

ESCUELA POLITECNICA NACIONAL

FACULTAD DE INGENIERIA ELECTRICA

TESIS PREVIA A LA OBTENCION DEL TITULO

DE INGENIERO EN

ELECTRONICA Y CONTROL

DISEÑO Y CONSTRUCCION DE UN PROTOTIPO

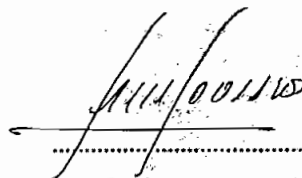
DE UN PROGRAMADOR PARA

MICROCONTROLADORES PIC16C5X

JUAN FRANCISCO SORIA AYALA

Quito, Enero de 1998

CERTIFICO QUE BAJO MI DIRECCION; LA
PRESENTE TESIS FUE REALIZADA EN SU
TOTALIDAD POR EL SEÑOR JUAN
FRANCISCO SORIA AYALA.

A handwritten signature in black ink, appearing to read 'Luis Corrales', written over a horizontal dotted line.

DR. LUIS CORRALES

AGRADECIMIENTO

AGRADEZCO A TODAS AQUELLAS PERSONAS QUE DE UNA U OTRA FORMA CONTRIBUYERON EN EL DESARROLLO DE LA PRESENTE TESIS, ESPECIALMENTE AL DR. LUIS CORRALES

DEDICATORIA

DEDICO TODO MI ESFUERZO, TANTO EN MI CARRERA COMO EN EL DESARROLLO DE LA PRESENTE TESIS A DIOS, A MIS PADRES, MI ESPOSA, MIS HIJOS Y AMIGOS QUE HAN SIDO QUIENES ME HAN ALENTADO PARA LLEGAR A ALCANZAR UNA DE MIS METAS TAN ANELADAS

INDICE

CAPITULO I	1
CARACTERISTICAS GENERALES DE LOS PIC16C5X	1
1.1 Introducción	1
1.2 Arquitectura	2
1.2.1 Bus de Datos	4
1.2.2 Osc1	5
1.2.3 Unidad Aritmética Lógica	5
1.2.4 Organización de los Registros	6
1.2.4.1 Registros de Direccionamiento Indirecto de Datos	6
1.2.4.2 Contador y Reloj en tiempo real	6
1.2.4.3 Contador de Programa	7
1.2.4.4 Registro de Estado	8
1.2.4.5 Registro selector de Registros (FSR)	9
1.2.4.6 Pórticos de Entrada y Salida	10
1.3 Distribución de Pines	11
1.4 Mapa de Memoria	13
1.4.1 Organización de la Memoria de Programa	13
1.4.2 Organización de la Memoria de Datos	14
1.5 Características	15
1.6 Modos de Direccionamientos	16
1.7 Set de Instrucciones	17
1.8 Registros Especiales	31
1.8.1 El Registro de trabajo W	31
1.8.2 El Registro TRIS	31
1.8.3 El Registro Opción	32

1.8.4	Registro de Configuración de los PICS	32
1.8.4.1	Opciones del Oscilador	33
1.8.5	Circuito de Reset	33
1.8.6	Circuito de Vigilancia "Watchdog"	34
1.8.7	Modo Sleep	34
1.8.8	Código de Protección	35
1.8.9	Localizaciones ID	35
CAPITULO II		36
METODOS DE PROGRAMACION		36
2.1	Programación de Memorias	37
2.2	Programación de las Memorias de los PICS	37
2.2.1	Descripción de los pines durante la programación de los PIC16C5X	38
2.2.2	Modos de Programar y verificar	39
2.2.3	Incremento de PC sin que programe/verifique	40
2.2.4	Verificación con el incremento del PC	40
2.2.5	Método de Programación	41
2.2.5.1	Ancho del pulso de programación	42
2.2.5.2	Direcciones especiales de memoria	42
2.2.5.3	Localización del código de usuario ID	42
2.2.6	Configuración del microcontrolador	45
2.2.7	Formatos de los datos en Hex para los PIC16C5X	46
2.2.7.1	Formato HXL/HXH	46
2.2.7.2	Formato Intel (HEX)	46
2.2.8	Características eléctricas en el modo de programación y verificación	47
2.2.8.1	Características en DC	47
2.2.8.2	Características en AC	48

CAPITULO III	50
DISEÑO DEL HARDWARE	50
3.1 Requerimientos generales	50
3.2 Diagrama de Bloques	51
3.3 Diseño Circuital	52
3.3.1 Interface para la comunicación entre el computador y el programador	54
3.3.2 Control de los voltajes aplicados al chip a ser programado	55
3.3.3 Fuentes de voltaje regulados	58
CAPITULO IV	62
DISEÑO DEL SOFTWARE	62
4.1 Descripción general	62
4.2 Opciones del Programador	63
4.3 Diagramas de Flujo	67
4.4 Desarrollo del Software	70
CAPITULO V	83
PRUEBAS Y RESULTADOS	83
CAPITULO VI	90
ANALISIS TECNICO ECONOMICO	90
CAPITULO VII	93
CONCLUSIONES	93
BIBLIOGRAFIA	95
REFERENCIAS	95
ANEXOS	96

CAPITULO I

CARACTERISTICAS GENERALES DE LOS PIC16C5X

1.1 INTRODUCCION

En 1980, aproximadamente, los fabricantes de circuitos integrados iniciaron la difusión de un nuevo circuito para medición, instrumentación y control de procesos, al que llamaron microcontrolador.

Un microcontrolador es un circuito integrado que contiene toda la estructura (arquitectura) de un microcomputador, esto es: CPU, RAM, ROM y circuitos de I/O.

En los últimos años han ganado un espacio considerable en el mercado del diseño de circuitos de instrumentación, medición y control lo cual podría servir a nuestro medio, tanto en la industria como en los laboratorios de instrumentación.

En la Tabla 1.1, se presentan algunas características resumidas de varios microcontroladores que se utilizan en el mercado:

	8031	8051	6805	68HC11	1654	Z8	BO196	HD647180
Fabricante	INT	INT	MOT	MOT	MC	ZILOG	INT	HI
Bus de Datos	8	8	8	8	8/12	8	16	8
Bus de Direcc.	16	interno	interno	interno	interno	16	16	interno
Pines	40	40	40	48/52	18	40	48/68	84
RAM	128	128	176	512	32x8	144	232	512
EPROM	NO	4K	4K	8K	512x12	20k	8K	16K
Timers	2	2	1	2	1	2	2	2
Puertos I/O	2	32	31	6	13	32	32	32
Seriales	1	1	1	2	NO	1	1	2
Interrupciones	2	5	5	18	NO	8	8	6
A/O	NO	NO	NO	SI	NO	NO	SI	NO
Otros	NO	NO	NO	EPROM	NO	BASIC	PWM	DMA

Tabla 1.1 Resumen de características entre diferentes microcontroladores que existen en el mercado

Los parámetros utilizados en la elaboración de la Tabla 1.1 son: el fabricante (Intel, Motorola, Micro Chip, Zilog, etc), existencia de buses de datos y direcciones para expansión externa, número de pines; cantidad de memoria RAM, ROM, temporizadores,

puertos seriales y puertos paralelos de entrada y salida (I/O).

En la Tabla 1.1 se presenta una muestra comparativa de algunos microcontroladores existentes en el mercado, durante la década del 80 al 90. Todos los meses, las casas fabricantes de semiconductores impulsan nuevos modelos; este proceso no se detendrá en muchos años. El diseñador de circuitos electrónicos debe analizar sus características, para seleccionar el componente más acorde con sus necesidades.

Ultimamente se están difundiendo en el mercado electrónico, la familia de microcontroladores PIC. Esta posee algunas ventajas sobre sus competidores, ya que presenta un set reducido de instrucciones, bajo consumo de potencia, tiene el modo de operación (SLEEP) para ahorro de energía, código de protección contra lectura y un bajo costo.

La familia de los PIC16C5X está formada por una amplia variedad de componentes con diferente capacidad de memoria, número de entradas/salidas (I/O), rangos variables de frecuencia y diferentes tipos de encapsulados los cuales están disponibles dependiendo de los requerimientos de aplicación y producción.

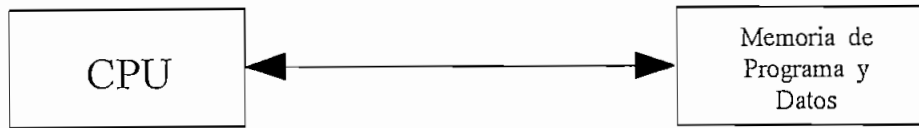
1.2 ARQUITECTURA

La elevada calidad de los microcontroladores de la familia PIC16C5X puede ser atribuida a su arquitectura basada en microprocesadores RISC.

La arquitectura (estructura interna) de los PIC16C5X se inicio en base a la llamada arquitectura **HARVARD**, la cual se basa en tener buses separados tanto para las instrucciones como para los datos. En la Figura 1.1, se muestra la diferencia entre la estructura tradicional de un computador o Von Neumann y la arquitectura Harvard.

En la estructura Harvard, al tener dos espacios de memoria diferentes uno para datos y otro para programas, además de la utilización de dos buses distintos: uno para la

comunicación entre la CPU y los datos y otro para la comunicación entre la memoria de programa y la CPU, permite tener un conjunto de instrucciones potentes diseñadas para controlar procesos a nivel de bit, byte y operaciones a alta velocidad. Esto se logra porque es posible superponer los ciclos de búsqueda y ejecución de las instrucciones.



Estructura de Von Neumann



Estructura de Harvard

Figura 1.1 Diagrama de Bloques de las Estructuras internas de los Controladores

Esto significa que, mientras una instrucción se ejecuta utilizando el bus de datos (8 bits), la siguiente se está leyendo desde la memoria de programa y cargándose en el registro de instrucciones utilizando el bus de instrucciones (12 bits).

La separación de los buses de datos e instrucciones en los PIC16C5X permite optimizar las instrucciones en palabras simples de 12 bits. La mayoría de las instrucciones se ejecutan en un solo ciclo de reloj excepto las instrucciones de salto que se ejecutan en dos ciclos de reloj. El conjunto de instrucciones está formado solamente por 33 instrucciones, de las cuales 18 están orientadas al trabajo con bytes, 4 para manipular bits y 11 van orientadas al control, por estas razones se dice que estos microcontroladores se comportan como procesadores tipo RISC (Reduced Instruction Set Computers).

La memoria interna de programa de los PIC16C58 y PIC16C57 es de 2K x 12, en los PIC16C55 y PIC16C56 es de 1K x 12 y en los PIC16C54 es de 512 x 12.

En los PIC16C5X se puede acceder directa o indirectamente tanto a los registros como a la memoria de datos. Todos los registros de funciones especiales incluyendo el contador de programa se encuentran mapeados en la memoria de datos.

En la Figura 1.2, se presenta el diagrama de bloques simplificado de los microcontroladores de la familia PIC16C5X

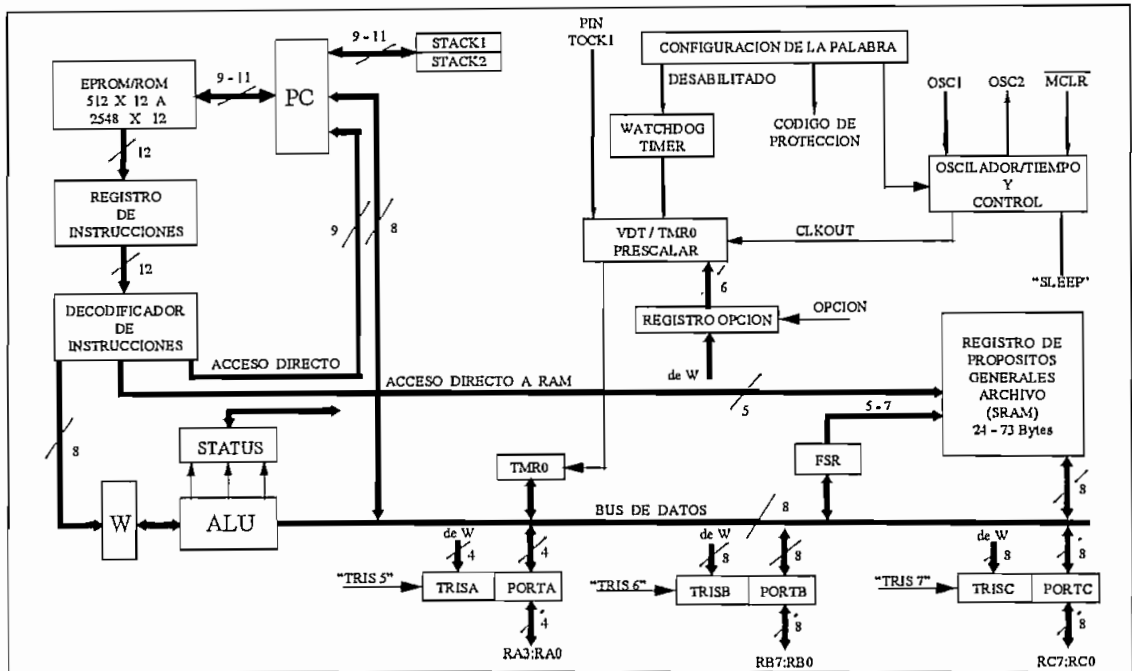


Figura 1.2 Diagrama de Bloques Simplificado de los Microcontroladores PIC16C5X

1.2.1 Bus de Datos:

Este bus es de 8 bits; es la vía central que intercomunica los siguientes elementos: el conjunto de registros de la memoria RAM, los puertos de entrada/salida (I/O), el registro de estado, la unidad aritmética - lógica (ALU), el registro de selección (FSR), el registro de preescalamiento del temporizador y el registro de trabajo (W) entre los más importantes.

1.2.2 OSC1

La entrada de reloj es el pin OSC1/CLKIN, se divide internamente por cuatro para generar cuatro señales de reloj Q1, Q2, Q3 y Q4 las cuales se encuentran en cuadratura, estas son necesarias para controlar las operaciones de búsqueda, decodificación y ejecución de las instrucciones.

Un ciclo de instrucción consiste de cuatro ciclos Q (Q1, Q2, Q3 y Q4). El ciclo de búsqueda empieza con el incremento del contador de programa en Q1. En el ciclo de ejecución, la instrucción buscada es retenida en el registro de instrucciones (IR) durante Q1, esta instrucción es decodificada y ejecutada durante el ciclo de Q2, Q3 y Q4. La memoria de datos es leída durante Q2 (operación de lectura) y escrita durante Q4. En la Figura 1.3 se muestra el diagrama de tiempo cuando se ejecuta una instrucción. Internamente el contador de programa (PC) es incrementado por Q1

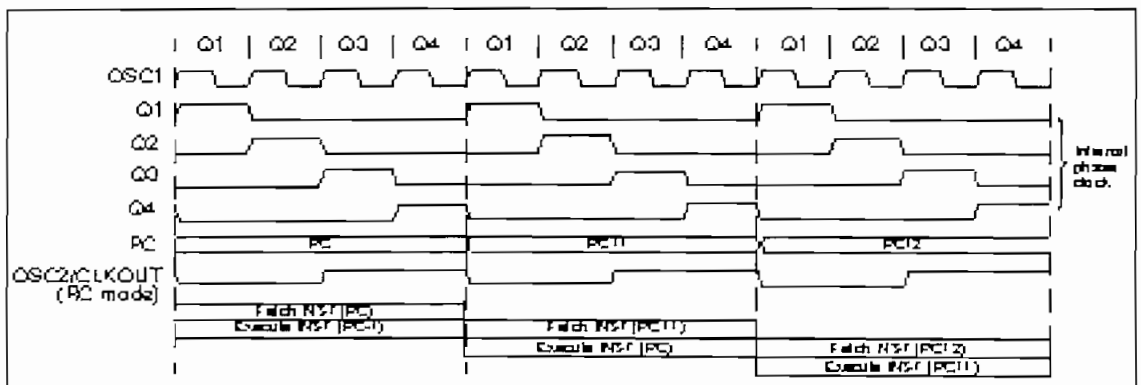


Figura 1.3 Diagrama de Tiempo de la ejecución de una Instrucción

1.2.3 Unidad aritmética lógica (ALU)

La unidad aritmética lógica (ALU) de los PIC es de 8 bits, contiene un registro temporal de trabajo (W) y las funciones lógicas necesarias para ejecutar operaciones aritméticas y operaciones de algebra de Boole entre el registro de trabajo W y cualquier otro registro.

1.2.4 Organización de los registros

Los registros especiales y las localidades de memoria, están organizadas en un conjunto o arreglo general. Hay 32 registros de RAM de 12 bits cada registro direccionables directamente y un sistema de bancos de memoria con 16 bytes cada uno, que se emplean cuando se tienen programas muy grandes y se requiere direccionar memorias mayores. Por ejemplo, hay 80 registros en el PIC16C57.

El conjunto de registros está dividido en dos grupos funcionales: los registros operacionales y los de propósito general. Los primeros incluyen el registro de selección (FSR), el registro contador de reloj (TOCK1), el contador de programa (PC), el registro de estado (STATUS) y los pórtricos de entrada/salida (I/O).

A continuación se hace un resumen de las características más relevantes de los mismos.

1.2.4.1 Registro de direccionamiento indirecto de datos

Es un registro que no está disponible en forma física. Se utiliza el contenido del registro FSR para seleccionar de manera indirecta uno de los 32 registros disponibles como registros de datos o registro puntero, según el tipo de instrucción.

1.2.4.2 Contador y reloj en tiempo real

En el registro TMR0 se puede escribir o leer como en cualquiera de los otros registros. Este registro es de 8 bits y puede ser incrementado por una señal externa aplicada al pin TOCK1 o por el reloj interno de instrucciones. Las aplicaciones más obvias de este registro son el conteo de eventos y la medición de tiempos. El modo Timer es seleccionado cuando se coloca un 0 lógico en el bit TOCS (bit 5 del registro option). El modo de contador se selecciona cuando seteamos el bit TOCS en 1 lógico.

En la Figura 1.4 se muestra un diagrama de bloque simplificado del modulo TMR0

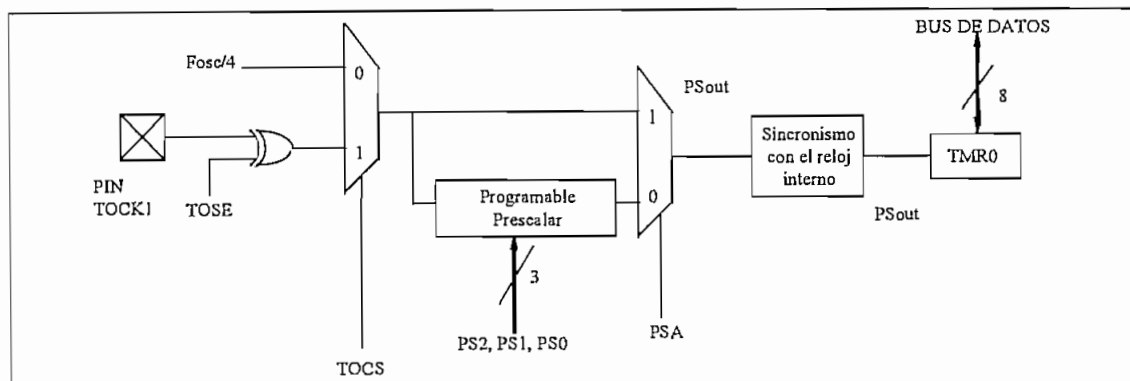


Figura 1.4 Diagrama de Bloques del TMR0

1.2.4.3 Contador de Programa

El contador de programa, PC, se utiliza para direccionar las palabras de 12 bits del programa del usuario, almacenadas en la memoria ROM/EPROM.

Según el tipo de microcontrolador que se utilice, el número de bits del registro PC y de la pila (Stack), es de 9 para los PIC16C54 y 16C55, de 10 bits para el PIC16C56 y 11 en el PIC16C57 que permite direccionar hasta 2048 palabras. Después de la condición de Reset el contador de programa se inicia con todos sus bits en "uno". Durante la ejecución de un programa se incrementa en uno el PC con cada instrucción, a menos que cambie como resultado de algunas instrucciones especiales (GOTO, CALL, RETLW, etc).

La instrucción GOTO permite el cambio directo de los 9 bits bajos del contador de programa (bits del 0 a 8). En los PIC16C56/PIC16C57 los dos bits altos del contador de programa (bits 10 y 9) se cargan con la selección de la página PA1 y PA0 (bits 6 y 5 del registro de estado); en esta forma, la instrucción GOTO permite saltar a cualquier posición del programa.

La instrucción CALL cambia directamente los 8 bits bajos del contador de programa (bits del 0 a 7) mientras que el noveno bit (bit 8) se pone en "cero". Por esta razón, todas las entradas a las subrutinas o saltos calculados están limitados a las primeras 256

posiciones de cualquier página de memoria de programa. El valor actual del PC, incrementado en uno, se carga en la pila. En los PIC16C56 y 16C57, los dos bits altos del PC (bits 9 y 10) se cargan con los bits de selección de la página PA1 y PA0.

La instrucción RETLW carga el contador de programa con el contenido de la parte alta de la pila. Si el PC es el destino de cualquier instrucción (por ejemplo MOVWF 2), el resultado calculado se cargará en los 8 bits bajos del contador de programa.

El direccionamiento relativo se logra sumando el contenido del registro de trabajo W al contador de programa (PC); éste es ampliamente utilizado para elaborar tablas de datos y para saltos.

1.2.4.4 Registro de Estado

Registro que contiene las condiciones de la unidad aritmética lógica ALU (bit de acarreo, bit de cero, etc), el Reset y los bits de preselección de página para el caso de memorias mayores a 512 palabras (para los PIC16C56/57).

Los bits TO y PD de este registro no se puede escribir, TO es el bit de superación o rebase del tiempo del temporizador, PD es el bit de bajo consumo.

Estos dos bits pueden consultarse para determinar si se ha producido la condición de Reset por el rebase del temporizador del circuito de vigilancia (watchdog), por la puesta en marcha después de una instrucción SLEEP o por la puesta en "cero" del terminal MCLR (borrado principal) del microcontrolador.

Los bits PA1 y PA0 seleccionan la página de memoria. La dirección directa especificada en la instrucción es solamente válida en la página particular de memoria que seleccionan PA1 y PA0. En la Figura 1.5 se muestra los bits que conforman el registro STATUS.

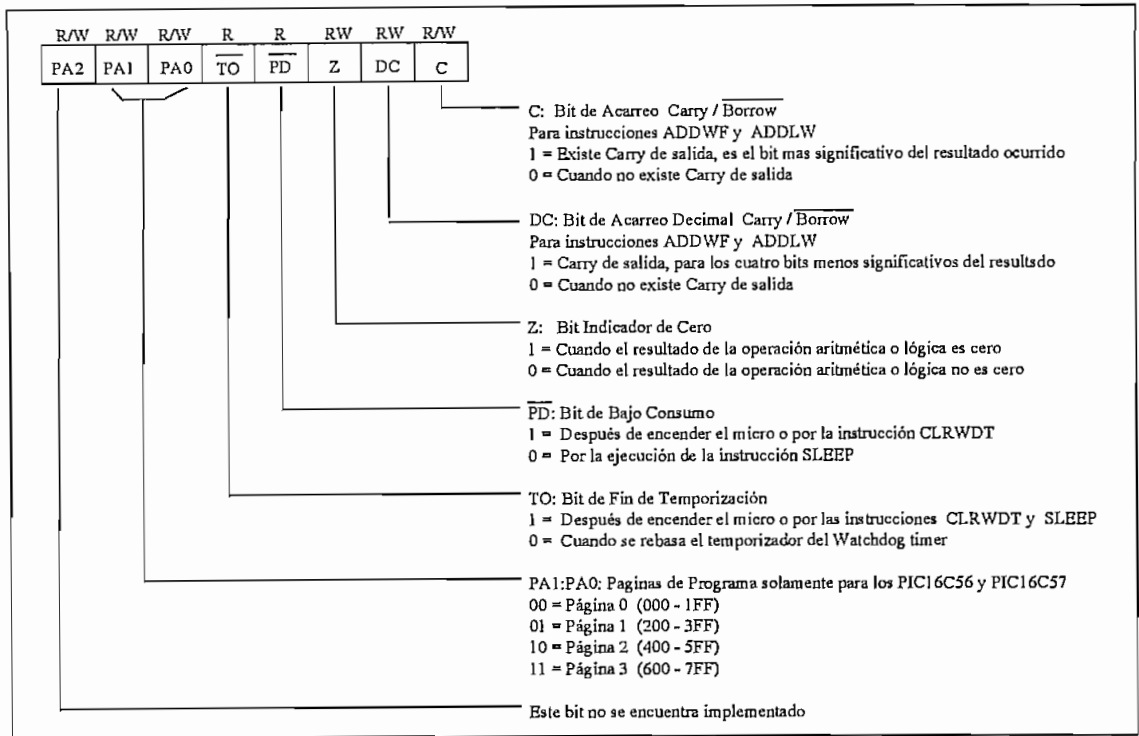


Figura 1.5 Registro STATUS

1.2.4.5 Registro selector de registros (FSR)

Se utiliza en asocio con el registro de direccionamiento indirecto de datos para seleccionar indirectamente uno de los 32 registros disponibles. El registro FSR es de 5 bits para los PIC 16C54/CR54/C55/C56 o de 7 bits para los PIC 16CR57.

Los bits de 0 al 4 del FSR se utilizan como puntero para direccionar la memoria de datos desde la dirección 00H hasta 1FH. Cuando el bit 4 de FSR es cero se puede direccionar desde 00H hasta 0FH esto es las 16 localidades de la parte baja de la memoria y si se pone este bit en uno lógico se podrá acceder a las 16 localidades de la parte alta de la memoria.

Los bits 5, 6 y 7 de FSR son de lectura y siempre se leen con el valor de "1".

Si en el programa no se utilizan llamadas indirectas, este registro se puede utilizar como un registro de propósito general de cinco bits. Para los PIC16C57 los bits 5 y 6 de

FSR permiten seleccionar el banco en uso (00 = banco 0, 01 = banco 1, 10 = banco 2 y 11 = banco 3).

Si el bit 4 de FSR es cero los bits 5 y 6 del mismo son ignorados, el bit 7 de FSR no esta implementado y siempre se lee como "1".

1.2.4.6 Pórticos de Entrada y Salida (I/O)

Los puertos de entrada/salida (I/O) puerto A, puerto B, y puerto C pueden escribirse y leerse bajo el control del programa, como cualquier otro registro. Sin embargo, las instrucciones de lectura leen siempre el estado del pin de entrada/salida sin importar que el pin esté definido como entrada o como salida. En las condiciones de Reset, todas las entradas/salidas quedan definidas como entradas y los registros de control de I/O (TRIS A, TRIS B y TRIS C) se actualizan al valor de "1". En la Figura 1.6 se muestra la estructura de un pin de entrada/salida.

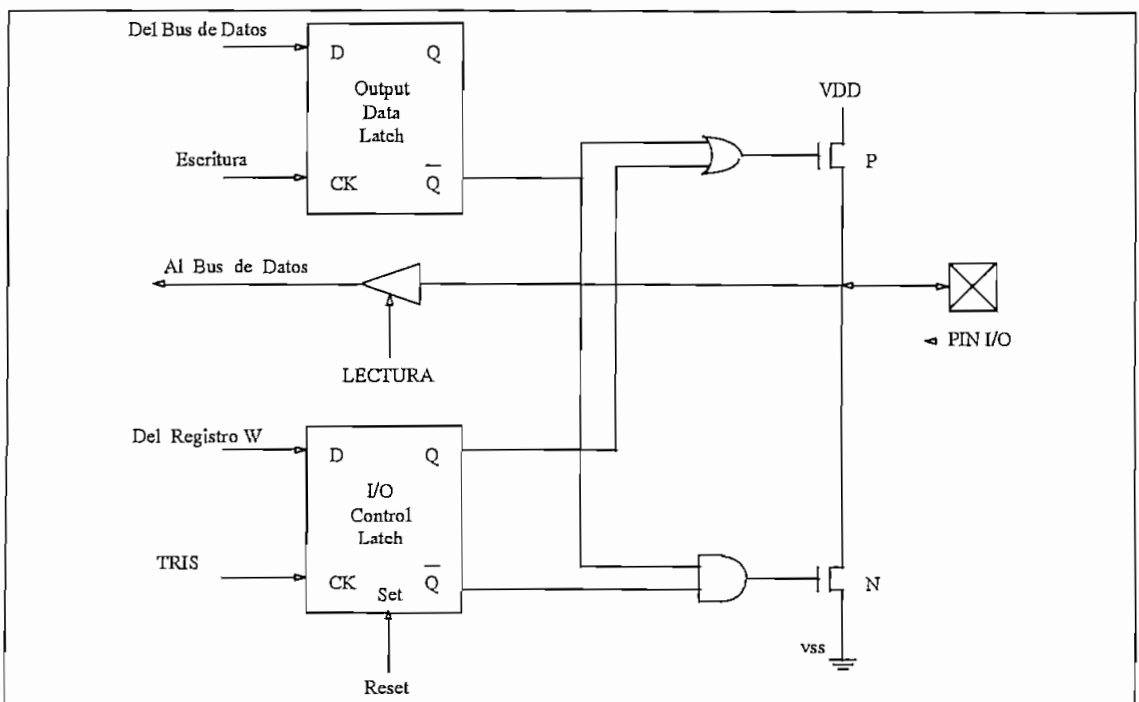


Figura 1.6 Circuito Equivalente para un Pin I/O

Para definir cualquier pin como salida, es necesario la ejecución de una instrucción

TRIS f con el correspondiente "cero" en el registro de trabajo W. Es conveniente, a lo largo de un programa, redefinir todos los pines de entrada y salida en intervalos regulares.

1.3 DISTRIBUCION DE PINES

En las siguientes tablas se presenta la descripción de la distribución de pines de los PIC16C5X.

DESCRIPCIÓN DE LOS PINES DE SALIDA PARA LOS PIC16C54/CR54/C56

Nombre	DIP,SOIC No.	SSOP No.	I/O/P Tipo	Buffer Tipo	Descripción
RA0	17	19	I/O	TTL	Pórtico Bidireccional de entrada y salida
RA1	18	20	I/O	TTL	
RA2	1	1	I/O	TTL	
RA3	2	2	I/O	TTL	
RB0	6	7	I/O	TTL	Pórtico Bidireccional de entrada y salida
RB1	7	8	I/O	TTL	
RB2	8	9	I/O	TTL	
RB3	9	10	I/O	TTL	
RB4	10	11	I/O	TTL	
RB5	11	12	I/O	TTL	
RB6	12	13	I/O	TTL	
RB7	13	14	I/O	TTL	
TOCK1	3	3	I	ST	Entrada de reloj o TMR0 Timer. Cuando no esta en uso se debería unir a VSS o VDD, esto reduce el consumo de corriente.
MCLR/Vpp	4	4	I	ST	Reset/Voltaje de entrada de programación. Este pin se activa en bajo. el voltaje sobre este pin no debe exceder el valor de VDD.
OSC1/CLKIN	16	18	I	ST	Entrada del oscilador de cristal, entrada de una fuente externa de reloj
OSC2/CLKOUT	15	17	O	----	Salida del oscilador de cristal. Se puede conectar un oscilador de cristal, un oscilador tipo RC. La frecuencia de salida de este pin debe ser 1/4 de la frecuencia del OSC1.
VDD	14	15,16	P	----	Fuente de suministro positiva para la lógica y los pines de entrada/salida
VSS	5	5,6	P	----	Referencia de tierra para la lógica y los pines de entrada/salida

DESCRIPCIÓN DE LOS PINES DE SALIDA PARA LOS PIC16C55/C57

Nombre	DIP,SOIC No.	SSOP No.	I/O/P Tipo	Buffer Tipo	Descripción
RA0 RA1 RA2 RA3	6 7 8 9	5 6 7 8	I/O I/O I/O I/O	TTL TTL TTL TTL	Pórtico Bidireccional de entrada y salida
RB0 RB1 RB2 RB3 RB4 RB5 RB6 RB7	10 11 12 13 14 15 16 17	9 10 11 12 13 15 16 17	I/O I/O I/O I/O I/O I/O I/O	TTL TTL TTL TTL TTL TTL TTL	Pórtico Bidireccional de entrada y salida
RC0 RC1 RC2 RC3 RC4 RC5 RC6 RC7	18 19 20 21 22 23 24 25	18 19 20 21 22 23 24 25	I/O I/O I/O I/O I/O I/O I/O	TTL TTL TTL TTL TTL TTL TTL	Pórtico Bidireccional de entrada y salida
TOCK1	1	2	I	ST	Entrada de reloj o TMR0 Timer. Cuando no esta en uso se debería unir a VSS o VDD, esto reduce el consumo de corriente.
MCLR/Vpp	28	28	I	ST	Reset /Voltaje de entrada de programación. Este pin se activa en bajo. el voltaje sobre este pin no debe exceder el valor de VDD.
OSC1/CLKIN	27	27	I	ST	Entrada del oscilador de cristal, entrada de una fuente externa de reloj
OSC2/CLKOUT	26	26	O	----	Salida del oscilador de cristal. Se puede conectar un oscilador de cristal, un oscilador tipo RC. La frecuencia de salida de este pin debe ser 1/4 de la frecuencia del OSC1.
VDD	2	3, 4	P	----	Fuente de suministro positiva para la lógica y los pines de entrada/salida
VSS	4	1,14	P	----	Referencia de tierra para la lógica y los pines de entrada/salida
NC	3, 5	----	----	----	No utilizado, no conectado

Legenda: I = Entrada, O = Salida, I/O = Entrada/Salida, P = Fuente, ---- = No usado.
TTL = Entrada TTL, ST = Entrada Schmit Trigger.

1.4 MAPA DE MEMORIA

La memoria en los microcontroladores PIC se divide en dos, una para los programas y otra para los datos.

1.4.1 ORGANIZACION DE LA MEMORIA DE PROGRAMA

Se tiene una memoria de programa (EPROM/ROM) de 512 palabras de 12 bits de ancho que pueden ser accedidas directamente. En la Figura 1.7 se muestra un esquema de como esta configurado el area de memoria de programa.

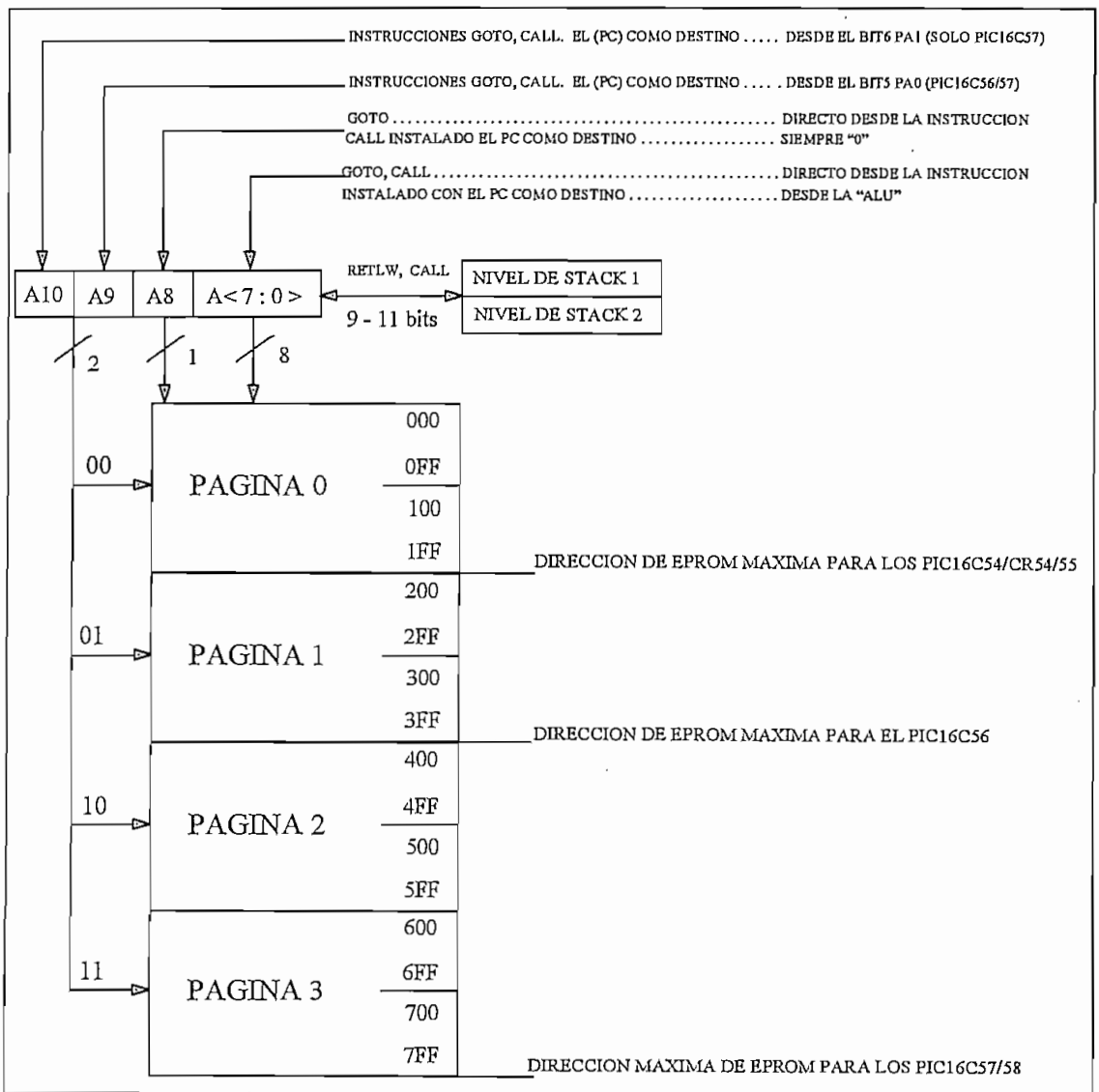


Figura 1.7 Mapa de Memoria de Programa

Para programas grandes que utilizan una mayor cantidad de memoria se deben utilizar dispositivos con mayor memoria, a estas localidades se puede acceder seleccionando 1 de las 4 páginas de 512 palabras habilitadas como se indica en la Figura 1.7.

1.4.2 ORGANIZACION DE LA MEMORIA DE DATOS

En la Figura 1.8 se muestra como está estructurado el área de memoria de datos.

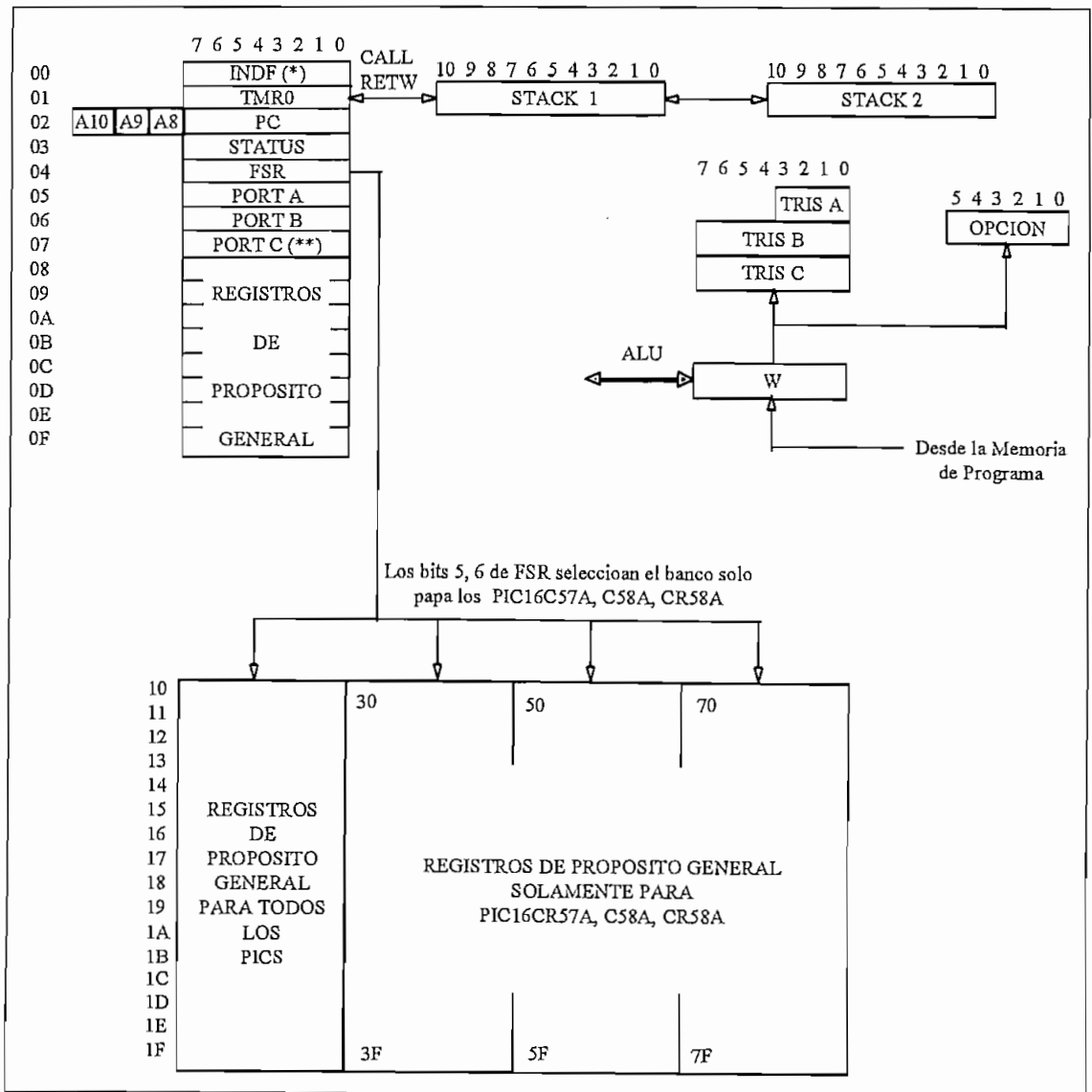


Figura 1.8 Mapa de Memoria de Datos

El bus de datos de 8 bits está conectado a dos funciones elementales básicas: el

archivo de registro compuesto de 80 localidades de 8 bits incluyendo pórtricos de entrada/salida y una unidad aritmética lógica (ALU) de 8 bits. Son accesibles directamente 32 bytes de RAM sobre un esquema de bancos, cada banco es de 16 bytes, son aplicados para acceder a memorias grandes de datos, como se muestra en la Figura 1.8.

Los datos pueden ser direccionados directamente o indirectamente usando el registro de selección de archivo (FSR).

El registro de archivos se divide en dos grupos, dependiendo de la función: los registros de función especial y los registros de propósito general. Los registros de función especial incluye al registro TIMER (TMR0), al contador de programa (PC), al registro STATUS, los registros de entrada/salida (Pórtricos) y el registro de selección de archivos (FSR). Los registros de propósito general son usados por datos y por información que permite el control de las instrucciones. Además, los registros de propósito general son usados para controlar la configuración de los pórtricos de entrada/salida y las opciones de preescalamiento.

1.5 CARACTERÍSTICAS

A continuación se presenta un resumen de las características eléctricas más importantes de los microcontroladores PIC16C5X, adicionalmente en el ANEXO 1 se presenta curvas características tanto para DC como para AC de la respuesta de estos microcontroladores a la frecuencia de oscilación, la temperatura, etc. Además se presentan las características eléctricas tanto para dispositivos de tipo comercial e industrial, hay que considerar que los valores que se van a presentar son afectados tanto por el tipo de oscilador que se está utilizando así como también por la frecuencia de operación o de trabajo

CARACTERISTICAS ELECTRICAS GENERALES

Rango de Valores máximos

Temperatura Ambiente	-55 C a +125 C
Voltaje sobre V_{DD} con respecto a V_{SS}	0 a +7.5V
Voltaje sobre MCLR con respecto a V_{SS}	0 a +14V
Voltaje sobre todos los otros pines con respecto a V_{SS}	-0.6V a $V_{DD} + 0.6V$
Potencia total de disipación	800 mW
Máxima corriente de salida en el pin V_{SS}	150 mA
Máxima corriente de entrada en el pin V_{DD}	50 mA
Máxima corriente de entrada en el pin TOCKI	$\pm 500 \mu A$
Máxima corriente de salida entregada por cualquier pin I/O	25 mA
Máxima corriente suministrada por cualquier pin I/O	20 mA
Máxima corriente suministrada por un solo pínico I/O	40 mA

1.6 MODOS DE DIRECCIONAMIENTO

Cada secuencia de instrucciones es controlada por el contador de programa (PC), como este se incrementa automáticamente al ejecutarse cada línea de programa, las operaciones de control del programa supone diferentes modos de direccionamiento como: directo, indirecto y relativo. Adicionalmente sobre el chip es empleado dos niveles de STACK lo cual permite un fácil uso de subrutinas.

Como se dijo anteriormente se tiene direccionamiento directo, indirecto, relativo. El direccionamiento directo está soportado mediante instrucciones especiales que cargan los datos de la memoria RAM en el registro de trabajo W o viceversa. Los datos pueden direccionarse de manera indirecta por medio del registro Selector, FSR. En la Figura 1.9 se muestra como interactúa el registro FSR con la memoria de datos. A continuación se presenta un ejemplo de como sería el direccionamiento indirecto así como también un esquema de como se utilizan los registros para este direccionamiento. En el siguiente

programa se borran las direcciones 20h hasta 2Fh de la memoria RAM.

```

movlw    0 x 10    ;Inicializa el puntero
movwf   FSR       ;
Next    clrf      INDF    ;borra localidades
        incf      FSR     ;incrementa el puntero
        btfsc    FSR,4   ;borrado todo?
        goto     Next    ;No, borra y va a la siguiente localidad de
                        ;memoria
    
```

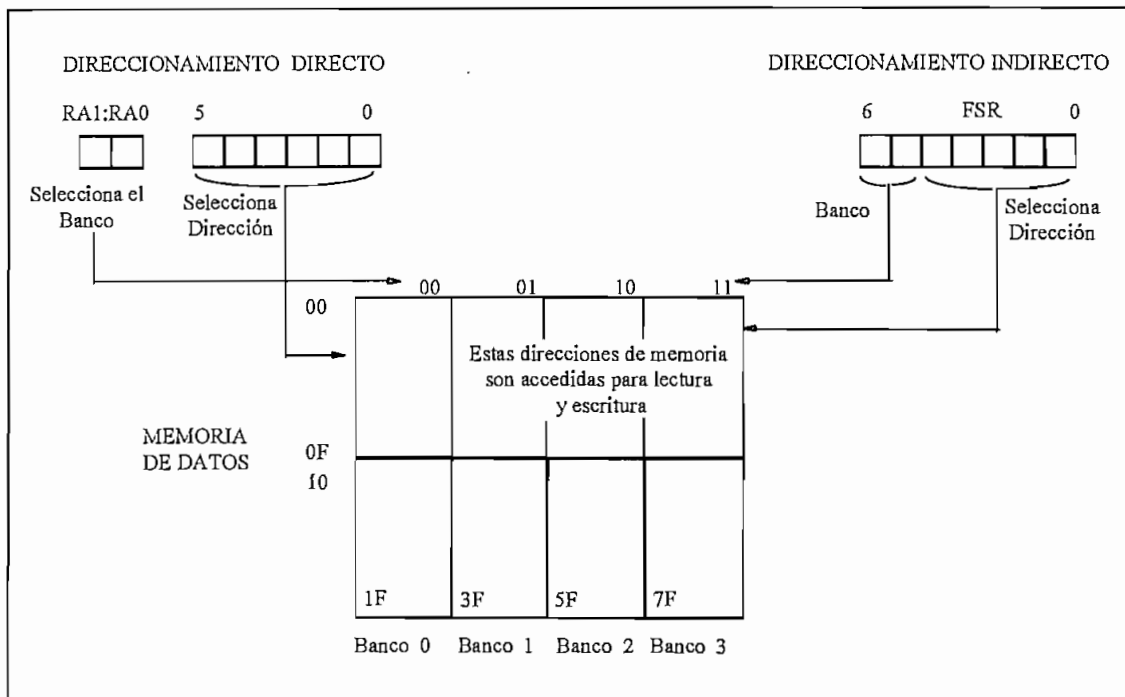


Figura 1.9 Direccionamiento Directo/Indirecto

1.7 SET DE INSTRUCCIONES

Son todas las órdenes o comandos que el microcontrolador puede interpretar y ejecutar.

Las instrucciones para los PIC16C5X tienen una longitud de 12 bits, en el que se encuentra el código del mнемónico y el código del operando (los datos son manipulados directamente en un registro localizado en la memoria). A continuación se presenta el set de instrucciones para estos microcontroladores:

ADDWF	Add W and f
Sintaxis:	[Etiqueta] ADDWF f,d
Operandos:	$0 \leq f \leq 127$ $d \in [0, 1]$
Operación:	$(W) + (f) \rightarrow (des)$
Banderas Afectadas:	C, DC, Z
Código de máquina:	0001 11df ffff
Descripción:	Suma el contenido del registro W y el registro "f". Si "d" es cero el resultado se copia en el registro W. Si "d" es uno el resultado se copia en el registro "f".
Ciclos de reloj:	1
Ejemplo:	ADDWF FSR, 0 Antes de la instrucción W = 17 FSR = C2 Después de la instrucción W = D9 FSR = C2
ANDWF	AND W con f
Sintaxis:	[Etiqueta] ANDWF f,d
Operandos:	$0 \leq f \leq 127$ $d \in [0, 1]$
Operación:	$(W) \text{ AND } (f) \rightarrow (des)$
Banderas Afectadas:	Z
Código de máquina:	0001 01df ffff
Descripción:	Se realiza la operación lógica AND entre el contenido del registro W y el registro "f". Si "d" es cero el resultado se copia en el registro W. Si "d" es uno el resultado se copia en el registro "f".
Ciclos de reloj:	1
Ejemplo:	ADDWF FSR, 1 Antes de la instrucción W = 17 FSR = C2 Después de la instrucción W = 17 FSR = C2
ANDLW	AND Dato con W
Sintaxis:	[Etiqueta] ANDLW k
Operandos:	$0 \leq k \leq 255$
Operación:	$(W) \text{ AND } (k) \rightarrow (W)$
Banderas Afectadas:	Z
Código de máquina:	1110 kkkk kkkk
Descripción:	Se realiza la operación lógica AND entre el contenido del registro W y el dato de ocho bits "k". El resultado se ubica en el registro W.

Ciclos de reloj: 1
Ejemplo: ANDLW 5F
 Antes de la instrucción
 W = A3
 Después de la instrucción
 W = 03

BCF **Borra Bit f**

Sintaxis: [Etiqueta] BCF f,b
Operandos: $0 \leq f \leq 127$
 $0 \leq b \leq 7$
Operación: $0 \rightarrow (f \ll b)$
Banderas Afectadas: Ninguno
Código de máquina: 0100 bbbf ffff
Descripción: El bit "b" en el registro "f" es borrado.
Ciclos de reloj: 1
Ejemplo: BCF FLAG_REG, 7
 Antes de la instrucción
 FLAG_REG = C7
 después de la instrucción
 FLAG_REG = 47

BSF **Setea un Bit de f**

Sintaxis: [Etiqueta] BSF f,b
Operandos: $0 \leq f \leq 127$
 $0 \leq b \leq 7$
Operación: $1 \rightarrow (f \ll b)$
Banderas Afectadas: Ninguno
Código de máquina: 0101 bbbf ffff
Descripción: Se setea el bit "b" del registro "f".
Ciclos de reloj: 1
Ejemplo: BSF FLAG_REG, 7
 Antes de la instrucción
 FLAG_REG = 0A
 después de la instrucción
 FLAG_REG = 8A

BTFSC **Prueba un Bit de f, salta y lo borra**

Sintaxis: [Etiqueta] BTFSC f,b
Operandos: $0 \leq f \leq 127$
 $0 \leq b \leq 7$
Operación: Salta si $(f \ll b) = 0$
Banderas Afectadas: Ninguno
Código de máquina: 0110 bbbf ffff
Descripción: Si el bit "b" en el registro "f" es cero entonces la siguiente instrucción es saltada. Si el bit "b" es uno la próxima instrucción es buscada, durante la ejecución de la presente instrucción y es descargada, y una instrucción de no

operación es ejecutada instantáneamente, siendo esta una instrucción de dos ciclos de reloj.

Ciclos de reloj:

Ejemplo:

1(2)
 Aquí BTFSC FLAG, 1
 Falso GOTO PROCESS_CODE
 Verdad -

Antes de la instrucción

PC = dirección de Aquí

Después de la instrucción

Si Flag «1» = 0

PC = Dirección (Verdad)

Si Flag «1» = 1

PC = Dirección (Falso)

BTFSS

Prueba un Bit de f, salta y lo setea

Sintaxis:

[Etiqueta] BTFSS f,b

Operandos:

$0 \leq l \leq 127$

$0 \leq b \leq 7$

Operación:

Salta si (f«b») = 1

Banderas Afectadas:

Ninguno

Código de máquina:

0111 bbbf ffff

Descripción:

Si el bit "b" en el registro "f" es uno entonces la siguiente instrucción es saltada. Si el bit "b" es uno, entonces la próxima instrucción es buscada, durante la ejecución de la presente instrucción y es descargada, y una instrucción de no operación es ejecutada instantáneamente, siendo esta una instrucción de dos ciclos de reloj.

Ciclos de reloj:

Ejemplo:

1(2)
 Aquí BTFSS FLAG, 1
 Falso GOTO PROCESS_CODE
 Verdad -

Antes de la instrucción

PC = dirección de Aquí

Después de la instrucción

Si Flag «1» = 0

PC = Dirección (Falso)

Si Flag «1» = 1

PC = Dirección (Verdad)

CALL

Subrutina CALL

Sintaxis:

[Etiqueta] CALL k

Operandos:

$0 \leq k \leq 2047$

Operación:

(PC) + 1 → Top of Stack

k → PC«8; 0»

(Status «6, 5») → PC «10, 9»

0 → PC «8»

Banderas Afectadas:

Ninguno

Código de máquina: 1001 kkkk kkkk
Descripción: La subrutina CALL. Primero la dirección de retorno (PC+1) se coloca en el stack. Los ocho bits de dirección son cargados inmediatamente en el PC bits del (7: 0) Los bits (10, 9) del PC son cargados en el STATUS (6: 5). El bit 8 del PC es borrado. Esta instrucción tiene dos ciclos de reloj.
Ciclos de reloj: 2
Ejemplo: Aquí CALL allá
 Antes de la instrucción
 PC = Dirección (Aquí)
 Después de la instrucción
 PC = Dirección (allá)
 TOS = Dirección (aquí)

CLRF **Borra f**

Sintaxis: [Etiqueta] CLRF f
Operandos: $0 \leq f \leq 127$
Operación: 00h → (f)
 1 → Z
Banderas Afectadas: Z
Código de máquina: 0000 011f ffff
Descripción: El contenido del registro "f" es borrado y se setea la bandera Z.
Ciclos de reloj: 1
Ejemplo: CLRF FLAG_REG
 Antes de la instrucción
 FLAG_REG = 5A
 Después de la instrucción
 FLAG_REG = 00
 Z = 1

CLRW **Borra el registro W**

Sintaxis: [Etiqueta] CLRW
Operandos: Ninguno
Operación: 00h → (W)
 1 → Z
Banderas Afectadas: Z
Código de máquina: 0000 0100 0000
Descripción: El registro W es borrado y se activa la bandera Z
Ciclos de reloj: 1
Ejemplo: CLRW
 Antes de la instrucción
 W = 5A
 Después de la instrucción
 W = 00
 Z = 1

CLRWDT	Borra el timer del circuito de vigilancia
Sintaxis:	[Etiqueta] CLRWDT
Operandos:	Ninguno
Operación:	00h → WDT 0 → WDT prescalado 1 → TO 1 → PD
Banderas Afectadas:	TO, PD
Código de máquina:	0000 0000 0100
Descripción:	La instrucción CLRWDT resetea el WDT . Siempre que se resetea se prescala el WDT. Los bits TO y PD del Status son seteados o activados.
Ciclos de reloj:	1
Ejemplo:	CLRWDT Antes de la instrucción WDT Counter = ? Después de la instrucción WDT Counter = 00 WDT prescala = 0 TO = 1 PO = 1
COMF	Complementa f
Sintaxis:	[Etiqueta] COMF f,d
Operandos:	$0 \leq f \leq 127$ $d \in [0, 1]$
Operación:	(f) → (des)
Banderas Afectadas:	Z
Código de máquina:	0010 01df ffff
Descripción:	El contenido del registro "f" es complementado. Si d es cero el resultado es enviado al registro W, y si d es uno el resultado de esta operación va al registro "f".
Ciclos de reloj:	1
Ejemplo:	COMF REG1, 0 Antes de la instrucción REG1 = 13 Después de la instrucción REG1 = 13 W = EC
DECF	Decrementa f
Sintaxis:	[Etiqueta] DECF f,d
Operandos:	$0 \leq f \leq 127$ $d \in [0, 1]$
Operación:	(f) - 1 → (des)
Banderas Afectadas:	Z
Código de máquina:	0000 11df ffff

Después de la instrucción
PC = dirección (Lazo)

INCF	Incrementa f
Sintaxis:	[Etiqueta] INCF f,d
Operandos:	$0 \leq f \leq 127$ $d \in [0, 1]$
Operación:	$((f) + 1 \rightarrow (des))$
Banderas Afectadas:	Z
Código de máquina:	0010 10df ffff
Descripción:	El contenido del registro "f" es incrementado. si "d" es cero el resultado es movido al registro w y si "d" es uno el resultado se presenta en el registro "f".
Ciclos de reloj:	1
Ejemplo:	INCF CNT, 1 Antes de la instrucción CNT = FF Z = 0 Después de la instrucción CNT = 00 Z = 1

INCFSZ	Incrementa f, salta si es cero
Sintaxis:	[Etiqueta] DECFSZ f,d
Operandos:	$0 \leq f \leq 127$ $d \in [0, 1]$
Operación:	$(f) + 1 \rightarrow (des)$, salta si el resultado = 0
Banderas Afectadas:	Ninguno
Código de máquina:	0011 11df ffff
Descripción:	El contenido del registro "f" es incrementado. Si d es cero el resultado es colocado en el registro W y si d es uno el resultado es presentado en el registro "f". Si el resultado del registro "f" es cero la siguiente instrucción no se ejecuta y en lugar de esta se ejecuta una instrucción NOP.
Ciclos de reloj:	1(2)
Ejemplo:	Aquí DECFSZ CNT, 1 GOTO LAZO Continua - - Antes de la instrucción PC = dirección (Aquí) Después de la instrucción CNT = CNT + 1 Si CNT = 0 PC = dirección (continua) Si CNT = 0 PC = dirección (Aquí + 1)

IORLW	OR entre un Dato con W
Sintaxis:	[Etiqueta] IORLW k
Operandos:	$0 \leq k \leq 255$
Operación:	(W).OR. (k) \rightarrow (W)
Banderas Afectadas:	Z
Código de máquina:	1101 kkkk kkkk
Descripción:	Se realiza la operación lógica OR entre el contenido del registro W y el dato de ocho bits "k". El resultado se ubica en el registro W.
Ciclos de reloj:	1
Ejemplo:	IORLW 35 Antes de la instrucción W = 9A Después de la instrucción W = BF
IORWF	OR entre los registros W con f
Sintaxis:	[Etiqueta] IORWF f,d
Operandos:	$0 \leq f \leq 127$ $d \in [0, 1]$
Operación:	(W) OR. (f) \rightarrow (des)
Banderas Afectadas:	Z
Código de máquina:	0001 00df ffff
Descripción:	Se realiza la operación lógica OR entre el contenido del registro W y el registro "f". Si "d" es cero el resultado se copia en el registro W. Si "d" es uno el resultado se copia en el registro "f".
Ciclos de reloj:	1
Ejemplo:	IORWF RESULT, 0 Antes de la instrucción RESUL = 13 W = 91 Después de la instrucción RESUL = 13 W = 93
MOVF	Mueve el registro f
Sintaxis:	[Etiqueta] MOVF f,d
Operandos:	$0 \leq f \leq 127$ $d \in [0, 1]$
Operación:	(f) \rightarrow (des)
Banderas Afectadas:	Z
Código de máquina:	0010 00df ffff
Descripción:	El contenido del registro "f" es movido al destino "d". si "d" es cero el destino es el registro W y si "d" es uno el destino es el registro "f". Usualmente se coloca a "d" en uno para poder comprobar los registros.
Ciclos de reloj:	1

Ejemplo: MOVF FSR, 0
Después de la instrucción
W = Valor del registro FSR

MOVLW **Mueve un Dato al registro W**

Sintaxis: [Etiqueta] MOVLW k
Operandos: $0 \leq k \leq 255$
Operación: (k) → (W)
Banderas Afectadas: Ninguna
Código de máquina: 1100 kkkk kkkk
Descripción: Los 8-bits del dato "k" es localizado en el registro W.
Ciclos de reloj: 1
Ejemplo: MOVLW 5A
Después de la instrucción
W = 5A

MOVWF **Mueve el contenido de W al registro f**

Sintaxis: [Etiqueta] MOVWF f
Operandos: $0 \leq f \leq 127$
Operación: (W) → (f)
Banderas Afectadas: Ninguna
Código de máquina: 0000 01df ffff
Descripción: Mueve el dato del registro W al registro "f".
Ciclos de reloj: 1
Ejemplo: MOVWF TEM_REG
Antes de la instrucción
TEMP_REG = FF
W = 4F
Después de la instrucción
TEMP_REG = 4F
W = 4F

NOP **No Operación**

Sintaxis: [Etiqueta] NOP
Operandos: Ninguna
Operación: No Operación
Banderas Afectadas: Ninguna
Código de máquina: 0000 0000 0000
Descripción: No operación.
Ciclos de reloj: 1
Ejemplo: NOP

OPTION **Carga el registro OPTION**

Sintaxis: [Etiqueta] OPTION
Operandos: Ninguna
Operación: (W) → OPTION

Banderas Afectadas: Ninguna
Código de máquina: 0000 0000 0010
Descripción: El contenido del registro W es cargado sobre el registro opción.
Ciclos de reloj: 1
Ejemplo: OPTION
 Antes de la instrucción
 W = 17
 Después de la instrucción
 OPTION = 17

RETLW **Retorna y coloca el dato en W**

Sintaxis: [Etiqueta] RETLW k
Operandos: $0 \leq k \leq 255$
Operación: (k) → (W)
 TOS → PC
Banderas Afectadas: Ninguna
Código de máquina: 1000 kkkk kkkk
Descripción: En el registro W se recupera los 8-bits del dato "k". El contador de programa PC es cargado con el valor alto del stack. Esta instrucción tiene dos ciclos.
Ciclos de reloj: 2
Ejemplo: CALL TABLA

```

TABLA
  ADDWF  PC
  RETLW  k1
  RETLW  k2
  -
  -
  RETLW  kn
  
```

Antes de la instrucción
 W = 07
 Después de la instrucción
 W = Valor de k7

RLF **Rota el archivo f a la izquierda con carry**

Sintaxis: [Etiqueta] RLF f,d
Operandos: $0 \leq f \leq 127$
 $d \in [0, 1]$
Operación: Mirar la descripción más abajo
Banderas Afectadas: C
Código de máquina: 0011 01df ffff
Descripción: El contenido del registro "f" es rotado un bit a la izquierda incluida la bandera del carry. Si "d" es cero el resultado es colocado en el registro W y si "d" es uno el resultado es colocado en el registro "f".

Ciclos de reloj: 1
Ejemplo: RLF REG1, 0
 Antes de la instrucción
 REG1 = 11100110
 C = 0
 Después de la instrucción
 REG1 = 11100110
 W = 11001100
 C = 1

RRF **Rota el archivo f a la derecha con carry**

Sintaxis: [Etiqueta] RRF f,d
Operandos: $0 \leq f \leq 127$
 $d \in [0, 1]$
Operación: Mirar la descripción más abajo
Banderas Afectadas: C
Código de máquina: 0011 00df ffff
Descripción: El contenido del registro "f" es rotado un bit a la derecha incluida la bandera del carry. Si "d" es cero el resultado es colocado en el registro W y si "d" es uno el resultado es colocado en el registro "f".

Ciclos de reloj: 1
Ejemplo: RRF REG1, 0
 Antes de la instrucción
 REG1 = 11100110
 C = 0
 Después de la instrucción
 REG1 = 11100110
 W = 01110011
 C = 1

SLEEP **Ingresa al modo Sleep**

Sintaxis: [Etiqueta] SLEEP
Operandos: Ninguno
Operación: 00h → WDT
 1 → TO
 0 → PD
Banderas Afectadas: TO, PD
Código de máquina: 0000 0000 0011
Descripción: El bit del status TO (Tiempo fuera) es seteado, el bit PD (Fuente baja) es borrada. El procesador entra en modo SLEEP cuando el oscilador deja de funcionar .

Ciclos de reloj: 1
Ejemplo: SLEEP

SUBWF **Resta W con f**

Sintaxis: [Etiqueta] SUBWF f,d
Operandos: $0 \leq f \leq 127$

	$d \in [0, 1]$
Operación:	$(f) - (W) \rightarrow (des)$
Banderas Afectadas:	C, DC, Z
Código de máquina:	0000 10df ffff
Descripción:	Se realiza la operación de la resta (utiliza el método del complemento de dos) el registro W del registro "f" . Si "d" es cero el resultado es colocado en el registro w y si "d" es uno el resultado es presentado en el registro "f".
Ciclos de reloj:	1
Ejemplo1:	<p>SUBWF REG1, 1</p> <p>Antes de la instrucción</p> <p>REG1 = 3</p> <p>w = 2</p> <p>C = ?</p> <p>Después de la instrucción</p> <p>REG1 = 1</p> <p>W = 2</p> <p>C = 1; el resultado es positivo</p>
Ejemplo2:	<p>Antes de la instrucción</p> <p>REG1 = 2</p> <p>w = 2</p> <p>C = ?</p> <p>Después de la instrucción</p> <p>REG1 = 0</p> <p>W = 2</p> <p>C = 1; el resultado es cero</p>
Ejemplo3:	<p>Antes de la instrucción</p> <p>REG1 = 1</p> <p>w = 2</p> <p>C = ?</p> <p>Después de la instrucción</p> <p>REG1 = FF</p> <p>W = 2</p> <p>C = 0; el resultado es negativo</p>

SWAPF	Swap f
Sintaxis:	[Etiqueta] SWAPF f,d
Operandos:	$0 \leq f \leq 127$ $d \in [0, 1]$
Operación:	$(f \ll 3; 0) \rightarrow (des \ll 7; 4)$ $(f \ll 7; 4) \rightarrow (des \ll 3; 0)$
Banderas Afectadas:	Ninguna
Código de máquina:	0011 10df ffff
Descripción:	Se intercambian los bits mas significativos con los bits menos significativos. Si "d" es cero el resultado es colocado en el registro W y si "d" es uno el resultado se presenta en el registro "f".
Ciclos de reloj:	1
Ejemplo:	SWAPF REG1, 0

Antes de la instrucción
 REG1 = A5
 Después de la instrucción
 REG1 = A5
 W = 5A

TRIS	Carga el registro TRIS
Sintaxis:	[Etiqueta] TRIS f
Operandos:	$5 \leq f \leq 7$
Operación:	(W) → Registro TRIS f
Banderas Afectadas:	Ninguna
Código de máquina:	0000 0000 0fff
Descripción:	El registro TRIS "f" (f = 5, 6 y 7) es cargado con el contenido del registro W.
Ciclos de reloj:	1
Ejemplo:	TRIS PORTA Antes de la instrucción W = A5 Después de la instrucción TRISA = A5
XORLW	OR Exclusivo Dato con W
Sintaxis:	[Etiqueta] XORLW k
Operandos:	$0 \leq k \leq 255$
Operación:	(W) .XOR. (k) → (W)
Banderas Afectadas:	Z
Código de máquina:	1111 kkkk kkkk
Descripción:	Se realiza la operación lógica OR EXCLUSIVO entre el contenido del registro W y el dato de ocho bits "k". El resultado se ubica en el registro W.
Ciclos de reloj:	1
Ejemplo:	XORLW AF Antes de la instrucción W = B5 Después de la instrucción W = 1A
XORWF	OR Exclusivo entre los registros W con f
Sintaxis:	[Etiqueta] XORWF f,d
Operandos:	$0 \leq f \leq 127$ $d \in [0, 1]$
Operación:	(W) XOR. (f) → (des)
Banderas Afectadas:	Z
Código de máquina:	0001 10df ffff
Descripción:	Se realiza la operación lógica OR EXCLUSIVA entre el contenido del registro W y el registro "f". Si "d" es cero el resultado se copia en el registro W. Si "d" es uno el resultado se copia en el registro "f".

Ciclos de reloj: 1
Ejemplo: XORWF REG, 0
Antes de la instrucción
REG = AF
W = B5
Después de la instrucción
REG = 1A
W = B5

1.8 REGISTROS ESPECIALES

Los Registros de propósito especial son: el registro de trabajo W, los registros TRIS A, TRIS B y TRIS C, el registro OPTION, el registro de configuración de los PICS, además se tiene los siguientes circuitos: el circuito de vigilancia Watchdog timer, el circuito de reset y la opción de poder seleccionar el y tipo de oscilador para los PICS que tienen memoria EPROM, en cambio en los PICS que son OTP (una sola vez programables) viene determinado o configurado el tipo de oscilador.

1.8.1 El registro de trabajo W

En las instrucciones de operaciones el registro W contiene el segundo operando además se utiliza intensamente para las operaciones que requieren transferencia interna de datos.

1.8.2 El registro TRIS

Los registros TRIS A, TRIS B y TRIS C, como se explicó, son registros que determinan si los pines de los puertos son entradas o salidas. El dispositivo de control de salida del registro es cargado con el contenido del registro W para ejecutarse la instrucción TRIS f. El valor de "f" es colocado en los bits del registro TRIS correspondiendo a una salida en el modo de alta impedancia. Al colocar un "0" el dato de salida es retenido en el

pin seleccionado

1.8.3 El registro OPTION

Se emplea para definir la asignación del "preescalador" (RTCC o Watchdog), el valor del mismo, la fuente y el flanco de la señal para hacer conteo. Este registro es de escritura solamente y tiene 8 bits de los cuales solo se utilizan 6 bits. Para ejecutar la instrucción OPTION, el contenido del registro W debe ser transferido al registro OPTION. En el Reset todos los bits del registro OPTION se ponen en "1". En la Figura 1.10 se muestra un esquema de la configuración de los bits de este registro.

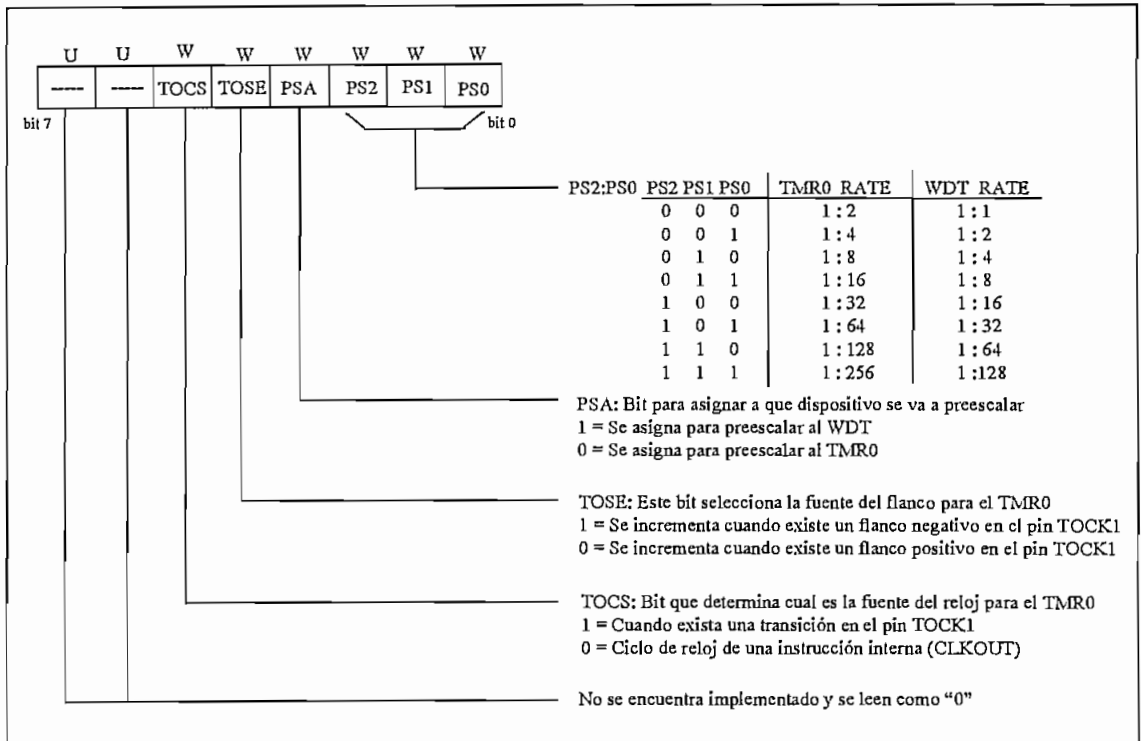


Figura 1.10 Registro OPTION

1.8.4 Registro de configuración de los PICS

Este registro de configuración consiste de 4 a 12 bits, dependiendo del dispositivo que se vaya a configurar. Estos bits de configuración pueden ser programados para seleccionar varios tipos de configuración. Dos bits permiten seleccionar el tipo de

oscilador, un bit habilita el Watchdog Timer y otro bit el código de protección. En la Figura 1.11 se muestra los bits del registro de configuración.

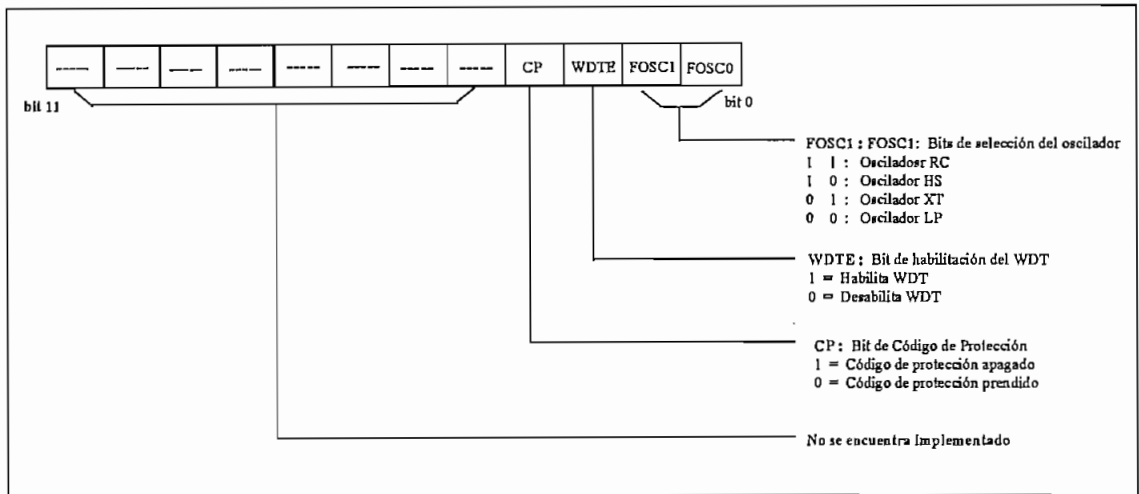


Figura 1.11 Configuración de los Microcontroladores

1.8.4.1 Opciones de oscilador

Hay cuatro tipos de oscilador en la familia de microcontroladores PIC16C5X: el XT; el HS o de alta velocidad; el LP o de baja velocidad y potencia; y el RC o de oscilador excitado por una red de resistencia y condensador. Los tipos XT, HS y LP requieren un cristal cerámico para establecer la oscilación. Los PIC OTP (programables una sola vez) se consiguen con una de las cuatro opciones de oscilador preprogramada; con los PICS basados en EPROM, el diseñador puede programar uno de los cuatro tipos de oscilador

1.8.5 Circuito de Reset

Los PIC16C5X tiene cuatro formas de producir el reset, los cuales son:

- Reset al momento del encendido (POR)
- MCLR reset durante la operación normal
- MCLR reset durante el SLEEP
- WDT reset producido por el desbordamiento del timer

Algunos registros no son afectados en la condición de reset, los PIC16C5X tienen

un circuito interno para generar el reset en el momento de la energización (POR), que evita el empleo de las tradicionales redes RC de reset. Para aprovechar el circuito POR, basta con unir el terminal MCLR a la fuente de +5 voltios. Cuando se trabaja con relojes de muy baja velocidad, es mejor utilizar la red RC externa.

1.8.6 Circuito de vigilancia "watchdog"

Se emplea para prevenir fallas catastróficas en la operación de un sistema cuando el PIC se bloquea por alguna razón. También se utiliza para temporizar o como detector de pulsos perdidos. La idea de funcionamiento de un circuito de vigilancia es la siguiente: este registro debe ser borrado por programa internamente, antes que el temporizador vigilante rebase su conteo y genere un reinicio total del microcontrolador (Reset). El circuito de vigilancia de los PIC no requiere reloj externo y opera con su propio oscilador RC.

1.8.7 Modo SLEEP

El modo SLEEP es seleccionado y ejecutado con la instrucción SLEEP, el dispositivo puede ser puesto en modo SLEEP (Ahorro de energía) por los siguientes eventos:

- Un reset externo de entrada; se produce cuando sobre el pin MCLR/Vpp aplicamos un 1 lógico.
- O por el desbordamiento del timer del WDT (esto si el WDT se encuentra habilitado).

Ambos eventos causan un reset del dispositivo, por tal motivo los bits \overline{TO} , \overline{PD} pueden ser usados para determinar la causa por la que se produjo el reset. El bit \overline{TO} es puesto en "cero" si el reset a ocurrido por el WDT, y el bit PD se pone en 1 lógico cuando

se produce el reset por el encendido del microcontrolador (POR) y el bit PD es borrado cuando se invoca a la instrucción SLEEP.

1.8.8 Código de Protección

El código (programa monitor) de la memoria de programa puede ser protegido al poner un "cero" en el bit del código de protección, del registro de configuración del microcontrolador.

Inmediatamente que el código de protección se habilita, todas las direcciones de la memoria no pueden ser leídas. Para dispositivos que tienen EPROM las localidades de la memoria de programa desde la 40h no pueden ser programadas en el futuro. Sin embargo las primeras 64 localidades, esto es desde la 00h hasta la 3Fh, pueden ser programadas.

1.8.9 Localizaciones ID

Cuatro localidades de memoria son designadas para las direcciones de ID, el usuario puede copiar en estas direcciones el checksum o otro código de identificación, estas direcciones no son asequibles durante la ejecución normal pero son leídas y escritas durante el proceso de verificación/programación. Se deben usar solamente los 4 bits menos significativos de las localidades ID.

CAPITULO II

METODOS DE PROGRAMACION

Una de las ventajas que tienen los sistemas digitales a los sistemas analógicos es la de poder almacenar información por períodos de tiempo largos o cortos, por esta razón los sistemas digitales son muy versátiles.

Uno de los medios para poder guardar la información son las memorias, en el mercado existen memorias que son solamente de lectura, memorias de acceso aleatorio y memorias de acceso secuencial, esto de acuerdo a la forma como se accede a la información y de acuerdo a la tecnología de fabricación pueden ser semiconductoras, magnéticas (como ejemplo las memorias de burbujas magnéticas), ópticas, etc.

Para nuestro estudio nos interesa las memorias que pueden ser programadas y que mantengan la información por períodos largos de tiempo, así se pierda o se interrumpa la fuente de energía, por este motivo las memorias que tienen este requerimiento son: las memorias ROM, PROM, EPROM Y EEPROM.

Las memorias ROM son memorias que solamente pueden ser leídas y vienen programadas por los fabricantes, las memorias PROM son memorias que pueden ser programadas por el usuario una sola vez y se las conoce como memorias OTP. Las dos memorias mencionadas anteriormente una vez programadas no se pueden borrar ya que el proceso es irreversible, si se requiere de un cambio se necesitará de un nuevo elemento. Por esta razón es que se crearon las memorias EPROM, las cuales se pueden programar varias veces.

A las memorias EPROM se las puede llevar a su condición inicial o borrarlas aplicando luz ultravioleta. Estas memorias sirven para desarrollar prototipos ya que se puede probar los programas monitores cuantas veces sea necesario hasta llegar a

depurarlos. Existen memorias EEPROM al igual que las memorias EPROM estas se pueden programar varias veces ya que se las puede regresar a la condición inicial por medios eléctricos.

2.1 PROGRAMACION DE MEMORIAS

Existen dos métodos para programar memorias, el primer método es utilizando la programación standard, la cual se basa en que el ancho del pulso de programación sea de 50 milisegundos; el otro método utilizan el algoritmo inteligente de programación, este método permite programar de una manera más rápida, debido a que el ancho del pulso de programación es menor al de la programación standard.

Para poder grabar información en una memoria básicamente se debe seguir el siguiente proceso: primero se debe colocar en el bus de direcciones la dirección en donde se desea grabar el dato, seguidamente el dato a ser grabado se coloca en el bus de datos de la memoria y por último se da un pulso de programación (V_{pp}) que tiene una cierta duración dependiendo del elemento a ser programado.

2.2 PROGRAMACION DE LAS MEMORIAS DE LOS PIC16C5X

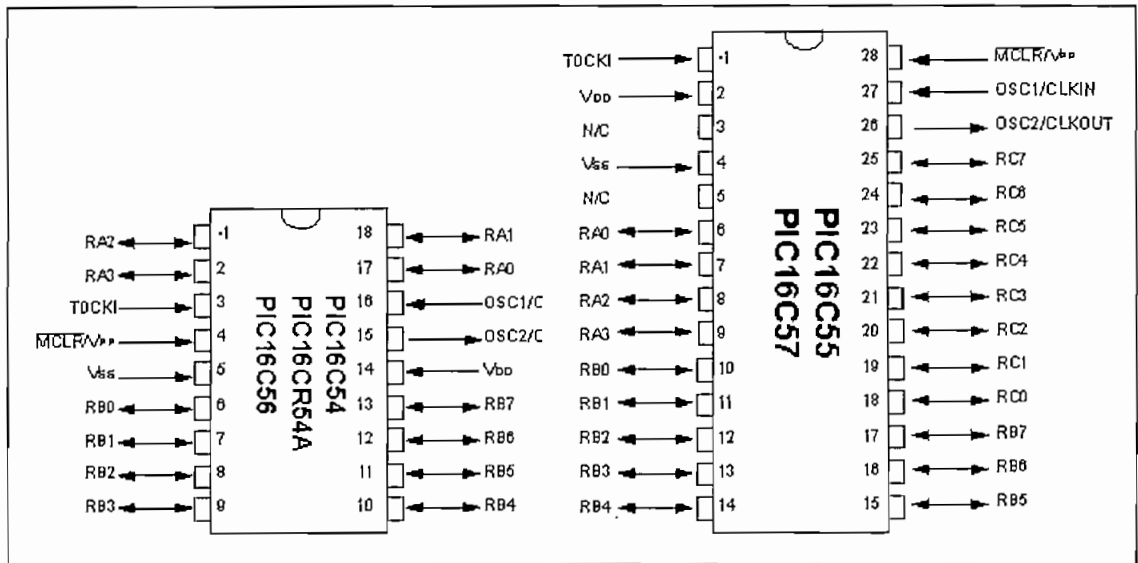
EL prototipo que se va a desarrollar permitirá programar las memorias EPROM y PROM de los microcontroladores de la serie PIC16C5X la cual, es una familia de microcontroladores CMOS con un chip EPROM para almacenar programas.

Debido a la arquitectura especial de estos microcontroladores (12-bits de ancho) y el bajo número de pines (18 pines), la metodología de la programación de la memoria EPROM es diferente para el estandar (ancho del byte) de EPROMS como por ejemplo el chip 27C256 que son memorias de 8-bits de ancho..

Se incluye las especificaciones de programación para los siguientes elementos:

PIC16C54, PIC16C58A, PIC16C56, PIC16C54A, PIC16CR54, PIC16C57, PIC16C55, PIC16CR57A, PIC16CR58A.

En la Figura 2.1 se muestra la distribución de pines de los microcontroladores antes mencionados.



programación (V_{pp}), mientras que el pin TOCK1 controla la habilitación de la programación /verificación del chip; además, el pin OSC1 controla el contador de programa PC el cual permitirá acceder a cualquier localidad (dirección) de la memoria.

2.2.2 MODOS DE PROGRAMAR Y VERIFICAR

La serie PIC16C5X usa el contador de programa interno (PC) para generar las direcciones de la EPROM, el pin OSC1 es llamado para incrementar el contador de programa (PC), el Voltaje de programación (V_{pp}) es suministrado a través del pin MCLR y el pin TOCK1 actúa como habilitador del chip, alternando entre programando y verificando.

El dato es aplicado o puede ser leído sobre el pórtilo A y el pórtilo B donde el bit más significativo es RB7 y el bit menos significativo el RA0.

El modo de programación/verificación es seleccionado por el cambio de nivel en el pin MCLR de un nivel bajo a un nivel alto, esto es desde V_L a $V_H=V_{pp}$. Al mismo tiempo, el pin TOCK1 es fijado a V_H y el pin OSC1 es fijado a V_L .

El contador de programa tiene un valor de "0 X FFF" porque MCLR tuvo antes un nivel bajo (V_L). Esta condición selecciona la dirección para poder configurar al microcontrolador, como primero verifica la primera dirección de la EPROM al ser accedida después de seleccionar el modo de programación.

Siempre que el pin MCLR esta en V_L , el dispositivo se encuentra en el estado de reset así como también los pines de entrada/salida.

Una vez incrementado el PC (al dar un pulso en el pin OSC1) se selecciona la dirección "0 x 000" para usar la memoria de programa; después se continúa con todas las otras direcciones de memoria desde la 001H hasta el final de la memoria.

2.2.3 INCREMENTO DEL PC SIN QUE PROGRAMA/VERIFIQUE

Después de seleccionar el modo de programación/verificación, poniendo el pin TOCK1 en bajo el programa de datos está presente sobre el pórtilco A y B en la dirección de memoria seleccionada por el contador de programa. La duración del tiempo que permanece en bajo el pin TOCK1 determina la longitud del pulso de programación.

Manteniendo el pin TOCK1 en bajo sin que cambie las señales sobre los pines MCLR y OSC1 coloca el contenido de la dirección de memoria seleccionada sobre el pórtilco A y el pórtilco B, para ser verificado, teniendo así un ciclo de programación exitoso. Este pulso de verificación sobre TOCK1 puede ser mucho más corto que el pulso de programación. Si la programación no fuese exitosa, TOCK1 puede ser puesto en bajo de nuevo para aplicar otro pulso de programación, seguido de nuevo por un pulso corto para la verificación.

Esta secuencia puede ser repetida varias veces como sea necesario para tener una programación exitosa. No se debe exceder de 25 pulsos o 25 ciclos, ya que esto podría dañar al elemento.

2.2.4 VERIFICACION CON EL INCREMENTO DEL PC

Si un ciclo de verificación muestra que la programación fue exitosa, el contador de programa puede ser incrementado al colocar un nivel alto sobre TOCK1, mientras que al mismo tiempo se da un pulso en alto sobre OSC1. Cuando TOCK1 y OSC1 están en alto, el contenido de la dirección de memoria seleccionada es colocada sobre el pórtilco A y B. El flanco de bajada de OSC1 hace que se incremente el contador de programa.

Una verificación rápida solamente con el incremento automático del PC puede ser ejecutada seleccionando el modo de programación/verificación como se describió anteriormente y colocando una señal de reloj en la entrada de OSC1. Si OSC1 está en alto,

la dirección de memoria seleccionada es colocada sobre el pórtico A y B inmediatamente en el flanco de bajada se incrementa el PC. De este modo la primera dirección puede ser verificada; después, seleccionando el modo de programación/verificación, se configura al microcontrolador. La dirección de la memoria de programa N puede ser alcanzada generando "N+1" flancos negativos sobre OSC1.

Cuando OSC1 es llevado a un nivel alto, de nuevo el contenido de la dirección [N] es colocada sobre el pórtico A y B, el tiempo que permanece el dato sobre los pórticos A y B es el tiempo que permanece en alto OSC1.

2.2.5 METODO DE PROGRAMACION

La técnica para programar se describe a continuación, la cual a sido diseñada para garantizar un buen margen de programación. Esta sin embargo requiere una variable, la fuente de energía para Vcc.

En esencia esta técnica requiere los siguientes pasos:

- 1.- Realizar un chequeo de la memoria si esta en blanco o vacía para $V_{DD}=5V$, si no esta en blanco reportar falla.
- 2.- Guardar el programa con pulsos de 100 us y verificar después de cada pulso para $V_{DD}=V_{DDP}$. Cuando $V_{DDP}=V_{DD}$ el rango referido durante la programación es de (4.5 a 5.5V).

Condiciones para programar:

a) $V_{PP} = 13.0V$ a $13.25V$.

$$V_{DD} = V_{DDP} = 4.5V \text{ a } 5.5V$$

V_{PP} debe ser $> V_{DD} + 7.25V$ para alcanzar que se active el modo de programación.

b) Condición de verificación

$$V_{DD} = V_{DDP}$$

$$V_{PP} > V_{DD} + 7.5V \text{ pero no debe exceder de } 13.25V.$$

Si se localiza una falla de programación después de "N" pulsos (se sugiere un programa de 25 pulsos) este reporta error como una falla de programación.

- 3.- Localizar todo el programa.
- 4.- Verificar todas las direcciones
- 5.- Reportar la programación

En la Figura 2.2 se muestra un diagrama de flujo del método de programación que se va a utilizar.

2.2.5.1 ANCHO DEL PULSO DE PROGRAMACION

Cuando se programa una celda de memoria de la EPROM, se requiere un pulso de programación (T_{PW}), se recomienda que sea de 100 us. El máximo número de pulsos de programación por dato debe ser limitado a 25.

2.2.5.2 DIRECCIONES ESPECIALES DE MEMORIA

El espacio de direcciones ID está solo habilitado si el dispositivo está en el modo de prueba o de verificación/programación. En el caso de operación normal la dirección de memoria 0x000 a 0xNNN puede ser accedido por el incremento del contador de programa.

La configuración del microcontrolador puede ser solo accedida inmediatamente después de pasar de un nivel bajo a un nivel alto en MCLR.

2.2.5.3 LOCALIZACION DEL CODIGO DE USUARIO ID

Por definición las primeras cuatro direcciones (TTT a TTT+3) son reservadas para el uso del usuario (cliente). Se recomienda que el cliente use solo los cuatro bits menos

significativos y el resto de bits llenarlos con "0"s.

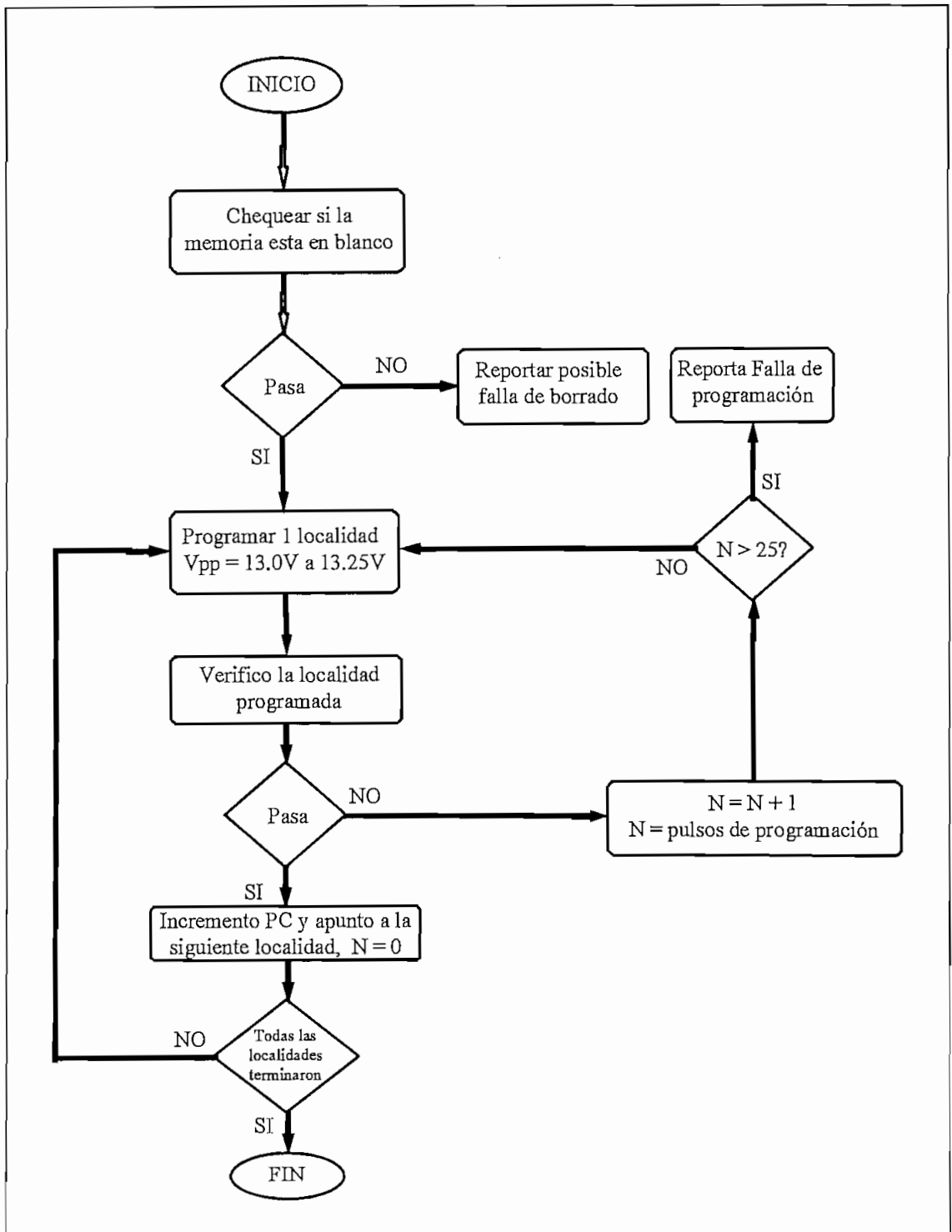


Figura 2.2 Diagrama de Flujo del Método de Programación

El usuario puede guardar el código de identificación en las direcciones de identificación y se puede habilitar la lectura de este código después de haber sido

programado el bit de código de protección. A continuación se presenta un ejemplo de código de usuario.

Si el código ID de usuario es 0xD1E2, las direcciones de ID deben tener los siguientes datos:

```
ID0: 0000 0000 1101
ID1: 0000 0000 0001
ID2: 0000 0000 1110
ID3: 0000 0000 0010
```

Leyendo estas cuatro direcciones de memoria, se debe tener sobre el pórtico A la siguiente secuencia de bits "1101", "0001", "1110", "0010" como es "D1E2".

En la Figura 2.3 se presenta el mapa de la memoria de programa en el modo de programación/verificación para los microcontroladores PIC16C5X.

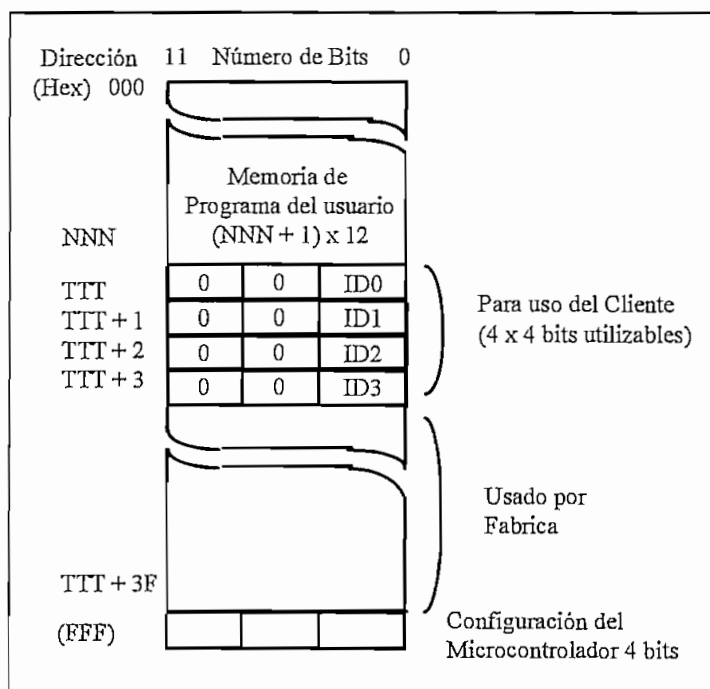


Figura 2.3 Mapa de Memoria de Programa en el Modo de Programación/Verificación

NNN: Dirección normal de la memoria EPROM. NNN = 1FF para los PIC16C54, 54A, 55.
 NNN = 3FF para los PIC16C56; y
 NNN = 7FF para los PIC16C57, 58A

TTT: Inicio de las direcciones especiales de la EPROM para el área de direcciones ID

2.2.6 CONFIGURACION DEL MICROCONTROLADOR

La configuración del microcontrolador sirve para seleccionar el tipo de oscilador, si está habilitado o deshabilitado el watchdog timer y para establecer el código de protección, requiere un pulso de programación de 10ms o una serie de pulsos de 100us es preferible a un pulso de 10ms. La dirección de la configuración del microcontrolador es la primera dirección después de haber accedido al modo de programación/verificación. Los bits que se utilizan son del 0 al 3 y los otros bits del 4 al 11 se leen como "1"s.

En la Figura 2.4 se presenta un mapa de los bits que conforman la dirección de la configuración de los microcontroladores.

	11	10	9	8	7	6	5	4	3	2	1	0
PIC16CR58A	CP	CP	CP	CP	CP	CP	CP	CP	CP	WDTE	FOSC1	FOSC0
PIC16C5X	----	----	----	----	----	----	----	----	CP	WDTE	FOSC1	FOSC0
	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	RA3	RA2	RA1	RA0

Figura 2.4 Mapa de Bits de Configuración de los Microcontroladores PIC16C5X

En la Tabla 2.2 se presenta el valor que deben tomar los bits de configuración para las diferentes opciones del microcontrolador.

RA3 - RA11 CP	RA2 WDTE	RA1 FOSC1	RA0 FOSC2	FUNCION	SELECCION
1	X	X	X	Memoria desprotegida	Por defecto
0	X	X	X	Memoria protegida	
X	1	X	X	Watchdog Timer habilitado	Por defecto
X	0	X	X	Watchdog Timer deshabilitado	
X	X	1	1	RC: Oscilador	Por defecto
X	X	1	0	HS: Cristal de alta velocidad	
X	X	0	1	XT: Cristal Standar	
X	X	0	0	LP: Cristal de baja frecuencia	

Tabla 2.2 Configuración de los Microcontroladores

1 = Aplicar un nivel alto en el pin I/O durante la programación

0 = Aplicar un nivel bajo en el pin I/O durante la programación

x = No importa

2.2.7 FORMATOS DE LOS DATOS EN HEX PARA LOS PIC16C5X

Los programadores de los PIC16C5X deben estar contruidos y diseñados para poder aceptar y enviar datos en uno de los siguientes formatos: formato de 8 bits separados (HXL/HXH) y el formato HEX Intel.

2.2.7.1 FORMATO (HXL/HXH)

En el formato (HXL/HXH), el dato se divide en dos partes HXL y HXH, este formato es como un formato normal de 8 bits, excepto que los bits bajos del dato son copiados en el archivo HXH y los bits altos del dato son copiados en el archivo HXL.

2.2.7.2 FORMATO INTEL (HEX)

Este formato produce un archivo de 8 bits con una combinación de byte bajo y alto. Cada dirección solo puede contener 8 bits, este formato es utilizado para transferir los programas en los programadores de EPROM de los microcontroladores PIC16C5X. A continuación se presenta un ejemplo de este formato:

```
:0400100000000000EC
:1000000000000000000000000000000000000000F0
:0400100000000000EC
:10003200000028004006800A800E800C80028016D
:06007200FF08FF08190A57
:00000001FF
:BBAAAATTHHHH.....HHHHCC
```

BB: Representa el número de datos que contiene la línea.

AAAA: Cuatro dígitos que representa la dirección de inicio en donde se deben grabar los datos.

TT: Estos dos dígitos nos indica si el archivo ha finalizado o no, siempre es "00" y si el archivo a finalizado será "01".

HH: Dos dígitos hexadecimales que representa el dato.

CC: Dos dígitos hexadecimales que nos indican el checksum.

En la primera línea del ejemplo nos indica que se tiene 4 datos, que empieza en la dirección 0010 esto es la dirección 16, que el separador entre direcciones y datos es 00 y que los datos tiene un valor de 00 además el checksum tiene un valor de EC, en la última línea de archivo se tiene que el número de datos es 00, la dirección es 0000 y se tiene que el separador entre la dirección de inicio y los datos es "01" lo cual indica que es el final del archivo.

2.2.8 CARACTERÍSTICAS ELÉCTRICAS EN EL MODO DE PROGRAMACIÓN Y VERIFICACIÓN.

2.2.8.1 CARACTERÍSTICAS EN DC

CARACTERÍSTICA	SIMB	Mín.	Typ.	Max.	Uni.	Condiciones
Suministro de voltaje durante la programación	VDD p	4.75	5.0	5.25	V	
Suministro de corriente para VDD	IDDp			25.0	mA	VDD = 5, Fosc1 = 5MHz
Suministro de voltaje durante la verificación	VDD v	VDD mín		VDD max		
Voltaje sobre MCLR en el modo de programación/verificación	VHH 1	VDD + 3		VDD + 7,25	V	
Voltaje sobre MCLR durante la programación	VHH 2	12.5		13.5	V	
Suministro de corriente para el voltaje de programación	IHH			100	mA	
Corriente en el pin MCLR durante la programación (TOCK1 = 0)	IHH2		10.0	25.0	mA	VHH = 13.5 V VDD = 6.0V
Voltaje de entrada en bajo	VIL	VSS		0.15VDD	V	
Voltaje de entrada en alto	VIH	0.85VDD	5.0	VDD	V	

2.2.8.2

CARACTERÍSTICAS EN AC

CARACTERÍSTICA	SIMB.	Min.	Typ	Max.	Uni.	CONDICIONES
Tiempo de subida de MCLR	TR	0.15	1.0	5	us	
Tiempo de bajada de MCLR	TF	0.5	2.0	5	us	
Tiempo de programar el modo	TPS	1.0			us	
Tiempo de acceso al dato	TACC			250	ns	
Tiempo de lectura del dato	TDS	1.0			us	
Tiempo de retención del dato	TDH	1.0			us	
Tiempo para habilitar la salida	TOE	0		100	ns	
Tiempo para deshabilitar la salida	TOZ	0		100	ns	
Ancho del pulso de programación	TPW	10.0			us	
Ancho del pulso de programación	TPWF		10.000		us	Solo para configuración
Tiempo de recuperación	TRC	10.0			us	
Frecuencia sobre OSC1	FOSC	DC		5	MHz	Para incrementar el PC

En la Figura 2.5 se presenta el diagrama de tiempo en el modo de programación/verificación, así como también en la Figura 2.6 se presenta la forma de onda de la velocidad de verificación.

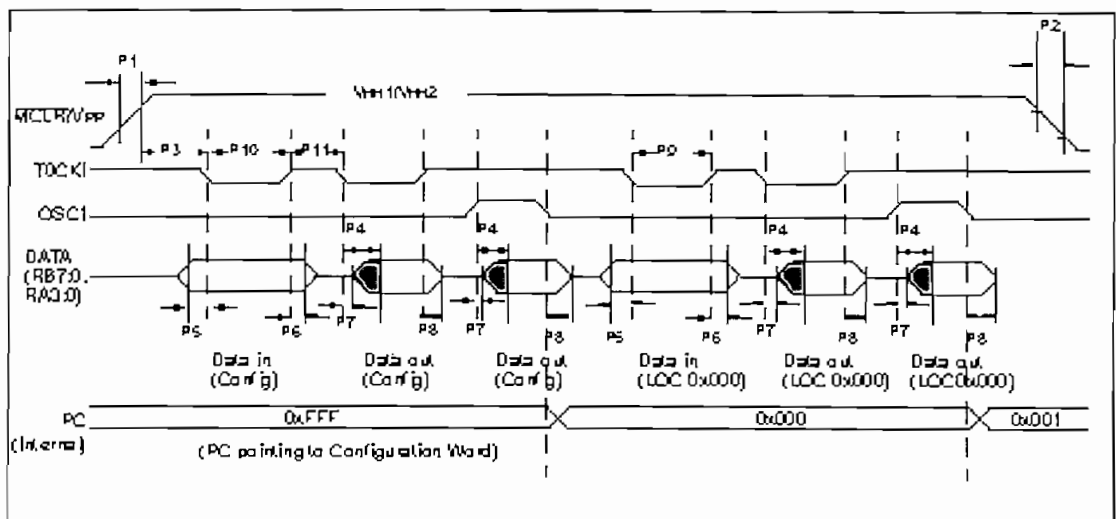


Figura 2.5 Diagrama de tiempos en el modo de programación/verificación

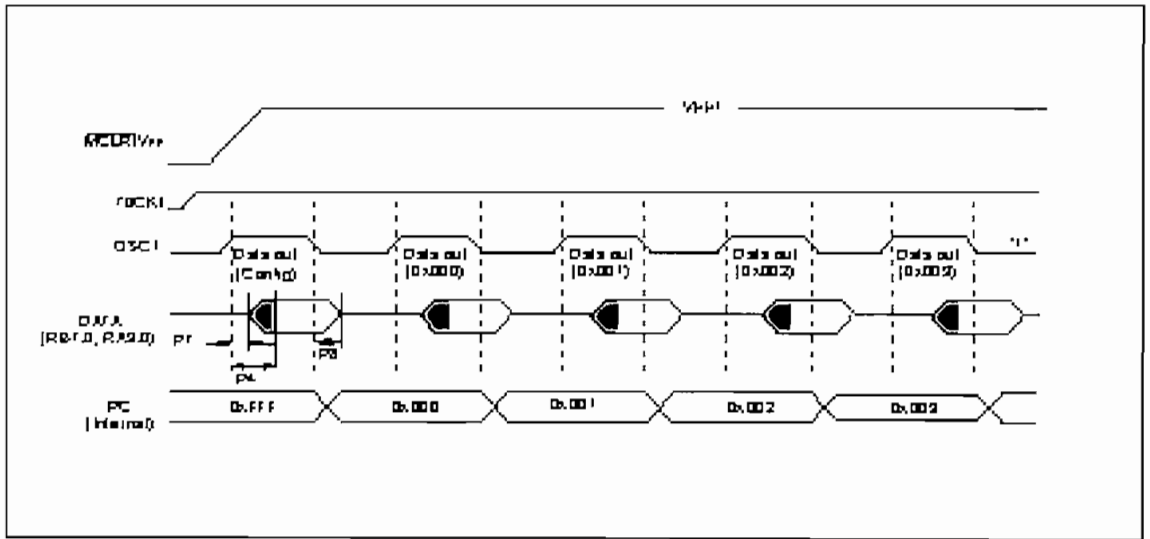


Figura 2.6 Diagrama de Tiempos en el modo de verificación

CAPITULO III

DISEÑO DEL HARDWARE

3.1 REQUERIMIENTOS GENERALES

Para la elaboración del proyecto planteado se deben tomar en cuenta los siguientes requerimientos:

- 1.- Se requiere de fuentes de alimentación tanto para obtener el voltaje de programación como para polarizar los diferentes elementos que conforman el programador.
- 2.- Se necesita una interfase de comunicación entre el programador y el computador.
- 3.- Señales de control, las cuales permitirán una programación óptima .

- 1.- Las fuentes de energía requeridas para el prototipo son:

V_{PP} que es el voltaje de programación, puede ser flexible entre 13.0V y 13.25V, este no debe exceder más de 14V y con una corriente límite aproximada de 100mA para no dañar ningún pin del microcontrolador.

V_{DD} puede estar entre 2.0V y 6.5V, para nuestro caso se utilizará una fuente de 5V para la alimentación de los chips y elementos que conforman el prototipo.

- 2.- La interfase que se utilizará para la comunicación entre el programador y el computador será un interfase serial RS232, para tal efecto se deberá adecuar las señales del puerto del computador a niveles utilizados por el microcontrolador (niveles TTL) y viceversa.

3.- Se utilizarán señales de control para determinar que elemento se va a programar, para la generación de las direcciones de la memoria EPROM, habilitación del modo de programación/verificación, así como también para el control de voltajes aplicados al elemento a ser programado.

Tanto los microcontroladores de la familia PIC16C5X de 18 pines como los de 28 pines utilizan los mismos pines para la programación y verificación, por tanto, mediante el control de los pines: TOCK1, MCLR, OSC1, RA0....RA3, RB0.....RB7, VDD Y VSS como se indica en la Tabla 2.1, será posible realizar las tareas de programación y verificación antes mencionadas.

Una ventaja de estos microcontroladores es que debido a su estructura solo se requiere de una sola línea para generar las direcciones de la memoria EPROM, esto a través del Pin OSC1 el mismo que al cambiar de estado de un nivel alto a un nivel bajo permite que se incremente la dirección en la memoria EPROM.

Por tanto para programar los microcontroladores de la familia PIC16C5X se requiere básicamente de 17 líneas las mismas que se distribuyen de la siguiente manera: 12 líneas para poder grabar y leer los datos de la memoria, 2 líneas para el control de la programación y 3 líneas para el control de las fuentes de polarización.

3.2 DIAGRAMA DE BLOQUES

En la Figura 3.1 se presenta el diagrama de bloques del programador, en el cual se ve claramente identificados tres bloques que son: la interfase serial entre el computador y el programador, la sección de control que es comandada por un microcontrolador y las fuentes de voltaje reguladas para la polarización y programación.

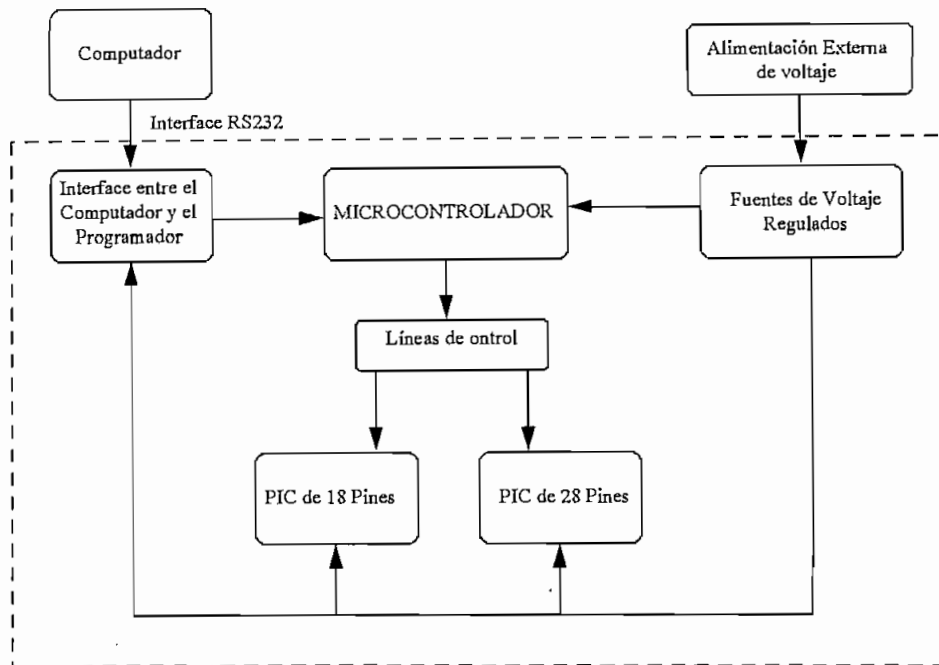


Figura 3.1 Diagrama de Bloques del Programador

3.3 DISEÑO CIRCUITAL

Como el programador es un equipo que debe cumplir con las tareas de escritura/lectura en memorias, controlar los voltajes aplicados a los pines de los microcontroladores a programar, así como también generar las direcciones en donde empieza la programación, se utilizará el microcontrolador INTEL 8751, el mismo que contendrá en su memoria EPROM el programa monitor para realizar las tareas antes indicadas.

El microcontrolador dispone de 4 puertos bidireccionales de 8 líneas cada uno, teniendo en total 32 líneas disponibles; este número de líneas es suficiente para la aplicación que se desea implementar.

Para nuestro caso se utilizarán 23 líneas, de las cuales 16 serán para grabar/leer datos desde la memoria a ser programada así como también para la generación de las señales que permitan programar el dispositivo, 5 se utilizaran para controlar los voltajes necesarios para la programación/verificación y las dos últimas líneas se usarán para la

comunicación serial, quedando 13 líneas disponibles las cuales se las puede utilizar para señalización.

El microcontrolador 8751 opera con un oscilador externo, el mismo que se implementa utilizando un cristal de frecuencia 7.15MHz, además se utiliza las líneas de comunicación RX y TX para la comunicación serial.

En la Tabla 3.1 se presentan los requerimientos de pines del microcontrolador 8751 que se utilizan para la programación de los microcontroladores de la familia PIC16C5X tanto para 28 pines como para 18 pines.

MICROCONTROLADOR 8751		MICROCONTROLADORES PIC DE 28 PINES		MICROCONTROLADORES PIC DE 18 PINES	
PIN	NOMBRE	PIN	NOMBRE	PIN	NOMBRE
1	P1.0	28	Habilita Vpp (MCLR)	--	---
2	P1.1	--	---	4	Habilita Vpp (MCLR)
3	P1.2	--	Habilita línea P0.3 - RA3	--	Desabilita línea P0.3 - RA3
4	P1.3	--	---	5	Conecto tierra
5	P1.4	2	Habilito VCC	14	Habilito VCC
6	P1.5	27	Genera Direcciones de EPROM (OSC1)	16	Genera Direcciones de EPROM (OSC1)
39	P0.0	6	RA0	1	RA2
38	P0.1	7	RA1	2	RA1
37	P0.2	8	RA2	3	TOCK1
36	P0.3	9	RA3	4	MCLR
35	P0.4	10	RB0	5	VSS
34	P0.5	11	RB1	6	RB0
33	P0.6	12	RB2	7	RB1
32	P0.7	13	RB3	8	RB2
21	P2.0	14	RB4	9	RB3
22	P2.1	15	RB5	10	RB4
23	P2.3	16	RB6	11	RB5
24	P2.4	17	RB7	12	RB6
25	P2.5	1	TOCK1	13	RB7
26	P2.6	--	---	17	RA0
27	P2.7	--	---	18	RA1
MICROCONTROLADOR		MAX232			
11	TXD	11	Pin de Transmisión uno (T1IN)		
10	RXD	12	Pin de Recepción uno (R1IN)		

Tabla 3.1 Conexión entre el microcontrolador 8751 y el microcontrolador a ser programado

Para el microcontrolador 8751 se implemento un circuito de "reset", que facilita

reinicializar el sistema en cualquier momento y ante cualquier situación. Se utilizó un transistor PNP al mismo que se lo hace trabajar en la zona de saturación y corte. Para asegurar que este trabaje en dichas zonas se colocó una resistencia de 1K en la base del mismo como se indica en la Figura 3.2 de esta manera se elimina el rebote producido en el pulsador. En el puerto P0 del microcontrolador se colocó resistencias de pull-up de 10K.

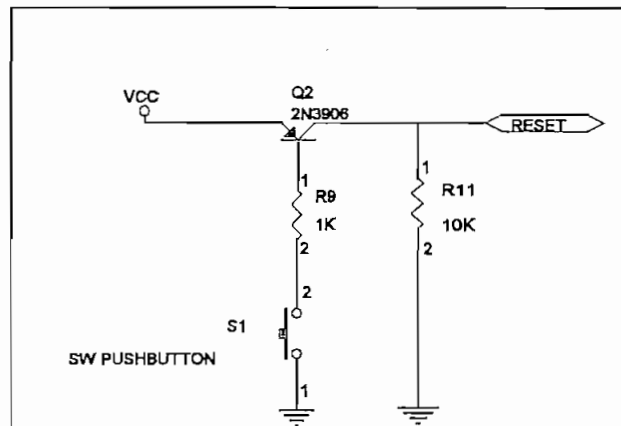


Figura 3.2 Circuito de Reset

3.3.1 INTERFASE PARA LA COMUNICACION ENTRE EL COMPUTADOR Y EL PROGRAMADOR

Para la comunicación con el computador, se utilizó el circuito integrado MAX232 el mismo que permite acoplar los niveles de voltaje de las señales que nos entrega el computador a niveles TTL utilizadas por el microcontrolador y viceversa, en la Figura 3.3 se muestra el esquema de como esta conectado este elemento así como también de los elementos adicionales dados por el fabricante que éste requiere para su adecuado funcionamiento.

Los condensadores C3, C4, C5, C6 y C7 tienen un valor de $10\mu\text{F}/20\text{V}$ este valor es el que recomienda el fabricante, en el ANEXO 2 se presenta las características del MAX232.

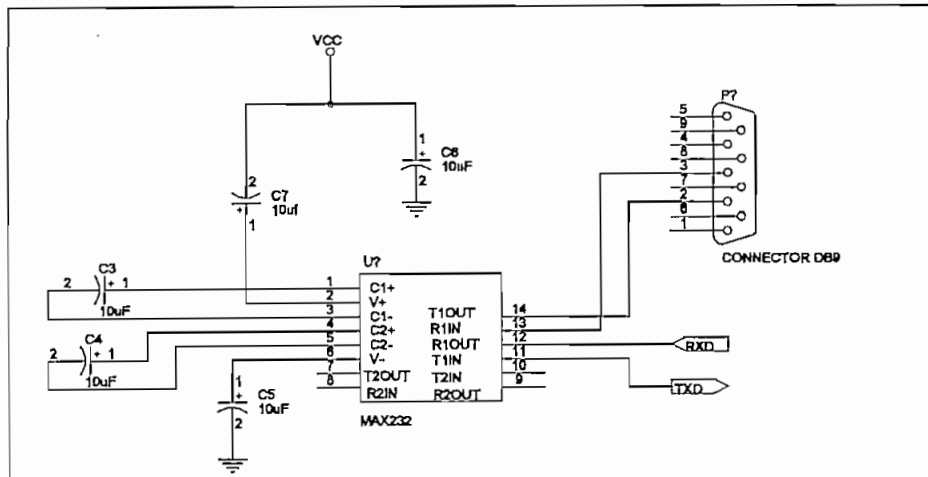


Figura 3.3 Circuito que realiza la Interfase para la Comunicación Serial

3.3.2 CONTROL DE LOS VOLTAJES APLICADOS AL CHIP A SER PROGRAMADO

Para poder tener un control sobre el voltaje de programación aplicado al chip a ser programado así como también sobre el voltaje de polarización, se utiliza el chip U4 74HC4066N (que es un switch bidireccional que permite transmitir o multiplexar señales analógicas como digitales) conjuntamente con el circuito integrado 74LS03N el mismo que es un integrado que contiene cuatro compuertas NAND open-colector. En la Figura 3.4 se muestra el esquema del circuito utilizado.

Debido a que se utilizan compuertas open-colector se requiere de resistencias " Pull-Up" las mismas que se conectan a la salida de cada compuerta. Como el nivel de corriente de salida es de 2mA y el voltaje que cae en la resistencia es de 13V el valor mínimo de la resistencia pull-up será de 6500Ω , para la presente aplicación se conectarán resistencias de $10\text{ K}\Omega$, con esto se limita la corriente que circula por la compuerta a 1.3mA, protegiendo de esta manera a la compuerta.

El funcionamiento del circuito es el siguiente: al tener un nivel bajo (0 lógico) en la entrada de la compuerta, se tiene un nivel alto a la salida de la misma; esto permite que el voltaje de 13V sea aplicado a pin de control del chip U4A (74HC4066). Esto hace que se

conecten las señales que se encuentran conectados en los pines A y B del chip y si se aplica un 1 lógico a la entrada de la compuerta NAND se tendrá a la salida 0 lógico; por ende, el voltaje aplicado a los pines de control del chip U4B o C (74HC4066) será de 0V desabilitando la conexión entre los pines A y B.

Además de los circuitos integrados mencionados anteriormente se utiliza el chip UA5 (7406) el cual es un buffer de corriente. Este elemento permite acoplar la señal que entrega el microcontrolador al transistor Q1 (PNP 2N3906) al mismo que se lo hace trabajar en corte/saturación como un switch, los valores de las resistencias R5 y R6 aseguran que este trabaje en la zona de corte y saturación, permitiendo de esta manera tener un control sobre el voltaje de polarización VDD aplicado al elemento que se va a programar. Por tanto, dependerá de la señal que se tenga a la entrada el buffer de corriente para tener habilitado o desabilitado VDD.

El transistor Q1 sirve además como elemento de protección para el circuito de programación.

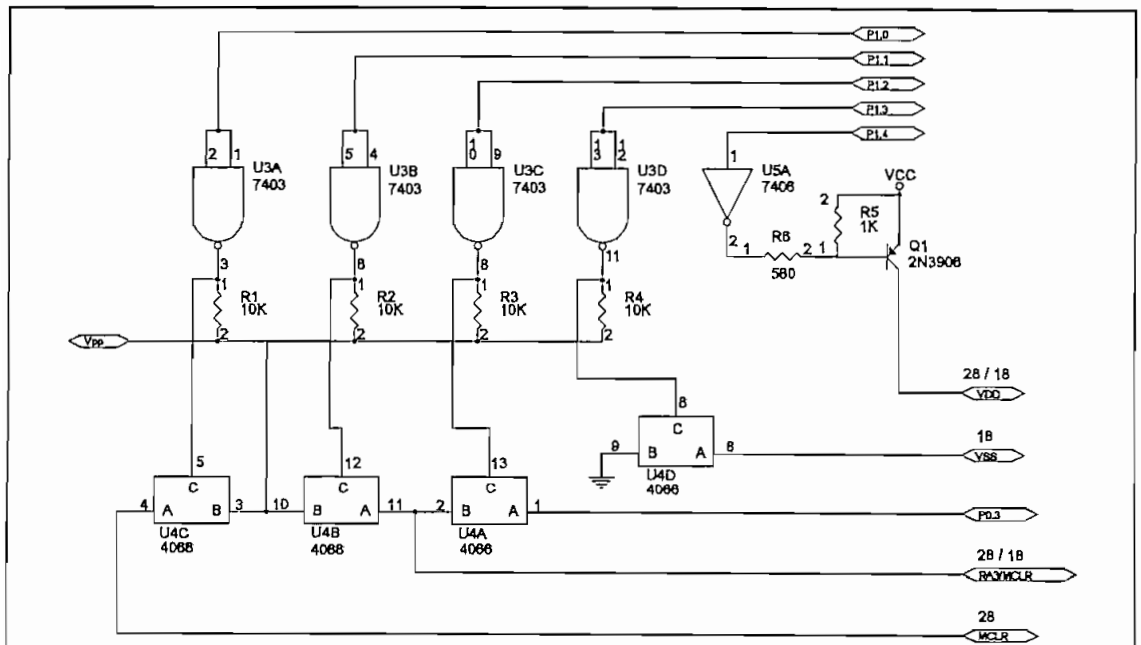


Figura 3.4 Circuito de Control

A la salida de Q1, esto es en el colector, se tiene un circuito de señalización el

mismo que indicará el momento en que se esté aplicando VDD a los microcontroladores

En la Tabla 3.2 se presenta como se encuentran codificadas las señales de control las mismas que dependiendo de su valor nos indicaran que elemento se va a programar.

P1.0	P1.1	P1.2	P1.3	P1.4	P1.5	ESTADO
1	1	1	1	0	0	Condición inicial
0	1	0	0	1	0	Programación PIC de 28 pines
1	0	1	0	1	0	Programación PIC de 18 pines

Tabla 3.2 Configuración de las Líneas de Control

Se llama condición inicial cuando no este habilitado ninguna señal hacia el elemento que se va a programar.

P1.0: Permite habilitar o deshabilitar el voltaje de programación para el PIC de 28 pines

P1.1: Habilita o deshabilita el voltaje de programación del PIC de 18 pines.

P1.2: Habilita al pin 9 del Sócalo de programación como línea de datos cuando esta en 0 lógico y cuando esta en 1 lógico este pin es línea de entrada para el voltaje de programación

P1.3: Habilita tierra para el PIC de 18 pines

P1.4: Habilita o deshabilita VDD al elemento a ser programado

P1.5 Es aplicado al pin OSC1 de los microcontroladores de 28 y 18 pines, el mismo que permite incrementar las direcciones en la memoria EPROM de los microcontroladores a ser programados.

Como la señal entregada por P1.5 es aplicada al mismo tiempo en los pines 27 (OSC1) y 21 (RC4) en los micros de 28 pines, esta señal aplicada a RC4 no influye en la programación de estos micros.

P0.2: Sirve como línea de datos para el PIC de 28 pines y como señal para habilitar la programación/verificación para el PIC de 18 pines.

P2.4: Al igual que P0.2 servirá para habilitar la señal de programación/verificación para el PIC de 28 pines y como línea de datos para el PIC de 18 pines dependiendo del micro que se haya seleccionado.

3.3.3 FUENTES DE VOLTAJE REGULADOS

Para este prototipo se requiere de dos fuentes de tensión regulada que cumplirán los siguientes propósitos:

- a) Para la polarización de los chips
- b) Para obtener el voltaje de programación

En el numeral 3.1 se dedujeron las especificaciones de las fuentes de voltaje: el sistema utilizará dos fuentes una de 5V y otra de 13.25V.

En la Figura 3.5 se muestra el diagrama de las fuentes que se van a utilizar. Para el voltaje de programación la fuente requerida debe ser capaz de entregar una corriente de 100mA y un voltaje de 13.25V. Para este propósito se utilizará el regulador LM317 (ECC956), este elemento es un regulador de voltaje que puede dar un voltaje de salida de 1.2V a 37V, con una corriente de 1.5A. El voltaje de salida viene dado por la relación siguiente:

$$V_{OUT} = 1.25 \left(1 + \frac{R2}{R1} \right)$$

Para el caso presente se requiere tener una tensión de salida de 13.25V los valores de R2 y R1 serían:

$$\text{Si } R2 = 270\Omega$$

$$R1 = 28.125\Omega$$

Como en el mercado no existen resistencias de 28.125Ω, se coló un potenciómetro de precisión de 1KΩ, el cual está calibrado para obtener el voltaje deseado.

Para la alimentación a los diferentes chips, se utilizará una fuente de 5V. El esquema del circuito se presenta en la Figura 3.5. Para obtener el voltaje de 5V se usará un regulador de voltaje el mismo que debe ser capaz de entregar la corriente que el circuito requiera. Para determinar la corriente que requiere el prototipo se debe sacar un listado de los elementos más representativos del circuito con sus respectivos consumos de corriente y sacar un total. En la Tabla 3.2 se presenta los elementos que conforman el prototipo con sus respectivos consumos de corriente

Elemento	Corriente (mA)
1 Led	10
Circuito de reset	4
8051	100
Elemento a programar PIC16C5X	100
74HC4066N	40
MAX232	40
74LS03N	40
74LS06	40
Total	374 mA

Tabla 3.2 Resumen del consumo de corriente del programador

Para la fuente de 5V se utilizó el regulador de voltaje LM7805 (ECG960) que permite tener una tensión de salida de 5V y una corriente de 1A.

A la entrada del regulador se coló un capacitor de 1000 μ F/25V para filtrar el voltaje de entrada al regulador, mientras que a la salida se usó un capacitor de 330 μ F/25V. Un capacitor de 0.1 μ F, fue añadido para eliminar ruidos. Para el voltaje de entrada se utilizó un transformador que permite reducir el voltaje de 110V AC a 18 V. El transformador se seleccionó con una capacidad de corriente de 2 amperios.

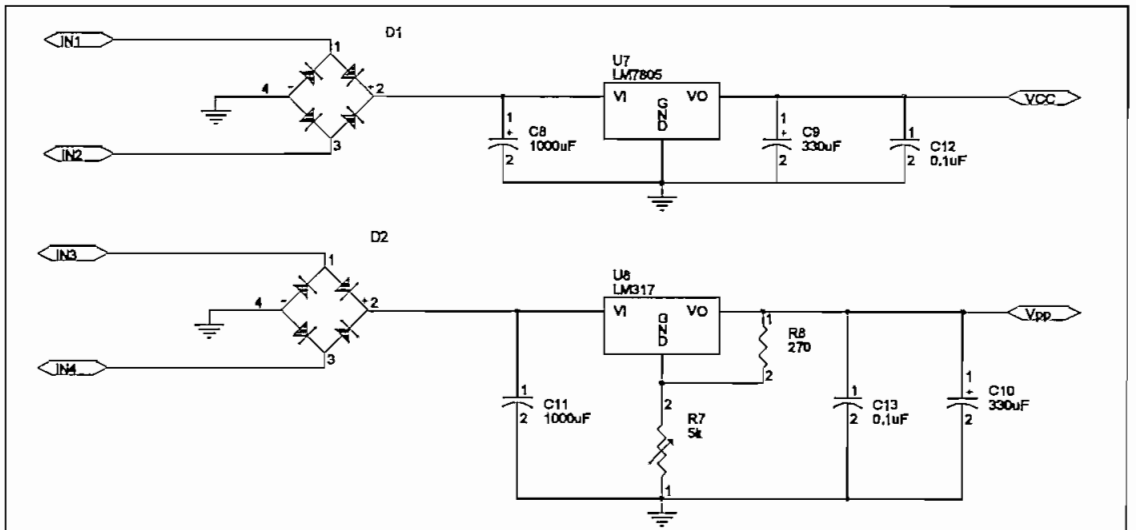


Figura 3.4 Fuentes de Alimentación de VDD (+5 V) y de Vpp (+13 V)

Para tener un mejor desacople entre la señal de Vcc y el ruido generado se colocó condensadores C14, C15, C16, C17, C18 de 0.1µF en los pines de Vcc de los chips .

Para la fabricación del circuito impreso se utilizará el programa Tango el mismo que nos permitirá realizar el ruteo del circuito a implementar

En el ANEXO 2 se presenta las características, de los chips utilizados en el diseño del programador, así como también los diagramas de los circuitos impresos.

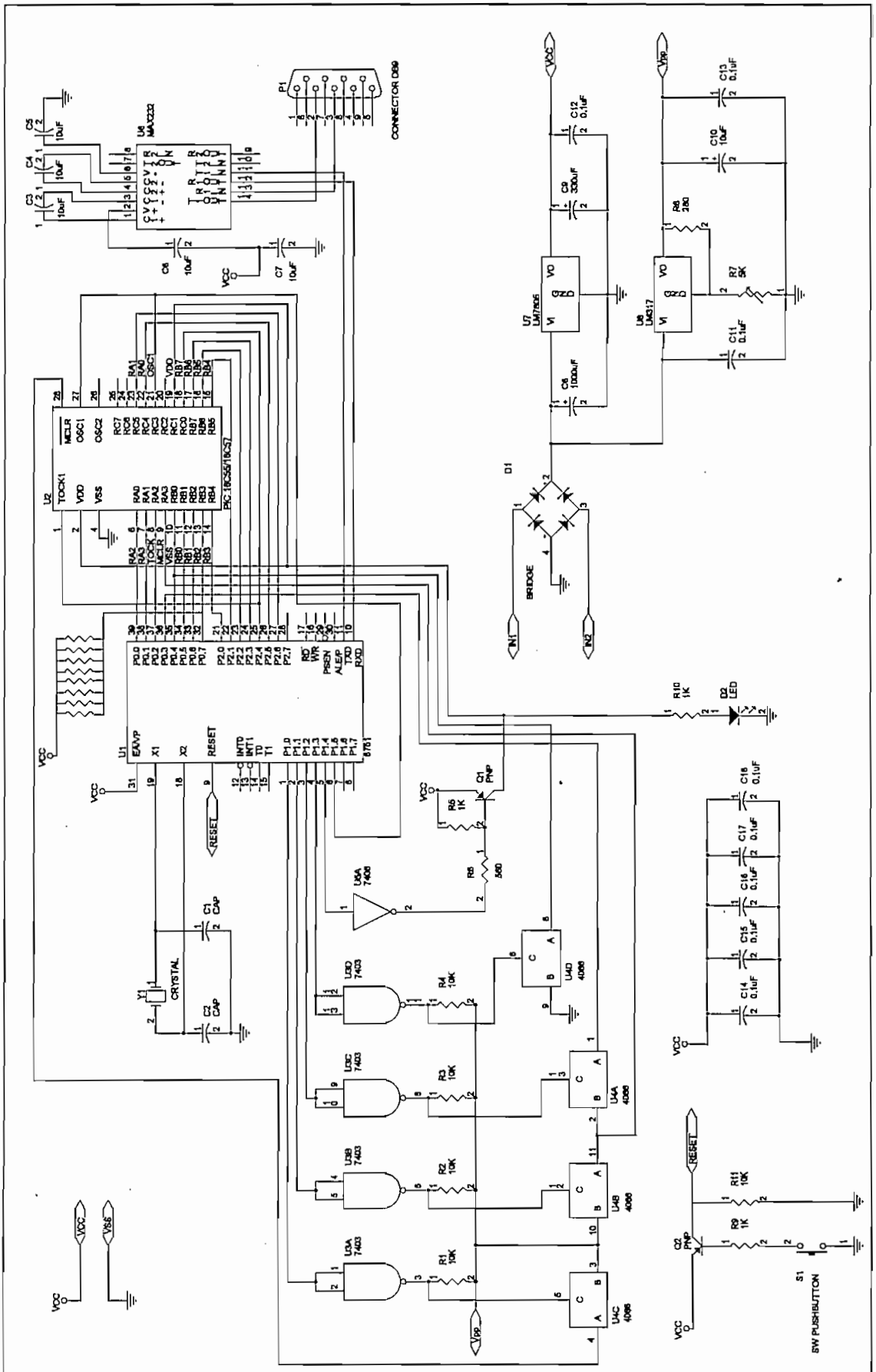


Figura 3.5 Diagrama del Programador

CAPITULO IV

DISEÑO DEL SOFTWARE

En este capítulo se describen los pasos para poder desarrollar el programa para el programador así como también el programa para el manejo del programador a través del computador.

En el desarrollo de ambos programas se tomaron en consideración características que ofrecen los programadores comerciales, particularmente aquellas opciones que buscan facilitar el manejo del programa por parte del usuario. También se buscó dotar a los programas de las opciones típicas que un usuario espera encontrar en la actualidad en un programa de computación, previamente adaptándolas donde era pertinente de acuerdo a las características propias del presente sistema.

El programador permitirá la programación de los microcontroladores CMOS de la familia PIC16C5X, pero en el desarrollo del software se debe considerar la posibilidad de una versión futura, que permitirá la programación de los microcontroladores PIC16CX, PIC16C71 y PIC16C84.

4.1 DESCRIPCION GENERAL

Como se mencionó, el software implementado se compone de dos partes el software para el programador y el software para el computador.

Para el software para el microcontrolador, se identifica que este debería realizar básicamente dos tareas que son:

- 1) Lectura de la memoria de los microcontroladores PIC16C5X; y
- 2) Programación/verificación de los microcontroladores PIC16C5X.

Los datos que reciba el programador servirían para poder determinar:

1. El elemento que se va a programar y por ende el tamaño de la memoria del microcontrolador.
2. Las líneas que se deben activar para poder alimentar correctamente a los PIC a programar; así como también el instante en que se requiere el pulso de programación.
3. La operación que se va a realizar: si es solo lectura de la memoria o la programación de la misma.

Como la comunicación se lo realiza por vía serial, se determinó que ésta debería ser en ambos sentidos, pero el que debe tomar la iniciativa es el computador quien siempre mandará un comando o dato en espera de una respuesta del programador.

Se determinó también que, al iniciar la aplicación, el programa debe chequear que exista comunicación entre el computador y el programador. Si no existe esta comunicación deberá desplegar un mensaje de error en la pantalla del computador.

4.2 OPCIONES DEL PROGRAMADOR

Tomando como guía los programas de programadores que existen en el mercado, se determinó que el programa a implementarse debería tener un barra de menú con las siguientes opciones: Archivo, Editar, Programar, Opciones y Ayuda, las mismas que se detallan a continuación:

Para el menú Archivo se pensó en las siguientes opciones:

■ ABRIR UN ARCHIVO *.HEX

Esta opción es necesaria para recuperar y cargar un archivo.HEX en el buffer de memoria.

- **GUARDAR**

Esta opción permitirá guardar el contenido del buffer de memoria en un archivo.

El formato con que es generado el archivo correspondería al de INTEL.

- **GUARDAR COMO**

Al igual que la opción anterior permitiría guardar el contenido del buffer de memoria en un archivo, en formato INTEL, pero con la diferencia en que el usuario podría guardar el archivo con otro nombre o seleccionar uno que ya existe.

- **SALIR**

Este comando permitirá salir del programa.

Para el menú Editar se pensó las opciones siguientes:

- **EDITAR BUFFER**

Este comando permitiría editar cualquier localidad de memoria. Si no se realiza ningún cambio el dato anterior permanecería.

- **BORRAR BUFFER**

Este comando permitiría borrar el buffer de memoria, escribiendo en las localidades de memoria el valor FFF en hexadecimal. Así, si leyéramos una memoria en blanco nos debería dar el valor de FFF en todas sus localidades de memoria.

- **BORRAR BITS DE CONFIGURACION**

Esta opción es muy útil para poner los valores por defecto en los bits de

configuración del microcontrolador, esto es: el tipo de oscilador en RC, con la opción del watchdog timer habilitado y el código de protección deshabilitado.

Para el menú Programar se pensó en las opciones siguientes:

■ **PROGRAMAR**

Este comando es el que permitiría programar el microcontrolador seleccionado. Debe observarse que para la programación, el microcontrolador debe tener su memoria en blanco o borrada. Además, es indispensable primero cargar en el buffer de memoria el archivo .HEX, que es el que contiene el programa a ser implementado.

Cada dato que se programe debe ser verificado, con esto se estaría asegurando una programación óptima.

Si la memoria del microcontrolador no se encuentra en blanco la programación no deberá continuar, dando un mensaje de programación no O.K.

■ **VERIFICAR**

Esta opción deberá leer el contenido de la memoria del microcontrolador y compararla con el contenido del buffer de memoria, el mismo que debe contener el archivo.HEX. Si el contenido no es igual deberá indicar la localidad en la cual los datos no coinciden. Siempre se considerará como dato correcto al que se encuentra en el buffer de memoria; al que proviene de la memoria del microcontrolador, es el que debe considerarse como sujeto de error.

■ **VERIFICAR SI ESTA EN BLANCO LA MEMORIA**

Esta opción permitirá chequear si la memoria se encuentra en blanco o no, para lo

cual se hará una lectura de la memoria y a cada localidad deberá compararla con el valor FFF(en hexadecimal) y si en la comparación resulta que los valores son diferentes se deberá presentar en pantalla un mensaje reflejando tal condición "LA MEMORIA NO ESTA BORRADA", por ejemplo. En caso contrario se presentaría en pantalla el mensaje siguiente "LA MEMORIA ESTA EN BLANCO".

■ **LEER MEMORIA**

Esta opción permitirá leer el contenido de la memoria del microcontrolador y presentarla en pantalla.

Para el menú Opciones se tiene los siguientes comandos:

■ **ESTABLECER COMUNICACION**

Con esta opción se podría determinar si existe comunicación entre el computador y el programador.

■ **SELECCIONAR PORTICO DE COMUNICACIONES**

Con esta opción se daría la posibilidad de seleccionar el p^ortico de comunicaciones serial que se va a utilizar en el computador para la comunicación.

■ **CONFIGURACION DE BITS**

Como los microcontroladores PIC16C5X tienen la posibilidad de trabajar con diferentes tipos de osciladores, se debe especificar que tipo de oscilador va a utilizar, así como también si va a estar habilitado o no el watchdog timer o el código de protección.

■ **SELECCIONAR MICROCONTROLADOR**

Esta opción es lógica ya que permitirá al usuario seleccionar el tipo de microcontrolador que desee programar, verificar o leer.

Para el menú Ayuda se tendrá las opciones siguientes:

■ **ACERCA DE**

Mostrará la información acerca de los derechos de autor, fecha de producción y versión del programador.

■ **AYUDA**

Esta opción servirá para consultar la operación de cada uno de los comandos que se desarrollen en el programador

4.3 DIAGRAMAS DE FLUJO

Para poder implementar las opciones antes mencionadas se desarrollaron los diagramas de flujo que se presentan a continuación, en los cuales se indica el orden lógico detrás de cada opción, tanto para el programador como para el computador.

DIAGRAMA DE FLUJO DEL PROGRAMA PARA EL PROGRAMADOR

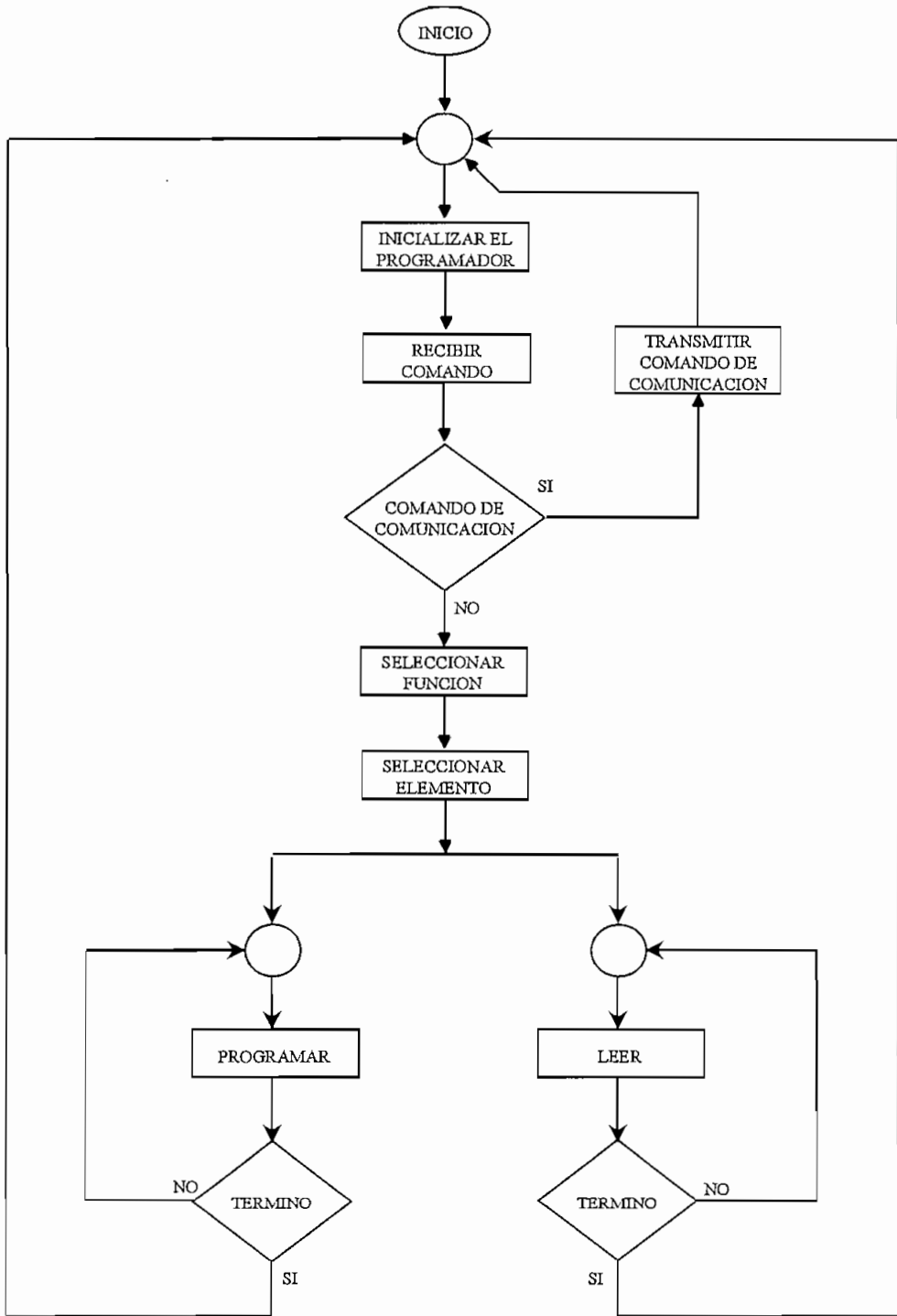


DIAGRAMA 1

DIAGRAMA DE FLUJO DEL PROGRAMA EN VISUAL BASIC

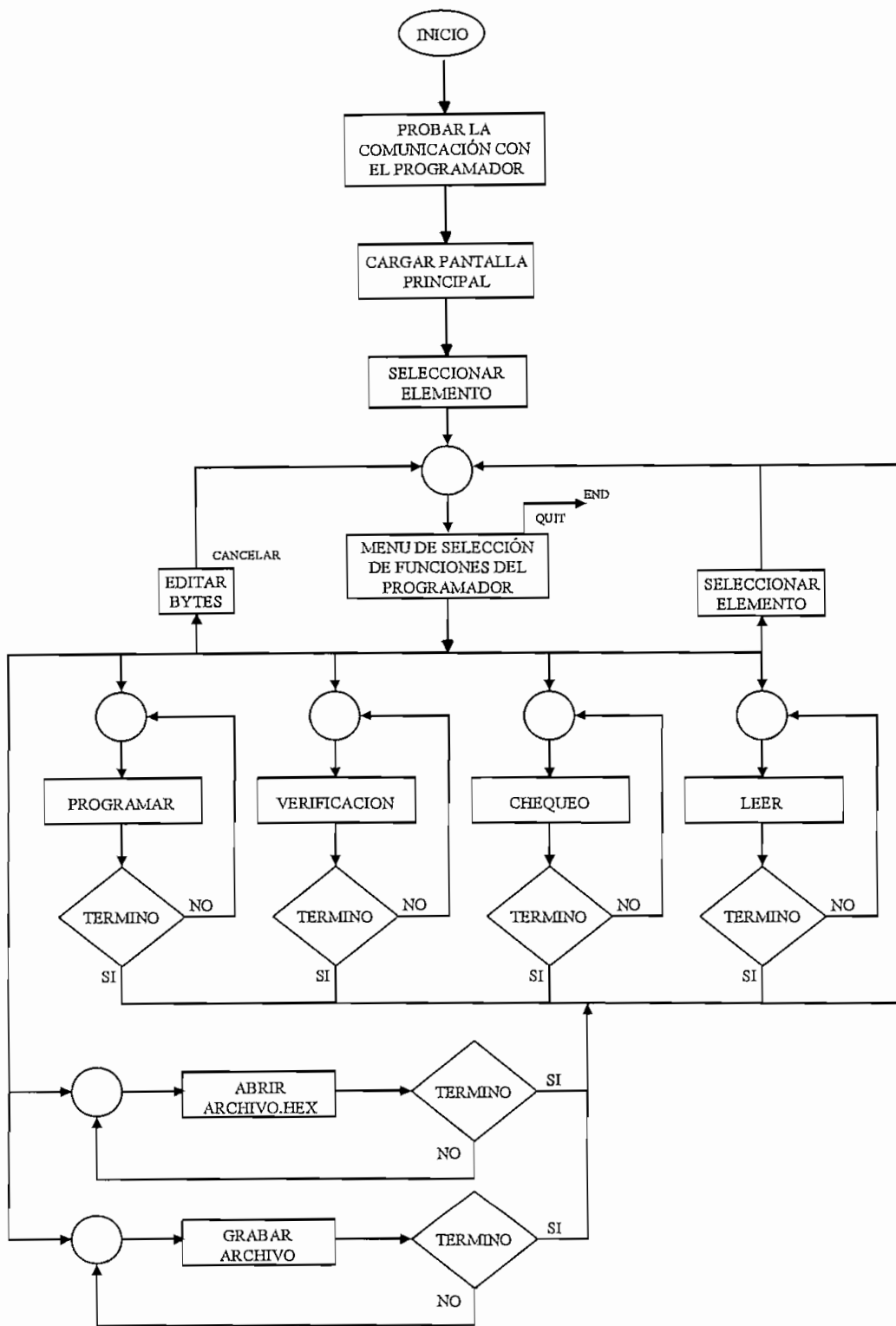


DIAGRAMA 2

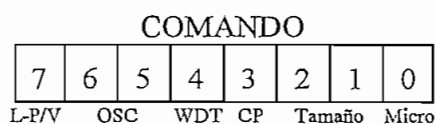
4.4 DESARROLLO DEL SOFTWARE

Antes de proceder al desarrollo de los algoritmos que produzcan los menús y opciones descritas anteriormente, es necesario indicar que en el desarrollo del código se tomaron en cuenta todas las normas y recomendaciones que existen en la industria del software. Por ejemplo, el programa deberá tener todas las seguridades debidas que le permita controlar el ingreso errado de datos por parte del usuario. El programa no deberá permitir que una acción considerada de riesgo se ejecute sin antes pedir una reconfirmación por parte del operador, etc.

El programa deberá permitir que la aplicación sea de fácil utilización, " amigable" al usuario, además de tener una presentación gráfica que permita al operador interactuar con el programador, en forma lógica y fácil.

Como el programador de los PICS debe saber que tarea va a realizar, así como también con que elemento va a trabajar se comienza por enviar desde el computador un comando, en el que se codifica toda la información que le permite conocer al programador si se va a realizar una operación de lectura o programación, con que elemento se va a trabajar, el tamaño de la memoria, si está habilitado el watchdog timer, si está habilitado el código de protección y la configuración del tipo de oscilador con el que va a trabajar el microcontrolador.

A continuación se presenta un esquema de los bits que conforma el comando que es enviado desde el computador al programador, con el propósito descrito.



Bit 0: Tipo de Micro: Si es 0, Micro de 18 pines.

Si es 1, Micro de 28 pines

Bit 1 y 2: Tamaño:

Bit 2	Bit 1	Tamaño
0	0	512
0	1	1 K
1	0	2 K
1	1	4 K

Bit 3: Código de protección: En 1 deshabilitado y en 0 habilitado

Bit 4: Watchdog Timer: En 0 deshabilitado y en 1 habilitado

Bit 5 y 6: Tipo de Oscilador:

Bit 6	Bit 5	Oscilador
0	0	LP
0	1	XT
1	0	HS
1	1	RC

Bit 7: Tipo de Operación: Si es 0 se va a realizar una operación de lectura

Si es 1 se va a realizar la programación del microcontrolador..

En el microcontrolador se utilizan los siguientes registros:

R7: Para el byte más significativo de la dirección en donde empieza la programación

R6: Para el byte menos significativo de la dirección en donde empieza la programación.

R3 y R4: Se utilizan para producir los retardos de los pulsos de programación.

Se emplean además las siguientes localidades de memoria:

7FH: En esta localidad se almacena el dato que es enviado por el computador.

7EH: Se utilizará para almacenar el dato recibido del computador a través de la localidad 7FH a la cual se la denominará comando.

7DH: En esta localidad se guarda el valor del tamaño de la memoria del microcontrolador a ser programado.

Adicionalmente se emplean los siguientes bits como banderas:

BSERIAL: Esta bandera se activa si el microcontrolador recibe alguna interrupción serial, por lo regular se utiliza para determinar si el programador recibió un dato del computador.

MICRO: Sirve para determinar que micro se va a programar, si es un micro de 18 pines esta bandera estará en cero y si es un micro de 28 pines esta bandera se pondrá en 1 lógico.

RESETEO: Permite inicializar el programador cuando se a producido algún error.

La Tabla 4.1 muestra como se encuentra codificado el tamaño de la memoria de los microcontroladores.

Tamaño	Valor en HEX	Valor en Decimal	Valor de Tamaño
512	01FF H	511	0200 H
1 K	03FF H	1023	0400 H
2 K	07FF H	2047	0800H

Tabla 4.1

En la localidad TAMAÑO se almacenará el valor del byte más significativo de la dirección, esto es 02H, 04H o 08H, que corresponde a memorias de 512, 1K y 2K respectivamente.

El objetivo del software para el computador es permitir tener un medio de enlace entre el usuario y el programador; por lo mismo, por las características que tiene dicho lenguaje de desarrollo, el software para el computador se lo desarrollo en Visual Basic .

A continuación se describen los algoritmos desarrollados en lenguaje estructurado. El código que se obtuvo se adjunta en el ANEXO 3.

En el algoritmo utilizado para la implementación del programa para el programador se tiene los siguientes procedimientos:

En el procedimiento de inicialización se tiene las siguientes tareas:

Inicialización

Asignar a las etiquetas las respectivas direcciones.
Programar el timer 1 para que pueda realizar la comunicación serial.
Inicializar el socket de programación.

Fin de Tarea

A <----> SBUF

Esperar que el computador envíe dato.
El dato recibido muevo al acumulador.
Verifico el dato recibido con dato esperado.
Retransmito el dato recibido si su verificación es correcta.

Fin de Tarea

A <----- SBUF

Esperar dato del computador.
Llamar a proceso de lectura o de programación.

Fin de Tarea

En el procedimiento de lectura se tiene las siguientes tareas:

Selección del elemento a ser leído

Determinar si el elemento a ser leído es de 18 o 28 pines.
Determinar el tamaño de la memoria.

Fin de Tarea

Configurar los pines del ZIF de acuerdo al elemento seleccionado

Habilitar las señales de control en los pines respectivos.

Fin de Tarea

Ir a la rutina particular de lectura e inicialización de la dirección con 0000H

Polarizar el elemento a ser leído.
Inicializar Contador de dirección.
Habilitar voltaje de programación.

Fin de Tarea

Leer dato

Sacar dato a los puertos P0 y P2.
Formar dato en el acumulador, si el elemento es de 18 pines.

Fin de Tarea

Envío de dato

Transmitir LSB del dato.
Esperar confirmación del computador.
Transmitir MSB del dato.
Esperar confirmación del computador.

Fin de Tarea

Incrementar la dirección

Generar pulso en el pin OSC1 del PIC.
Terminar lectura si dirección es igual al registro tamaño, si no volver a leer y transmitir dato.

Fin de Tarea

Regresar el ZIF a condiciones iniciales

- Quitar polarización del elemento a ser leído.
- Regresar los pines del socket de programación a su estado inicial.
- Esperar dato del computador.

Fin de Tarea

En el procedimiento de programación se tiene las siguientes tareas:

Selección del elemento a ser programado

- Determinar si el elemento a ser programado es de 18 o 28 pines.
- Determinar el tamaño de la memoria.

Fin de Tarea.

Configurar los pines del ZIF de acuerdo al elemento seleccionado.

- Habilitar las señales de control en los pines respectivos.

Fin de Tarea.

Ir a la rutina particular de programación.

- Polarizar el elemento a ser programado.
- Habilitar voltaje de programación.
- Inicializar contador de direcciones.
- De acuerdo al comando recibido programar los bits de configuración del micro y leer la configuración del micro para su transmisión.

Fin de Tarea

Recibir el número de datos a programar

- Recibir el número de datos a programar y guardar en el registro R5
- Si el número de datos a programar es igual a cero, regresar a esperar un nuevo comando de lo contrario retransmitir el dato para su validación.

Fin de Tarea

Recibir la dirección

- Recibir la dirección en donde va a empezar la programación y este valor guardar en los registros R7 y R6.
- Recibir el separador entre la dirección de inicio de la programación y los datos.

Fin de Tarea

Activar el voltaje de programación

- Aplico el voltaje de programación

Fin de Tarea

Sacar dirección

- Dependiendo de los valores que se tienen en los registros R6 y R7 se aplica pulsos en el pin OSC1 del micro a ser programado para llegar a la dirección en donde empieza la programación .

Fin de Tarea

Sacar el dato a ser programado

- Recibir el byte LSB del dato y sacar a los pines respectivos de los pódicos P0 y P2.
- Recibir el byte MSB del dato y sacar a los pines del pódico P2.

Fin de Tarea

Aplicar pulso de programación

- Aplicar el pulso de programación de 10 useg.

Fin de Tarea

Verificación y envío del dato

- Entrar al modo de verificación para leer el dato programado

Sacar el dato a los p \acute{o} rticos P0 y P2 y transmitir al computador para su validaci3n.

Fin de Tarea

Incrementar la direcci3n

Programado y verificado el dato se aplica un pulso sobre el pin OSC1 del micro para acceder a la siguiente direcci3n a ser programada

Decremento el Registro R5

Si R5 es diferente de cero volver al procedimiento de sacar dato a ser programado

Fin de Tarea

Recepci3n y envi3 de checksum

Si R5 es igual a cero recibir el checksum y retransmitirlo

Deshabilitar el voltaje de programaci3n.

Desconectar VCC del ZIF de programaci3n.

Esperar dato del computador.

Fin de Tarea

Recepci3n del n \acute{u} mero de datos a programar

Si el n \acute{u} mero de datos a programar es cero:

Regresar el ZIF a condiciones iniciales

Esperar un nuevo comando.

Si el n \acute{u} mero de datos es diferente de cero:

R5 toma el nuevo valor de n \acute{u} mero de datos a programar

Regresar al procedimiento de recibir direcci3n de inicio de programaci3n.

Fin de Tarea

En el algoritmo utilizado para la implementaci3n del programa para el computador en Visual Basic se tienen los siguientes procedimientos:

Procedimiento para probar la comunicaci3n:

Open Com

Borrar el buffer de recepci3n.

Abrir p \acute{o} rtico de comunicaciones.

Si no existe ning \acute{u} n problema con el p \acute{o} rtico de comunicaciones continuar \acute{a} recibir dato enviado del programador .

Si el dato recibido es igual al enviado presentar el mensaje de comunicaci3n O.K. , caso contrario se mostrar \acute{a} el mensaje error de comunicaci3n y permitir \acute{a} seleccionar otro p \acute{o} rtico de comunicaciones.

Fin de Tarea

Como resultado del c3digo desarrollado, cada vez que exista un error en la comunicaci3n se presentar \acute{a} en pantalla la Figura 4.1, la misma que permite cambiar el p \acute{o} rtico de comunicaciones.

En el procedimiento para cargar la pantalla principal se tiene las siguientes tareas:

Cargar la pantalla principal

Descarga de la memoria de la pantalla de presentaci3n

Inicializaci3n de variables

Generaci3n de los controles y barra de men \acute{u} s

Presentaci3n de la grilla con el valor de FFF H en cada celda

Fin de Tarea

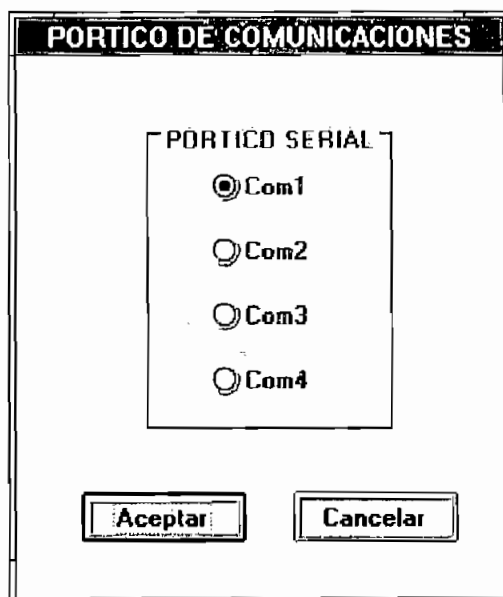


Figura 4.1 Selección del Pórtico Serial de Comunicaciones

Del código implementado se tiene la pantalla principal del programador, la misma que se presenta en la Figura 4.2

Abrir archivo a ser programado

Abrir archivo que se encuentra en la propiedad Filemane del cuadro de dialogo.
Si no existe ningún archivo indicar mediante un mensaje de error y salir del procedimiento.

Fin de Tarea

En el procedimiento para la programación se tiene las siguientes tareas:

Abrir archivo a ser programado

Abrir archivo que se encuentra en la propiedad Filemane del cuadro de dialogo.
Si no existe ningún archivo indicar mediante un mensaje de error y salir del procedimiento.

Fin de Tarea

Open Com

Borrar el buffer de recepción.
Abrir pórtico de comunicaciones.
Si no existe ningún problema con el pórtico de comunicaciones continuará con la programación , caso contrario se mostrará el mensaje seleccionar otro pórtico de comunicaciones .

Fin de Tarea

Envío de comando de programación

Dependiendo del elemento seleccionado se transmitirá el comando respectivo.

Fin de Tarea

Recibir configuración del micro

Esperar a que lleguen datos al buffer de recepción del computador.
Si no se recibe ningún dato o se excede de cierto tiempo se presentará el mensaje de error de programación.
El dato recibido se lo compara con el comando enviado (en el mismo se halla codificado la configuración del micro a ser programada).
Si la comparación resulta exitosa sigue con el procedimiento de programación, si no se presenta el

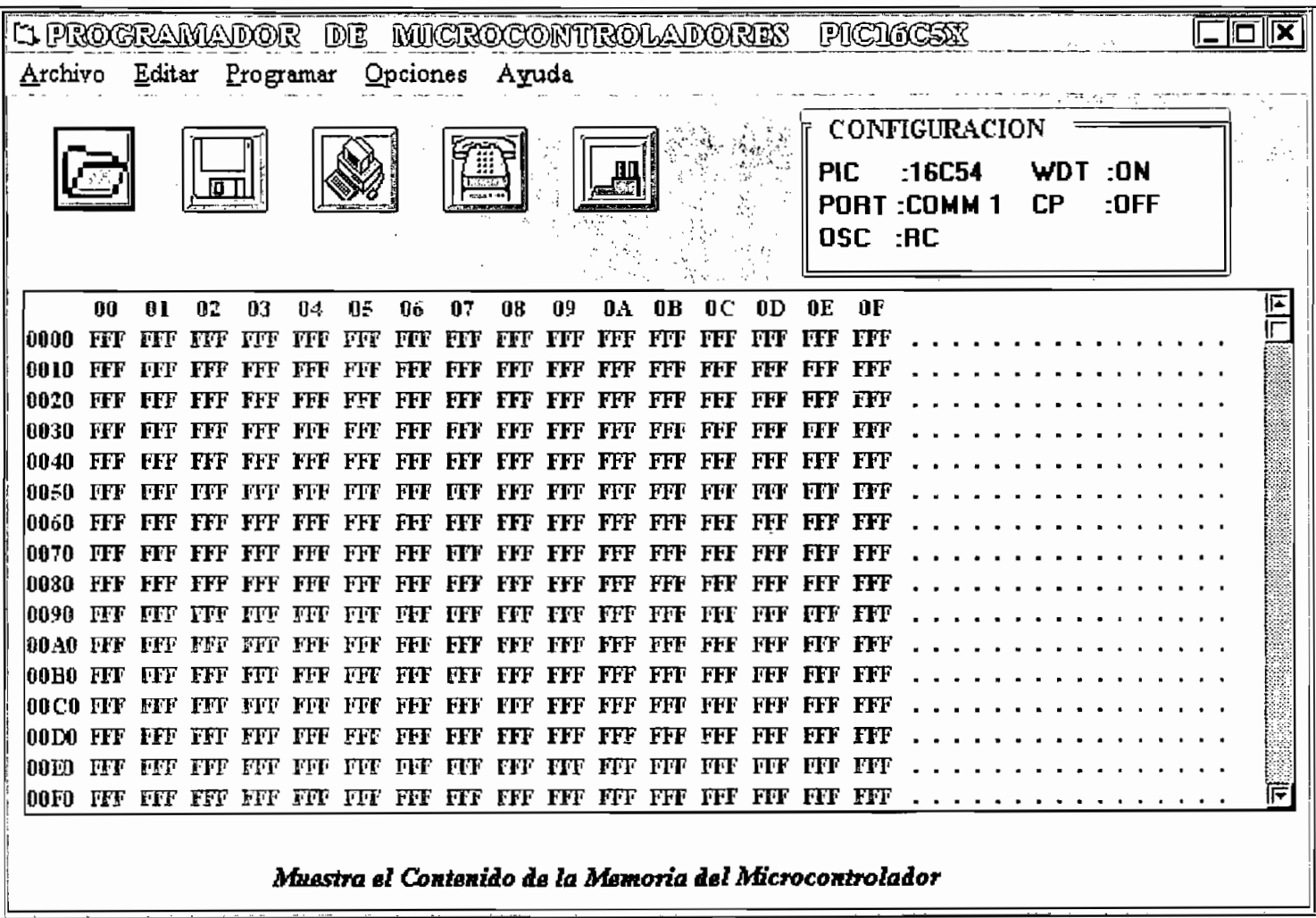


Figura 4.2 Programador de Microcontroladores PIC16CSX

Transmitir datos a ser programados y recibir datos enviados por el programador

Borrar buffer de recepción y transmisión.

Leer archivo a ser programado

Terminar con la programación y presentar el mensaje de programación O.K., si es fin de archivo

Leer los dos puntos.

Leer el número de datos que contiene la línea a ser programada

Salir del procedimiento de programación y presentar el mensaje de programación O.K., si el número de datos a ser programado es cero.

Transmitir el dato y validar el mismo.

Leer la dirección en donde empieza la programación, si la validación es correcta. Presentar el mensaje de error de programación, si la validación es incorrecta.

Transmitir dirección.

Validar dirección. Si la validación es exitosa se procederá a leer los datos que contiene la línea, caso contrario se presentará el mensaje de error de programación.

Transmitir los datos a ser programados.

Validar datos transmitidos, si validación es incorrecta, presentar mensaje de error de programación y salir del procedimiento; si validación es correcta verificar si se han programado todos los datos; si es afirmativo presentar el mensaje de programación O.K., de lo contrario volver a transmitir los datos que faltan.

Fin de Tarea

En el procedimiento para la verificación se tiene las siguientes tareas:

Open Com

Borrar el buffer de recepción.

Abrir pórtico de comunicaciones.

Si no existe ningún problema con el pórtico de comunicaciones continuará con la verificación caso contrario se mostrará el mensaje que se debe seleccionar otro pórtico de comunicaciones .

Fin de Tarea

Envío de comando de lectura

Dependiendo del elemento seleccionado se transmitirá el comando respectivo.

Fin de Tarea

Leer dato de configuración del micro

Esperar a que lleguen datos al buffer de recepción del computador.

Si no se recibe ningún dato o se excede de cierto tiempo se presentará el mensaje de error de verificación de la memoria.

El dato recibido se lo compara con el comando enviado (en el mismo se halla codificado la configuración del micro a ser leída).

Si la comparación resulta exitosa sigue con el procedimiento de verificación, caso contrario setear las banderas de error de verificación, para presentar en pantalla el error ocurrido posteriormente.

Fin de Tarea

Inicializar la dirección

Inicializar la dirección con cero del vector donde se van a guardar los datos enviados por el programador.

Fin de Tarea

Recibir dato de la dirección i

Borrar buffer de recepción

Leer dato enviado por el programador

Guardar dato en el vector Dato3\$()

Fin de Tarea

Validación del dato recibido

Comparar dato del buffer de memoria con el dato enviado por el programador, si estos son iguales incrementar la dirección

Verificar si es la última dirección de la memoria; si es la última localidad de memoria mostrar

el mensaje de verificación O.K.

Si no son iguales, guardar la dirección y el dato en que se produjo el error y setear las banderas de error de verificación.

Incrementar la dirección, si la dirección es igual al tamaño de la memoria, presentar en pantalla los datos donde la verificación no fue O.K.; si no se ha llegado a la última dirección de la memoria volver a recibir el dato del programador para su verificación.

Fin de tarea

En el procedimiento para chequear si la memoria se encuentra en blanco y para la lectura de la misma se tiene las siguientes tareas:

Open Com

Borrar el buffer de recepción.

Abrir pórtico de comunicaciones.

Si no existe ningún problema con el pórtico de comunicaciones continuará con la lectura caso contrario se mostrará el mensaje seleccionar otro pórtico de comunicaciones .

Fin de Tarea

Envío de comando de lectura

Dependiendo del elemento seleccionado se transmitirá el comando respectivo.

Fin de Tarea

Leer dato de configuración del micro

Esperar a que llegue la configuración del micro al buffer de recepción del computador.

Si no se recibe ningún dato o se excede de cierto tiempo se presentará el mensaje de error de lectura de la memoria.

Fin de Tarea

Inicializar la dirección

Inicializar la dirección con cero del vector donde se van a guardar los datos enviados por el programador.

Fin de Tarea

Recibir dato de la dirección i

Borrar buffer de recepción.

Leer dato enviado por el programador.

Guardar dato en el vector Dato1\$(i).

Si bandera verificación es igual a cero procedo a leer.

Incrementar dirección.

Comparar la dirección con el tamaño de la memoria.

Si son iguales salir del procedimiento de lectura.

Si no son iguales volver al procedimiento de recibir el dato de la dirección i.

Si bandera verificación es igual a uno procedo a verificar si la memoria se encuentra en blanco.

Comparar el dato guardado en Dato1\$(i) con FFFH.

Si son iguales.

Incrementar la dirección.

Comparar la dirección con el tamaño de la memoria.

Si son iguales presentar en pantalla el mensaje de Memoria en Blanco.

Si no son iguales volver al procedimiento de recibir dato de la dirección i.

Si no son iguales.

Presentar en pantalla el mensaje Memoria no se encuentra en Blanco.

Regresar a la pantalla principal.

Fin de Tarea

En el procedimiento para abrir un archivo.hex se tiene las siguientes tareas:

Abrir Archivo For Input.

- Setear el control Dialog para que se presente el formato de abrir un archivo.
- Si no se ingresa ningún nombre o no se selecciona el archivo, salir del procedimiento
- Abrir el archivo seleccionado como #1.
- Revisar si EOF(1) es verdadero.
 - Si es verdadero salir de del procedimiento.
 - Si es falso proceder a leer el archivo seleccionado.

Fin de tarea

Leer el archivo seleccionado

- Leer los dos puntos.
- Leer el número de datos que contiene la línea del archivo y grabar en la variable Numdatos.
- Leer la dirección en donde empieza los datos y grabar en la variable dirección.
- Leer el separador entre datos y dirección de inicio.
- Leer el dato y almacenarlo en el buffer de memoria a través del vector Dato2\$().
- Número de datos que contiene la línea, no es igual al número de datos almacenados en Dato2\$() proceder a leer un nuevo dato, si son iguales leer el checksum.
- Verificar si es fin de archivo.
 - Si es verdadero salir de del procedimiento.
 - Si es falso proceder a leer la siguiente línea del archivo seleccionado.
- Codificar el vector Dato2\$() que contiene datos hexadecimales de 8 bits a 12bits en un nuevo vector Dato1\$().

Fin de tarea

Como resultado del código implementado se tiene la Figura 4.3.

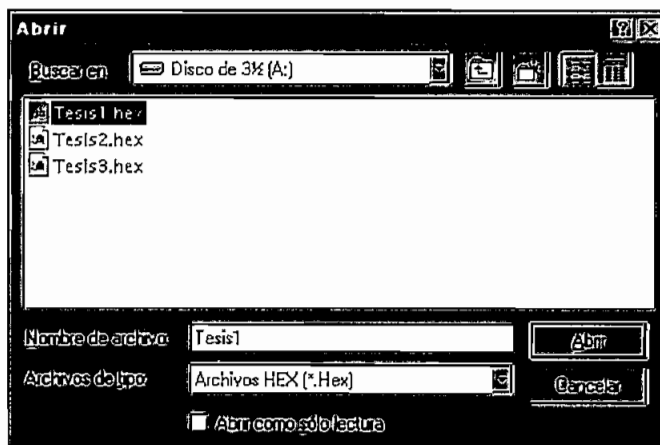


Figura 4.3 Pantalla que permite buscar el archivo que se desea abrir

En el procedimiento para poder guardar en un archivo el contenido del buffer de memoria se tiene las siguientes tareas:

Abrir Archivo For Output

- Setear el control Dialog para que se presente el formato de guardar un archivo.
- Si se a modificado el buffer de memoria el archivo que se va a generar tendrá el nombre progpic.hex en caso contrario salir del procedimiento.
- Abrir el archivo que se va a generar.

Fin de tarea

Generación del archivo.hex

- Seleccionar la Fila uno y la columna cero de la grilla en donde se encuentran los datos.
- Guardar el dato de la dirección en la variable direcc1\$.
- Grabar en el archivo :,10,direcc1\$,00.

Incrementar la columna.
 Leer el dato de la celda de la grilla.
 Decodificar el dato de la celda para poder obtener dos datos de 8 bits y grabar estos datos en el archivo.
 Verificar si el valor de la columna es 16.
 Si es falso proceder a leer el dato de la celda.
 Si es verdadero:
 Incrementar la fila.
 Grabar 00 y fin de línea en el archivo
 Verificar si el valor de la fila corresponde a la última fila de la grilla.
 Si es falso volver a la columna cero de la grilla y regresar al paso número dos del procedimiento.
 Si es verdadero grabar en el archivo :00000001FF.

Fin de tarea

En el procedimiento para editar el buffer de memoria se tiene las siguientes tareas:

Inicializar dirección 0000H

Inicializar el valor de la dirección con 000H.
 Si se desea se puede cambiar el valor de la dirección

Fin de tarea

Presentar el dato de la dirección

Presentar en pantalla el valor que contiene la dirección seleccionada.

Fin de Tarea

Ingreso de dato

Ingresar dato.
 Si dato ingresado no es un valor hexadecimal mostrar mensaje de error.
 Verificar si se procede a aceptar el dato ingresado.
 Si se acepta el ingreso del dato, modificar el valor que contiene dicha dirección y presentar en pantalla el cambio efectuado.
 Incrementar el valor de la dirección y volver al paso uno del procedimiento.
 Si no se acepta el dato, no se modificara el valor de la dirección y se procederá a salir del procedimiento.

Fin de Tarea

El resultado del código implementado se muestra en la Figura 4.4.

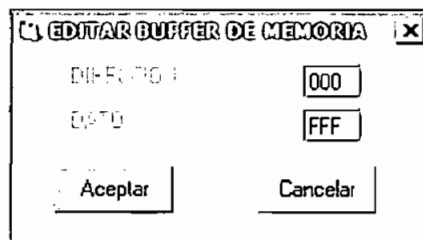


Figura 4.4 Editor de memoria

Tanto para la programación, lectura y chequeo de la memoria se debe seleccionar el micro con que se va a trabajar, del código implementado se obtienen las Figuras 4.5 y 4.6

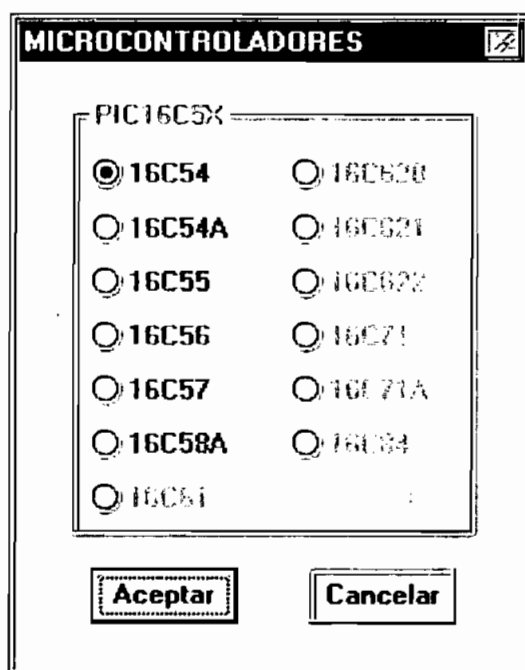


Figura 4.5 Microcontroladores PIC16C5X



Figura 4.6 Configuración del Microcontrolador

El listado de los programas implementados tanto para el programador como para el computador se adjunta a la presente tesis en Anexos, de igual manera el manual del usuario.

CAPITULO V

PRUEBAS Y RESULTADOS

La familia de los microcontroladores PIC16C5X están constituidos por los elementos que se presentan en la Tabla 5.1:

Elemento	16C54	16C54A	16C55	16C56	16C57	16C58A
No. de Pines	18	18	28	18	28	18
Memoria	512	512	512	1KB	2KB	2KB

Tabla 5.1 Familia de los PIC16C5X

Del universo de microcontroladores que se tiene se ha escogido para realizar las pruebas a los microcontroladores PIC16C55 y PIC16C58A, con este tipo de selección se estaría cubriendo los dos tipos de microcontroladores, esto es, los micros de 18 pines como los micros de 28 pines, además se esta cubriendo el espectro de memoria con el que se cuenta (512 Bytes a 2 KB).

Se tomo esta decisión debido a que no se puede contar con todos los elementos antes mencionados, debido al costo que representaría tener todos ellos.

Para probar el sistema se procedió a realizar las pruebas que se detallan a continuación:

1. Prueba de Lectura de la memoria.
2. Prueba de Programación (escritura); y
3. Prueba de Verificación.

Antes de realizar las pruebas se debe seleccionar el dispositivo con el que se va a trabajar, además las pruebas son las mismas para ambos microcontroladores.

1) Prueba de Lectura.- En esta prueba se realizaron dos sub-pruebas:

La primera consiste en usar un PIC con su memoria en blanco y con el programador se procede a utilizar la opción de verificar si esta en blanco dando como resultado que la memoria se encuentra en blanco..

La segunda prueba consiste en leer el contenido de la memoria del microcontrolador para luego el contenido de la misma presentarla en pantalla, con el resultado de estas dos pruebas se determino el estado de la memoria del microcontrolador PIC

2) Prueba de Programación.- Para poder realizar ésta prueba se debe comprobar que la memoria del micro a programar este en blanco o borrada, esto se determinó con la prueba que se realizo anteriormente.

Posteriormente se procedió a cargar en el buffer de memoria el archivo a programar, con la opción abrir archivo del programador, para ésta prueba se utilizo el archivo ejemplo.hex:

ejemplo.hex

```
:1000000072007300080C3400090203043003030675  
:10001000F20132033303F402060A00084000020032  
:0C00200006023000060229000009100A48  
:0203FE000E0AE5  
:00000001FF
```

El resultado de abrir el archivo ejemplo.hex se muestra en la Figura 5.1, tanto en forma hexadecimal como ASCII.

A continuación se procedió a programar la memoria del microcontrolador, con la opción de programar del menú de programación (o utilizando el comando F5). Dando como resultado el mensaje de Programación O.K.

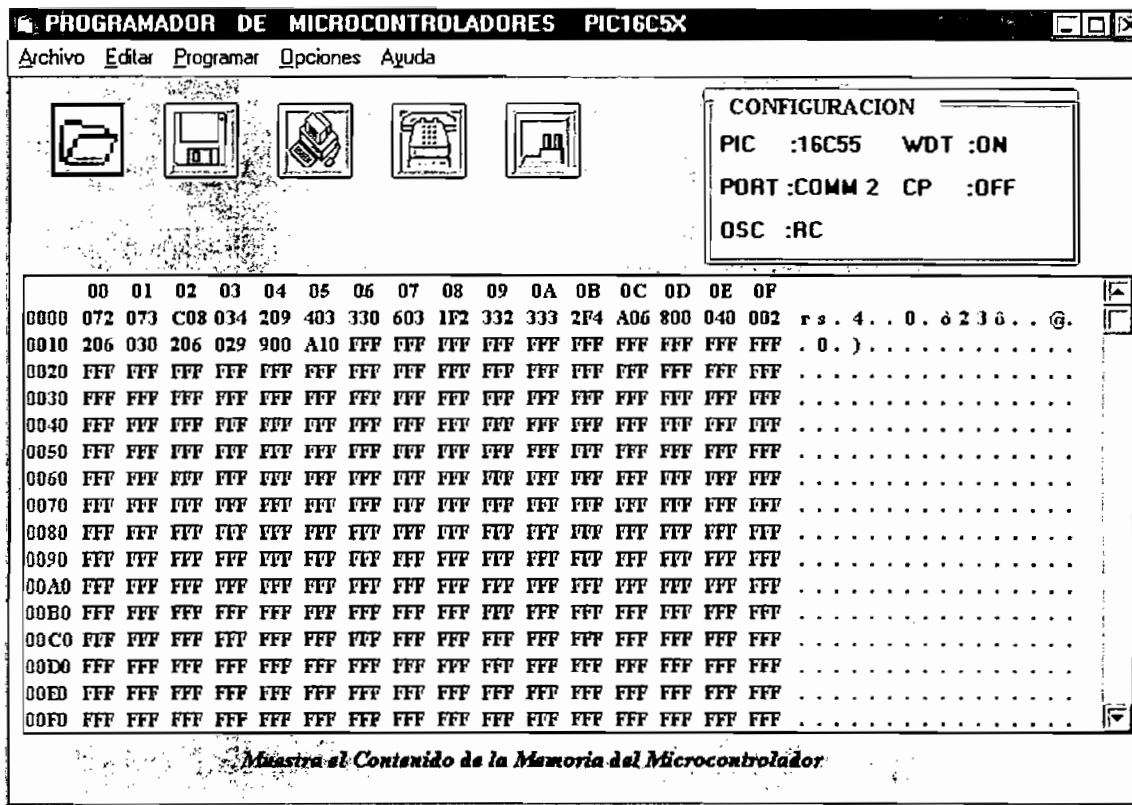


Figura 5.1 Presentación del archivo ejemplo.hex a ser programado

3) Prueba de Verificación.- Después de haber programado la memoria del microcontrolador se realizó la verificación de la memoria, la misma que consiste en comparar los valores que se tiene en el buffer de memoria del computador con los valores que se leen de la memoria del microcontrolador.

- El resultado de la verificación presento en pantalla el mensaje de Verificación O.K.

Con el archivo ejemplo.hex cargado en el buffer de memoria se procedió a editar la dirección 0000H, cambiando su valor de 072H a 241H y la dirección 0001H de 0073H a 454H. Adicionalmente se cambio la condición de WDT de "ON" a "OFF". Obviamente la verificación debía dar como resultado, "Verificación no O.K.", lo que efectivamente ocurrió así. El programador fué desarrollado de tal forma que permite ver en pantalla las direcciones que tienen error, así como también los valores verdaderos y erróneos de los bits

de configuración, como se indica en la Figura 5.2.

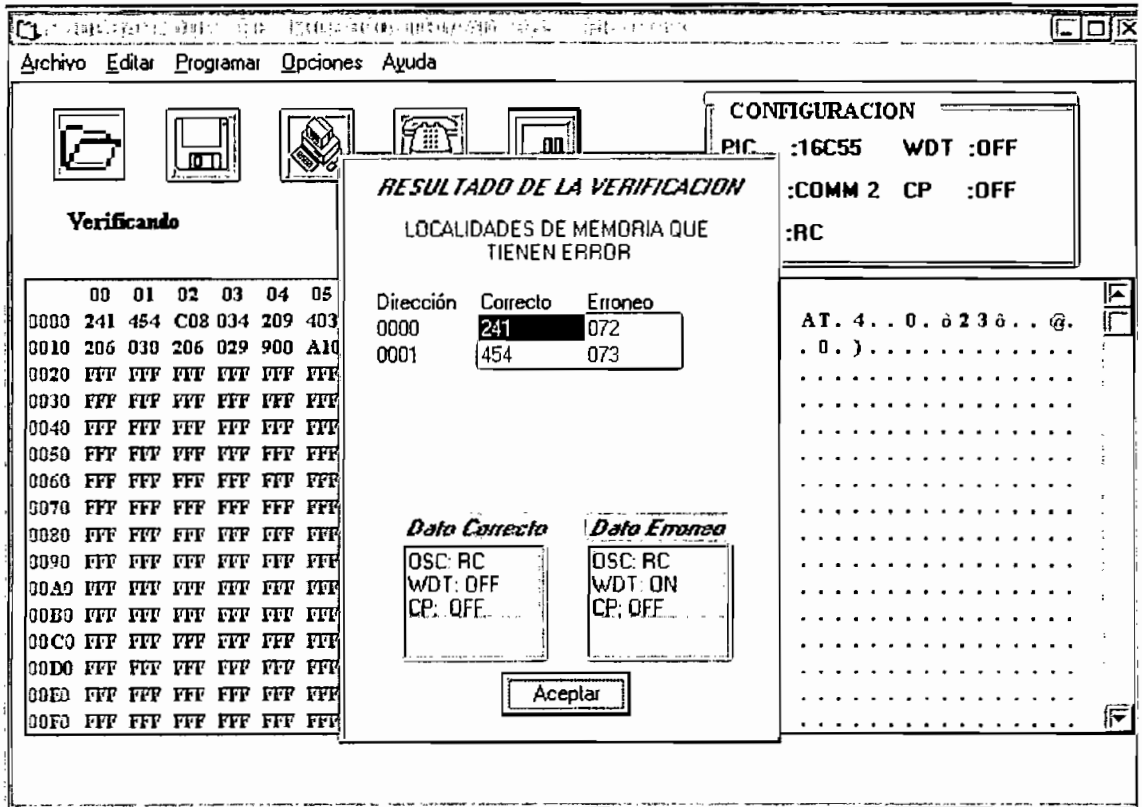


Figura 5.2 Verificación de la memoria después de los cambios realizados

Para mayor seguridad se procedió a cargar en el buffer de memoria del computador el archivo tutor.hex, y se procedió a realizar nuevamente la prueba de verificación.

tutor.hex:

```
:02000000502886
:1000A000FF30A0000130A1000730A2005A20A20BAF
:1000B00056285028A10E210EA10EA0026020003467
:0A00C000A10E210EA10EA1070034CD
:00000001FF
```

Al realizar la prueba de verificación se obtuvo una verificación no O.K., y en pantalla se presentan las direcciones en las cuales los valores no coinciden, como se muestra en las Figuras 5.3 y 5.4.

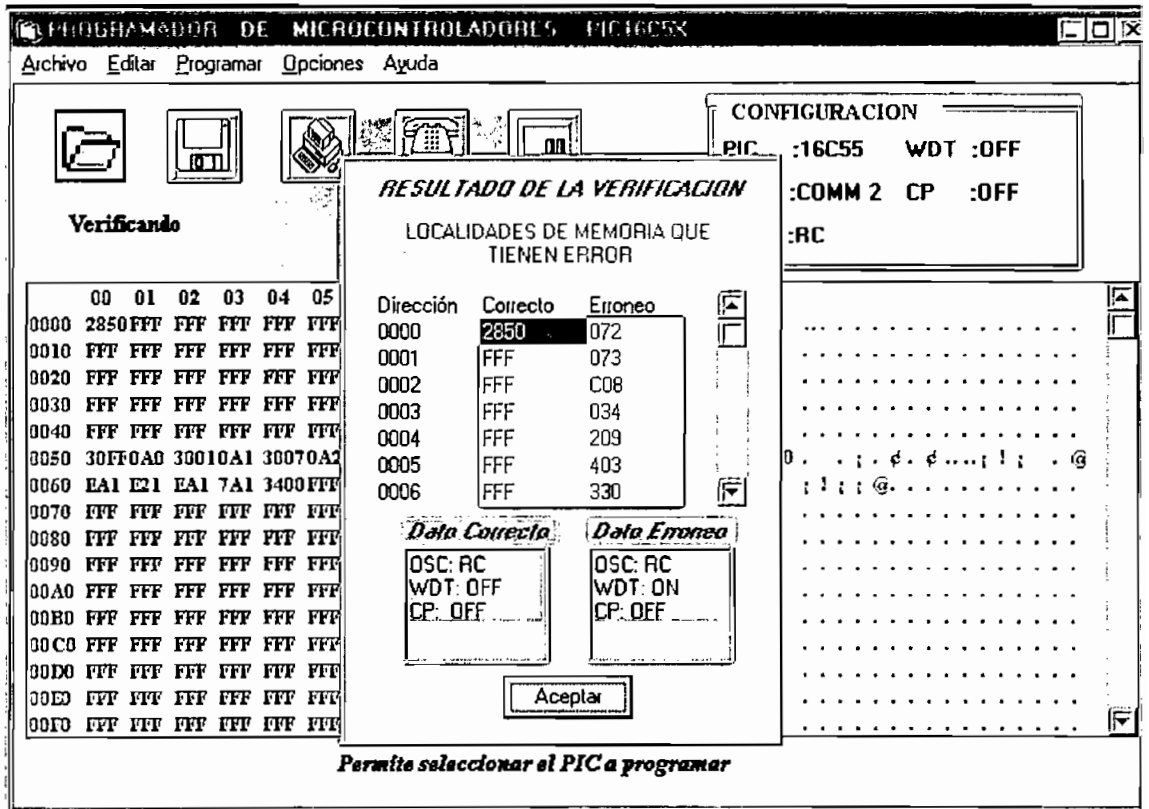


Figura 5.3 Resultado de la Verificación con el archivo Tutor.hex

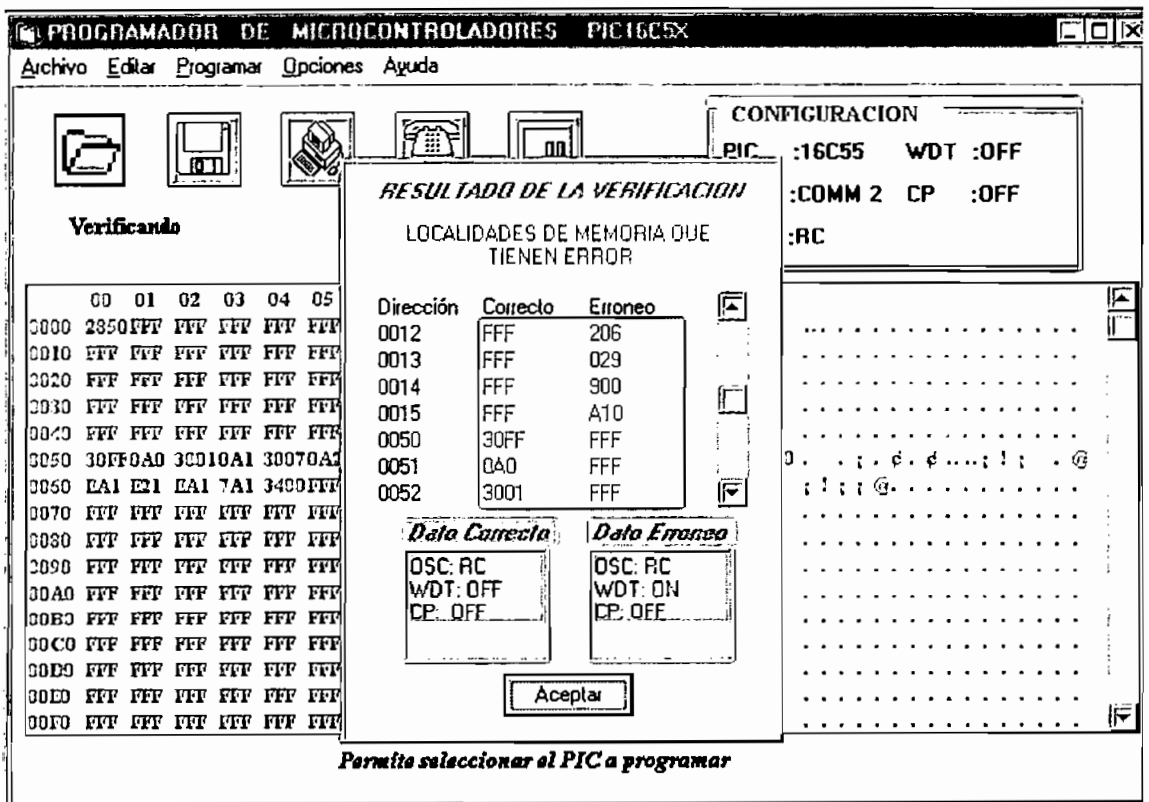


Figura 5.4 Continuación del resultado de la Verificación con el archivo Tutor.hex

Se procedió luego a utilizar la opción guardar; y el contenido del buffer de memoria se almacenó en el archivo que se avía abierto en ese instante. Con la opción guardar como el contenido del buffer de memoria es grabado en un archivo cuyo nombre es puesto por el usuario. Para comprobar si lo descrito anteriormente ocurre efectivamente como se indica se procedió a cargar el archivo ejemplo.hex en el buffer de memoria, para luego correr la opción de grabar obteniéndose el siguiente resultado.

archivo inicial:

```
:1000000072007300080C3400090203043003030675
:10001000F20132033303F402060A00084000020032
:0C00200006023000060229000009100A48
:0203FE000E0AE5
:00000001FF
```

archivo después de correr la opción guardar o guardar como

```
:1000000072007300080C3400090203043003030600
:10001000F20132033303F402060A00084000020000
:1000200006023000060229000009100AFF0FFF0F00
:10003000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10004000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10005000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10006000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10007000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10008000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10009000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:1000A000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:1000B000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:1000C000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:1000D000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:1000E000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:1000F000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10010000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10011000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10012000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10013000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10014000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10015000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10016000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10017000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10018000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10019000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:1001A000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
```

```
:1001B000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:1001C000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:1001D000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:1001E000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:1001F000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10020000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10021000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10022000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10023000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10024000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10025000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10026000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10027000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10028000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10029000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:1002A000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:1002B000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:1002C000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:1002D000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:1002E000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:1002F000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10030000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10031000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10032000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10033000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10034000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10035000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10036000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10037000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10038000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:10039000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:1003A000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:1003B000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:1003C000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:1003D000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:1003E000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F00
:1003F000FF0FFF0FFF0FFF0FFF0FFF0FFF0FFF0F0E0A00
:00000001FF
```

Para el sistema implementado el archivo que se genera con la opción guardar o guardar como, al ser cargado en el buffer de memoria del computador se presenta igual que el archivo original.

CAPITULO VI

ANALISIS TECNICO ECONOMICO

Se procedió a sacar un listado de todos los elementos utilizados en el en equipo con sus respectivo precio, tomando como referencia los precios de los elementos dados por JDR Electronics en el catalogo No.300 de Agosto, los precios se encuentra en dólares.

Elemento	Cantidad	Precio Unitario	Precio Total
ZIF de 28 pines	1	16	16
CI INTEL 8751H	1	20	20
CI MAX232	1	3	5
CI SN74LS03	1	0,75	0,75
CI 74HC4066N	1	0,75	0,75
CI SN7406	1	0,8	0,80
CRISTAL DE 7.15MHz	1	1,09	1,09
LM317 (Regulador de 13V)	1	0,69	0,69
LM7805 (Regulador de 5V)	1	0,49	0,49
Resistencia de 10K de ¼ de Watio	15	0,0179	0,2685
Resistencia de 1K de ¼ de Watio	4	0,0179	0,0716
Resistencia de 280 de ¼ de Watio	1	0,0179	0,0179
Resistencia de 560 de ¼ de Watio	1	0,0179	0,0179
Condensador de 30pF	2	0,25	0,5
Condensador de 1.0uF de tantalio	5	0,2	1
Condensador de 0.1uF	6	0,2	1,2
Condensador de 1000uF/25V	1	1,09	1,09
Condensador de 350uF/25V	2	0,3	0,6
Transistor 2N2222	2	0,25	0,5
Potenciometro de !k de precisión	1	0,85	0,85
Puente Rectificador de 1.5Amp.	2	0,1	0,2
Led	1	0,11	0,11
Pulsador	1	0,39	0,39
Conector DB9 (Hembra)	1	0,45	0,45
Disipador	1	0,49	0,49
Socket de 40 pines	1	1,19	1,19
Socket de 14 pines	3	0,59	1,77
Socket de 16 pines	1	0,60	0,6
Transformador	1	9	9
Fusible	1	0,5	0,5
Porta fusible	1	1	1

Elemento	Cantidad	Precio Unitario	Precio Total
Tarjeta	1	25	25
Caja	1	10	10
TOTAL			102,368

El precio total del equipo es de ciento dos dólares con 3 centavos.

Las bondades que presenta el equipo son las siguientes:

1. Puede ser utilizado con cualquier computador que posea una interfase serial RS232 y que tenga el sistema operativo Windows o Windows 95.
2. No se necesita abrir el computador para poder instalar el equipo.
3. Utiliza un programa de fácil utilización, el mismo que incluye un archivo de ayuda, donde se encuentra indicado la utilización de los comandos.
4. Este equipo puede ser ampliado en el futuro para programar los siguientes microcontroladores: PIC16C61, PIC16C620, PIC16C621, PIC16C622, PIC16C71, PIC16C71A y PIC16C84, con sólo modificar el programa de soporte.
5. Como el equipo se encuentra totalmente documentado, el soporte técnico se encuentra a la mano, lo que permite un funcionamiento permanente.
6. El hardware del equipo es sencillo y fácil de comprender, lo que permite que sea reparado con mucha facilidad..
7. Es fácil de instalar.
8. No requiere mucho espacio para su instalación
9. El programa trabaja en un ambiente Windows.
10. Tiene la posibilidad de grabar el código de protección
11. Realiza todas las operaciones de un programador. Estas son: leer, programar, verificar, chequear, editar bytes, grabar en archivo, mostrar contenido de la memoria del micro tanto en hex. como en ASCII.
12. Acepta formato INTEL para ser grabado.

En el mercado se ofrecen muchos programadores, desde programadores universales hasta programadores específicos. Entre los programadores específicos se tienen el programador PICSTART16B el mismo que tiene un costo de 285 dólares. Este programador se parece al equipo implementado, pero al hacer una comparación de precios se puede ver que el programador implementado es mucho más barato lo cual justifica su implementación.

El precio tomado para la implementación del equipo no incluye el costo de desarrollo, esto es costo Ing/hora.

El equipo se pudo haber implementado en tres meses, calculando a 333.33 dólares (1'500.000 sucres) por mes, tendríamos que agregar al costo del equipo, la suma de 1000 dólares, dando como resultado que el costo total del equipo de 1102,3 dólares, cabe recalcar que este costo se debería prorratear para el número de unidades vendidas.

CAPITULO VII

CONCLUSIONES

De la experiencia adquirida en la implementación del equipo y de los resultados obtenidos en las pruebas se presentan las siguientes conclusiones:

En la comunicación serial la velocidad de transmisión/recepción es baja, sin embargo esta reducción de velocidad no es tan crítica ya que en el proceso de programación se utilizan retardos. Esta desventaja se ve compensada por su bajo costo de implementación.

En la comunicación serial, no se utilizó líneas de control (Hand-Shake) debido a que la aplicación es sencilla.

Con la utilización del MAX232 nos evitamos utilizar fuentes de tensión de ± 12 V requeridas para la interfase serial RS-232, reduciéndose de esta manera la complejidad del circuito. Además su utilización facilita la implementación de la comunicación serial.

El programador desarrollado está expuesto a daños debido a que el estado del elemento a ser programado/verificado no se puede determinar, aun viniendo del fabricante, por este motivo, la confiabilidad del equipo esta sujeto al estado en el que se encuentre el chip a ser utilizado. El caso más crítico sería si usáramos un chip que tenga el pin de Vcc con el pin de GND en corto, esto provocaría que el transistor Q1 se quemara por un exceso de corriente, ya que este es el encargado de suministrar el voltaje de polarización a chip a ser programado/verificado. Este elemento puede ser fácilmente cambiado.

Se utiliza un solo socket de programación tanto para los micros de 18 pines como para los micros de 28 pines, por tal motivo se debe tener muy en cuenta que la elección del elemento a ser programado, realizada en el computador, esté de acuerdo al chip que va a ser colocado en el socket de programación para evitar dañarlo.

El potenciómetro que permite calibrar el voltaje de programación (V_{pp}) no debe ser

manipulado, ya que este se encuentra calibrado para un valor de V_{pp} adecuado, el incremento de este voltaje puede llegar a destruir el chip a ser programado.

Las resistencias de pull-up que se colocaron en el p rtico cero del microcontrolador 8751H, evita que el estado del p rtico se encuentre indeterminado, entre un valor de cero l gico y uno l gico.

Para evitar que existan corrientes de fuga a trav s de los microcontroladores PIC, se inicializ  los p rticos P0 y P2 del microcontrolador 8751H con el valor de 00H, de  sta manera se evitar  el da ar o deteriorar el chip a ser programado.

De las pruebas realizadas de verificaci n y programaci n se deduce que el equipo es 100% seguro.

Al tener un equipo que permite una f cil conexi n y desconexi n, as  como tambi n una f cil transportaci n por su tama o, resulta una opci n conveniente frente a programadores cuyo funcionamiento va ligado a tarjetas que deben ser instaladas dentro del computador.

Si bien es cierto que la implementaci n del presente programador a llevado su tiempo, sin embargo es garantizado en el sentido de que se encuentra documentado tanto a nivel de hardware como de software, lo cual permitir  el mantenimiento, soporte adecuado y futuras mejoras, ventajas que no se tienen con un equipo tra do del extranjero.

BIBLIOGRAFIA

- 1 **INTEL MICROCONTROLLERS HANDBOOK**, Intel Literature Sales, Santa Clara CA. 1991.
- 2 **MICROCHIP PIC16/17 MICROCONTROLLER DATABOOK**, Microchip Technology 1994/1995.
- 3 **MICROCHIP EMBEDDED CONTROL HANDBOOK**, Microchip Technology 1995/1996.
- 4 **LINEAR DATABOOK**, National Semiconductor Corporation, 1982
- 5 **MAXIN HANDBOOK**, Maxin, 1995
- 6 **INTEL COMPONENT DATA CATALOG**, Intel Literature Sales Santa Clara, 1979.

REFERENCIAS

- 1 Microcomputer Journal, May/June, 1994
- 2 Microcomputer Journal, September/October, 1994
- 3 <http://www.microchip.com>
- 4 <http://www.ibars.com>
- 5 <http://www.eetoolbox.com/gatopic.htm>
- 6 <http://www.lancs.ac.uk/personas/cpaarne/pic/pic.htm>

ANEXOS

ANEXO 1
CARACTERISTICAS DE LOS
MICROCONTROLADORES PIC16C5X

PIC16C5X

D.2 PIC16C5X Family of Devices

Device	Clock		Memory		Peripherals		Features		
	Maximum Frequency of Operation (MHz)	Program Memory (K12 words)	EPROM	ROM	RAM Data Memory (bytes)	Timer Module(s)	I/O Pins	Voltage Range (Volts)	Number of Instructions
PIC16C52	4	384	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, SOIC
PIC16C54	20	512	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, SOIC; 20-pin SSOP
PIC16C54A	20	512	—	25	TMR0	12	2.0-6.25	33	18-pin DIP, SOIC; 20-pin SSOP
PIC16CR54A	20	—	512	25	TMR0	12	2.0-6.25	33	18-pin DIP, SOIC; 20-pin SSOP
PIC16C55	20	512	—	24	TMR0	20	2.5-6.25	33	28-pin DIP, SOIC, SSOP
PIC16C56	20	1K	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, SOIC; 20-pin SSOP
PIC16C57	20	2K	—	72	TMR0	20	2.5-6.25	33	28-pin DIP, SOIC, SSOP
PIC16CR57B	20	—	2K	72	TMR0	20	2.5-6.25	33	28-pin DIP, SOIC, SSOP
PIC16C58A	20	2K	—	73	TMR0	12	2.0-6.25	33	18-pin DIP, SOIC; 20-pin SSOP
PIC16CR58A	20	—	2K	73	TMR0	12	2.5-6.25	33	18-pin DIP, SOIC; 20-pin SSOP

All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

TABLE 10-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS (RC, XT & 10) AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)

OSC	PIC16C5X-RC	PIC16C5X-XT	PIC16C5X-10
RC	VDD: 3.0 V to 6.2 V IDD: 3.3 mA max. at 5. V IPD: 9 µA max. at 3.0 V, WDT dis Freq: 4 MHz max.	N/A	N/A
XT	VDD: 3.0V to 6.25V IDD: 1.8 mA typ. at 5.5V IPD: 0.6 µA typ. at 3.0V, WDT dis Freq: 4 MHz max.	VDD: 3.0V to 6.25V IDD: 3.3 mA max. at 5.5V IPD: 9 µA max. at 3.0V, WDT dis Freq: 4 MHz max.	N/A
HS	VDD: 4.5V to 5.5V IDD: 9.0 mA typ. at 5.5V IPD: 0.6 µA typ. at 3.0V, WDT dis Freq: 20 MHz max.	N/A	VDD: 4.5V to 5.5V IDD: 10 mA max. at 5.5V IPD: 9 µA max. at 3.0V, WDT dis Freq: 10 MHz max.
LP	VDD: 2.5V to 6.25V IDD: 15 µA typ. at 3.0V IPD: 0.6 µA typ. at 3.0V, WDT dis Freq: 40 kHz max.	VDD: 2.5V to 6.25V IDD: 15 µA typ. at 3.0V IPD: 0.6 µA typ. at 3.0V, WDT dis Freq: 40 kHz max.	VDD: 2.5V to 6.25V IDD: 15 µA typ. at 3.0V IPD: 0.6 µA typ. at 3.0V, WDT dis Freq: 40 kHz max.

The shaded sections indicate oscillator selections which should work by design, but are not tested. It is recommended that the user select the device type from information in unshaded sections.

TABLE 10-2: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS (HS, LP & JW) AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)

OSC	PIC16C5X-HS	PIC16C5X-LP	PIC16C5X/JW
RC	N/A	N/A	VDD: 3.0V to 6.25V IDD: 3.3 mA max. at 5.5V IPD: 9 µA max. at 3.0V, WDT dis Freq: 4 MHz max.
XT	N/A	N/A	VDD: 3.0V to 6.25V IDD: 3.3 mA max. at 5.5V IPD: 9 µA max. at 3.0V, WDT dis Freq: 4 MHz max.
HS	VDD: 4.5V to 5.5V IDD: 20 mA max. at 5.5V IPD: 9 µA max. at 3.0V, WDT dis Freq: 20 MHz max.	N/A	VDD: 4.5V to 5.5V IDD: 20 mA max. at 5.5V IPD: 9 µA max. at 3.0V, WDT dis Freq: 20 MHz max.
LP	VDD: 2.5V to 6.25V IDD: 15 µA typ. at 3.0V IPD: 0.6 µA typ. at 3.0V, WDT dis Freq: 40 kHz max.	VDD: 2.5V to 6.25V IDD: 32 µA max. at 32 kHz, 3.0V IPD: 9 µA max. at 3.0V, WDT dis Freq: 40 kHz max.	VDD: 2.5V to 6.25V IDD: 32 µA max. at 32 kHz, 3.0V IPD: 9 µA max. at 3.0V, WDT dis Freq: 40 kHz max.

The shaded sections indicate oscillator selections which should work by design, but are not tested. It is recommended that the user select the device type from information in unshaded sections.

10.1 DC Characteristics: PIC16C5X-RC, XT, 10, HS, LP (Commercial)

DC Characteristics Power Supply Pins		Standard Operating Conditions (unless otherwise specified) Operating Temperature $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$				
Characteristic	Sym	Min	Typ ⁽¹⁾	Max	Units	Conditions
Supply Voltage PIC16C5X-RC PIC16C5X-XT PIC16C5X-10 PIC16C5X-HS PIC16C5X-LP	V _{DD}	3.0 3.0 4.5 4.5 2.5		6.25 6.25 5.5 5.5 6.25	V V V V V	Fosc = DC to 4 MHz Fosc = DC to 4 MHz Fosc = DC to 10 MHz Fosc = DC to 20 MHz Fosc = DC to 40 kHz
RAM Data Retention Voltage ⁽²⁾	V _{DR}		1.5*		V	Device in SLEEP Mode
V _{DD} Start Voltage to ensure Power-On Reset	V _{POR}		V _{SS}		V	See Section 7.4 for details on Power-On Reset
V _{DD} Rise Rate to ensure Power-On Reset	SV _{DD}	0.05*			V/rms	See Section 7.4 for details on Power-On Reset
Supply Current ⁽³⁾ PIC16C5X-RC ⁽⁴⁾ PIC16C5X-XT PIC16C5X-10 PIC16C5X-HS PIC16C5X-LP	I _{DD}		1.8 1.8 4.8 4.8 9.0 15	3.3 3.3 10 10 20 32	mA mA mA mA mA μA	Fosc = 4 MHz, V _{DD} = 5.5V Fosc = 4 MHz, V _{DD} = 5.5V Fosc = 10 MHz, V _{DD} = 5.5V Fosc = 10 MHz, V _{DD} = 5.5V Fosc = 20 MHz, V _{DD} = 5.5V Fosc = 32 kHz, V _{DD} = 3.0V, WDT disabled
Power Down Current ⁽⁵⁾	I _{PD}		4.0 0.6	12 9	μA μA	V _{DD} = 3.0V, WDT enabled V _{DD} = 3.0V, WDT disabled

* These parameters are characterized but not tested.

Note 1: Data in the Typical ("Typ") column is based on characterization results at 25°C. This data is for design guidance only and is not tested.

2: This is the limit to which V_{DD} can be lowered in SLEEP mode without losing RAM data.

3: The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, bus rate, internal code execution pattern, and temperature also have an impact on the current consumption.

a) The test conditions for all I_{DD} measurements in active operation mode are:

OSC1 = external square wave, from rail-to-rail; all I/O pins tristated, pulled to V_{SS}, T0CKI = V_{DD}, MCLR = V_{DD}; WDT enabled/disabled as specified.

b) For standby current measurements, the conditions are the same, except that the device is in SLEEP mode.

4: Does not include current through R_{ext}. The current through the resistor can be estimated by the formula: $I_R = V_{DD}/2R_{ext}$ (mA) with R_{ext} in k Ω

5: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to V_{DD} and V_{SS}.

10.2 DC Characteristics: PIC16C5X-RCI, XTI, 10I, HSI, LPI (Industrial)

DC Characteristics Power Supply Pins		Standard Operating Conditions (unless otherwise specified) Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$				
Characteristic	Sym	Min	Typ ⁽¹⁾	Max	Units	Conditions
Supply Voltage PIC16C5X-RCI PIC16C5X-XTI PIC16C5X-10I PIC16C5X-HSI PIC16C5X-LPI	VDD	3.0 3.0 4.5 4.5 2.5		6.25 6.25 5.5 5.5 6.25	V V V V V	Fosc = DC to 4 MHz Fosc = DC to 4 MHz Fosc = DC to 10 MHz Fosc = DC to 20 MHz Fosc = DC to 40 kHz
RAM Data Retention Voltage ⁽²⁾	VDR		1.5*		V	Device in SLEEP mode
VDD Start Voltage to ensure Power-On Reset	VPOR		VSS		V	See Section 7.4 for details on Power-On Reset
VDD Rise Rate to ensure Power-On Reset	SVDD	0.05*			V/ms	See Section 7.4 for details on Power-On Reset
Supply Current⁽³⁾ PIC16C5X-RCI ⁽⁴⁾ PIC16C5X-XTI PIC16C5X-10I PIC16C5X-HSI PIC16C5X-LPI	IDD		1.8 1.8 4.8 4.8 9.0 19	3.3 3.3 10 10 20 40	mA mA mA mA mA μA	Fosc = 4 MHz, VDD = 5.5V Fosc = 4 MHz, VDD = 5.5V Fosc = 10 MHz, VDD = 5.5V Fosc = 10 MHz, VDD = 5.5V Fosc = 20 MHz, VDD = 5.5V Fosc = 32 kHz, VDD = 3.0V, WDT disabled
Power Down Current⁽⁵⁾	IPD		5.0 0.6	14 12	μA μA	VDD = 3.0V, WDT enabled VDD = 3.0V, WDT disabled

* These parameters are characterized but not tested.

Note 1: Data in the Typical ("Typ") column is based on characterization results at 25°C. This data is for design guidance only and is not tested.

2: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

3: The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, bus rate, internal code execution pattern, and temperature also have an impact on the current consumption.

a) The test conditions for all IDD measurements in active operation mode are:

OSC1 = external square wave, from rail-to-rail; all I/O pins tristated, pulled to VSS, T0CKI = VDD, MCLR = VDD; WDT enabled/disabled as specified.

b) For standby current measurements, the conditions are the same, except that the device is in SLEEP mode.

4: Does not include current through Rext. The current through the resistor can be estimated by the formula: $I_R = V_{DD}/2R_{ext}$ (mA) with Rext in kΩ

5: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.

10.3 DC Characteristics: PIC16C5X-RCE, XTE, 10E, HSE, LPE (Automotive)

DC Characteristics Power Supply Pins		Standard Operating Conditions (unless otherwise specified) Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$				
Characteristic	Sym	Min	Typ ⁽¹⁾	Max	Units	Conditions
Supply Voltage PIC16C5X-RCE PIC16C5X-XTE PIC16C5X-10E PIC16C5X-HSE PIC16C5X-LPE	VDD	3.25 3.25 4.5 4.5 2.5		6.0 6.0 5.5 5.5 6.0	V V V V V	Fosc = DC to 4 MHz Fosc = DC to 4 MHz Fosc = DC to 10 MHz Fosc = DC to 16 MHz Fosc = DC to 40 kHz
RAM Data Retention Voltage ⁽²⁾	VDR		1.5*		V	Device in SLEEP mode
VDD Start Voltage to ensure Power-On Reset	VPOR		VSS		V	See Section 7.4 for details on Power-On Reset
VDD rise rate to ensure Power-On Reset	SVDD	0.05*			V/ms	See Section 7.4 for details on Power-On Reset
Supply Current ⁽³⁾ PIC16C5X-RCE ⁽⁴⁾ PIC16C5X-XTE PIC16C5X-10E PIC16C5X-HSE PIC16C5X-LPE	IDD		1.8 1.8 4.8 4.8 9.0 25	3.3 3.3 10 10 20 55	mA mA mA mA mA μA	Fosc = 4 MHz, VDD = 5.5V Fosc = 4 MHz, VDD = 5.5V Fosc = 10 MHz, VDD = 5.5V Fosc = 10 MHz, VDD = 5.5V Fosc = 16 MHz, VDD = 5.5V Fosc = 32 kHz, VDD = 3.25V, WDT disabled
Power Down Current ⁽⁵⁾	IPD		5.0 0.8	22 18	μA μA	VDD = 3.25V, WDT enabled VDD = 3.25V, WDT disabled

* These parameters are characterized but not tested.

Note 1: Data in the Typical ("Typ") column is based on characterization results at 25°C. This data is for design guidance only and is not tested.

2: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

3: The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, bus rate, internal code execution pattern, and temperature also have an impact on the current consumption.

a) The test conditions for all IDD measurements in active operation mode are:

OSC1 = external square wave, from rail-to-rail; all I/O pins tristated, pulled to VSS, T0CKI = VDD, MCLR = VDD; WDT enabled/disabled as specified.

b) For standby current measurements, the conditions are the same, except that the device is in SLEEP mode.

4: Does not include current through Rext. The current through the resistor can be estimated by the formula: $I_R = V_{DD}/2R_{ext}$ (mA) with Rext in kΩ

5: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.

10.4 **DC Characteristics:** PIC16C5X-RC, XT, 10, HS, LP (Commercial)
PIC16C5X-RCI, XTI, 10I, HSI, LPI (Industrial)

DC Characteristics All Pins Except Power Supply Pins		Standard Operating Conditions (unless otherwise specified) Operating Temperature 0°C ≤ TA ≤ +70°C (commercial) -40°C ≤ TA ≤ +85°C (industrial) Operating Voltage VDD range is described in Section 10.1, Section 10.2 and Section 10.3.				
Characteristic	Sym	Min	Typ ⁽¹⁾	Max	Units	Conditions
Input Low Voltage I/O ports \overline{MCLR} (Schmitt Trigger) T0CKI (Schmitt Trigger) OSC1 (Schmitt Trigger)	VIL	VSS VSS VSS VSS VSS		0.2 VDD 0.15 VDD 0.15 VDD 0.15 VDD 0.3 VDD	V V V V V	Pin at hi-impedance PIC16C5X-RC only ⁽⁴⁾ PIC16C5X-XT, 10, HS, LP
Input High Voltage I/O ports \overline{MCLR} (Schmitt Trigger) T0CKI (Schmitt Trigger) OSC1 (Schmitt Trigger)	VIH	0.45 VDD 2.0 0.36 VDD 0.85 VDD 0.85 VDD 0.85 VDD 0.7 VDD		VDD VDD VDD VDD VDD VDD VDD	V V V V V V V	For all VDD ⁽⁵⁾ 4.0V < VDD ≤ 5.5V ⁽⁵⁾ VDD > 5.5V PIC16C5X-RC only ⁽⁴⁾ PIC16C5X-XT, 10, HS, LP
Hysteresis of Schmitt Trigger inputs	VHYS	0.15VDD*			V	
Input Leakage Current ^(2,3) I/O ports \overline{MCLR} T0CKI OSC1	IIL	-1 -5 -3 -3	0.5 0.5 0.5 0.5	+1 +5 +3 +3	μA μA μA μA	For VDD ≤ 5.5V VSS ≤ VPIN ≤ VDD, Pin at hi-impedance VPIN = VSS + 0.25V VPIN = VDD VSS ≤ VPIN ≤ VDD VSS ≤ VPIN ≤ VDD, PIC16C5X-XT, 10, HS, LP
Output Low Voltage I/O ports OSC2/CLKOUT	VOL			0.6 0.6	V V	IOI = 8.7 mA, VDD = 4.5V IOI = 1.6 mA, VDD = 4.5V, PIC16C5X-RC
Output High Voltage I/O ports ⁽³⁾ OSC2/CLKOUT	VOH	VDD - 0.7 VDD - 0.7			V V	IOH = -5.4 mA, VDD = 4.5V IOH = -1.0 mA, VDD = 4.5V, PIC16C5X-RC

* These parameters are characterized but not tested.

Note 1: Data in the Typical ("Typ") column is based on characterization results at 25°C. This data is for design guidance only and is not tested.

- The leakage current on the \overline{MCLR}/VPP pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltage.
- Negative current is defined as coming out of the pin.
- For PIC16C5X-RC devices, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16C5X be driven with external clock in RC mode.
- The user may use the better of the two specifications.

10.5 DC Characteristics: PIC16C5X-RC, XT, 10, HS, LP (Automotive)

DC Characteristics All Pins Except Power Supply Pins		Standard Operating Conditions (unless otherwise specified) Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ Operating Voltage V_{DD} range is described in Section 10.1, Section 10.2 and Section 10.3.				
Characteristic	Sym	Min	Typ ⁽¹⁾	Max	Units	Conditions
Input Low Voltage I/O ports $\overline{\text{MCLR}}$ (Schmitt Trigger) T0CKI (Schmitt Trigger) OSC1 (Schmitt Trigger)	VIL	V_{SS} V_{SS} V_{SS} V_{SS}		0.15 V_{DD} 0.15 V_{DD} 0.15 V_{DD} 0.3 V_{DD}	V V V V	Pin at hi-impedance PIC16C5X-RC only ⁽⁴⁾ PIC16C5X-XT, 10, HS, LP
Input High Voltage I/O ports $\overline{\text{MCLR}}$ (Schmitt Trigger) T0CKI (Schmitt Trigger) OSC1 (Schmitt Trigger)	VIH	0.45 V_{DD} 2.0 0.36 V_{DD} 0.85 V_{DD} 0.85 V_{DD} 0.85 V_{DD} 0.7 V_{DD}		V_{DD} V_{DD} V_{DD} V_{DD} V_{DD} V_{DD}	V V V V V V	For all V_{DD} ⁽⁵⁾ $4.0\text{V} < V_{DD} \leq 5.5\text{V}$ ⁽⁵⁾ $V_{DD} > 5.5\text{V}$ PIC16C5X-RC only ⁽⁴⁾ PIC16C5X-XT, 10, HS, LP
Hysteresis of Schmitt Trigger inputs	VHYS	0.15 V_{DD} *			V	
Input Leakage Current ^(2,3) I/O ports $\overline{\text{MCLR}}$ T0CKI OSC1	IIL	-1 -5 -3 -3	0.5 0.5 0.5 0.5	+1 +5 +3 +3	μA μA μA μA	For $V_{DD} \leq 5.5\text{V}$ $V_{SS} \leq V_{PIN} \leq V_{DD}$, Pin at hi-impedance $V_{PIN} = V_{SS} + 0.25\text{V}$ $V_{PIN} = V_{DD}$ $V_{SS} \leq V_{PIN} \leq V_{DD}$ $V_{SS} \leq V_{PIN} \leq V_{DD}$, PIC16C5X-XT, 10, HS, LP
Output Low Voltage I/O ports OSC2/CLKOUT	VOL			0.6 0.6	V V	$I_{OL} = 8.7\text{ mA}$, $V_{DD} = 4.5\text{V}$ $I_{OL} = 1.6\text{ mA}$, $V_{DD} = 4.5\text{V}$, PIC16C5X-RC
Output High Voltage I/O ports ⁽³⁾ OSC2/CLKOUT	VOH	$V_{DD} - 0.7$ $V_{DD} - 0.7$			V V	$I_{OH} = -5.4\text{ mA}$, $V_{DD} = 4.5\text{V}$ $I_{OH} = -1.0\text{ mA}$, $V_{DD} = 4.5\text{V}$, PIC16C5X-RC

* These parameters are characterized but not tested.

Note 1: Data in the Typical ("Typ") column is based on characterization results at 25°C. This data is for design guidance only and is not tested.

- The leakage current on the $\overline{\text{MCLR}}/V_{PP}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltage.
- Negative current is defined as coming out of the pin.
- For PIC16C5X-RC devices, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16C5X be driven with external clock in RC mode.
- The user may use the better of the two specifications.

11.0 DC AND AC CHARACTERISTICS - PIC16C54/55/56/57

The graphs and tables provided in this section are for design guidance and are not tested or guaranteed. In some graphs or tables the data presented are outside specified operating range (e.g., outside specified VDD range). This is for information only and devices will operate properly only within the specified range.

The data presented in this section is a statistical summary of data collected on units from different lots over a period of time. "Typical" represents the mean of the distribution while "max" or "min" represents (mean + 3σ) and (mean - 3σ) respectively, where σ is standard deviation.

FIGURE 11-1: TYPICAL RC OSCILLATOR FREQUENCY vs. TEMPERATURE

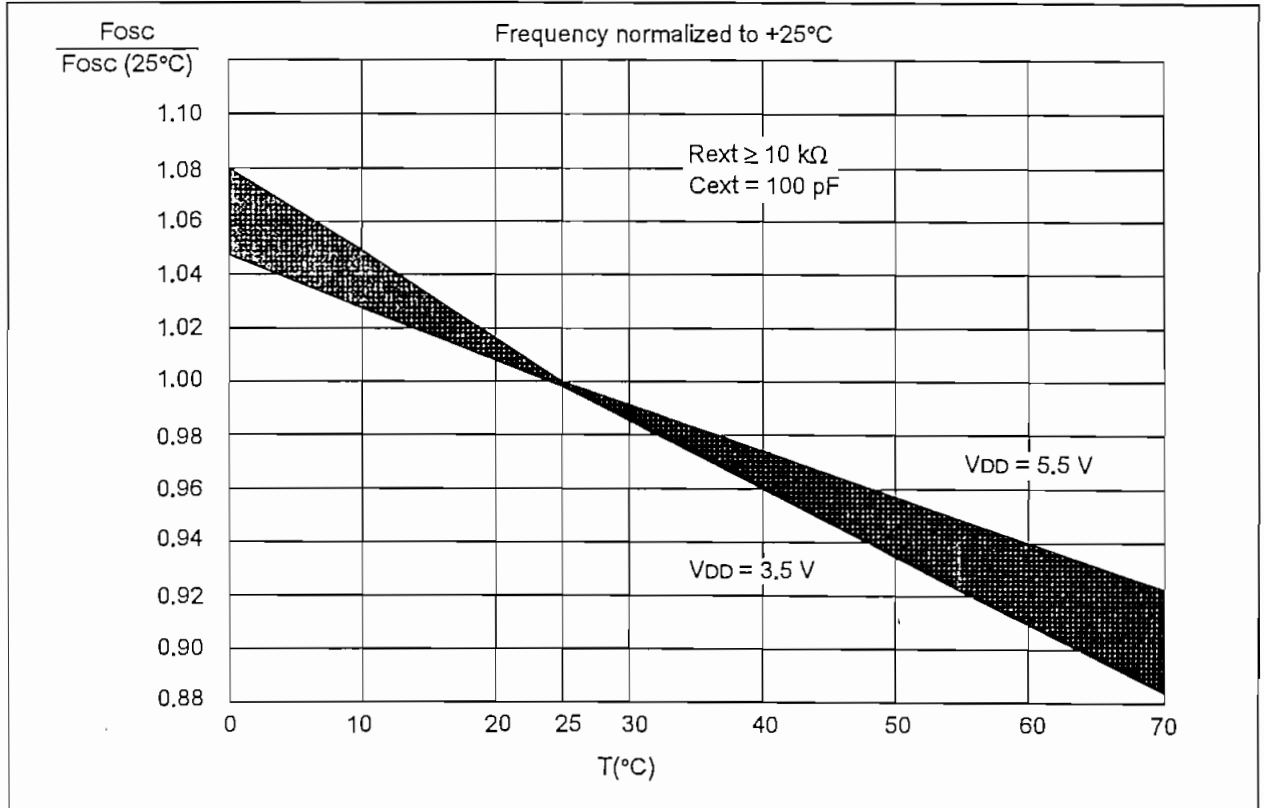


TABLE 11-1: RC OSCILLATOR FREQUENCIES

Cext	Rext	Average Fosc @ 5 V, 25°C	
		Frequency	Percentage Variation
20 pF	3.3 k	4.973 MHz	± 27%
	5 k	3.82 MHz	± 21%
	10 k	2.22 MHz	± 21%
	100 k	262.15 kHz	± 31%
100 pF	3.3 k	1.63 MHz	± 13%
	5 k	1.19 MHz	± 13%
	10 k	684.64 kHz	± 18%
	100 k	71.56 kHz	± 25%
300 pF	3.3 k	660 kHz	± 10%
	5.0 k	484.1 kHz	± 14%
	10 k	267.63 kHz	± 15%
	160 k	29.44 kHz	± 19%

The frequencies are measured on DIP packages.

The percentage variation indicated here is part-to-part variation due to normal process distribution. The variation indicated is ±3 standard deviation from average value for VDD = 5 V.

FIGURE 11-2: TYPICAL RC OSCILLATOR FREQUENCY vs. V_{DD} , $C_{EXT} = 20\text{pF}$

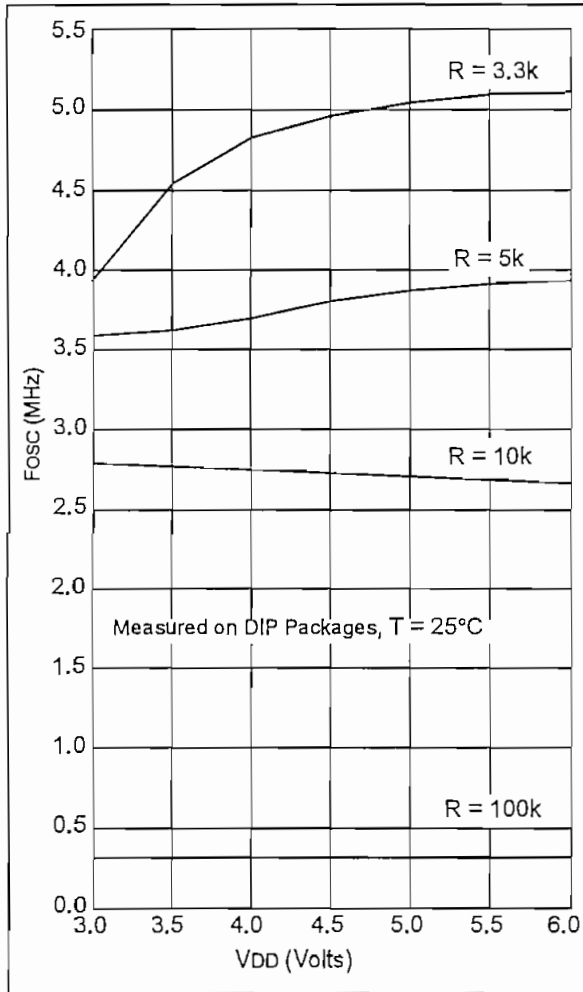


FIGURE 11-3: TYPICAL RC OSCILLATOR FREQUENCY vs. V_{DD} , $C_{EXT} = 100\text{pF}$

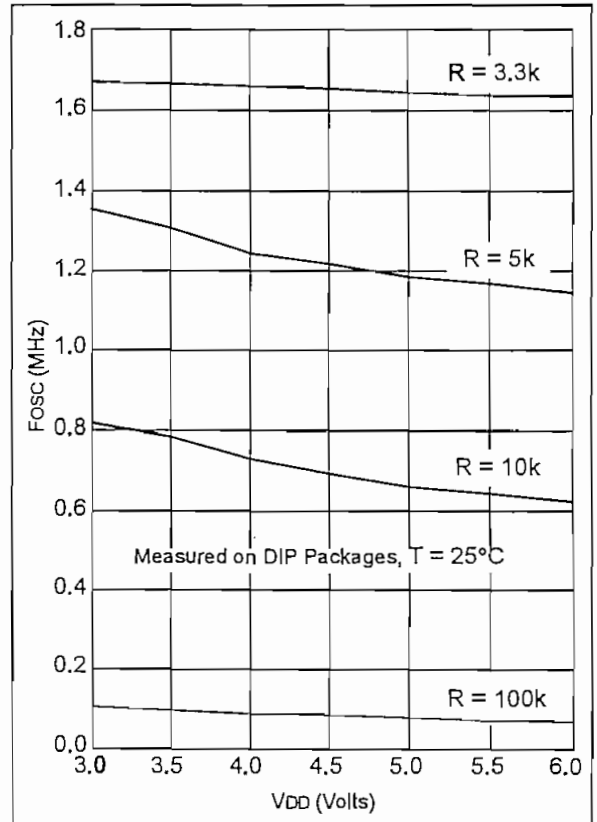


FIGURE 11-4: TYPICAL RC OSCILLATOR FREQUENCY vs. V_{DD} , $C_{EXT} = 300\text{pF}$

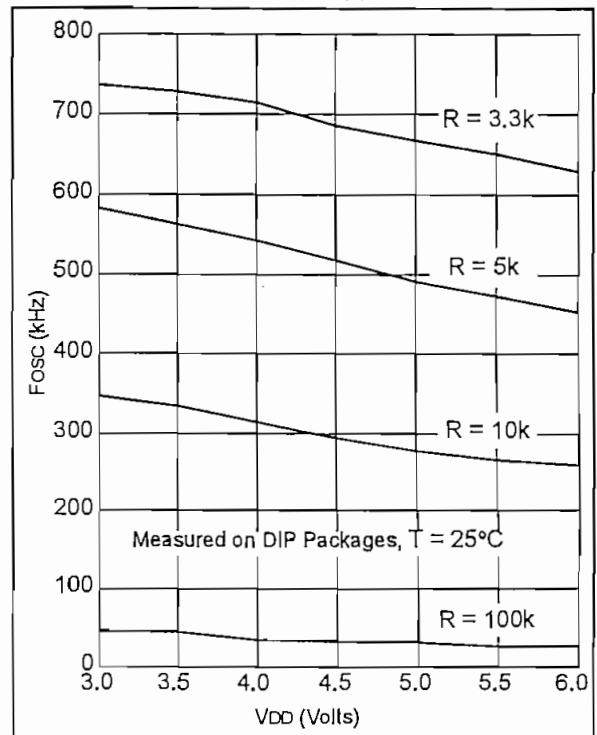


FIGURE 11-5: TYPICAL I_{PD} vs. V_{DD} , WATCHDOG DISABLED

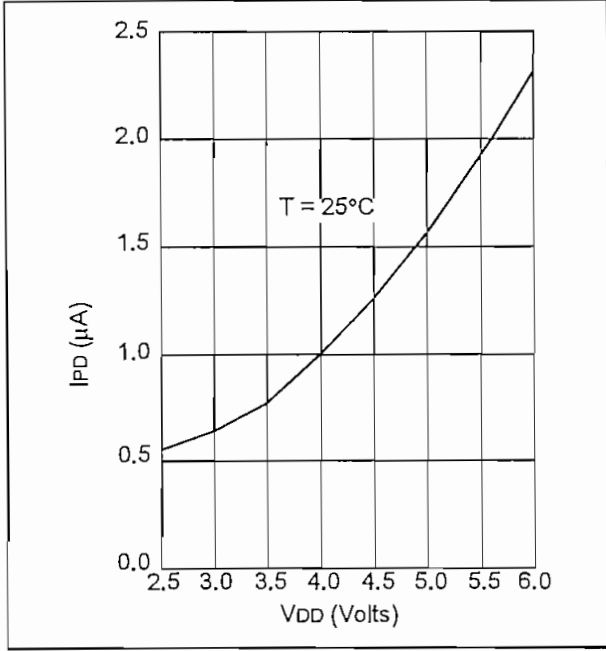


FIGURE 11-7: TYPICAL I_{PD} vs. V_{DD} , WATCHDOG ENABLED

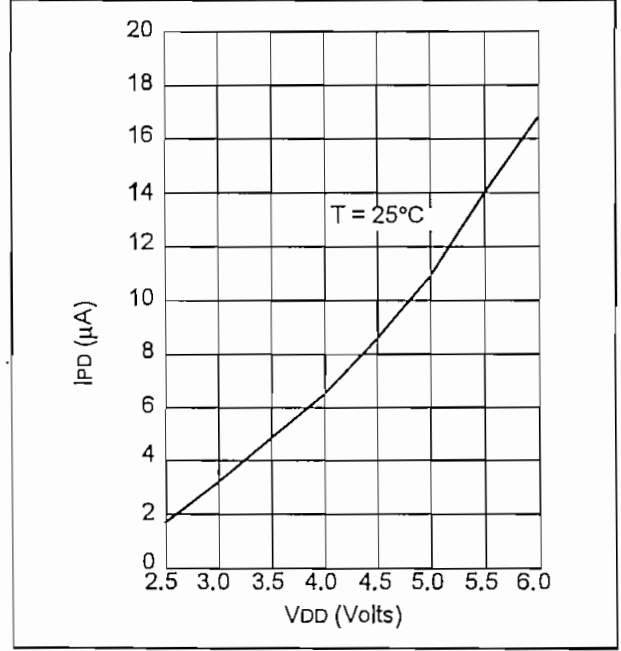


FIGURE 11-6: MAXIMUM I_{PD} vs. V_{DD} , WATCHDOG DISABLED

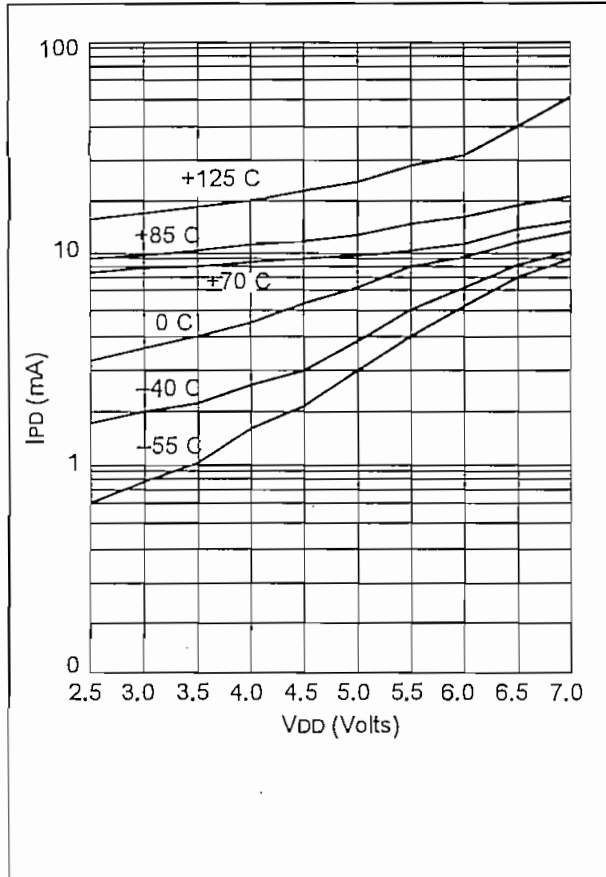


FIGURE 11-8: MAXIMUM I_{PD} vs. V_{DD} , WATCHDOG ENABLED

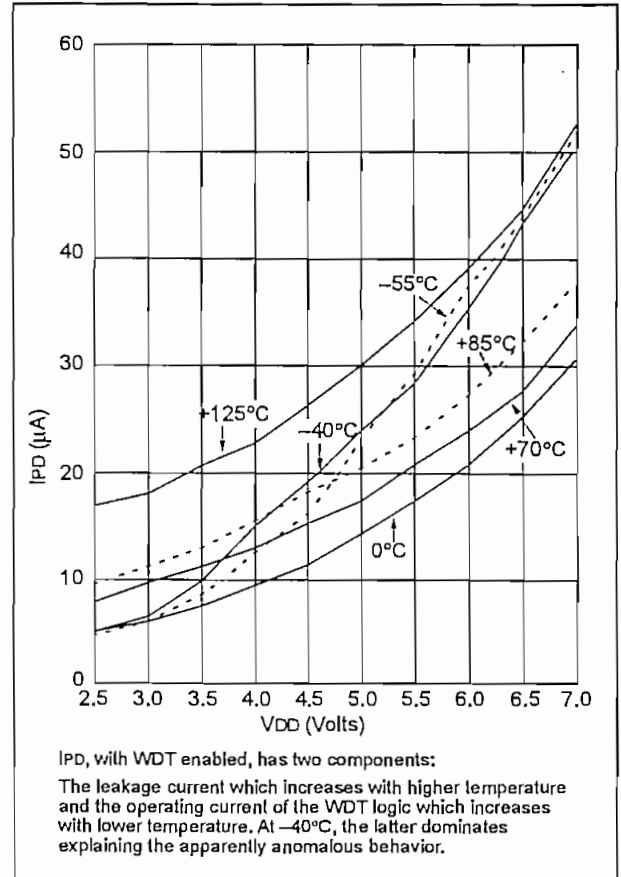


FIGURE 11-9: V_{TH} (INPUT THRESHOLD VOLTAGE) OF I/O PINS vs. V_{DD}

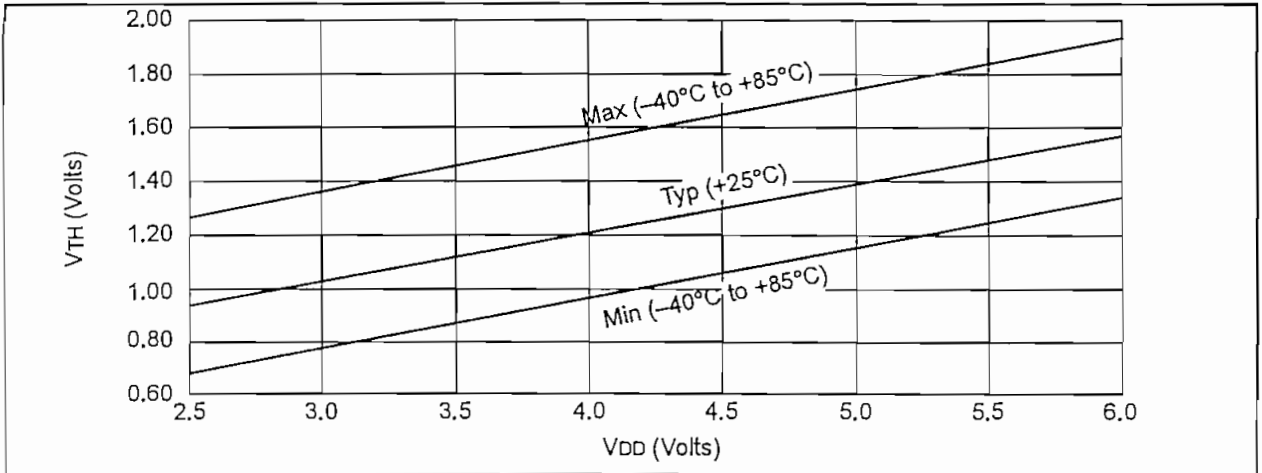


FIGURE 11-10: V_{IH} , V_{IL} OF MCLR, T0CKI AND OSC1 (IN RC MODE) vs. V_{DD}

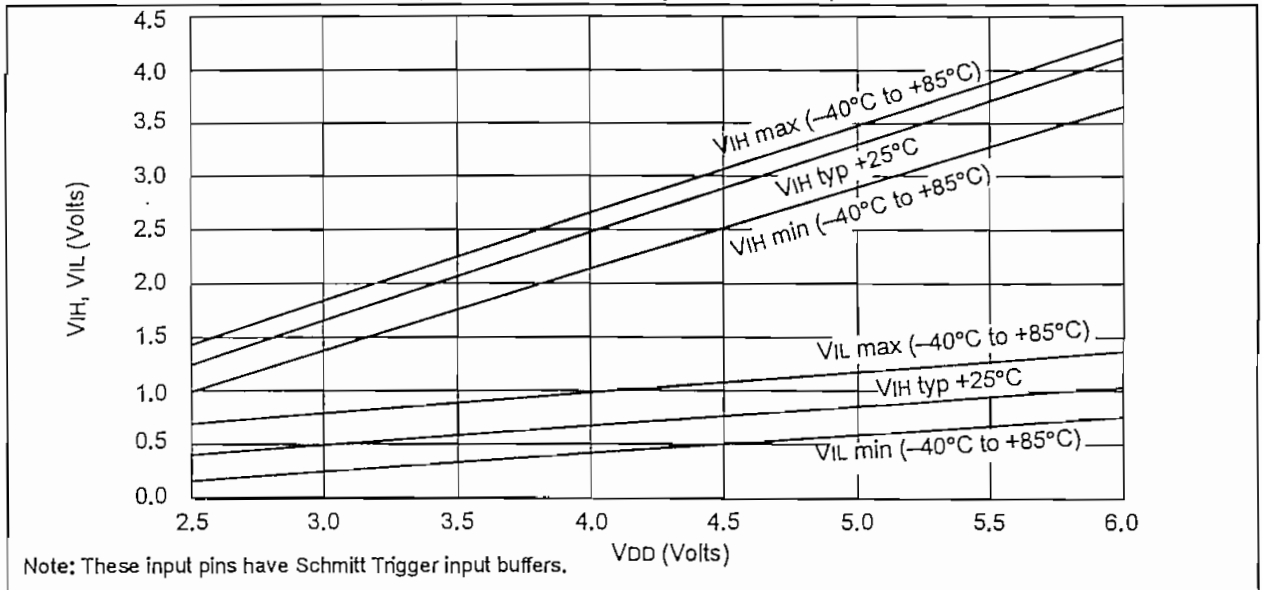


FIGURE 11-11: V_{TH} (INPUT THRESHOLD VOLTAGE) OF OSC1 INPUT (IN XT, HS, AND LP MODES) vs. V_{DD}

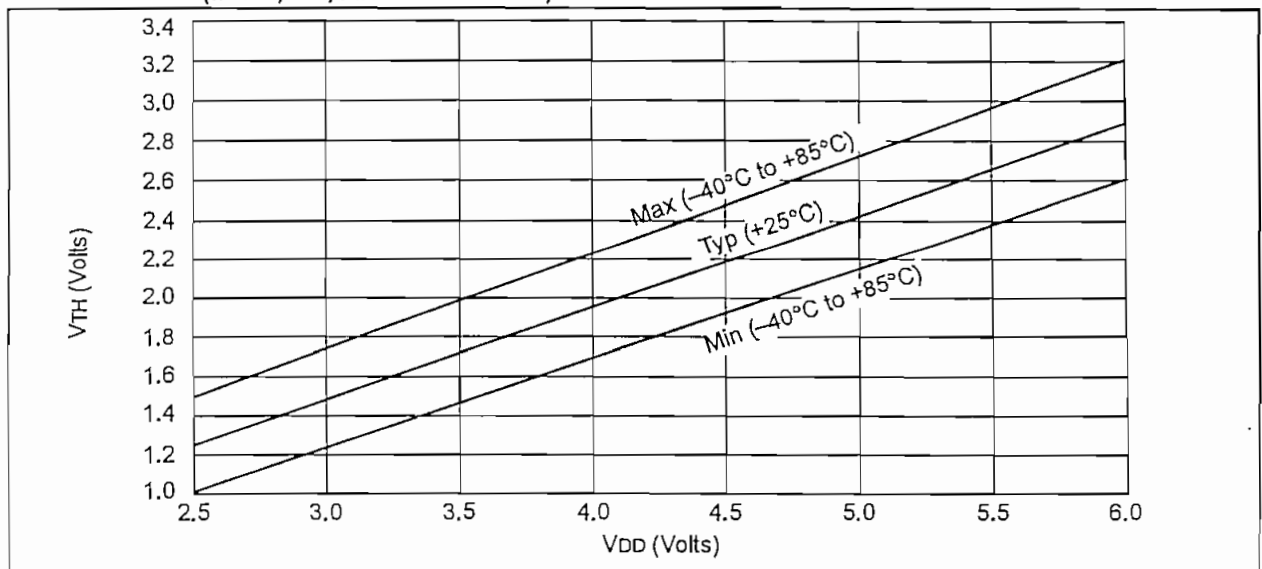


FIGURE 11-12: TYPICAL I_{DD} vs. FREQUENCY (EXTERNAL CLOCK, 25°C)

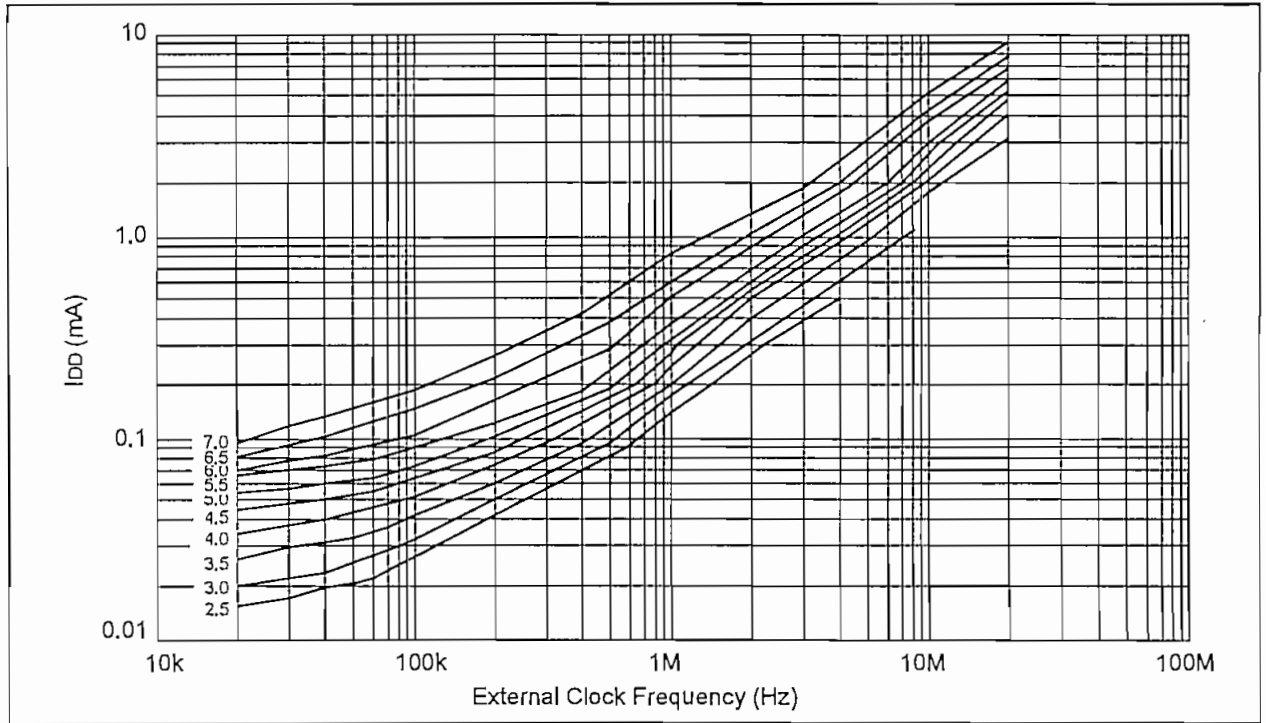


FIGURE 11-13: MAXIMUM I_{DD} vs. FREQUENCY (EXTERNAL CLOCK, -40°C TO +85°C)

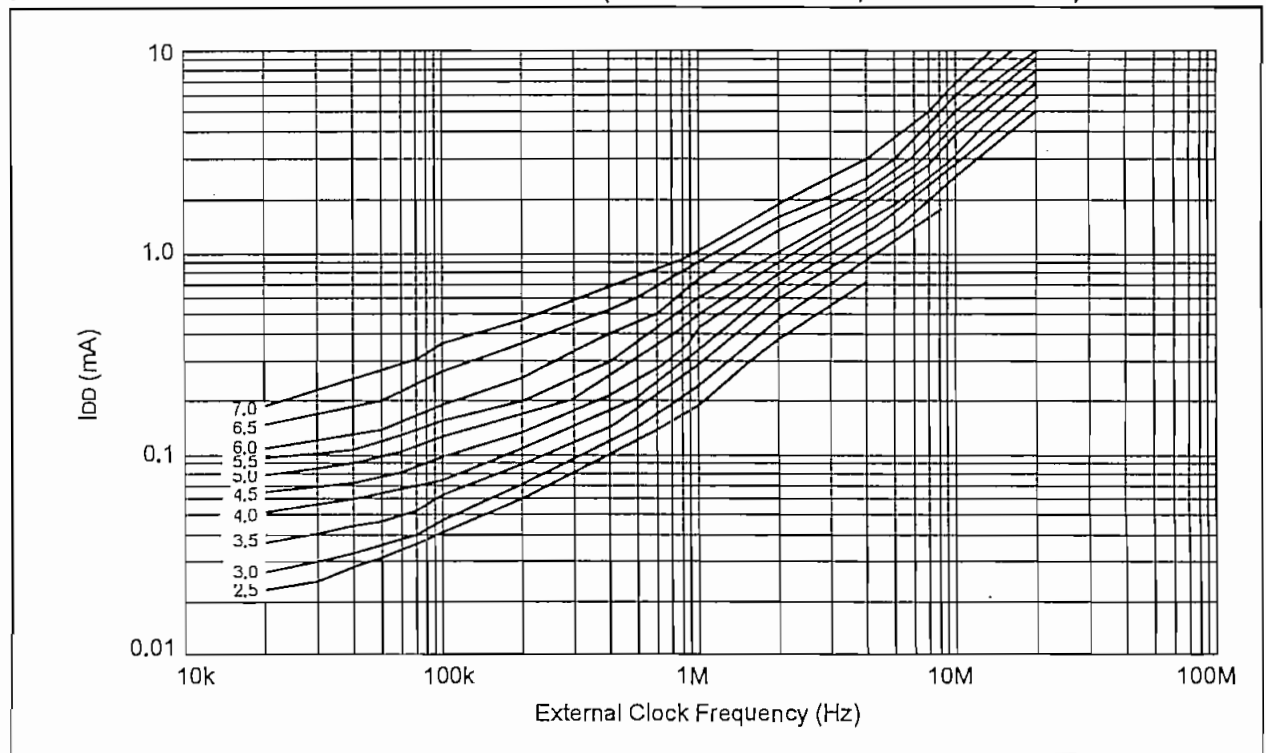


FIGURE 11-14: MAXIMUM I_{DD} vs. FREQUENCY (EXTERNAL CLOCK -55°C TO +125°C)

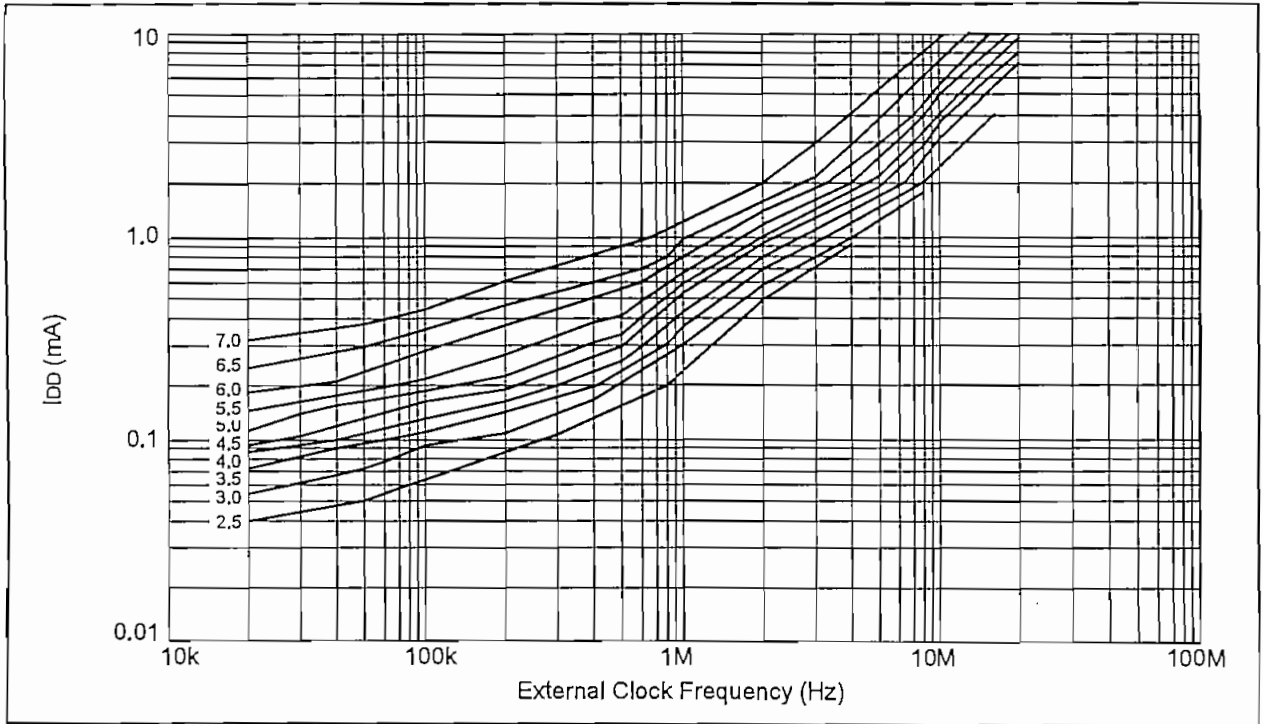


FIGURE 11-15: WDT TIMER TIME-OUT PERIOD vs. V_{DD}

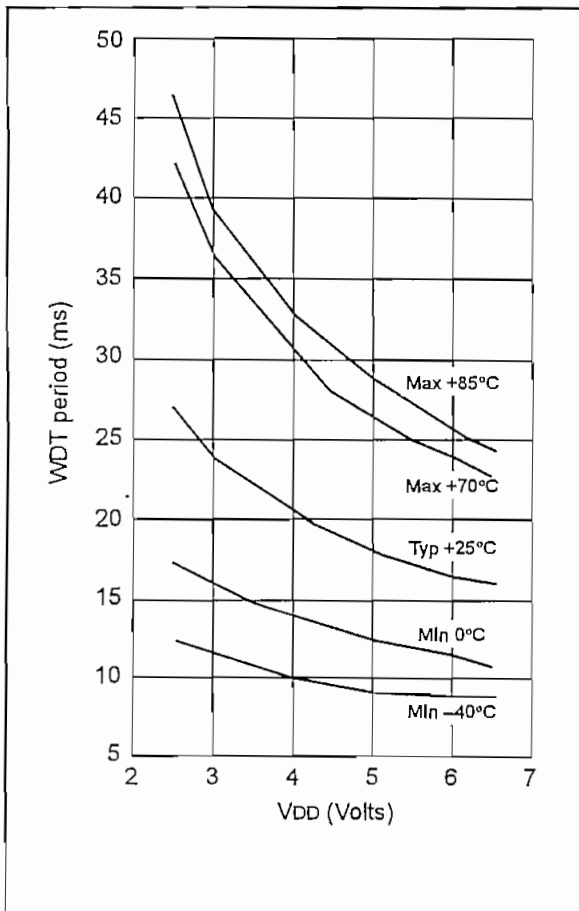


FIGURE 11-16: TRANSCONDUCTANCE (gm) OF HS OSCILLATOR vs. V_{DD}

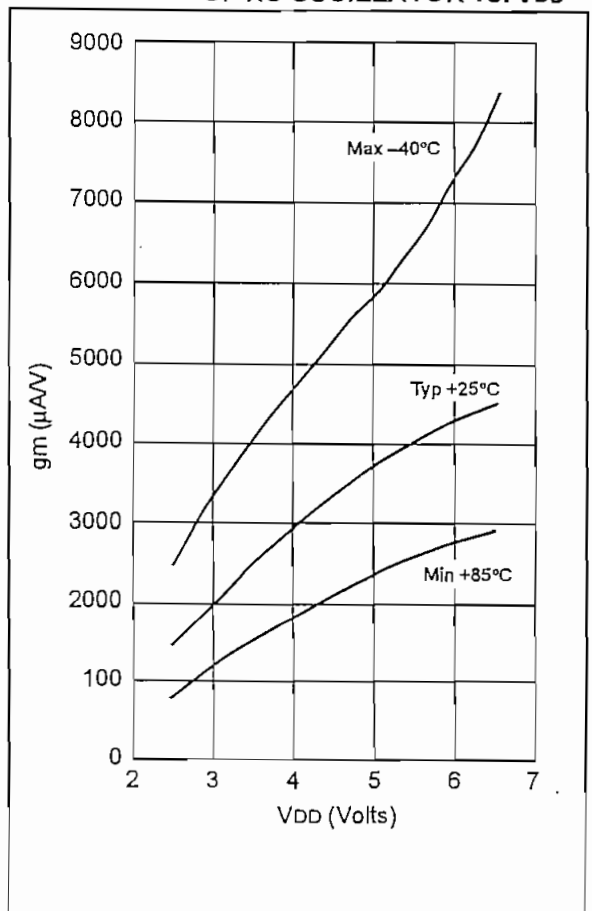


FIGURE 11-17: TRANSCONDUCTANCE (gm) OF LP OSCILLATOR vs. VDD

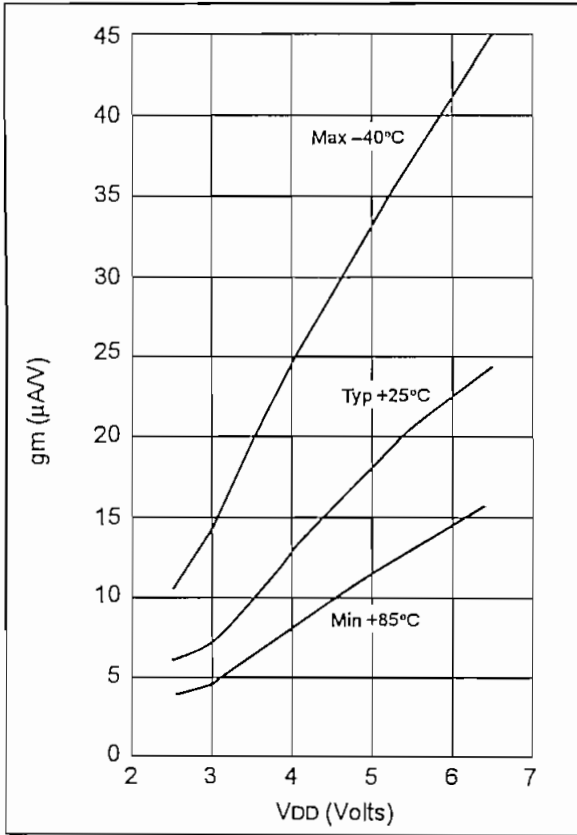


FIGURE 11-19: TRANSCONDUCTANCE (gm) OF XT OSCILLATOR vs. VDD

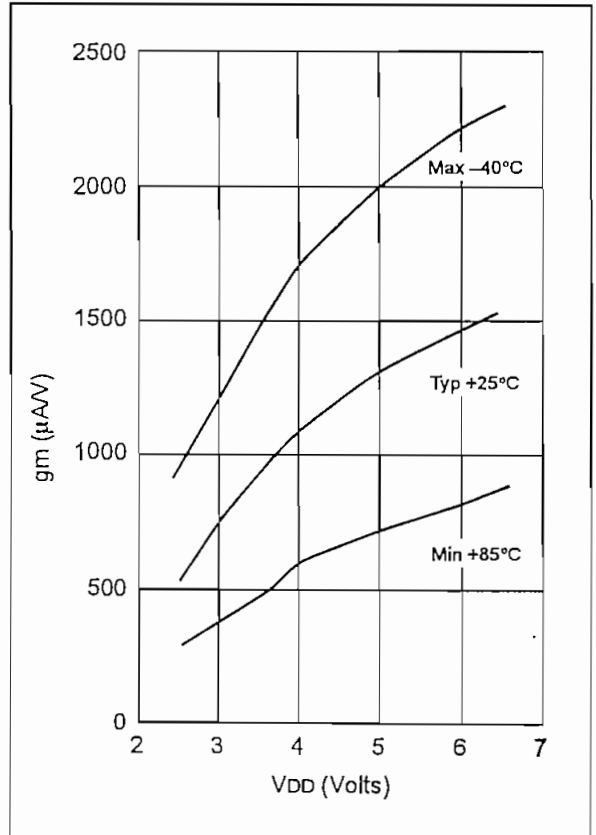


FIGURE 11-18: IOH vs. VOH, VDD = 3 V

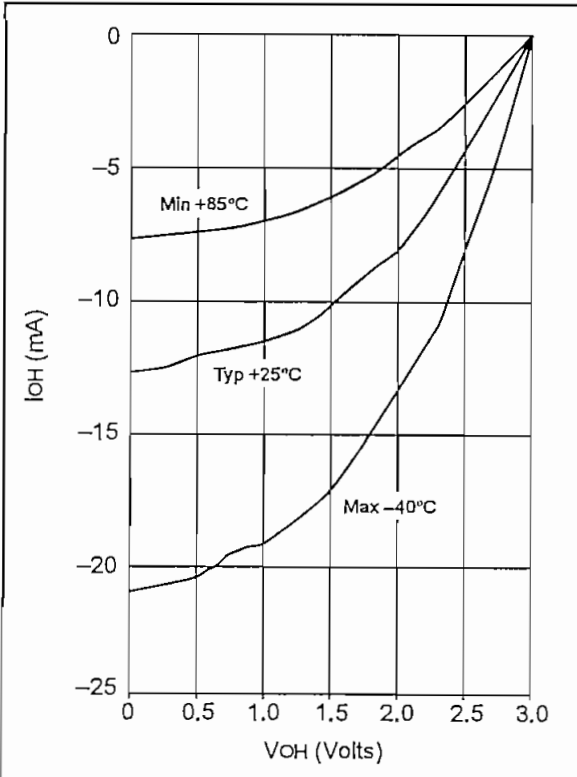


FIGURE 11-20: IOH vs. VOH, VDD = 5 V

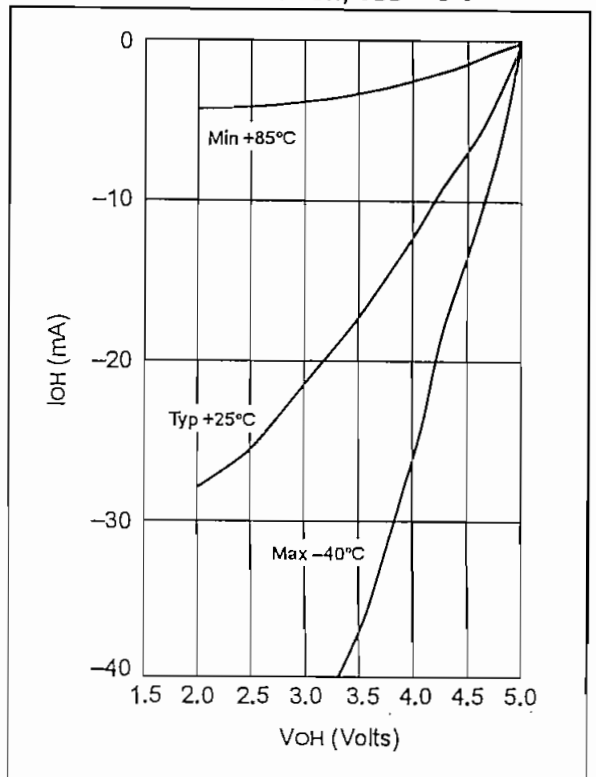


FIGURE 11-21: I_{OL} vs. V_{OL}, V_{DD} = 3 V

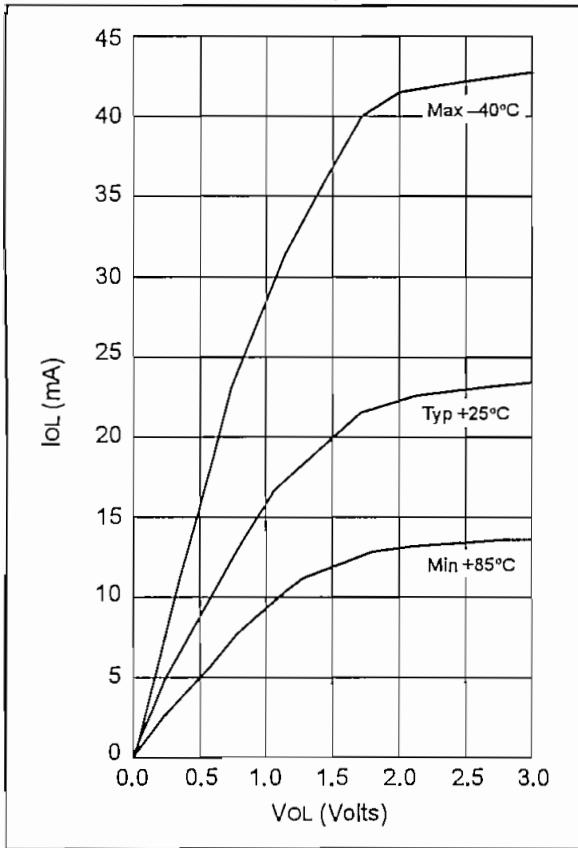


FIGURE 11-22: I_{OL} vs. V_{OL}, V_{DD} = 5 V

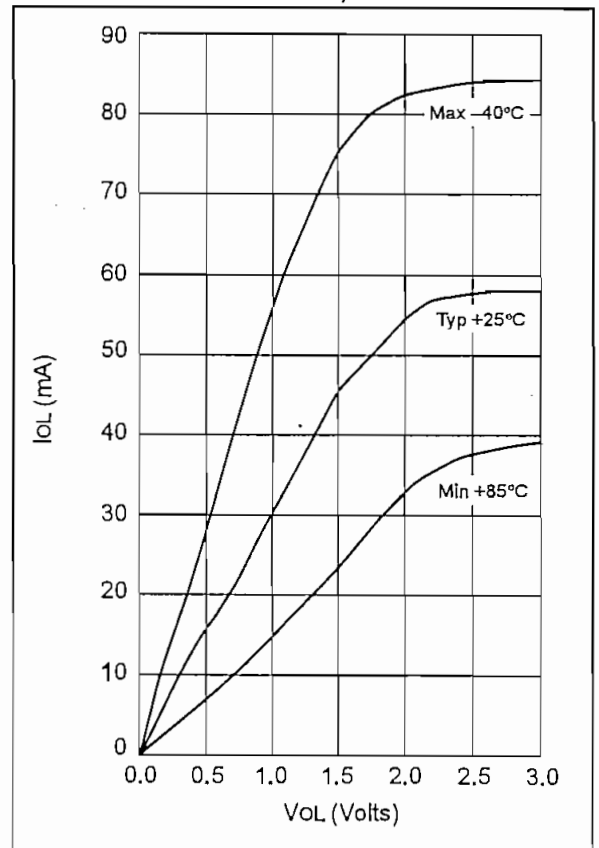


TABLE 11-2: INPUT CAPACITANCE FOR PIC16C54/56

Pin	Typical Capacitance (pF)	
	18L PDIP	18L SOIC
RA port	5.0	4.3
RB port	5.0	4.3
MCLR	17.0	17.0
OSC1	4.0	3.5
OSC2/CLKOUT	4.3	3.5
T0CKI	3.2	2.8

All capacitance values are typical at 25°C. A part-to-part variation of ±25% (three standard deviations) should be taken into account.

TABLE 11-3: INPUT CAPACITANCE FOR PIC16C55/57

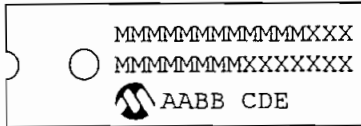
Pin	Typical Capacitance (pF)	
	28L PDIP (600 mil)	28L SOIC
RA port	5.2	4.8
RB port	5.6	4.7
RC port	5.0	4.1
MCLR	17.0	17.0
OSC1	6.6	3.5
OSC2/CLKOUT	4.6	3.5
T0CKI	4.5	3.5

All capacitance values are typical at 25°C. A part-to-part variation of ±25% (three standard deviations) should be taken into account.

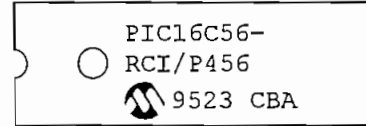
13.0 PACKAGING INFORMATION

13.1 Package Marking Information

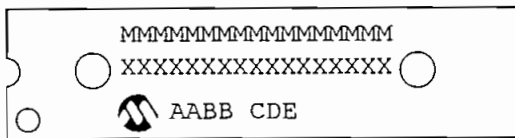
18-Lead PDIP



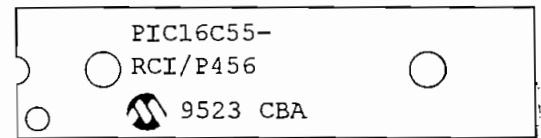
Example



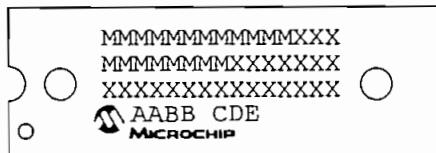
28-Lead Skinny PDIP (.300")



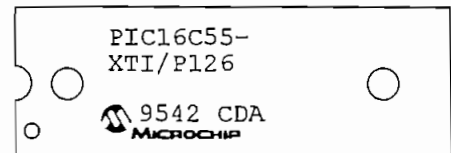
Example



28-Lead PDIP (.600")



Example

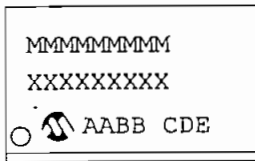


Legend:	MM...M	Microchip part number information
	XX...X	Customer specific information*
	AA	Year code (last two digits of calendar year)
	BB	Week code (week of January 1 is week '01')
	C	Facility code of the plant at which wafer is manufactured C = Chandler, Arizona, U.S.A., S = Tempe, Arizona, U.S.A.
	D	Mask revision number
	E	Assembly code of the plant or country of origin in which part was assembled
Note:	In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.	

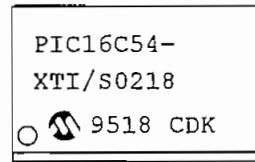
* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask rev#, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

PIC16C5X

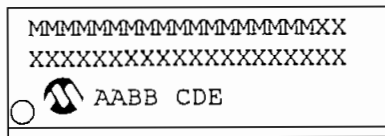
18-Lead SOIC



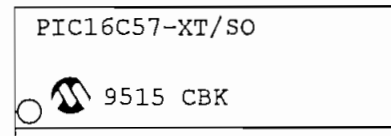
Example



28-Lead SOIC



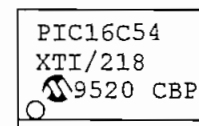
Example



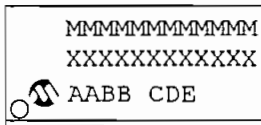
20-Lead SSOP



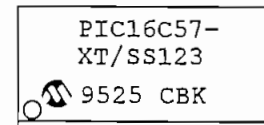
Example



28-Lead SSOP



Example

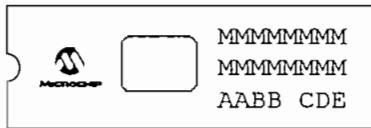


Legend: MM...M	Microchip part number information
XX...X	Customer specific information*
AA	Year code (last two digits of calendar year)
BB	Week code (week of January 1 is week '01')
C	Facility code of the plant at which wafer is manufactured C = Chandler, Arizona, U.S.A., S = Tempe, Arizona, U.S.A.
D	Mask revision number
E	Assembly code of the plant or country of origin in which part was assembled

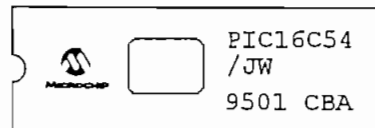
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.

* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask rev#, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

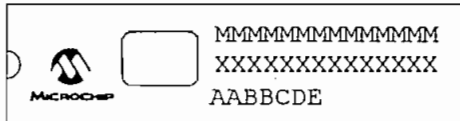
18-Lead CERDIP Windowed



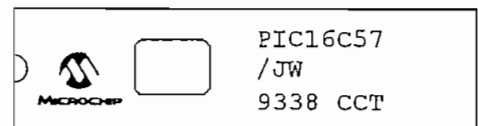
Example



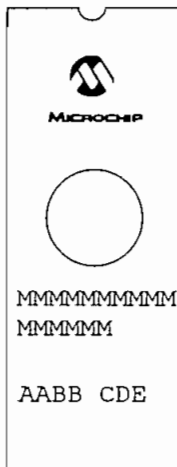
28-Lead CERDIP Skinny Windowed



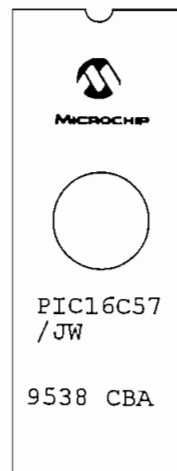
Example



28-Lead CERDIP Windowed



Example

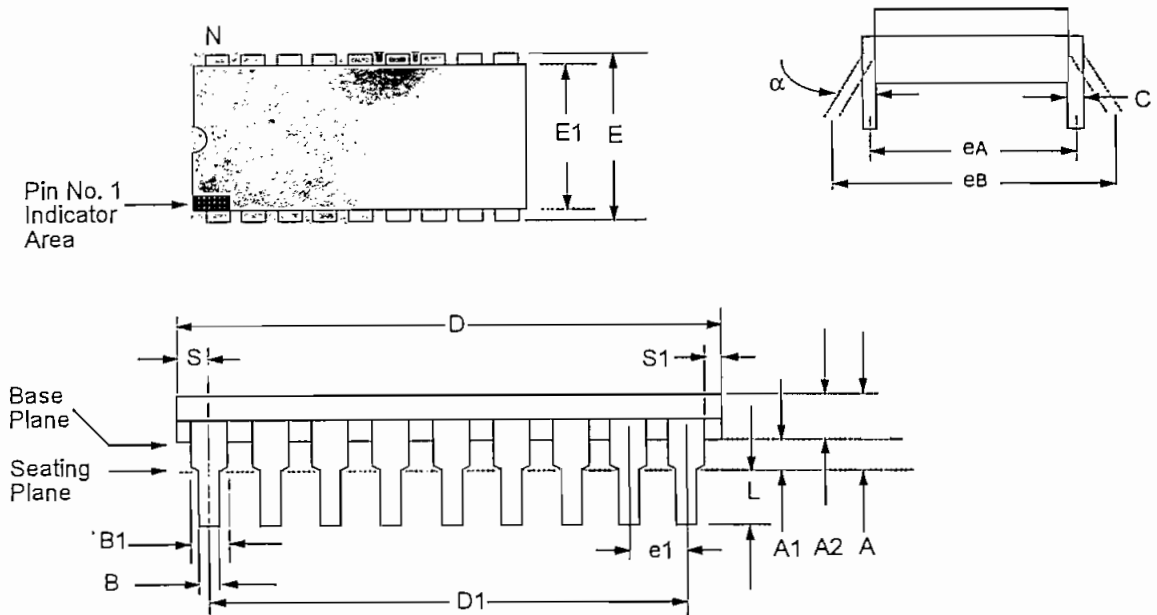


Legend: MM...M	Microchip part number information
XX...X	Customer specific information*
AA	Year code (last two digits of calendar year)
BB	Week code (week of January 1 is week '01')
C	Facility code of the plant at which wafer is manufactured C = Chandler, Arizona, U.S.A., S = Tempe, Arizona, U.S.A.
D	Mask revision number
E	Assembly code of the plant or country of origin in which part was assembled
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.	

* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask rev#, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

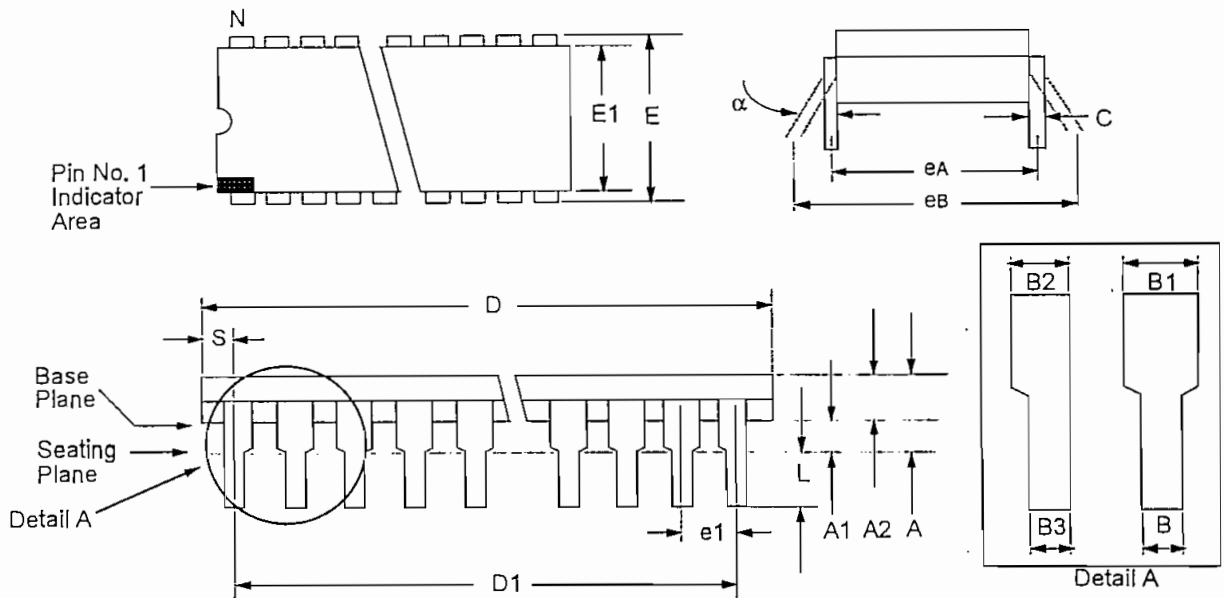
PIC16C5X

13.2 18-Lead Plastic Dual In-Line (PDIP) - 300 mil



Package Group: Plastic Dual In-Line (PLA)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	—	4,064		—	0,160	
A1	0,381	—		0,015	—	
A2	3,048	3,810		0,120	0,150	
B	0,355	0,559		0,014	0,022	
B1	1,524	1,524	Reference	0,060	0,060	Reference
C	0,203	0,381	Typical	0,008	0,015	Typical
D	22,479	23,495		0,885	0,925	
D1	20,320	20,320	Reference	0,800	0,800	Reference
E	7,620	8,255		0,300	0,325	
E1	6,096	7,112		0,240	0,280	
e1	2,489	2,591	Typical	0,098	0,102	Typical
eA	7,620	7,620	Reference	0,300	0,300	Reference
eB	7,874	9,906		0,310	0,390	
L	3,048	3,556		0,120	0,140	
N	18	18		18	18	
S	0,889	—		0,035	—	
S1	0,127	—		0,005	—	

13.3 28-Lead Plastic Dual In-Line (PDIP) - 300 mil



Package Group: Plastic Dual In-Line (PLA)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	3.632	4.572		0.143	0.180	
A1	0.381	—		0.015	—	
A2	3.175	3.556		0.125	0.140	
B	0.406	0.559		0.016	0.022	
B1	1.016	1.651	Typical	0.040	0.065	Typical
B2	0.762	1.016	4 places	0.030	0.040	4 places
B3	0.203	0.508	4 places	0.008	0.020	4 places
C	0.203	0.331	Typical	0.008	0.013	Typical
D	34.163	35.179		1.385	1.395	
D1	33.020	33.020	Reference	1.300	1.300	Reference
E	7.874	8.382		0.310	0.330	
E1	7.112	7.493		0.280	0.295	
e1	2.540	2.540	Typical	0.100	0.100	Typical
eA	7.874	7.874	Reference	0.310	0.310	Reference
eB	8.128	9.652		0.320	0.380	
L	3.175	3.683		0.125	0.145	
N	28	—		28	—	
S	0.584	1.220		0.023	0.048	

EPROM/ROM Memory Programming/Verify Specification

4.0 PROGRAM/VERIFY MODE ELECTRICAL CHARACTERISTICS

4.1 DC Program Characteristics

TABLE 4-1: DC CHARACTERISTICS (TA = +10°C TO +40°C) (25°C IS RECOMMENDED)

Parameter No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
PD1	VDDP	Supply voltage during programming	4.75	5.0	5.25	V	Note 1
PD2	IDDP	Supply Current (from VDD)			25.0	mA	VDD = 5.0V, Fosc1 = 5MHz
PD3	VDDV	Supply Voltage during verify	VDDmin		VDDmax		
PD4	VHH1	Voltage on MCLR to stay in Program/Verify Modes	VDD + 3		VDD+7.25	V	
PD5	VHH2	Voltage on MCLR during programming	12.5		13.5	V	
PD6	IHH	Supply current from programming voltage source			100	mA	
PD7	IHH2	Current into MCLR pin during programming (TOCKI=0)		10.0	25.0	mA	VHH = 13.5V, VDD = 6.0V
PD8	VIL	Input Low Voltage	Vss		0.15VDD	V	
PD9	VIH	Input High Voltage	0.85VDD	5.0	VDD	V	

Note 1: Device must be verified at minimum and maximum operating voltages specified in the data sheet.

4.2 AC Program and Test Mode Characteristics

TABLE 4-2: AC CHARACTERISTICS (TA = +10°C TO +40°C, VDD = 5.0V ± 5%) (25°C IS RECOMMENDED)

Parameter No.	Symbol	Characteristics	Min	Typ	Max	Units	Conditions
P1	TR	MCLR Rise Time	0.15	1.0	8	μs	
P2	TF	MCLR Fall Time	0.5	2.0	8	μs	
P3	TPS	Program Mode Setup Time	1.0			μs	
P4	TACC	Data Access Time			250	ns	
P5	TDS	Data Setup Time	1.0			μs	
P6	TDH	Data Hold Time	1.0			μs	
P7	TOE	Output Enable Time	0		100	ns	
P8	TOZ	Output Disable Time	0		100	ns	
P9	TPW	Programming Pulse Width	10.0	100		μs	
P10	TPWF	Programming Pulse Width		10,000		μs	Configuration Word only
P11	TRC	Recovery Time	10.0			μs	
P12	FOSC	Frequency on OSC1	DC		5	MHz	For incrementing of the PC

PIC16C5X

FIGURE 4-1: PROGRAMMING AND VERIFY TIMING WAVEFORM

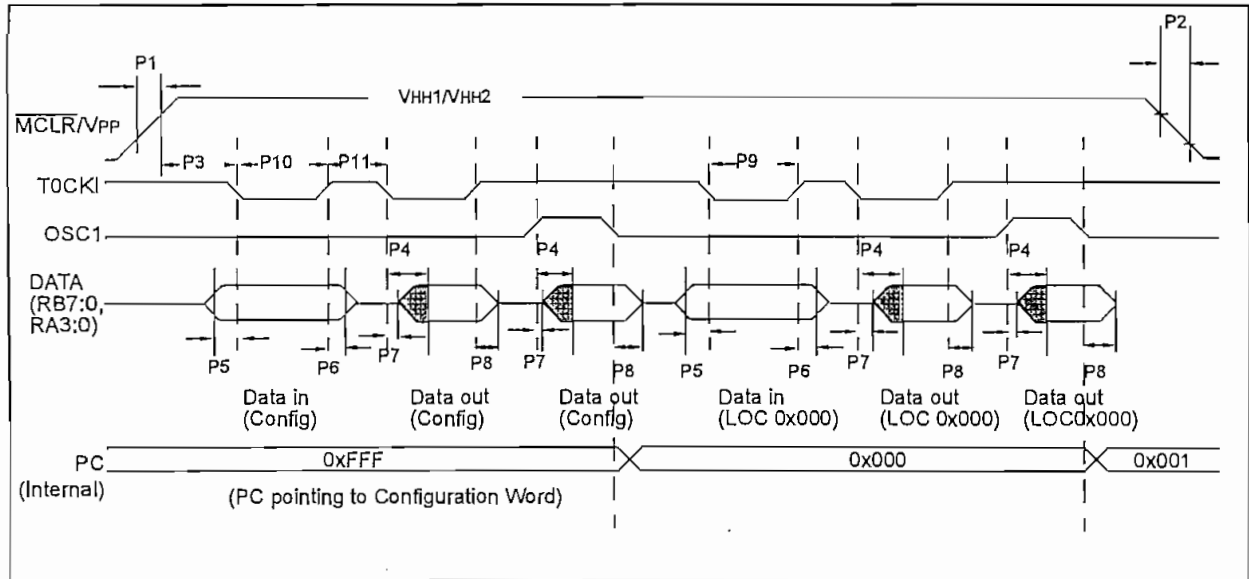
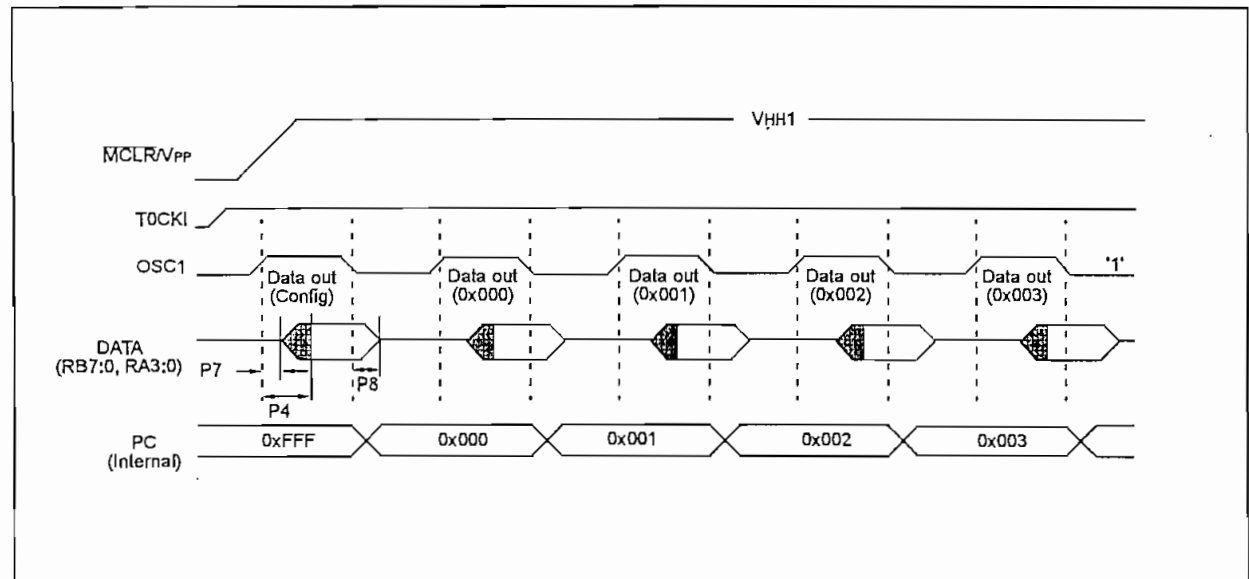


FIGURE 4-2: SPEED VERIFY TIMING WAVEFORM



PIN COMPATIBILITY

Devices that have the same package type and VDD, Vss and \overline{MCLR} pin locations are said to be pin compatible. This allows these different devices to operate in the same socket. Compatible devices may only require minor software modification to allow proper operation in the application socket (ex., PIC16C56 and PIC16C61 devices). Not all devices in the same package size are pin compatible; for example, the PIC16C62 is compatible with the PIC16C63, but not the PIC16C55.

Pin compatibility does not mean that the devices offer the same features. As an example, the PIC16C54 is pin compatible with the PIC16C71, but does not have an A/D converter, weak pull-ups on PORTB, or interrupts.

TABLE D-1: PIN COMPATIBLE DEVICES

Pin Compatible Devices	Package
PIC12C508, PIC12C509	8-pin
PIC16C54, PIC16C54A, PIC16CR54A, PIC16C56, PIC16C58A, PIC16CR58A, PIC16C61, PIC16C554, PIC16C556, PIC16C558 PIC16C620, PIC16C621, PIC16C622, PIC16C710, PIC16C71, PIC16C711, PIC16F83, PIC16CR83, PIC16C84, PIC16F84A, PIC16CR84	18-pin 20-pin
PIC16C55, PIC16C57, PIC16CR57B	28-pin
PIC16C62, PIC16CR62, PIC16C62A, PIC16C63, PIC16C72, PIC16C73, PIC16C73A	28-pin
PIC16C64, PIC16CR64, PIC16C64A, PIC16C65, PIC16C65A, PIC16C74, PIC16C74A	40-pin
PIC17C42, PIC17CR42, PIC17C42A, PIC17C43, PIC17CR43, PIC17C44	40-pin
PIC16C923, PIC16C924	64/68-pin

ANEXO 2
CARACTERISTICAS DE LOS CHIPS UTILIZADOS
EN EL PROGRAMADOR Y DIAGRAMAS DE LOS
IMPRESOS

TYPES SN54L03
QUADRUPLE 2-INPUT POSITIVE-NAND GATES WITH OPEN-COLLECTOR OUTPUTS

recommended operating conditions

PARAMETER	TEST CONDITIONS	SN54L03	MIN	TYP	MAX
V _{CC} Supply voltage		4.5	5	5.5	V
V _{IH} High-level input voltage		2			V
V _{IL} Low-level input voltage					V
V _{OH} High-level output voltage					V
I _{OL} Low-level output current					mA
T _A Operating free-air temperature		-15			125

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
V _{IH}	V _{CC} = MIN, V _L = 0.6 V, V _{OH} = 5.5 V	2			V
V _{OL}	V _{CC} = MIN, V _{IH} = 2 V, I _{OL} = 2 mA	0.15	0.4	0.7	V
I _I	V _{CC} = MAX, V _I = 5.5 V		0.1		mA
I _{IH}	V _{CC} = MAX, V _I = 2.4 V		10		μA
I _{IL}	V _{CC} = MAX, V _I = 0.3 V		-0.18		μA
I _{OCH}	V _{CC} = MAX, V _I = 0 V		0.44	0.8	mA
I _{OCL}	V _{CC} = MAX, V _I = 4.5 V		1.18	2.04	mA

For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions. All typical values are at V_{CC} = 5 V, T_A = 25°C.

switching characteristics, V_{CC} = 5 V, T_A = 25°C (see note 2)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t _{PHL}	A or B	Y	R _L = 4 kΩ, C _L = 50 pF	90	90	90	ns
t _{PHL}	A or B	Y	R _L = 4 kΩ, C _L = 50 pF	30	30	30	ns

NOTE 2: See General Information Section for load circuits and voltage waveforms.

QUADRUPLE 2-INPUT POSITIVE-NAND GATES WITH OPEN-COLLECTOR OUTPUTS

recommended operating conditions

PARAMETER	TEST CONDITIONS	SN54L03	MIN	TYP	MAX	UNIT
V _{CC} Supply voltage		4.5	5	5.5	V	
V _{IH} High-level input voltage		2			V	
V _{IL} Low-level input voltage					V	
V _{OH} High-level output voltage					V	
I _{OL} Low-level output current					mA	
T _A Operating free-air temperature		-55			125	

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
V _{IH}	V _{CC} = MIN, I _I = 18 mA, V _{OH} = 5.5 V	2			V
V _{OL}	V _{CC} = MIN, V _{IH} = 2 V, I _{OL} = 4 mA	0.25	0.4	0.7	V
I _I	V _{CC} = MIN, V _{IH} = 2 V, I _{OL} = 8 mA		0.1		mA
I _{IH}	V _{CC} = MAX, V _I = 7 V		20		μA
I _{IL}	V _{CC} = MAX, V _I = 2.7 V		-0.4		μA
I _{OCH}	V _{CC} = MAX, V _I = 0.4 V		0.8	1.6	mA
I _{OCL}	V _{CC} = MAX, V _I = 4.5 V		2.4	4.4	mA

For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions. All typical values are at V_{CC} = 5 V, T_A = 25°C.

switching characteristics, V_{CC} = 5 V, T_A = 25°C (see note 2)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t _{PHL}	A or B	Y	R _L = 2 kΩ, C _L = 15 pF	17	17	17	ns
t _{PHL}	A or B	Y	R _L = 2 kΩ, C _L = 15 pF	15	15	15	ns

NOTE 2: See General Information Section for load circuits and voltage waveforms.

QUADRUPEL 2-INPUT POSITIVE-NAND GATES WITH OPEN-COLLECTOR OUTPUTS

recommended operating conditions

PARAMETER	SN54S04		SN74S04	
	MIN	MAX	MIN	MAX
V _{CC} Supply voltage	4.5	5.5	4.75	5.25
V _{IH} High-level input voltage	2	3	2	3
V _{IL} Low-level input voltage	0.8	1.5	0.8	1.5
V _{OH} High-level output voltage	0.8	1.5	0.8	1.5
I _{OL} Low-level output current	20	20	20	20
T _A Operating free-air temperature	-55	125	0	70

Electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS	MIN (TYP) MAX	
		MIN	MAX
V _{IH} High-level input voltage	V _{CC} - MIN, I _I = 0.8 mA	2	3
V _{IL} Low-level input voltage	V _{CC} - MIN, V _{IC} = 0.8 V, I _I = 0.5 V	0.8	1.5
V _{OH} High-level output voltage	V _{CC} - MIN, V _{IH} = 2 V, I _{OL} = 20 mA	0.8	1.5
I _{OL} Low-level output current	V _{CC} - MAX, V _I = 0.5 V	0.5	2
t _{PHL} Propagation delay time (low to high)	V _{CC} - MAX, V _I = 2.7 V	11	15
t _{PLH} Propagation delay time (high to low)	V _{CC} - MAX, V _I = 0.8 V	10	15
t _{CH} Rise time	V _{CC} - MAX, V _I = 0 V	2	7
t _{FCL} Fall time	V _{CC} - MAX, V _I = 4.5 V	6	13.2

For conditions shown in MIN or MAX, use the appropriate value specified under recommended operating conditions. T_A is typical unless otherwise specified.

Switching characteristics, V_{CC} = 5 V, T_A = 25°C (see note 2)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS		MIN	TYP	MAX	UNIT
			R _L	C _L				
t _{PHL}	A or B	Y	280 Ω	15 pF	2	5	7.5	ns
t _{PLH}	A or B	Y	280 Ω	50 pF	7	7.5	11	ns

NOTE 2: See General Information Section for load circuit and voltage waveforms.

TYPES SN5404, SN54H04, SN54L04, SN54LS04, SN54S04, SN7404, SN74H04, SN74L04, SN74LS04, SN74S04 HEX INVERTERS

REVISIONS: REVISED DECEMBER 1983

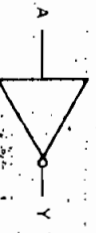
- Package Options include Both Plastic and Ceramic Chip Carriers in Addition to Plastic and Ceramic DIPs
- Dependable Texas Instruments Quality and Reliability

These devices contain six independent inverters.

The SN5404, SN54H04, SN54L04, SN54LS04 and SN54S04 are characterized for operation over the full factory temperature range of -55°C to 125°C. The SN7404, SN74H04, SN74L04 and SN74S04 are characterized for operation from 0°C to 70°C.

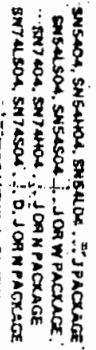
INPUTS	OUTPUT
A	Y
H	L
L	H

Logic diagram (each inverter)

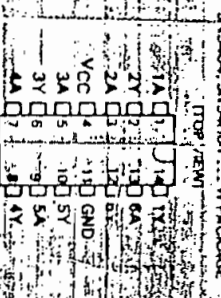


positive logic

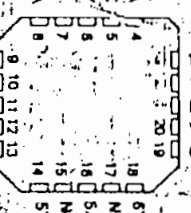
Y = A-bar



(TOP VIEW)



(TOP VIEW)



NC - No internal connection

TYPES SN54S02, SN74S02
QUADUPLE 2-INPUT POSITIVE-NOR GATES

TYPES SN54S03, SN74S03
QUADUPLE 2-INPUT POSITIVE-NAND GATES WITH OPEN-COLLECTOR OUTPUTS

Recommended operating conditions

PARAMETER	TEST CONDITIONS	SN54S02			SN74S02		
		MIN	TYP	MAX	MIN	TYP	MAX
V _{CC} Supply Voltage	V _{CC} = 5 V, I _A = 25°C	4.5	5	5.5	4.75	5	5.25
V _{IH} High-level input voltage	V _{IH} = 0.9 V, I _{IH} = 1 mA	2			2		
V _{IL} Low-level input voltage	V _{IL} = 0.5 V, I _{IL} = 1 mA			0.8			0.8
I _{OH} High-level output current	V _{OH} = 2.7 V, I _{OL} = 20 mA			-1			
I _{OL} Low-level output current	V _{OH} = 2.7 V, I _{OL} = 20 mA			20			20
T _A Operating free-air temperature		-55		125	0		70

Electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS	SN54S02		SN74S02	
		MIN	MAX	MIN	MAX
V _{IK}	V _{CC} = 5 V, I _{IH} = 18 mA	-12		-12	
V _{OH}	V _{CC} = 5 V, I _{IH} = 0.9 V, I _{OH} = 1 mA	2.5	3.4	2.7	3.4
V _{OL}	V _{CC} = 5 V, I _{IH} = 2 V, I _{OL} = 20 mA		0.5		0.5
I _I	V _{CC} = 5 V, V _I = 5.5 V		1		1
I _{II}	V _{CC} = 5 V, V _I = 2.7 V		50		50
I _{IL}	V _{CC} = 5 V, V _I = 0.5 V		-2		-2
I _{OS3}	V _{CC} = 5 V	-40	-100	-40	-100
I _{OS2}	V _{CC} = 5 V, V _I = 0 V	17	29	17	29
I _{OS1}	V _{CC} = 5 V				
I _{OSL}	V _{CC} = 5 V	26	45	26	45

For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
 All values are at V_{CC} = 5 V, T_A = 25°C.
 If not more than one output should be shorted at a time, and the duration of the shorted circuit should not exceed one second.
 NOTE 2: One input at 4.5 V, all others at GND.
 Switching characteristics, V_{CC} = 5 V, T_A = 25°C (see note 3).

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN		TYP		MAX	
t _{PLH}	A or B	Y	R _L = 280 Ω, C _L = 15 pF			3.5		5.5	
t _{PLH}	A or B	Y	R _L = 280 Ω, C _L = 50 pF			5		5	
t _{PHL}	A or B	Y	R _L = 280 Ω, C _L = 50 pF			5		5	

NOTE 3: See General Information Section for load circuits and voltage waveforms.

- Package Options Include Both Plastic and Ceramic Chip Carriers in Addition to Plastic and Ceramic DIPs
- Dependable Texas Instruments Quality and Reliability

Description

These devices contain four independent 2-input NAND gates. The open-collector outputs require pull-up resistors to perform correctly. They may be connected to other open-collector outputs to implement active-low wired-OR or active-high wired-AND functions. Open-collector devices are often used to generate higher V_{OH} levels.
 The SN54S03, SN54S03 and SN74S03 are characterized for operation over the full military temperature range of -55°C to 125°C. The SN74S03, SN74LS03 and SN74S03 are characterized for operation from 0°C to 70°C.

FUNCTION TABLE (each gate)

INPUTS	OUTPUT
A B	Y
H H	L
L H	H
H L	H
L L	H

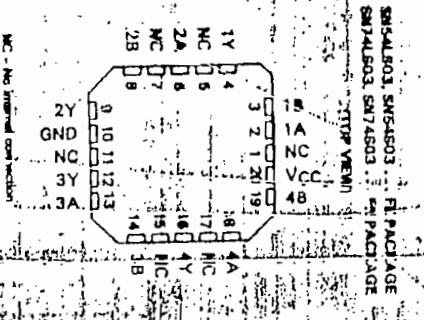
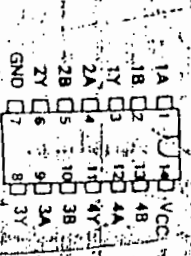
logic diagram (each gate)



positive logic

$Y = \overline{A \cdot B}$ or $Y = \overline{A} + \overline{B}$

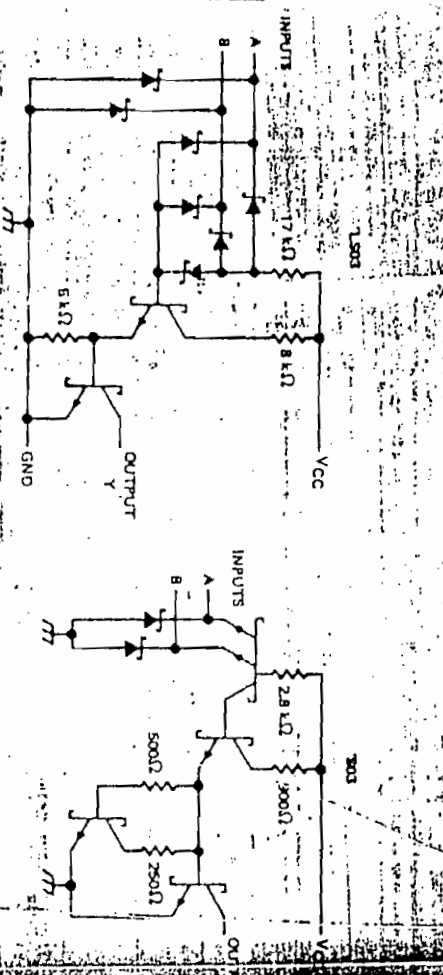
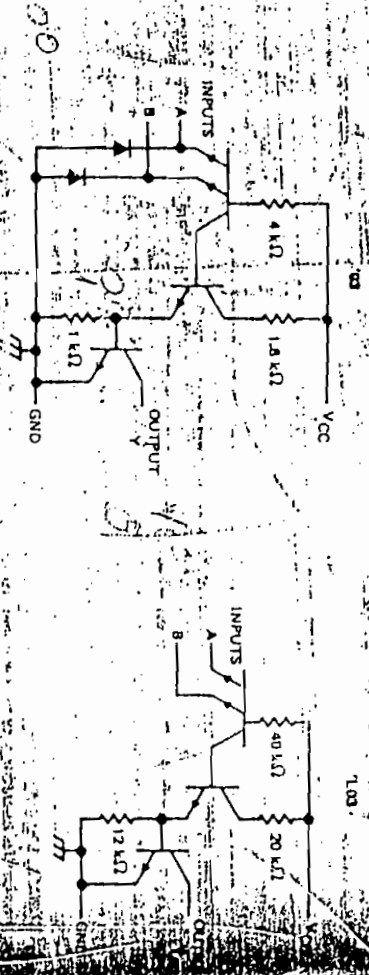
SN54S03, SN74S03 ... J PACKAGE
 SN54S03, SN74S03 ... J OM PACKAGE
 SN74LS03, SN74S03 ... D, OM N PACKAGE



NC - No internal connection

QUADRUPLE 2-INPUT POSITIVE-NAND GATES WITH OPEN-COLLECTOR OUTPUTS

schematics (each gate)



Resistor values shown are nominal.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

- Supply voltage, VCC (see Note 1): L03, L503, S03
- Input voltage: L03, L03, S03
- Off-state output voltage: L03
- Operating free-air temperature range: L03, S03
- Storage temperature range: SN74A

QUADRUPLE 2-INPUT POSITIVE-NAND GATES WITH OPEN-COLLECTOR OUTPUTS

recommended operating conditions

Parameter	MIN		MAX		UNIT
	TYP	MIN	TYP	MAX	
VCC Supply voltage		4.5	5.5	4.75	V
High-level input voltage		0.8	1.5	1.5	V
Low-level input voltage		0.1	0.1	0.1	V
High-level output voltage		0.1	0.1	0.1	V
Low-level output voltage		0.1	0.1	0.1	V
Output current		0	0	0	mA

Electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

Parameter	TEST CONDITIONS	MIN	TYP	MAX	UNIT
V _{OL}	V _{CC} = MIN, I _L = 12 mA	0.1	0.1	0.1	V
V _{OH}	V _{CC} = MIN, V _I = 0.8 V, V _O H = 5.5 V	0.1	0.1	0.1	V
I _{OL}	V _{CC} = MIN, V _I = 2 V, I _{OL} = 16 mA	0.2	0.2	0.4	mA
I _{OH}	V _{CC} = MAX, V _I = 5.5 V	0.1	0.1	0.1	mA
t _{PLH}	V _{CC} = MAX, V _I = 2.4 V	15	15	15	ns
t _{PLL}	V _{CC} = MAX, V _I = 0.4 V	4	4	8	ns
t _{COH}	V _{CC} = MAX, V _I = 0 V	12	12	12	ns
t _{COL}	V _{CC} = MAX, V _I = 4.5 V				

NOTE 2: See General Information Section for load circuit and voltage waveforms.

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t _{PLH}	A or B	Y	R _L = 400 Ω, C _L = 15 pF	1.8	4.5	15	ns
t _{PLL}	A or B	Y	R _L = 400 Ω, C _L = 15 pF	1.8	7.5	15	ns



+5V-Powered Multi-Channel RS-232 Drivers/Receivers

MAX220-MAX249

ABSOLUTE MAXIMUM RATINGS – MAX220/222/232A/233A/242/243

Supply Voltage (V _{CC})	-0.3V to +6V	16-Pin Narrow SO (derate 8.70mW/°C above +70°C)	696mW
Input Voltages		16-Pin Wide SO (derate 9.52mW/°C above +70°C)	762mW
V _{IN}	-0.3V to (V _{CC} - 0.3V)	18-Pin Wide SO (derate 9.52mW/°C above +70°C)	762mW
R _{IN}	±30V	20-Pin Wide SO (derate 10.00mW/°C above +70°C)	?
T _{OUT} (Note 1)	±15V	16-Pin CERDIP (derate 10.00mW/°C above +70°C)	800mW
Output Voltages		18-Pin CERDIP (derate 10.53mW/°C above +70°C)	842mW
T _{OUT}	±15V	Operating Temperature Ranges:	
R _{OUT}	-0.3V to (V _{CC} + 0.3V)	MAX2__AC__, MAX2__C__	0°C to +70°C
Driver/Receiver Output Short Circuited to GND	Continuous	MAX2__AE__, MAX2__E__	-40°C to +85°C
Continuous Power Dissipation (T _A = +70°C)		MAX2__AM__, MAX2__M__	-55°C to +125°C
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C)	842mW	Storage Temperature Range	-65°C to +160°C
18-Pin Plastic DIP (derate 11.11mW/°C above +70°C)	889mW	Lead Temperature (soldering, 10 sec)	+300°C
20-Pin Plastic DIP (derate 8.00mW/°C above +70°C)	440mW		

Note 1: Input voltage measured with T_{OUT} in high-impedance state, \overline{SHDN} or V_{CC} = 0V.

Stresses beyond those listed under 'Absolute Maximum Ratings' may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS – MAX220/222/232A/233A/242/243

(V_{CC} = +5V ±10%, C₁-C₄ = 0.1μF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 TRANSMITTERS						
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to GND		±5	±8		V
Input Logic Threshold Low				1.4	0.8	V
Input Logic Threshold High			2	1.4		V
Logic Pull-Up/Input Current	SHDN = V _{CC}			5	40	μA
	SHDN = 0V			±0.01	±1	
Output Leakage Current	V _{CC} = 5.5V, SHDN = 0V, V _{OUT} = ±15V			±0.01	±10	μA
	V _{CC} = SHDN = 0V, V _{OUT} = ±15V			±0.01	±10	
Data Rate	Except MAX220, normal operation			200	116	kbits/sec
	MAX220			22	20	
Transmitter Output Resistance	V _{CC} = V ₊ = V ₋ = 0V, V _{OUT} = ±2V		300	10M		Ω
Output Short-Circuit Current	V _{OUT} = 0V		±7	±22		mA
RS-232 RECEIVERS						
RS-232 Input Voltage Operating Range					±30	V
RS-232 Input Threshold Low	V _{CC} = 5V	Except MAX243 R _{2IN}	0.8	1.3		V
		MAX243 R _{2IN} (Note 2)	-3			
RS-232 Input Threshold High	V _{CC} = 5V	Except MAX243 R _{2IN}		1.8	2.4	V
		MAX243 R _{2IN} (Note 2)		-0.5	-0.1	
RS-232 Input Hysteresis	Except MAX243, V _{CC} = 5V, no hyst. in shdn.		0.2	0.5	1	V
	MAX243			1		
RS-232 Input Resistance			3	5	7	kΩ
TTL/CMOS Output Voltage Low	I _{OUT} = 3.2mA			0.2	0.4	V
TTL/CMOS Output Voltage High	I _{OUT} = -1.0mA		3.5	V _{CC} - 0.2		V
TTL/CMOS Output Short-Circuit Current	Sourcing V _{OUT} = GND		-2	-10		mA
	Sinking V _{OUT} = V _{CC}		10	30		
TTL/CMOS Output Leakage Current	SHDN = V _{CC} or EN = V _{CC} , 0V ≤ V _{OUT} ≤ V _{CC}			±0.05	±10	μA

+5V-Powered Multi-Channel RS-232 Drivers/Receivers

MAX220-MAX249

ELECTRICAL CHARACTERISTICS – MAX220/222/232A/233A/242/243 (continued)

(VCC = +5V ±10%, C1-C4 = 0.1µF, TA = TMIN to TMAX, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS	
EN Input Threshold Low				1.4	0.8	V	
EN Input Threshold High			2.0	1.4		V	
POWER SUPPLY							
Operating Supply Voltage			4.5		5.5	V	
VCC Supply Current (SHDN = VCC), Figures 5, 6, 9, 18	No load	MAX220		0.5	2	mA	
		MAX222/232A/233A/242/243		4	10		
	3kΩ load both outputs	MAX220		12			
		MAX222/232A/233A/242/243		15			
Shutdown Supply Current	MAX222/242	TA = +25°C		0.1	10	µA	
		TA = 0°C to +70°C		2	50		
		TA = -40°C to +85°C		2	50		
		TA = -55°C to +125°C		35	100		
SHDN Input Leakage Current					±1	µA	
SHDN Threshold Low				1.4	0.8	V	
SHDN Threshold High			2.0	1.4		V	
AC CHARACTERISTICS							
Transition Slew Rate		CL = 50pF to 2500pF, RL = 3kΩ to 7kΩ, VCC = 5V, TA = +25°C, measured from +3V to -3V or -3V to +3V	MAX222/232A/233A/242/243	6	12	30	V/µs
		MAX220	1.5	3	30		
Transmitter Propagation Delay TTL to RS-232 (Normal Operation), Figure 1	tPHLT		MAX222/232A/233A/242/243		1.3	3.5	µs
			MAX220		4	10	
	tPLHT		MAX222/232A/233A/242/243		1.5	3.5	
			MAX220		5	10	
Receiver Propagation Delay RS-232 to TTL (Normal Operation), Figure 2	tPHLR		MAX222/232A/233A/242/243		0.5	1	µs
			MAX220		0.6	3	
	tPLHR		MAX222/232A/233A/242/243		0.6	1	
			MAX220		0.8	3	
Receiver Propagation Delay RS-232 to TTL (Shutdown), Figure 2	tPHLS		MAX242		0.5	10	µs
	tPLHS		MAX242		2.5	10	
Receiver-Output Enable Time, Figure 3	tER		MAX222/242		125	500	ns
Receiver-Output Disable Time, Figure 3	tDR		MAX222/242		160	500	ns
Transmitter-Output Enable Time (SHDN goes high), Figure 4	tET		MAX222/242, 0.1µF caps (Includes charge-pump start-up)		250		µs
Transmitter-Output Disable Time (SHDN goes low), Figure 4	tDT		MAX222/242, 0.1µF caps		600		ns
Transmitter + to - Propagation Delay Difference (Normal Operation)	tPHLT-tPLHT		MAX222/232A/233A/242/243		300		ns
			MAX220		2000		
Receiver + to - Propagation Delay Difference (Normal Operation)	tPHLR-tPLHR		MAX222/232A/233A/242/243		100		ns
			MAX220		225		

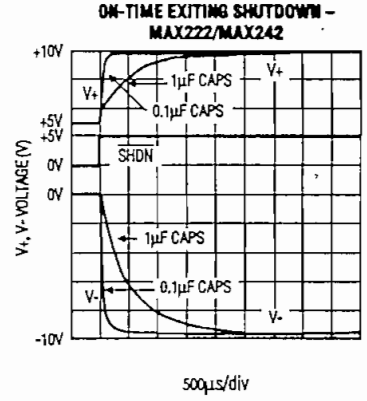
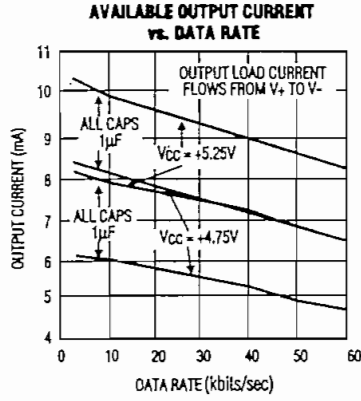
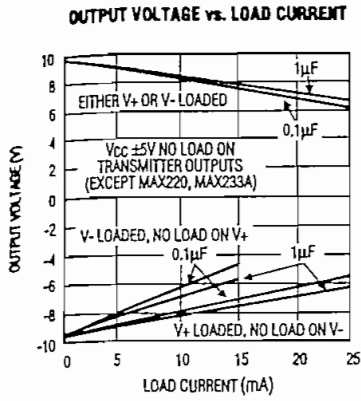
Note 2: MAX243 R2OUT is guaranteed to be low when the R2IN is ≥ 0V or is floating.

OUTPUT VOLTAGE (V)

+5V-Powered Multi-Channel RS-232 Drivers/Receivers

Typical Operating Characteristics

MAX220/222/232A/233A/242/243



MAX220-MAX249

2

+5V-Powered Multi-Channel RS-232 Drivers/Receivers

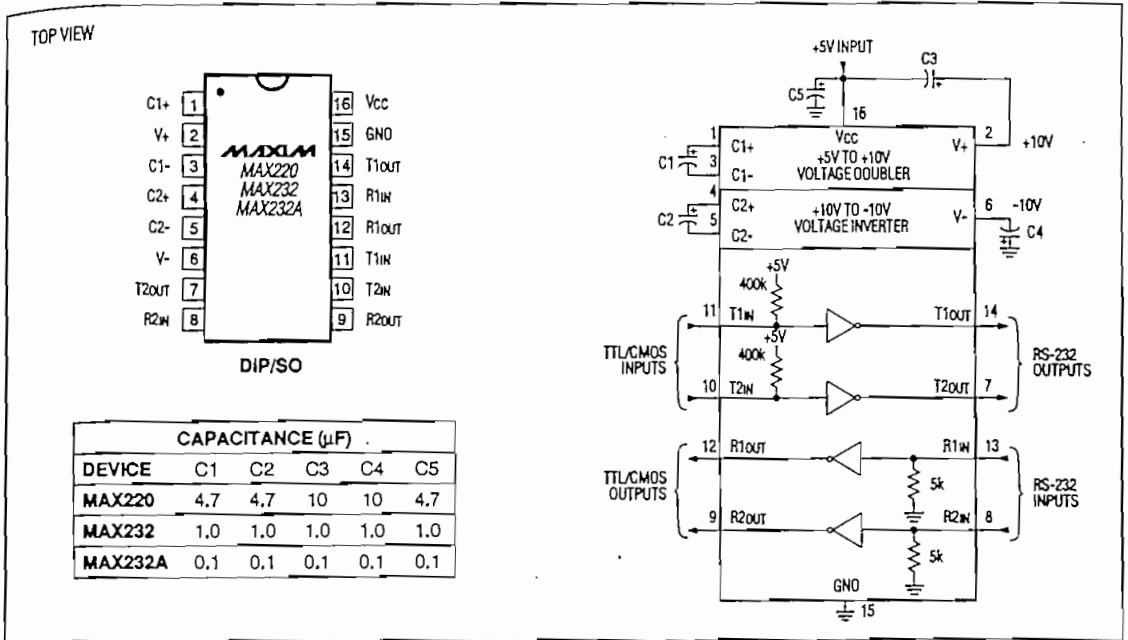


Figure 5. MAX220/232/232A Pin Configuration and Typical Operating Circuit

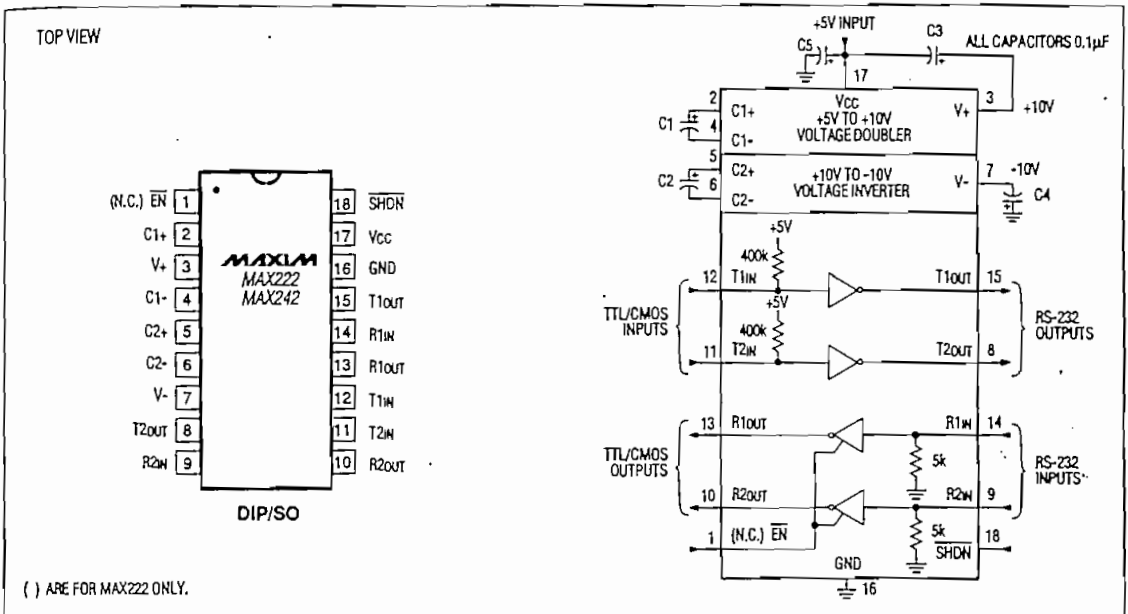


Figure 6. MAX222/MAX242 Pin Configuration and Typical Operating Circuit

CD4066BM/CD4066BC Quad Bilateral Switch

General Description

The CD4066BM/CD4066BC is a quad bilateral switch intended for the transmission or multiplexing of analog or digital signals. It is pin-for-pin compatible with CD4016BM/CD4016BC, but has a much lower "ON" resistance, and "ON" resistance is relatively constant over the input-signal range.

- Extremely low "OFF" switch leakage: 0.1 nA (typ.) @ $V_{DD} - V_{SS} = 10V, T_A = 25^\circ C$
- Extremely high control input impedance: $10^{12} \Omega$ (typ.)
- Low crosstalk between switches: -50 dB (typ.) @ $f_{is} = 0.9 MHz, R_L = 1 k\Omega$
- Frequency response, switch "ON": 40 MHz (typ.)

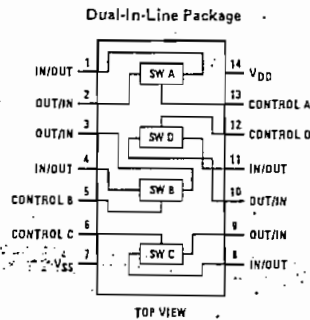
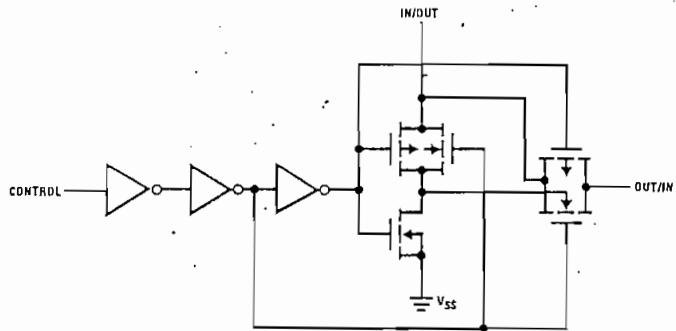
Features

- Wide supply voltage range: 3V to 15V
- High noise immunity: $0.45 V_{DD}$ (typ.)
- Wide range of digital and analog switching: $\pm 7.5 V_{PEAK}$
- "ON" resistance for 15V operation: 80 Ω
- Matched "ON" resistance over 15V signal input: $\Delta R_{ON} = 5 \Omega$ (typ.)
- "ON" resistance flat over peak-to-peak signal range
- High "ON"/"OFF" output voltage ratio: 65 dB (typ.) @ $f_{is} = 10 kHz, R_L = 10 k\Omega$
- High degree linearity: 0.1% distortion (typ.) @ $f_{is} = 1 kHz, V_{is} = 5 V_{p-p}, V_{DD} - V_{SS} = 10V, R_L = 10 k\Omega$

Applications

- Analog signal switching/multiplexing
 - Signal gating
 - Squelch control
 - Chopper
 - Modulator/Demodulator
 - Commutating switch
- Digital signal switching/multiplexing
- CMOS logic implementation
- Analog-to-digital/digital-to-analog conversion
- Digital control of frequency, impedance, phase, and analog-signal-gain

Schematic and Connection Diagrams



Absolute Maximum Ratings

Notes 1 and 2)

V _{DD} Supply Voltage	-0.5V to +18V
V _{IH} Input Voltage	-0.5V to V _{DD} + 0.5V
T _S Storage Temperature Range	-65°C to +150°C
P _D Package Dissipation	500 mW
T _L Lead Temperature (Soldering, 10 seconds)	300°C

Recommended Operating Conditions

(Note 2)

V _{DD} Supply Voltage	3V to 15V
V _{IH} Input Voltage	0V to V _{DD}
T _A Operating Temperature Range	-55°C to +125°C
CD4066BM	-40°C to +85°C
CD4066BC	

DC Electrical Characteristics CD4066BM (Note 2)

Parameter	Conditions	-55°C		25°C			125°C		Units
		Min	Max	Min	Typ	Max	Min	Max	
I _{DD} Quiescent Device Current	V _{DD} = 5V		0.25		0.01	0.25		7.5	μA
	V _{DD} = 10V		0.5		0.01	0.5		15	μA
	V _{DD} = 15V		1.0		0.01	1.0		30	μA
Signal Inputs and Outputs									
R _{ON} "ON" Resistance	R _L = 10 kΩ to $\frac{V_{DD}-V_{SS}}{2}$ V _C = V _{DD} , V _{IS} = V _{SS} to V _{DD} V _{DD} = 5V V _{DD} = 10V V _{DD} = 15V		2000 400 220		270 120 80	2500 500 280		3500 550 320	Ω
ΔR _{ON} Δ "ON" Resistance Between any 2 of 4 Switches	R _L = 10 kΩ to $\frac{V_{DD}-V_{SS}}{2}$ V _C = V _{DD} , V _{IS} = V _{SS} to V _{DD} V _{DD} = 10V V _{DD} = 15V				10 5				Ω
I _{OL} Input or Output Leakage Switch "OFF"	V _C = 0 V _{IS} = 15V and 0V, V _{OS} = 0V and 15V		±50		±0.1	±50		±500	nA
Control Inputs									
V _{IL} Low Level Input Voltage	V _{IS} = V _{SS} and V _{DD} V _{OS} = V _{DD} and V _{SS} I _{IS} = ±10 μA V _{DD} = 5V V _{DD} = 10V V _{DD} = 15V		1.5 3.0 4.0		2.25 4.5 6.75	1.5 3.0 4.0		1.5 3.0 4.0	V
V _{IH} High Level Input Voltage	V _{DD} = 5V V _{DD} = 10V (see note 6) V _{DD} = 15V	3.5 7.0 11.0		3.5 7.0 11.0	2.75 5.5 8.25		3.5 7.0 11.0		V
I _{IN} Input Current	V _{DD} - V _{SS} = 15V V _{DD} > V _{IS} > V _{SS} V _{DD} > V _C > V _{SS}		±0.1		±10 ⁻⁵	±0.1		±1.0	μA

DC Electrical Characteristics CD4066BC (Note 2)

Parameter	Conditions	-40°C		25°C			85°C		Units
		Min	Max	Min	Typ	Max	Min	Max	
I _{DD} Quiescent Device Current	V _{DD} = 5V		1.0		0.01	1.0		7.5	μA
	V _{DD} = 10V		2.0		0.01	2.0		15	μA
	V _{DD} = 15V		4.0		0.01	4.0		30	μA

AC Electrical Characteristics (Continued)

$T_A = 25^\circ\text{C}$, $t_r = t_f = 20\text{ ns}$ and $V_{SS} = 0\text{ V}$ unless otherwise noted

Parameter	Conditions	Min.	Typ.	Max.	Units
Feedthrough — Switch "OFF" (Frequency at -50 dB)	$V_{DD} = 5.0\text{ V}$, $V_{CC} = V_{SS} = -5.0\text{ V}$, $R_L = 1\text{ k}\Omega$, $V_{IS} = 5.0\text{ V}_{p-p}$, 20 Log_{10} , $V_{OS}/V_{IS} = -50\text{ dB}$, (Figure 4)		1.25		
Crosstalk Between Any Two Switches (Frequency at -50 dB)	$V_{DD} = V_{C(A)} = 5.0\text{ V}$; $V_{SS} = V_{C(B)} = -5.0\text{ V}$, $R_L = 1\text{ k}\Omega$, $V_{IS(A)} = 5.0\text{ V}_{p-p}$, 20 Log_{10} , $V_{OS(B)}/V_{IS(A)} = -50\text{ dB}$, (Figure 5)		0.9		MHz
Crosstalk; Control Input to Signal Output	$V_{DD} = 10\text{ V}$, $R_L = 10\text{ k}\Omega$, $R_{IN} = 1.0\text{ k}\Omega$, $V_{CC} = 10\text{ V}$ Square Wave, $C_L = 50\text{ pF}$ (Figure 6)		150		mV _{p-p}
Maximum Control Input	$R_L = 1.0\text{ k}\Omega$, $C_L = 50\text{ pF}$, (Figure 7) $V_{OS(f)} = \frac{1}{2}V_{OS}(1.0\text{ kHz})$ $V_{DD} = 5.0\text{ V}$ $V_{DD} = 10\text{ V}$ $V_{DD} = 15\text{ V}$		6.0 8.0 8.5		MHz MHz MHz
C_{IS} Signal Input Capacitance			8.0		pF
C_{OS} Signal Output Capacitance	$V_{DD} = 10\text{ V}$		8.0		pF
C_{IOS} Feedthrough Capacitance	$V_C = 0\text{ V}$		0.5		pF
C_{IN} Control Input Capacitance			5.0	7.5	pF

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the devices should be operated at these limits. The tables of "Recommended Operating Conditions" and "Electrical Characteristics" provide conditions for actual device operation.

Note 2: $V_{SS} = 0\text{ V}$ unless otherwise specified.

Note 3: These devices should not be connected to circuits with the power "ON".

Note 4: In all cases, there is approximately 5 pF of probe and jig capacitance in the output; however, this capacitance is included in C_L wherever it is specified.

Note 5: V_{IS} is the voltage at the in/out pin and V_{OS} is the voltage at the out/in pin. V_C is the voltage at the control input.

Note 6: Conditions for V_{IHC} :

a) $V_{IS} = V_{DD}$, $I_{OS} = \text{standard B series } I_{OH}$ b) $V_{IS} = 0\text{ V}$, $I_{OL} = \text{standard B series } I_{OL}$

AC Test Circuits and Switching Time Waveforms

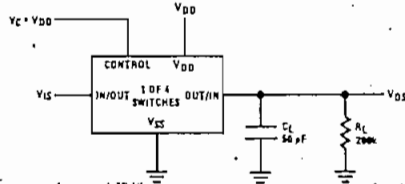


FIGURE 1. t_{pHL} , t_{pLH} Propagation Delay Time Signal Input to Signal Output

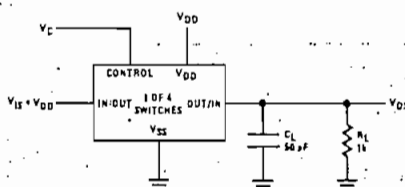
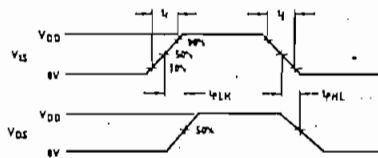


FIGURE 2. t_{pZH} , t_{pHZ} Propagation Delay Time Control to Signal Output

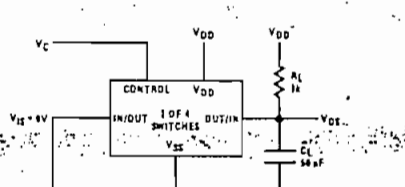
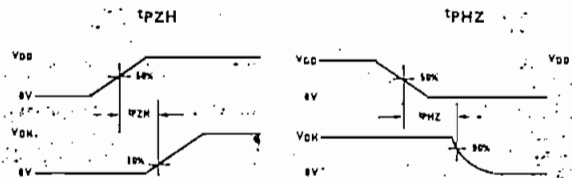
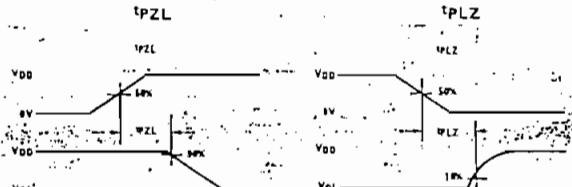


FIGURE 3. t_{pZL} , t_{pLZ} Propagation Delay Time Control to Signal Output



AC Test Circuits and Switching Time Waveforms (Cont'd.)

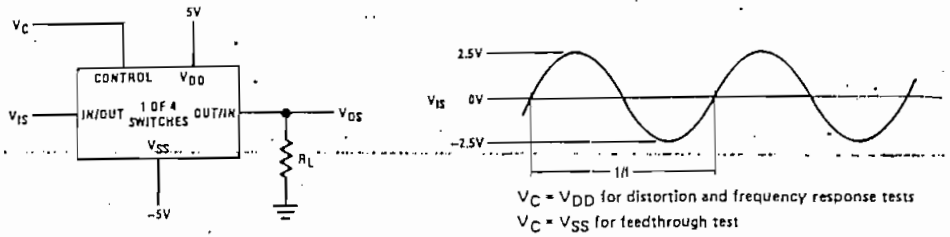


FIGURE 4. Sine Wave Distortion, Frequency Response and Feedthrough

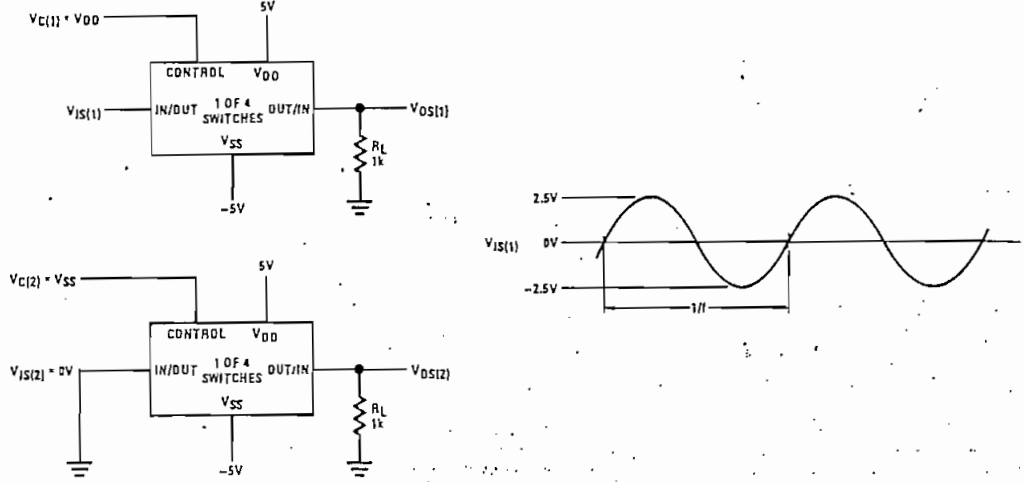


FIGURE 5. Crosstalk Between Any Two Switches

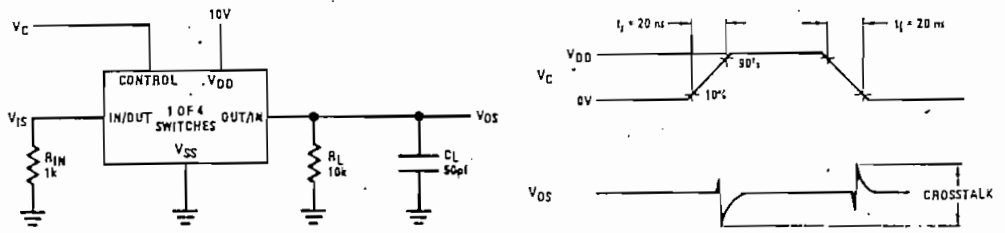


FIGURE 6. Crosstalk: Control Input to Signal Output

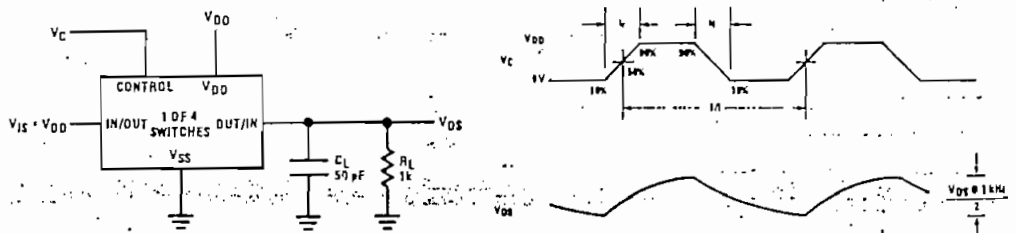
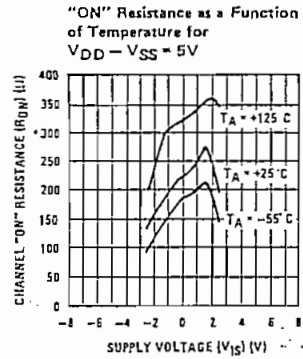
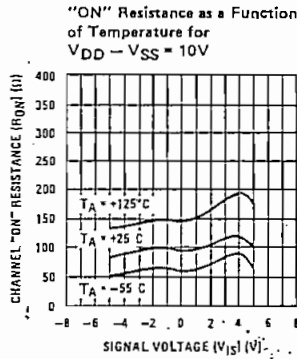
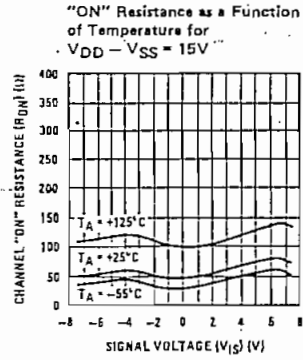
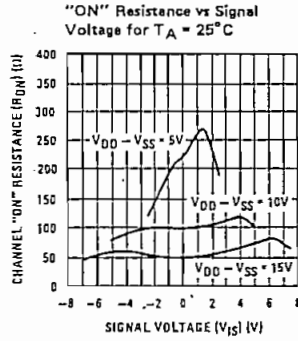


FIGURE 7. Maximum Control Input Frequency

Typical Performance Characteristics



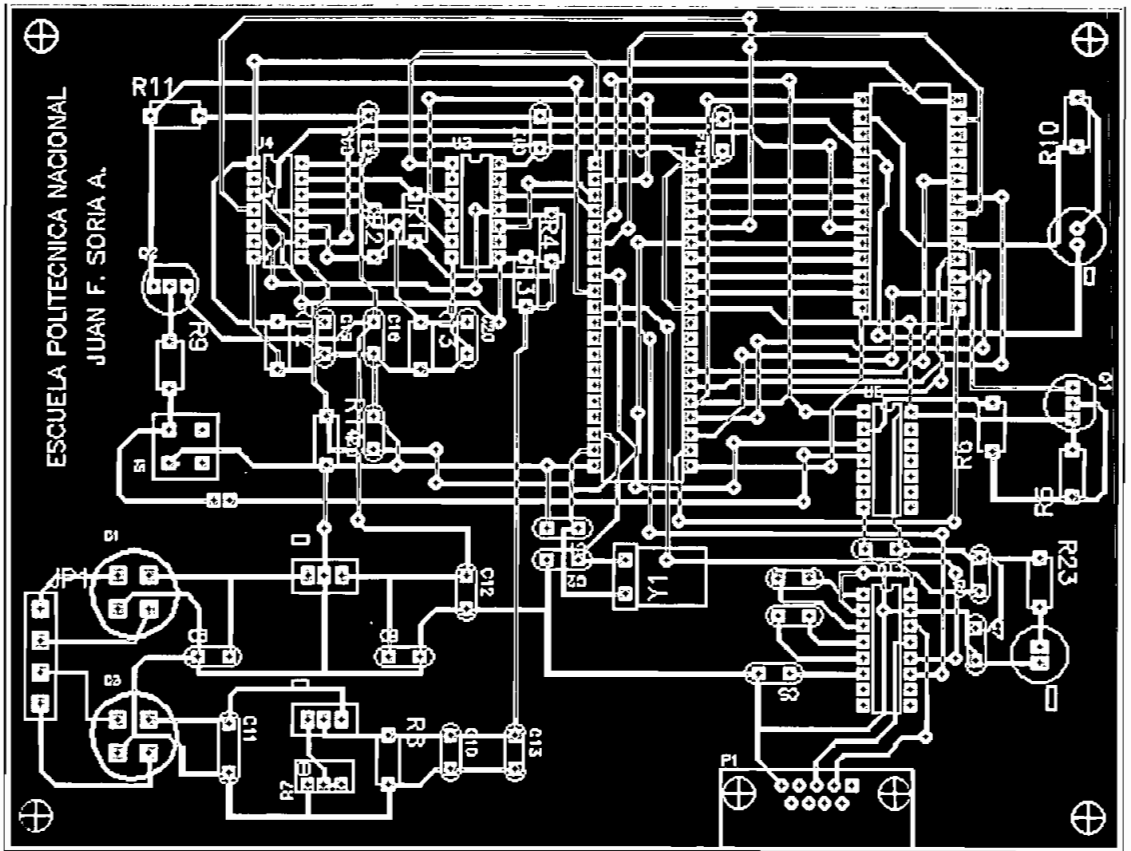
Special Considerations

In applications where separate power sources are used to drive V_{DD} and the signal input, the V_{DD} current capability should exceed V_{DD}/R_L (R_L = effective external load of the 4 CD4066BM/CD4066BC bilateral switches). This provision avoids any permanent current flow or clamp action on the V_{DD} supply when power is applied or removed from CD4066BM/CD4066BC.

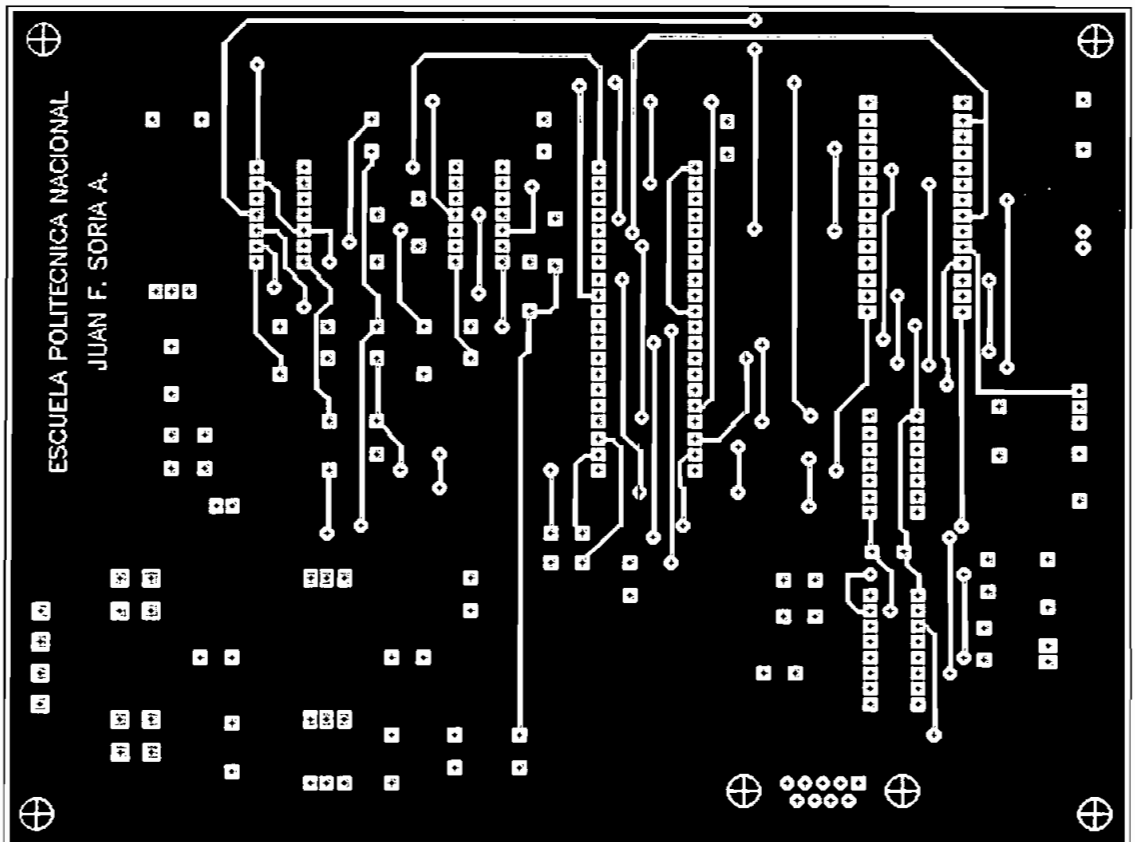
In certain applications, the external load-resistor current may include both V_{DD} and signal-line components. To

avoid drawing V_{DD} current when switch current flows into terminals 1, 4, 8 or 11, the voltage drop across the bidirectional switch must not exceed 0.6V at $T_A \leq 25^\circ\text{C}$, or 0.4V at $T_A > 25^\circ\text{C}$ (calculated from R_{ON} values shown).

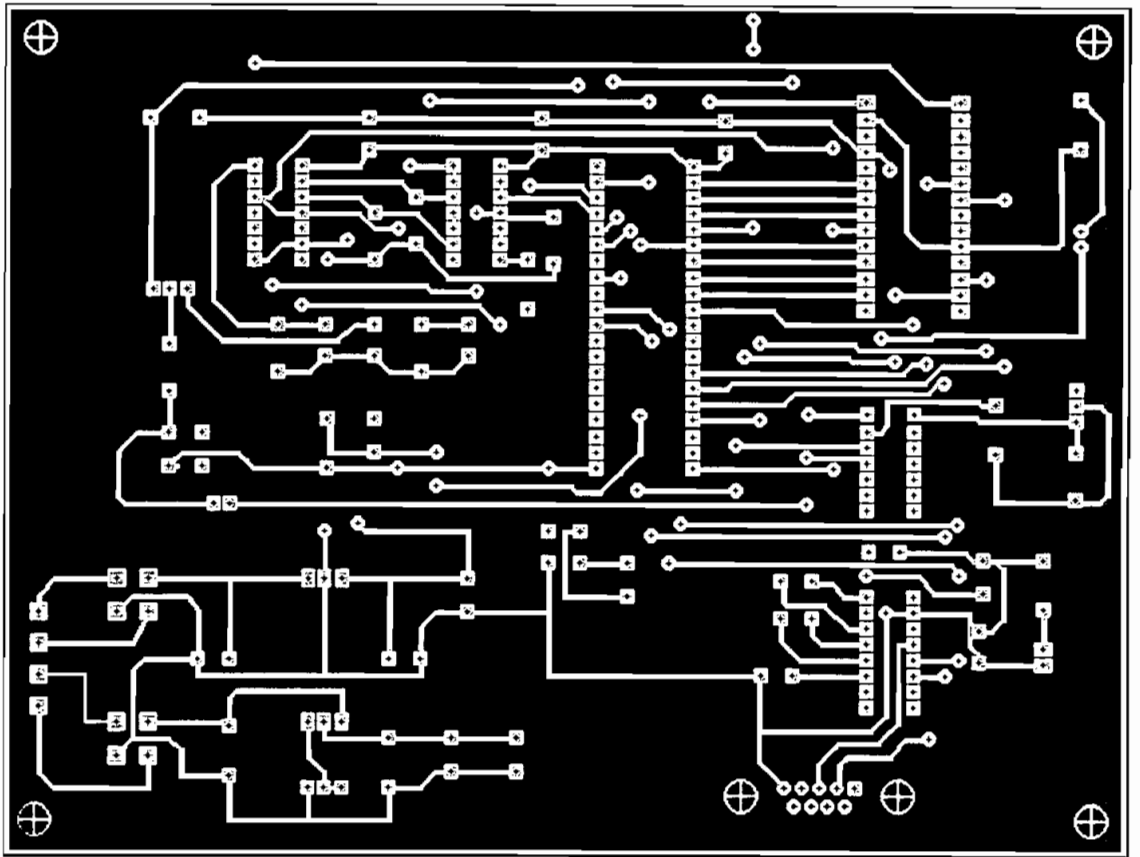
No V_{DD} current will flow through R_L if the switch current flows into terminals 2, 3, 9 or 10.



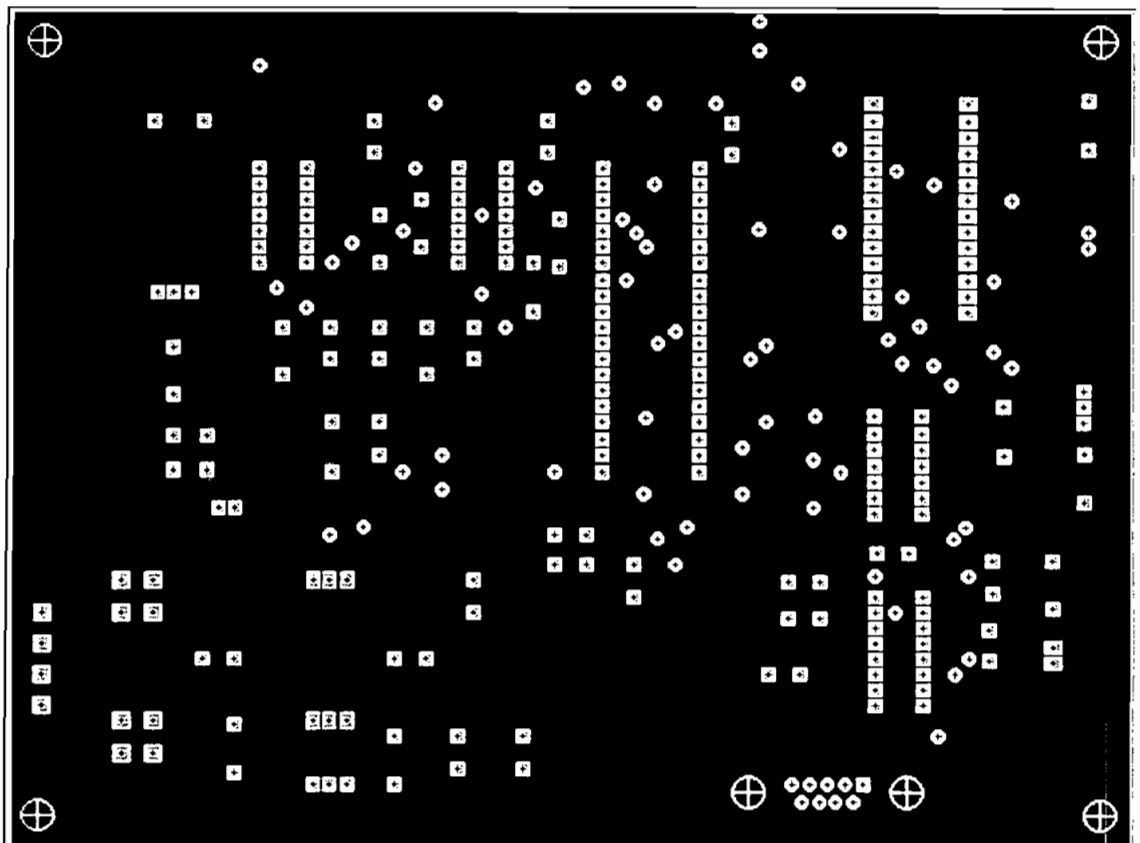
TARJETA PRINCIPAL DEL PROGRAMADOR PROGPIC



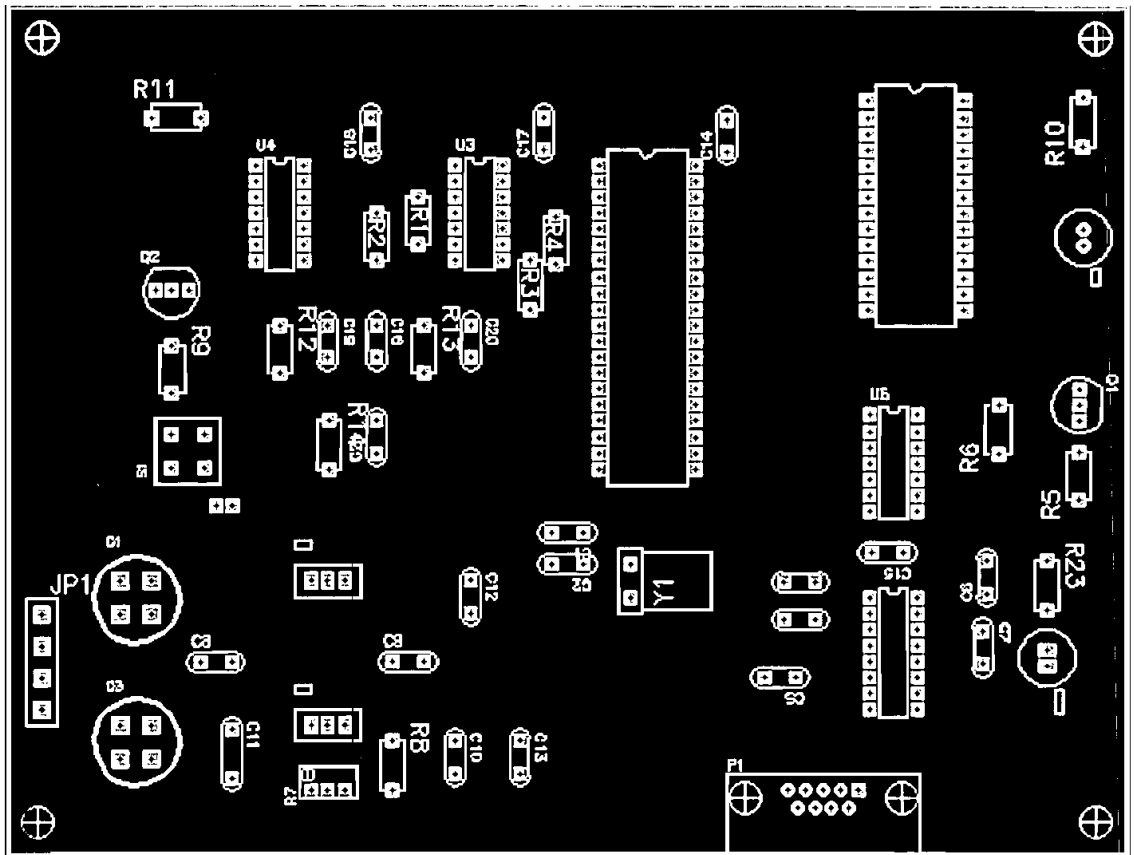
TARJETA POR EL LADO DE LOS COMPONENTES



TARJETA POR LADO DE LADO DE LAS SUELDAS



VIAS Y PADS DE LA TARJETA



DISTRIBUCION DE LOS ELEMENTOS EN LA TARJETA