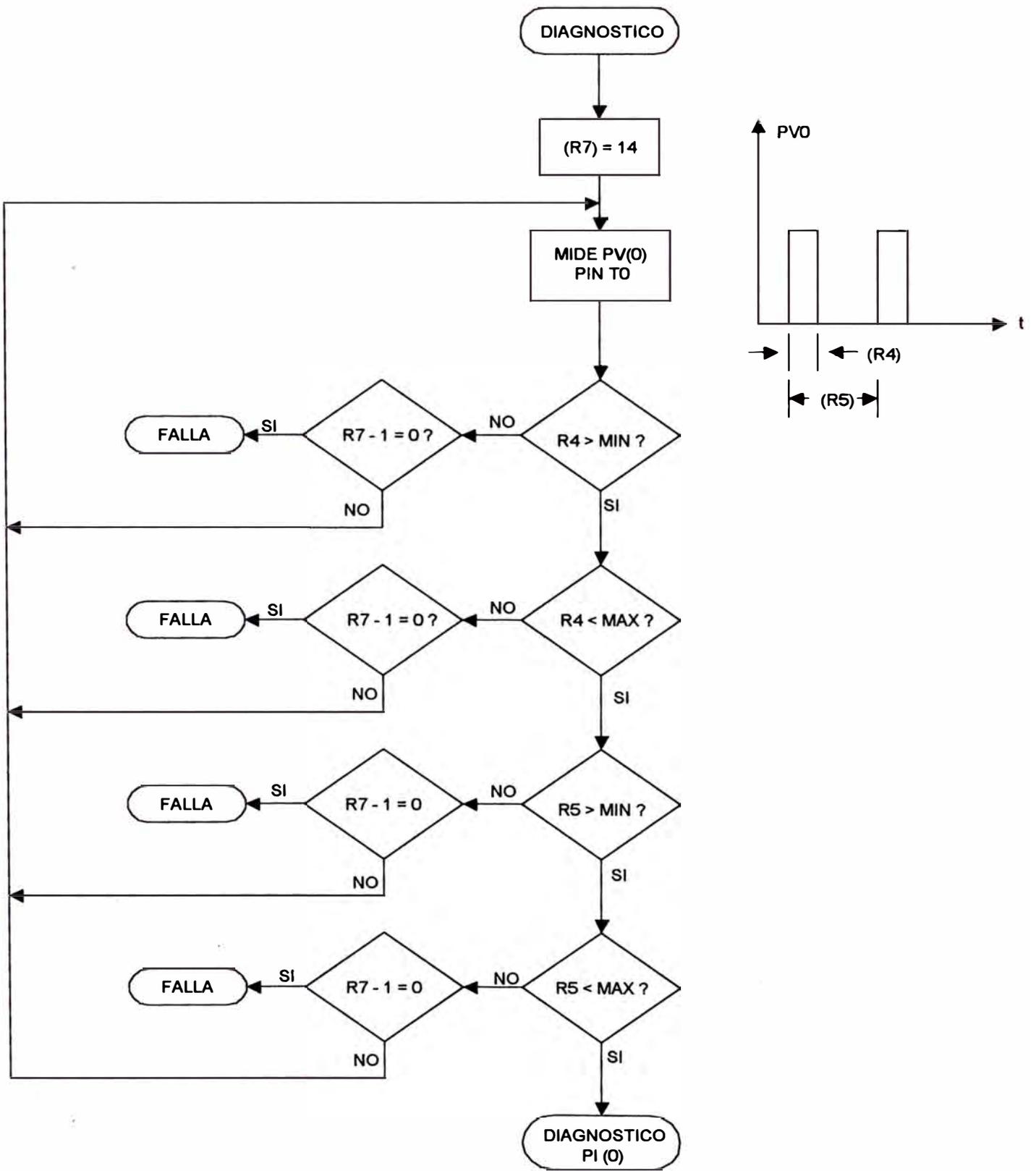


Fig. 19

Diagrama de flujo general

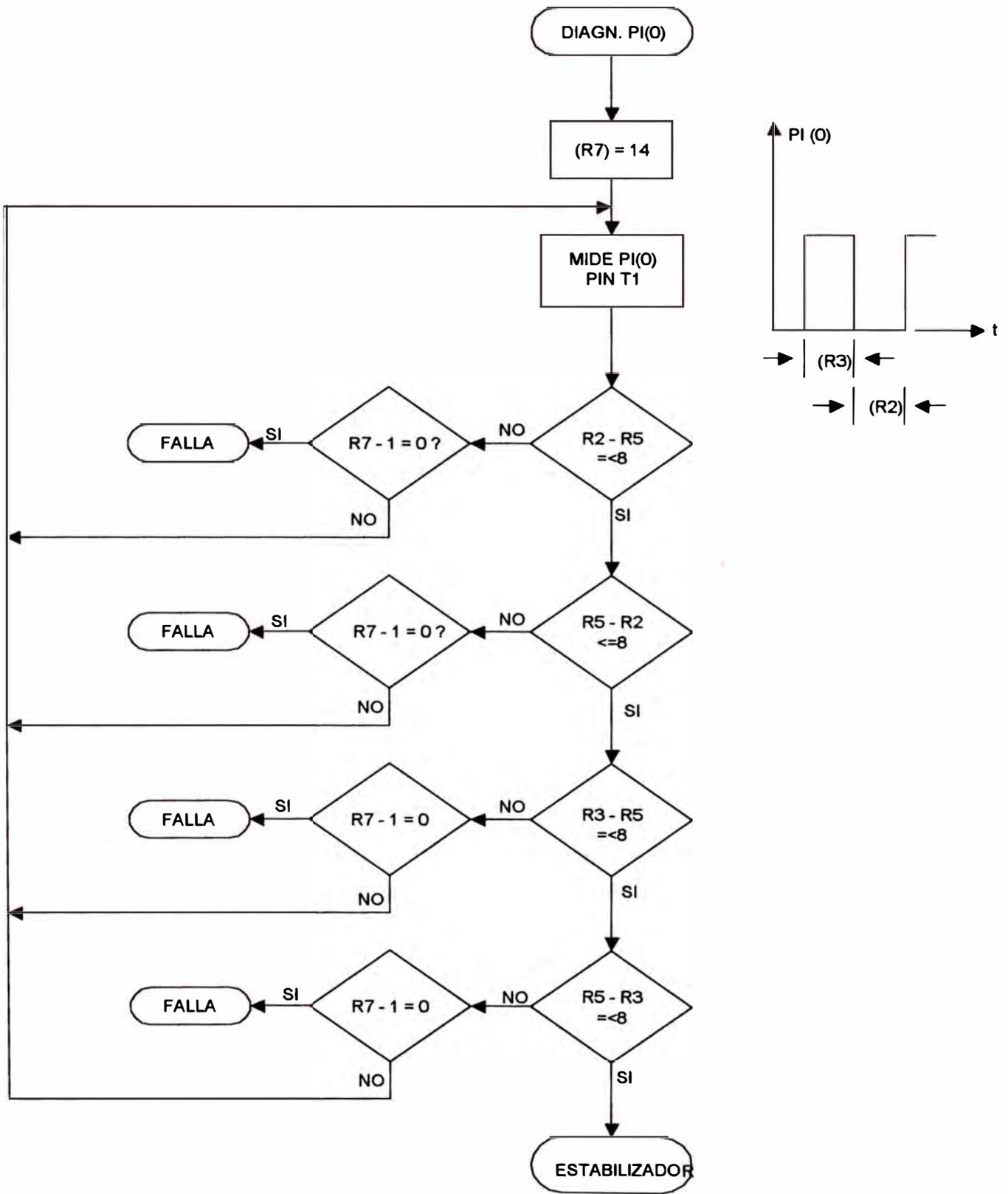


Fig. 20

Diagrama de flujo general

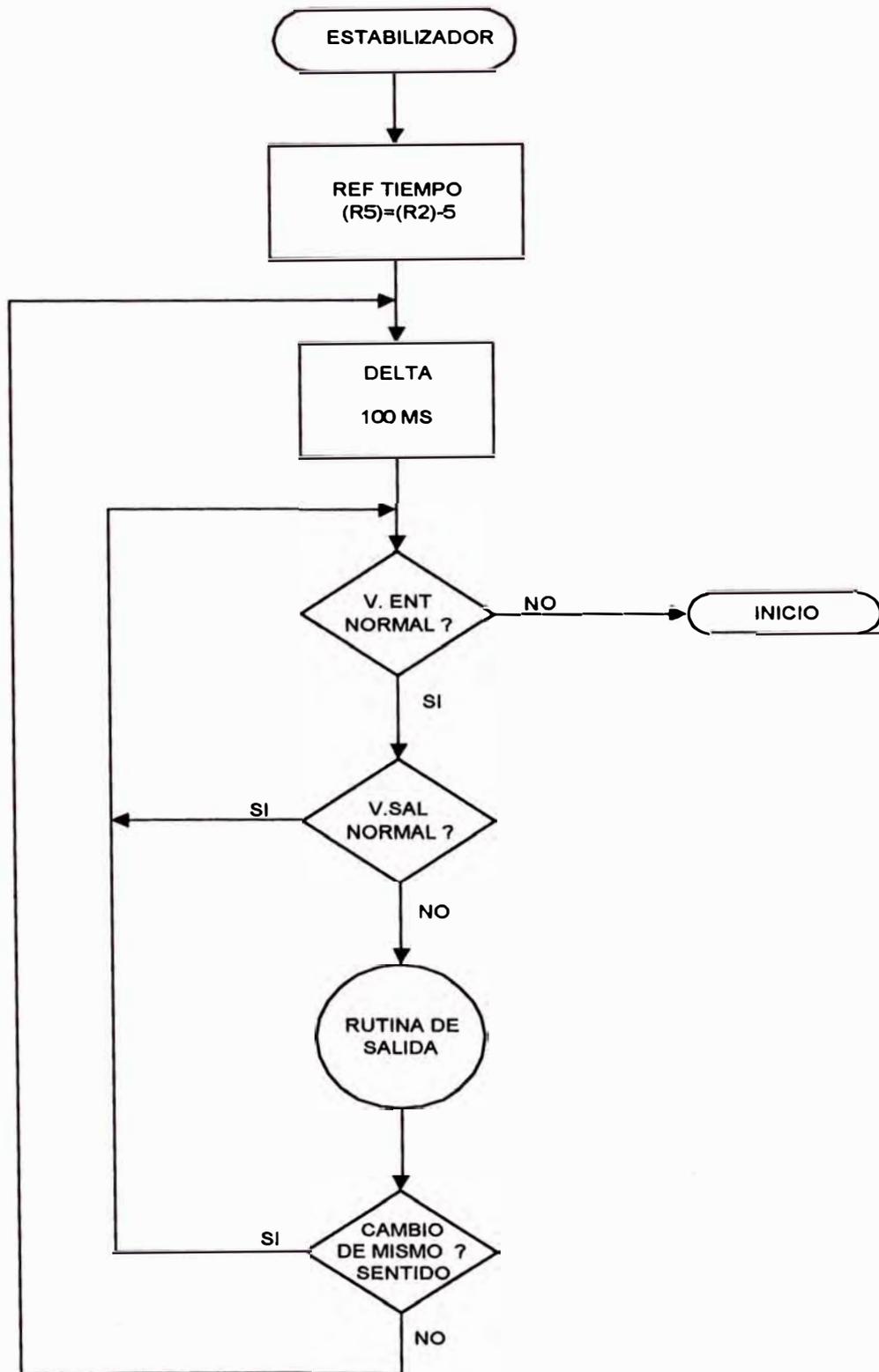


Fig. 21

Diagrama de flujo general

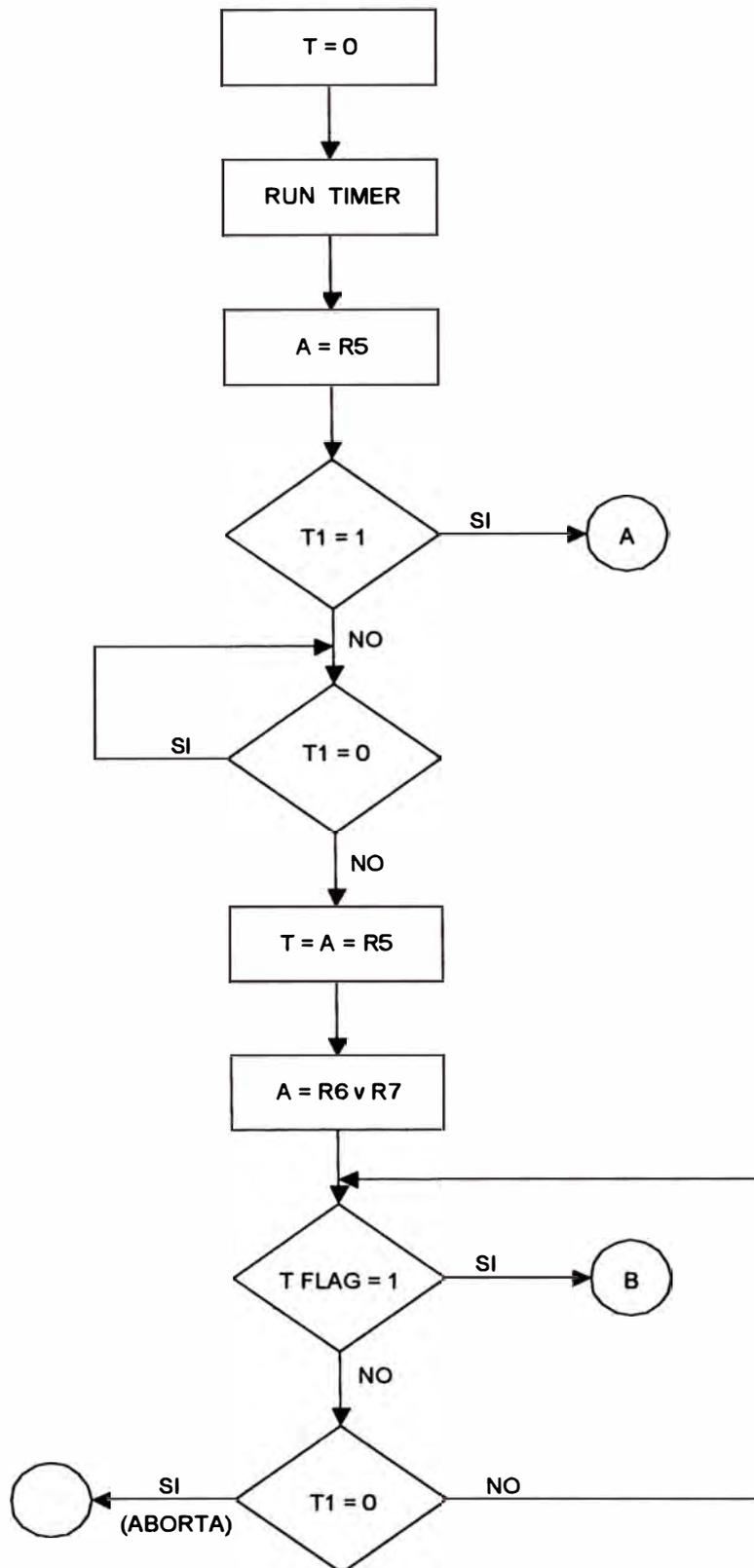


Fig. 22

Diagrama de flujo general

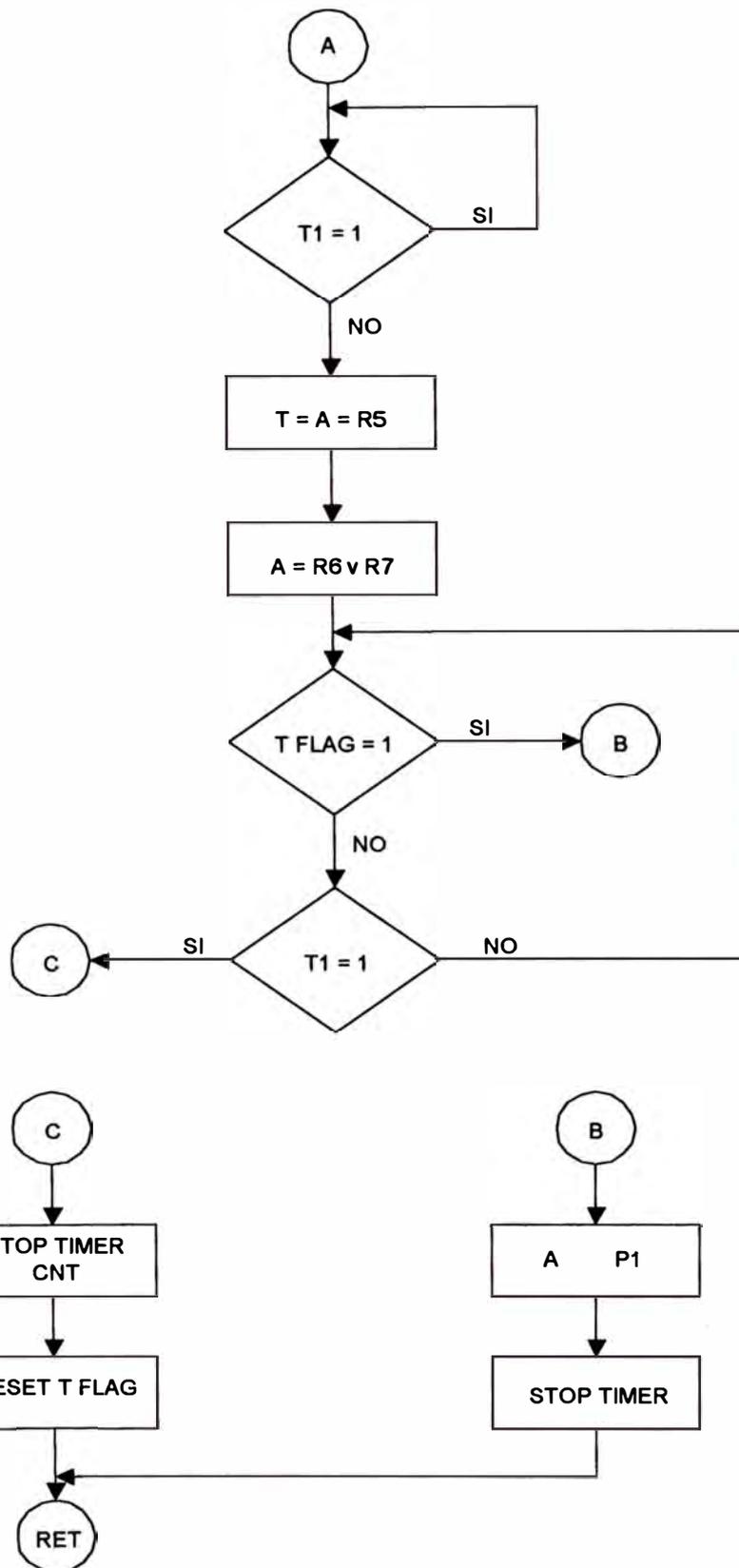


Fig. 23

Diagrama de flujo general

#### 4.12.4 Programa en ensamblador para el sistema de control

Programa del sistema de estabilizador t-4x4uc

```

=====
LOC      OBJ      MNEMONICO      COMENTARIOS
=====
                                                \\SERVICIOS DE
                                                INTERRUPTS
000      15              DIS I
001      04 0B      JMP $00B
003      15              DIS I          INT\
004      35              DIS TCNT I
005      24 80      JMP $180          FALLA# 1
007      15              DIS I          T INT\
008      35              DIS TCNT I
009      24 90      JMP $190          FALLA # 2
-----
                                                \\PROGRAMA DE
                                                DIAGNOSTICO
00B      05              EN I
00C      23 F7      MOV  A, #F7
00E      3A              OUTL P2,A
00F      39              OUTL P1,A
010      09              IN  A, P1
011      D2 15      JB6 $015          MODO ESTAB.
013      24 30      JMP $130          MODO PRUEBA
015      B2 1B      JB5 $01B          AUTOTIMER
017      92 0C      JB4 $00C          PULSADOR

```

```

019    04 1D    JMP  $01D
01B    14 E8    CALL $0E8    DELAY 5 SEG.
01D    0A      IN  A,P2
01E    D2 0C    JB6  $00C    V.ENT MUY ALTO
020    F2 0C    JB7  $00C    V.ENT MUY BAJO
022    23 FB    MOV  A,#FB
024    3A      OUTL P2,A
025    23 70    MOV  A.#70
027    39      OUTL P1,A
028    14 D8    CALL $0D8    DELAY 1 SEG.
02A    BF 14    MOV  R7,#14
02C    34 70    CALL $170    DELAY 100 ms
02E    34 A0    CALL $1A0    [ R4 Y R5
030    97      CLR  C
031    23 FD    MOV  A,#FD
033    6C      ADD  A,R4
034    F6 3A    JC   $03A    PV(0) MIN OK
036    EF 2C    DJNZ R7,$02C
038    04 07    JMP  $007    FALLA # 2
03A    97      CLR
03B    23 F8    MOV  A,#F8
03D    6C      ADD  A,R4
03E    E6 44    JNC  $044    PV(0) MAX  OK
040    EF 2C    DJNZ R7,$02C
042    04 07    JMP  $007    FALLA # 2
044    23 C7    MOV  A,#C7
046    6D      ADD  A,R5
047    F6 4D    JC   $04D    PV(0) MIN OK

```

```

049   EF 2C      DJNZ R7,$02C
04B   04 07      JMP $007          FALLA # 2
04D   97         CLR C
04E   23 BD      MOV A,#BD
050   6D         ADD A,R5
051   E6 57      JNC $057          PV(0) MAX OK
053   EF 2C      DJNZ R7,$02C
055   04 07      JMP $007          FALLA # 2
057   BF 14      MOV R7,#14
059   34 70      CALL $170        DELAY 100 ms
05B   34 C0      CALL $1C0        [ R2  Y R3 ]
05D   97         CLR C
05E   FD         MOV A,R5
05F   03 08      ADD A,#08
061   37         CPL A
062   6A         ADD A,R2
063   E6 69      JNC $069
065   EF 59      DJNZ R7,$059
067   04 07      JMP $007          FALLA # 2
069   FA         MOV A,R2
06A   03 08      ADD A,#08
06C   37         CPL A
06D   6D         ADD A,R5
06E   E6 74      JNC $074          PI(0) OK
070   EF 59      DJNZ R7,$059
072   04 07      JMP $007          FALLA # 2
074   FD         MOV A,R5
075   03 08      ADD A,#08

```

```

077 37      CPL A
078 6B      ADD A,R3
079 E6 7F   JNC $07F
07B EF 59   DJNZ R7,$059
07D 04 07   JMP $007      FALLA # 2
07F FB      MOV A,R3
080 03 08   ADD A,#08
082 37      CPL A
083 6D      ADD A,R5
084 E6 8A   JNC $08A      DIAGNOST. OK
086 EF 59   DJNZ R7,$059
088 04 07   JMP $007      FALLA # 2

```

-----

\\PROGRAMA DE  
ESTABILIZADOR

```

08A 34 E0   CALL 1E0
08C 00      NOP
08D 00      NOP
08E 00      NOP
08F BE 00   MOV R6,#00   VAR. DE SALIDA
091 BF 70   MOV R7,#70   MASK
093 85      CLR F0
094 95      CPL F0
095 A5      CLR F1
096 B9 00   MOV R1,#00
098 BA 01   MOV R2,#01
09A 34 70   CALL $170    DELAY 100 ms
09C 0A      IN A,P2

```

09D	D2 0C	JB6 \$00C	V. ENT ALTO
09F	F2 0C	JB7 \$00C	V. ENT BAJO
0A1	36 A1	JT0 \$0A1	PV(0)=1
0A3	26 A3	JNT0 \$0A3	PV(0)=0
0A5	0A	IN A,P2	
0A6	92 AC	JB4 \$0AC	Vout ALTO
0A8	B2 CE	JB5 \$0CE	Vout BAJO
0AA	04 9C	JMP \$09C	
0AC	B2 09	JB5 \$009	FALLA # 2
0AE	B6 9C	JF0 \$09C	LIM BAJO
0B0	CE	DEC R6	
0B1	FA	MOV A,R2	
0B2	A9	MOV R1,A	
0B3	BA 00	MOV R2,#00	
0B5	34 00	CALL \$100	SALIDA
0B7	85	CLR FO	
0B8	A5	CLR F1	
0B9	23 0F	MOV A,#0F	
0BB	5E	ANL A,R6	
0BC	96 C1	JNZ \$0C1	
0BE	95	CPL F0	
0BF	04 CB	JMP \$0C8	
0C1	FE	MOV A,R6	
0C2	17	INC A	
0C3	53 0F	ANL A,#0F	
0C5	96 C8	JNZ \$0C8	
0C7	B5	CPL F1	
0C8	FA	MOV A,R2	

```

0C9    D9          XRL A,R1
0CA    96 9A      JNZ $09A
0CC    04 9C      JMP $09C
0CE    76 9C      JF1 $09C
0D0    1E          INC R6
0D1    FA          MOV A,R2
0D2    A9          MOV R1,A
0D3    BA 01      MOV R2,#01
0D5    04 B5      JMP $0B5

```

---

```

\\RUTINA DE
      RETARDO 1 Seg

```

```

0D8    35          DIS CNT I
0D9    23 00      MOV A,#00
0DB    B8 1D      MOV R0,#1D
0DD    62          MOV T,A
0DE    55          STRT T
0DF    16 E3      JTF $0E3
0E1    04 DF      JMP $0DF
0E3    E8 DF      DJNZ R0,$0DF
0E5    65          STOP TCNT
0E6    83          RET

```

---

```

\\RUTINA DE
      RETARDO 5 Seg.

```

```

0E8    BB 05      MOV R3,#05
0EA    14 D8      CALL $0D8
0EC    EB EA      DJNZ R3,$0EA

```

OEE 83 RET

---

\\RUTINA DE

RETARDO 33 ms.

OF0 23 00 MOV A, #00  
 OF2 62 MOV T, A  
 OF3 55 STRT T  
 OF4 16 F8 JTF \$OF8  
 OF6 04 F4 JMP \$OF4  
 OF8 65 STOP TCNT  
 OF9 83 RET

---

\\RUTINA DE

RETARDO 100 ms.

170 B8 03 MOV R0, #03  
 172 14 F0 CALL \$OF0  
 174 E8 72 DJNZ R0, \$172  
 176 83 RET

---

\\RUTINA DE

SALIDA

100 23 00 MOV A, #00  
 102 62 MOV T, A  
 103 55 STRT T  
 104 FD MOV A, R5  
 105 56 15 JT1 \$115  
 107 45 07 JNT1 \$107  
 109 62 MOV T, A

```

10A  FE      MOV A,R6
10B  4F      ORL A,R7
10C  16 12   JTF $112      /SE SALE
10E  46 20   JNT1 $120     /SE FRUSTRA LA SALIDA
110  24 0C   JMP $10C
112  39      OUTL P1,A      /SE SALE
113  65      STOPTCNT
114  83      RET
115  56 15   JT1 $115
117  62      MOV T,A
118  FE      MOV A,R6
119  4F      ORL A,R7
11A  16 12   JTF $112      /SE SALE
11C  56 20   JT1 $120     /SE FRUSTRA LA SALIDA
11E  24 1A   JMP $11A
120  09      IN A,P1
121  53 0F   ANL A,#0F
123  AE      MOV R6,A
124  65      STOP TCNT
125  16 27   JTF $127      /RESET TF
127  83      RET

```

-----  
 \\MODO PRUEBA

```

130  15      DIS I
131  35      DIS TCNT I
132  B9 0F   MOV R1,#0F
134  BE 70   MOV R6,#70
136  FE      MOV A,R6

```

```

137    39          OUTL P1,A
138    34 50      CALL $150      \TEST Y DELTA
13A    1E          INC R6
13B    E9 36      DJNZ R1,$136
13D    B9 0F      MOV R1,#0F
13F    BE 7F      MOV R6,#7F
141    FE          MOV A,R6
142    39          OUTL P1,A
143    34 50      CALL $150      \TEST Y DELTA
145    CE          DEC R6
146    E9 41      DJNZ R1,$141
148    24 32      JMP $132

```

-----  
 \TEST Y DELTA

```

150    23 00      MOV A,#00
152    62          MOV T,A
153    B8 64      MOV R0,#64
155    55          STRT T
156    16 64      JTF $164
158    86 5F      JNI $15F
15A    23 FB      MOV A,#FB
15C    3A          OUTL P2,A
15D    24 56      JMP $156
15F    23 F7      MOV A,#F7
161    3A          OUTL P2,A
162    24 56      JMP $156
164    E8 56      DJNZ R0,$156
166    FE          MOV A,R6

```

```

167    43 80    ORL A,#80
169    39      OUTL P1,A
16A    16 6E    JTF 6E
16C    24 6A    JMP $16A
16E    65      STOP TCNT
16F    83      RET

```

---

\\RUTINA DE

RETARDO 100 ms.

```

170    B8 03    MOV R0,#03
172    14 F0    CALL $0F0
174    E8 72    DJNZR0 , $172
176    83      RET

```

---

\\RUTINA

FALLA # 1

```

180    23 F0    MOV A,#F0
182    39      OUTL P1,A
183    23 F7    MOV A,#F7
185    3A      OUTL P2,A
186    14 D8    CALL $0D8    DELAY 1 Seg.
188    23 F3    MOV  A,#F3
18A    3A      OUTL P2,A
18B    14 D8    CALL $0D8    DELAY 1 Seg.
18D    24 80    JMP $180

```

---

\\RUTINA FALLA # 2

```

190    23 F0    MOV A,#F0

```

```

192    39          OUTL P1,A
193    23 FB      MOV A,#FB
195    3A          OUTL P2,A
196    14 D8      CALL $0D8          DELAY 1 Seg.
198    23 FF      MOV A,#FF
19A    3A          OUTL P2,A
19B    14 D8      CALL $0D8          DELAY 1 Seg.
19D    24 90      JMP $190

```

-----

\\RUTINA DE MEDICION DEL  
PULSO DE VOLTAJE

```

1A0    23 00      MOV A,#00          [ R4 Y R5 ]
1A2    62          MOV T,A
1A3    25          EN TCNT I
1A4    55          STRT T
1A5    36 A5      JT0 $1A5
1A7    26 A7      JNT0 $1A7
1A9    62          MOV T,A
1AA    36 AA      JT0 $1AA
1AC    42          MOV A,T
1AD    AC          MOV R4,A
1AE    26 AE      JNT0 $1AE
1B0    42          MOV A,T
1B1    AD          MOV R5,A
1B2    35          DIS TCNT I
1B3    65          STOP TCNT
1B4    83          RET

```

-----

\\RUTINA DE MEDICION DEL  
PULSO DE CORRIENTE

1C0	23 00	MOV A,#00	[ R2 Y R3 ]
1C2	62	MOV T,A	
1C3	25	EN TCNT I	
1C4	55	STRT T	
1C5	46 C5	JNT1 \$1C5	
1C7	56 C7	JT1 \$1C7	
1C9	62	MOV T,A	
1CA	46 CA	JNT1 \$1CA	
1CC	42	MOV A,T	
1CD	AA	MOV R2,A	
1CE	23 00	MOV A,#00	
1D0	62	MOV T,A	
1D1	56 D1	JT1 \$1D1	
1D3	46 D3	JNT1 \$1D3	
1D5	62	MOV T,A	
1D6	56 D6	JT1 \$1D6	
1D8	42	MOV A,T	
1D9	AB	MOV R3,A	
1DA	35	DIS TCNT I	
1DB	65	STOP TCNT	
1DC	83	RET	
1E0	FA	MOV A,R2	
1E1	6B	ADD A,R3	
1E2	97	CLR	
1E3	67	RRC A	

```
1E4    37          CPL A
1E5    03 08      ADD A,#08
1E7    AD        MOV R5,A
1E8    83        RET
```

#### 4.12.5 Análisis del programa

El programa consiste de un programa principal que maneja los servicios de interrupción, el mecanismo de diagnóstico y el programa de estabilizador; además se tiene 12 subrutinas como se menciona a continuación:

Una subrutina de retardo de 1 seg. Localizada en la dirección 0D8h.

Una subrutina de retardo de 5 seg. Localizada en la dirección 0E8h.

Una subrutina de retardo de 33 mseg. Localizada en la dirección 0F0h.

Una subrutina de retardo de 100 mseg. Localizada en la dirección 170h.

Una subrutina de salida localizada en la dirección 100h.

Una subrutina de modo prueba localizada en la dirección 130h.

Una subrutina de test y delta localizada en la dirección 150h.

Una subrutina de falla 1 localizada en la dirección 180h.

Una subrutina de falla 2 localizada en la dirección 190h.

Una subrutina de medición de voltaje de cruce por cero

(PV0) en la dirección 1A0h.

Una subrutina de medición de corriente de cruce por cero (PI0) localizada en la dirección 1C0h.

Una subrutina localizada en la dirección 1E0h.

Las subrutinas de retardo se han diseñado aprovechando el contador timer interno del propio microcontrolador.

La rutina de modo prueba empieza deshabilitando las interrupciones externas y la interrupción del contador timer luego carga el registro R1 con el valor 0Fh y el registro R6 con el valor 70h, después se carga el acumulador con el contenido del registro R6, que posteriormente a su vez es sacado al puerto P1 cuyo efecto es activar a los decodificadores para proporcionar la combinación cero para la conmutación de los triacs, posteriormente se incrementa el contenido del registro R6 y se decrementa el contenido del registro R1, causando como efecto seleccionar la combinación 1 de los triacs y así sucesivamente hasta llegar a la combinación 15 con lo cual se demuestra la realización de la prueba en forma ascendente; posteriormente también lo hace en forma descendente repitiendo esta serie de pruebas hasta que el modo prueba sea desactivado.

Estas combinaciones se pueden observar mediante leds que se encuentran en cada salida de los decodificadores, la función de éstos leds es mostrar que combinación de triacs está trabajando en un momento dado.

Para que se active la rutina de modo prueba el P1.6 del puerto 1 del microcontrolador debe estar en el nivel cero.

La rutina de test y delta inicia cargando el timer con el valor cero desde el acumulador y el registro R0 con el valor 64h, luego verifica el valor de TF, y si está en 1 verifica la entrada de interrupción si ésta entrada es 1 significa que no existe ninguna interrupción entonces el acumulador se carga con el valor FBh para pasar posteriormente al puerto 2 produciendo como efecto el prendido del led normal, si el valor de la entrada de interrupción es cero el acumulador se carga con el valor F7h que luego lo pasa al puerto 2 produciendo el efecto de prender al led auxiliar indicando algún error, en caso de error se deshabilita los decodificadores que controlan a los triacs Inhibiendo a los triacs.

La rutina de medición de voltaje de cruce por cero empieza cargando el timer con cero e iniciando a correr el timer, si el timer encuentra al pulso de voltaje de cruce por cero en el nivel alto espera a que baje y si encuentra en un nivel bajo espera a que suba y justo cuando sube el timer se carga con cero, luego el timer sigue corriendo y cuando el pulso baja el timer se carga en el acumulador capturando de esta manera el valor del pulso en el estado alto que se carga en el registro R4, el timer sigue corriendo mientras el pulso esté en estado bajo y justo cuando sube el timer se carga en el acumulador, capturando así el valor deseado que luego se carga en el registro R5, en este proceso lo que se hace es contar cuantos pulsos de 0.134 mseg. Se captura ya sea cuando el PV0 está en alto o en bajo, los valores de los registros R4 y R5 son cargados

desde el timer durante el funcionamiento dinámico.

La subrutina de medición de corriente de paso por cero, si al iniciar el timer encuentra al pulso en un nivel bajo espera que suba o si encuentra en un nivel alto espera a que baje y justo cuando baja el acumulador que inicialmente era cero se carga en el timer empieza a contar el timer y justo cuando el pulso sube el timer se carga al acumulador capturando de esta manera el valor que es cargado en el registro R2; luego el timer se carga nuevamente con cero y si encuentra al pulso en el nivel alto espera a que baje o si encuentra en el nivel bajo espera a que suba y justo cuando sube empieza a correr el timer hasta que el pulso baje donde el timer se carga al acumulador capturando de esta manera el valor que es cargado en el registro R3.

#### **4.12.6 Simulación mediante Spice**

Al realizar la simulación con los circuitos de sensado de voltaje de salida y del pulso de voltaje de cruce por cero, utilizando el paquete Spice se ha obtenido el gráfico de la Fig. 24, donde se muestra los niveles de VH, VL y del PV0.

#### **4.12.7 Implementación**

##### **4.12.7.1 Circuito Impreso**

El circuito impreso es una placa de material aislante recubierto por una de sus caras con una capa de cobre, hay también circuitos impresos que disponen de cobre, por ambas caras, pero el tipo mas corriente es el que solo aparece cobreado por un solo lado. Esta capa de cobre es

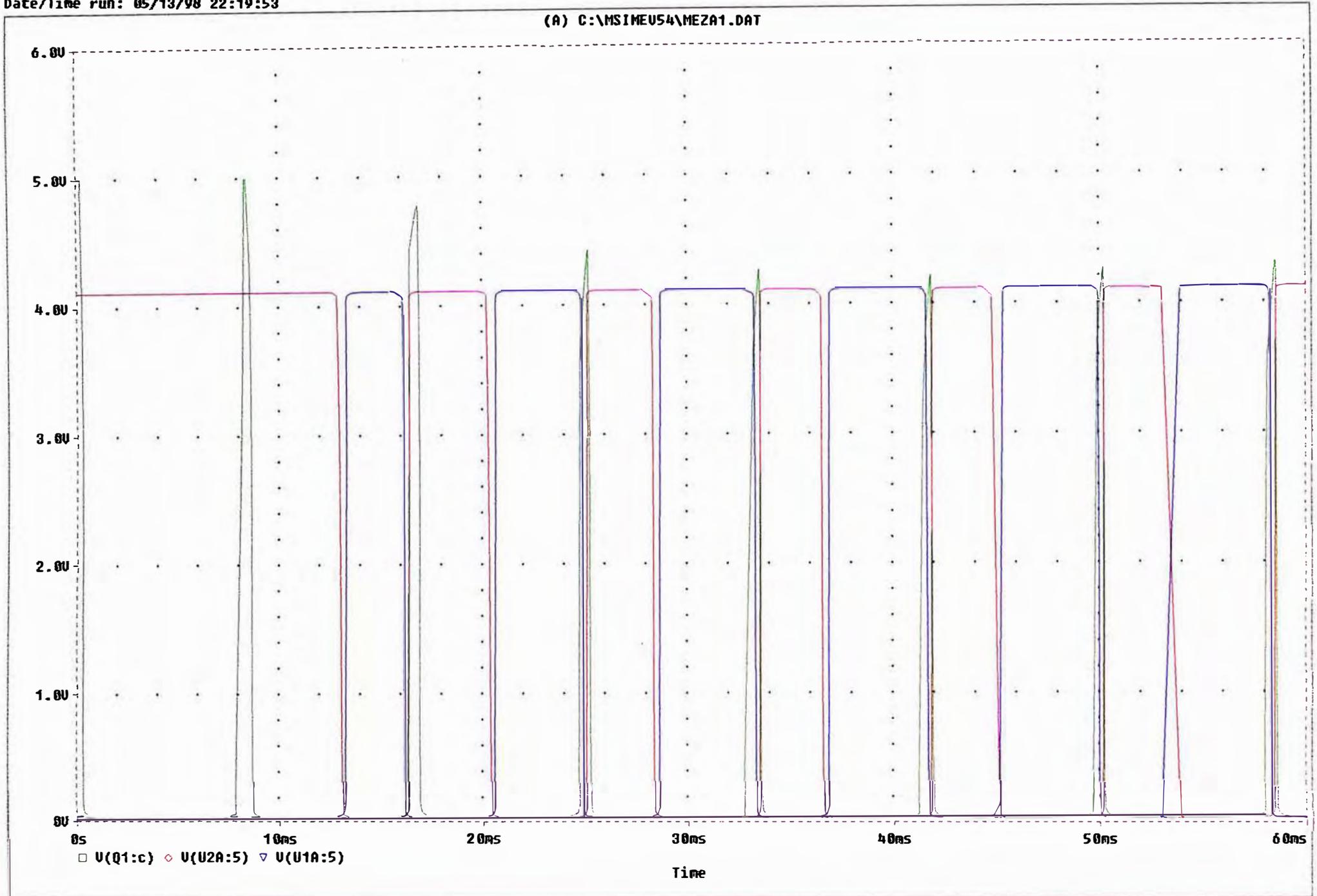


Fig. 24 Formas de onda de VH, VL y PV0

totalmente uniforme y por lo tanto, en esta situación sería como una especie de cortocircuito total no aprovechable. Para llegar a la placa útil y montarla, es preciso someterla a un proceso químico que elimine toda la toda la parte de cobre innecesaria dejando así, únicamente las pistas de unión entre los distintos componentes.

Una que se disponga de la placa con las pistas que forman las uniones, es indispensable efectuar los agujeros necesarios que permitan el paso de los terminales de los componentes desde la parte aislante hacia la del cobre, soldando los alambres o terminales mediante estaño sobre las pistas.

#### **4.12.7.2 Lado de pistas y lado de componentes**

Todo el cableado que precisa un montaje para la interconexión de sus componentes, es lo que forma el circuito impreso.

Partiendo del diagrama circuital todo el cableado correspondiente entre los componentes allí presentes se ha realizado mediante el uso del paquete llamado SMARTWORK, dicho paquete nos ha permitido realizar las pistas que unen los diversos componentes. La impresión de las pistas, se observan en las figuras 27 y 30.

Mediante un proceso fotográfico del plano de pistas, se obtiene el fotolito que es una película, la cual nos permitirá realizar la impresión de las pistas sobre la placa de material aislante en la cara cobreada.

Todo este proceso de serigrafía y químico posterior, da como resultado el circuito impreso acabado que luego se

harán los huecos respectivos.

Mediante el SMARTWORK se obtiene también los esquemas de los componentes y del SILKSCREEN.

Este último nos ha permitido imprimir sobre el circuito (proceso serigráfico) la ubicación de los componentes y sus respectivas letras indicadoras.

En la fig. 27, se muestra el dibujo del circuito impreso de una sola cara, visto desde el lado del cobre, para la tarjeta de control y en la fig. 30 para la tarjeta de fuente.

La fig. 25, muestra el dibujo del SILKSCREEN de tarjeta de control y la fig. 28 para la tarjeta de fuente.

En la fig. 26, se muestra el dibujo del lado de los componentes, para la tarjeta de control y en la fig. 29, para la tarjeta de fuente, así mismo en estos dibujos se muestran los puentes hechos para la unión de algunas pistas separadas, con el objeto de cumplir con los diagramas circuitales correspondientes.

#### **4.12.7.3 Montaje**

Para el montaje del sistema se tiene una tarjeta de control, una tarjeta de las fuentes de alimentación y el autotransformador de 8 devanados en el primario.

La tarjeta de control es del tipo de cara simple cuyas dimensiones son de 6.10" x 5.95" ó 15.5 x 15.11 cm hacemos notar que el montaje de esta tarjeta consta de 33 puentes, que se debe realizar antes de proceder a montar los componentes en su orden habitual.

Se debe tener mucho cuidado con las pistas que pasan

entre las patillas de los circuitos integrados. Es aconsejable utilizar un soldador de punta fina y poner mucho cuidado al realizar la soldadura de dichas patillas.

Ahora se pueden montar todas las resistencias, comprobando su valor óhmico, que debemos comparar siempre con las indicaciones de la lista de componentes correspondientes a cada circuito.

Después de las resistencias conviene instalar todos los diodos de silicio, con los que hay que tener mucho cuidado con su polaridad, ya que un sólo diodo montado en sentido contrario impedirá que el circuito funcione, dañando además, algún transistor o algún circuito integrado.

Después se montan aquellos diodos que tienen cuerpo de plástico donde el terminal del cátodo esta marcado con una franja blanca o de color plateado.

El siguiente paso es soldar los zócalos de los circuitos integrados, según muestra el dibujo serigráfico, teniendo cuidado en respetar la muesca, que nos sirve como referencia del pin número 1.

Después de comprobar que todos los diodos se han montado correctamente, podemos empezar a montar todos los condensadores de poliester teniendo cuidado en no confundir un valor con otro. Para continuar el montaje, podemos instalar todos los condensadores electrolíticos, teniendo cuidado de situar el terminal positivo en el orificio del circuito impreso marcado como positivo.

Luego se pueden montar todos los transistores, orientando la parte plana de sus cuerpos como figura en el

dibujo serigrafiado del circuito impreso; igualmente se hace con los fotoacopladores, y los triacs que van colocados sobre una barra disipadora; para terminar con el montaje de esta tarjeta se instalan todos los conectores; y luego se montan los circuitos integrados en sus zócalos correspondientes.

Con respecto a la tarjeta de fuentes de alimentación, también es del tipo de cara simple y cuyas dimensiones son de 6" x 5.20" ó 15 x 13.2 cm. en este caso también se siguen los pasos mencionados anteriormente para el montaje de las resistencias, condensadores y diodos de silicio además se tiene el integrado regulador VR1 que es el 7805 tiene que ir sujeto sobre una aleta de refrigeración.

Para terminar el montaje de esta tarjeta se instalan los transformadores y los conectores.

La tarjeta de control se coloca en forma vertical, utilizando dos soportes para sujetarlo dentro del módulo prototipo.

La tarjeta de fuentes de alimentación se coloca horizontalmente dentro del módulo.

El autotransformador de 8 devanados en el primario también se coloca horizontalmente en la base del módulo.

La interconexión entre la tarjeta de control, la tarjeta de las fuentes de alimentación y el autotransformador se realiza a través de los conectores.

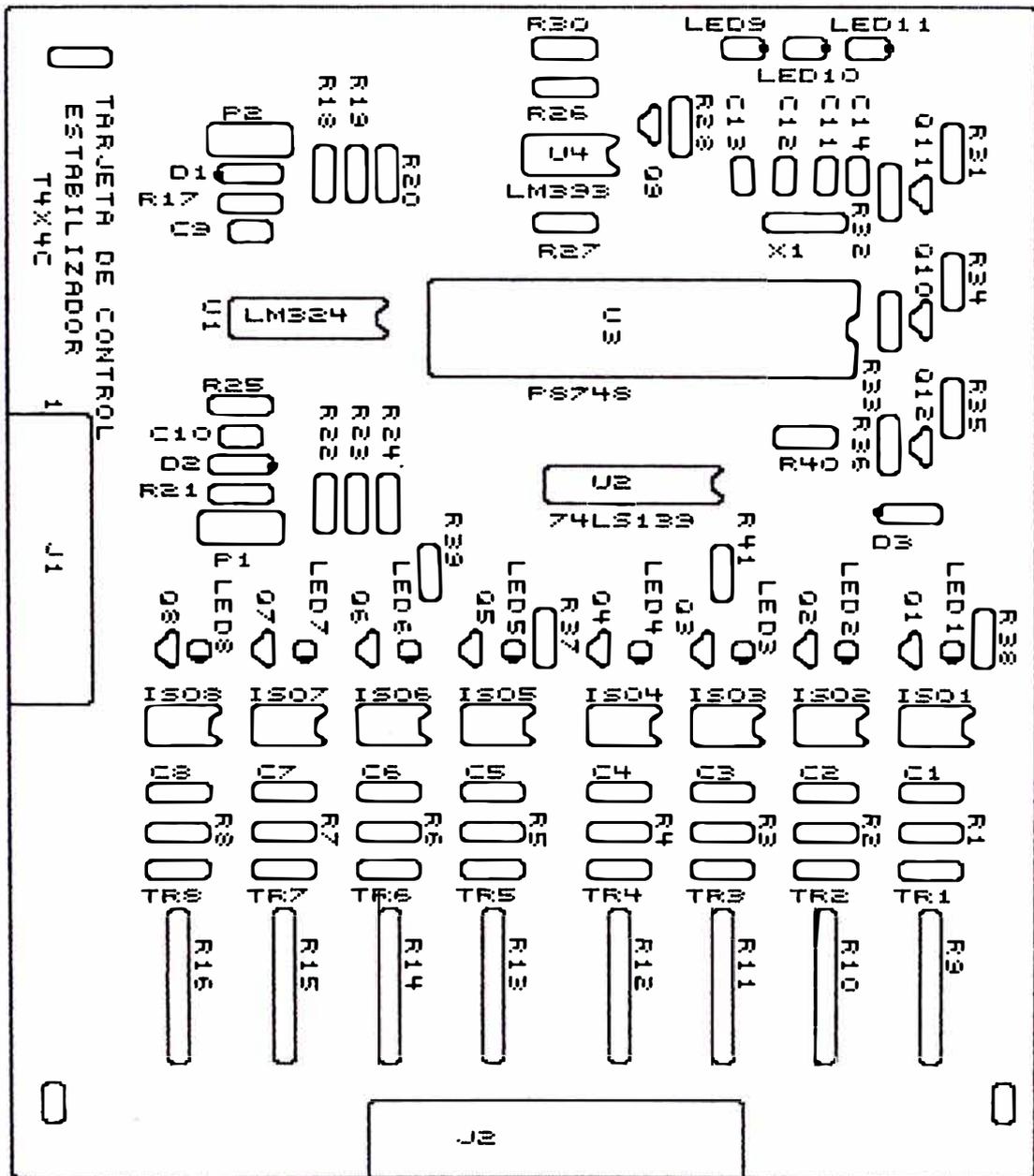


Fig. 25

Esquema del circuito impreso silkscreen  
de la tarjeta de control

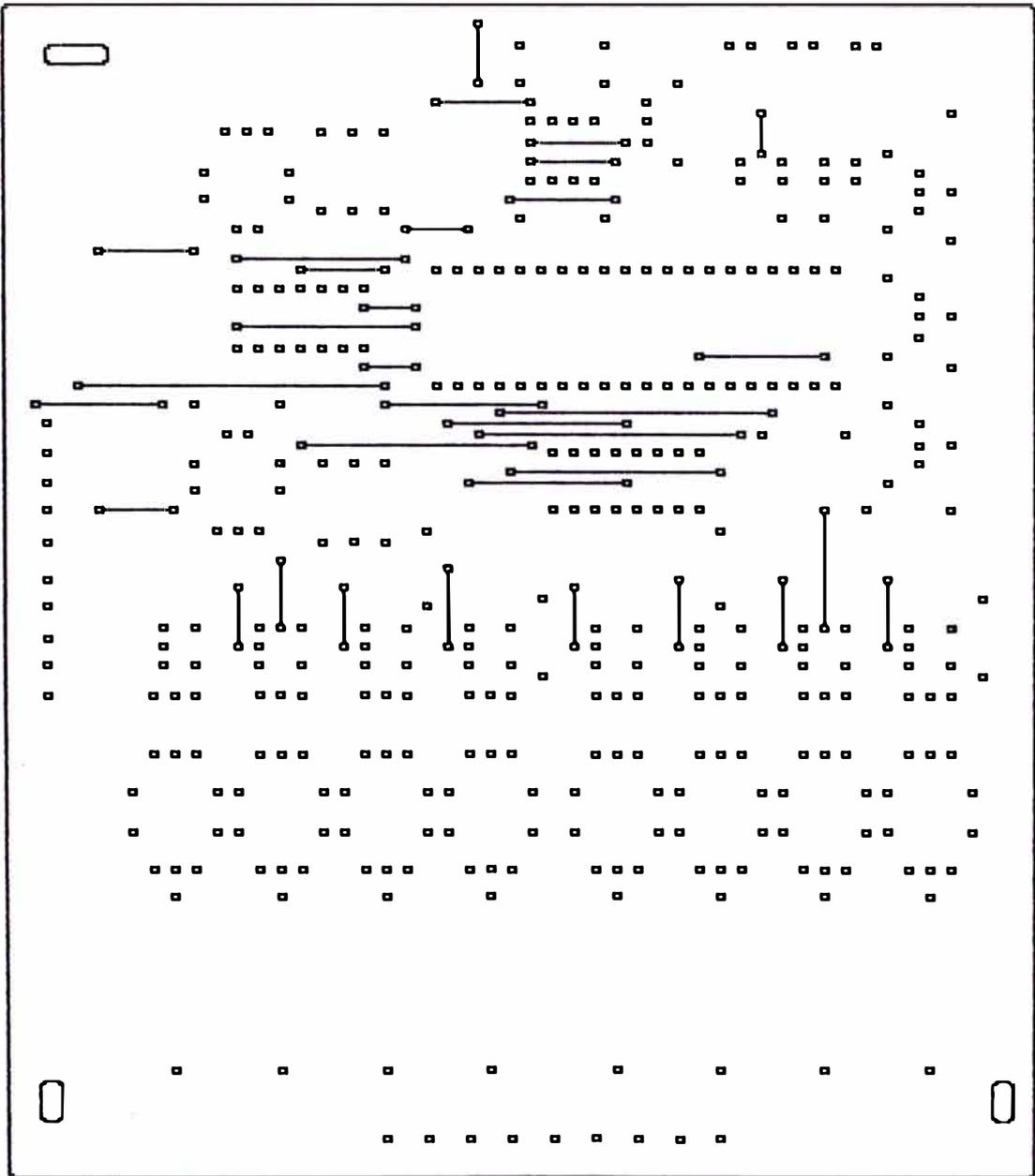


Fig. 26

Esquema del circuito impreso lado de componentes  
de la tarjeta de control

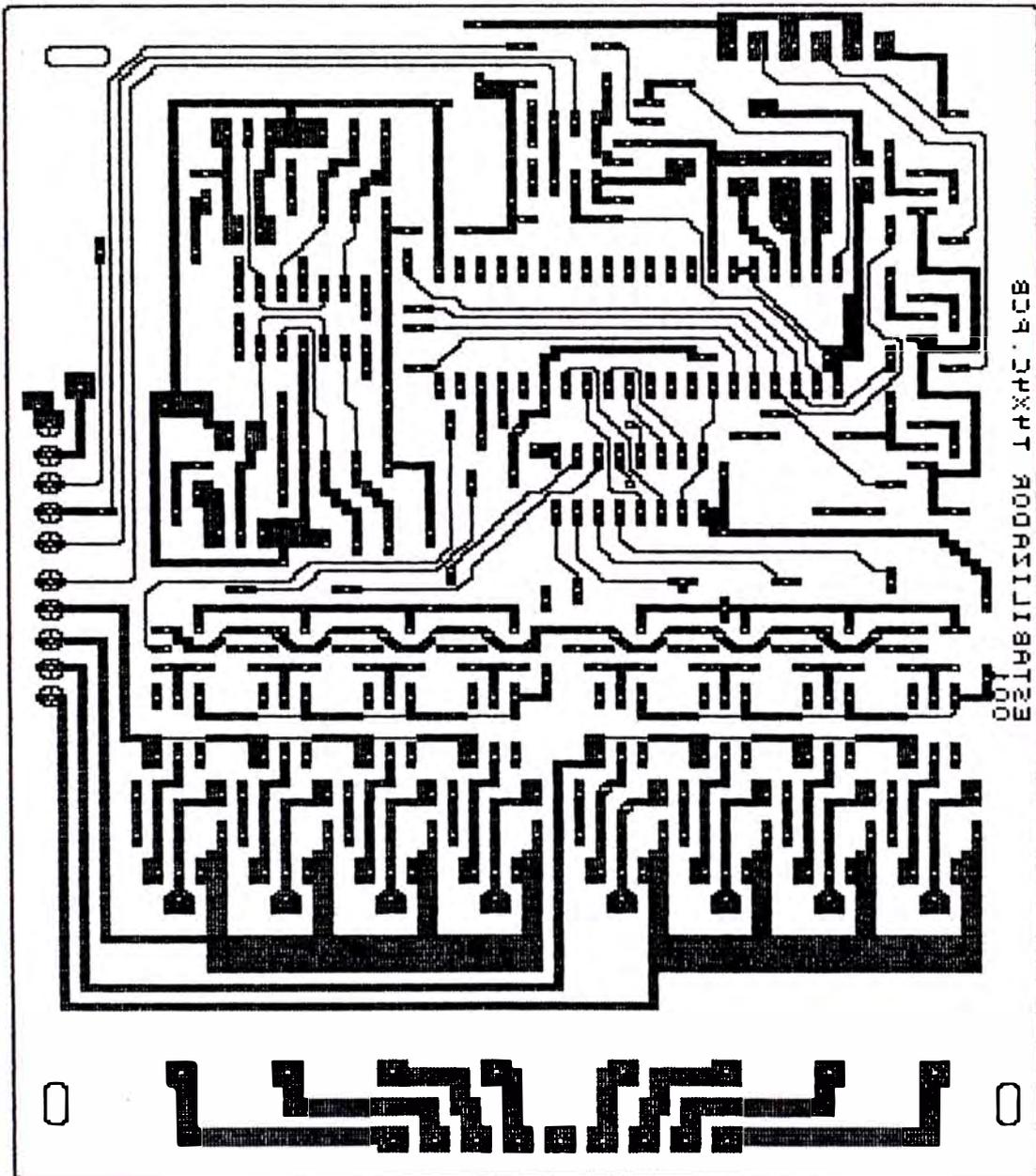


Fig. 27

Esquema del circuito impreso lado del cobre  
de la tarjeta de control

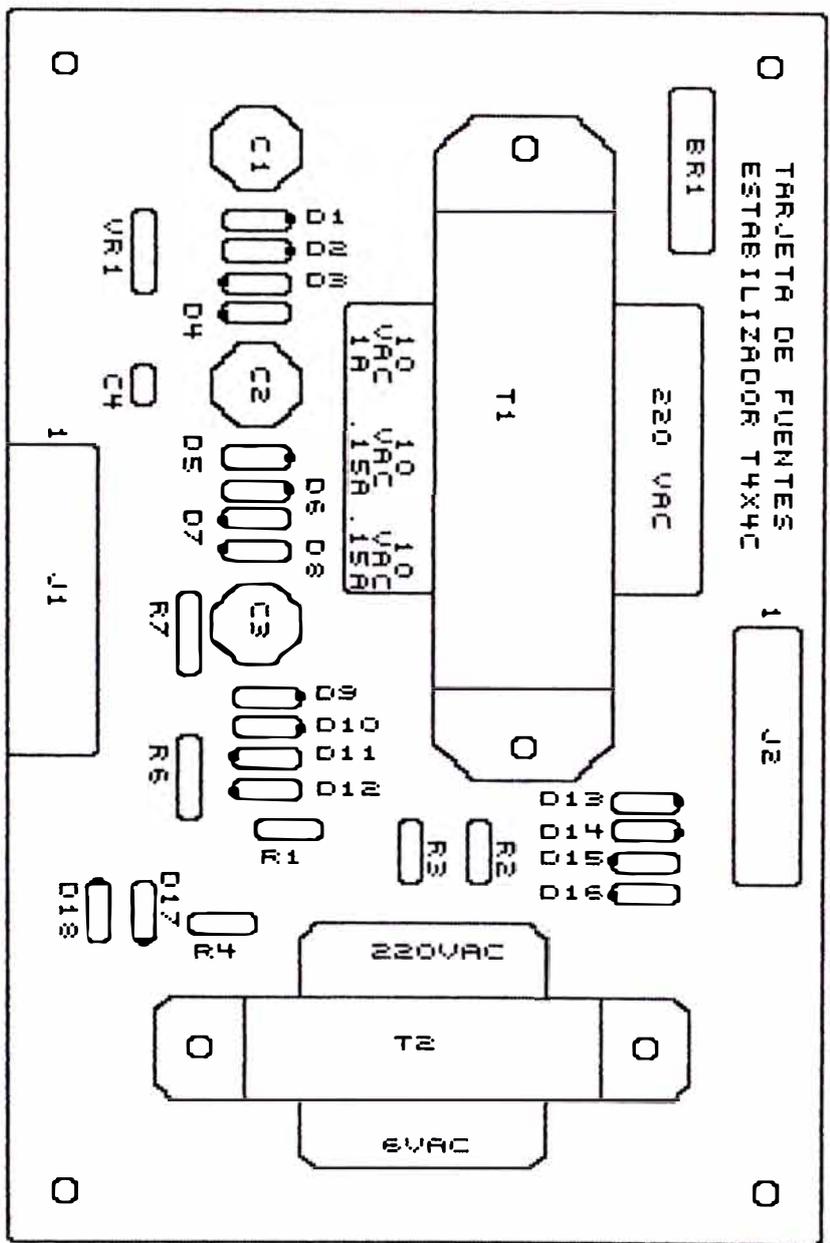


Fig. 28

Esquema del circuito impreso silkscreen  
de la tarjeta de fuente

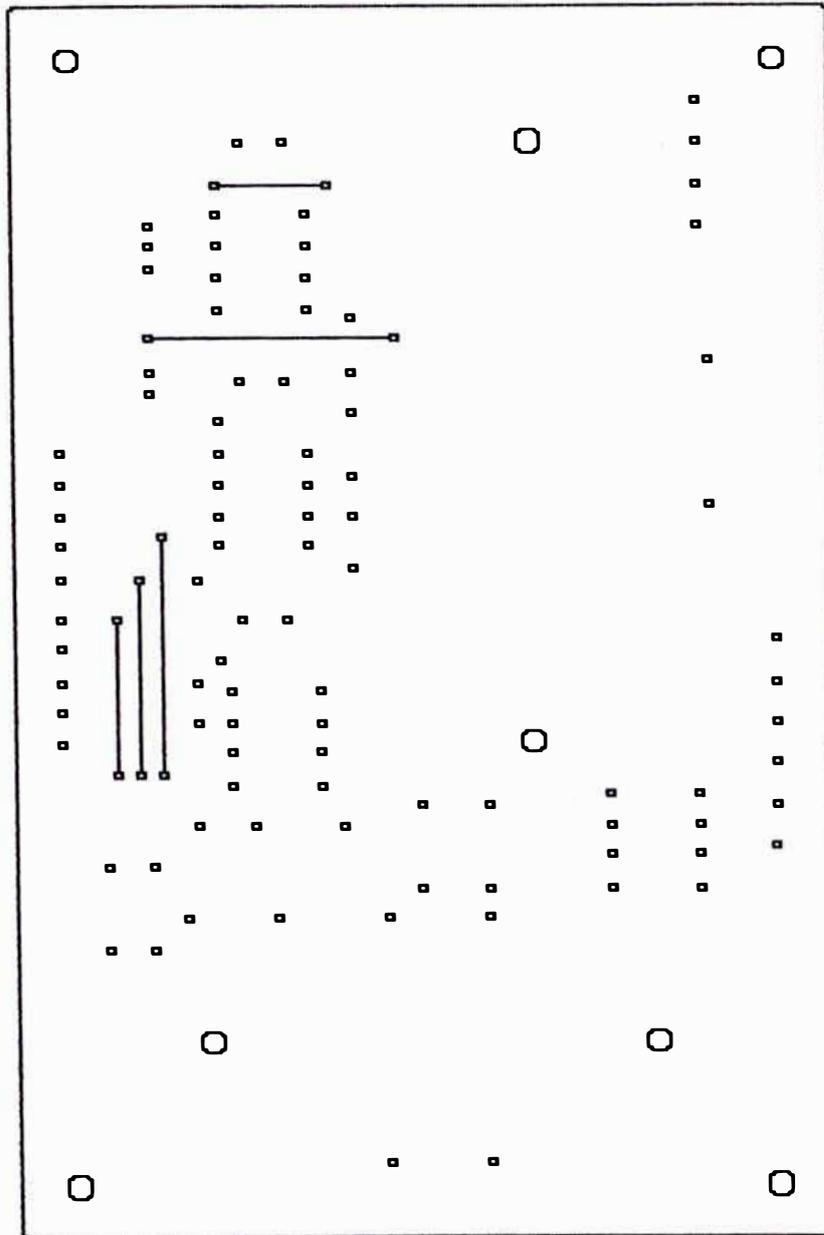


Fig. 29

Esquema del circuito impreso lado de componentes  
de la tarjeta de fuente

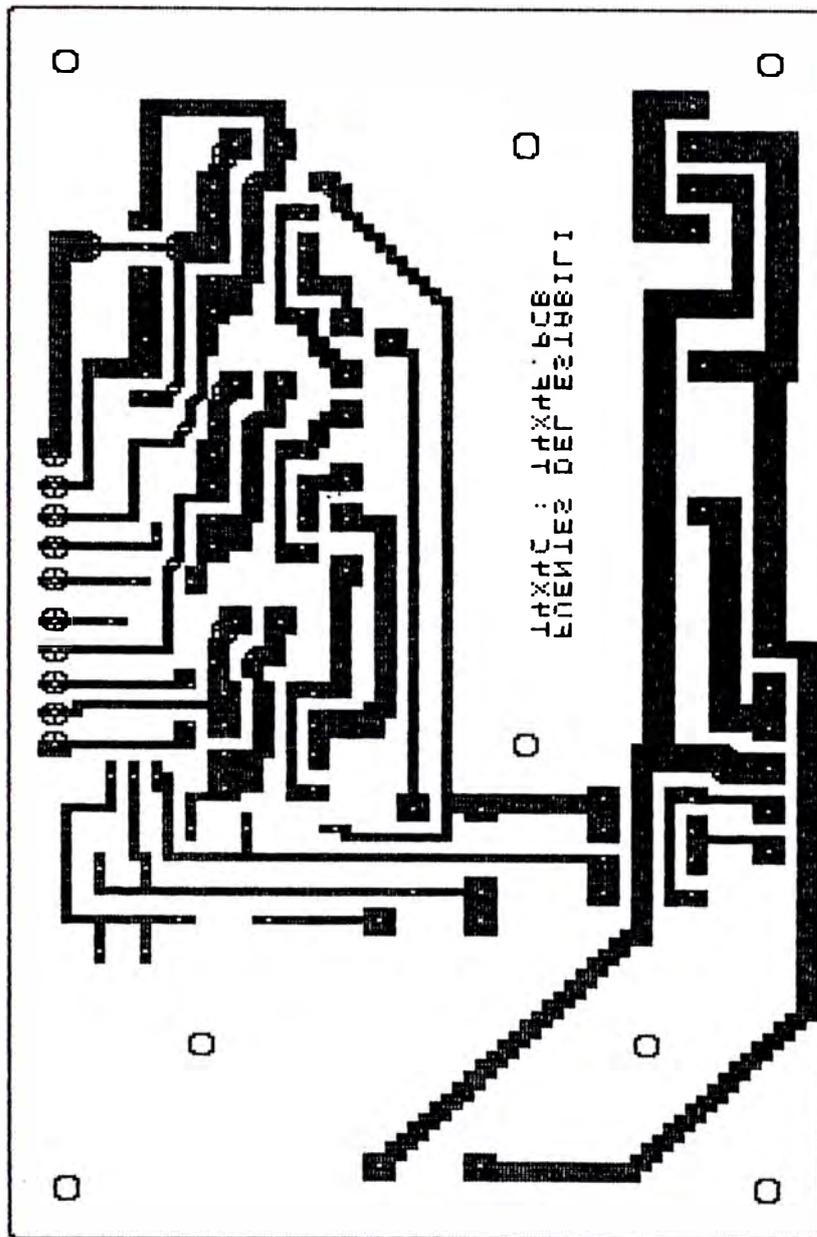


Fig. 30

Esquema del circuito impreso lado del cobre  
de la tarjeta de fuente

#### 4.12.7.4 Lista de componentes

##### TARJETA DE FUENTES DE ALIMENTACION

###### Resistencias:

R1 = 10 K $\Omega$           1/4 W.

R2 = 10 K $\Omega$           1/4 W.

R3 = 10 K $\Omega$           1/4 W.

R4 = 10 K $\Omega$           1/4 W.

R6 = 100 $\Omega$             2 W.

R7 = 100 $\Omega$             2 W.

###### Condensadores:

C1 = 1000 uF    elect.    25 Volt.

C2 = 1000 uF    elect.    25 Volt.

C3 = 1000 uF    elect.    25 Volt.

C4 = 10 uF        elect.    25 Volt.

C5 = 47 uF        elect.    25 Volt.

###### Diodos

D1 - D4 = Diodo rectificador 1N4004

D5 - D8 = Diodo rectificador 1N4004

D9 - D12 = Diodo rectificador 1N4004

D13 - D16 = Diodo rectificador 1N4004

D17 - D18 = Diodo

BR1 = Puente rectificador

VR1 = L7805CV, regulador de voltaje positivo + 5 Volt.

T1 = Transformador de 220 volt. a 10 volt. 15 A. con 3 devanados en el secundario.

T2 = Transformador de 220 volt. a 6 volt.

##### TARJETA DE CONTROL

###### Resistencias:

R1 = 100 $\Omega$	1/2 W.
R2 = 100 $\Omega$	1/2 W.
R3 = 100 $\Omega$	1/2 W.
R5 = 100 $\Omega$	1/2 W.
R6 = 100 $\Omega$	1/2 W.
R7 = 100 $\Omega$	1/2 W.
R8 = 100 $\Omega$	1/2 W.
R9 = 0.47 $\Omega$	5 W.
R10 = 0.47 $\Omega$	5 W.
R11 = 0.47 $\Omega$	5 W.
R12 = 0.47 $\Omega$	5 W.
R13 = 0.47 $\Omega$	5 W.
R14 = 0.47 $\Omega$	5 W.
R15 = 0.47 $\Omega$	5 W.
R16 = 0.47 $\Omega$	5 W.
R17 = 18 K $\Omega$	1/4 W.
R18 = 1.8 K $\Omega$	1/4 W.
R19 = 1 K $\Omega$	1/4 W.
R20 = 2.2 K $\Omega$	1/4 W.
R21 = 18 K $\Omega$	1/4 W.
R22 = 2.2 K $\Omega$	1/4 W.
R23 = 100 $\Omega$	1/4 W.
R24 = 2.7 K $\Omega$	1/4 W.
R25 = 33 K $\Omega$	1/4 W.
R26 = 5.1 K $\Omega$	1/4 W.
R27 = 5.1 K $\Omega$	1/4 W.

R28 = 33 K $\Omega$  1/4 W.

R30 = 33 K $\Omega$  1/4 W.

R31 = 470 $\Omega$  1/4 W.

R32 = 10 K $\Omega$  1/4 W.

R33 = 10 K $\Omega$  1/4 W.

R34 = 470 $\Omega$  1/4 W.

R35 = 470 $\Omega$  1/4 W.

R36 = 10 K $\Omega$  1/4 W.

R37 = 470 $\Omega$  1/4 W.

R38 = 330 $\Omega$  1/4 W.

R39 = 470 $\Omega$  1/4 W.

R40 = 10 K $\Omega$  1/4 W.

R41 = 330 $\Omega$  1/4 W.

Potenciómetros:

P1 = 50 K $\Omega$

P2 = 50 K $\Omega$

Condensadores

C1 = 0.1  $\mu$ F Poliester 250 volt.

C2 = 0.1  $\mu$ F Poliester 250 volt.

C3 = 0.1  $\mu$ F Poliester 250 volt.

C4 = 0.1  $\mu$ F Poliester 250 volt.

C5 = 0.1  $\mu$ F Poliester 250 volt.

C6 = 0.1  $\mu$ F Poliester 250 volt.

C7 = 0.1  $\mu$ F Poliester 250 volt.

C8 = 0.1  $\mu$ F Poliester 250 volt.

IS01 - IS08 = Optoacoplador T9123

TR1 - TR8 = Triac BTA08

U1A - U1D = Comparadores LM329N  
U4A - U4B = Comparadores DBL393  
U2A - U2B = Decodificadores 74LS139  
U3 = Microcontrolador D8749H  
Q1 - Q8 = Transistores 2N3908  
Q12 = Transistor 2N3904  
Q10 - Q11 = Transistores 2N3904  
Q9 = Transistor 2N3904  
LED1 - LED8 = LED  
LED9 - LED10 = LED

Transformador con 8 devanados en el primario

#### **4.12.8 Limitaciones del diseño**

El tamaño, peso y costo del transformador del equipo va creciendo a medida que aumenta la potencia.

Los triacs se emplean hasta los 400 Amp. Y para valores mayores se debe usar tiristores, ya que el tiristor es un dispositivo unidireccional y baja la potencia, porque solo conduce en la comba positiva pero con ciertos arreglos de tiristores se consigue el objetivo, pero aumentaría la complejidad del circuito y por lo tanto baja la confiabilidad, porque cuanto más componentes el sistema no es más confiable, osea de que el sistema falla por conexiones o por acoplo.

Cuando aumenta la potencia los fotoacopladores serían reemplazados por transformadores y por consiguiente aumentaría el volumen y el costo.

**4.12.9 Presupuesto del estabilizador de voltaje usando un microcontrolador.**

ITEM	DESCRIPCION Y/O ESPECIFICACION	CANT.	PRECIO UNIT.S/	PREC. TOT.S/
R1-R4	10k $\Omega$ , 1/W, resistencia	07	0.1	0.7
R6-R7	100 $\Omega$ , 2W, resistencia	02	0.1	0.2
C1-C3	1000 $\mu$ f, 25V electro. Cond.	03	0.6	1.8
C4	10 $\mu$ f, 25V electro. Cond.	01	0.5	0.5
C5	47 $\mu$ f, 25V electro. Cond.	01	0.5	0.5
D1-D16	1N4004, diodo rectificador	16	0.1	1.6
D17-D18	Diodo	02	0.1	0.2
BR1	Puente rectificador	01	2.0	2.0
VR1	L7805CV, regula de voltaje	01	2.0	2.0
	Tarjeta	01	2.0	2.0
T1	Transformador de 220/10 Volt. 15 Amp. Con 3 devan. En el secundario	01	5.0	5.0
T2	Transformad. De 220/6 volt	01	3.0	3.0
R1-R8	100 $\Omega$ , 1/2W, resistencia	08	0.1	0.8
R9-R16	0.47 $\Omega$ , 5W, resistencia	08	0.1	0.8
R18	1.8 K $\Omega$ , 1/4W, resistencia	02	0.1	0.2
R17-R21	18K $\Omega$ , 1/4W, resistencia	02	0.1	0.2
R19	1K $\Omega$ , 1/4W, resistencia	01	0.1	0.2
R20-R22	22K $\Omega$ , 1/4W resistencia	02	0.1	0.2
R23	100 $\Omega$ , 1/4W, resistencia	01	0.1	0.1
R24	2.7K $\Omega$ , 1/4W, resistencia	01	0.1	0.1

ITEM	DESCRIPCION Y/O ESPECIFICACION	CANT.	PRECIO UNIT.S/	PREC. TOT.S/
R25-R28 R30	33k $\Omega$ , 1/4W, resistencia	03	0.1	0.3
R26-R27	5.1K $\Omega$ , 1/4W, resistencia	02	0.1	0.2
R31	470 $\Omega$ , 1/4W, resistencia	05	0.1	0.5
R34-R35				
R37-R39				
R32-R33	10K $\Omega$ , 1/4W, resistencia	04	0.1	0.4
R36-R40				
R41	330 $\Omega$ , 1/4W, resistencia	01	0.1	0.1
P1-P2	50K $\Omega$ poten. De precisión	02	2.0	4.0
C1-C8	0.1 $\mu$ f, 250V.poliest.cond.	08	0.6	4.8
IS01-				
IS08	T9123 optoacoplador	08	2.84	22.72
TR1-TR8	BTA08, Triac	08	2.5	20.00
U1A-U1D	LM329N, comparador	01	2.0	2.00
U4A-U4B	DBL393, comparador	01	1.50	1.50
U2A-U2B	74LS139, decodificador	01	1.50	1.50
U3	D8748H, microcontrolador	01	10.00	10.00
Q1-Q8	2N3908, transistor	08	0.3	2.4
Q9-Q12	2N3904, transistor	04	0.3	1.2
LED1-				
LED8	Led, pequeños	08	0.2	1.6
LED9-				
LED11	Led, normales	03	0.3	0.9

ITEM	DESCRIPCION Y/O ESPECIFICACION	CANT.	PRECIO UNIT.S/	PREC. TOT.S/
	Autotransformador de			
	8 devanados en el primario	01	50.0	50.00
	Borneras	03	1.00	3.00
	Tarjeta	01	3.0	3.00
	Tomacorriente	02	1.0	2.00
	Filtro	01	7.0	7.00
	Interruptor	01	2.0	2.00
	Caja	01	10.0	10.00
Costo total en soles			S/.173.22	
Costo total en dólares			\$ 60.99	

## CONCLUSIONES Y RECOMENDACIONES

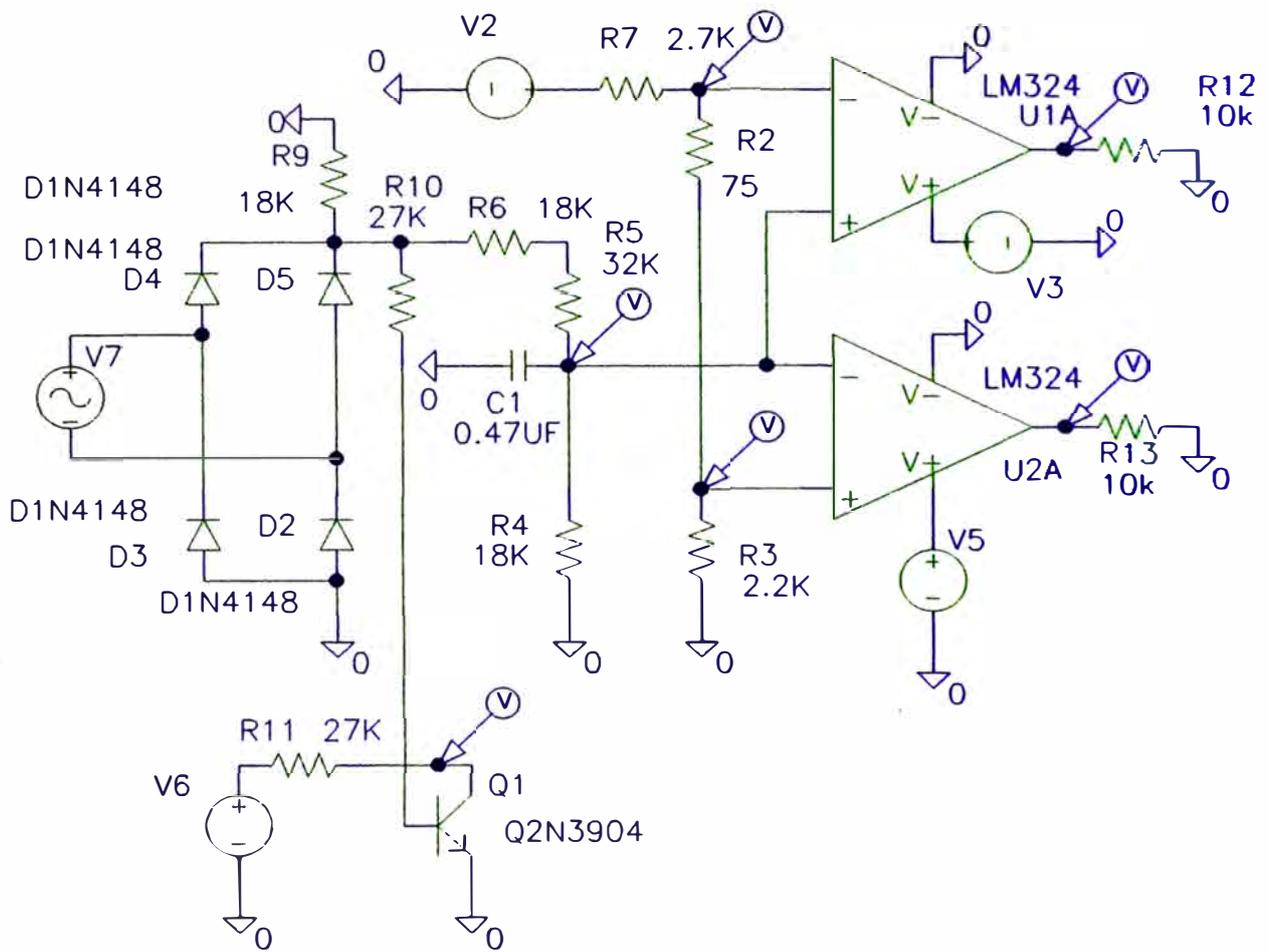
1. Uno de los objetivos de esta tesis, ha sido el de mantener una tensión de salida lo más estable posible, para ello el microcontrolador monitorea constantemente el voltaje de salida, y si hay alguna variación se corre la rutina de salida que se encarga de efectuar la corrección respectiva.
2. La mayoría de los estabilizadores de voltaje existentes en el mercado tienen su funcionamiento basado en la medición de las variaciones del voltaje de entrada que va a una serie de comparadores y al variar el voltaje de entrada conmutan uno u otro relé para hacer la compensación; mientras el funcionamiento del proyecto propuesto realiza el sensado del voltaje de salida y se realimenta al circuito de control, donde dependiendo de las variaciones del voltaje de salida se ejecuta una rutina, cuya función será la de mantener constante la salida.
3. El diseño propuesto en comparación con los sistemas tradicionales realiza los cambios con la salida, lo cual garantiza de que cualquier variación del voltaje de salida por efecto de carga será inmediatamente comunicado al microcontrolador para su respectiva solución.

4. La diferencia entre los estabilizadores híbridos y los de estado sólido, es fundamentalmente teniendo en cuenta los elementos de conmutación, si la conmutación se realiza con relé serán estabilizadores híbridos mientras que si la conmutación es a través de tiristores o triac, el estabilizador será de estado sólido.
5. Los estabilizadores que usan relé tienen algunas desventajas tales como: ocupan mayor volumen, baja velocidad de conmutación, producen chisporroteo y es peligroso el uso en ambientes inflamables, son susceptibles al ambiente de trabajo, por ejemplo el polvo con el cual pueden ensuciarse los contactos y como ventaja soportan a las radiaciones.
6. Los estabilizadores que usan triac, tienen las siguientes ventajas: alta velocidad de conmutación, no hay chisporroteo, inmune al polvo, y como desventajas son sensibles a la temperatura y radiación.
7. Este diseño puede ser generalizado para un sistema trifásico donde por cada fase se implementaría una tarjeta de control correspondiente; ya que la tarjeta de control tiene un costo aproximado de 30 dólares lo cual no influye mucho en el costo y para bajar costos se colocaría 5 triacs por cada fase para la conmutación.
8. La filosofía de diseño de este proyecto se basa en que el voltaje de salida se realimenta al circuito de control, además el sensado de la corriente que pasa

por los triacs, también se toma como parámetro de entrada para el control, entonces con estas dos informaciones se tienen los datos necesarios para hacer la conmutación en el instante preciso, el momento de hacer dicha conmutación es cuando la corriente por los triacs pasa por cero.

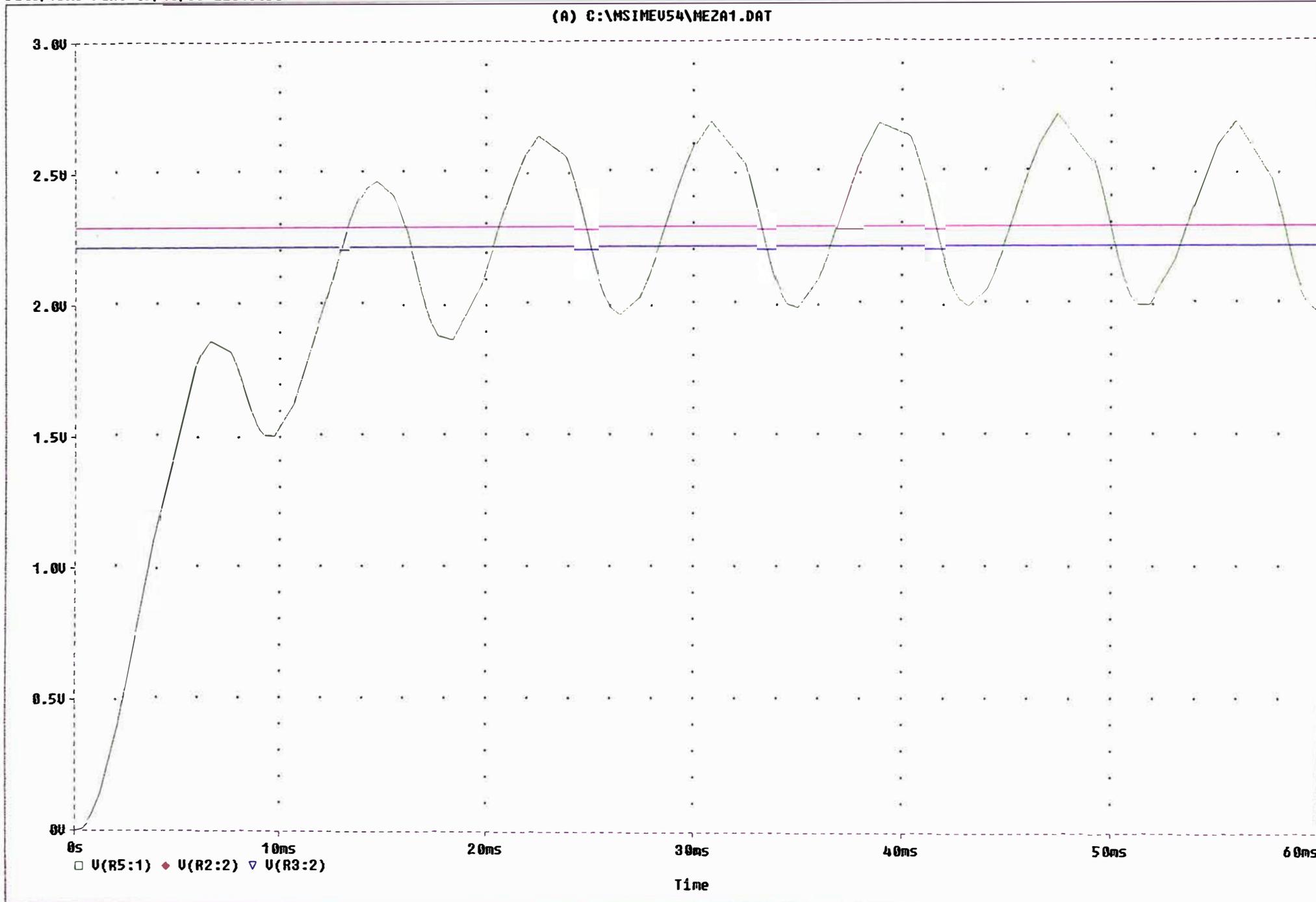
9. El circuito de control fundamentalmente está formado por el microcontrolador 8748 el cual posee una EPROM de 1 Kbyte, una RAM de 64 Bytes, 2 puertos de entrada y salida y un timer internamente, la utilización de este dispositivo para la implementación del proyecto resalta también la innovación respecto a los sistemas tradicionales.
10. El sistema por sus características de diseño y de implementación ha dado como un resultado de un sistema fiable, sencillo y sobre todo económico enmarcándose dentro de las características de todo diseño electrónico.

**ANEXO A**  
**CIRCUITO USADO PARA SIMULACION Y RESULTADOS GRAFICOS**  
**OBTENIDOS**

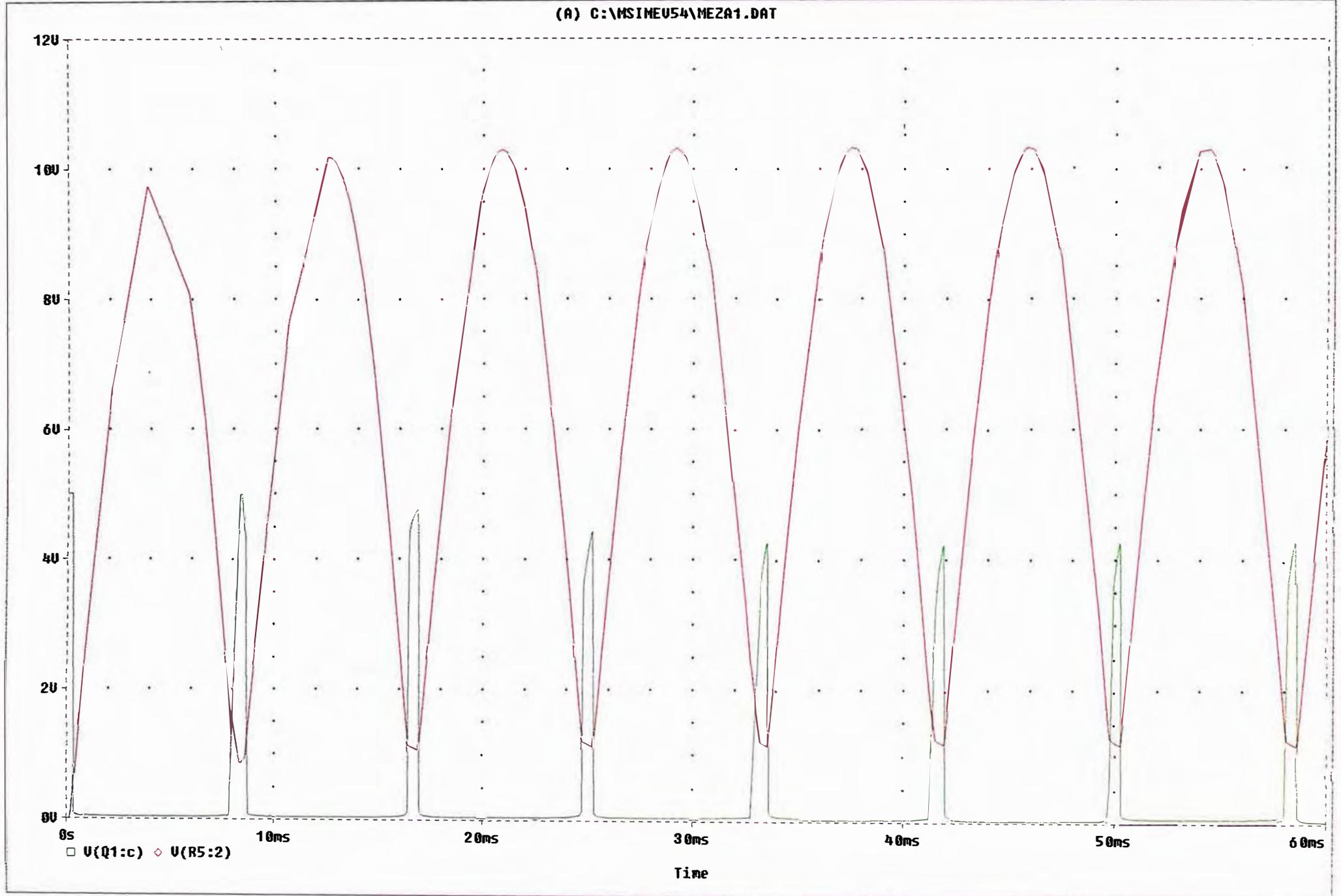


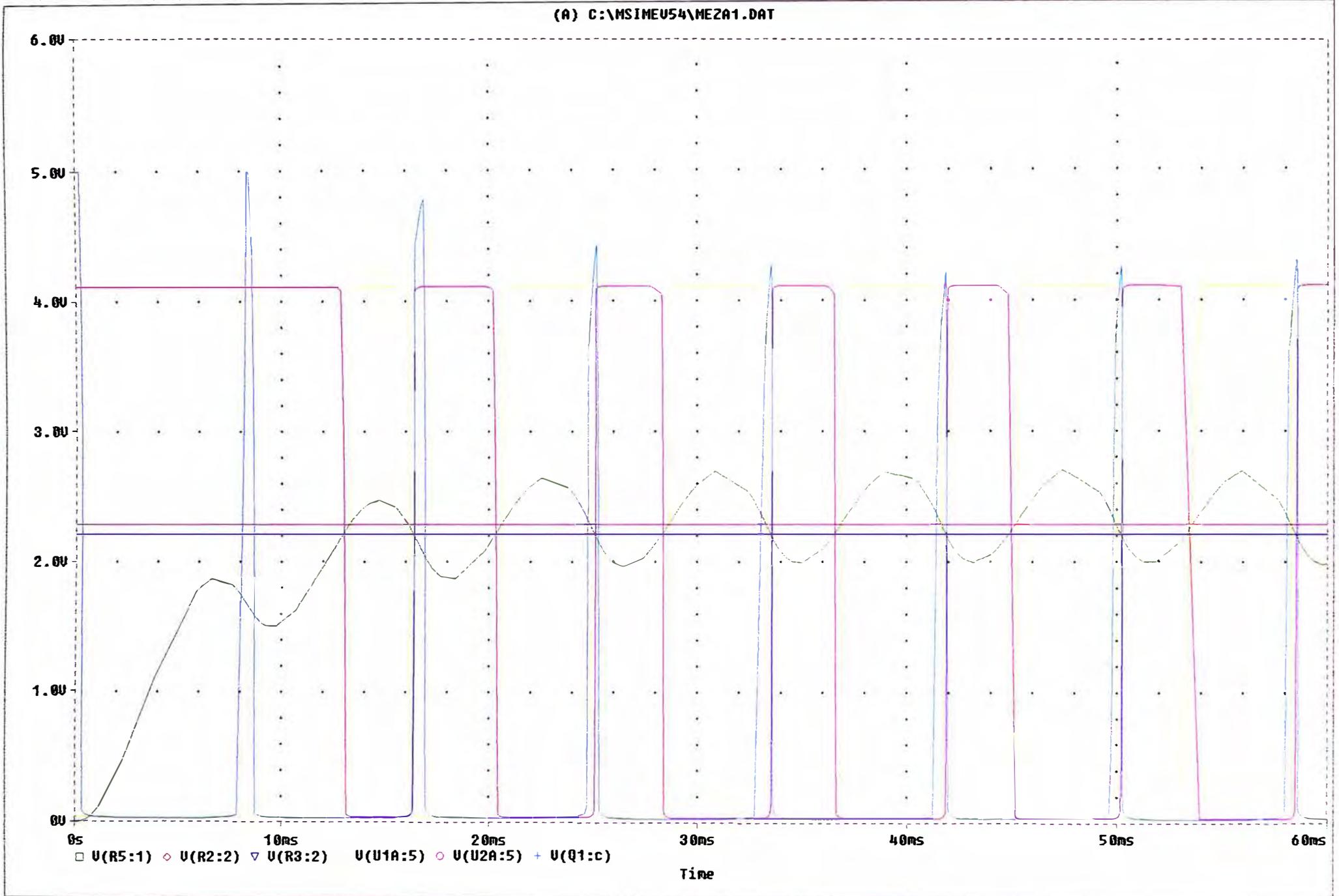
Circuito usado para simulación

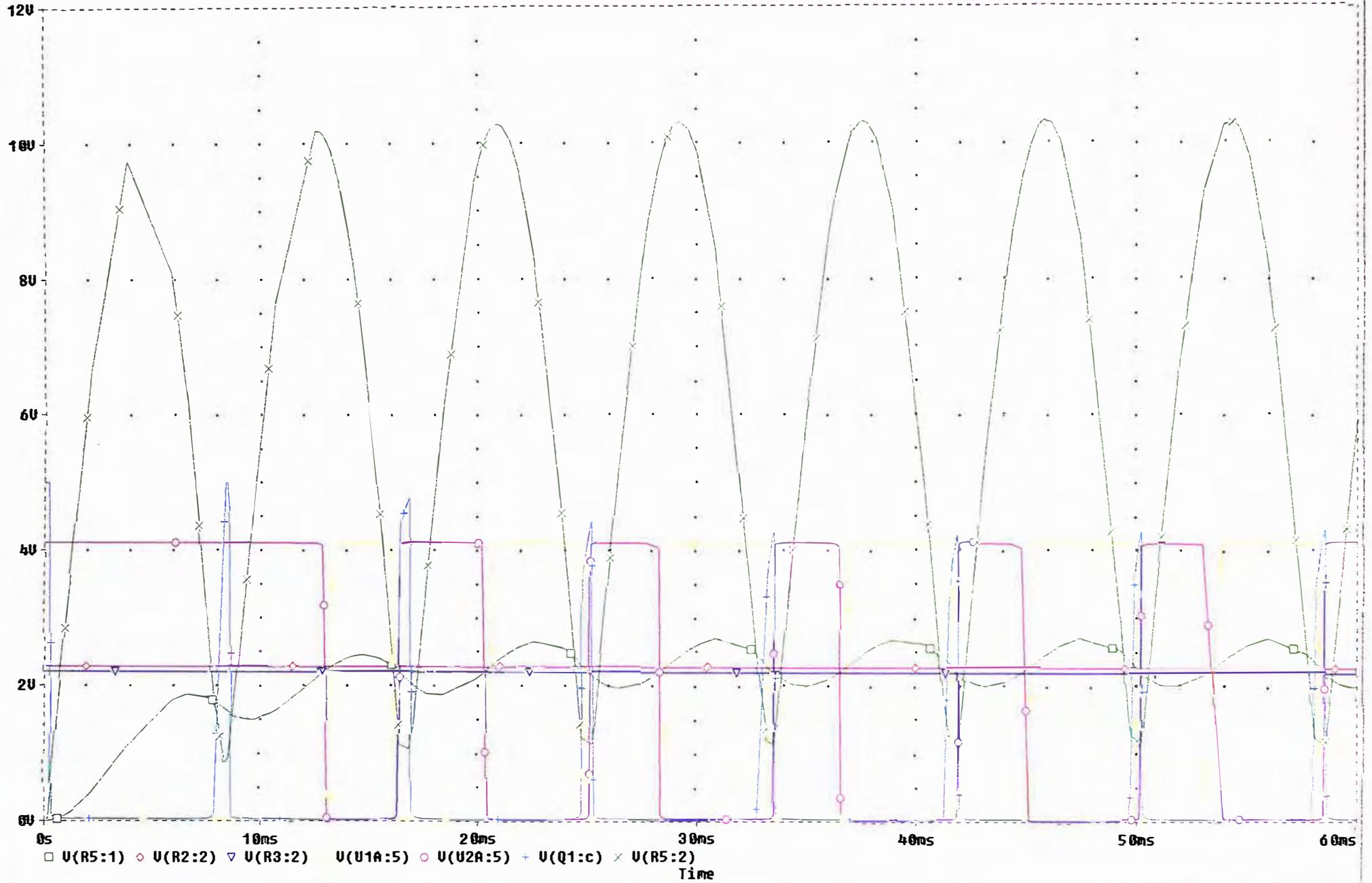
(A) C:\MSIMEU54\MEZA1.DAT



(A) C:\MSIMEU54\MEZA1.DAT







**ANEXO B**

**DIAGRAMA DE TIEMPOS DE LAS INSTRUCCIONES**

INSTRUCTION	CYCLE 1					CYCLE 2				
	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
IN A P	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	-	*INCREMENT TIMER	-	-	READ PORT	-	-	-
OUTL P, A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	-	*INCREMENT TIMER	OUTPUT TO PORT	-	-	-	-	-
ANL P, = DATA	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	-	*INCREMENT TIMER	READ PORT	FETCH IMMEDIATE DATA	-	INCREMENT PROGRAM COUNTER	*OUTPUT TO PORT	-
ORL P, = DATA	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	-	*INCREMENT TIMER	READ PORT	FETCH IMMEDIATE DATA	-	INCREMENT PROGRAM COUNTER	*OUTPUT TO PORT	-
INS A, BUS	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	-	INCREMENT TIMER	-	-	READ PORT	-	-	-
OUTL BUS, A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	-	INCREMENT TIMER	OUTPUT TO PORT	-	-	-	-	-
ANL BUS, = DATA	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	-	*INCREMENT TIMER	READ PORT	FETCH IMMEDIATE DATA	-	INCREMENT PROGRAM COUNTER	*OUTPUT TO PORT	-
ORL BUS, = DATA	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	-	*INCREMENT TIMER	READ PORT	FETCH IMMEDIATE DATA	-	INCREMENT PROGRAM COUNTER	*OUTPUT TO PORT	-
MOVX @R, A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT RAM ADDRESS	INCREMENT TIMER	OUTPUT DATA TO RAM	-	-	-	-	-
MOVX A, @R	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT RAM ADDRESS	INCREMENT TIMER	-	-	READ DATA	-	-	-
MOVD A, P <sub>1</sub>	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT OP CODE/ADDRESS	INCREMENT TIMER	-	-	READ P2 LOWER	-	-	-
MOVD P <sub>1</sub> , A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT OP CODE/ADDRESS	INCREMENT TIMER	OUTPUT DATA TO P2 LOWER	-	-	-	-	-
ANL P, A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT OP CODE/ADDRESS	INCREMENT TIMER	OUTPUT DATA	-	-	-	-	-
ORL P, A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT OP CODE/ADDRESS	INCREMENT TIMER	OUTPUT DATA	-	-	-	-	-
J(CONDITIONAL)	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	SAMPLE CONDITION	*INCREMENT SAMPLE	-	FETCH IMMEDIATE DATA	-	UPDATE PROGRAM COUNTER	-	-
START START CNT	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	-	-	START COUNTER	-	-	-	-	-
STOPTCNT	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	-	-	STOP COUNTER	-	-	-	-	-
ENI	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	-	* ENABLE INTERRUPT	-	-	-	-	-	-
DISI	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	-	* DISABLE INTERRUPT	-	-	-	-	-	-
ENTO CLK	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	-	* ENABLE CLOCK	-	-	-	-	-	-

\*VALID INSTRUCTION ADDRESSES ARE OUTPUT AT THIS TIME IF EXTERNAL PROGRAM MEMORY IS BEING ACCESSED.  
 (1) IN LATER MCS-48 DEVICES T1 IS SAMPLED IN S4.

Figure 11. 8048AH/8049AH Instruction Timing Diagram

**ANEXO C**

**JUEGO DE INSTRUCCIONES DEL 8048AH - 8748AH**

**ADD A,R<sub>r</sub> Add Register Contents to Accumulator**Encoding: 

0	1	1	0
---	---	---	---

1	r	r	r
---	---	---	---

 68H-6FH

Description: The contents of register 'r' are added to the accumulator. Carry is affected.

Operation:  $(A) \leftarrow (A) + (R_r)$   $r = 0-7$ Example: ADDR: ADD A,R6 ;ADD REG 6 CONTENTS  
;TO ACC**ADD A,@R<sub>i</sub> Add Data Memory Contents to Accumulator**Encoding: 

0	1	1	0
---	---	---	---

0	0	0	i
---	---	---	---

 60H-61H

Description: The contents of the resident data memory location addressed by register 'i' bits 0-5\*\* are added to the accumulator. Carry is affected.

Operation:  $(A) \leftarrow (A) + ((R_i))$   $i = 0-1$ Example: ADDM: MOV R0, #01FH ;MOVE '1F' HEX TO REG 0  
ADD A, @R0 ;ADD VALUE OF LOCATION  
;31 TO ACC**ADD A,#data Add Immediate Data to Accumulator**Encoding: 

0	0	0	0
---	---	---	---

0	0	1	1
---	---	---	---

d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>	d <sub>4</sub>
----------------	----------------	----------------	----------------

d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>
----------------	----------------	----------------	----------------

 03H

Description: This is a 2-cycle instruction. The specified data is added to the accumulator. Carry is affected.

Operation:  $(A) \leftarrow (A) + \text{data}$ Example: ADDI: ADD A,#ADDER: ;ADD VALUE OF SYMBOL  
;ADDER' TO ACC**ADDC A,R<sub>r</sub> Add Carry and Register Contents to Accumulator**Encoding: 

0	1	1	1
---	---	---	---

1	r	r	r
---	---	---	---

 78H-7FH

Description: The content of the carry bit is added to accumulator location 0 and the carry bit cleared. The contents of register 'r' are then added to the accumulator. Carry is affected.

Operation:  $(A) \leftarrow (A) + (R_r) + (C)$   $r = 0-7$ Example: ADDRGC: ADDC A,R4 ;ADD CARRY AND REG 4  
;CONTENTS TO ACC\*\* 0-5 in 8048AH/8748H  
0-6 in 8049AH/8749H  
0-7 in 8050AH**ADDC A,@R<sub>i</sub> Add Carry and Data Memory Contents to Accumulator**Encoding: 

0	1	1	1
---	---	---	---

0	0	0	i
---	---	---	---

 70H-71H

Description: The content of the carry bit is added to accumulator location 0 and the carry bit cleared. Then the contents of the resident data memory location addressed by register 'i' bits 0-5\*\* are added to the accumulator. Carry is affected.

Operation:  $(A) \leftarrow (A) + ((R_i)) + (C)$   $i = 0-1$ Example: ADDMC: MOV R1,#40 ;MOVE '40' DEC TO REG 1  
ADDC A,@R1 ;ADD CARRY AND LOCATION 40  
;CONTENTS TO ACC**ADDC A,@data Add Carry and Immediate Data to Accumulator**Encoding: 

0	0	0	1
---	---	---	---

0	0	1	1
---	---	---	---

d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>	d <sub>4</sub>
----------------	----------------	----------------	----------------

d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>
----------------	----------------	----------------	----------------

 13H

Description: This is a 2-cycle instruction. The content of the carry bit is added to accumulator location 0 and the carry bit cleared. Then the specified data is added to the accumulator. Carry is affected.

Operation:  $(A) \leftarrow (A) + \text{data} + (C)$ Example: ADDC A,#225 ;ADD CARRY AND '225' DEC  
;TO ACC**ANL A,R<sub>r</sub> Logical AND Accumulator with Register Mask**Encoding: 

0	1	0	1
---	---	---	---

1	r	r	r
---	---	---	---

 58H-5FH

Description: Data in the accumulator is logically ANDed with the mask contained in working register 'r'.

Operation:  $(A) \leftarrow (A) \text{ AND } (R_r)$   $r = 0-7$ Example: ANDREG: ANL A,R3 ;AND' ACC CONTENTS WITH MASK  
;IN REG 3**ANL A,@R<sub>i</sub> Logical AND Accumulator with memory Mask**Encoding: 

0	1	0	1
---	---	---	---

0	0	0	i
---	---	---	---

 50H-51H

Description: Data in the accumulator is logically ANDed with the mask 1 contained in the data memory location referenced by register 'i' bits 0-5\*\*.

Operation:  $(A) \leftarrow (A) \text{ AND } ((R_i))$   $i = 0-1$ Example: ANDDM: MOV R0,#03FH ;MOVE '3F' HEX TO REG 0  
ANL A, @R0 ;AND' ACC CONTENTS WITH  
;MASK IN LOCATION 63\*\* 0-5 in 8048AH/8748H  
0-6 in 8049AH/8749H  
0-7 in 8050AH

**ANL A,#data Logical AND Accumulator with Immediate Mask**

Encoding: 

0	1	0	1
---	---	---	---

0	0	0	1
---	---	---	---

d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>	d <sub>4</sub>
----------------	----------------	----------------	----------------

d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>
----------------	----------------	----------------	----------------

 53H

Description: This is a 2-cycle instruction. Data in the accumulator is logically ANDed with an immediately-specified mask.

Operation: (A) ← (A) AND data

Examples: ANDID: ANL A,#0AFH ;'AND' ACC CONTENTS  
;WITH MASK 10101111  
ANL A,#3 + X/Y ;'AND' ACC CONTENTS  
;WITH VALUE OF EXP  
;'3 + XY/Y'

**ANL BUS,#data\* Logical AND BUS with Immediate Mask**

Encoding: 

1	0	0	1
---	---	---	---

1	0	0	0
---	---	---	---

d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>	d <sub>4</sub>
----------------	----------------	----------------	----------------

d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>
----------------	----------------	----------------	----------------

 98H

Description: This is a 2-cycle instruction. Data on the BUS port is logically ANDed with an immediately-specified mask. This instruction assumes prior specification of an 'OUTL BUS, A' instruction.

Operation: (BUS) ← (BUS) AND data

Example: ANDBUS: ANL BUS,#MASK ;'AND' BUS CONTENTS  
;WITH MASK EQUAL VALUE  
;OF SYMBOL 'MASK'

**ANL Pp,#data Logical AND Port 1-2 with Immediate Mask**

Encoding: 

1	0	0	1
---	---	---	---

1	0	p	p
---	---	---	---

d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>	d <sub>4</sub>
----------------	----------------	----------------	----------------

d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>
----------------	----------------	----------------	----------------

 99H-9AH

Description: This is a 2-cycle instruction. Data on port 'p' is logically ANDed with an immediately-specified mask.

Operation: (Pp) ← (Pp) AND DATA p = 1-2

Example: ANDP2: ANL P2,#0F0H ;'AND' PORT 2 CONTENTS  
;WITH MASK 'F0' HEX  
;(CLEAR P20-23)

\* For use with internal program memory ONLY.

**ANLD Pp,A Logical AND Port 4-7 with Accumulator Mask**

Encoding: 

1	0	0	1
---	---	---	---

1	1	p	p
---	---	---	---

 9CH-9FH

Description: This is a 2-cycle instruction. Data on port 'p' is logically ANDed with the digit mask contained in accumulator bits 0-3.

Operation: (Pp) ← (Pp) AND (A0-3) p = 4-7

Note: The mapping of port 'p' to opcode bits 0-1 is as follows:

Port	0-1
4	00
5	01
6	10
7	11

Example: ANDP4: ANLD P4,A ;'AND' PORT 4 CONTENTS  
;WITH ACC BITS 0-3

**CALL address Subroutine Call**

Encoding: 

a <sub>10</sub>	a <sub>9</sub>	a <sub>8</sub>	1
-----------------	----------------	----------------	---

0	1	0	0
---	---	---	---

a <sub>7</sub>	a <sub>6</sub>	a <sub>5</sub>	a <sub>4</sub>
----------------	----------------	----------------	----------------

a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>
----------------	----------------	----------------	----------------

Page Hex Op Code

0	14
1	34
2	54
3	74
4	94
5	B4
6	D4
7	F4

Description: This is a 2-cycle instruction. The program counter and PSW bits 4-7 are saved in the stack. The stack pointer (PSW bits 0-2) is updated. Program control is then passed to the location specified by 'address'. PC bit 11 is determined by the most recent SEL MB instruction.

A CALL cannot begin in locations 2046-2047 or 4094-4095. Execution continues at the instruction following the CALL upon return from the subroutine.

Operation: ((SP)) ← (PC), (PSW<sub>4-7</sub>)  
(SP) ← (SP) + 1  
(PC<sub>8-10</sub>) ← (addr<sub>8-10</sub>)  
(PC<sub>0-7</sub>) ← (addr<sub>0-7</sub>)  
(PC<sub>11</sub>) ← DBF

MCS<sup>®</sup>-48 INSTRUCTION SET

**Example:** Add three groups of two numbers. Put subtotals in locations 50, 51 and total in location 52.

```

MOV R0,#50      ;MOVE '50' DEC TO ADDRESS
                ;REG 0
BEGADD: MOV A,R1 ;MOVE CONTENTS OF REG 1
                ;TO ACC
          ADD A,R2 ;ADD REG 2 TO ACC
          CALL SUBTOT ;CALL SUBROUTINE 'SUBTOT'
          ADDC A,R3 ;ADD REG 3 TO ACC
          ADDC A,R4 ;ADD REG 4 TO ACC
          CALL SUBTOT ;CALL SUBROUTINE 'SUBTOT'
          ADDC A,R5 ;ADD REG 5 TO ACC
          ADDC A,R6 ;ADD REG 6 TO ACC
          CALL SUBTOT ;CALL SUBROUTINE 'SUBTOT'
SUBTOT: MOV @R0,A ;MOVE CONTENTS OF ACC TO
                ;LOCATION ADDRESSED BY
                ;REG 0
          INC R0   ;INCREMENT REG 0
          RET     ;RETURN TO MAIN PROGRAM

```

## CLR A Clear Accumulator

Encoding: 

0	0	1	0
---	---	---	---

0	1	1	1
---	---	---	---

 27H

Description: The contents of the accumulator are cleared to zero.

Operation:  $A \leftarrow 0$

## CLR C Clear Carry Bit

Encoding: 

1	0	0	1
---	---	---	---

0	1	1	1
---	---	---	---

 97H

Description: During normal program execution, the carry bit can be set to one by the ADD, ADDC, RLC, CPL C, RRC, and DAA instructions. This instruction resets the carry bit to zero.

Operation:  $C \leftarrow 0$

## CLR F1 Clear Flag 1

Encoding: 

1	0	1	0
---	---	---	---

0	1	0	1
---	---	---	---

 A5H

Description: Flag 1 is cleared to zero.

Operation:  $(F1) \leftarrow 0$

MCS<sup>®</sup>-48 INSTRUCTION SET

## CLR F0 Clear Flag 0

Encoding: 

1	0	0	0
---	---	---	---

0	1	0	1
---	---	---	---

 85H

Description: Flag 0 is cleared to zero.

Operation:  $(F0) \leftarrow 0$

## CPL A Complement Accumulator

Encoding: 

0	0	1	1
---	---	---	---

0	1	1	1
---	---	---	---

 37H

Description: The contents of the accumulator are complemented. This is strictly a one's complement. Each one is changed to zero and vice-versa.

Operation:  $(A) \leftarrow \text{NOT}(A)$

Example: Assume accumulator contains 01101010.

CPLA: CPL A ;ACC CONTENTS ARE COMPLEMENTED TO 10010101

## CPL C Complement Carry Bit

Encoding: 

1	0	1	0
---	---	---	---

0	1	1	1
---	---	---	---

 A7H

Description: The setting of the carry bit is complemented; one is changed to zero, and zero is changed to one.

Operation:  $(C) \leftarrow \text{NOT}(C)$

Example: Set C to one; current setting is unknown.

CTO1: CLR C ;C IS CLEARED TO ZERO  
CPL C ;C IS SET TO ONE

## CPL F0 Complement Flag 0

Encoding: 

1	0	0	1
---	---	---	---

0	1	0	1
---	---	---	---

 95H

Description: The setting of flag 0 is complemented; one is changed to zero, and zero is changed to one.

Operation:  $F0 \leftarrow \text{NOT}(F0)$

## CPL F1 Complement Flag 1

Encoding: 

1	0	1	1
---	---	---	---

0	1	0	1
---	---	---	---

 B5H

Description: The setting of flag 1 is complemented; one is changed to zero, and zero is changed to one.

Operation:  $(F1) \leftarrow \text{NOT}(F1)$

## DA A Decimal Adjust Accumulator

Encoding: 

0	1	0	1	0	1	1	1
---	---	---	---	---	---	---	---

 57H

Description: The 8-bit accumulator value is adjusted to form two 4-bit Binary Coded Decimal (BCD) digits following the binary addition of BCD numbers. The carry bit C is affected. If the contents of bits 0-3 are greater than nine, or if AC is one, the accumulator is incremented by six.

The four high-order bits are then checked. If bits 4-7 exceed nine, or if C is one, these bits are increased by six. If an overflow occurs, C is set to one.

Example: Assume accumulator contains 10011011.

```
DA A                ;ACC Adjusted to 00000001
                   ;WITH C SET
```

C	AC	7	4	3	0		
0	0	1	0	0	1	1	0
				0	0	0	0
						1	1
0	1	1	0	1	0	0	0
				0	1	1	0
1	0	0	0	0	0	0	0

ADD SIX TO BITS 0-7

ADD SIX TO BITS 4-7

OVERFLOW TO C

## DEC A Decrement Accumulator

Encoding: 

0	0	0	0	0	1	1	1
---	---	---	---	---	---	---	---

 07H

Description: The contents of the accumulator are decremented by one. The carry flag is not affected.

Operation: (A) ← (A) - 1

Example: Decrement contents of external data memory location 63.

```
MOV R0,#3FH        ;MOVE '3F' HEX TO REG 0
MOVX A,@R0         ;MOVE CONTENTS OF
                   ;LOCATION 63 TO ACC
DEC A              ;DECREMENT ACC
MOVX @R0,A         ;MOVE CONTENTS OF ACC TO
                   ;LOCATION 63 IN EXPANDED
                   ;MEMORY
```

## DEC Rr Decrement Register

Encoding: 

1	1	0	0	1	r	r	r
---	---	---	---	---	---	---	---

 C8H-CFH

Description: The contents of working register 'r' are decremented by one.

Operation: (Rr) ← (Rr) - 1      r = 0-7

Example: DEC R1      ;DECREMENT CONTENTS OF REG 1

## DISI External Interrupt

Encoding: 

0	0	0	1	0	1	0	1
---	---	---	---	---	---	---	---

 15H

Description: External interrupts are disabled. A low signal on the interrupt input pin has no effect.

## DIS TCNTI Disable Timer/Counter Interrupt

Encoding: 

0	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---

 35H

Description: Timer/counter interrupts are disabled. Any pending timer interrupt request is cleared. The interrupt sequence is not initiated by an overflow, but the timer flag is set and time accumulation continues.

## DJNZ Rr, address Decrement Register and Test

Encoding: 

1	1	1	0	1	r	r	r
---	---	---	---	---	---	---	---

a <sub>7</sub>	a <sub>6</sub>	a <sub>5</sub>	a <sub>4</sub>	a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

 E8H-EFH

Description: This is a 2-cycle instruction. Register 'r' is decremented, then tested for zero. If the register contains all zeros, program control falls through to the next instruction. If the register contents are not zero, control jumps to the specified 'address'.

The address in this case must evaluate to 8-bits, that is, the jump must be to a location within the current 256-location page.

Example: (Rr) ← (Rr) - 1      r = 0-7

If Rr not 0  
(PC<sub>0-7</sub>) ← addr

Note: A 12-bit address specification does not cause an error if the DJNZ instruction and the jump target are on the same page. If the DJNZ instruction begins in location 255 of a page, it must jump to a target address on the following page.

Example: Increment values in data memory locations 50-54.

```
MOV R0,#50        ;MOVE '50' DEC TO ADDRESS
                  ;REG 0
MOV R3,#5         ;MOVE '5' DEC TO COUNTER
                  ;REG 3
INCR: INC @R0     ;INCREMENT CONTENTS OF
                  ;LOCATION ADDRESSED BY
                  ;REG 0
INC R0            ;INCREMENT ADDRESS IN REG 0
DJNZ R3, INCR    ;DECREMENT REG 3 — JUMP TO
                  ;'INCR' IF REG 3 NONZERO
NEXT —          ;'NEXT' ROUTINE EXECUTED
                  ;IF R3 IS ZERO
```

**EN I Enable External Interrupt**Encoding: 

0	0	0	0	0	1	0	1
---	---	---	---	---	---	---	---

 05H

Description: External interrupts are enabled. A low signal on the interrupt input pin initiates the interrupt sequence.

**EN TCNTI Enable Timer/Counter Interrupt**Encoding: 

0	0	1	0	0	1	0	1
---	---	---	---	---	---	---	---

 25H

Description: Timer/counter interrupts are enabled. An overflow of the timer/counter initiates the interrupt sequence.

**ENT0 CLK Enable Clock Output**Encoding: 

0	1	1	1	0	1	0	1
---	---	---	---	---	---	---	---

 75H

Description: The test 0 pin is enabled to act as the clock output. This function is disabled by a system reset.

Example: EMTST0: ENT0 CLK ;ENABLE T0 AS CLOCK OUTPUT

**IN A,Pp Input Port or Data to Accumulator**Encoding: 

0	0	0	0	1	0	p	p
---	---	---	---	---	---	---	---

 09H-0AH

Description: This is a 2-cycle instruction. Data present on port 'p' is transferred (read) to the accumulator.

Operation: (A) ← (Pp)                      p = 1-2

INP12: IN A,P1                            ;INPUT PORT 1 CONTENTS TO ACC

          MOV R6,A                        ;MOVE ACC CONTENTS TO REG 6

          IN A,P2                        ;INPUT PORT 2 CONTENTS TO ACC

          MOV R7,A                        ;MOVE ACC CONTENTS TO REG 7

**INC A Increment Accumulator**Encoding: 

0	0	0	1	0	1	1	1
---	---	---	---	---	---	---	---

 17H

Description: The contents of the accumulator are incremented by one. Carry is not affected.

Operation: (A) ← (A) + 1

Example: Increment contents of location 100 in external data memory.

```
INCA: MOV R0,#100      ;MOVE '100' DEC TO ADDRESS REG 0
      MOVX A,@R0       ;MOVE CONTENTS OF LOCATION
                       ;100 TO ACC
      INC A            ;INCREMENT A
      MOVX @R0,A       ;MOVE ACC CONTENTS TO
                       ;LOCATION 101
```

**INC R<sub>r</sub> Increment Register**Encoding: 

0	0	0	1	1	r	r	r
---	---	---	---	---	---	---	---

 18H-1FH

Description: The contents of working register 'r' are incremented by one.

Operation: (Rr) ← (Rr) + 1                      r = 0-7

Example: INCR0: INC R0 ;INCREMENT CONTENTS OF REG 0

**INC @R<sub>i</sub> Increment Data Memory Location**Encoding: 

0	0	0	1	0	0	0	i
---	---	---	---	---	---	---	---

 10H-11H

Description: The contents of the resident data memory location addressed by register 'i' bits 0-5\*\* are incremented by one.

Operation: ((Ri)) ← ((Ri)) + 1                      i = 0-1

Example: INCDM: MOV R1,#03FH ;MOVE ONES TO REG 1

          INC @R1 ;INCREMENT LOCATION 63

**INS A,BUS\* Strobed Input of BUS Data to Accumulator**Encoding: 

0	0	0	0	1	0	0	0
---	---	---	---	---	---	---	---

 08H

Description: This is a 2-cycle instruction. Data present on the BUS port is transferred (read) to the accumulator when the RD pulse is dropped. (Refer to section on programming memory expansion for details.)

Operation: (A) ← (BUS)

Example: INPBUS: INS A,BUS ;INPUT BUS CONTENTS TO ACC

\* For use with internal program memory ONLY.

\*\* 0-5 in 8048AH/8748H

    0-6 in 8049AH/8749H

    0-7 in 8050AH

**JBb address Jump If Accumulator Bit Is Set**

Encoding: 

b <sub>2</sub> b <sub>1</sub> b <sub>0</sub> 1	0 0 1 0	a <sub>7</sub> a <sub>6</sub> a <sub>5</sub> a <sub>4</sub>	a <sub>3</sub> a <sub>2</sub> a <sub>1</sub> a <sub>0</sub>
--	---------	---	---

 76H

Accumulator Bit	Hex Op Code
0	12
1	32
2	52
3	72
4	92
5	B2
6	D2
7	F2

Description: This is a 2-cycle instruction. Control passes to the specified address if accumulator bit 'b' is set to one.

Operation:  $(PC_{0-7}) \leftarrow \text{addr}$  If Bb = 1  
 $(PC) = (PC) + 2$  If Bb = 0

Example: JB4IS1: JB4 NEXT ;JUMP TO 'NEXT' ROUTINE  
 ;IF ACC BIT 4 = 1

**JC address Jump If Carry Is Set**

Encoding: 

1 1 1 1	0 1 1 0	a <sub>7</sub> a <sub>6</sub> a <sub>5</sub> a <sub>4</sub>	a <sub>3</sub> a <sub>2</sub> a <sub>1</sub> a <sub>0</sub>
---------	---------	---	---

 F6H

Description: This is a 2-cycle instruction. Control passes to the specified address if the carry bit is set to one.

Operation:  $(PC_{0-7}) \leftarrow \text{addr}$  If C = 1  
 $(PC) = (PC) + 2$  If C = 0

Example: JC1: JC OVFLOW ;JUMP TO 'OVFLOW' ROUTINE  
 ;IF C = 1

**JF0 address Jump If Flag 0 Is Set**

Encoding: 

1 0 1 1	0 1 1 0	a <sub>7</sub> a <sub>6</sub> a <sub>5</sub> a <sub>4</sub>	a <sub>3</sub> a <sub>2</sub> a <sub>1</sub> a <sub>0</sub>
---------	---------	---	---

 B6H

Description: This is a 2-cycle instruction. Control passes to the specified address if flag 0 is set to one.

Operation:  $(PC_{0-7}) \leftarrow \text{addr}$  If F0 = 1  
 $(PC) = (PC) + 2$  If F0 = 0

Example: JF0IS1: JF0 TOTAL ;JUMP TO 'TOTAL' ROUTINE IF F0 = 1

**JF1 address Jump If Flag 1 Is Set**

Encoding: 

0 1 1 1	0 1 1 0	a <sub>7</sub> a <sub>6</sub> a <sub>5</sub> a <sub>4</sub>	a <sub>3</sub> a <sub>2</sub> a <sub>1</sub> a <sub>0</sub>
---------	---------	---	---

 76H

Description: This is a 2-cycle instruction. Control passes to the specified address if flag 1 is set to one.

Operation:  $(PC_{0-7}) \leftarrow \text{addr}$  If F1 = 1  
 $(PC) = (PC) + 2$  If F1 = 0

Example: JF1IS1: JF1 FILBUF ;JUMP TO 'FILBUF'  
 ;ROUTINE IF F1 = 1

**JMP address Direct Jump within 2K Block**

Encoding: 

a <sub>10</sub> a <sub>9</sub> a <sub>8</sub> 0	0 1 0 0	a <sub>7</sub> a <sub>6</sub> a <sub>5</sub> a <sub>4</sub>	a <sub>3</sub> a <sub>2</sub> a <sub>1</sub> a <sub>0</sub>
---	---------	---	---

Page	Hex Op Code
0	04
1	24
2	44
3	64
4	84
5	A4
6	C4
7	E4

Description: This is a 2-cycle instruction. Bits 0-10 of the program counter are replaced with the directly-specified address. The setting of PC bit 11 is determined by the most recent SELECT MB instruction.

Operation:  $(PC_{8-10}) \leftarrow \text{addr } 8-10$   
 $(PC_{0-7}) \leftarrow \text{addr } 0-7$   
 $(PC_{11}) \leftarrow \text{DBF}$

Example: JMP SUBTOT ;JUMP TO SUBROUTINE 'SUBTOT'  
 JMP \$-6 ;JUMP TO INSTRUCTION SIX  
 ;LOCATIONS BEFORE CURRENT  
 ;LOCATION  
 JMP 2FH ;JUMP TO ADDRESS '2F' HEX

**JMPP @A Indirect Jump within Page**

Encoding: 

1 0 1 1	0 0 1 1
---------	---------

 B3H

Description: This is a 2-cycle instruction. The contents of the program memory location pointed to by the accumulator are substituted for the 'page' portion of the program counter (PC bits 0-7).

Operation:  $(PC_{0-7}) \leftarrow ((A))$

Example: Assume accumulator contains 0FH.

JMPPAG: JMPP @A ;JUMP TO ADDRESS STORED IN  
;LOCATION 15 IN CURRENT PAGE

#### JNC address Jump If Carry Is Not Set

Encoding: 

1	1	1	0
---	---	---	---

0	1	1	0
---	---	---	---

a <sub>7</sub>	a <sub>6</sub>	a <sub>5</sub>	a <sub>4</sub>
----------------	----------------	----------------	----------------

a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>
----------------	----------------	----------------	----------------

 86H

Description: This is a 2-cycle instruction. Control passes to the specified address if the carry bit is not set, that is, equals zero.

Operation:  $(PC_{0-7}) \leftarrow \text{addr}$  If C = 0  
 $(PC) = (PC) + 2$  If C = 1

Example: JCO: JNC NOVFO ;JUMP TO 'NOVFO' ROUTINE  
;IF C = 0

#### JNI address Jump If Interrupt Input Is Low

Encoding: 

1	0	0	0
---	---	---	---

0	1	1	0
---	---	---	---

a <sub>7</sub>	a <sub>6</sub>	a <sub>5</sub>	a <sub>4</sub>
----------------	----------------	----------------	----------------

a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>
----------------	----------------	----------------	----------------

 86H

Description: This is a 2-cycle instruction. Control passes to the specified address if the interrupt input signal is low (= 0), that is, an external interrupt has been signaled. (This signal initiates an interrupt service sequence if the external interrupt is enabled.)

Operation:  $(PC_{0-7}) \leftarrow \text{addr}$  If I = 0  
 $(PC) = (PC) + 2$  If I = 1

Example: LOC 3: JNI EXTINT ;JUMP TO 'EXTINT' ROUTINE  
;IF I = 0

#### JNT0 address Jump If Test 0 Is Low

Encoding: 

0	0	1	0
---	---	---	---

0	1	1	0
---	---	---	---

a <sub>7</sub>	a <sub>6</sub>	a <sub>5</sub>	a <sub>4</sub>
----------------	----------------	----------------	----------------

a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>
----------------	----------------	----------------	----------------

 26H

Description: This is a 2-cycle instruction. Control passes to the specified address, if the test 0 signal is low.

Operation:  $(PC_{0-7}) \leftarrow \text{addr}$  If T0 = 0  
 $(PC) = (PC) + 2$  If T0 = 1

Example: JT0LOW: JNT0 60 ;JUMP TO LOCATION 60 DEC  
;IF T0 = 0

#### JNT1 address Jump If Test 1 Is Low

Encoding: 

0	1	0	0
---	---	---	---

0	1	1	0
---	---	---	---

a <sub>7</sub>	a <sub>6</sub>	a <sub>5</sub>	a <sub>4</sub>
----------------	----------------	----------------	----------------

a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>
----------------	----------------	----------------	----------------

 46H

Description: This is a 2-cycle instruction. Control passes to the specified address, if the test 1 signal is low.

Operation:  $(PC_{0-7}) \leftarrow \text{addr}$  If T1 = 0  
 $(PC) = (PC) + 2$  If T1 = 1

#### JNZ Address Jump If Accumulator Is Not Zero

Encoding: 

1	0	0	1
---	---	---	---

0	1	1	0
---	---	---	---

a <sub>7</sub>	a <sub>6</sub>	a <sub>5</sub>	a <sub>4</sub>
----------------	----------------	----------------	----------------

a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>
----------------	----------------	----------------	----------------

 96H

Description: This is a 2-cycle instruction. Control passes to the specified address if the accumulator contents are nonzero at the time this instruction is executed.

Operation:  $(PC_{0-7}) \leftarrow \text{addr}$  If A ≠ 0  
 $(PC) = (PC) + 2$  If A = 0

Example: JACCN0: JNZ 0ABH ;JUMP TO LOCATION 'AB' HEX  
;IF ACC VALUE IS NONZERO

#### JTF address Jump If Timer Flag Is Set

Encoding: 

0	0	0	1
---	---	---	---

0	1	1	0
---	---	---	---

a <sub>7</sub>	a <sub>6</sub>	a <sub>5</sub>	a <sub>4</sub>
----------------	----------------	----------------	----------------

a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>
----------------	----------------	----------------	----------------

 16H

Description: This is a 2-cycle instruction. Control passes to the specified address if the timer flag is set to one, that is, the timer/counter register has overflowed. Testing the timer flag resets it to zero. (This overflow initiates an interrupt service sequence if the timer-overflow interrupt is enabled.)

Operation:  $(PC_{0-7}) \leftarrow \text{addr}$  If TF = 1  
 $(PC) = (PC) + 2$  If TF = 0

Example: JTF1: JTF TIMER ;JUMP TO 'TIMER' ROUTINE  
;IF TF = 1

#### JT0 address Jump If Test 0 Is High

Encoding: 

0	0	1	1
---	---	---	---

0	1	1	0
---	---	---	---

a <sub>7</sub>	a <sub>6</sub>	a <sub>5</sub>	a <sub>4</sub>
----------------	----------------	----------------	----------------

a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>
----------------	----------------	----------------	----------------

 36H

Description: This is a 2-cycle instruction. Control passes to the specified address if the test 0 signal is high (= 1).

Operation:  $(PC_{0-7}) \leftarrow \text{addr}$  If T0 = 1  
 $(PC) = (PC) + 2$  If T0 = 0

Example: JT0HI: JT0 53 ;JUMP TO LOCATION 53 DEC  
;IF T0 = 1

MCS<sup>®</sup>-48 INSTRUCTION SET

JT1 address Jump If Test 1 Is High

Encoding: 

0	1	0	1
---	---	---	---

0	1	1	0
---	---	---	---

a <sub>7</sub>	a <sub>6</sub>	a <sub>5</sub>	a <sub>4</sub>
----------------	----------------	----------------	----------------

a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>
----------------	----------------	----------------	----------------

 56H

Description: This is a 2-cycle instruction. Control passes to the specified address if the test 1 signal is high (= 1).

Operation: (PC<sub>0-7</sub>) ← addr                    If T1 = 1  
              (PC) = (PC) + 2                If T1 = 0

Example: JT1HI: JT1 COUNT                    ;JUMP TO 'COUNT' ROUTINE  
  ;IF T1 = 1

JZ address Jump If Accumulator Is Zero

Encoding: 

1	1	0	0
---	---	---	---

0	1	1	0
---	---	---	---

a <sub>7</sub>	a <sub>6</sub>	a <sub>5</sub>	a <sub>4</sub>
----------------	----------------	----------------	----------------

a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>
----------------	----------------	----------------	----------------

 C6H

Description: This is a 2-cycle instruction. Control passes to the specified address if the accumulator contains all zeros at the time this instruction is executed.

Operation: (PC<sub>0-7</sub>) ← addr                    If A = 0  
              (PC) = (PC) + 2                If A ≠ 1

Example: JACCO: JZ 0A3H                    ;JUMP TO LOCATION 'A3' HEX  
  ;IF ACC VALUE IS ZERO

MOV A,#data Move Immediate Data to Accumulator

Encoding: 

0	0	1	0
---	---	---	---

0	0	1	1
---	---	---	---

a <sub>7</sub>	a <sub>6</sub>	a <sub>5</sub>	a <sub>4</sub>
----------------	----------------	----------------	----------------

a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>
----------------	----------------	----------------	----------------

 23H

Description: This is a 2-cycle instruction. The 8-bit value specified by 'data' is loaded in the accumulator.

Operation: (A) ← data

Example: MOV A,#0A3H                    ;MOVE 'A3' HEX TO ACC

MOV A,PSW Move PSW Contents to Accumulator

Encoding: 

1	1	0	0
---	---	---	---

0	1	1	1
---	---	---	---

 C7H

Description: The contents of the program status word are moved to the accumulator.

Operation: (A) ← (PSW)

Example: Jump to 'RB1SET' routine if PSW bank switch, bit 4, is set.  
          BSCHK: MOV A,PSW                    ;MOVE PSW CONTENTS TO ACC  
  JB4 RB1SET                    ;JUMP TO 'RB1SET' IF ACC BIT 4 = 1

MCS<sup>®</sup>-48 INSTRUCTION SET

MOV A,R<sub>r</sub> Move Register Contents to Accumulator

Encoding: 

1	1	1	1
---	---	---	---

1	r	r	r
---	---	---	---

 F8H-FFH

Description: 8-bits of data are removed from working register 'r' into the accumulator.

Operation: (A) ← (R<sub>r</sub>)                    r = 0-7

Example: MAR: MOV A,R3                    ;MOVE CONTENTS OF REG 3 TO ACC

MOV A,@R<sub>i</sub> Move Data Memory Contents to Accumulator

Encoding 

1	1	1	1
---	---	---	---

0	0	0	i
---	---	---	---

 F0H-F1H

Description: The contents of the resident data memory location addressed by bits 0-5\*\* of register 'i' are moved to the accumulator. Register 'i' contents are unaffected.

Operation: (A) ← ((R<sub>i</sub>))                    i = 0-1

Example: Assume R1 contains 00110110.  
          MADM: MOV A,@R1                    ;MOVE CONTENTS OF DATA MEM  
  ;LOCATION 54 TO ACC

MOV A,T Move Timer/Counter Contents to Accumulator

Encoding: 

0	1	0	0
---	---	---	---

0	0	1	0
---	---	---	---

 42H

Description: The contents of the timer/event-counter register are moved to the accumulator.

Operation: (A) ← (T)

Example: Jump to "EXIT" routine when timer reaches '64', that is, when bit 6 set—  
          assuming initialization 64,  
          TIMCHK: MOV A,T                    ;MOVE TIMER CONTENTS TO ACC  
  JB6 EXIT                    ;JUMP TO 'EXIT' IF ACC BIT 6 = 1

MOV PSW,A Move Accumulator Contents to PSW

Encoding: 

1	1	0	1
---	---	---	---

0	1	1	1
---	---	---	---

 D7H

Description: The contents of the accumulator are moved into the program status word. All condition bits and the stack pointer are affected by this move.

Operation: (PSW) ← (A)

Example: Move up stack pointer by two memory locations, that is, increment the  
          pointer by one.  
          INCPTR: MOV A,PSW                    ;MOVE PSW CONTENTS TO ACC  
  INC A                    ;INCREMENT ACC BY ONE  
  MOV PSW,A                    ;MOVE ACC CONTENTS TO PSW

**MOV R<sub>r</sub>,A** Move Accumulator Contents to RegisterEncoding: 

1	0	1	0	1	r	r	r
---	---	---	---	---	---	---	---

 A8H-AFH

Description: The contents of the accumulator are moved to register 'r'.

Operation: (Rr) ← (A) r = 0-7

Example: MRA: MOV R0,A ;MOVE CONTENTS OF ACC TO REG 0

**MOV R<sub>r</sub>,#data** Move Immediate Data to RegisterEncoding: 

1	0	1	1	1	r <sub>2</sub>	r <sub>1</sub>	r <sub>0</sub>
---	---	---	---	---	----------------	----------------	----------------

d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>	d <sub>4</sub>	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

 B8H-BFH

Description: This is a 2-cycle instruction. The 8-bit value specified by 'data' is moved to register 'r'.

Operation: (Rr) ← data r = 0-7

Examples: MIR4: MOV R4,#HEXTEN ;THE VALUE OF THE SYMBOL  
 ;'HEXTEN' IS MOVED INTO REG 4  
 MIR 5: MOV R5,#PI\*(R\*R) ;THE VALUE OF THE EXPRESSION  
 ;'PI\*(R\*R)' IS MOVED INTO REG 5.  
 MIR 6: MOV R6, #0ADH ;'AD' HEX IS MOVED INTO REG 6

**MOV @R<sub>i</sub>,A** Move Accumulator Contents to Data MemoryEncoding: 

1	0	1	0	0	0	0	i
---	---	---	---	---	---	---	---

 A0H-A1H

Description: The contents of the accumulator are moved to the resident data memory location whose address is specified by bits 0-5\*\* of register 'i'. Register 'i' contents are unaffected.

Operation: ((Ri)) ← (A) i = 0-1

Example: Assume R0 contains 00000111.  
 MDMA: MOV @R0,A ;MOVE CONTENTS OF ACC TO  
 ;LOCATION 7 (REG 7)

**MOV @R<sub>i</sub>,#data** Move Immediate Data to Data memoryEncoding: 

1	0	1	1	0	0	0	i
---	---	---	---	---	---	---	---

d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>	d <sub>4</sub>	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

 B0H-B1H

Description: This is a 2-cycle instruction. The 8-bit value specified by 'data' is moved to the resident data memory location addressed by register 'i', bits 0-5\*\*.

Operation: ((Ri)) ← data i = 0-1

Examples: Move the hexadecimal value AC3F to locations 62-63.  
 MIDM: MOV R0,#62 ;MOVE '62' DEC TO ADDR REG 0  
 MOV @R0,#0ACH ;MOVE 'AC' HEX TO LOCATION 62  
 INC R0 ;INCREMENT REG 0 to '63'  
 MOV @R0,#3FH ;MOVE '3F' HEX TO LOCATION 63

\*\* 0-5 in 8048AH/8748H  
 0-6 in 8049AH/8749H

**MOV T,A** Move Accumulator Contents to Timer/CounterEncoding: 

0	1	1	0	0	0	1	0
---	---	---	---	---	---	---	---

 62H

Description: The contents of the accumulator are moved to the timer/event-counter register.

Operation: (T) ← (A)

Example: Initialize and start event counter.  
 INITEC: CLR A ;CLEAR ACC TO ZEROS  
 MOV T,A ;MOVE ZEROS TO EVENT COUNTER  
 START CNT ;START COUNTER

**MOVD A,Pp** Move Port 4-7 Data to AccumulatorEncoding: 

0	0	0	0	1	1	p	p
---	---	---	---	---	---	---	---

 0CH-0FH

Description: This is a 2-cycle instruction. Data on 8243 port 'p' is moved (read) to accumulator bits 0-3. Accumulator bits 4-7 are zeroed.

Operation: (0-3) ← (Pp) p = 4-7  
(4-7) ← 0

Note: Bits 0-7 of the opcode are used to represent ports 4-7. If you are coding in binary rather than assembly language, the mapping is as follows:

Bits 1 0	Port
0 0	4
0 1	5
1 0	6
1 1	7

Example: INPPT5: MOVD A,P5 ;MOVE PORT 5 DATA TO ACC  
 ;BITS 0-3, ZERO ACC BITS 4-7

**MOVD Pp,A** Move Accumulator Data to Port 4-7Encoding: 

0	0	1	1	1	p	p
---	---	---	---	---	---	---

 3CH-3FH

Description: This is a 2-cycle instruction. Data in accumulator bits 0-3 is moved (written) to 8243 port 'p'. Accumulator bits 4-7 are unaffected. (See NOTE above regarding port mapping.)

Operation: (Pp) ← (A<sub>0-3</sub>) P = 4-7

Example: Move data in accumulator to ports 4 and 5.  
 OUTP45: MOVD P4,A ;MOVE ACC BITS 0-3 TO PORT 4  
 SWAP A ;EXCHANGE ACC BITS 0-3 and 4-7  
 MOVD P5,A ;MOVE ACC BITS 0-3 TO PORT 5

MCS<sup>®</sup>-48 INSTRUCTION SET**MOVP A,@A Move Current Page Data to Accumulator**Encoding: 

1	0	1	0	0	0	1	1
---	---	---	---	---	---	---	---

 A3H

Description: The contents of the program memory location addressed by the accumulator are moved to the accumulator. Only bits 0-7 of the program counter are affected, limiting the program memory reference to the current page. The program counter is restored *following* this operation.

Operation:  $(PC_{0-7}) \leftarrow (A)$   
 $(A) \leftarrow ((PC))$

Note: This is a 1-byte, 2-cycle instruction. If it appears in location 255 of a program memory page, @A addresses a location in the *following* page.

Example: MOV128: MOV A,#128 ;MOVE '128' DEC TO ACC  
 MOVP A,@A ;CONTENTS OF 129th LOCATION IN  
 ;CURRENT PAGE ARE MOVED TO ACC

**MOVP3 A,@A Move Page 3 Data to Accumulator**Encoding: 

1	1	1	0	0	0	1	1
---	---	---	---	---	---	---	---

 E3H

Description: This is a 2-cycle instruction. The contents of the program memory location (within page 3) addressed by the accumulator are moved to the accumulator. The program counter is restored following this operation.

Operation:  $(PC_{0-7}) \leftarrow (A)$   
 $(PC_{8-11}) \leftarrow 0011$   
 $(A) \leftarrow ((PC))$

Example: Look up ASCII equivalent of hexadecimal code in table contained at the beginning of page 3. Note that ASCII characters are designated by a 7-bit code; the eighth bit is always reset.

TABSCH: MOV A,#0B8H ;MOVE 'B8' HEX TO ACC (10111000)  
 ANL A,#7FH ;LOGICAL AND ACC TO MASK BIT  
 ;7 (00111000)  
 MOVP3 A,@A ;MOVE CONTENTS OF LOCATION '38'  
 ;HEX IN PAGE 3 TO ACC (ASCII '8')

Access contents of location in page 3 labelled TAB1.

Assume current program location is not in page 3.

TABSCH: MOV A,#LOW TAB 1 ;ISOLATE BITS 0-7 OF LABEL  
 ;ADDRESS VALUE  
 MOVP3 A,@A ;MOVE CONTENTS OF PAGE 3  
 ;LOCATION LABELED 'TAB1' TO ACC

MCS<sup>®</sup>-48 INSTRUCTION SET**MOVX A,@R<sub>i</sub> Move External-Data-Memory Contents to Accumulator**Encoding: 

1	0	0	0	0	0	0	i
---	---	---	---	---	---	---	---

 80H-81H

Description: This is a 2-cycle instruction. The contents of the external data memory location addressed by register 'i' are moved to the accumulator. Register 'i' contents are unaffected. A read pulse is generated.

Operation:  $(A) \leftarrow ((R_i))$   $i = 0-1$

Example: Assume R1 contains 01110110.  
 MAXDM: MOVX A,@R1 ;MOVE CONTENTS OF LOCATION  
 ;118 TO ACC

**MOVX @R<sub>i</sub>,A Move Accumulator Contents to External Data Memory**Encoding: 

1	0	0	1	0	0	0	i
---	---	---	---	---	---	---	---

 90H-91H

Description: This is a 2-cycle instruction. The contents of the accumulator are moved to the external data memory location addressed by register 'i'. Register 'i' contents are unaffected. A write pulse is generated.

Operation:  $((R_i)) \leftarrow A$   $i = 0-1$

Example: Assume R0 contains 11000111.  
 MXDMA: MOVX @R0,A ;MOVE CONTENTS OF ACC TO  
 ;LOCATION 199 IN EXPANDED  
 ;DATA MEMORY

**NOP The NOP Instruction**Encoding: 

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

 00H

Description: No operation is performed. Execution continues with the following instruction.

**ORL A,R<sub>r</sub> Logical OR Accumulator With Register Mask**Encoding: 

0	1	0	0	1	r	r	r
---	---	---	---	---	---	---	---

 48H-4FH

Description: Data in the accumulator is logically ORed with the mask contained in working register 'r'.

Operation:  $(A) \leftarrow (A) \text{ OR } (R_r)$   $r = 0-7$

Example: ORREG: ORL A,R4 ;'OR' ACC CONTENTS WITH  
 ;MASK IN REG 4

MCS<sup>®</sup>-48 INSTRUCTION SET

**ORL A,@R<sub>i</sub> Logical OR Accumulator With Memory Mask**

Encoding: 

0	1	0	0
---	---	---	---

0	0	0	i
---	---	---	---

 40H-41H

Description: Data in the accumulator is logically ORed with the mask contained in the resident data memory location referenced by register "i", bits 0-5\*\*.

Operation: (A) ← (A) OR ((R<sub>i</sub>))                      i = 0-1

Example: ORDM: MOV R0,#3FH                      ;MOVE '3F' HEX TO REG 0  
           ORL A,@R0                              ;'OR' AC CONTENTS WITH MASK  
   ;IN LOCATION 63

**ORL A,#data Logical OR Accumulator With Immediate Mask**

Encoding: 

0	1	0	0
---	---	---	---

0	0	1	1
---	---	---	---

d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>	d <sub>4</sub>
----------------	----------------	----------------	----------------

d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>
----------------	----------------	----------------	----------------

 43H

Description: This is a 2-cycle instruction. Data in the accumulator is logically ORed with an immediately-specified mask.

Operation: (A) ← (A) OR data

Example: ORID: ORL A,#'X'                      ;'OR' ACC CONTENTS WITH MASK  
   ;01011000 (ASCII VALUE OF 'X')

**ORL BUS,#data\* Logical OR BUS With Immediate Mask**

Encoding: 

1	0	0	0
---	---	---	---

1	0	0	0
---	---	---	---

d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>	d <sub>4</sub>
----------------	----------------	----------------	----------------

d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>
----------------	----------------	----------------	----------------

 82H

Description: This is a 2-cycle instruction. Data on the BUS port is logically ORed with an immediately-specified mask. This instruction assumes prior specification on an 'OUTL BUS,A' instruction.

Operation: (BUS) ← (BUS) OR data

Example: ORBUS: ORL BUS,#HEXMSK            ;'OR' BUS CONTENTS WITH MASK  
   ;EQUAL VALUE OF SYMBOL 'HEXMSK'

**ORL Pp,#data Logical OR Port 1 or 2 With Immediate Mask**

Encoding: 

1	0	0	0
---	---	---	---

1	0	p	p
---	---	---	---

d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>	d <sub>4</sub>
----------------	----------------	----------------	----------------

d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>
----------------	----------------	----------------	----------------

 89H-8AH

Description: This is a 2-cycle instruction. Data on port 'p' is logically ORed with an immediately-specified mask.

Operation: (Pp) ← (Pp) OR data                      p = 1-2

Example: ORP1: ORL P1, #0FFH                ;'OR' PORT 1 CONTENTS WITH MASK  
   ;'FF' HEX (SET PORT 1 TO ALL ONES)

\* For use with internal program memory ONLY.

\*\* 0-5 in 8048AH/8748H  
 0-6 in 8049AH/8749H  
 0-7 in 8050AH

MCS<sup>®</sup>-48 INSTRUCTION SET

**ORLD Pp,A Logical OR Port 4-7 With Accumulator Mask**

Encoding: 

1	0	0	0
---	---	---	---

1	1	p	p
---	---	---	---

 8CH-8FH

Description: This is a 2-cycle instruction. Data on port 'p' is logically ORed with the digit mask contained in accumulator bits 0-3.

Operation: (Pp) ← (Pp) OR (A<sub>0-3</sub>)                      p = 4-7

Example: ORP7: ORLD P7,A                      ;'OR' PORT 7 CONTENTS WITH ACC  
   ;BITS 0-3

**OUTL BUS,A\* Output Accumulator Data to BUS**

Encoding: 

0	0	0	0
---	---	---	---

0	0	1	0
---	---	---	---

 02H

Description: This is a 2-cycle instruction. Data residing in the accumulator is transferred (written) to the BUS port and latched. The latched data remains valid until altered by another OUTL instruction. Any other instruction requiring use of the BUS port (except INS) destroys the contents of the BUS latch. This includes expanded memory operations (such as the MOVX instruction). Logical operations on BUS data (AND, OR) assume the OUTL BUS,A instruction has been issued previously.

Operation: (BUS) ← (A)

Example: OUTLBP: OUTL BUS, A                ;OUTPUT ACC CONTENTS TO BUS

**OUTL Pp,A Output Accumulator Data to Port 1 or 2**

Encoding: 

0	0	1	1
---	---	---	---

1	0	p	p
---	---	---	---

 39H-3AH

Description: This is a 2-cycle instruction. Data residing in the accumulator is transferred (written) to port 'p' and latched.

Operation: (Pp) ← (A)                                      p = 1-2

Example: OUTLP: MOV A,R7                      ;MOVE REG 7 CONTENTS TO ACC  
   OUTL P2,A                                      ;OUTPUT ACC CONTENTS TO PORT 2  
   MOV A, R6                                      ;MOV REG 6 CONTENTS TO ACC  
   OUTL P1,A                                      ;OUTPUT ACC CONTENTS TO PORT 1

\* For use with internal program memory ONLY.



**RRC A Rotate Right through Carry**Encoding: 

0	1	1	0
0	1	1	1

 E7-**Description:** The contents of the accumulator are rotated right one bit. Bit 0 replaces the carry bit; the carry bit is rotated into the bit 7 position.**Operation:** (An) ← (An + 1)            n = 0-6  
(A7) ← (C)  
(C) ← (A<sub>0</sub>)**Example:** Assume carry is not set and accumulator contains 10110001.  
RRTC: RRC A                               ;CARRY IS SET AND ACC  
  ;CONTAINS 01011000**SEL MB0 Select Memory Bank 0**Encoding: 

1	1	1	0
0	1	0	1

 E5H**Description:** PC bit 11 is set to zero on next JMP or CALL instruction. All references to program memory addresses fall within the range 0-2047.**Operation:** (DBF) ← 0**Example:** Assume program counter contains 834 Hex.  
SEL MB0                               ;SELECT MEMORY BANK 0  
JMP S+20                               ;JUMP TO LOCATION 58 HEX**SEL MB1 Select Memory Bank 1**Encoding: 

1	1	1	1
0	1	0	1

 F5H**Description:** PC bit 11 is set to one on next JMP or CALL instruction. All references to program memory addresses fall within the range 2048-4095.**Operation:** (DBF) ← 1**SEL RB0 Select Register Bank 0**Encoding: 

1	1	0	0
0	1	0	1

 C5H**Description:** PSW bit 4 is set to zero. References to working registers 0-7 address data memory locations 0-7. This is the recommended setting for normal program execution.**Operation:** (BS) ← 0**SEL RB1 Select Register Bank 1**Encoding: 

1	1	0	1
0	1	0	1

 D5H**Description:** PSW bit 4 is set to one. References to working registers 0-7 address data memory locations 24-31. This is the recommended setting for interrupt service routines, since locations 0-7 are left intact. The setting of PSW bit 4 in effect at the time of an interrupt is restored by the RETR instruction when the interrupt service routine is completed.**Operation:** (BS) ← 1**Example:** Assume an external interrupt has occurred, control has passed to program memory location 3, and PSW bit 4 was zero before the interrupt.

<b>Operation:</b> LOC3: JN1 INIT  INIT: MOV R7,A  SEL RB1 MOV R7,#0FAH  SEL RB0 MOV A,R7 RETR	;JUMP TO ROUTINE 'INIT' IF ;INTERRUPT INPUT IS ZERO ;MOVE ACC CONTENTS TO ;LOCATION 7 ;SELECT REG BANK 1 ;MOVE 'FA' HEX TO LOCATION 31  ;SELECT REG BANK 0 ;RESTORE ACC FROM LOCATION 7 ;RETURN — RESTORE PC AND PSW
--	---

**STOP TCNT Stop Timer/Event-Counter**Encoding: 

0	1	1	0
0	1	0	1

 65H**Description:** This instruction is used to stop both time accumulation and event counting.

**Example:** Disable interrupt, but jump to interrupt routine after eight overflows and stop timer. Count overflows in register 7.

```

START: DIS TCNTI      ;DISABLE TIMER INTERRUPT
      CLR A          ;CLEAR ACC TO ZEROS
      MOV T,A        ;MOVE ZEROS TO TIMER
      MOV R7,A       ;MOVE ZERCS TO REG 7
      STRT T         ;START TIMER
MAIN:  JTF COUNT     ;JUMP TO ROUTINE 'COUNT'
      ;IF TF = 1 AND CLEAR TIMER FLAG
      JMP MAIN      ;CLOSE LOOP
COUNT: INC R7       ;INCREMENT REG 7
      MOV A,R7      ;MOVE REG 7 CONTENTS TO ACC
      JB3 INT       ;JUMP TO ROUTINE 'INT' IF ACC
      ;BIT 3 IS SET (REG 7 = 8)
      JMP MAIN      ;OTHERWISE RETURN TO ROUTINE
      ;MAIN

INT:  STOP TCNT     ;STOP TIMER
      JMP 7H        ;JUMP TO LOCATION 7 (TIMER)
      ;INTERRUPT ROUTINE

```

#### STRT CNT Start Event Counter

Encoding: 0 1 0 0 0 1 0 1 45H

**Description:** The test 1 (T1) pin is enabled as the event-counter input and the counter is started. The event-counter register is incremented with each high-to-low transition on the T1 pin.

**Example:** Initialize and start event counter. Assume overflow is desired with first T1 input.

```

STARTC: EN TCNTI    ;ENABLE COUNTER INTERRUPT
      MOV A,#0FFH   ;MOVE 'FF'HEX (ONES) TO ACC
      MOV T,A       ;MOVES ONES TO COUNTER
      STRT CNT      ;ENABLE T1 AS COUNTER
      ;INPUT AND START

```

#### STRT T Start Timer

Encoding: 0 1 0 1 0 1 0 1 55H

**Description:** Timer accumulation is initiated in the timer register. The register is incremented every 32 instruction cycles. The prescaler which counts the 32 cycles is cleared but the timer register is not.

**Example:** Initialize and start timer.

```

STARTT: CLR A      ;CLEAR ACC TO ZEROS
      MOV T,A      ;MOVE ZEROS TO TIMER
      EN TCNTI     ;ENABLE TIMER INTERRUPT
      STRT T       ;START TIMER

```

#### SWAP A Swap Nibbles within Accumulator

Encoding: 0 1 0 0 0 1 1 1 47H

**Description:** Bits 0-3 of the accumulator are swapped with bits 4-7 of the accumulator.

**Operation:**  $(A_{4-7}) \rightleftharpoons (A_{0-3})$

**Example:** Pack bits 0-3 of locations 50-51 into location 50.

```

PCKDIG: MOV R0, #50 ;MOVE '50' DEC TO REG 0
      MOV R1, #51   ;MOVE '51' DEC TO REG 1
      XCHD A,@R0    ;EXCHANGE BITS 0-3 OF ACC
      ;AND LOCATION 50
      SWAP A        ;SWAP BITS 0-3 AND 4-7 OF ACC
      XCHD A,@R1    ;EXCHANGE BITS 0-3 OF ACC AND
      ;LOCATION 51
      MOV @R0,A     ;MOVE CONTENTS OF ACC TO
      ;LOCATION 50

```

#### XCH A,R<sub>r</sub> Exchange Accumulator-Register Contents

Encoding: 0 0 1 0 1 r r r 28H-2FH

**Description:** The contents of the accumulator and the contents of working register 'r' are exchanged.

**Operation:**  $(A) \rightleftharpoons (R_r)$  r = 0-7

**Example:** Move PSW contents to Reg 7 without losing accumulator contents.

```

XCHAR7: XCH A,R7   ;EXCHANGE CONTENTS OF REG 7
      ;AND ACC
      MOV A,PSW    ;MOVE PSW CONTENTS TO ACC
      XCH A,R7     ;EXCHANGE CONTENTS OF REG 7
      ;AND ACC AGAIN

```

**XCH A,@R<sub>i</sub>** Exchange Accumulator and Data Memory ContentsEncoding: 

0	0	1	0	0	0	i
---	---	---	---	---	---	---

 20H-21H

Description: The contents of the accumulator and the contents of the resident data memory location addressed by bits 0-5\*\* of register 'i' are exchanged. Register 'i' contents are unaffected.

Operation:  $(A) \rightleftharpoons ((Ri))$  i = 0-1

Example: Decrement contents of location 52.

```

DEC52: MOV R0,#52      :MOVE '52' DEC TO ADDRESS REG 0
        XCH A,@R0      :EXCHANGE CONTENTS OF ACC
                          :AND LOCATION 52
        DEC A           :DECREMENT ACC CONTENTS
        XCH A,@R0      :EXCHANGE CONTENTS OF ACC
                          :AND LOCATION 52 AGAIN

```

**XCHD A,@R<sub>i</sub>** Exchange Accumulator and Data Memory 4-Bit DataEncoding: 

0	0	1	1	0	0	0	i
---	---	---	---	---	---	---	---

 30H-31H

Description: This instruction exchanges bits 0-3 of the accumulator with bits 0-3 of the data memory location addressed by bits 0-5\*\* of register 'i'. Bits 4-7 of the accumulator, bits 4-7 of the data memory location, and the contents of register 'i' are unaffected.

Operation:  $(A_{0-3}) \rightleftharpoons ((Ri_{0-3}))$  i = 0-1

Example: Assume program counter contents have been stacked in locations 22-23.

```

XCHNIB: MOV R0,#23     :MOVE '23' DEC TO REG 0
        CLR A          :CLEAR ACC TO ZEROS
        XCHD A,@R0     :EXCHANGE BITS 0-3 OF ACC AND
                          :LOCATION 23 (BITS 8-11 OF PC ARE
                          :ZEROED, ADDRESS REFERS
                          :TO PAGE 0)

```

**XRL A,R<sub>r</sub>** Logical XOR Accumulator With Register MaskEncoding: 

1	1	0	1	1	r	r	r
---	---	---	---	---	---	---	---

 D8H-DFH

Description: Data in the accumulator is EXCLUSIVE ORed with the mask contained in working register 'r'.

Operation:  $(A) \oplus (A) \text{ XOR } (Rr)$  r = 0-7

Example: XORREG: XRL A,R5 ;'XOR' ACC CONTENTS WITH  
;MASK IN REG 5

\*\* 0-5 in 8048AH/8748H  
0-6 in 8049AH/8749H  
0-7 in 8050AH

**XRL A,@R<sub>i</sub>** Logical XOR Accumulator With Memory MaskEncoding: 

1	1	0	1	0	0	0	i
---	---	---	---	---	---	---	---

 D0H-D1H

Description: Data in the accumulator is EXCLUSIVE ORed with the mask contained in the data memory location addressed by register 'i', bits 0-5.\*\*

Operation:  $(A) \oplus (A) \text{ XOR } ((Ri))$  i = 0-1

Example: XORDM: MOV R1,#20H ;MOVE '20' HEX TO REG 1  
XRL A,@R1 ;'XOR' ACC CONTENTS WITH MASK  
;IN LOCATION 32

**XRL A,#data** Logical XOR Accumulator With Immediate MaskEncoding: 

1	1	0	1	0	0	1	1
---	---	---	---	---	---	---	---

d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>	d <sub>4</sub>	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

 D3H

Description: This is a 2-cycle instruction. Data in the accumulator is EXCLUSIVE ORed with an immediately-specified mask.

Operation:  $(A) \oplus (A) \text{ XOR data}$ 

Example: XORID: XOR A,#HEXTEN ;XOR CONTENTS OF ACC WITH MASK  
;EQUAL VALUE OF SYMBOL 'HEXTEN'

\*\* 0-5 in 8048AH/8748H  
0-6 in 8049AH/8749H  
0-7 in 8050AH

**ANEXO D**

**CARACTERISTICAS GENERALES DEL 8748H**

# 8748H/8035H/8749H/8039H HMOS-E SINGLE-COMPONENT 8-BIT MICROCOMPUTER

- High Performance HMOS-E
- Interval Timer/Event Counter
- Two Single Level Interrupts
- Single 5-Volt Supply
- Over 96 Instructions; 90% Single Byte
- Compatible with 8080/8085 Peripherals
- Easily Expandable Memory and I/O
- Up to 1.35  $\mu$ s Instruction Cycle; All Instructions 1 or 2 Cycles

The Intel 8749H/8039H/8748H/8035H are totally self-sufficient, 8-bit parallel computers fabricated on single silicon chips using Intel's advanced N-channel silicon gate HMOS-E process.

The family contains 27 I/O lines, an 8-bit timer/counter, on-chip RAM and on-board oscillator/clock circuits. For systems that require extra capability, the family can be expanded using MCS<sup>®</sup>-80/MCS<sup>®</sup>-85 peripherals.

These microcomputers are designed to be efficient controllers as well as arithmetic processors. They have extensive bit handling capability as well as facilities for both binary and BCD arithmetic. Efficient use of program memory results from an instruction set consisting mostly of single byte instructions and no instructions over 2 bytes in length.

Device	Internal Memory	
8039H	none	128 x 8 RAM
8035H	none	64 x 8 RAM
8749H	2K x 8 EPROM	128 x 8 RAM
8748H	1K x 8 EPROM	64 x 8 RAM

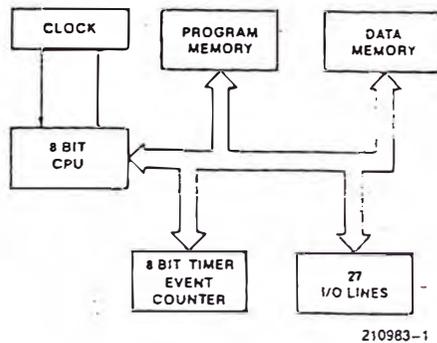


Figure 1.  
Block Diagram

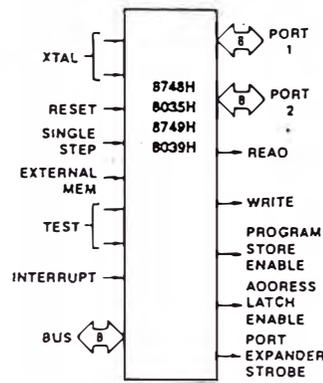


Figure 2.  
Logic Symbol

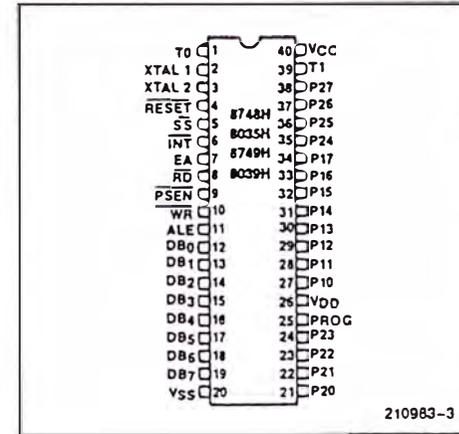


Figure 3. Pin Configuration

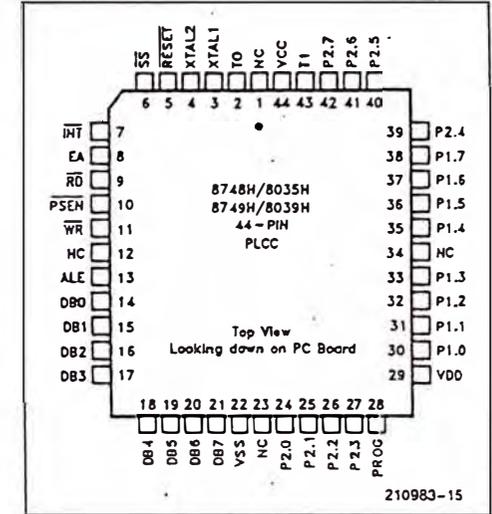


Figure 4. Pad Configuration

Table 1. Pin Description (40-Pin DIP)

Symbol	Pin No.	Function	Device
V <sub>SS</sub>	20	Circuit GND potential.	All
V <sub>DD</sub>	26	+5V during normal operation.	All
		Programming power supply (+21V).	8748H 8749H
V <sub>CC</sub>	40	Main power supply; +5V during operation and programming.	All
PROG	25	Output strobe for 8243 I/O expander.	All
		Program pulse (+18V) input pin during programming.	8748H 8749H (See Note)
P10-P17 Port 1	27-34	8-bit quasi-bidirectional port.	All
P20-P23 P24-P27 Port 2	21-24 35-38	8-bit quasi-bidirectional port. P20-P23 contain the four high order program counter bits during an external program memory fetch and serve as a 4-bit I/O expander bus for 8243.	All
DB0-DB7 BUS	12-19	True bidirectional port which can be written or read synchronously using the RD, WR strobes. The port can also be statically latched. Contains the 8 low order program counter bits during an external program memory fetch, and receives the addressed instruction under the control of PSEN. Also contains the address and data during an external RAM data store instruction, under control of ALE, RD, and WR.	All
T0	1	Input pin testable using the conditional transfer instructions JT0 and JNT0. T0 can be designated as a clock output using ENT0 CKL instruction.	All
		Used during programming.	8748H 8749H
T1	39	Input pin testable using the JT1, and JNT1 instructions. Can be designated the timer/counter input using the STRT CNT instruction.	All
INT	6	Interrupt input. Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. Also testable with conditional jump instruction. (Active	All

Table 1. Pin Description (40-Pin DIP) (Continued)

Symbol	Pin No.	Function	Device
RD	8	Output strobe activated during a BUS read. Can be used to enable data onto the bus from an external device. Used as a read strobe to external data memory. (Active low)	All
RESET	4	Input which is used to initialize the processor. (Active low) (Non TTL V <sub>IH</sub> ) Used during programming.	All 8748H 8749H
WR	10	Output strobe during a bus write. (Active low) Used as write strobe to external data memory.	All
ALE	11	Address latch enable. This signal occurs once during each cycle and is useful as a clock output. The negative edge of ALE strobes address into external data and program memory.	All
PSEN	9	Program store enable. This output occurs only during a fetch to external program memory. (Active low.)	All
SS	5	Single step input can be used in conjunction with ALE to "single step" the processor through each instruction.	All
EA	7	External access input which forces all program memory fetches to reference external memory. Useful for emulation and debug. (Active high.) Used during (18V) programming.	All 8748H 8749H
XTAL1	2	One side of crystal input for internal oscillator. Also input for external source. (Non TTL V <sub>IH</sub> .)	All
XTAL2	3	Other side of crystal input.	All

NOTE:

On the 8749H/8039H, PROG must be clamped to V<sub>CC</sub> when not programming. A diode should be used when using an 8243; otherwise, a direct connection is permissible.

Table 2. Instruction Set

Mnemonic	Description	Bytes	Cycles	Mnemonic	Description	Bytes	Cycles
<b>ACCUMULATOR</b>				<b>ACCUMULATOR (Continued)</b>			
ADD A, R	Add register to A	1	1	XRL A, @R	Exclusive or data memory to A	1	1
ADD A, @R	Add data memory to A	1	1	XR L A, #data	Exclusive or immediate to A	2	2
ADD A, #data	Add immediate to A	2	2	INC A	Increment A	1	1
ADDC A, R	Add register with carry	1	1	DEC A	Decrement A	1	1
ADDC A, @R	Add data memory with carry	1	1	CLR A	Clear A	1	1
ADDC A, #data	Add immediate with carry	2	2	CPL A	Complement A	1	1
ANL A, R	And register to A	1	1	DA A	Decimal adjust A	1	1
ANL A, @R	And data memory to A	1	1	SWAP A	Swap nibbles of A	1	1
ANL A, #data	And immediate to A	2	2	RL A	Rotate A left	1	1
ORL A, R	Or register to A	1	1	RLC A	Rotate A left through carry	1	1
ORL A, @R	Or data memory to A	1	1	RR A	Rotate A right	1	1
ORL A, #data	Or immediate to A	2	2	RRC A	Rotate A right through carry	1	1
XRL A, R	Exclusive or register to A	1	1	<b>INPUT/OUTPUT</b>			
XRL A, @R	Exclusive or data memory to A	1	1	IN A, P	Input port to A	1	2
XRL A, #data	Exclusive or immediate to A	2	2	OUTL P, A	Output A to port	1	2
XRLA, R	Exclusive or register to A	1	1	ANL P, #data	And immediate	2	2

Table 2. Instruction Set (Continued)

Mnemonic	Description	Bytes	Cycles	Mnemonic	Description	Bytes	Cycles
<b>REGISTERS</b>				<b>DATA MOVES</b>			
INC R	Increment register	1	1	MOV A, R	Move register to A	1	1
INC @R	Increment data memory	1	1	MOV A, @R	Move data memory to A	1	1
DEC R	Decrement register	1	1	MOV A, #data	Move immediate to A	2	2
<b>BRANCH</b>				MOV R, A	Move A to register	1	1
JMP addr	Jump unconditional	2	2	*MOV @R, A	Move A to data memory	1	1
JMPP @A	Jump indirect	1	2	MOV R, #data	Move immediate to register	2	2
DJNZ R, addr	Decrement register and skip	2	2	MOV @R, #data	Move immediate to data memory	2	2
JC addr	Jump on carry = 1	2	2	MOV A, PSW	Move PSW to A	1	1
JNC addr	Jump on carry = 0	2	2	MOV PSW, A	Move A to PSW	1	1
JZ addr	Jump on A zero	2	2	XCH A, R	Exchange A and register	1	1
JNZ addr	Jump on A not zero	2	2	XCH A, @R	Exchange A and data memory	1	1
JT0 addr	Jump on T0 = 1	2	2	XCHD A, @R	Exchange nibble of A and register	1	1
JNT0 addr	Jump on T0 = 0	2	2	MOVX A, @R	Move external data memory to A	1	2
JT1 addr	Jump on T1 = 1	2	2	MOVX @R, A	Move A to external data memory	1	2
JNT1 addr	Jump on T1 = 0	2	2	MOVP A, @A	Move to A from current page	1	2
JF0 addr	Jump on F0 = 1	2	2	MOVP3 A, @A	Move to A from page 3	1	2
JF1 addr	Jump on F1 = 1	2	2	<b>TIMER/COUNTER</b>			
JTF addr	Jump on timer flag	2	2	MOV A, T	Read timer/counter	1	1
JNI addr	Jump on INT = 0	2	2	MOV T, A	Load timer/counter	1	1
JBb addr	Jump on accumulator bit	2	2	STRT T	Start timer	1	1
<b>SUBROUTINE</b>				STRT CNT	Start counter	1	1
CALL addr	Jump to subroutine	2	2	STOP TCNT	Stop timer/counter	1	1
RET	Return	1	2	EN TCNTI	Enable timer/counter interrupt	1	1
RETR	Return and restore status	1	2	DIS TCNTI	Disable timer/counter interrupt	1	1
<b>FLAGS</b>				<b>CONTROL</b>			
CLR C	Clear carry	1	1	EN I	Enable external interrupt	1	1
CPL C	Complement carry	1	1	DIS I	Disable external interrupt	1	1
CLR F0	Clear flag 0	1	1	SEL RB0	Select register bank 0	1	1
CPL F0	Complement flag 0	1	1	SEL RB1	Select register bank 1	1	1
CLR F1	Clear flag 1	1	1	SEL MB0	Select memory bank 0	1	1
CPL F1	Complement flag 1	1	1	SEL MB1	Select memory bank 1	1	1
				ENT0 CLK	Enable clock output on T0	1	1

**ABSOLUTE MAXIMUM RATINGS\***

Ambient Temperature Under Bias ..... 0°C to 70°C  
 Storage Temperature ..... -65°C to +150°C  
 Voltage On Any Pin With Respect to Ground ..... -0.5V to +7V  
 Power Dissipation ..... 1.0 Watt

*\*Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

**NOTICE:** Specifications contained within the following tables are subject to change.

**D.C. CHARACTERISTICS**  $T_A = 0^\circ\text{C to } 70^\circ\text{C}; V_{CC} = V_{DD} = 5V \pm 10\%; V_{SS} = 0V$

Symbol	Parameter	Limits			Unit	Test Conditions	Device
		Min	Typ	Max			
V <sub>IL</sub>	Input Low Voltage (All Except RESET, X1, X2)	-0.5		0.8	V		All
V <sub>IL1</sub>	Input Low Voltage (RESET, X1, X2)	-0.5		0.6	V		All
V <sub>IH</sub>	Input High Voltage (All Except XTAL1, XTAL2, RESET)	2.0		V <sub>CC</sub>	V		All
V <sub>IH1</sub>	Input High Voltage (X1, X2, RESET)	3.8		V <sub>CC</sub>	V		All
V <sub>OL</sub>	Output Low Voltage (BUS)			0.45	V	I <sub>OL</sub> = 2.0 mA	All
V <sub>OL1</sub>	Output Low Voltage (RD, WR, PSEN, ALE)			0.45	V	I <sub>OL</sub> = 1.8 mA	All
V <sub>OL2</sub>	Output Low Voltage (PROG)			0.45	V	I <sub>OL</sub> = 1.0 mA	All
V <sub>OL3</sub>	Output Low Voltage (All Other Outputs)			0.45	V	I <sub>OL</sub> = 1.6 mA	All
V <sub>OH</sub>	Output High Voltage (BUS)	2.4			V	I <sub>OH</sub> = -400 μA	All
V <sub>OH1</sub>	Output High Voltage (RD, WR, PSEN, ALE)	2.4			V	I <sub>OH</sub> = -100 μA	All
V <sub>OH2</sub>	Output High Voltage (All Other Outputs)	2.4			V	I <sub>OH</sub> = -40 μA	All
I <sub>L1</sub>	Leakage Current (T1, INT)			±10	μA	V <sub>SS</sub> ≤ V <sub>IN</sub> ≤ V <sub>CC</sub>	All
I <sub>L11</sub>	Input Leakage Current (P10-P17, P20-P27, EA, SS)			-500	μA	V <sub>SS</sub> + 0.45 ≤ V <sub>IN</sub> ≤ V <sub>CC</sub>	All
I <sub>L12</sub>	Input Leakage Current RESET	-10		-300	μA	V <sub>SS</sub> ≤ V <sub>IN</sub> ≤ 3.8V	All
I <sub>L0</sub>	Leakage Current (BUS, T0) (High Impedance State)			±10	μA	V <sub>SS</sub> ≤ V <sub>IN</sub> ≤ V <sub>CC</sub>	All
I <sub>DD</sub> + I <sub>CC</sub>	Total Supply Current*						
		80	100		mA		8035H
		95	110		mA		8039H
		80	100		mA		8048H
		95	110		mA		8749H

NOTE:

\*I<sub>CC</sub> + I<sub>DD</sub> is measured with all outputs disconnected; SS, RESET, and INT equal to V<sub>CC</sub>; EA equal to V<sub>SS</sub>.

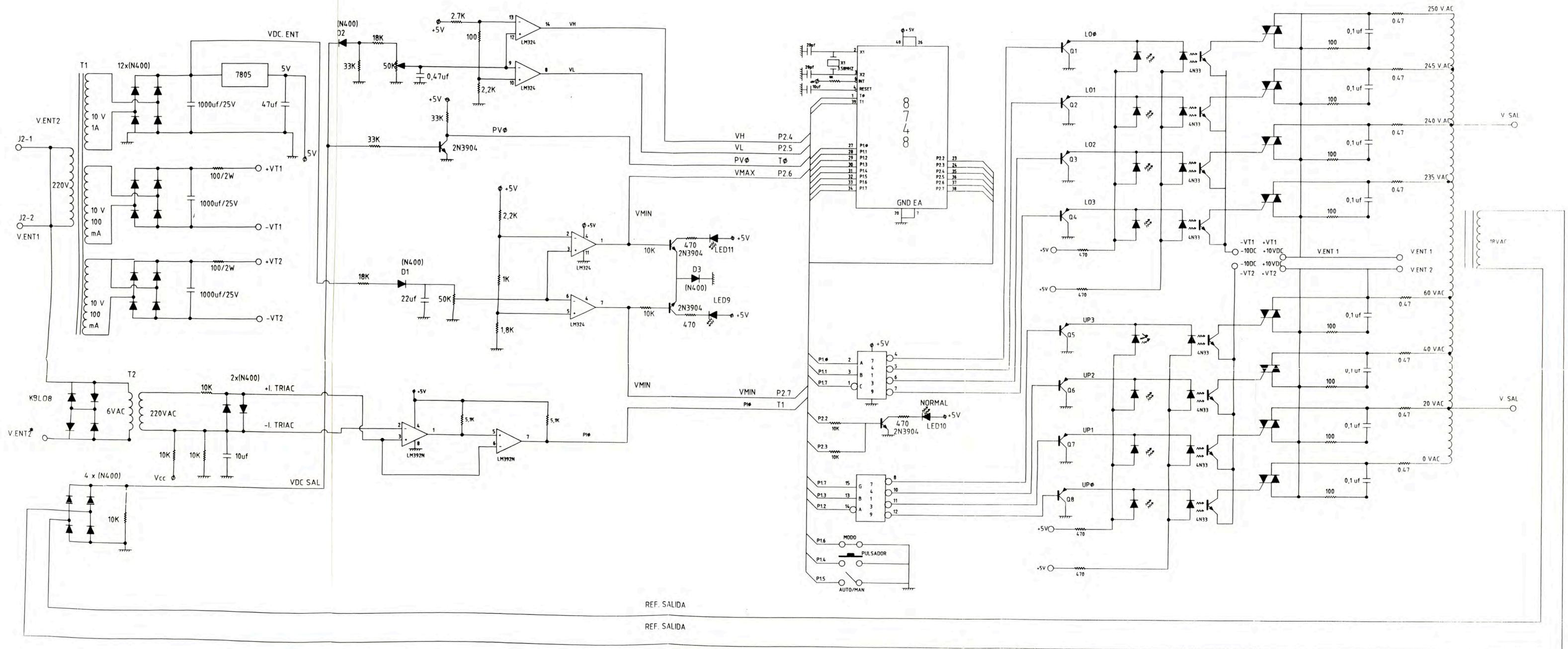
**A.C. CHARACTERISTICS**  $T_A = 0^\circ\text{C to } 70^\circ\text{C}; V_{CC} = V_{DD} = 5V \pm 10\%; V_{SS} = 0V$

Symbol	Parameter	f(t) (Note 3)	11 MHz		Unit	Conditions (Note 1)
			Min	Max		
t	Clock Period	1/xtal freq	90.9	1000	ns	(Note 3)
t <sub>LL</sub>	ALE Pulse Width	3.5t - 170	150		ns	
t <sub>AL</sub>	Addr Setup to ALE	2t - 110	70		ns	(Note 2)
t <sub>LA</sub>	Addr Hold from ALE	t - 40	50		ns	
t <sub>CC1</sub>	Control Pulse Width (RD, WR)	7.5t - 200	480		ns	
t <sub>CC2</sub>	Control Pulse Width (PSEN)	6t - 200	350		ns	
t <sub>DW</sub>	Data Setup before WR	6.5t - 200	390		ns	
t <sub>WD</sub>	Data Hold after WR	t - 50	40		ns	
t <sub>DR</sub>	Data Hold (RD, PSEN)	1.5t - 30	0	110	ns	
t <sub>RD1</sub>	RD to Data In	6t - 170		375	ns	
t <sub>RD2</sub>	PSEN to Data In	4.5t - 170		240	ns	
t <sub>AW</sub>	Addr Setup to WR	5t - 150	300		ns	
t <sub>AD1</sub>	Addr Setup to Data (RD)	10.5t - 220		730	ns	
t <sub>AD2</sub>	Addr Setup to Data (PSEN)	7.5t - 200		460	ns	
t <sub>AFC1</sub>	Addr Float to RD, WR	2t - 40	140		ns	(Note 2)
t <sub>AFC2</sub>	Addr Float to PSEN	0.5t - 40	10		ns	(Note 2)
t <sub>LAFC1</sub>	ALE to Control (RD, WR)	3t - 75	200		ns	
t <sub>LAFC2</sub>	ALE to Control (PSEN)	1.5t - 75	60		ns	
t <sub>CA1</sub>	Control to ALE (RD, WR, PROG)	t - 65	25		ns	
t <sub>CA2</sub>	Control to ALE (PSEN)	4t - 70	290		ns	
t <sub>CP</sub>	Port Control Setup to PROG	1.5t - 80	50		ns	
t <sub>PC</sub>	Port Control Hold to PROG	4t - 260	100		ns	
t <sub>PR</sub>	PROG to P2 Input Valid	8.5t - 120		650	ns	
t <sub>PF</sub>	Input Data Hold from PROG	1.5t	0	140	ns	
t <sub>DP</sub>	Output Data Setup	6t - 290	250		ns	
t <sub>PD</sub>	Output Data Hold	1.5t - 90	40		ns	
t <sub>PP</sub>	PROG Pulse Width	10.5t - 250	700		ns	
t <sub>PL</sub>	Port 2 I/O Setup to ALE	4t - 200	160		ns	
t <sub>LP</sub>	Port 2 I/O Hold to ALE	0.5t - 30	15		ns	
t <sub>PV</sub>	Port Output from ALE	4.5t + 100		510	ns	
t <sub>PRR</sub>	T0 Rep Rate	3t	270		ns	
t <sub>CY</sub>	Cycle Time	15t	1.36	15.0	μs	

NOTES:

- Control outputs CL = 80 pF; BUS outputs CL = 150 pF.
- BUS High Impedance Load 20 pF.
- f(t) assumes 50% duty cycle on X1, X2. Max clock period is for a 1 MHz crystal input.

**ANEXO E**  
**ESQUEMA DEL ESTABILIZADOR DE VOLTAJE**  
**USANDO UN MICROCONTROLADOR**



REF. SALIDA  
REF. SALIDA

UNIVERSIDAD NACIONAL DE INGENIERIA		
FACULTAD INGENIERIA ELECTRICA Y ELECTRONICA		TITULO ESQUEMA DEL ESTABILIZADOR DE VOLTAJE USANDO UN MICROCONTROLADOR
TAMANO A1	NOMBRE JUAN MEZA ARRIETA	REVISADO
FECHA mayo DE 1,998	HOJA 01 DE 01	ANEXO- E

## BIBLIOGRAFIA

1. Introducción a los microcontroladores de 16 bits familia 96.  
José Adolfo Gonzáles Vásquez.
2. Introducción a los microcontroladores.  
José Adolfo Gonzáles Vásquez.
3. Microprocesadores y microcontroladores aplicados a la industria.  
Torres Portero.
4. Microprocesadores.  
José María Angulo.
5. Microprocesadores, fundamentos, diseño y aplicaciones en la industria y en los microcomputadores.  
José María Angulo.
6. Sistemas controlados por computador.  
Karl y Astrom.
7. Embedded controller hand book.  
Intel.
8. CMOS data book.  
Motorola semiconductor products, inc.
9. Lineal data book, linear applications, hand book.  
National Semiconductor Corporation.
10. TTL logic data book.  
Texas instruments.
11. Tiristores y triacs.  
Henry Lilen.
12. Los tiristores y sus circuitos de comando.  
R. Berlioux.
13. Circuitos integrados lineales y amplificadores operacionales.  
Robert F. Coughlin.
14. Los microprocesadores Intel.  
Barry Brey.

15. Ensamblador básico.  
Alberto Rojas Ponce.
16. Manual de circuitos integrados TTL.  
Don Lancaster.
17. Arquitectura de computadoras.  
Morris Mano.
18. Electrónica digital integrada.  
Herbert Taub, Donald Schilling.
19. The TTL data book for design engineers.  
Texas instruments incorporated.
20. Organización y diseño de computadores.  
David Patterson - John Hennessy.