

ESCUELA POLITECNICA NACIONAL
FACULTAD DE INGENIERIA ELECTRICA

SISTEMA DIDACTICO DE DESARROLLO BASADO
EN EL MICROCONTROLADOR M68HC11

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

"ANGEL VINICIO ACOSTA VILLACIS"

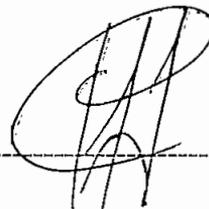
INGENIERO EN ELECTRONICA Y
TELECOMUNICACIONES

"WILSON WILFRIDO PASTUISACA ORELLANA"

Quito, Junio, 1998

CERTIFICACION

CERTIFICO QUE EL PRESENTE TRABAJO DE TESIS HA SIDO DESARROLLADO EN SU TOTALIDAD POR LOS SEÑORES ANGEL VINICIO ACOSTA VILLACIS Y WILSON WILFRIDO PASTUISACA ORELLANA, BAJO MI DIRECCION.

A handwritten signature in black ink, appearing to be 'B. Ledesma', written over a horizontal dashed line.

ING. BOLIVAR LEDESMA

DIRECTOR DE TESIS

AGRADECIMIENTO

A la Escuela Politécnica Nacional, a la Facultad de Ingeniería Eléctrica, a nuestro Director de Tesis: Ing. Bolívar Ledesma y a todos los profesores que colaboraron para culminar con éxito nuestra formación a nivel superior, y el desarrollo del presente trabajo de tesis.

DEDICATORIA

A Dios por su infinita bondad, ya que sin su ayuda me sería imposible alcanzar las metas que me he planteado.

A mis padres y hermanos por su constante apoyo y motivación en todos los momentos de mi vida.

Angel Vinicio Acosta Villacís

A Dios, mi familia y a todos quienes me incentivaron y motivaron para culminar mis estudios universitarios con éxito. De corazón:

Wilson Wilfrido Pastuisaca Orellana

INTRODUCCION

En éste trabajo se describe en detalle todas las características del sistema didáctico de desarrollo V&W – HC11, de su principal componente: el microcontrolador MC68HC11, de la tarjeta principal "M68HC11EVB" y de la tarjeta de periféricos V&W – IN/OUT. El módulo V&W – HC11 es un verdadero sistema didáctico de desarrollo, de fácil manejo y diseñado para trabajar en el laboratorio de Control con Microprocesadores y para implementar cualquier sistema físico real que involucre adquisición de datos, control de procesos, procesamiento digital de señales, comunicaciones, etc.

El sistema se complementa con un programa Monitor, el cual sirve para depurar, evaluar y ejecutar los programas implementados por el usuario. Además se han desarrollado nueve prácticas demostrativas cuyos algoritmos tanto en lenguaje ensamblador para el microcontrolador, como en Quick Basic para el computador se han resuelto tomando en cuenta la optimización del espacio de memoria ocupado por el programa y del tiempo de ejecución del mismo.

En cada una de las prácticas se ha ido comprobando el correcto funcionamiento de todas y cada una de las partes del módulo, tratando de potenciar al máximo sus características como son: entradas y salidas analógicas y digitales, sistemas de comunicaciones, pórtricos de entrada – salida de datos, periféricos manejados como localidades de memoria externa, interrupciones, temporizadores y contadores.

INDICE

ITEM

Pg. No.

CAPITULO I EL MICROCONTROLADOR MC68HC11

1.1	INTRODUCCIÓN.....	2
1.2	LA FAMILIA DE MICROCONTROLADORES MC68HC11.....	3
1.2.1	Características.....	3
1.2.2	Descripción General.....	4
1.2.3	Módulo del Programador.....	5
1.2.4	Familia de Microcontroladores MC68HC11.....	6
1.3	MODOS DE OPERACIÓN Y DESCRIPCIÓN DE PINES.....	7
1.3.1	Descripciones de las señales de los pines.....	7
1.3.2	Modos de Operación.....	15
1.4	MEMORIAS INTERNAS DEL MICROCONTROLADOR.....	19
1.4.1	Mapas de Memoria.....	20
1.4.2	Registro INIT.....	21
1.4.3	Memoria ROM.....	22
1.4.4	Memoria RAM.....	23
1.4.5	Memoria EEPROM.....	23
1.5	EL CPU Y SUS MODOS DE DIRECCIONAMIENTO.....	27
1.5.1	Registros del CPU.....	27
1.5.2	Modos de Direccionamiento.....	30
1.6	PÓRTICOS DE ENTRADA SALIDA DE DATOS.....	32
1.6.1	Pórticos C y D (entradas y salidas de propósito general).....	32
1.6.2	Pórticos A, B y E (entradas o salidas como dirección específica).....	33
1.6.3	Habilitación simple de entradas y salidas.....	33
1.6.4	Entradas y salidas paralelas en el modo de HANDSHAKE.....	35
1.6.5	Registro de control I/O paralelas (PIOC).....	37
1.7	INTERFAZ DE COMUNICACIÓN SERIAL SINCRÓNICA.....	39
1.7.1	Características.....	39
1.7.2	Descripciones de señales SPI.....	40
1.7.3	Descripción Funcional.....	41
1.7.4	Registro del SPI.....	44
1.8	INTERFAZ DE COMUNICACIÓN SERIAL ASINCRÓNICA.....	48
1.8.1	Características.....	48
1.8.2	Formato de Datos.....	49
1.8.3	Características del inicio de Recepción.....	50
1.8.4	Datos Recibidos (RxD).....	51
1.8.5	Datos Transmitidos TxD.....	52
1.8.6	Descripción funcional del SCI.....	52
1.8.7	Registros del SCI.....	53
1.9	CONVERSION ANÁLOGO DIGITAL.....	62
1.9.1	Proceso de conversión.....	62
1.9.2	Asignación de canales.....	62

1.9.3	Modos de Operación.....	63
1.9.4	Registros de Control de estado del A/D.....	64
1.9.5	Registros de resultados (ADR1, ADR2, ADR3 y ADR4).....	66
1.9.6	Selección de Reloj y energía del convertor.....	66
1.10	TEMPORIZADORES Y CONTADORES.....	67
1.10.1	Temporizador Programable.....	67
1.10.2	Interrupción en tiempo Real.....	77
1.10.3	Acumulador de Pulsos.....	77
1.11	INTERRUPCIONES.....	80
1.11.1	Interrupciones de Software.....	81
1.11.2	Interrupciones de código Ilegal.....	82
1.11.3	Bits de máscara de Interrupción en el registro de código (CCR).....	82
1.11.4	Estructura de prioridades.....	83
1.11.5	Registro de prioridad de Interrupción (HPRIO).....	84
1.12	SISTEMAS DE RESET Y MODOS DE BAJO CONSUMO.....	86
1.12.1	Sistemas de Reset.....	86
1.12.2	Modos de bajo consumo.....	94
1.13	SET DE INSTRUCCIONES EN DETALLE	

CAPITULO 2

TARJETA PRINCIPAL M68HC11EVB

2.1	INTRODUCCIÓN.....	98
2.2	CARACTERÍSTICAS Y ESPECIFICACIONES.....	99
2.2.1	Características.....	99
2.2.2	Especificaciones.....	102
2.3	PREPARACIÓN DE HARDWARE E INSTRUCCIONES DE INSTALACION.....	103
2.3.1	Preparación de Hardware.....	103
2.3.2	Instrucciones de Instalación.....	108
2.4	DESCRIPCIÓN DE HARDWARE.....	112
2.4.1	Descripción general.....	112
2.4.2	El Microcontrolador.....	113
2.4.3	Amplificador de pórticos PRU.....	113
2.4.4	Memoria.....	114
2.4.5	Decodificación y Demultiplexación de Direcciones.....	114
2.4.6	Circuitos de Interfaz seria RS-232.....	115
2.5	INSTRUCCIONES DE OPERACIÓN.....	116
2.5.1	Limitaciones.....	116
2.5.2	Procedimientos de Operación.....	117
2.5.3	Formato de línea de Comandos.....	118
2.5.4	Comandos del Programa Monitor.....	119
2.6	EL PROGRAMA MONITOR BUFFALO.....	122
2.6.1	Descripción del programa.....	122
2.6.2	Vectores de Interrupción.....	127

CAPITULO 3 TARJETA DE ENTRADAS Y SALIDAS

3.1 MAPA DE MEMORIA DE PERIFÉRICOS.....	130
3.2 CIRCUITO DE CONVERSIÓN DIGITAL ANALOGA.....	131
3.3 CIRCUITO DE VISUALIZACIÓN NUMÉRICA.....	133
3.4 CIRCUITO DE VISUALIZACIÓN ALFANUMÉRICA.....	136
3.5 CIRCUITO DE TECLADO MATRICIAL Y TECLAS.....	139
3.6 CIRCUITO DE ENTRADAS Y SALIDAS DIGITALES.....	144
3.6.1 Circuito de entradas digitales.....	144
3.6.2 Circuito de salidas digitales.....	145
3.7 DISEÑO DE LA TARJETA V&W – IN/OUT.....	146
3.8 ESPECIFICACIONES DEL MÓDULO.....	148
3.8.1 Descripción General.....	148
3.8.2 Especificaciones del equipo.....	149
3.8.3 Fuente de alimentación.....	150
3.8.4 Otras características.....	150

CAPITULO 4 DESARROLLO DE PRACTICAS

4.1 PRÁCTICA NO. 1: FAMILIARIZACIÓN CON EL EQUIPO.....	153
4.2 PRÁCTICA NO. 2: UTILIZACIÓN DEL PROGRAMA MONITOR BUFFALO.....	157
4.3 PRÁCTICA NO. 3: INTERRUPCIONES, TEMPORIZADORES Y CONTADORES.....	161
4.4 PRÁCTICA NO. 4: COMUNICACIÓN SERIAL.....	164
4.5 PRÁCTICA NO. 5: MANEJO DE DISPLAY ALFANUMÉRICO.....	166
4.6 PRÁCTICA NO. 6: UTILIZACIÓN DE CONVERSORES ANÁLOGO DIGITALES.....	168
4.7 PRÁCTICA NO. 7: SÍNTESIS DE SEÑALES DIGITALES.....	170
4.8 PRÁCTICA NO. 8: IMPLEMENTACIÓN DIGITAL DE ALGORITMOS DE CONTROL.....	172
4.9 PRÁCTICA NO. 9: IMPLEMENTACIÓN DE FILTROS DIGITALES.....	174

CAPITULO 5	
CONCLUSIONES Y RECOMENDACIONES.....	178

BIBLIOGRAFIA	181
---------------------------	------------

APENDICES

APENDICE A: CARACTERISTICAS ELECTRICAS DEL MC68HC11

APENDICE B: LISTA DE INSTRUCCIONES DEL PROGRAMA MONITOR

APENDICE C: ARCHIVOS "S-RECORDS"

APENDICE D: CIRCUITO IMPRESO DE LA TARJETA V&W - IN/OUT

APENDICE E: DIAGRAMA DE TARJETAS

CAPITULO No. 1

EL MICROCONTROLADOR MC68HC11

1.1 INTRODUCCION

En éste capítulo se realiza una descripción detallada del microcontrolador MC68HC11 (MCU), elemento principal en el funcionamiento del sistema didáctico de desarrollo V&W-HC11.

En la sección 1.2 se realiza una descripción de la familia de microcontroladores MC68HC11. En la sección 1.3 se analizan los distintos modos de operación del microcontrolador y su asignación de pines. Las secciones 1.4 y 1.5 están dedicadas al análisis del CPU, modos de direccionamiento y memorias internas del microcontrolador. La sección 1.6 aborda todo lo referente a los pórtricos de entrada y salida del microcontrolador. En las secciones 1.7 y 1.8 se realiza una descripción pormenorizada de los sistemas de comunicación serial sincrónicos y asincrónicos. La sección 1.9 analiza todo lo referente al sistema de conversión análogo - digital, mientras que en las secciones 1.10 y 1.11 se describen los sistemas de temporización, interrupciones y contadores. En la sección 1.12 se da una revisión a los sistemas de reset y modos de bajo consumo. Finalmente en la sección 1.13 se describe el funcionamiento de todas y cada una de las instrucciones de programación del microcontrolador MC68HC11.

1.2 LA FAMILIA DE MICROCONTROLADORES M68HC11

Primeramente realizaremos una descripción del microcontrolador más representativo de la familia de microcontroladores M68HC11 como es el MC68HC11A8, para posteriormente proceder a la descripción de las principales características de los demás microcontroladores.

1.2.1 CARACTERISTICAS

Entre las principales características del microcontrolador MC68HC11A8 tenemos:

CARACTERISTICAS DE HARDWARE

.8 Kbytes de memoria ROM

.512 Bytes de memoria EEPROM

.256 Bytes de memoria RAM

.Sistema de temporización de 16 bits.

4 periodos preescalares programables.

3 funciones de captura de entrada.

5 funciones de comparación de salida.

. Circuito acumulador de pulsos de 8 bits

. Interface de Comunicación Serial NRZ (SCI).

. Interface de periféricos serial (SPI).

. 8 canales de convertidores A/D de 8 bits.

. Circuito de interrupción en tiempo real.

. Sistema watchdog (COP) de computación propia de operación.

CARACTERISTICAS DE SOFTWARE.

- . Set de instrucciones M6800/M6801.
- . División entera y fraccional de 16x16.
- . Manipulación de bits.
- . Modo de espera (WAIT).
- . Modo de parada (STOP).

1.2.2 DESCRIPCION GENERAL

En este microcontrolador se incluyen: Ocho canales de conversión A/D de ocho bits de resolución, un Interfaz de comunicación serial (SCI) asincrónica y un Interfaz serial sincrónica (SPI). El principal sistema de temporización es de 16 bits de corrido-libre, tiene 3 líneas de captura de entrada, 5 líneas de comparación de salida y una función de interrupción en tiempo real. Un subsistema acumulador de pulsos de 8 bits puede contar eventos externos o medir periodos externos.

Una circuitería de monitoréo para protección en contra de errores en el sistema está incluida en el chip. Un sistema Watchdog (COP) computa las operaciones propias es decir protege contra fallas de software. Un sistema de monitoreo de reloj genera una acción de reset en caso de pérdida de reloj o corridas de reloj a frecuencias bajas. Un circuito de detección de códigos ilegales genera una interrupción (no enmascarable) cuando algún código ilegal es detectado.

Dos modos de operación de control de software, espera y paro (WAIT y STOP) están disponibles para ahorro de consumo de energía.

Un diagrama de bloques del MC68HC11 está mostrado en la Fig.1.1

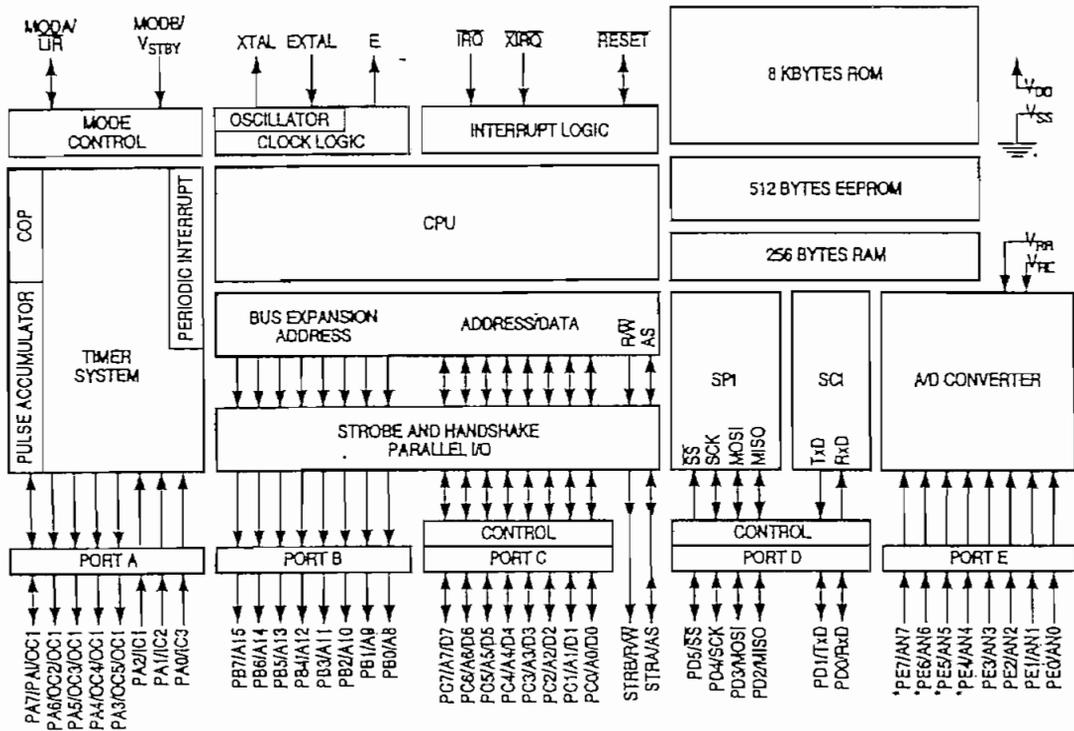


FIGURA No 1.1

DIAGRAMA DE BLOQUES DEL MC68HC11A8

1.2.3 MODELO DEL PROGRAMADOR

Además de ejecutar todas las instrucciones del M6800 y M6801, el MC68HC11A8, este microcontrolador permite la ejecución de 91 nuevos códigos. La fig.1.2 muestra los 7 registros del CPU los cuales están disponibles para el usuario.

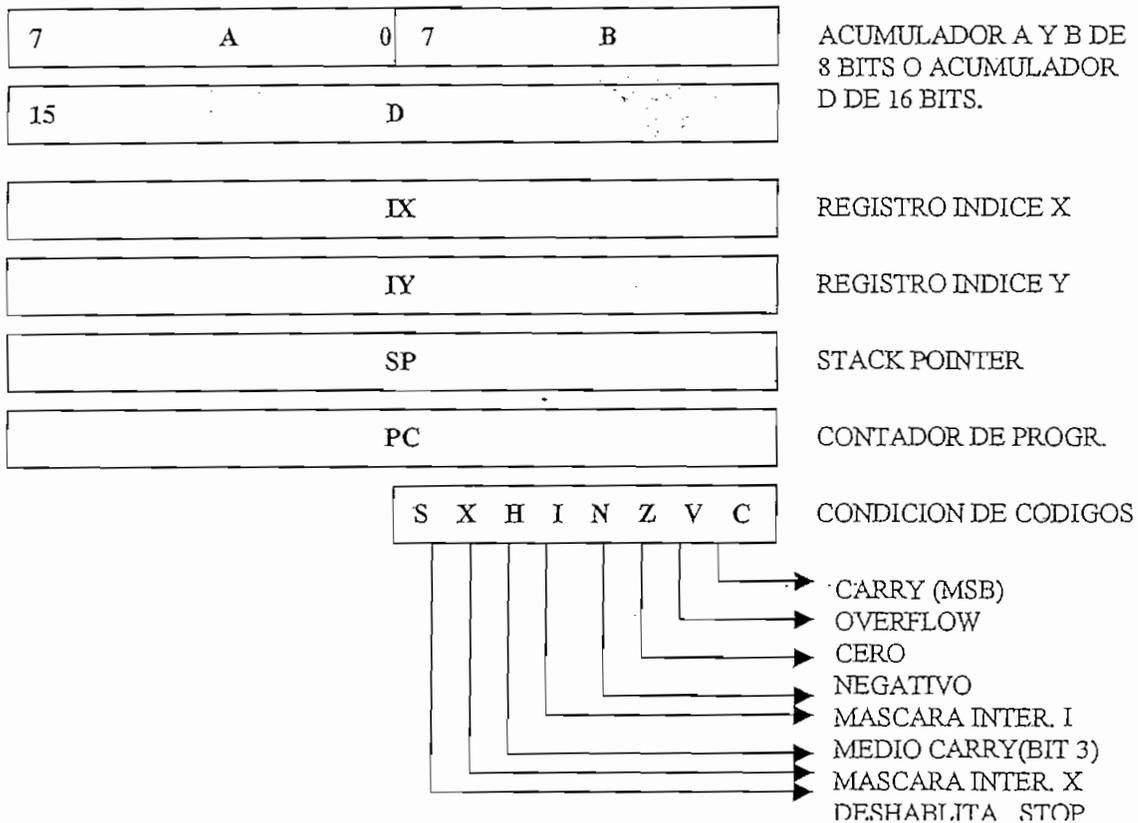


FIGURA 1.2

REGISTROS DEL CPU

1.2.4 FAMILIA DE MICROCONTROLADORES MC68HC11

La tabla 1.1 y los siguientes párrafos resumen los principales miembros de la familia de microcontroladores M68HC11. A continuación se describe la versión MC68HC11A8 y puede ser usada como una referencia general para varias otras versiones de la familia M68HC11, sin embargo con la excepción del CPU, algunos miembros más nuevos difieren grandemente de la familia de Microcontroladores MC68HC11A8 y su respectiva literatura técnica deberá ser revisada.

DISPOSITIVO	RAM	ROM	EPROM	EEPROM	COMENTARIOS
MC68HC11A8	256	8K	0	512	TIMER DE 16 BITS, 8 CANALES A/D
MC68HC11A7	256	8K	0	0	DE 8 BITS, SCI, SPI
MC68HC11A1	256	0	0	512	
MC68HC11A0	256	0	0	0	
MC68HC11D3	192	4K	0	0	TIMER DE 16 BITS, SCI, SPI
MC68HC711D3	192	0	4K	0	
MC68HC11D0	192	0	0	0	
MC68HC11ED0	512	0	0	0	TIMER DE 16 BITS, SCI, SPI
MC68HC11E9	512	12K	0	512	TIMER DE 16 BITS, 8 CANALES A/D
MC68HC711E9	512	0	12K	512	DE 8 BITS, SCI, SPI
MC68HC11E8	512	12K	0	0	
MC68HC11E1	512	0	0	512	
MC68HC11E0	512	0	0	0	
MC68HC811E2	256	0	0	2048	TIMER DE 16 BITS, 8 CANALES A/D. 2K EEPROM, SCI, SPI
MC68HC11E20	768	20K	0	512	TIMER DE 16 BITS, 8 CANALES A/D DE 8 BITS
MC68HC711E20	768	0	20K	512	, SCI, SPI, 20K ROM/EPROM
MC68HC11F1	1024	0	0	512	BUS NO MULTIPLEXADO, 8 CANALES A/D, 4 CS, SCI, SPI
MC68HC11G7	512	24K	0	0	BUS NO MULTIPLEXADO, 8 CANALES A/D, 4 CS, SCI, SPI
MC68HC11G5	512	16K	0	0	4 CANALES PWM, 66 PINES DE I/O
MC68HC711G5	512	0	16K	0	
MC68HC11G0	512	0	0	0	
MC68HC11K4	768	24K	0	640	BUS NO MULTIPLEXADO, EXPANSION DE MEMORIA 1 MB
MC68HC711K4	768	0	24K	640	8 CANALES A/D, 4 CANALES PWM, 4 CHIP SELECTS
MC68HC11K3	768	24K	0	0	
MC68HC11K1	768	0	0	640	
MC68HC11K0	768	0	0	0	
MC68HC11KA4	768	24K	0	640	BUS NO MULTIPLEXADO, 8 CANALES A/D, SCI, SPI,
MC68HC711KA4	768	0	24K	640	4 CANALES PWM
MC68HC11KA2	1024	32K	0	640	
MC68HC711KA2	1024	0	32K	640	
MC68HC11L6	512	16K	0	512	BUS MULTIPLEXADO, TIMER DE 16 BITS, 8 CANALES A/D
MC68HC711L6	512	0	16K	512	SCI, SPI
MC68HC11L5	512	16K	0	0	
MC68HC11L1	512	0	0	512	
MC68HC11L0	512	0	0	0	
MC68HC11M2	1280	32K	0	640	BUS NO MULTIPLEXADO, 8 CANALES A/D, 4 CANAL. PWM
MC68HC711M2	1280	0	32K	640	DMA, COOPROCESADOR MATEMATICO, SCI, SPI
MC68HC11N4	768	24K	0	640	BUS NO MULTIPLEXADO, 12 CAN. A/D, 2 CAN. D/A,
MC68HC711N4	768	0		640	6 CAN. PWM, COOPROCESADOR MATEMATICO, SCI, SPI
MC68HC11P2	1024	32K	0	640	BUS NO MULTIPLEXADO, PLL, 8 CAN. PWM
MC68HC711P2	1024	0	32K	640	3 SCI, SPI, 62 PINES I/O

TABLA No 1.1
FAMILIA DE MICROCONTROLADORES M68HC11

Varias de las series de dispositivos dentro de la familia HC11 tienen versiones x1 y x0. Estas son idénticas a los principales miembros pero tienen algunos de sus recursos deshabilitados. Por ejemplo, un MC68HC11A1 es idéntico al MC68HC11A8, excepto que su ROM está deshabilitada. Un MC68HC11A0 tiene deshabilitado los arreglos de EPROM y EEPROM, como se muestra en la tabla 1.1.

Casi todas las series dentro de la familia M68HC11 tienen una versión de EPROM y una versión de ROM.

Cualquier dispositivo en la familia M68HC11 que tiene un 7 precedente al 11 es un dispositivo que tiene EPROM en lugar de ROM versión (MC68HC711E9).

1.3 MODOS DE OPERACIÓN Y DESCRIPCIÓN DE PINES.

Cuando el microcontrolador está operando en el modo expandido multiplexado, 18 pines cambian de función para soportar una multiplexación del bus de datos y direcciones.

1.3.1 DESCRIPCIONES DE LAS SEÑALES EN LOS PINES

A continuación se realiza una descripción de las señales de entrada y salida, aunque también describiremos estas señales en otras secciones donde serán explicadas con mayor detalle.

1.3.1.1 ENERGÍA DE ENTRADA (VDD) Y TIERRA (VSS)

La energía es abastecida al microcontrolador usando los pines: VDD y VSS en donde VDD es la entrada de energía positiva y VSS es tierra, aunque el MC68HC11A8 es un dispositivo CMOS, las transiciones de señal muy rápidas pueden estar presentes en cualquiera de estos

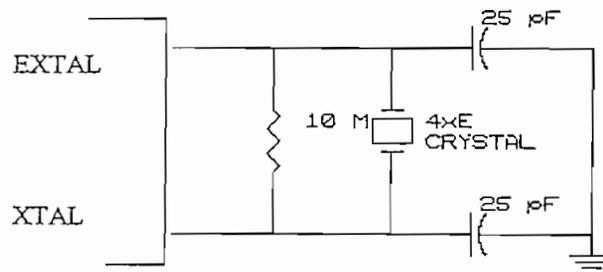
pinos, por lo que se debe incluir un capacitor cerámico de 0.1uf entre los pines de VDD y VSS el cual debe estar físicamente contiguo a uno de esos 2 pines para evitar interferencias.

1.3.1.2 RESET

Esta es una señal de control bidireccional de nivel activo bajo, y es usado como entrada para inicializar el MC68HC11A8 a un estado conocido de inicio, y también como salida de drenaje-abierto (open – drain) para indicar que una falla interna ha sido detectada en el sistema monitor de reloj o en el circuito de computo de operación apropiada (COP).

1.3.1.3 MANEJADOR DE CRISTAL Y ENTRADA DE RELOJ EXTERNO (XTAL, EXTAL)

Estos dos pines provéen el interfaz para un cristal o un reloj compatible CMOS para controlar la circuiteria de generación de reloj interno. La frecuencia establecida a esos pines será 4 veces mas alta que el rango de reloj interno deseado. El pin XTAL puede dejarse indeterminado cuando se usa una entrada de reloj compatible CMOS externa al pin EXTAL. Sin embargo, una resistencia de carga de 10K a 100K puesta a tierra podrá ser usada para reducir la emisión de ruido de RFI. La salida XTAL normalmente solo maneja un cristal.



Donde: E = frecuencia de reloj interna del microcontrolador.

FIGURA No 1.3

CONEXION COMUN DEL CRISTAL

1.3.1.4 SALIDA DE RELOJ (E)

Esta es la conexión de salida de la generación interna de reloj, que puede ser usada como una referencia de sincronización. La frecuencia de salida de reloj es actualmente un cuarto de la frecuencia de entrada en los pines XTAL Y EXTAL. La señal de reloj es detenida cuando el MCU está en modo STOP.

1.3.1.5 SOLICITUD DE INTERRUPCION ($\overline{\text{IRQ}}$).

La entrada de $\overline{\text{IRQ}}$ sirve para realizar una petición de interrupción asincrónica en el MC68HC11A8. Es seleccionable por el programador (registro OPTION) con una selección de: activado por estado bajo o activado por flanco negativo, y es siempre configurado para ser activado por estado bajo por la acción de reset. El pin IRQ requiere una resistencia externa de Pullup a VDD.

1.3.1.6 INTERRUPCION NO ENMASCARABLE ($\overline{\text{XIRQ}}$)

Esta entrada de interrupción es no enmascarable después de la inicialización del reset. Durante el reset, el bit X es seteado en el registro de código de condiciones CCR y cualquier interrupción es enmascarada hasta que sea habilitada por Software en el CCR. La entrada $\overline{\text{XIRQ}}$ es activada por nivel bajo y requiere una resistencia externa de Pullup conectada a VDD.

1.3.1.7 MODO A/REGISTRO DE INSTRUCCIÓN DE CARGA Y MODO B/VOLTAJE DE STANDBY (MODA/LIR, MODB/VSTBY).

Durante el RESET, MODA y MODB son usadas para seleccionar uno de los cuatro modos de operación. Referirse tabla 1.2

Después que el modo de operación ha sido seleccionado, el pin LIR provee una salida para indicar que una instrucción está iniciada. Todas las instrucciones son parte de una serie de

ciclos de reloj. La señal de LIR va a bajo durante los primeros ciclos de reloj para cada instrucción (OPCODE buscado). Esta salida es usada como una ayuda en la depuración de un programa.

La señal VSTBY es usada como la entrada de energía de standby para la memoria RAM.

MODB	MODA	MODO DE TRABAJO
1	0	Single Chip
1	1	Expandido Multiplexado
0	0	Especial Bootstrap
0	1	Especial Test

TABLA No 1.2

MODOS DE TRABAJO DEL MICROCONTROLADOR MC68HC11

1.3.1.8 VOLTAJES DE REFERENCIA PARA EL CONVERTIDOR A/D (VRL, VRH)

Estas dos entradas proveen los voltajes de referencia alto y bajo para el convertidor A/D

1.3.1.9 STROBE B Y READ / WRITE (STRB/R/W)

Esta señal actúa como una salida de habilitación programable (Strobe B) en el modo “single – chip” Refierace a la sección 1.6 Entradas y Salidas de datos para información adicional. Mientras que en el modo de trabajo expandido multiplexado actúa como un controlador de la dirección de transferencia en el bus de datos externo (R/W).

Un nivel bajo en el pin R/W indica que los datos serán escritos al bus de datos externos, y una señal en alto indica que un ciclo de lectura está en proceso.

Las señales R/\overline{W} junto con la señal de E clock podrán ser usadas como una señal de habilitación de escritura para una memoria RAM externa.

1.3.1.10 STROBE A Y STROBE DE DIRECCIONES (STRA/AS)

Esta señal actúa como una entrada de habilitación programable STRA en el modo de operación “single – chip”, y sirve para trabajar con otros dispositivos de entrada - salida paralela.

En el modo de operación multiplexado, la salida AS es usada para demultiplexar las direcciones y señales de datos en el pòrtico C. Refierace a la sección 1.3.2.2- Modo de operación Expandido Multiplexado para información adicional.

1.3.1.11 SEÑALES DE LOS PORTICOS.

La tabla 1.3 muestra un resumen de las señales de los pòrticos de los 40 pines y como ellos están relacionados con los modos de operación.

1.3.1.11.1 PORTICO A. El pòrtico A puede ser configurado para trabajar con tres funciones de captura de entrada (IC1,IC2,IC3), 4 funciones de comparación de salida (OC2,OC3,OC4,OC5), y ambas: Una entrada del acumulador de pulsos (PAI) o una quinta función de comparación de salida. Refierace a la sección 1.10.1 Timer Programable para información adicional.

Cualquier pin del pòrtico A que no es usado para estas funciones de entrada - salida de temporización, pueden ser usadas como líneas de entrada o salida de propósito general.

1.3.1.11.2 PORTICO B. Mientras el microcontrolador está trabajando en el modo de operación “single – chip” todos los pines del pòrtico B son pines de salida de propósito general, pero cuando esta trabajando en modo expandido - multiplexado, todos los pines del pòrtico B actúan como señales de salida de direccionamiento de orden alto. Durante cada ciclo del MCU, los bits de las direcciones desde el 8 hasta el 15 son sacados en los pines PBO-PB7 respectivamente.

BIT DEL PORTICO	MODO SINGLE CHIP Y MODO BOOTSTRAP	EXPANDIDO MULTIPLEXADO Y MODO ESPECIAL DE TEST
A-0	PA0/IC3	PA0/IC3
A-1	PA1/IC2	PA1/IC2
A-2	PA2/IC1	PA2/IC1
A-3	PA3/OC5 o OC1	PA3/OC5 o OC1
A-4	PA4/OC4 o OC1	PA4/OC4 o OC1
A-5	PA5/OC3 o OC1	PA5/OC3 o OC1
A-6	PA6/OC2 o OC1	PA6/OC2 o OC1
A-7	PA7/OC2 o OC1	PA7/OC2 o OC1
B-0	PB0	A8
B-1	PB1	A9
B-2	PB2	A10
B-3	PB3	A11
B-4	PB4	A12
B-5	PB5	A13
B-6	PB6	A14
B-7	PB7	A15
C-0	PC0	A0/D0
C-1	PC1	A1/D1
C-2	PC2	A2/D2
C-3	PC3	A3/D3
C-4	PC4	A4/D4
C-5	PC5	A5/D5
C-6	PC6	A6/D6
C-7	PC7	A7/D7
D-0	PD0/RxD	PD0/RxD
D-1	PD1/TxD	PD1/TxD
D-2	PD2/MISO	PD2/MISO
D-3	PD3/MOSI	PD3/MOSI
D-4	PD4/SCK	PD4/SCK
D-5	PD5/SS, STRA, STRB	PD5/SS, AS, R/W
E-0	PE0/AN0	PE0/AN0
E-1	PE1/AN1	PE1/AN1
E-2	PE2/AN2	PE2/AN2
E-3	PE3/AN3	PE3/AN3
E-4	PE4/AN5	PE4/AN5
E-5	PE5/AN5	PE5/AN5
E-6	PE6/AN6	PE6/AN6
E-7	PE7/AN6	PE7/AN6

TABLA 1.3

RESUMEN DE LAS SEÑALES DE LOS PORTICOS

1.3.1.11.3 PORTICO C. Mientras el microcontrolador esté trabajando en el modo de operación “single – chip” todos los pines, son pines de entrada / salida de propósito general. Las entradas del pòrtico C pueden ser retenidas en un registro si se realiza una transición de entrada a la señal STRA. EL pòrtico C puede también ser usado para trabajo en modos de operación full handshake para I/O paralelas donde la entrada STRA. y la salida STRB actúan como líneas de control de handshake.

Cuando el microcontrolador está trabajando en el modo de operación expandido multiplexado, todos los pines están configurados para trabajar con señales de direcciones y datos en forma multiplexada. Durante el direccionamiento en la parte respectiva de cada ciclo del MCU, los bits 0 hasta 7 de las direcciones son puestos en las líneas de salida PC0-PC7. Durante los datos, en la parte del ciclo del MCU (E alta), los pines 0 hasta 7 son señales de datos bidireccionales (D0-D7). La dirección de los datos en los pines del pòrtico C son definidos por la señal R/W.

1.3.1.11.4 PORTICO D. Los pines del Pòrtico D, desde el cero hasta el cinco pueden ser usados como señales de I/O de propósito general. Los pines del pòrtico D alternativamente sirven como señales para el interfaz serial de comunicaciones (SCI) y el interfaz serial de periféricos (SPI) cuando esos subsistemas son habilitados.

El pin PD0 es la señal de entrada de recepción de datos (RxD) para el interfaz de comunicación serial (SCI) .

El pin PD1 es la señal de salida de transmisión de datos TxD para el SCI

Los pines PD2 hasta PD5 son dedicados al SPI, PD2 es la señal master -in – slave- out (MISO), PD3 es la señal de master -out - slave - in (MOSI) PD4 es la señal de reloj (SCK) y PD5 es la entrada de selección de slave (esclavo) SS .

1.3.1.11.5 PORTICO E. El p3rtico E es usado como entrada de prop3sito general I/O o como canales de entrada an3logo - digital (A/D). La lectura del p3rtico E durante el tiempo de muestreo de conversi3n A/D, podr3a causar peque1as perturbaciones y afectar la exactitud de los resultados. Si necesitamos muy alta exactitud, debe evitarse la lectura del p3rtico E durante las conversiones.

1.3.2 MODOS DE OPERACI3N.

Hay 4 modos de operaci3n para el MC68MC11A8 que son: modo de operaci3n "single - chip", modo de operaci3n expandido - multiplexado, modo de operaci3n especial bootstrap, modo de operaci3n especial de prueba (test).

La tabla 1.2 muestra como seleccionar el modo de operaci3n.

1.3.2.1 MODO DE OPERACI3N SINGLE - CHIP

En el modo de operaci3n de "single - chip", el MC68HC11A8 funciona sin direccionamiento externo o bus de datos; el p3rtico B, el p3rtico C, la l3nea strobe A y la l3nea strobe B funcionan como se1ales de I/O de prop3sito general y handshake. Referirse a la secci3n 1.6.4 I/O Paralelas para informaci3n adicional.

1.3.2.2 MODO DE OPERACI3N EXPANDIDO - MULTIPLEXADO

En el modo de operaci3n expandido - multiplexado, el MC68MC11A8 tiene la capacidad de acceder a 64 kbyte de espacio de direcciones. Este espacio de direccionamiento incluye las mismas direcciones de memoria del microcontrolador usado para el modo de operaci3n de "single - chip", mas dispositivos perif3ricos y memorias externas. La expansi3n del bus es hecha sobre el p3rtico B y el p3rtico C y se1ales de control AS y R/W. La figura 1.4 muestra un camino recomendado para la multiplexaci3n de las direcciones de orden bajo y los datos en el p3rtico C.

Las señales de direcciones, R/W y AS están activas y son válidas para todos los ciclos del bus incluyendo el acceso a localidades de memoria externa.

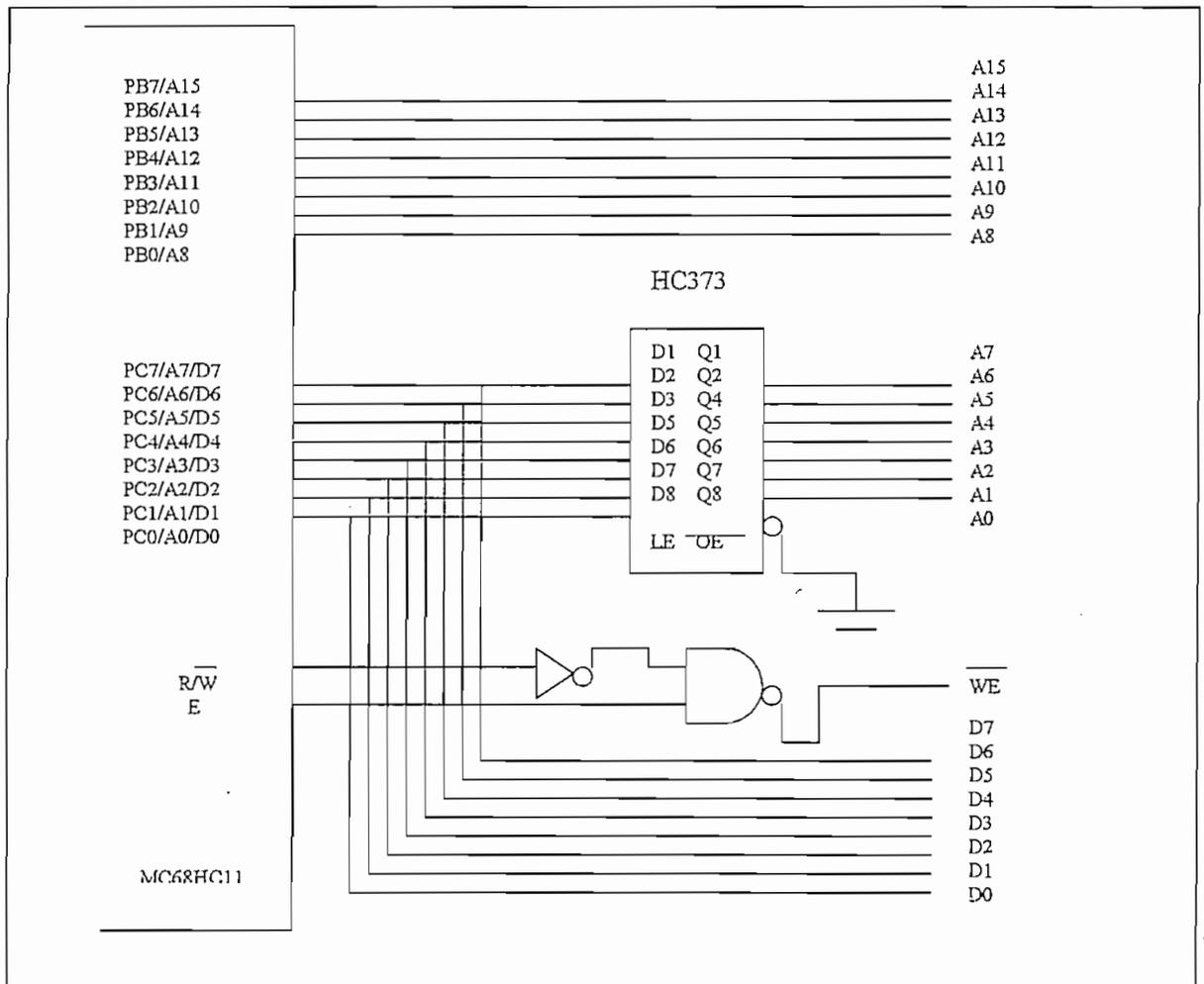


FIGURA No. 1.4

DEMULTIPLEXADO DE DIRECCIONES Y DATOS.

1.3.2.3 MODO DE OPERACION ESPECIAL BOOTSTRAP

El modo bootstrap es considerado un modo especial de operación totalmente diferente al modo normal de operación de “single – chip”. Este modo de operación es muy versátil porque no

hay limitaciones en el programa de propósito especial que puede ser cargado en la RAM interna . El programa cargado "BOOT" está contenido en los 192 bytes de ROM de bootstrap. Esta ROM es habilitada solo si el MCU es reseteado en modo de operación bootstrap, y aparece como un espacio de memoria interna en las localidades \$BF40 - \$BFFF. El programa cargado boot de inicialización usará el SCI para cargar un programa de 256 bytes dentro de la RAM en el chip en las localidades \$0000-\$00FF. Después que el carácter de la dirección \$00FF es recibido el control pasa automáticamente al programa cargado en las localidades \$0000.

El MC68HC11A8 se comunica a través del p \acute{o} rtico SCI. Después del reset en modo de operación especial bootstrap el SCI correrá a una frecuencia de $E/16$ (7812 baudios para un reloj E de 2MHZ). Al sacar el caracter \$FF por el transmisor SCI, la memoria EEPROM es borrada. Si el borrado no tiene éxito, \$FF es sacado otra vez y el borrado es intentado otra vez. Si se ha tenido éxito en el borrado de la memoria EEPROM, toda la memoria RAM es escrita con \$FF y luego el registro CONFIG es borrado. El programa cargado boot ahora procede como si las partes no han tenido modo de seguridad.

Una vez completado el proceso de borrado indicado en el párrafo anterior, un caracter de ruptura es sacado por el transmisor SCI; Para normal uso del programa cargado boot, el usuario envía \$FF al receptor SCI. El caracter \$FF no debe ser repetido a través del transmisor SCI.

Ahora el usuario debe descargar 256 bytes de datos de programa y serán puestos en el inicio de la memoria RAM en las localidades \$0000. Cuando la carga es completa el programa salta a las localidades \$0000 y comenzará ejecutando el código.

DIRECCION	VECTOR
00C4	SCI
00C7	SPI
00CA	ENTRADA DEL ACUMULADOR DE PULSOS
00CD	OVERFLOW DEL ACUMULADOR DE PULSOS
00D0	EVERFLOW DEL TIMER
00D3	OUTPUT COMPARE 5 DEL TIMER
00D6	OUTPUT COMPARE 4 DEL TIMER
00D9	OUTPUT COMPARE 3 DEL TIMER
00DC	OUTPUT COMPARE 2 DEL TIMER
00DF	OUTPUT COMPARE 1 DEL TIMER
00E2	INPUT CAPTURE 3 DEL TIMER
00E5	INPUT CAPTURE 2 DEL TIMER
00E8	INPUT CAPTURE 1 DEL TIMER
00EB	INTERRUPCION EN TIEMPO REAL
00EE	IRQ
00F1	XIRQ
00F4	SWI
00F7	ILEGAL OPCODE
00FA	FALLAS DEL COP
00FD	MONITOREO DE RELOJ
BF40	RESET

TABLA No 1.4

VECTORES DE INTERRUPCION (MODO BOOTSTRAP)

Si el pin de transmisión SCI va a ser usado, una resistencia externa de pullup es requerida porque los pines del p rtico D son configurados para operaciones wire – or (drenaje-abierto).

En modo operando especial bootstrap los vectores de interrupci n est n direccionados a la RAM como se muestra en la tabla 1.4. Esto permite al usuario usar interrupciones utilizando una tabla de saltos. Por ejemplo para usar la interrupci n SWI una instrucci n de salto ser  ubicado en las localidades \$00F4, \$00F5, y \$00F6.

Cuando una SWI es encontrada, el vector (el cual est  en el programa de ROM cargado boot) dirigir  el control de programa a la localidad \$00F4 en la RAM, la cual tiene una instrucci n de salto a la rutina de servicio.

1.3.2.4.- MODO DE OPERACI N ESPECIAL DE PRUEBA.

El modo de operaci n especial de prueba es usado para pruebas de fabricaci n. Este modo es muy similar al modo de operaci n expandido - multiplexado. En el modo de operaci n especial de prueba, el reset y los vectores de interrupci n son tra dos de la memoria externa de las localidades \$BFC0 - \$BFFF y no de \$FFC0 - \$FFFF. No hay tiempo l mite para proteger los registros TMSK2, OPTION, e INIT, as  esos registros pueden ser escritos repetidamente. Tamb n el registro especial TEST1 es habilitado el cual permite que varias de las funciones de test de fabricaci n sean invocadas.

Este modo especial de test no es recomendado para ser usado por un usuario final porque reduce la seguridad del sistema.

1.4. MEMORIAS INTERNAS DEL MICROCONTROLADOR.

Esta secci n describe las memorias ROM, RAM y EPROM del microcontrolador.

El registro INIT permite a la memoria RAM y a los 64 registros de control del microcontrolador ser movidos y acomodados a las necesidades de una aplicación particular.

1.4.1 MAPAS DE MEMORIA

La composición de los mapas de memoria para cada modo de operación son mostrados en la figura 1.5. Las localidades son mostradas en la parte izquierda y los contenidos de esas localidades se muestran a la derecha. Esos modos incluyen, “single chip”, expandido multiplexado, especial bootstrap y especial de test.

En el modo de operación de “single – chip” no se genera direccionamiento externo.

En el modo expandido multiplexado, las localidades son básicamente las mismas como el modo de operación simple, sin embargo; las localidades entre las áreas rayadas (designadas EXT) son para direccionamiento externo de memoria y dispositivos de I/O. La localización de memoria en el modo de operación especial bootstrap es similar a las localidades de memoria del modo simple de operación, excepto que el programa bootstrap está en las localidades \$BF40 hasta \$BFFF cuando es habilitado. El reset y los vectores de interrupción están direccionados en \$BFC0 - \$BFFF, estos vectores direccionados están dentro de los 192 bytes de memoria usada para el programa bootstrap.

El modo de operación especial de prueba tiene un mapa de memoria similar como del modo de operación expandido - multiplexado excepto que el reset y los vectores de interrupción son *localizados* en la memoria externa en las localidades \$BFC0 - \$BFFF.

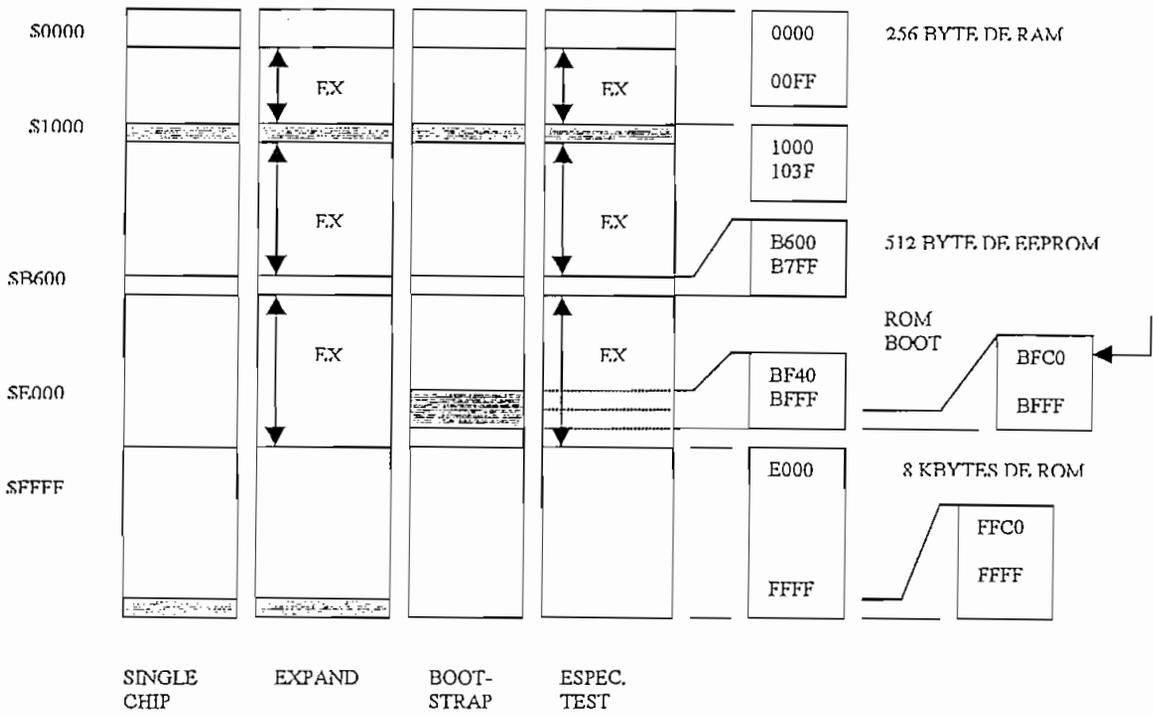
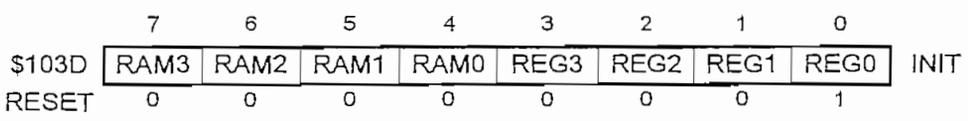


FIGURA No 1.5

MAPAS DE MEMORIA

1.4.2 REGISTRO INIT

El registro INIT es un registro de 8 bit de propósito general el cual puede ser usado durante la inicialización para cambiar las localidades de inicio de la memoria RAM y los registros de control dentro del mapa de memoria del microcontrolador. Este puede ser escrito una vez dentro de los 64 primeros ciclos de reloj inicial después de un reset, después de eso llega a ser un registro solo de lectura.



La dirección de inicio cuando no se especifica otra dirección para la RAM interna es \$0000 y para los 64 registros de control es \$1000 (el registro INIT es fijado a \$01 en el reset). Los primeros 4 bits del registro INIT especifican el inicio de dirección para los 256 bytes de memoria RAM y los 4 bits bajos del registro INIT especifican el inicio de las direcciones de los registros de control.

Note que si la memoria RAM es localizada en: \$E000 o \$F000, esta entra en conflicto con la memoria ROM interna (no hay conflicto si el bit ROMON en el registro de configuración es cero) la RAM tomará prioridad y la ROM llegará a ser inaccesible.

También si los 64 registros de control son relocalizados de modo que ellos tengan conflicto con la RAM y/o ROM, los 64 registros de control toman prioridad y las memorias RAM y/o ROM llegarán a ser inaccesibles.

1.4.3 MEMORIA ROM

La memoria ROM interna ocupa los 8 Kbytes altos del mapa de memoria (\$E000-\$FFFF) esta memoria ROM es deshabilitada cuando el bit ROMON del registro CONFIG está encendido. El bit ROMON es implementado con una celda EEPROM y es programado usando el mismo procedimiento para programar el chip de EEPROM. Para mas información referirse al Registro de configuración de sistema (CONFIG).

Hay también 192 bytes enmascarados adicionales en la memoria ROM del MC68MC11A8. Este bootstrap de ROM de programa controla la operación del modo especial de operación bootstrap y solo es habilitado después del RESET durante el modo de operación especial bootstrap. Para más información referirse a la sección 1.3.2.3. Modo de operación BOOTSTRAP

1.4.4 MEMORIA RAM

Los 256 bytes de memoria RAM interna pueden ser reubicados durante la reinicialización del sistema por escritura en el registro INIT. En el reset si no se especifica otra dirección, la memoria RAM se ubica en las posiciones \$0000 hasta \$00FF. Esta RAM es implementada con celdas estáticas que retienen el contenido durante los modos de espera y paro. Este contenido se retiene a través de una fuente de energía de respaldo de baja corriente colocada al pin MODB/V_{STBY}.

1.4.5 MEMORIA EEPROM

Los 512 bytes de memoria EEPROM son localizadas desde \$B600 hasta \$B7FF y tienen el mismo tiempo de ciclo de lectura como la memoria ROM interna. El mecanismo de escritura o programación para la memoria EEPROM es controlado por el registro PPROG. La memoria EEPROM es deshabilitada cuando el bit EEON en el registro CONFIG es 0. El bit EEON es implementado con una celda EEPROM.

El estado de borrado de un byte de la memoria EEPROM es \$FF. Al programar cambian unos a ceros. Si algún bit en una localidad necesita ser cambiado de cero a uno, el byte debe ser borrado antes que sea programado. Si un nuevo byte de datos no tiene unos en posiciones de bits las cuales fueron ya programadas o ceros, es aceptable programar los nuevos datos sin borrar el primer byte de memoria EPROM. Por ejemplo programar \$50 a la localidad la cual estaba en \$55 cambiaría la localidad en \$50.

El tiempo recomendado para borrar es 10 milisegundos cuando el reloj E es de 2 MHz y será incrementado a 20 milisegundos cuando E está entre 1 MHz y 2 MHz.

Cuando el reloj E es menor a 1 MHz, la fuente de reloj para la carga deberá ser switchheada desde el sistema de reloj a un reloj de oscilador RC. Esto será hecho por Seteo del bit CSEL del registro OPTION. Un periodo de 10 milisegundos será permitido después del seteo del bit CSEL permitiendo la estabilización del sistema.

Note que el bit CSEL también controla un reloj para el subsistema de conversión análogo digital.

1.4.5.1 REGISTRO DE CONTROL DE PROGRAMACION EEPROM (PPROG).

Este registro de 8 bits es usado para controlar la programación y borrado de los 512 bytes de memoria EEPROM. El RESET limpia este registro, así la memoria EEPROM es configurada para lecturas normales.

	7	6	5	4	3	2	1	0	
\$103B	ODD	EVEN	0	BYTE	ROB	ERASE	EELAT	EEPGM	PPROG
RESET	0	0	0	0	0	0	0	0	

ODD: Utilizado solo en el modo de prueba

EVEN: Utilizado solo en el modo de prueba

Bit 5: No implementado

BYTE: Selecciona el borrado de byte.

0 = borrado de fila o borrado en conjunto

1 = borra un solo byte

ROW: Selecciona el borrado de fila.

0 = Borrado en conjunto

1 = Borrado de fila

ERASE: Selecciona el modo de borrado

0 = Lectura normal o programa

1 = Modo de borrado

EELAT: Control del LATCH de EEPROM

0 = Dirección de EEPROM y configuración de datos para modo de lectura.

1 = Dirección de EEPROM y configuración de datos para programación/borrado.

EEPGM: Habilitación del voltaje de programación de la EEPROM

0 = Apagado el voltaje de programación

1 = Encendido el voltaje de programación

Si una prueba es hecha seteando ambos: los bits EELAT y EEGM en el mismo ciclo de escritura ninguno será seteado. Si una escritura a una dirección de memoria EEPROM es ejecutada mientras que el bit EEGM es seteado la escritura es ignorada y la operación actual de programación no será afectada. Estas seguridades fueron incluidas para prevenir cambios accidentales en la memoria EEPROM.

1.4.5.2 REGISTRO DE CONFIGURACION DE SISTEMA (CONFIG)

La configuración del registro de control (CONFIG) es implementado con celdas de EEPROM y controlan la presencia de ROM y EEPROM en el mapa de memoria, así como el sistema de vigilancia COP y el modo de seguridad.

	7	6	5	4	3	2	1	0	
\$103F	0	0	0	0	NOSEC	NOCOP	ROMON	EEON	CONFIG

RESET

Bits 7.6.5 y 4: No implementados

NOSEC: Bit de deshabilitación del modo de seguridad.

0 = Modo de seguridad Habilitado

1 = Modo de seguridad deshabilitado

NOCOP: Deshabilitación del sistema COP

0 = Habilitación del sistema de vigilancia COP

1 = Deshabilitación del sistema de vigilancia COP

ROMON: Habilitación de la memoria ROM en el microcontrolador.

Cuando este bit es borrado, los 8K de ROM es deshabilita. En el modo de “single-chip”, los 8K de ROM interna son habilitados sin tomar en cuenta el estado del bit ROMON.

EEON: Habilitación de la memoria EEPROM en el microcontrolador.

Cuando está borrado, los 512 Bytes de EEPROM son deshabilitados.

1.4.5.2.1 PROGRAMACION Y BORRADO DE EL REGISTRO CONFIG

Como el registro CONFIG es implementado con celdas EEPROM, cuidado especial se debe tener para borrar y programar este registro. La programación sigue el mismo procedimiento como programar un byte en los 512 bytes de memoria EEPROM excepto que la dirección del registro CONFIG es usado. Cuando el registro CONFIG es borrado, los 512 bytes de vectores de EEPROM son también borrados.

1.4.5.2.2 OPERACIONES PARA CAMBIAR EL REGISTRO DE CONFIGURACION.

El registro CONFIG consiste de un byte de EEPROM y latches estáticos. Este registro controla la configuración de inicio del MCU.

El procedimiento para cambiar el registro CONFIG es como sigue.

1.- Borrar el registro CONFIG

No efectuar un reset en este tiempo

2.- Programar el nuevo valor al registro CONFIG

3.- Efectuar en reset para que la nueva configuración tome efecto.

1.5 CPU Y MODOS DE DIRECCIONAMIENTO

Esta sección provee una descripción de los registros del CPU y los modos de direccionamiento del MC68HC11.

1.5.1 REGISTROS DEL CPU

Además de ser capaz de ejecutar todas las instrucciones del M6800 y M6801 el MC68HC11A8 usa un mapa de opcodes de 4 páginas para permitir la ejecución de 91 nuevos opcodes. Siete registros son discutidos en los siguientes párrafos, los cuales están disponibles para el programador como muestra la figura No. 1.2

1.5.1.1 ACUMULADORES A Y B

El acumulador A y el acumulador B son registros de 8 bits de propósito general para retener operaciones y resultados de cálculos aritméticos o manipulación de datos. Estos 2 acumuladores pueden ser concatenados en un simple acumulador de 16 bits llamado el acumulador D

1.5.1.2 REGISTRO INDICE X (IX)

El registro IX de 16 bits es usado para modo de direccionamiento indexado. Este provee un valor indexado de 16 bits, el cual es añadido a un offset de 8 bits dado en una instrucción para crear una dirección efectiva. El registro IX puede ser usado como un controlador o como un registro de almacenamiento temporal.

1.5.1.3 REGISTRO INDICE Y (IY)

El registro IY de 16 bits es usado para modo de direccionamiento indexado similar al registro IX, sin embargo todas las instrucciones que usan el registro IY requieren un byte extra de código de máquina y un ciclo extra de tiempo de ejecución.

1.5.1.4 STACK POINTER (SP)

El puntero de pila (SP) es un registro de 16 bits que contiene las direcciones de la próxima localidad libre en la pila. El STACK es configurado como registro de lectura, escritura de una secuencia de la última entrada y primera salida lo cual permite que datos importantes sean almacenados durante interrupciones y llamadas a subrutinas. Cada vez que un nuevo byte es añadido al stack (una instrucción PUSH), el SP es decrementado mientras que, cada vez que un byte es removido desde el stack (una instrucción PULL) el SP es incrementado.

1.5.1.5 CONTADOR DE PROGRAMA (PC)

El contador de programa es un registro de 16 bits que contiene la dirección de la próxima instrucción a ser ejecutada.

1.5.1.6 REGISTRO DE CODIGO DE CONDICIONES (CCR)

Este es un registro de 8 bits en el cual cada bit significa el resultado de una instrucción ejecutada. Estos bits pueden ser individualmente verificados por un programa y una acción específica puede tomarse como resultado de una verificación. A continuación se presentan los diferentes bits del CCR.

1.5.1.6.1 CARRY (C)

El bit C es seteado si hubo un carry en la unidad aritmética lógica (ALU) durante la última operación aritmética lógica. EL bit C es también afectado durante instrucciones de desplazamiento y rotación.

1.5.1.6.2 OVERFLOW / DESBORDAMIENTO (V).

El bit de desbordamiento es seteado si hubo un desbordamiento aritmético, como resultado de una operación, de otro modo el bit V es encerado.

1.5.1.6.3 ZERO (Z).

El bit cero es seteado si el resultado de la última operación aritmética lógica, u operaciones de manipulación de datos fue cero.

1.5.1.6.4 NEGATIVO (N).

El bit negativo es seteado si el resultado de la última operación aritmética lógica u operaciones de manipulación de datos fue negativo; de otro modo el bit es encerado.

Un resultado es negativo si el bit más significativo es uno.

1.5.1.6.5 INTERRUPCION ENMASCARABLE (I).

El bit I de interrupción enmascarable es seteado por hardware o instrucciones de programa para deshabilitar todas las fuentes de interrupción enmascarables

1.5.1.6.6 MEDIO CARRY (H)

EL bit medio carry es seteado cuando ocurre un carry entre los bits 3 y 4 de la unidad aritmética lógica durante una instrucción ADD, ABA y ADC, de otro modo el bit H es encerrado.

1.5.1.6.7 INTERRUPCION ENMASCARABLE (X).

El bit X de la interrupción enmascarable es seteado solo por (RESET o XIRQ); y es borrado solo por instrucciones de programa (TAP o RTI)

1.5.1.6.8 DESHABILITACION DE PARADA (S).

El bit de deshabilitación de parada es seteado para deshabilitar la instrucción STOP, y borrado para habilitar la instrucción STOP, La instrucción STOP es tratada como no operación (NOP) si el bit S es fijado .

1.5.2 MODOS DE DIRECCIONAMIENTO

Seis modos de direccionamiento pueden ser usados para referirse a la memoria; ellos incluyen direccionamiento inmediato, directo, indexado.

Los siguientes párrafos proveen una descripción de cada modo de direccionamiento y una discusión del prebyte. En estas descripciones el término dirección efectiva es usado para indicar la dirección en memoria desde la cual el argumento es traído o almacenado, o desde la cual la ejecución es procesada.

1.5.2.1 DIRECCIONAMIENTO INMEDIATO

En el modo de direccionamiento inmediato, el argumento actual está contenido en el (los) byte(s) inmediatamente seguido de la instrucción. Estas son instrucciones de dos, tres o cuatro (si el prebyte es requerido) bytes.

1.5.2.2 DIRECCIONAMIENTO DIRECTO

En el modo de direccionamiento directo el byte menos significativo de la dirección del operando está contenida en un simple byte siguiendo el opcode y el byte mas significativo es asumido como \$00. El direccionamiento directo permite al usuario acceder desde \$0000 hasta \$00FF usando instrucciones de 2 bytes y el tiempo de ejecución es reducido por eliminación del acceso a la memoria adicional. En la mayoría de aplicaciones, esta área de 256 bytes son reservados para datos referenciados frecuentemente. En el MC68HC11A8, el software puede

configurar el mapa de memoria de modo que la memoria RAM interna y/o registros internos, o espacios de memoria externa pueden ocupar estas direcciones.

1.5.2.3 DIRECCIONAMIENTO EXTENDIDO

En el modo de direccionamiento extendido, el segundo y tercer bytes contiene las direcciones absolutas del operando. Estas son instrucciones de 3 o 4 (si el prebyte es requerido) bytes: 1 o 2 para el opcode y 2 para la dirección efectiva.

1.5.2.3 DIRECCIONAMIENTO INDEXADO

En el modo de direccionamiento indexado, uno de los registros (X o Y) son usados para calcular la dirección efectiva. En este caso, la dirección efectiva es variable y depende de dos factores: Primero.- el contenido corriente de los registros índices (X o Y) que son usados y; Segundo.- el offset asignado de 8 bits contenido en la instrucción. Este modo de direccionamiento permite referenciar una localidad de memoria en los 64 Kbytes de espacio de direcciones. Estos son usualmente instrucción de 2 o 3 (si el prebyte es requerido) bytes, el opcode mas los 8 bits de offset.

1.5.2.4 DIRECCIONAMIENTO INHERENTE

En el modo de direccionamiento inherente, toda la información esta contenida en el opcode. Los operandos (si los hay) son registros y no se requiere referenciar a ninguna localidad de memoria. Estas son usualmente instrucciones de 1 o 2 bytes.

1.5.2.5 DIRECCIONAMIENTO RELATIVO

El modo de direccionamiento relativo es usado para instrucciones de salto. Si la condición de salto es verdadera, el contenido de los 8 bits del byte que sigue al opcode (el offset) es

añadido al contador del programa para formar la dirección de salto efectivo, de otro modo, el control procede a la próxima instrucción. Estas son usualmente instrucciones de 2 bytes.

1.5.2.6 PREBYTE

Para expandir el número de instrucciones usadas en el MC68HC11A8, un prebyte de instrucción tiene que ser añadido a ciertas instrucciones. Las instrucciones afectadas son usualmente asociadas al registro índice Y. Los opcodes de instrucción que no requieren prebyte serán consideradas como página 1 de todos los mapas de opcodes. Los opcodes restantes serán considerados como páginas 2, 3 y 4 del mapa de opcode y requerirán un prebyte; \$18 para la página 2, \$1A para la página 3, y \$CD para la página 4.

1.6 PORTICOS DE ENTRADAS Y SALIDAS DE DATOS.

El MC68HC11A8 tiene 40 pines de I/O distribuidos como 5 pórticos de 8 bits. Todos esos pines sirven para múltiples funciones, dependiendo del modo de operación y de los valores cargados en los registros de control.

1.6.1 PORTICOS C y D (ENTRADAS Y SALIDAS DE PROPOSITO GENERAL)

Cada línea de entrada o salida del pórtico tiene un bit asociado a un registro específico de datos del pórtico y un registro de dirección de datos del pórtico. Los bits del registro de dirección de datos son usados para especificar la dirección primaria de datos para cada línea de I/O; Cuando una línea de salida es leída, el valor en la entrada del pin es regresada. Cuando una línea es configurada como una entrada, ese pin llega a ser una entrada de alta impedancia. Si una escritura es ejecutada a un línea de entrada el valor no afecta el pin de I/O pero es registrado en un latch interno. Cuando la línea llega a ser una salida, este valor aparece en el pin de I/O. Los bits del

registro de dirección de datos son borrados por el reset y los pines de I/O son configurados como entradas.

Los pines AS y R/W son dedicados al control del bus mientras se está operando en el modo expandido - multiplexado, o habilitación de I/O paralelas (STRA y STRAB) mientras está en el modo de operación de simple chip.

1.6.2 PORTICOS A, B y E (ENTRADAS O SALIDAS COMO DIRECCION ESPECIFICA)

Las líneas para los pórtricos A, B y E (excepto para el bit 7 para el pórtrico A) tienen dirección de datos fijas. Cuando el pórtrico A está siendo usado para I/O de propósito general, bits 0, 1, y 2 son configurados solo como entradas, una escritura en esas líneas no tienen efecto. Los bits 3, 4, 5 y 6 de el pórtrico A son configurados solo como salidas y si leemos de esas líneas regresa el nivel sensado en las entradas. El bit 7 del pórtrico A puede ser configurado como una entrada o como una salida de propósito general usando el bit DDRA7 en el registro de control del acumulador de pulsos. Cuando el pórtrico B está siendo usado para salida de propósito general es configurado solo como salida, y las lecturas de esas líneas regresarán el nivel sensado en la entrada de esos pines . El pórtrico E contiene 8 entradas de canales A/D, pero estas líneas pueden ser también usadas como entradas digitales de propósito general. Escrituras a la dirección del pórtrico E no tiene efecto.

1.6.3 HABILITACION SIMPLE DE ENTRADAS Y SALIDAS

El modo de habilitación de I/O paralelas es invocado y controlado por el registro de control de I/O paralelas (PIOC). Este modo es seleccionado cuando el bit de handshake (HNDS) en el registro PIOC es borrado.

El pórtrico C llega a ser un pórtrico de entrada habilitado con la línea STRA.

También el p rtico B llega ser un p rtico de salida habilitado con la l nea STRB como la habilitaci n de salida. El sentido l gico de la salida STRB es seleccionado por el bit B de habilitaci n inversa (INVB) en el registro PIOC.

1.6.3.1 HABILITACION DE ENTRADA DEL PORTICO C

En este modo hay dos direcciones donde el p rtico C puede ser le do, el registro de datos del portico C PORTC y alternativamente el registro del p rtico C latcheado (PORTCL). El registro de direcci n de datos todav a controla la direcci n de todas las l neas del p rtico C. Incluso cuando el modo de la direcci n de entrada es seleccionado, alguna o todas las l neas del p rtico C pueden ser usadas para I/O de prop sito general.

La l nea STRA es usada como una detecci n de entrada y la selecci n de cambio de nivel para habilitar STRA es el bit (EGA) del registro PIOC, el mismo define el nivel de subida o bajada. Cuando el nivel seleccionado es detectado en el pin STRA, el nivel l gico de las l neas en el p rtico C son latcheadas en el registro PORTCL y la bandera de Strobe A (STAF) en el registro PIOC es seteado. Si el bit de habilitaci n de interrupci n de strobe A (STAI) en el PIOC es tambi n seteado, una interrupci n interna es generada. La bandera STAF es autom ticamente limpiada por lectura del registro PIOC seguido por una lectura del registro PORTCL. El dato es retenido en el registro PORTCL sin tomar en cuenta si la bandera ha sido o no limpiada.

1.6.3.2 HABILITACION DE SALIDA DEL PORTICO B

En este modo, el pin STRB es una habilitaci n de salida la cual es pulsada por dos periodos de reloj cada vez que hay una escritura al p rtico B. El bit INVB en el registro PIOC controla la polaridad del pulso en la l nea STRB.

1.6.4 ENTRADAS Y SALIDAS PARALELAS EN EL MODO DE HANDSHAKE

Los modos de handshake total de I/O paralelas involucran al p rtico C, STRA y STRB. Hay dos modos b sicos (entrada y salida) y una variaci n adicional en el modo de salida de handshake que permite 3 estados de operaci n del p rtico C. En todos los modos de handshake, STRA es detector de nivel de entrada y STRB es una l nea de salida "listo" de handshake.

1.6.4.1 PROTOCOLO DE HANDSHAKE DE ENTRADA.

En el protocolo de handshake de entrada, el p rtico C es un p rtico de entrada que retiene la informaci n, STRA detecta el nivel para iniciar la retenci n desde un sistema externo que est  manejando al p rtico C, y STRB es una l nea de salida "listo" controlada por la l gica interna del microcontrolador.

Cuando una condici n de "listo" es reconocida, el dispositivo externo ubica los datos en las l neas del p rtico C, luego env a pulsos a la l nea STRA. El nivel activo en la l nea STRA retiene los datos del p rtico C en el registro PORTCL, seteando la bandera STAF (opcionalmente causando una interrupci n) y no toma en cuenta la l nea STRB. Este desertamiento de la l nea STRB autom ticamente inhibe el dispositivo externo de habilitar un nuevo dato en el p rtico C. Leyendo el registro de retenci n del p rtico C PORTCL (independiente de la limpieza de la bandera STAF) se pone en "listo" la l nea STRB, indicando que el nuevo dato puede ser ahora aplicado al p rtico C.

La l nea STRB puede ser configurado por el bit de control PSL como un pulso de salida (modo de pulsos) o como una salida de nivel (modo entrelazado).

Los bits del registro de direcci n de datos del p rtico C ser n borrados por cada l nea que sea usada como una l nea de entrada retenida.

La escritura, sea al registro de datos del p rtico PORTC o alternativamente al registro del p rtico C retenido (PORTCL) se env a al mismo p rtico C de salida sin afectar la entrada de habilitaci n de handshake.

1.6.4.2 PROTOCOLO DE HANDSHAKE DE SALIDA.

En el protocolo de handshake de salida el p rtico C es un p rtico de salida, STRB es una salida “lista” y STRA es el pin de se al de entrada de reconocimiento con nivel sensible usada para indicar al MCU que los datos de salida han sido aceptados por el dispositivo externo.

El MCU ubica los datos en las l neas de salida del p rtico C y luego indica que los datos est n estables y disponibles a trav s de la l nea STRB. El dispositivo externo luego procesa los datos disponibles y pulsa la l nea STRA indicando que el nuevo dato puede ser ubicado en las l neas de salida del p rtico C. El borde activo en la l nea STRA causa que la l nea STRB quede sin seguro y la bandera de estado STAF sea seteado. En respuesta al bit STAF seteado el programa transfiere el nuevo dato de salida del p rtico C. Escribiendo datos en el registro PORTCL se causa que estos aparezcan en las l neas del p rtico C y aseguren la l nea STRB.

Hay una variaci n del protocolo de handshake de salida que permite la operaci n de 3 estados en el p rtico C, por lo que es posible conectar este p rtico de 8 bits a otros dispositivos de 3 estados sin partes adicionales.

Mientras la l nea de entrada STRA es inactiva, todas las l neas del p rtico C obedecen la direcci n de datos especificado por el registro de direcci n de datos de modo que las l neas configuradas como entradas est n en alta impedancia. Cuando la l nea STRA es activada, todas las l neas del p rtico C son forzadas a salidas sin tomar en cuenta los valores en el registro de direcci n de datos.

1.6.5 REGISTRO DE CONTROL I/O PARALELAS (PIOC)

Las funciones I/O de handshake paralelas están disponibles solo en el modo de operación de "single - chip". EL PIOC es un registro de lectura y escritura excepto por el séptimo bit que es solo de lectura. La tabla-1.5 muestra un resumen de las operaciones de I/O con handshake

	STAI	CWOM	INVB
0	INHIBE INTERRUPCION STAF	SALIDA NORMAL PORT C	NIVEL ACTIVO BAJ0 (STRB)
1	HABILITACION INT. STAF	SALIDA DRENAJE-ABIERTO PORTICO C	NIVEL ACTIVO ALTO (STRB)

	SECUENCIA DE BORRADO STAF	HNDS	ON	PLS	EGA	PORT C	PORT B
MODO HABILITACION SIMPLE	LECTURA PIOC CON STAF=1 LUEGO LECTURA PORTCL	0	X	X		ENTRADAS LACHEADA EN EL PORTCL EN CUALQUIER NIVEL ACTIVO EN STRA	PULSOS EN STRB ESCRIBE EN EL PORTICO B
ENTRADA DE HANDSHAKE	LECTURA PIOC CON STAF=1 LUEGO LECTURA PORTCL	1	0	0=STRB NIVEL ACTIVO 1=STRB PULSO ACTIVO		ENTRADAS LACHEADA EN EL PORTCL EN CUALQUIER NIVEL ACTIVO EN STRA	PORTICO NORMAL DE SALIDA. NO AFECTADA EN MODO DE HANDSHAKE
SALIDA DE HANDSHAKE	LECTURA PIOC CON STAF=1 LUEGO ESCRIBIR PORTCL	1	1	0=STRB NIVEL ACTIVO 1=STRB PULSO ACTIVO	<p>DDRC NIVEL ACTIVO STRA</p>	MANEJADO COMO SALIDA SI STRA ESTA EN NIVEL ACTIVO.	PORTICO NORMAL DE SALIDA. NO AFECTADA EN MODO DE HANDSHAKE

TABLA No 1.5

RESUMEN DE LAS OPERACIONES DE I/O DE HANDSHAKE

	7	6	5	4	3	2	1	0	
\$1,002	STAF	STI	CWOM	HNDS	OIN	PLS	EGA	INVB	PIOC
RESET	0	0	0	0	0	U	1	1	

STAF: Bandera de estado de interrupción de strobe A

Este bit es fijado cuando se detecta un nivel strobe A

STAI: Máscara de habilitación de interrupción de strobe A

CWOM: Modo wire-or (drenaje abierto)

Afecta a todos los pines del pórtilo C

0 = Pórtilo C como salidas normales CMOS

1 = Pórtilo C como salidas drenaje- abierto

HNDS: Modo de Handshake

0 = Modo de habilitación simple

1 = Modo de entrada o salida de Handshake

OIN: Entrada o salida de Handshake.

0 = Handshake de entrada

1 = Handshake de salida

PLS: Operación de handshake (pulso/interlocked)

Este bit no tiene significado si HNDS = 0

0 = Selecciona el handshake interlocked (STRB activado hasta que se detecte el nivel en STRA)

1 = Selecciona el handshake por pulsos (STRB pulsado por dos ciclos de reloj)

EGA: Nivel activo para Strobe A

0 = Selecciona nivel de bajada para STRA

1= Selecciona nivel de subida para STRA

INVB: Strobe B invertida

0 = Nivel activo tiene un cero lógico

1 = Nivel activo tiene un uno lógico

1.7 INTERFAZ DE COMUNICACIÓN SERIAL SINCRONICA (SPI)

1.7.1 CARACTERISTICAS

El Interfaz de periféricos serial (SPI) es un Interfaz sincrónico el cual permite a varios microcontroladores SPI o periféricos tipo SPI ser interconectados entre sí. En un Interfaz de periférico serial, se requiere señales independientes para datos y reloj. En el formato SPI, el reloj no está incluido en los datos y debe ser suministrado como una señal separada. El sistema SPI del MC68HC11A8 puede ser configurado como maestro o como esclavo.

El SPI tiene las siguientes características.

- . Full Duplex, Transferencia Sincrónica a 3 hilos
- . Operación de Maestro o esclavo
- . Frecuencia de bit como Maestro 1.5 MHz (Máximo)
- . Frecuencia de bit como esclavo 3 MHz (Máximo)
- . Cuatro Bits de velocidad programables como maestro
- . Polaridad y Fase de reloj programable
- . Bandera de interrupción de fin de transmisión
- . Bandera de protección de colisión de escritura

. Protección para modo de fallo Master - Master

1.7.2 DESCRIPCIONES DE SEÑALES SPI

Las cuatro señales básicas SPI (MISO, MOSI, SCK Y SS) son discutidas en los siguientes párrafos. Cada señal es descrita para los modos de maestro y esclavo.

Todas las líneas de salida tienen su correspondiente bit de habilitación en un registro de dirección de datos. Si este bit es borrado, la línea es desconectada del SPI y llega a ser una línea de entrada de propósito general.

1.7.2.1 MAESTRO IN ESCLAVO OUT (MISO)

LA línea MISO es configurado como una entrada en un dispositivo master y como una salida en un dispositivo esclavo, ésta es una de las dos líneas que transfieren datos seriales en una dirección, el bit mas significativo es enviado primero. La línea MISO de un dispositivo esclavo es ubicado en un estado de alta impedancia, si el esclavo no está seleccionado.

1.7.2.2 MAESTRO OUT ESCLAVO IN (MOSI)

La línea MOSI es configurado como una salida en un dispositivo maestro y como una entrada en un dispositivo esclavo, esta es la otra línea que transfiere datos seriales en una dirección con el bit más significativo enviado primero.

1.7.2.3 RELOJ SERIAL (SCK)

El reloj es usado para el movimiento sincronizado de datos , tanto para entrada como para salida de los dispositivos a través de las líneas MOSI y MISO. Los dispositivos maestros y esclavos son capaces de intercambiar un byte de información durante una secuencia de 8 ciclos de reloj. Desde que el SCK es generado por el dispositivo maestro, esta línea llega a ser una entrada en un dispositivo esclavo.

Los dispositivos maestro y esclavo deben operar con la misma sincronización. El dispositivo maestro siempre ubica datos en la línea MOSI medio ciclo antes que el reloj (SCK), para que el dispositivo esclavo retenga el dato.

Dos bits (SPRO y SPR1) en el SPCR del dispositivo maestro seleccionan el rango de reloj. En el dispositivo esclavo, SPRO y SPR1 no tiene efecto en la operación del SPI.

1.7.2.4 SELECCIÓN DE ESCLAVO (\overline{SS})

La línea de entrada de selección de esclavo (\overline{SS}) es usada para seleccionar un dispositivo esclavo. Esta tiene que ser de baja prioridad en la transferencia de datos y debe permanecer bajo mientras dura la transferencia.

La línea \overline{SS} en el maestro debe ser alta. Si ésta va a bajo, una bandera de error de modo por falla (MODF) es fijado en el registro de estado de periférico serial (SPSR).

1.7.3 DESCRIPCION FUNCIONAL

La figura 1.6 muestra un diagrama de bloques de la circuitería del Interfaz de periférico serial. Cuando un dispositivo maestro transmite datos al dispositivo esclavo vía la línea MOSI, el dispositivo esclavo responde enviando datos al dispositivo maestro vía la línea MISO del maestro. Esto implica una transmisión full duplex con datos de salida y entrada sincronizado con la misma señal de reloj. Un bit de estado simple (SPIF) es usado para indicar que las operaciones de I/O han sido completados.

Si una escritura es ejecutada durante la transferencia de datos, la transferencia ocurre ininterrumpidamente, y la escritura será inexistente. Esta condición causará que el bit de estado de

colisión de escritura (WCOL) en el SPSR sea seteado. Después que un byte de datos es desplazado, la bandera SPIF del SPSR es seteado.

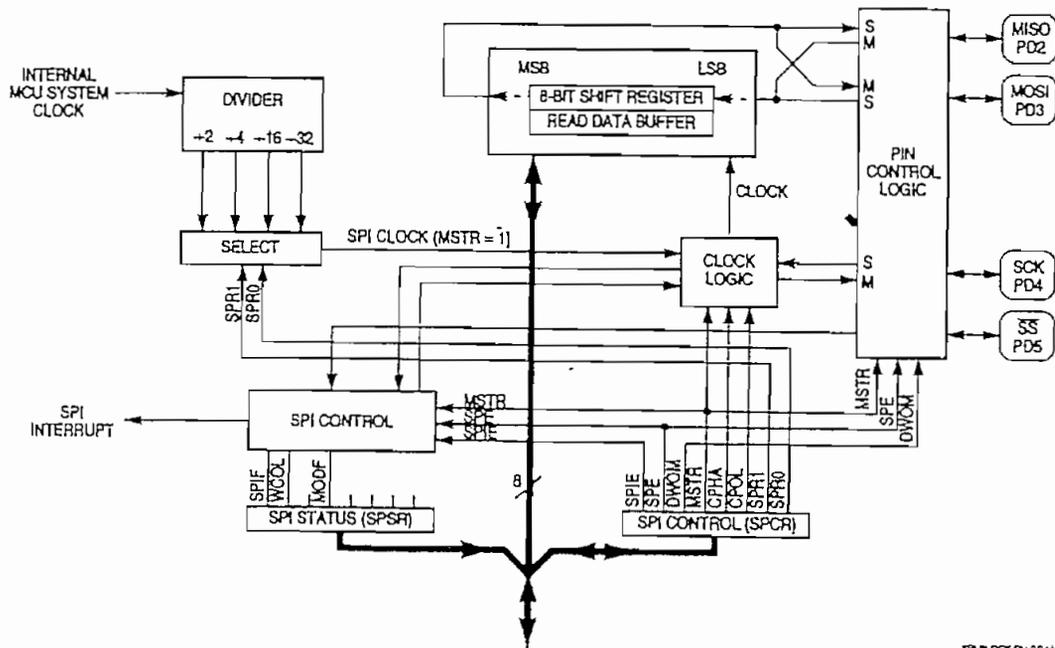


FIGURA No 1.6

DIAGRAMA DE BLOQUES DEL INTERFAZ DE PERIFERICOS SERIAL

En el modo maestro, el pin SCK es una salida. Esta línea desocupada estará en alto o bajo (dependiendo del bit CPOL en el registro SPCR), hasta que los datos sean escritos en el registro de desplazamiento, luego ocho ciclos de reloj son generados para desplazar los ocho bits de datos, y la línea SCK vuelve a estar desocupada.

En el modo esclavo, el esclavo inicia la recepción lógica con una señal baja en el pin SS y una entrada de reloj en el pin SCK, así el esclavo es sincronizado con el maestro. Los datos del maestro son recibidos serialmente por el esclavo en la línea MOSI y cargado en el registro de desplazamiento de 8 bits. Después que el registro de desplazamiento es cargado son transferidos paralelamente al buffer de lectura. Durante un ciclo de escritura, los datos son escritos en el registro de desplazamiento, luego el esclavo espera por un tren de reloj desde el maestro al registro de datos de salida en la línea MISO del esclavo.

La figura 1.7 ilustra la interconexión maestro - esclavo para las líneas MOSI, MISO ,SCK

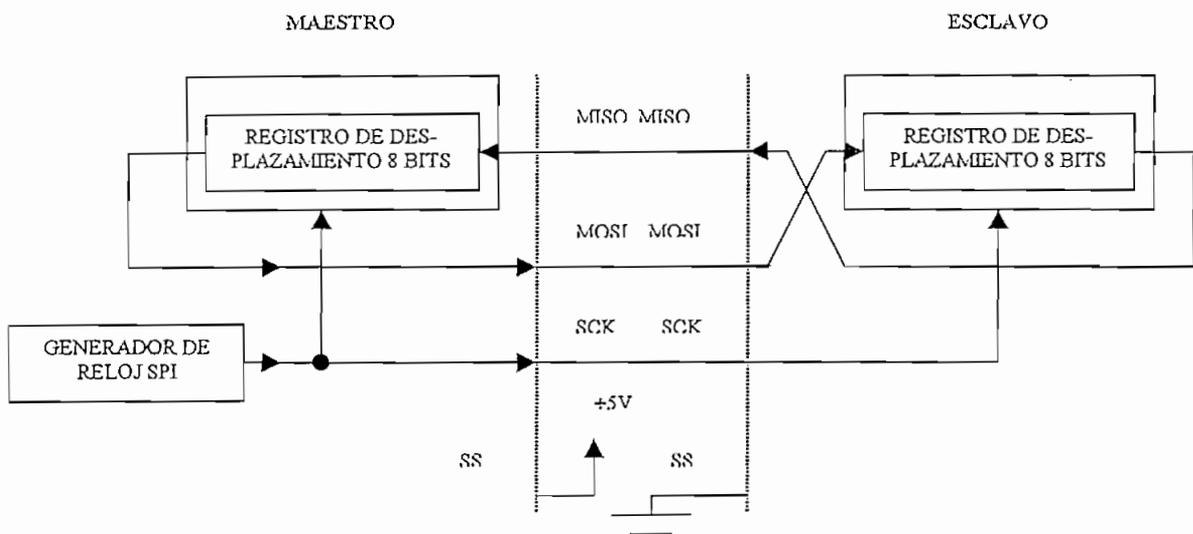


FIGURA. No 1.7

INTERCONEXION MAESTRO - ESCLAVO DEL INTERFAZ SERIAL SINCRONICO

1.7.4 REGISTROS DEL SPI

Hay 3 registros en el interfaz serial de periféricos, los cuales proveen control, estados, y funciones de registrar datos. Estos registros son llamados registros de control de periférico serial (SPCR), registro de I/O de datos de periférico serial (SPDR) y son descritos a continuación.

1.7.4.1 REGISTRO DE CONTROL DE PERIFERICO SERIAL

	7	6	5	4	3	2	1	0	
\$1028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
RESET	0	0	0	0	0	1	U	U	

SPIE: Habilitación de periférico serial

0= deshabilitado las interrupciones SPIF

1= Se genera la interrupción SPI si SPIF=1

SPE: Habilitación del sistema de periféricos serial

0= Apagado sistema SPI

1= Encendido del sistema SPI

DWON: Opción de modo WIRE - OR pòrtico D

DWON Afectan todos los 6 pines a la vez

0= Pòrtico D son salidas CMOS normales

1= Pòrtico D con salidas a drenaje abierto

MSTR: Selección de modo maestro

0= Modo Esclavo

1= Modo Maestro

CPOL: Polaridad de Reloj

Cuando el bit de polaridad de reloj es borrado y los datos no han sido transferidos, un estado constante bajo es producido en el pin SCK del dispositivo maestro. Contrariamente, si este bit es seteado, el pin SCK estará en alto. Este bit es también usado en conjunto con el bit de control de fase de reloj, para producir el reloj de datos deseado entre el maestro esclavo.

CPHA: Fase de Reloj

El bit de fase de reloj, en conjunto con el bit CPOL controla la relación reloj - datos entre el maestro y el esclavo. EL bit CPHA selecciona una de las 2 diferencias fundamentales de protocolos de reloj. Cuando CPHA=0 el reloj de desplazamiento es igual a la función lógica OR de la señal SCK con la señal \overline{SS} . Tan pronto como \overline{SS} llega a bajo la transacción comienza y el primer cambio sobre SCK invoca la primera muestra de datos. Cuando CPHA=1 el pin SS puede pensarse como una simple habilitación de salida.

SPR1	SPR0	DIVISION PARA EL PROCESADOR INTERNO
0	0	2
0	1	4
1	0	16
1	1	32

TABLA No 1.6

SELECCION DEL RANGO DE VELOCIDAD PARA EL SPI

SPR1 y SPR0 Selecciona el rango de reloj del SPI.

Estos 2 bits del SPI seleccionan uno de los 4 rangos de velocidad (Tabla 1.6) que serán usados como reloj si el dispositivo es maestro, sin embargo no tienen efecto en el modo esclavo.

1.7.4.2 REGISTRO DE ESTADO DE PERIFERICO SERIAL (SPSR)

	7	6	5	4	3	2	1	0	
\$1,029	SPIF	WCOL	0	MODF	0	0	0	0	SPSR
RESET	0	0	0	0	0	0	0	0	

SPIF.- Bandera de transferencia completa del SPI.

El bit de bandera de transferencia de datos de periférico serial es seteado cuando se completa una transferencia de datos entre el microcontrolador y el dispositivo externo. Si el SPIF va a alto, y el SPIE es seteado, la interrupción de periférico serial es generada. El borrado del bit SPIF es realizado por la lectura del SPSR seguido por un acceso del SPDR.

WCOL.- Colisión de Escritura.

El bit de colisión de escritura es seteado cuando un intento es hecho para escribir al registro de datos de periféricos serial mientras la transferencia de datos toma lugar. Si CPHA es cero una transferencia esta siendo iniciada cuando SS va a bajo y la transferencia finaliza cuando SS va a alto después de 8 ciclos de reloj SCK. Cuando CPHA=1 una transferencia comienza al primer ciclo SCK llegando a ser activo mientras SS está bajo y la transferencia finaliza cuando la bandera SPIF es seteada. La limpieza del bit WCOL se da al realizar una lectura del SPSR seguido por un acceso al SPDR.

Bit 5 No Implementado

Cuando leemos este bit siempre se obtiene cero.

MODF: Modo de Falla

La bandera de modo de falla indica que puede haber conflicto por multi- maestro para controlar el sistema, y permite una salida de la operación del sistema por un reset o estado del sistema por fallas. El bit MODF esta normalmente borrado, y es seteado solo cuando el

dispositivo maestro tiene al pin SS en bajo, seteando el bit MODF se afecta al sistema de Interfaz de periférico serial de la siguiente forma:

- 1.- Una interrupción es generada si SPIE=1
- 2.- El bit SPE es limpiado. Esto deshabilita el SPI
- 3.- El bit MSTR es limpiado, así forzamos al dispositivo a ponerse en el modo esclavo
- 4.- Los bits DDRD para los 4 pines SPI son forzados a cero.

Para borrar el bit MODF se debe leer el registro SPSR, seguido por una escritura al SPSR. Los bits de control SPE y MSTR pueden ser restaurados por el usuario por software a su original estado después que el bit MODF es limpiado. Es también necesario restaurar el DDRD después de un modo de falla.

Bits 3..0 No Implementados

Lecturas de estos bits siempre serán cero

1.7.4.3 REGISTRO I/O DE DATOS DE PERIFERICO SERIAL

El registro I/O de datos de periférico serial es usado para transmitir y recibir datos en el bus serial. Solo una escritura a este registro iniciará la transmisión/recepción de otro byte, y esto solo ocurrirá en el dispositivo maestro. Al completar la transmisión de un byte, el bit de estado SPIF es seteado en el maestro y en el esclavo.

Cuando el usuario lee del registro I/O de datos de periféricos serial, un buffer es realmente leído. El primer SPIF debe ser limpiado a la vez que una segunda transferencia de datos desde el registro de desplazamiento al buffer de lectura es iniciada.

Una escritura al registro I/O de datos de periférico serial no pasa el dato al buffer y ubica directamente el dato en el registro de desplazamiento para transmisión.

1.8 INTERFAZ DE COMUNICACIÓN SERIAL ASINCRÓNICA (SCI)

1.8.1 Características.

El SCI es un Interfaz de comunicación serial asincrónica full- dúplex con un formato standard de no retorno a cero NRZ (un bit de inicio, 8 o 9 bits de datos y un bit de parada) y una variedad de velocidades (baud rate).

El transmisor y receptor SCI son funcionalmente independientes, pero usan los mismos formatos de datos y velocidades.

CARACTERÍSTICAS DEL SISTEMA SCI A DOS HILOS.

- . Formato standard NRZ (no retorno a cero).
- . Método de detección de errores el cual incluye detección de ruido de duración sobre 1/16 de tiempo del bit.
- . Operación full dúplex
- . Software programable para 32 diferentes velocidades
- . Software seleccionable para la longitud de la palabra (8 o 9 bits por palabra)
- . Bits de habilitación para Transmisión y Recepción
- . Cuatro bits de habilitación separados disponibles para control de interrupciones

CARACTERÍSTICAS DEL RECEPTOR SCI

- . Función de inicio de recepción (por línea en reposo o por bit de dirección)
- . Detección de línea en reposo
- . Detección de errores
- . Detección de ruido
- . Detección de sobrecorrido

. Bandera de llenado de registro de datos del receptor.

CARACTERISTICAS DEL TRANSMISOR SCI

. Bandera de registro de datos de transmisión vacío

. Bandera de transmisión completa.

. Ruptura de envío.

1.8.2 FORMATO DE DATOS

Los datos transmitidos o recibidos serialmente son transferidos al bus de datos interno desde el pin de entrada de recepción (RxD), o desde el bus interno al pin de salida de transmisión de datos (TxD).

El formato de datos “no retorno a cero” (NRZ) es mostrado en la figura No 1.8 y se basa en los siguientes criterios.

- 1.- La línea en reposo es llevada a un estado de prioridad uno lógico para transmisión o recepción de un carácter
- 2.- Un bit de inicio (cero lógico) es usado para indicar el inicio de una estructura.
- 3.- El primer bit de datos transmitido o recibido es el menos significativo.
- 4.- Un bit de parada (uno lógico) es usado para indicar el final de la estructura. Una estructura consiste de un bit de inicio, un carácter de 8 o 9 bits y un bit de parada .
- 5.- Una ruptura es definida como la transmisión o recepción de al menos un tiempo en bajo igual al de una estructura completa.

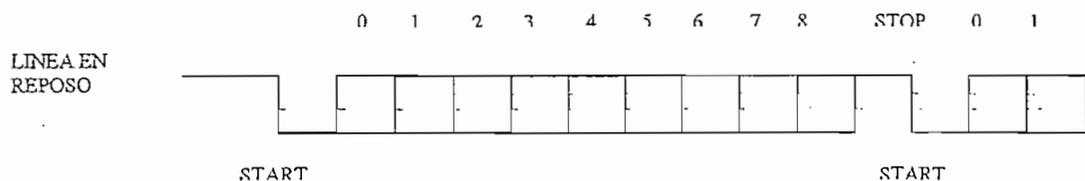


FIGURA No 1.8

FORMATO DE DATOS

1.8.3 CARACTERISTICAS DEL INICIO DE RECEPCION.

El Software en cada recepción evalúa el primer carácter de cada mensaje.

Un receptor SCI en espera puede ser configurado usando el bit de control WAKE en el registro de control de comunicación serial (SCCR1) usando ambos métodos: despertar por línea en reposo o despertar por marca de dirección.

Al despertar por línea de reposo, un receptor dormido despierta tan pronto como la línea de Rx llega al estado de reposo. El estado de reposo está definido como una lógica continua alta en la línea de Rx para 10 u 11 tiempos de bits. Los sistemas que usan este tipo de despertar deben proveer al menos un tiempo de reposo igual al de un carácter entre mensajes para despertar un receptor dormido; pero no deben permitir un tiempo igual o mayor en reposo entre caracteres dentro de un mensaje.

Al despertar por marca de direcciones, el bit más significativo (MSB) en un carácter está usado para indicar que el carácter es una dirección (1) o un dato (0). El receptor dormido despertará cuando un carácter de dirección es recibido. Los sistemas que usan este método para despertar pondrán en uno lógico el MSB del primer carácter en cada mensaje y dejarán limpio el

MSB para los otros caracteres del mensaje. Los períodos de reposo pueden estar presentes dentro de un mensaje para este método de despertar.

1.8.4 DATOS RECIBIDOS (R x D)

Datos recibidos son los datos seriales los cuales son aplicados a través de la línea de entrada RxD al bus interno. La circuitería de reloj de recepción de entrada es de un rango igual a 16 veces el rango de velocidad y este tiempo es referido como el reloj RT.

Una vez que un bit de inicio válido es detectado, el bit de inicio, cada bit de datos, y el bit de parada son muestreados 3 veces en intervalos iguales a RT en las posiciones 8 RT, 9 RT y 10 RT (1RT es la posición donde se espera que el bit inicie), como muestra la figura 1.9. Los valores son determinados por lógica de votos, en la cual se toma el valor de la mayoría de muestras.

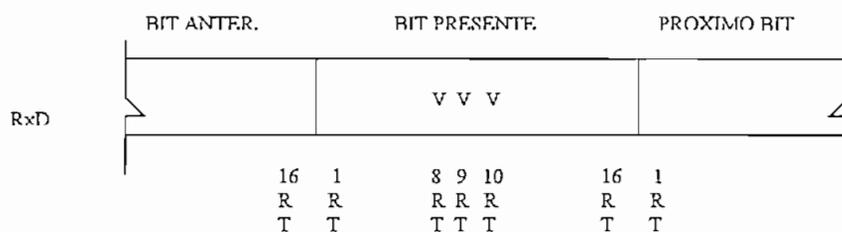


FIGURA 1.9

TECNICA DE MUESTREO ASOCIADO EN TODOS LOS BITS

1.8.4.1 DETECCION DEL BIT DE INICIO

Cuando en la entrada RxD se detecta un estado bajo, se verifica en 3 muestras mas la permanencia del estado. Si al menos 2 de esas 3 verificaciones de muestras detecta un cero lógico, un bit de inicio ha sido detectado, de otro modo la línea será asumida en reposo.

Una bandera de ruido es seteada si las 3 muestras de verificación no están en cero lógico.

1.8.5 DATOS TRANSMITIDOS (TxD)

Datos transmitidos son los datos que estando en el bus de datos interno, van a ser aplicados a través del interfaz de comunicación serial a la línea de salida TxD. El transmisor genera una señal de reloj igual a $1/16$ del reloj de muestreo de recepción RT.

1.8.6 DESCRIPCION FUNCIONAL DEL SCI.

Un diagrama de bloques del SCI es mostrado en la figura No.1.10. El usuario tiene bits de opción en el registro de control del interface serial uno (SCCR1) para determinar el método de despertar (bit WAKE) y la longitud de palabra (M bit) del SCI, El registro de control de comunicación serial dos (SCCR2) provee bits de control los cuales habilitan individualmente el transmisor y receptor (TE y RE respectivamente) habilitan al sistema de interrupción (TIE, TCIE, ILIE) y provee la habilitación del bit de despertar (RWU) y el envío de ruptura con el código de bit (SBK). El registro de elección de velocidad (BAUD) permite al usuario seleccionar las diferentes velocidades, las cuales pueden ser usadas para el control de velocidad de transmisión y recepción.

La transmisión de datos es iniciada por una escritura en el registro de datos de comunicación serial (SCDR). La habilitación a la transmisión es ejecutada, los datos registrados en el SCDR son transferidos al registro de desplazamiento de transmisión de datos. Esta transferencia de datos fija el bit TDRE del registro de estado (SCSR) del SCI y puede generar una interrupción si la interrupción de transmisión es habilitada. La transferencia de datos al registro de desplazamiento de transmisión es sincronizado con la señal de reloj mostrada en la

figura No 1.11. Todos los datos son transmitidos iniciando con el LSB. Al finalizar la transmisión de un dato, el bit de transmisión completa (TC) del SCSR es seteado, y una interrupción puede ser generada si la interrupción de transmisión completa está habilitada. Si el transmisor es deshabilitado y el dato o una señal de ruptura ha sido enviado, el bit TC también será seteado. Esto también generará una interrupción si el bit TCIE es seteado. Si el transmisor es deshabilitado en la mitad de la transmisión, ese carácter será completado antes que el transmisor pierda el control del pin de T x D.

Cuando el SCDR es leído, este contiene el último byte recibido, con tal que el receptor esté habilitado. El bit RDRF del SCSR es seteado para indicar que un byte de dato ha sido transferido desde el registro de desplazamiento serial de entrada al SCDR, lo cual puede causar una interrupción, si la interrupción de recepción está habilitada. La transferencia de datos desde el registro de desplazamiento serial de entrada al SCDR es sincronizada por el de reloj de recepción. Los bits OR (over - run), NF (ruido) o bit de error OF(estructura) del SCSR pueden ser seteados si ocurren errores de recepción de datos.

Una interrupción de la línea de reposo es generada si la interrupción de la línea de reposo es habilitada y el bit IDLE (el cual detecta la transmisión en la línea en reposo) del SCSR es seteado.

1.8.7 REGISTROS DEL SCI

Hay 5 registros usados en el Interfaz de comunicación serial y las referencias serán hechas al diagrama de bloques mostrado en la figura No.1.10

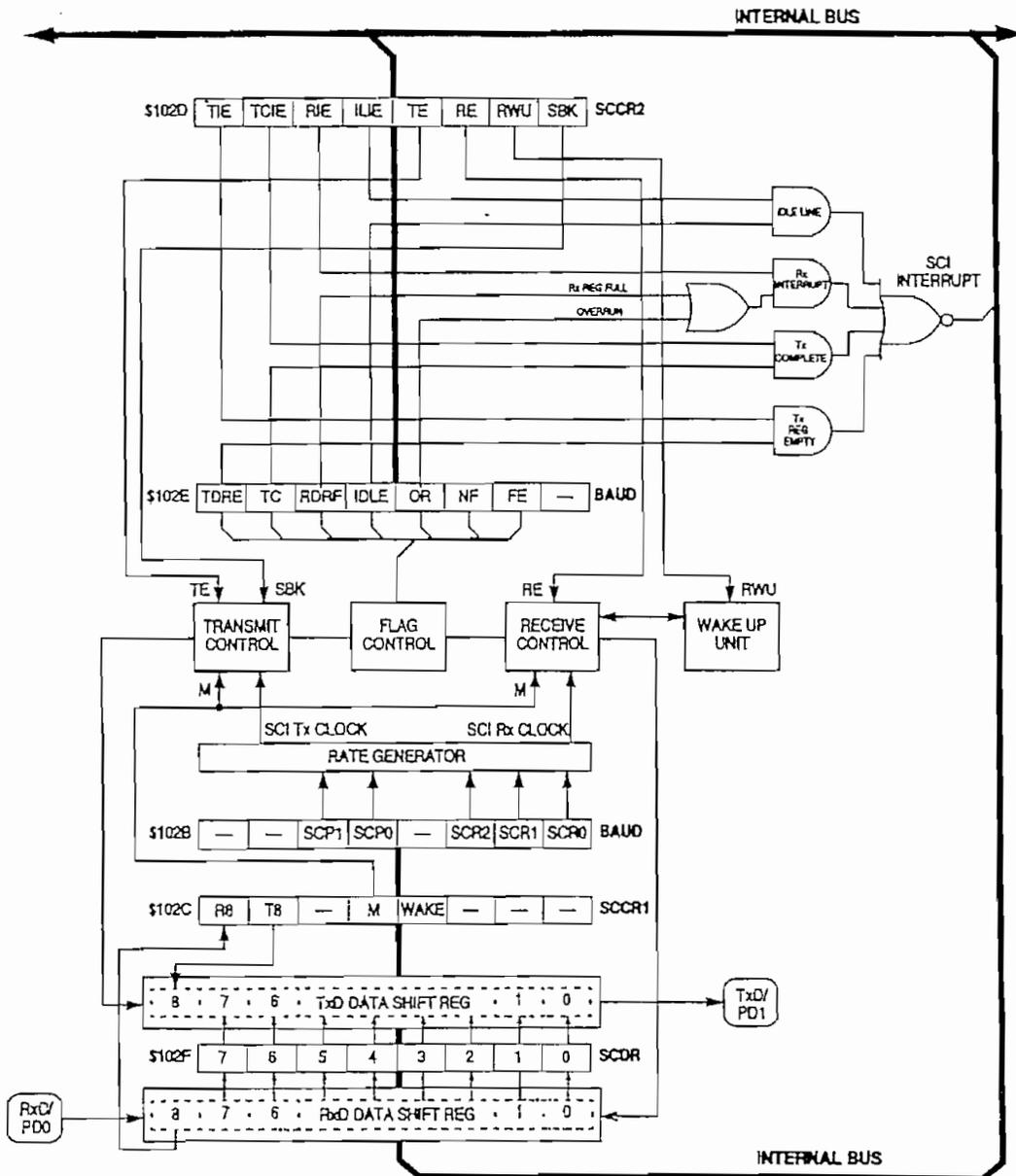


FIGURA No.1.10

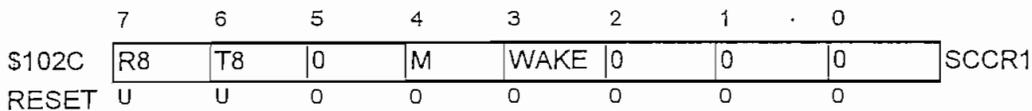
DIAGRAMA DE BLOQUES DEL INTERFAZ DE COMUNICACIÓN SERIAL.

1.8.7.1 REGISTRO DE DATOS DE COMUNICACIÓN SERIAL (SCDR)

El registro de datos de comunicación serial ejecuta 2 funciones; este actúa como el registro de datos de recepción cuando este es leído y como el registro de datos de transmisión cuando es escrito.

1.8.7.2 REGISTRO DE CONTROL DE COMUNICACIÓN SERIAL 1 (SCCR1)

El registro de control de comunicación serial 1 (SCCR1) contiene los bits de control los cuales: 1) Determinan la longitud de la palabra y 2) Seleccionan el método usado para las características de despertar o inicio de recepción.



R8: Dato recibido - 8 bits

Si el bit M está en uno lógico, el bit R8 provee una localidad de almacenamiento para el noveno bit en el caracter de datos recibidos.

T8: Datos transmitido - 8 bits.

Si el bit M está en uno lógico, este bit provee una localidad de almacenamiento para el noveno bit en el caracter de datos transmitidos

Bit 5: No implementado

Lectura a este bit nos da cero.

M: Longitud del caracter SCI.

0 = 1 bit de inicio, 8 bits de datos, 1 bit de parada.

1 = 1 bit de inicio, 9 bits de datos, 1 bit de parada.

WAKE: Selección el método de inicio de recepción.

0 = Línea en reposo

1 = Address Mark

Bits 2-0: No implementados.(lecturas a estos bits serán siempre cero)

1.8.7.3 REGISTRO DE CONTROL DE COMUNICACIÓN SERIAL 2 (SCCR2)

El registro de control de comunicación serial 2 (SCCR2) contiene los bits de control los cuales habilitan o deshabilitan las funciones del SCI individuales.

	7	6	5	4	3	2	1	0	
\$102d	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCCR2
RESET	0	0	0	0	0	0	0	0	

TIE – Habilitación de interrupción de transmisión

0 = Deshabilitada la interrupción TDRE.

1 = Se genera la Interrupción SCI sí TDRE = 1

TCIE – Habilitación de interrupción de transmisión completa.

0 = Deshabilitado la interrupción TC.

1 = Se genera la interrupción SCI sí TC = 1

RIE- Habilitación de Interrupción de recepción

0= Las interrupciones RDRF y OR deshabilitadas

1= Se genera la Interrupción SCI sí RDRF u OR= 1

ILIE- Habilitación de interrupción de la línea de reposo

0= Deshabilitación de interrupción IDLE

1= Se genera la Interrupción SCI sí IDLE = 1

TE- Habilitación de transmisión

Cuando el bit de habilitación de transmisión es seteado, el registro de desplazamiento de salida es aplicado a la línea de T x D. Dependiendo del estado del bit de control M (SCCR1), un preámbulo de 10 (M=0) u 11 (M= 1) unos consecutivos son transmitidos cuando el software setea el bit TE desde un estado de borrado. Después de cargar el último byte en el registro de datos de comunicación serial y recibiendo la bandera TDRE, el usuario puede borrar TE. La transmisión del último byte será completada antes que el transmisor ceda el control del pin T x D. Mientras el transmisor está activo, el control del registro de dirección de datos para el bit 1 del pórtilo D es sobrepuesto y la línea es forzada a ser una salida.

RE- Habilidad de recepción

Cuando el bit de habilitación de recepción es seteado, la recepción está habilitado. Cuando RE es borrado, la recepción es deshabilitada y todos los bits de estado asociado con la recepción (RDRF, IDLE, OR, NF, FE) son inhibidos. Mientras la recepción es habilitada, el control del registro de dirección de datos para el bit 0 del pórtilo D es sobrepuesto y la línea es forzada a ser una entrada.

RWU- Despertar del Receptor

Cuando el bit de inicio de recepción es seteado por el software de usuario, esto pone al receptor en reposo y habilita la función de despertar “wake - up”.

SBK- Envío de ruptura (Break).

Si el bit envío de break es consecutivamente seteado y limpiado, el transmisor envía 10(M=0) u 11(M=1) ceros y luego revierte el reposo o envía el dato.

En la terminación del código de break, el transmisor envía al menos un bit alto para garantizar el reconocimiento de un bit de inicio válido.

1.8.7.4 REGISTRO DE ESTADO DE COMUNICACIÓN SERIAL (SCSR).

El registro de estado de comunicación serial (SCSR) provee entradas al circuito lógico de interrupción para la generación de interrupción SCI.

	7	6	5	4	3	2	1	0	
\$102E	TDRE	TC	RDRF	IDLE	OR	NF	FE	0	SCSR
RESET	1	1	0	0	0	0	0	0	

TDRE - Registro de datos de transmisión vacío

El bit del registro de transmisión de datos es seteado para indicar que el contenido del registro de datos de comunicación serial ha sido transferido al registro de desplazamiento serial. Este bit es limpiado por lectura del SCSR seguido por una escritura al SCDR

TC: transmisión completa

El bit de transmisión completa es seteado al final de un dato, preámbulo, o condición de break sí:

- 1) TE = 1, TDRE=1 y no hay datos pendientes, preámbulo, o ruptura a ser transmitidos.
- 2) TE = 0, y el dato, preámbulo o break en el registro de desplazamiento de transmisión ha sido transmitido.

El bit TC es borrado por lectura del SCSR (con TC fijado) seguido por una escritura al SCDR

RDRF: REGISTRO DE DATOS DE RECEPCION LLENO

Es seteado cuando el registro serial de desplazamiento es transferido al SCDR

El bit RDRF es limpiado cuando el SCSR es leído (con el RDRF fijado) seguido por una lectura del SCDR.

IDLE. DETECCIÓN DE LÍNEA EN REPOSO

El bit de detección de línea en reposo, cuando es seteado indica que el receptor ha detectado una línea en reposo. El bit IDLE es limpiado por lectura al SCSR seguido por una lectura del SCDR.

OR . ERROR OVERRUN.

El bit de error over-run es seteado cuando el próximo byte está listo a ser transferido del registro de desplazamiento al SCDR el cual está ya lleno ($RDRF = 1$). Cuando ocurre un error de over-run, el dato el cual causa el over-run se pierde y el dato que estuvo ya listo en el SCDR no es afectado. El bit OR es borrado cuando el SCSR es leído, seguido por una lectura del SCDR.

NF- BANDERA DE RUIDO

EL bit de bandera de ruido es seteado si hay ruido en cualquiera de los bits recibidos, incluyendo los bits de inicio y parada. El bit NF no es seteado hasta que la bandera RDRF es seteada. El bit NF es limpiado cuando el SCSR es leído (con el NF fijado) seguido por una lectura al SCDR.

FE ERROR DE ESTRUCTURA

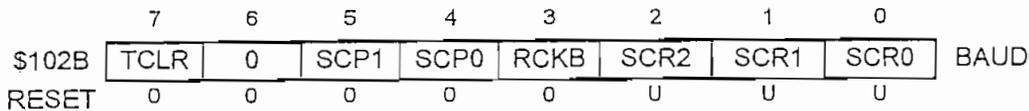
El bit de error de estructura es seteado cuando el bit de parada no es detectado en el caracter de datos recibido. El bit FE es seteado al mismo tiempo que el RDRF es seteado. Si el byte recibido causa ambos errores de estructura y over-run, el procesador solo reconocerá el error de over-run. La bandera de error de estructura inhibe la transferencia de datos en el SCDR hasta que éste sea borrado. El bit FE es borrado cuando el SCSR es leído seguido por una lectura del SCDR.

BIT 0- no Implementado

Una lectura de este bit siempre será cero.

1.8.7.5 REGISTRO DE VELOCIDAD (BAUD)

El registro de velocidad selecciona las diferentes velocidades las cuales son usadas como velocidades de control para el transmisor y el receptor. Los bits SCP0-SCP1 funcionan como preescalares para los bits SCR0, SCR1 y SCR2, juntos esos 5 bits proveen múltiples combinaciones para dar diferentes velocidades a una frecuencia de cristal dado.



TCLR . limpieza del contador de velocidad (válido solo en el modo de prueba)

TCLR es cero y no puede ser usado mientras está en el modo normal de operación.

SCP1 y SCP0. Selecciona el valor preescalar para determinar la velocidad del SCI.

El reloj E es dividido por los factores mostrados en la tabla No 1.7 Esta salida preescalar provee una entrada a un divisor el cual es controlado por los bits SCR2- SCR0.

SCP1	SCP0	PROCESADOR INTERNO DIVISION DEL RELOJ POR
0	0	1
0	1	3
1	0	4
1	1	13

TABLA No 1.7

PRIMER PREESCALADO

SCR2, SCR1 Y SCRO- Utilizados para seleccionar la velocidad. Estos 3 bits seleccionan la velocidad para la transmisión y recepción, La salida preescalar descrita arriba es dividida por los factores mostrados en la tabla No 1.8

SCR2	SCR1	SCR0	SALIDA PREESCALAR DIVISION POR
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

TABLA No 1.8

PREESCALADO FINAL

RCKB- Chequeo de reloj de rango de velocidad (válido solo en el modo de prueba)

Este bit es usado durante las pruebas de fabricación.

El diagrama mostrado en la figura No1.10 ilustra la cadena de división para obtener el rango de velocidad (baud rate). Note que hay una división para 16 entre el reloj de recepción (RT) y el reloj de transmisión:

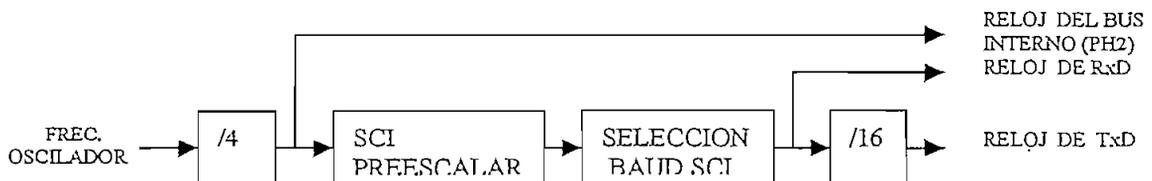


FIGURA No 1.11

DIVISIÓN DEL RANGO DE GENERACIÓN

1.9 CONVERTOR ANALOGO DIGITAL

El MC68HC11A8 incluye 8 canales de entrada multiplexados, de conversión análogo - digital (A/D), con muestreo y retención para minimizar los errores de conversión causados por cambios rápidos en la señal de entrada. Dos líneas dedicadas (VRL, VRH) son dadas para las referencias de voltaje de entrada. Estos pines pueden ser conectados a una fuente separada o aislada para asegurar la precisión de la conversión A/D.

El convertidor A/D de 8 bits tiene un error total de ± 1 del bit menos significativo y acepta entradas analógicas las cuales están en el rango de VRL a VRH. La conversión es especificada y probada para VRL=0 y VRH=5V \pm 10%: sin embargo, la caracterización de laboratorio sobre el rango máximo de temperatura indica poca o ninguna degradación si VRH - VRL es tan bajo como 2.5 o máximo 3 V. El sistema A/D puede ser operado con VRH abajo de VDD y o VRL sobre VSS. Cada conversión es acompañada de 32 ciclos de reloj E del MCU, siendo la frecuencia de reloj E mayor que 750 KHz, para sistemas los cuales operan a un rango de reloj menor que 750 KHz, un oscilador interno RC debe ser usado para el reloj del sistema A/D. El oscilador interno R-C es seleccionado por seteo del bit CSEL en el registro OPTION.

1.9.1 PROCESO DE CONVERSION

En el sistema de conversión análogo digital, un voltaje de entrada igual a VRL se convierte en \$00 y un voltaje de entrada igual a VRH se convierte en \$FF, sin indicación de desbordamiento.

1.9.2 ASIGNACION DE CANALES

Un multiplexor permite al simple convertidor A/D seleccionar una de las 16 señales analógicas. 8 de éstas corresponden a las líneas de entrada del pòrtico E del MCU, 4 canales son

modo que una comparación OC1 que tenga éxito no afecte ninguno de estos pines de I/O. Las otras comparaciones de salida son configuradas para no afectar los pines I/O en comparaciones que tengan éxito. Todos los 3 circuitos de captura de entrada son deshabilitados. La bandera de interrupción de overflow del timer y todas las 8 banderas de funciones de interrupción son limpiadas. Las 9 interrupciones de timer son deshabilitadas desde que sus bits de máscara son borrados.

1.12.1.2.5 INTERRUPCION EN TIEMPO REAL.- La bandera de interrupción en tiempo real es borrado y automáticamente las interrupciones de hardware son enmascaradas. Los bits de control de velocidad de comunicación serial son borrados después del reset y pueden ser inicializados por software antes que el sistema de interrupción en tiempo real sea usado.

1.12.1.2.6 ACUMULADOR DE PULSOS .- El sistema de acumulador de pulsos es deshabilitado en el reset de modo que el pin de entrada PA1 será un pin de entrada de propósito general.

1.12.1.2.7 COP. EL sistema de vigilancia (watchdog) es habilitado si el bit de control NOCOP en el registro de control de configuración del sistema (CONFIG) es borrado y deshabilitado si NOCOP es seteado. El reloj para el COP es seteado para el tiempo de duración mas pequeño.

1.12.1.2.7 I/O SERIAL SCI. La condición de reset del sistema SCI es independiente del modo de operación. En el reset el rango de velocidad SCI es indeterminado y debe ser establecido por una escritura de software al registro BAUD. Todas las interrupciones de Tx y Rx son enmascaradas, el transmisor y el receptor (Tx y Rx) son deshabilitados de modo que los pines del pòrtico A son líneas de I/O de propósito general. EL formato de estructura SCI es inicializado en formato de 8 bits . Las funciones de break y wake up (inicio de recepción) son deshabilitadas. Los bits de estado TDRE y TC en el registro de estado del SCI son seteados indicando que no hay datos de

transmisión pendientes ni en el registro de datos de transmisión ni en el registro de desplazamiento de transmisión serial. Los bits de estado RDRF, IDLE , OR, NF, y FE son todos borrados.

1.12.1.2.9 I/O SERIAL SPI.- El sistema SPI es deshabilitado por el reset, los pines del pòrtico asociados con estas funciones serán líneas de I/O de propósito general.

1.12.1.2.10 CONVERTIDOR A/D. El sistema de conversión A/D es indeterminado después del reset, la bandera de conversión completa es limpiada, el bit ADPU es borrado, así el sistema A/D es deshabilitado.

1.12.1.2.11 SISTEMA.- Los controles de programación de la memoria EEPROM son todos deshabilitados de modo que el sistema de memoria es configurado para operaciones normales de lectura. La interrupción de más alta prioridad por defaults será al pin externo IRQ. Los bits RBOOT, SMOD y MDA en el registro HPRI0 reflejan el estado de las entradas MODB y MODA en el instante de subida del reset. El sistema monitor de reloj es deshabilitado pues CME=0

1.12.1.3.- RESET POR OPERACIÓN IMPROPIA.

El MCU incluye un sistema de vigilancia (WATCHDOG) para protección contra fallas de software. Para usar un sistema de vigilancia de computación apropiada de operación (COP), una secuencia de reset de temporización será ejecutada en un periodo regular base de modo que el temporizador de vigilancia nunca quede fuera de tiempo.

La función interna COP, incluye bits de control especial lo cual permite la especificación de uno de los 4 periodos de tiempo fuera e incluso permite ser deshabilitado completamente. El sistema COP tiene un vector de reset separado.

El bit de control NOCOP, el cual determina si un tiempo fuera, generado por el sistema de vigilancia causa o no un reset, es implementado en una celda EEPROM en el registro CONFIG. Una vez programado este bit se mantiene seteado (o borrado) aun cuando no hay energía y las funciones COP son habilitadas y deshabilitadas independientes del software residente.

Los otros bits de control en el registro OPTION seleccionan uno de los 4 periodos de tiempo fuera para el timer COP.

La condición de reset de los bits CR1 y CR0 son borrados, lo cual corresponde al periodo de tiempo fuera mas corto.

La secuencia requerida para resetear el temporizador WATCHDOG es

- 1.- Escribir \$55 al registro reset de COP (COPRST) en \$103A, seguida por
- 2.- Escribir \$AA en la misma dirección.

1.12.1.4.- RESET POR MONITOR DE RELOJ

La función del monitor de reloj es habilitada por el bit de control CME en el registro OPTION. Cuando CME es borrado las funciones del monitor de reloj son deshabilitados, y cuando el bit CME es seteado, la función del monitor de reloj detecta la ausencia de un reloj E para más de un cierto periodo de tiempo. El periodo de tiempo fuera será entre 5 y 100 microsegundos. Esto significa que un reloj de 200KHz o más, nunca causará fallas de monitoreo de reloj y un reloj de 10KHz o menos definitivamente causa fallas de monitoreo de reloj. Esto implica que sistemas operando cerca o sobre un rango de 200KHz no usará las funciones de monitoreo de reloj.

Si se da una detección de un reloj bajo o ausencia de reloj, el circuito monitor de reloj causará un reset del sistema. Este reset es dado a conocer a un sistema externo vía al pin bidireccional RESET.

Consideraciones especiales son necesarias cuando usamos una función STOP y el sistema monitor de reloj en un mismo programa. Ya que la función STOP causa que el reloj se pare, la función del monitor de reloj generará una secuencia de reset si este es habilitado en el tiempo que es ingresado en el modo STOP.

1.12.1.5 REGISTRO DE CONFIGURACION (OPTION)

Este es un registro especial de 8 bits de propósito especial que es usado durante la inicialización para configurar el sistema interno. Con la excepción de los bits 7,6 y 3 (ADPU, CSEL y CME) los cuales pueden ser escritos o leídos en cualquier tiempo, este registro puede ser escrito solo una vez después del reset y de ahí en adelante es un registro de solo lectura. Si una escritura no es ejecutada en esta localidad dentro de los 64 ciclos de reloj después del reset, los bits IRQE, DLY CR1 y CRO llegan a ser bits solo de lectura del sistema lo cual minimiza los posibles cambios accidentales a la configuración del sistema. Mientras está en el modo especial de test, el mecanismo de protección de este registro es fijado y todos los bits del registro OPTION pueden ser escritos repetidamente.

	7	6	5	4	3	2	1	0	
\$1,039	ADPU	CSEL	IRQE	DLY	CME	0	CR1	CR0	OPTION
RESET	0	0	0	1	0	0	0	0	

ADPU-A/D Energía del A/D

Este bit controla la operación del convertidor análogo digital. Cuando ADPU es borrado, el sistema A/D es apagado y las conversiones no ingresan información verdadera.. Para usar el

sistema A/D, este bit deberá setearse, se da un retardo de 100 ms después que el ADPU es encendido para permitir la estabilización del sistema.

CSEL- Selección de fuente de reloj

Este bit determina la fuente de reloj para el conversor A/D y el sistema de carga de la memoria EEPROM. Cuando este bit es cero, el reloj del MCU maneja al conversor A/D y el sistema de carga de la memoria EEPROM. Cuando CSEL=1 un oscilador separado RC es habilitado y el sistema de reloj es alrededor de 2MHz. Cuando se trabaja con un reloj bajo 1MHz, CSEL puede ser alto para programar o borrar la EEPROM. Cuando se trabaja bajo los 750 KHz, CSEL debe ser alto para conversiones A/D.

IRQE - IRQ Nivel de Detección

Cuando este bit es borrado el pin IRQ es configurado para operación de nivel sensible (nivel bajo), y cuando es seteado, el pin IRQ es configurado solo para detección de cambio de nivel (flanco negativo).

DLY- Retardo de Encendido STOP

Este bit es seteado durante el reset y controla si un retardo será o no necesario antes que el MCU se reanude después de un periodo de STOP. Cuando DLY es seteado, un ciclo de retardo de 4064 E es impuesto para permitir la estabilización del oscilador y cuando DLY es borrado éste retardo no se da.

CME Habilitación del Monitor de Reloj

Este bit puede ser escrito o leído en cualquier tiempo y controla si un circuito monitor de reloj interno dispara o no una secuencia de reset. Cuando CME es borrado, el circuito de

monitoreo de reloj es deshabilitado y cuando CME es seteado el circuito de monitoreo es habilitado. El reset limpia el bit CME.

BIT2 No implementado

Lecturas a este bit siempre nos dará cero

CR1 y CR0 Seleccionan la frecuencia de temporización para el sistema COP

1.12.2.- MODOS DE BAJO CONSUMO .

El MCU contiene 2 modos programables de bajo consumo WAIT y STOP

1.12.2.1.- INSTRUCCION WAIT

La instrucción WAIT pone al MCU en el modo de bajo consumo, manteniendo la ejecución del oscilador. Si se da la ejecución de una instrucción WAI, el estado del MCU es guardado en el stack, y se para la ejecución del programa. El estado de espera puede ser abandonado solo por una interrupción no enmascarable o por la acción de un RESET. Si el bit I es seteado (interrupción enmascarada) y el COP es deshabilitado, el sistema de temporización será apagado reduciendo el consumo de energía

1.12.2.2.- INSTRUCCIÓN STOP

La instrucción STOP ubica al MCU en el modo de más bajo consumo de energía, esto se da si el bit S en el registro de código de condiciones es limpiado. Si el bit S es seteado, el modo STOP es deshabilitado y la instrucción STOP es tratada como instrucciones de no operación NOP. En el modo de STOP todos los relojes incluyendo el oscilador interno son parados causando que todos los procesos sean detenidos. La recuperación del modo de STOP puede ser acompañada de un RESET, una interrupción XIRQ o una interrupción no enmascarable IRQ. Cuando XIRQ es usado, el MCU sale del modo STOP sin hacer caso del estado del bit X en el

CCR sin embargo, la secuencia de recuperación difiere dependiendo del estado del bit X. Si el bit X es limpiado el MCU inicia con la secuencia de apilamiento cargando el normal servicio del requerimiento de XIRQ. Si el bit X es seteado, el proceso continuará inmediatamente después de la instrucción STOP. Un reset siempre causará una salida desde el modo de STOP, y el inicio de operación del MCU es determinado por el vector de reset.

1.13 SET DE INSTRUCCIONES EN DETALLE.

La información completa y detallada del set de instrucciones para todos los microcontroladores de la familia M68HC11 de Motorola serán explicadas en el Manual de “Listado de Programas de las prácticas Implementadas y Set de Instrucciones del MC68HC11”, cada una acompañada de un cuadro de resumen de los cambios (si los hubiere) de los estados de los bits en el registro de código de condiciones CCR.

La nomenclatura presentada a continuación será usada en el Apéndice G

ACCA = Acumulador A.

ACCB = Acumulador B.

ACCX = Acumulador A ó B.

ACCD = Doble acumulador, acumulador A concatenado con acumulador B, donde A es el byte más significativo.

CCR = Registro de Código de condiciones.

IX = Registro índice X, 16 bits.

IY = Registro índice Y, 16 bits.

PC = Contador de programa, 16 bits.

SP = Puntero de Pila, 16 bits.

- opr = operando
- msk = mascara usada en instrucciones de manipulaci3n de bits.
- rel = offset relativo usado en instrucciones de salto
- \oplus = operaci3n l3gica OR - exclusivo.

La simbolog3a utilizada para el CCR es la siguiente:

- S = Modo de STOP deshabilitado, bit 7.
- X = M3scara de interrupci3n X, bit 6.
- H = Carry intermedio, bit 5.
- I = M3scara de interrupci3n I, bit 4.
- N = Indicador negativo, bit 3.
- Z = Indicador de cero, bit 2.
- V = Indicador de desbordamiento en complemento de dos, bit 1.
- C = Carry/Llevo, bit 0.
- = Bit no afectado.
- 1 = Bit forzado a uno
- 0 = Bit forzado a cero
- $\hat{\uparrow}$ = Bit seteado o borrado de acuerdo a resultados de operaci3n.
- \Downarrow = El bit puede cambiar de uno a cero, mantenerse en cero, mantenerse en uno

como resultado de una operaci3n, pero no puede cambiar de cero a uno.

Adem3s en el ap3ndice G se incluir3n cuadros de res3menes de las operaciones ciclo a ciclo de cada instrucci3n.

CAPITULO No. 2

TARJETA PRINCIPAL "M68HC11EVB"

2.1.- INTRODUCCION

En este capítulo se describe todos los detalles de la tarjeta "M68HC11EVB" en cuanto a sus características, especificaciones, instalación, funcionamiento e instrucciones de operación, así como también se dará una explicación del programa monitor BUFFALO.

La tarjeta "MC68HC11EVB" fabricada por la casa Motorola, ha sido diseñada para aprovechar todas y cada una de las características que presentan los microcontroladores de la familia M68HC11.

En el numeral 2.2 se hace un análisis de las características y especificaciones del equipo, luego en el numeral 2.3 se indica como realizar las conexiones del equipo para su perfecto funcionamiento, así como las respectivas instrucciones de instalación. En el numeral 2.4 se realiza una explicación pormenorizada del hardware de la tarjeta y su funcionamiento, luego en el numeral 2.5 se realiza una descripción de todas las instrucciones que permitan realizar los trabajos de ensamblaje y depuración de los programas de aplicaciones realizadas por el usuario. En el numeral 2.6 se realiza una descripción de como manejar el programa monitor BUFFALO y su configuración respectiva.

2.2.- CARACTERÍSTICAS Y ESPECIFICACIONES.

2.2.1.- CARACTERÍSTICAS.

Entre las principales características de la tarjeta "M68HC11EVB" se tienen las siguientes:

- Método simple de depuración y ensamblaje de código de usuario.
- Sistema de Evaluación abierto (conectores para entrada y salida de datos).
- Ensamblador en línea.
- Capacidad de descargar archivos a través de un Host Computer.
- Circuitería para evaluación y depuración de programas basado en el microcontrolador MC68HC11A1FN.
- Adaptador para interface de comunicación asincrónica (ACIA) basado en el circuito integrado MC6850.
- Pórticos de Entrada - Salida (Terminal y Host Computer) compatible con interfaces de comunicación serial RS-232.C

El componente principal de la tarjeta "M68HC11EVB" es el microcontrolador M68HC11 (Información detallada de este microcontrolador la encontramos en el capítulo número uno). Para demostrar todas las capacidades de este microcontrolador se cuenta con un programa monitor llamado "BUFFALO" (Bit User Fast Friendly Aid to Logical Operations), el cual se encuentra grabado en la memoria EPROM externa. La tarjeta "M68HC11EVB" funciona ya sea en el modo de depuración o en el modo de evaluación.

El primer modo de operación permite al programador depurar el código de los programas utilizando el programa monitor BUFFALO. El código de programa es ensamblado utilizando el ensamblador en línea del programa monitor, o en un computador y luego descargado a la RAM externa de la tarjeta, localidades desde \$C000 hasta \$D000. El programa monitor BUFFALO es también usado para depurar el código de programa ensamblado.

El segundo modo de operación permite al usuario evaluar el código de programa en un ambiente de sistema abierto. El EVB emula el modo de trabajo “single chip” (ver capítulo número 1) aun cuando el EVB opera siempre en el modo de trabajo expandido - multiplexado.

El interfaz de sistema abierto se lo obtiene a través de los circuitos integrados MCU y PRU como se puede ver en la figura número 2.1, además se cuenta con dos interfaces de comunicación serial RS-232C (Host Computer y Terminal) para permitir operaciones de comunicación y transferencia de datos entre el EVB y un computador o cualquier otro sistema provisto de este interfaz. La selección del Baud Rate es independiente para los pórfticos de entrada - salida Host y Terminal. El Baud Rate del pórftico Terminal es seleccionado por Hardware entre 300 y 9600 en pasos discretos, mientras que el Host esta fijado a 9600.

El EVB requiere de una fuente de energía con salidas de voltaje continuo de +5 [V], +12 [V] y -12 [V] y otros accesorios que se listan en la siguiente tabla:

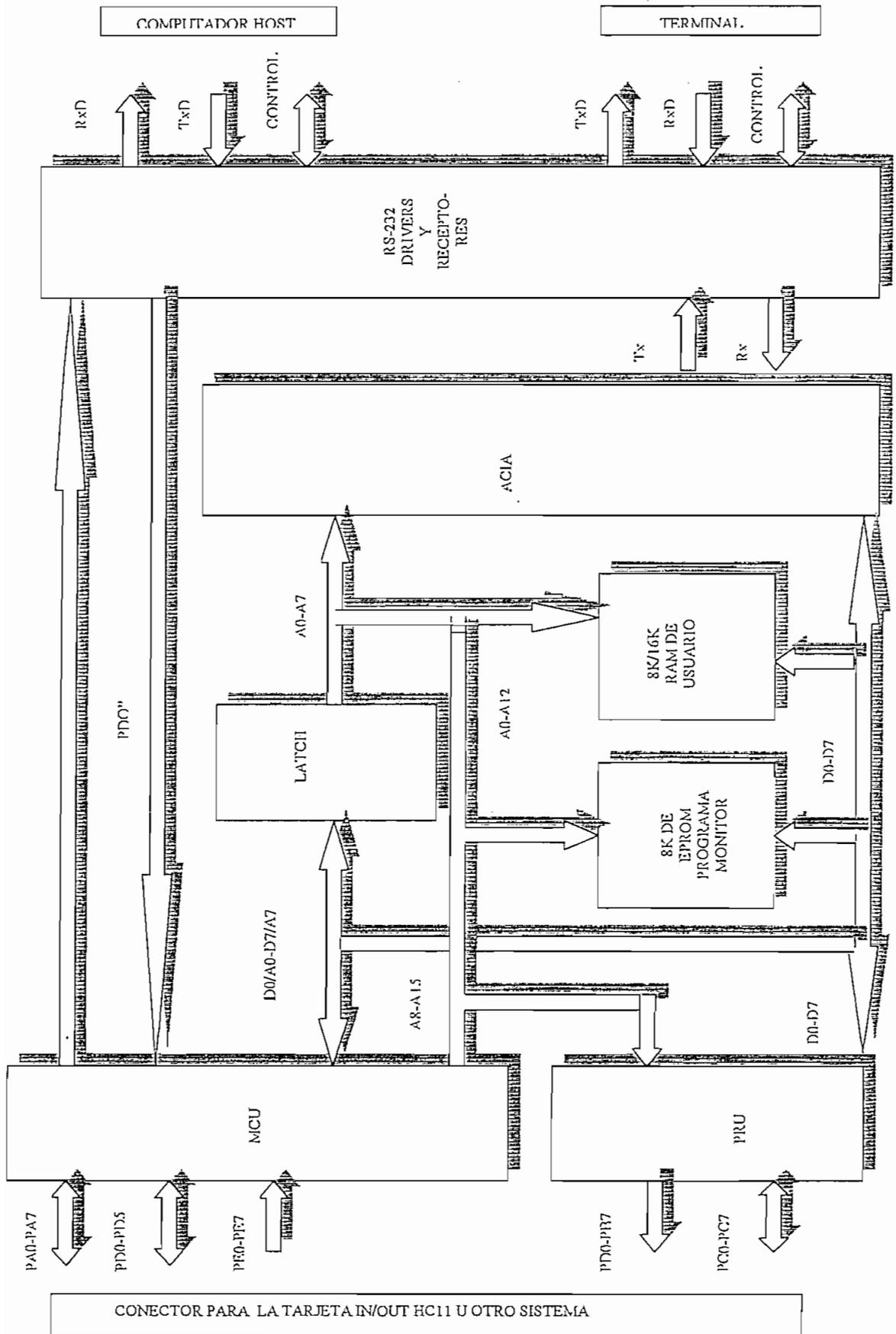


FIGURA NO. 2.1 DIAGRAMA DE BLOQUES DE TARJETA M68HC11EVB

TABLA 2.1

Requerimientos Externos de la Tarjeta "M68HC11EVB"

DISPOSITIVO	ESPECIFICACIONES
Fuente de energía de +5, +12, -12 V _{DC}	
Terminal	RS-232C compatible
Host Computer	RS-232C compatible
Cable para conexión con sistema abierto	cable plano #3350 -60 con conector 3M #3414-7060
Cable para conexión con Terminal/Host Computer	2 cable plano 3M #3365 -25 con conectores 25 "D" macho y hembra

2.2.2.- ESPECIFICACIONES.

En la tabla 2.2 se pueden ver las principales especificaciones de la tarjeta "M68HC11EVB", en la cual se enumeran los elementos primordiales, así como también las condiciones de operación, dimensiones y características de corriente y voltaje de la fuente de alimentación.

TABLA 2.2
ESPECIFICACIONES

	ESPECIFICACIONES
MCU	MC68HC11A1FN
PRU	MC68HC24FN
ACIA	MC68B50
PORTICOS DE ENTRADA SALIDA:	
Terminal	RS-232C compatible
Host Computer	RS-232C compatible
MCU sistema abierto	HCMOS -TTL compatible
TEMPERATURA:	
Operación	0 a 50 °C
Almacenamiento	-40 a +85 °C
HUMEDAD RELATIVA	(0 a 90) %
REQUERIMIENTOS DE ENERGIA	+5 V _{DC} _ 0.5 [A] máximo +12 V _{DC} _ 0.1 [A] máximo -12 V _{DC} _ 0.1 [A] máximo
DIMENSIONES	
Ancho	11.75 [cm]
Largo	17.8 [cm]

2.3.- PREPARACION DE HARDWARE E INSTRUCCIONES DE INSTALACION

2.3.1.- PREPARACION DE HARDWARE

“La tarjeta “M68HC11EVB” contiene cuatro conectores, un pulsador y seis jumpers como se puede ver en la figura número 2.2. Todos y cada uno de ellos cumplen una función específica la cual será explicada en detalle en los siguientes párrafos:

El conector P1 facilita la interconexión de la tarjeta a un sistema abierto, es decir permite tener la mayoría de entradas y salidas al, y del microcontrolador accesibles al usuario. Los conectores P2 y P3 facilitan la interconexión de la tarjeta a equipos de computadores que cumplan funciones de Terminal y Host respectivamente. El conector P4 facilita la interconexión de una fuente de energía para la tarjeta.

El pulsador S1 es un control del RESET para el usuario.

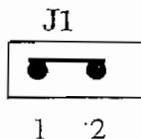
Las funciones de los jumpers J1 a J6 se observa en la tabla numero 2.3

TABLA 2.3
FUNCIONES DE LOS JUMPERS EN LA TARJETA "M68HC11EVB"

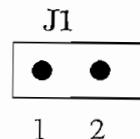
JUMPER	FUNCION
J1	Selección del Reset
J2	Selección del Reloj
J3	Habilitación de Periféricos
J4	Selección de ejecución del programa
J5	Selección del Baud Rate del terminal
J6	Habilitación de la señal Rx del pórtico Host

2.3.1.1.- Jumper de selección de RESET (J1)

Este jumper es usado para conectar un sistema de RESET externo a través del pin 17 del conector P1.



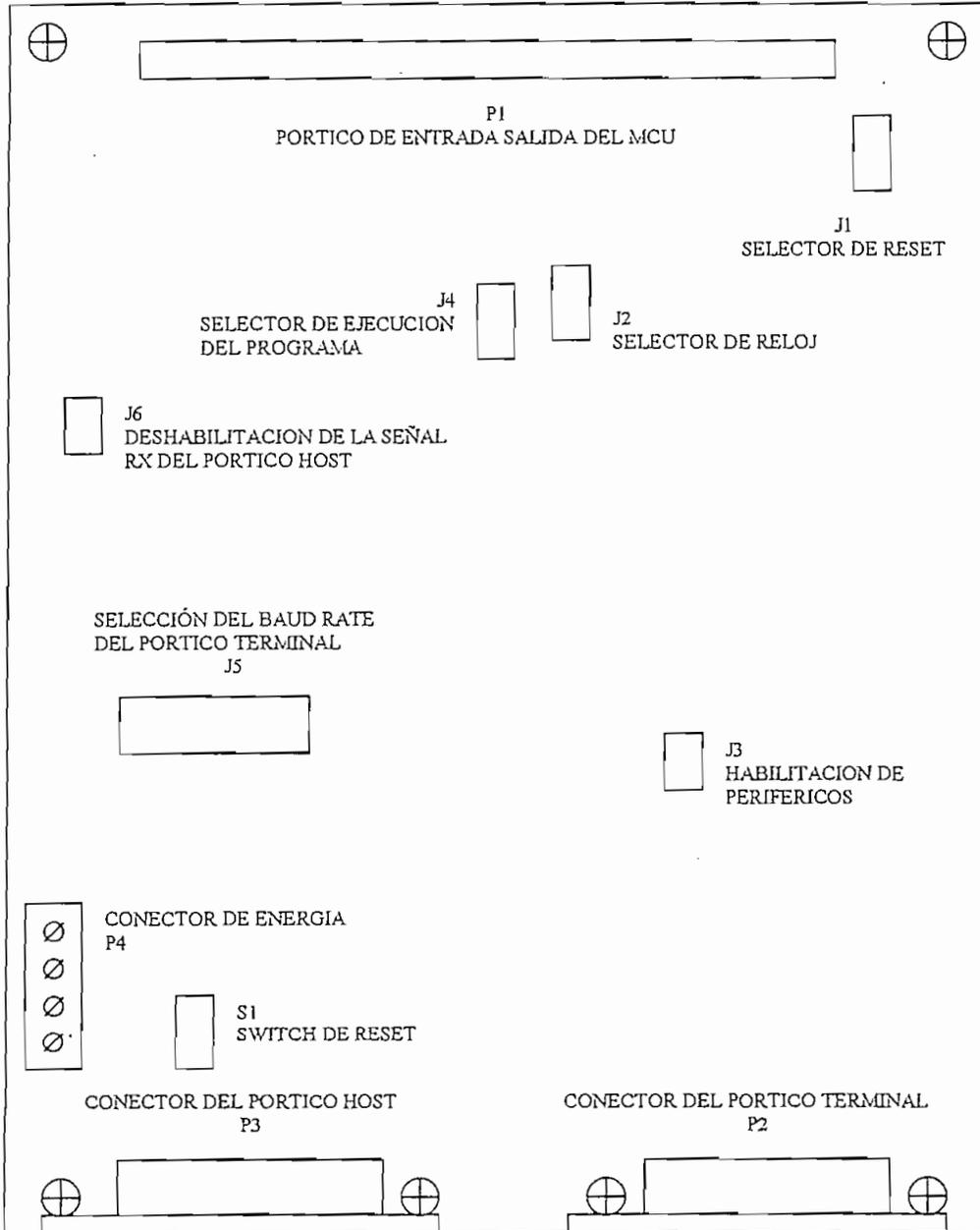
Reset Externo Activado



Reset Externo Desactivado

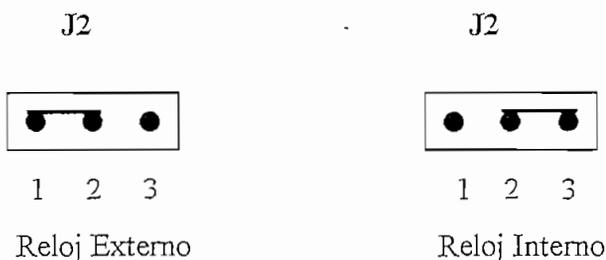
FIGURA No. 2.2

DIAGRAMA DE LOCALIZACION DE CONECTORES, JUMPERS Y PULSADOR EN LA TARJETA "M68HC11EVB"



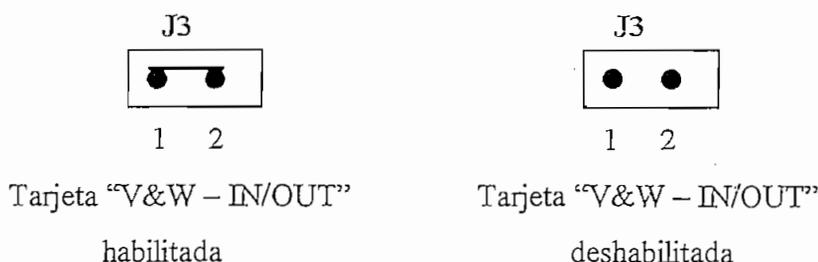
2.3.1.2.- Jumper de Selección del Reloj (J2).

Este jumper es usado para seleccionar la fuente de reloj usada por la tarjeta “M68HC11EVB”, es decir si ésta es interna o externa. La fuente de reloj interna de la tarjeta es un cristal de 8 MHz, el cual nos da un “bus rate” de 2 MHz.



2.3.1.3.- Jumper de Habilitación de Periféricos Manejados como Localidades de Memoria Externa (J3).

Este jumper es usado para aplicar la señal de habilitación de localidades de memoria externa (\$6000 - \$7FFF) en la cual se encuentran ubicados los periféricos externos, es decir es la habilitación para la tarjeta “V&W – IN/OUT”.



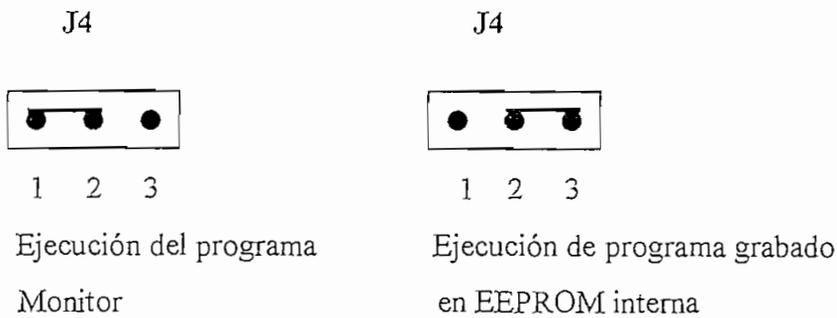
2.3.1.4.- Jumper de selección de ejecución del programa (J4).

Usado para determinar si el programa a ser ejecutado es el programa monitor BUFFALO o si es ejecutado el programa residente en la memoria EEPROM interna.

Luego de presionar el pulsador de RESET, el programa monitor analiza el estado del pin PE0, si un estado bajo es detectado, el programa monitor es ejecutado, pero si el estado del pin

PE0 es alto, el programa monitor automáticamente salta a la memoria EEPROM interna (localidad \$B600) para ejecutar el programa grabado en esta localidad de memoria.

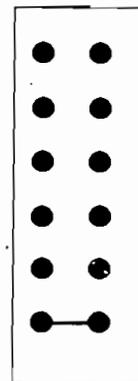
Si la línea PE0 es usada para operaciones del conversor análogo digital, la condición de ejecución del programa no es adecuado. Para solucionar este problema, programar las tres primeras localidades de la EEPROM con \$7E, \$E0 y \$0A respectivamente (Al cargar estas localidades con los valores especificados arriba, se produce un salto a la dirección \$E00A del programa monitor, en donde se realiza la habilitación del conversor análogo digital). Luego retire el jumper J4 y proceda a operar con el conversor análogo digital.



2.3.1.5.- Jumper de selección del Baud Rate del pòrtico Terminal (J5)

Este jumper es usado para seleccionar el Baud Rate del pòrtico de entrada - salida Terminal Port (P2)

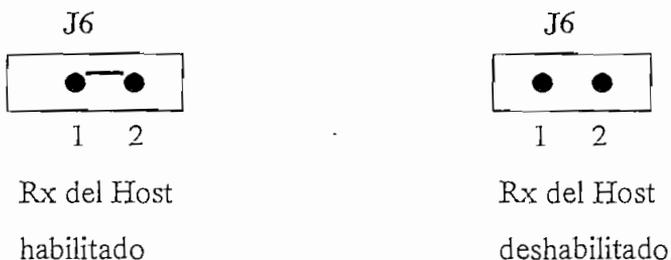
BAUD RATE	POSICION DEL JUMPER
300	1
600	2
1200	3
2400	4
4800	5
9600	6



J5

2.3.1.6.- Jumper de Habilitación de Señal Rx del Pórtico Host. (J6)

Este jumper es usado para deshabilitar la línea Rx del pórtico Host (conector P3, pin 2), para cuando la tarjeta es usada para comunicación a través del pórtico P1 de la tarjeta "M68HC11EVB".



2.3.2.- INSTRUCCIONES DE INSTALACION.

2.3.2.1.- Descripción de Conectores.

En los gráficos 2.3 a 2.6 se puede visualizar todos y cada uno de los pines de los pórticos P1 a P4 del EVB, cuyas funciones serán explicadas en detalle en el numeral 2.4 de este capítulo.

FIGURA No. 2.3
CONECTOR P4

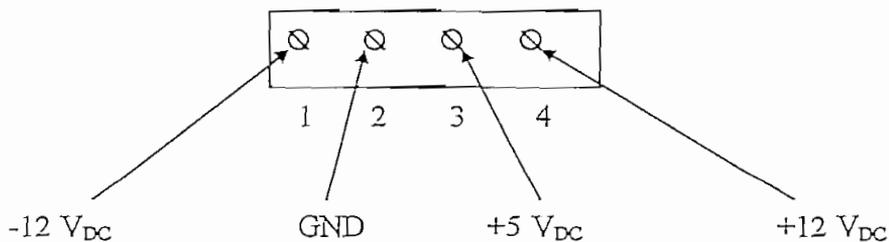


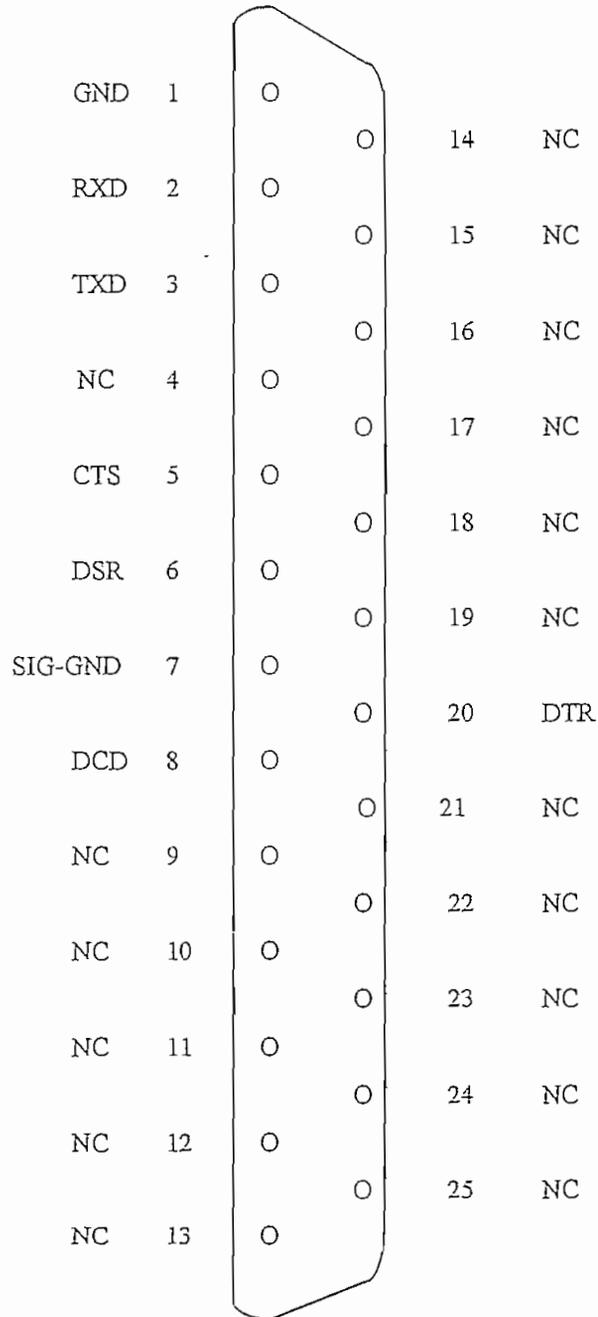
FIGURA No. 2.4

CONECTOR P1

GND	1	O	O	2	MODEB
NC	3	O	O	4	STRA
E	7	O	O	8	XTAL
PC0	9	O	O	10	PC1
PC2	11	O	O	12	PC3
PC4	13	O	O	14	PC5
PC6	15	O	O	16	PC7
RESET	17	O	O	18	XIRQ*
IRQ*	19	O	O	20	PD0
PD1	21	O	O	22	PD2
PD3	23	O	O	24	PD4
PD5	25	O	O	26	VDD
PA7	27	O	O	28	PA6
PA5	29	O	O	30	PA4
PA3	31	O	O	32	PA2
PA1	33	O	O	34	PA0
PB7	35	O	O	36	PB6
PB5	37	O	O	38	PB4
PB3	39	O	O	40	PB2
PB1	41	O	O	42	PB0
PE0	43	O	O	44	PE4
PE1	45	O	O	46	PE5
PE2	47	O	O	48	PE6
PE3	49	O	O	50	PE7
VRL	51	O	O	52	VRH
NC	53	O	O	54	NC
NC	55	O	O	56	NC
NC	57	O	O	58	NC
NC	59	O	O	60	NC

FIGURA No. 2.5

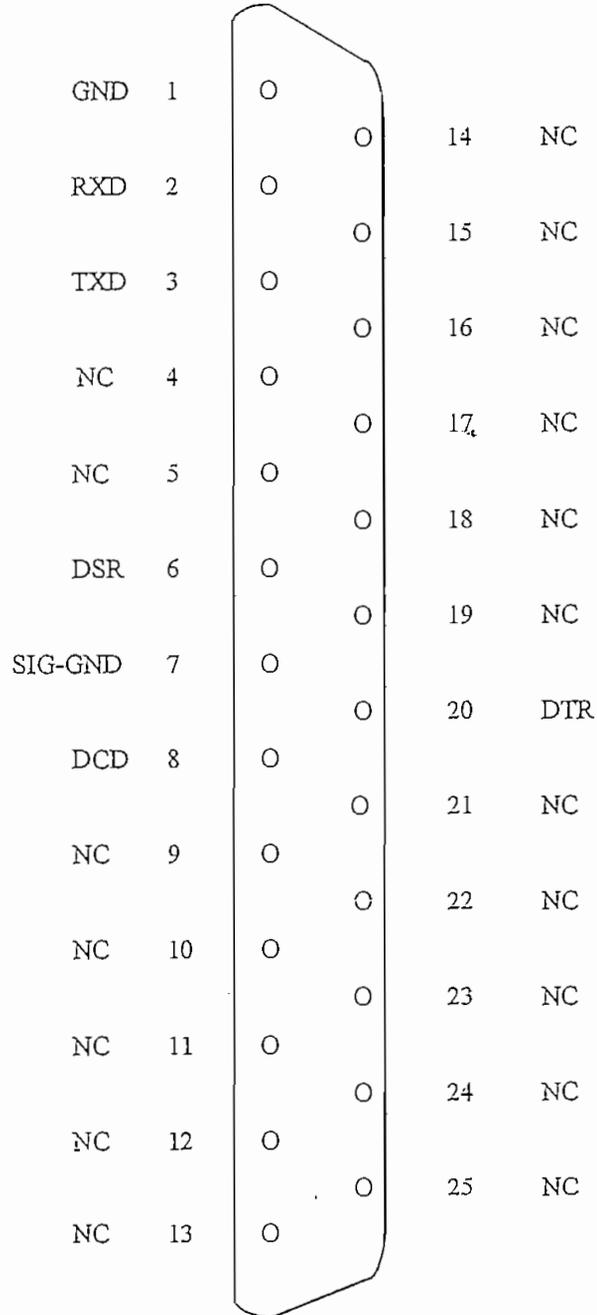
CONECTOR P2



TERMINAL

FIGURA No. 2.6

CONECTOR P3



HOST

2.3.2.2.- Procedimientos de Chequeo de Instalación de la Tarjeta “M68HC11EVB”

El siguiente procedimiento se debe seguir para verificar el funcionamiento correcto de la tarjeta “M68HC11EVB” para trabajo en operaciones básicas.

- Conectar la fuente de energía al conector P4 de acuerdo a la figura 2.6
- Conectar un computador terminal al conector P2, y chequear que el EVB este configurado para el mismo Baud Rate que el computador.
- En el computador realizar:

 Ingresar al programa Kermit desde DOS o desde cualquier versión de WINDOWS y correr los comandos:

```
    Kermit-MS>SET PORT 1 (O 2).J
```

```
    Kermit-MS>SET BAUD 9600 (u Otro Baud Rate).J
```

```
    Kermit-MS>CONNECT.J
```

 Luego de lo cual le aparecerá el siguiente mensaje:

```
    [Conecting to Host, type Control -] C to return to PC]
```

- Al encender la fuente de energía se produce un Power On Reset, esta condición causa al MCU y a los circuitos de entrada salida de usuario a ser reseteados, y el programa monitor BUFFALO es invocado, desplegándose en la pantalla del computador el siguiente mensaje:

 BUFFALO 2.2 (ext) - Bit User Friendly Aid to Logical Operation

2.4.- DESCRIPCION DE HARDWARE

2.4.1.- DESCRIPCION GENERAL.

El control de los modos de operación evaluación y depuración del EVB está dado por medio del programa monitor BUFFALO (residente en la memoria EPROM externa al microcontrolador) utilizando el pòrtico terminal. El interfaz de sistema abierto está dado por los pines de salida del microcontrolador directos al pòrtico P1 y del amplificador de pòrticos PRU.

Además se cuenta con los pórtricos Terminal y Host destinados a comunicación y transferencia de datos entre el EVB y computadores conectados como Terminal y Host respectivamente.

Todos los elementos y su interconexión dentro de la tarjeta “M68HC11 EVB” lo podemos ver en el apéndice E.

2.4.2.- EL MICROCONTROLADOR.

El microcontrolador MC68HC11A1FN (referido en el apéndice E como U10) trabaja en el modo de operación expandido multiplexado lo cual se consigue colocando una señal de $+5 V_{DC}$ a los pines MODA y MODB del MCU.

Un cristal de 8 MHz es colocado sobre sus pines XTAL y EXTAL permitiendo así generar una señal de reloj interna (E-clock) de 2 MHz.

Se debe notar que el programa monitor usa las localidades de RAM interna del MCU (\$0036 a \$00FF) y que los registros de control están ubicados en las localidades (\$1000 a \$103F).

2.4.3.- AMPLIADOR DE PORTICOS PRU

La tarjeta “M68HC11EVB” trabaja en el modo de operación expandido multiplexado, por lo que un dispositivo amplificador de pórtricos M68HC24 (referido en el apéndice E como U1) es usado para recuperar los pórtricos de entrada PB y PC del microcontrolador y las líneas de control STRA y STRB usadas para el modo de operación “single – chip”.

El amplificador de pórtricos recupera las líneas requeridas en el modo “single – chip” para tenerlas accesibles al usuario en el sistema abierto del EBV conector P1.

El sistema constituido por un microcontrolador M68HC11 trabajando en el modo expandido, un amplificador de pórtricos MC68HC24, un retenedor octal HC373, y una memoria EPROM externa trabaja como si un microcontrolador MC68HC11A8 lo hiciera en el modo “single – chip”, lo que permite que un programa de aplicación sea desarrollado y probado antes de ser grabado en una memoria ROM fija.

La lógica del M68HC11 fue específicamente diseñada para permitir la emulación de las funciones del modo “single – chip” utilizando el MC68HC24. Primero, las direcciones asociadas a los pórtricos PB y PC y sus funciones de entrada - salida de handshake son tratadas como direcciones externas cuando el MCU esta trabajando en el modo expandido - multiplexado. Luego

las interrupciones asociadas con el sistema de entrada - salida de handshake son vectorizados en la misma dirección que el vector IRQ, así la salida de interrupción del MC68HC24 es conectado a la entrada de interrupción IRQ del microcontrolador, y el software escrito en un sistema expandido que incluye un MC68HC24 operará exactamente como si estuviera escrito en la memoria ROM interna de un MC68HC11A8 trabajando en el modo "single - chip".

2.4.4.- MEMORIA

El mapa de memoria del EVB es un mapa de diseño simple, y se lo puede ver en la tabla número 2.4.

Se cuenta con una RAM de usuario en la cual se depura el código de programa, es decir corregir errores del programa antes de pasarlo a la memoria EEPROM y ejecutarlo en el modo de Evaluación.

Para evaluar programas que normalmente deberían estar almacenados en una memoria ROM o EPROM se cuenta con una memoria RAM de 8 Kbytes (referido en el apéndice E como U5) con un tiempo de acceso de 250 ns. Para habilitar los periféricos que serán tratados como memoria externa se debe colocar el jumper J3 con lo cual se podrá tener acceso a la tarjeta "V&W - IN/OUT".

2.4.5.- DECODIFICACION Y DEMULTIPLEXION DE DIRECCIONES.

El circuito integrado MC74HC138 (U6) realiza la decodificación de direcciones, segmentando la memoria en 8 bloques de 8 Kbytes.

El bus bajo de direcciones (D0..D7) y la línea de datos son demultiplexados usando el circuito integrado MC74HC373 (U2), para así comunicarse con las memorias EPROM, RAM periféricos externos y el ACIA. El PRU usa una entrada multiplexada directa del MCU.

TABLA No. 2.4
MAPA DE MEMORIA DEL EVB

RAM DE USUARIO	\$0000 - \$0035
STACK POINTER	\$0036 - \$004A
VARIABLES DEL PROGRAMA MONITOR	\$004B - \$00C3
TABLA DE VECTORES DE SALTO	\$00C4 - \$00FF
PRU Y REGISTROS DE DATOS Y CONTROL	\$1000 - \$17FF
NO USADO	\$1800 - \$3FFF
FLIP-FLOP DECODIFICADOR	\$4000 - \$5FFF
PERIFERICOS EXTERNOS	\$6000 - \$7FFF
NOT USED	\$8000 - \$97FF
TERMINAL ACIA	\$9800 - \$9FFF
NO USADO	\$A000 - \$B5FF
EEPROM	\$B600 - \$B7FF
NO USADO	\$B800 - \$BFFF
RAM DE USUARIO	\$C000 - \$DFFF
EPROM PARA EL PROGRAMA MONITOR	\$E000 - \$FFFF

2.4.6.- CIRCUITOS DE INTERFAZ SERIAL RS - 232C.

El EVB usa el circuito integrado MC68B50 (U9) ACIA para establecer comunicación con un terminal, este circuito integrado es un manejador (driver) de interfaz RS - 232C, y la conexión se la realiza a través del pórtico de entrada - salida terminal (P2). El baud rate es seleccionable por hardware en pasos discretos entre 300 y 9600 bauds al cambiar la posición del jumper J5.

Otro sistema con manejador de interfaz serial RS - 232C es obtenido del SCI del microcontrolador el cual trabaja a 9600 baud usando un reloj (E-clock) de 2 MHz, sin embargo este baud rate puede ser cambiado por software al reprogramar el registro BAUD. Las líneas de

entrada - salida de este sistema de comunicación las encontramos en el p rtico Host (P3). El switch digital MC74HC4076 (U7) y el flip - flop tipo D MC74HC74 (U11) son usados para determinar si los datos recibidos en el pin PD0 (RXD) del microcontrolador vienen del p rtico Host o del sistema abierto (P1). El control de la l nea de recepci n por parte del p rtico Host se da al escribir un uno l gico en el bit cero en alguna direcci n comprendida entre \$4000 y \$5FFF). De otro modo al escribir un cero en el bit cero en el mismo rango de direcciones, se conecta el p n RXD del microcontrolador al sistema abierto (P1).

Como las l neas de Handshake del RS232 no son usadas, un retardo de aproximadamente 300 milisegundos se da entre caracteres sucesivos enviados al computador Host durante la ejecuci n del comando LOAD en el programa monitor.

2.5.- INSTRUCCIONES DE OPERACI N

2.5.1.- LIMITACIONES.

La precauci n principal que se debe tener es cuando se programe o borre localidades EEPROM del microcontrolador. El bit ROMON del registro CONFIG es borrado para deshabilitar la ROM interna del microcontrolador, permitiendo as  al programa monitor BUFFALO contenido en la memoria EPROM externa, controlar las operaciones realizadas en el EVB.

El SCI del microcontrolador ha sido seteado para trabajar a 9600 bauds, usando un reloj E-clock del bus externo de 2 MHz. Este baud rate puede ser cambiado por software, reprogramando el registro BAUD en la subrutina ONSCI del programa monitor BUFFALO (o en cualquier programa).

El programa monitor usa la RAM interna del MCU desde la localidad \$0036 hasta la localidad \$00FF, los registros de control son localizados en las direcciones desde \$1000 hasta \$103F.

El programa monitor usa la salida Output Compare cinco OC5 para la instrucci n TRACE, de ah  que OC5 no deber a ser usado en rutinas creadas por el usuario.

El EVB permite al usuario usar todas las características del software de evaluación del BUFFALO, sin embargo debería notarse que el BUFFALO usa localidades de RAM interna del MCU dejando solo 54 bytes para el usuario (\$0000 - \$0035).

2.5.2.- PROCEDIMIENTOS DE OPERACIÓN.

El EVB es una herramienta de depuración y evaluación simple, diseñada para trabajar en los dos modos de trabajo arriba mencionados. El jumper J4 es usado para determinar si el programa monitor BUFFALO es ejecutado, o si éste debe realizar un salto a la EEPROM interna del microcontrolador. Ver sección 2.3.

Al presionar el botón de RESET, el programa monitor detecta el estado de la línea PE0. Si un estado bajo es detectado, el programa monitor es ejecutado y su indicador mostrado en la pantalla del computador, pero si de otro modo un estado alto es detectado, el programa monitor automáticamente saltará a la memoria EEPROM (\$B600) y ejecutará el programa de usuario grabado a partir de ésta localidad, sin desplegar ningún carácter en la pantalla del computador.

Existen dos modos de operación del EVB, éstos son el modo de Depuración y el modo de Evaluación.

2.5.2.1.- Modo de Depuración.

Este modo de operación permite depurar el código de usuario bajo el control del programa monitor. El código de usuario puede ser ensamblado por dos métodos. El primer método es el ensamblaje de código usando el ensamblador en línea del programa monitor BUFFALO en la RAM de usuario (\$C000 - \$DFFF). El segundo método es ensamblar el código en un computador y luego descargar el código a la RAM de usuario del EVB. El formato del archivo a ser descargado debe ser "Motorola S-Records". Ver apéndice C.

Por lo tanto el programa monitor al trabajar en este modo es usado para depurar el código de usuario ensamblado.

2.5.2.2.- Modo de Evaluación

El segundo modo de operación permite al usuario evaluar el código de programa en un sistema abierto, utilizando la memoria del microcontrolador MC68HC11. Esto se completa reubicando el código de las localidades (\$C000 - \$DFFF) a (\$E000 - \$FFFF), las localidades de memoria RAM (\$0000 - \$00FF) y EEPROM (\$B600 a \$B7FF) están también disponibles al

usuario. EL EVB entonces emula un equivalente EPROM de la memoria ROM en el modo de operación “Single – Chip”. Ver sección 1.3.

El EVB emula como si estuviera en un modo de operación “single – chip”, aun cuando siempre la operación se da en el modo de operación Expandido - Multiplexado.

2.5.3.- FORMATO DE LA LINEA DE COMANDOS.

El formato de la línea de comandos es como sigue:

><comando>[<parámetros>](↵).

donde:

>	Indicador del comando monitor
<comando>	mnemonico del comando
<parámetros>	expresión o direcciones
(↵)	Tecla “enter”

El formato de la línea de comandos es definido usando caracteres especiales los cuales tienen el siguiente significado:

< > encierra sintaxis de variables

[] encierra campos opcionales

[].. encierra campos de comandos repetidos.

Estos últimos caracteres no son ingresados por el usuario, son únicamente para propósitos de definición.

Todos los números ingresados son interpretados como hexadecimal.

Todos los comandos pueden ser ingresados indistintamente con letras mayúsculas o minúsculas. Automáticamente son convertidos a mayúsculas.

Un máximo de 35 caracteres puede ser ingresado en una línea de comandos. A partir del carácter número 36, el programa monitor automáticamente termina el programa.

Después que un comando ha sido ingresado, presionar la tecla “enter” si se quiere repetir el comando.

2.5.4.- COMANDOS DEL PROGRAMA MONITOR

Los comandos del programa monitor Buffalo se listan en orden alfabético en la tabla número 2.5, y a continuación se describen en detalle todos los comandos tabulados. Sin embargo existen teclas de funciones específicas:

(CTRL)A Salir del modo transparente o ensamblador.

(CTRL)B Envía comandos de parada al Host en el modo transparente.

(CTRL)H Espacio en blanco.

(CTRL)W Espera, imagen del monitor congelada. La ejecución se restablece al presionar cualquier tecla.

(DELETE) Abandonar o cancelar un comando.

(↵) Ingresar un comando o repetir el último comando.

ASM <dirección> .- Permite ingresar al modo ensamblador/desensamblador interactivo, <dirección> es la dirección de inicio de la operación de ensamblado (sino se especifica la dirección, la operación de ensamblado se la realiza en la RAM interna.).

BF <dirección1> <dirección2> <dato> .- Permite llenar un bloque de memoria con un byte específico.<dirección1> dirección de inicio del bloque, <dirección2> dirección de fin de bloque, <dato> valor Hexadecimal a ser llenado.

BR <dirección> .- Permite ubicar puntos de parada en la ejecución de un programa.

BULK .- Permite borrar todo el contenido de la EEPROM interna del microcontrolador (\$B600-\$B7FF).

BULKALL .- Permite borrar todo el contenido de la EEPROM interna del microcontrolador (\$B600-\$B7FF) y adicionalmente borra el registro de configuración (CONFIG) localidad (\$103F).

TABLA No. 2.5
COMANDOS DEL PROGRAMA MONITOR

COMANDO	DESCRIPCION
ASM [<dirección>]	Ensamblar/Desensamblar
BF <dirección1><dirección2><dato>	Llenar un bloque de memoria con un dato
BR <dirección>	Colocar un punto de parada
BULK	Borrar toda la memoria EEPROM
BULKALL	Borrar memoria EEPROM y registro CONFIG
CALL [<dirección>]	Ejecutar subrutina
G [<dirección>]	Ejecutar programa
HELP	Desplegar comandos del programa monitor
LOAD	Descargar archivos con formato S – records.
MD [<dirección1>[<dirección2>]]	Desplegar mapa de memoria
MM [<dirección>]	Modificar memoria
MOVE <dirección1><dirección2><desti.>	Mover memoria a una nueva localidad
P	Continuar luego de un punto de parada
RM [p,y,x,a,b,]	Modificar registros
T	Ejecutar instrucción por instrucción
TM	Ingresa a modo transparente
VERIFY	Comparar memoria descargada

CALL <dirección>.- Permite ejecutar una subrutina , <dirección> es la dirección donde inicia la subrutina .

G <dirección> .- Permite la ejecución de un programa , <dirección> es la dirección donde inicia la ejecución del programa.

HELP .- Despliega los comandos del programa monitor.

LOAD T .- Permite descargar archivos en formato S-Records (archivos ensamblados con formato nombre.S19), vía el p \acute{o} rtico terminal (P2).

MD <direcci3n1> <direcci3n2> .- Despliega en la pantalla el contenido de un bloque de memoria. <direcci3n1> direcci3n de inicio del bloque de memoria <direcci3n2> direcci3n final del bloque.

MM <direcci3n> .- Permite examinar y modificar el contenido de localidades de memoria <direcci3n> Es la localidad de memoria donde se inicia la modificaci3n del contenido de la misma. **Tener precauci3n al modificar los registros internos del microcontrolador.**

MOVE <direcci3n1> <direcci3n2> <destino>.- Permite copiar los contenidos de un bloque de memoria a otro bloque de memoria . Este comando es \acute{u} til para programar la memoria EEPROM interna del microcontrolador.<direcci3n1> direcci3n de inicio del bloque <direcci3n2> direcci3n final del bloque <destino> direcci3n de inicio donde se mueve el bloque.

P .- Permite continuar la ejecuci3n de un programa luego de un comando BR (Break point).

RM.- Permite modificar el contenido de los registros: Contador de Programa (P), Registro Indice (Y), Registro Indice (X), Acumulador (A), Acumulador (D) y Stack Pointer.

T <n>.- Permite la ejecuci3n de un programa paso a paso <n> es el n \acute{u} mero en Hexadecimal de instrucciones a ejecutar.

TM .- Este comando permite conectar el p \acute{o} rtico Host al p \acute{o} rtico Terminal del EVB con lo cual se permite comunicaci3n directa entre el computador Terminal y el computador Host. Todas las entradas y salidas entre los p \acute{o} rticos son ignorados por el EVB hasta que el car \acute{a} cter de EXIT sea ingresado por el terminal.

VERIFY.- Similar al comando LOAD excepto que el comando Verify permite al EVB comparar los datos de los archivos S - records descargados, con los datos almacenados en memoria.

2.6.- EL PROGRAMA MONITOR

2.6.1.- DESCRIPCION DEL PROGRAMA

En este capítulo se da una descripción general del programa monitor BUFFALO lo cual servirá al usuario para entender la estructura básica del programa, y para modificar o adaptar el programa para aplicaciones específicas.

El programa monitor dado para el EVB llamado BUFFALO (Bit User Fast Friendly Ad to Logical Operations) sirve para comunicar el EVB por medio del circuito integrado MC6850 Adaptador de Interfaz de Comunicación Asíncrona ACIA y por el interfaz de comunicación serial SCI del microcontrolador. El listado del programa monitor en detalle lo encontramos en el apéndice B.

El programa monitor se encuentra en la memoria EPROM externa al microcontrolador en las localidades (\$E000 a \$FFFF). El microcontrolador que viene en el EVB tiene el bit ROMON del registro de configuración CONFIG borrado, por lo tanto se encuentra deshabilitada la memoria ROM interna del microcontrolador.

El tener el programa monitor en la memoria EPROM externa es una gran ventaja, porque ésta permite al usuario añadir instrucciones para adaptar el programa monitor para aplicaciones específicas. El módulo del programa monitor incluye todas las partes requeridas por alguno de los módulos de comandos individuales. De este modo el módulo principal es un núcleo del programa monitor BUFFALO y consta de cinco partes que son:

- 1.- Inicialización
- 2.- Interprete de Comandos
- 3.- Rutinas de Entrada - Salida
- 4.- Subrutinas de Utilidad
- 5.- Tabla de Comandos

2.6.1.1.- Inicialización.

Esta parte del módulo principal contiene todos los códigos de inicialización del reset. En esta parte se configuran las localidades de memoria RAM, y el canal de entrada - salida para el pórtico terminal. Para configurar el pórtico de entrada - salida terminal, el programa monitor debe

determinar si el terminal es conectado al SCI o a un ACIA externo. Esta prueba se la realiza enviando un mensaje de señal a los dos p \acute{o} rticos y luego esperando la respuesta de cualquiera de los p \acute{o} rticos. Cuando el BUFFALO recoge la se \acute{n} al de retorno de alg \acute{u} n p \acute{o} rtico, \acute{e} ste es usado en adelante para todas las subsecuentes operaciones de entrada - salida por el p \acute{o} rtico terminal.

2.6.1.2.- Interprete de Comandos.

El interprete de comandos nos sirve para leer los caracteres ASCII del buffer del p \acute{o} rtico terminal hasta que sea recibido el car \acute{a} cter ASCII de la tecla "enter" o el car \acute{a} cter ASCII de la tecla "slash (/)".

El campo de comandos es entonces sacado del buffer de entrada y ubicado en el buffer de comandos. Una tabla de comandos es buscada y si coinciden los caracteres ingresados con alg \acute{u} n comando de los de la tabla de comandos, el correspondiente m \acute{o} dulo de comando es llamado como una subrutina. Todos los comandos regresan el control al interprete de comandos luego de completar su operaci \acute{o} n.

2.6.1.3.- Rutinas de Entrada - Salida.

La secci \acute{o} n de entrada - salida del m \acute{o} dulo principal consiste de un grupo de subrutinas de supervisi \acute{o} n y tres grupos de rutinas de manejo ("DRIVERS"). Las rutinas de supervisi \acute{o} n son INIT, INPUT y OUPUT. Esas rutinas determinan cual rutina de manejo debe ser llamada para realizar la acci \acute{o} n espec \acute{i} fica.

Cada grupo de rutinas de manejo consiste de una rutina de inicializaci \acute{o} n, una rutina de entrada y una de salida. Un grupo de rutinas de manejo es para el p \acute{o} rtico SCI y esas rutinas son llamadas ONSCI, INSCI y OUTSCI. El segundo grupo de rutinas de manejo es para un DUART y esas rutinas son llamadas ONUART, INUART y OUTUART. El tercer grupo de rutinas de manejo es para un ACIA y esas rutinas son llamadas ONACIA, INACIA y OUTACIA.

Todas las entradas y salidas de comunicaci \acute{o} n son controladas por tres localidades de memoria RAM: IODEV, EXTDEV y HOSTDEV.

EXIDEV especifica el tipo de dispositivo externo: 0 = ninguno, 1 = ACIA, 2=DUART.

HOSTDEV especifica cual p \acute{o} rtico de entrada salida es usado para comunicaciones por el p \acute{o} rtico Host: 0 = SCI, 1 = ACIA, 3 = DUARTB.

IODEV indica al supervisor de rutinas cuales subrutinas de manejo usar: 0 = SCI, 1= ACIA, 2 = DUARTA, 3 = DUARTB.

La rutina de inicialización INIT configura el pòrtico con un formato de transmisión serial de 8 bits de datos, 1 bit de parada y ninguno de paridad. Para el SCI , el baud rate es seteado a 9600 para un cristal de 8 MHz. (E - clock de 2 MHz), sin embargo un Baud Rate diferente puede ser alcanzado al modificar el registro BAUD (\$102B). Para mayor información respecto a este registro, ver capítulo 1 sección 1.8.

La rutina de entrada INPUT lee del pòrtico especificado. Si un carácter es recibido, el carácter es pasado al acumulador A. Si ningún carácter es recibido, un cero es puesto en el acumulador A. Esta rutina no espera recibir algún carácter antes de retornar.

La rutina de salida OUPUT toma los caracteres ASCII cargados en el acumulador A y escribe al pòrtico de entrada - salida especificado.

Esta rutina espera hasta que el carácter sea transmitido antes de retornar.

2.6.1.4.- Subrutinas de Utilidad.

Existen varias subrutinas que están disponibles al usuario para realizar tareas de entrada - salida. Una tabla de saltos ha sido ubicada en la memoria EPROM exactamente sobre los vectores de interrupción (\$FFA0 a \$FFD2).

Para usar esas subrutinas, ejecutar una instrucción de salto a subrutina (JSR) a la dirección apropiada de acuerdo a la tabla de saltos. Si no se especifica otra cosa, todas las acciones realizadas con estas subrutinas son enviadas al pòrtico terminal. El redireccionamiento del pòrtico de entrada - salida se lo alcanza al poner el valor deseado en la localidad de memoria RAM IODEV. Ver sección 2.6.1.3 de este capítulo.

Las subrutinas disponibles al usuario son las siguientes:

UPCASE: Convierte los caracteres (contenidos en el acumulador A) de letras minúsculas en sus respectivos caracteres de letras mayúsculas.

WCHEK: Examina el valor del carácter en el acumulador A y setea el bit Z del CCR si el carácter es un espacio en blanco (espacio, coma, tab).

DCHEK: Examina el valor del carácter en el acumulador A y setea el bit Z del CCR si el carácter es un delimitador (enter o espacio en blanco).

INIT: Inicializa el p rtico de entrada - salida

INPUT: Lee de los p rticos de entrada - salida.

OUTPUT: Escribe en los p rticos de entrada - salida.

OUTLHLF: Convierte el contenido de la mitad izquierda del acumulador A en c digo ASCII y lo saca al p rtico terminal.

OUTRHLF: Convierte el contenido de la mitad derecha del acumulador A en c digo ASCII y lo saca al p rtico terminal.

OUTA: Saca el car cter ASCII del contenido del acumulador A.

OUT1BYT: Convierte el byte binario de la direcci n contenida en el registro X en dos caracteres ASCII y los saca. Al retornar, el puntero X apunta a la siguiente direcci n.

OUT1BSP: Convierte el byte binario de la direcci n contenida en el registro X en dos caracteres ASCII y los saca seguido de un espacio. Al retornar, el puntero X apunta a la siguiente direcci n.

OUT2BSP: Convierte dos bytes binarios consecutivos iniciando en la direcci n apuntada por el registro X a cuatro caracteres ASCII y los pone en el p rtico de salida, seguido por un espacio. Al retornar el puntero X apunta a la siguiente direcci n.

OUTCRLF: Saca el car cter ASCII de la tecla enter.

OUTSTRG: Saca una cadena de caracteres ASCII de los bytes indicados por el registro  ndice X hasta que el car cter sea un fin de transmisi n (\$04).

OUTSTRGO: Lo mismo que OUTSTRG excepto que no se carga el car cter de la tecla enter.

INCHAR: Ingresa los caracteres ASCII del acumulador A y regresa. Esta rutina se queda en un lazo hasta que el car cter sea recibido.

VECINIT: Usada durante la inicializaci n para definir el  rea de los vectores de interrupci n en la memoria RAM. Esta rutina o alguna similar debe ser incluida en un programa de usuario que sea accedido por el salto a la rutina \$B600 (EEPROM interna) del programa monitor BUFFALO.

Los saltos a las subrutinas de utilidad para realizar tareas de entrada salida se muestran en la tabla n mero 2.6

TABLA No. 2.6

TABLA DE SALTOS A SUBROUTINAS DE UTILIDAD

SUBROUTINA	DIRECCION
UPCASE	\$FFA0
WCHEK	\$FFA3
DCHEK	\$FFA6
INIT	\$FFA9
INPUT	\$FFAC
OUTPUT	\$FFAF
OUTLHLF	\$FFB2
OUTRHLF	\$FFB5
OUTA	\$FFB8
OUT1BYT	\$FFBB
OUT1BSP	\$FFBE
OUT2BSP	\$FFC1
OUTCRLF	\$FFC4
OUTSTRG	\$FFC7
OUTSTRGO	\$FFCA
INCHAR	\$FFCD
VECINIT	\$FFD0

2.6.1.5.- Tabla de Comandos.

La tabla de comandos consiste de tres líneas por cada entrada. El primer byte es el número de caracteres en el nombre del comando, la segunda entrada es el carácter ASCII del nombre del comando. La tercera entrada es la dirección de inicio del módulo de comando. Como un ejemplo tenemos:

FCB 3	tres caracteres en el nombre del comando.
FCC 'ASM'	Cadena ASCII del nombre del comando.
FDB #ASM	Salto a la dirección del módulo del comando.

Cada comando en el programa monitor BUFFALO es un módulo individual, de modo que para añadir o quitar algún comando, todo lo que se requiere es incluir un nuevo módulo de comando o borrar un módulo existente (borrar también la entrada en la tabla de comandos) respectivamente.

2.6.2.- VECTORES DE INTERRUPCION.

Para acceder a los vectores de interrupción residentes en la memoria RAM interna del microcontrolador se tiene que:

A cada vector se le ha asignado un campo de tres bytes ubicado en las direcciones \$00C4 a \$00FF del mapa de memoria del EVB. Es aquí donde el programa monitor espera que resida la RAM interna del microcontrolador. Cada vector consta de tres bytes los cuales son usados para incluir saltos a las respectivas rutinas de servicio. En la tabla número 2.7 se listan los vectores de interrupción y sus respectivos campos asociados.

Para utilizar los vectores especificados en la tabla número 2.7, el usuario debe insertar un código de salto extendido en los tres bytes del campo del vector requerido. Por ejemplo, si se quiere habilitar la interrupción IRQ se procede:

- 1) Ubicar \$7E (JMP) en la localidad \$00EE
- 2) Ubicar la dirección de la rutina de servicio en las localidades \$00EF y \$00F0 teniendo:

```
$00EE      7E 80 00   JMP   IRQ   SERVICE
```

Durante la inicialización el programa monitor chequea el primer byte de cada grupo de tres localidades de memoria. Si no se encuentra el código de salto (\$7E), el programa monitor realiza un salto a la subrutina STOPIT. Esto sirve para asegurarse que no se inicialicen interrupciones las cuales podrían causar operaciones no deseadas durante el encendido y apagado del EVB. Si una interrupción es accidentalmente dada, la subrutina STOPIT fuerza a parar la secuencia de instrucciones a ser ejecutada. El usuario puede reemplazar alguno de los saltos a STOPIT con un salto a una subrutina de servicio escrita por el usuario.

Si el botón de reset S1 es presionado, el programa monitor no sobrescribe esas instrucciones de salto del usuario, así no se necesita redireccionar los saltos, cada vez que se aplasta el botón de reset S1.

TABLA No. 2.7
VECTORES DE INTERRUPCION

VECTOR DE INTERRUPCION	LOCALIDAD DE MEMORIA
SERIAL COMMUNICATIONS INTERFACE	S 00C4 - S00C6
SERIAL PERIPHERICAL INTERFACE (SPI)	S 00C7 - S00C9
PULSE ACCUMULADOR INPUT EDGE	S 00CA - S00CC
PULSE ACCUMULADOR OVERFLOW	S 00CD - S00CF
TIMER OVERFLOW	S00D0 - S00D2
TIMER OUPUT COMPARE 5	S00D3 - S00D5
TIMER OUPUT COMPARE 4	S00D6 - S00D8
TIMER OUPUT COMPARE 3	S00D9 - S00DB
TIMER OUPUT COMPARE 2	S00DC - S00DE
TIMER OUPUT COMPARE 1	S00DF - S00E1
TIMER INPUT COMPARE 3	S00E2 - S00E4
TIMER INPUT COMPARE 2	S00E5 - S00E7
TIMER INPUT COMPARE 1	S00E8 - S00EA
REAL TIME INTERRUPT	S00EB - S00ED
IRQ	S00EE - S00F0
XIRQ	S00F1 - S00F3
SOFTWARE INTERRUPT (SWI)	S00F4 - S00F6
ILLEGAL OPCODE	S00F7 - S00F9
COMPUTER OPERATING PROPERLY (COP)	S00FA - S00FC
CLOCK MONITOR	S00FD - S00FF

CAPITULO No. 3

TARJETA DE ENTRADAS Y SALIDAS

3.1.- MAPA DE MEMORIA DE PERIFERICOS.

La tarjeta de entradas y salidas será en su mayoría manejada como localidades de memoria externa al microcontrolador, por lo que se ha asignado un bloque de memoria de 8 Kbytes dentro de los 64 Kbytes totales del mapa de memoria. La habilitación de la localidad de memoria asignada para manejo de periféricos externos se la realiza a través de la salida Y3 del demultiplexor (U6) de la tarjeta principal M68HC11EVB.

El mapa de memoria de los periféricos externos ubicados en la tarjeta de entradas y salidas se muestra en la tabla 3.1.

LOCALIDAD DE MEMORIA	ENTRADAS	SALIDAS
\$6000-\$63FF	TECLADO MATRICIAL	DISPLAY (LCD)
\$6400-\$67FF	TECLAS	CONVERSION D/A 1
\$6800-\$6BFF	ENTRADAS DIGITALES	SALIDAS DIGITALES
\$6C00-\$6FFF	NO ASIGNADAS	CONVERSION D/A 2
\$7000-\$7FFF	NO ASIGNADAS	NO ASIGNADAS

TABLA No. 3.1

MAPA DE MEMORIA DE PERIFERICOS EXTERNOS

3.2 DISEÑO DEL CIRCUITO DE CONVERSION DIGITAL - ANALOGA.

Se incluye dos sistemas de conversión D/A de similares características: A/D1 y A/D2.

Para los circuitos de conversión digital - analógica se emplea como elemento principal el convertor digital análogo: DAC0830LCN, por ser un elemento que se lo encuentra en el mercado local y por ser un convertor diseñado para trabajar directamente con los microprocesadores más comunmente utilizados, en este caso el MC68HC11.

Este convertor utiliza tecnología CMOS para bajo consumo de potencia y bajos errores de corrientes de salida, sin embargo un circuito especial permite la entrada de señales con niveles lógicos TTL. Un doble buffer permite a este convertor tener a la salida un voltaje correspondiente a una palabra digital, mientras se retiene la siguiente palabra digital.

Dado que este convertor tiene salidas de corriente se ha diseñado un circuito que permita tener salidas de voltaje que son más utilizadas. El convertor D/A internamente cuenta con una red de resistencias, la cual puede ser usada como una red de switcheo de voltaje, así el voltaje de referencia es conectado a uno de los terminales de salida de corriente, en este caso I_{OUT1} para tener el control directo de los bits de entrada al convertor, y el voltaje de salida es tomado del pín V_{REF} como se puede ver en la figura No. 3.1

. Dado que la impedancia de salida del convertor es un poco alta (10 a 20 $K\Omega$) y para acoplamiento de impedancias a la entrada, se usa el circuito integrado LM324, así se tiene una baja impedancia de salida del circuito de conversión D/A, como se puede ver en el circuito de la figura No. 3.2.

Con las conexiones en las líneas de habilitación ILE, XFER, CS, WR1 y WR2 (Ver "National Semiconductor Data Book", Capítulo No. 8) realizadas como se muestra en el circuito

de la figura No. 3.2, se define el trabajo del convertor en el modo de conversión continua, y el valor a la salida cambia únicamente cuando se tenga un cero lógico en las señales de CS y WR1.

Con $V_{DD} = 12 [V]$, $R_{20} = 1.2 K\Omega$ se da la corriente necesaria al diodo zener D_{z1} para su correcto funcionamiento, $I_z = V_{DD} / R_{20} = 10 [mA]$. El voltaje del diodo zener es $V_{z1} = 5.1 [V]$

Y dado que:

$$V_A = V_{z1}$$

Entonces $V_A = 5.1 [V]$ (Voltaje medido en la práctica $V_{z1} = 4.99 [V]$), y:

$$V_{OUT} = \frac{D * V_A}{256} \quad ;$$

Donde D varía de 0 a 255 en pasos de uno, de acuerdo a los valores lógicos en las entradas digitales.

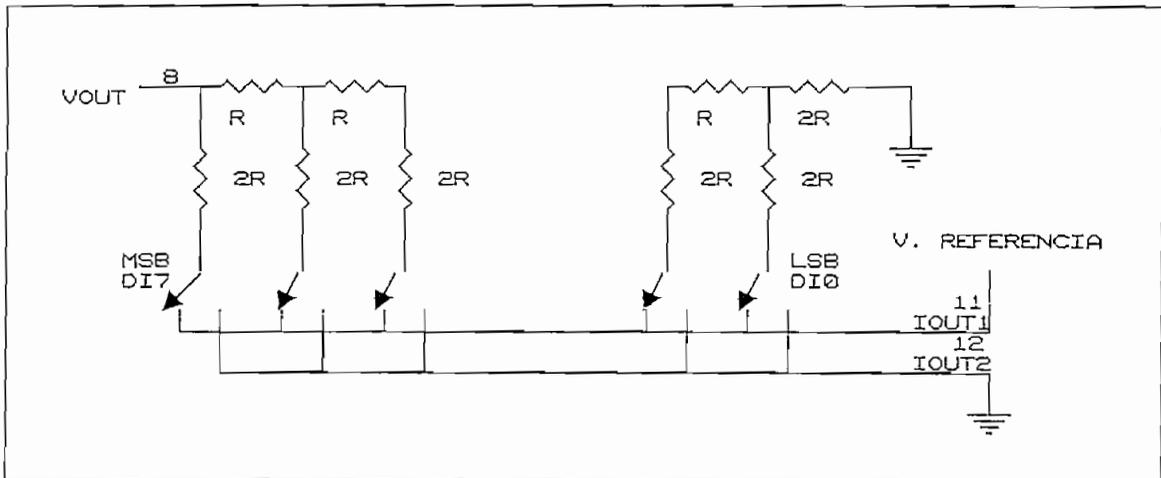


FIGURA No. 3.1

CIRCUITO INTERNO DEL CONVERTOR D/A

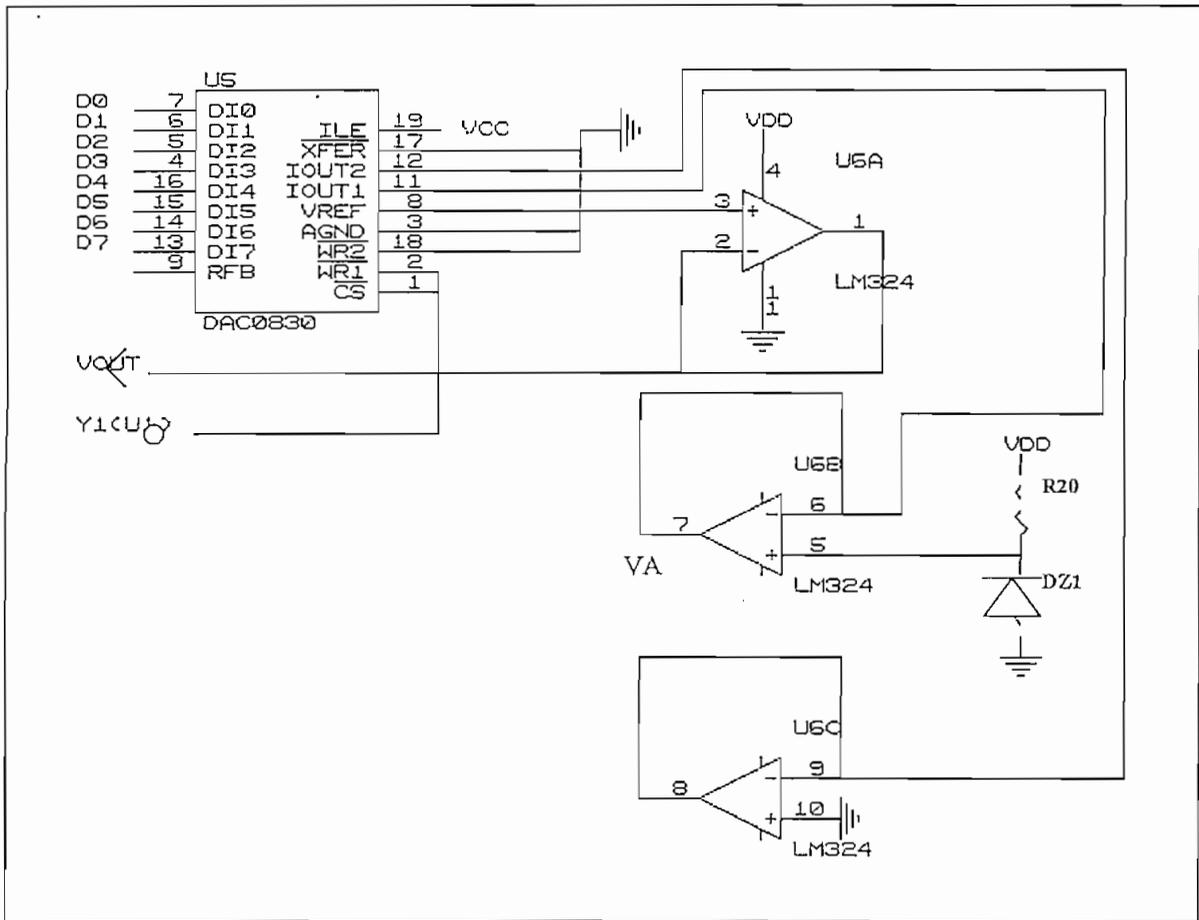


FIGURA No. 3.2

SISTEMA DE CONVERSION DIGITAL ANALOGO (A/D1 y A/D2)

3.3 CIRCUITO DE VISUALIZACION NUMERICA.

Este circuito está compuesto básicamente por una pantalla de cristal liquido de 4 dígitos hexadecimales y el manejador de display, y es tratado como localidad de memoria externa (localidades \$6000 - \$63FF).

El microcontrolador puede manejar los 4 displays del LCD de manera independiente, por lo que a su vez se ha establecido la siguiente asignación de dígitos y localidad de memoria correspondiente como se muestra en la tabla 3.2.

LOCALIDAD DE MEMORIA	DIGITOS
\$6000	UNIDAD
\$6001	DECENA
\$6002	CENTENA
\$6003	MIL

TABLA No. 3.2

ASIGNACION DE LOCALIDADES DE MEMORIA A DIGITOS DEL DISPLAY

El ICM7211M es un decodificador de 4 dígitos, con salidas a 7 segmentos y bits para direccionamiento de los dígitos, es controlado por las entradas de Chip Select para permitir un interfaz directo de alta velocidad con el microcontrolador MC68HC11 como se muestra en la figura No 3.3.

En la tabla 3.3 se muestra la entrada binaria y la correspondiente salida a siete segmentos.

CODIGO BINARIO (IN)				CODIGO HEXDECIMAL
B3	B2	B1	B0	(OUT)
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	E
1	1	1	1	F

TABLA No. 3.3

ENTRADAS Y SUS RESPECTIVAS SALIDAS EN EL DISPLAY

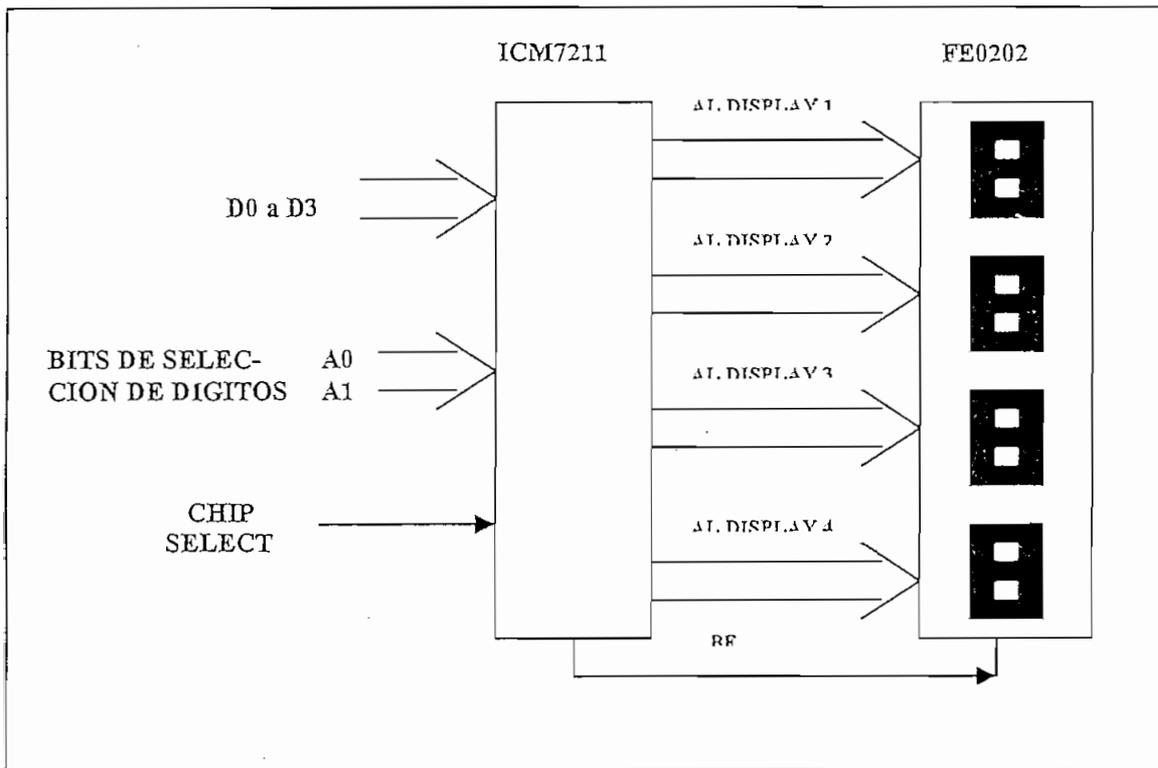


FIGURA No. 3.3

CIRCUITO DE VISUALIZACION NUMERICA

3.4.- CIRCUITO DE VISUALIZACION ALFANUMERICA.

Este circuito lo conforman: 1.) un display de cristal líquido que tiene 2 filas con 16 columnas cada una, lo que nos permite visualizar hasta 32 caracteres alfanuméricos; Internamente tiene el circuito de decodificación lo que da la posibilidad de ser accedido por el microcontrolador. 2.) Un potenciómetro que controla el nivel de contraste del display.

Las principales características del módulo LCD son:

- Ingreso de datos en formato de 4 (y 8) bits y 3 señales de control.

- Selección de ingreso de datos de izquierda a derecha y viceversa.
- Control de cursor.
- Set de caracteres ASCII extendido.
- Control del nivel de contraste.

El módulo de display LTN211 tiene dos registros de 8 bits:

Registro de instrucciones para almacenar la dirección del dato, como también para almacenar ciertas instrucciones propias como por ejemplo borrado del display, y Registro de datos, para almacenar los datos enviados por el microcontrolador.

El módulo LTN211 recibe el código del carácter desde el microprocesador y lo pasa a la RAM de datos del display, transforma cada código de carácter a una matriz patrón de caracteres y luego es mostrado este carácter en la pantalla de cristal líquido. Esto nos indica que para mostrar un carácter, la posición es enviada vía el bus de datos desde el microcontrolador al módulo LTN211 y luego es escrito en el registro de instrucciones. Un código de caracteres es luego enviado y escrito en el registro de datos del módulo.

Uno de los dos registros es seleccionado con la señal \overline{RS} .

Este módulo está provisto de una ROM generadora de caracteres, la cual puede generar hasta 160 códigos de caracteres patrones que tienen una correspondencia con los códigos de caracteres enviados por el microcontrolador.

Para interactuar con el módulo LTN211 existe la señal R/W, la cual permite escribir los códigos del carácter o a su vez acceder a los caracteres existentes en la ROM (0 = escritura; 1 = lectura).

Se tiene 4 líneas de orden alto (D4-D7) para entrada y salida de datos desde o hacia el microprocesador respectivamente.

Por lo anteriormente expuesto, utilizamos el p rtico B para manejar el m dulo LCD; cuya asignaci n de pines se muestra en la tabla No 3.4.

PB.7	PB.6	PB.5	PB.4	PB.3	PB.2	PB.1	PB.0
NC	E	RS	R/W	D7	D6	D5	D4
				D3	D2	D1	D0

TABLA No 3.4

DISPLAY ALFANUMERICO Y ASIGNACION DE PINES DEL PORTICO B

El voltaje de control de contraste es aplicado al terminal Vo. El panel del display es manejado por una diferencia de voltaje entre V_{DD} y Vo. En la figura No. 3.4 se muestran las conexiones respectivas, POT1 es ajustado hasta obtener el brillo m s adecuado de la pantalla LCD. R3 y R5 son resistencias que nos permiten proteger al elemento y mantener el nivel de brillo dentro de rangos aceptables de intensidad. El valor de  stas resistencias es 330 [ ].

Adem s es necesario indicar que para manejar este display alfanum rico se realizaron subrutinas en lenguaje ensamblador del MC68HC11 que son transparentes para el usuario y que facilitan su utilizaci n, se puede sacar mensajes, resultados de operaciones aritm ticas, etc. El tratamiento detallado de estas subrutinas se lo implementa en la pr ctica No 5 (CAPITULO No. 4).

Las funciones de estas subrutinas se muestran a continuaci n:

INITDIS: Inicializa el modo de transferencia de datos al display en cuatro bits, habilita las dos filas, encendido del cursor y habilita el incremento del cursor después de cada escritura.

CLEAR: Permite el borrado total del display.

MCURSOR: Permite posicionar el cursor en algún lugar del display.

SENDM: Permite escribir uno o varios caracteres a partir de la posición indicada por el cursor.

PRINTD: Permite escribir un número desde cero hasta nueve en la posición especificada por el cursor.

Las conexiones respectivas del circuito de visualización se muestran en la figura No 3.4

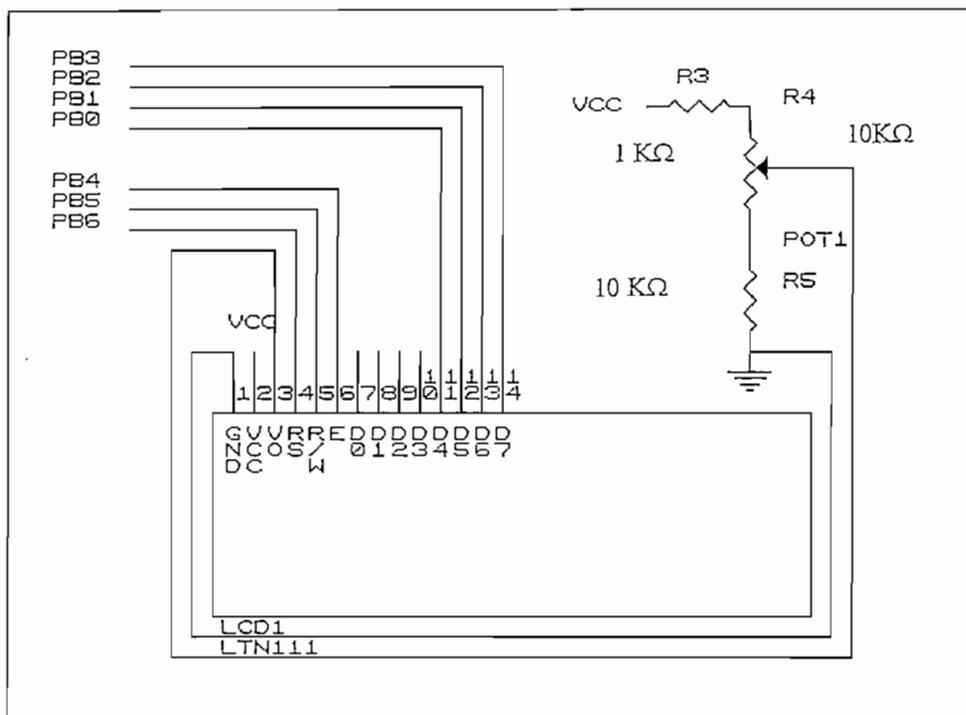


FIGURA No 3.4

CIRCUITO DE VISUALIZACION ALFANUMERICA

3.5.- CIRCUITO DE TECLADO MATRICIAL Y TECLAS

3.5.1.- CIRCUITO DE TECLADO MATRICIAL

El circuito para ingreso de datos al microcontrolador a través de teclado tiene como elemento principal el circuito integrado 74C922 codificador de teclado y un teclado matricial de

16 teclas. El codificador de teclado permite tener a las salidas el código binario de la tecla presionada, el cual se resume en la tabla 3.5. El pin DAV (Data Available) del decodificador de teclado sirve para dar la señal de interrupción al microcontrolador, para indicarle que se ha presionado una tecla. A la salida del decodificador de teclado se ha colocado un buffer el cual permitirá pasar el dato de la tecla presionada, solo cuando el microcontrolador envíe una señal de lectura a la dirección asignada al teclado matricial (\$6000 a \$63FF).

El circuito implementado se puede ver en la figura No. 3.5.

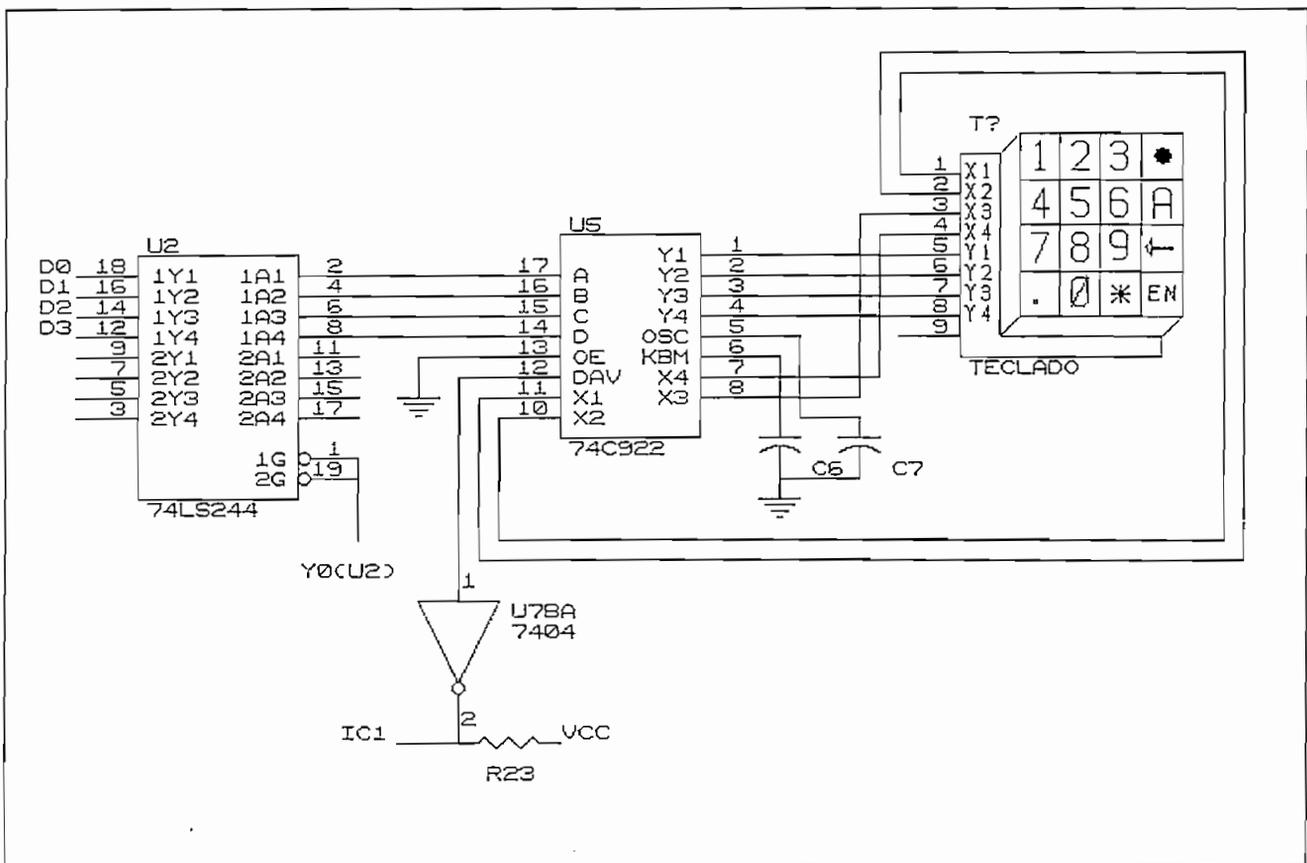


FIGURA No. 3.5

CIRCUITO DE INGRESO DE DATOS A TRAVES DE TECLADO

A la salida del pín DAV del decodificador de teclado, se ha puesto un inversor pues aquí se genera un uno lógico cuando se presiona una tecla y para generar la interrupción se necesita un cero lógico. Además para evitar perturbaciones debido al ruido que puede generar niveles de salida inestables se ha colocado la resistencia R23 de 10 K Ω a Vcc.

POSICION DE LA TECLA	SALIDA A	SALIDA B	SALIDA C	SALIDA D
0 Y1,X1	0	0	0	0
1 Y1,X2	1	0	0	0
2 Y1,X3	0	1	0	0
3 Y1,X4	1	1	0	0
4 Y2,X1	0	0	1	0
5 Y2,X2	1	0	1	0
6 Y2,X3	0	1	1	0
7 Y2,X4	1	1	1	1
8 Y3,X1	0	0	0	1
9 Y3,X2	1	0	0	1
10 Y3,X3	0	1	0	1
11 Y3,X4	1	1	0	1
12 Y4,X1	0	0	1	1
13 Y4,X2	1	0	1	1
14 Y4,X3	0	1	1	1
15 Y4,X4	1	1	1	1

TABLA No. 3.5

RELACION ENTRE TECLA PRESIONADA Y CODIGO DE SALIDA

De acuerdo a especificaciones de las hojas de datos del codificador, el capacitor C7 debe ser 10 veces el valor de C6 y de valores bajos, entonces se ha elegido $C6 = 0.01$ [uF] y $C7 = 0.1$ [uF].

3.5.2.- CIRCUITO DE TECLAS.

El circuito para ingreso de datos a través de teclas individuales como se puede ver en la figura número 3.6 consta de teclas simples que lo único que hacen es conectar a cero lógico la correspondiente línea del bus de datos cuando no se presiona la tecla y a uno lógico cuando se la presiona.

A la salida de las teclas se ha colocado un buffer el cual permitirá pasar el dato de la tecla presionada, solo cuando el microcontrolador envíe una señal de lectura a la dirección asignada a las teclas (\$6400 a \$67FF).

Los diodos D19 a D22 establecen una sola dirección de la corriente y evita que se pueda generar un cortocircuito entre la tecla presionada y cualquiera de las teclas no presionadas. Las resistencias R7 a R11 son de $2\text{ K}\Omega$ y sirven para limitar la corriente de ingreso al buffer 74LS244. A la salida de los diodos, se ha puesto un inversor pues aquí se genera un uno lógico cuando se presiona una tecla y para generar interrupción en el microcontrolador se necesita un cero lógico.

El código binario generado en el bus de datos al presionar las teclas es simple y se resume en la tabla número 3.6

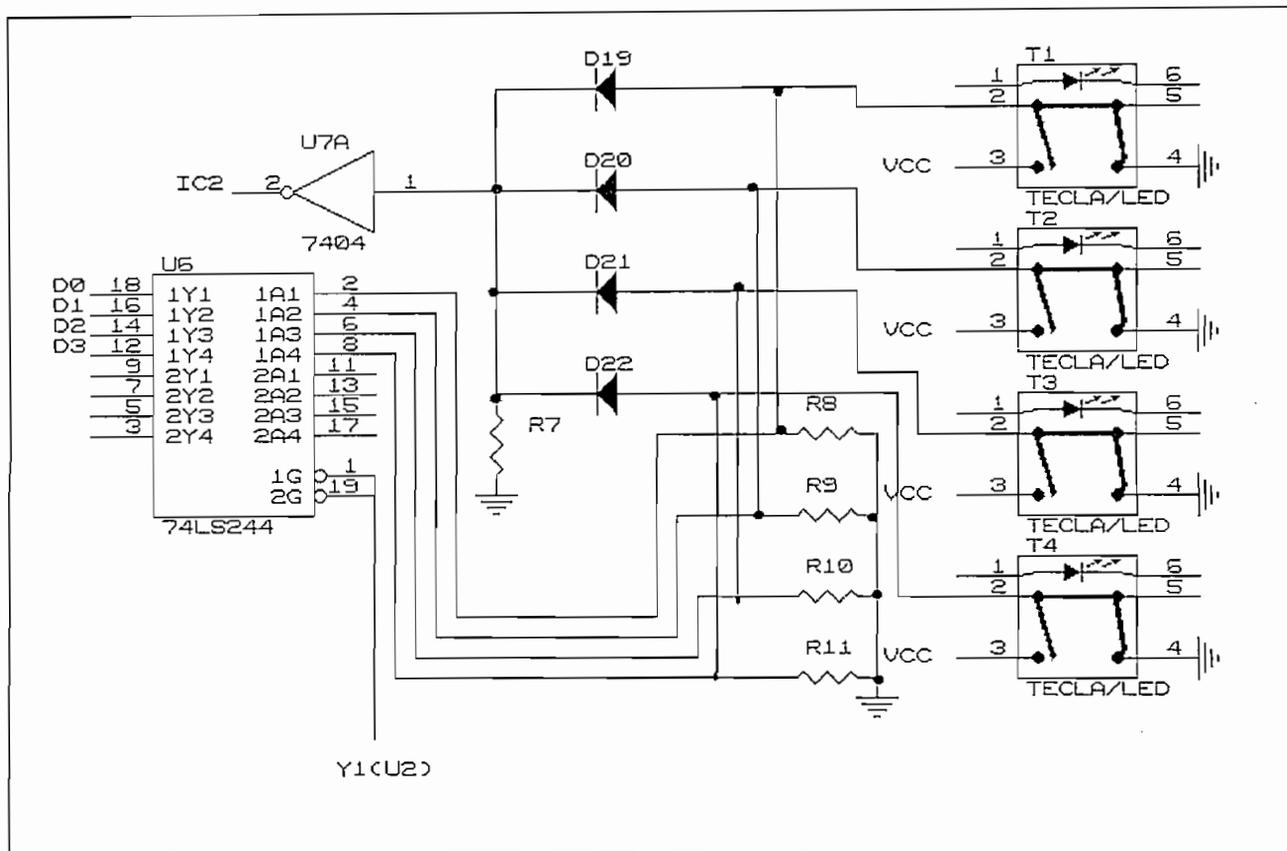


FIGURA No. 3.6

CIRCUITO DE ENTRADA DE DATOS A TRAVES DE TECLAS

TECLA PRESIONADA	D0	D1	D2	D3
TECLA 1	1	0	0	0
TECLA 2	0	1	0	0
TECLA 3	0	0	1	0
TECLA 4	0	0	0	1

TABLA No. 3.6

RELACION ENTRE TECLA PRESIONADA Y CODIGO GENERADO

3.6 CIRCUITOS DE ENTRADAS Y SALIDAS DIGITALES.

3.6.1 CIRCUITO DE ENTRADAS DIGITALES.

Este circuito dispone de 8 switches (S1-S8) de dos posiciones para el ingreso de señales digitales (0L o 1L). Es manejado como localidad de memoria externa (\$6000-\$63FF). Estos switches permiten el ingreso de un byte de datos, la entrada al bus de datos se lo realiza a través del buffer SN74LS244. Este elemento es un buffer octal con salida de tres estados lo que posibilita manejar direccionamiento de memorias y trabajar con buses de datos.

El nivel lógico aplicado a las entradas del buffer es 3.4 [V] (nivel TTL alto), puesto que además el circuito incluye LEDS indicadores de estado lógico que esta siendo ingresado. Se ha establecido un circuito para dar la corriente necesaria al diodo led, y que evita el consumo excesivo de corriente como se muestra en la figura No 3.7; donde el valor de las resistencias es:

$$R = (V_{DD} - V_{LED})/I_{LED}$$

$$R = (5V - 1.6V)/10mA$$

$$R = 340\Omega$$

El valor normalizado que más se aproxima es $R = 330 \Omega$, valor que toman las resistencias desde R12 hasta R19

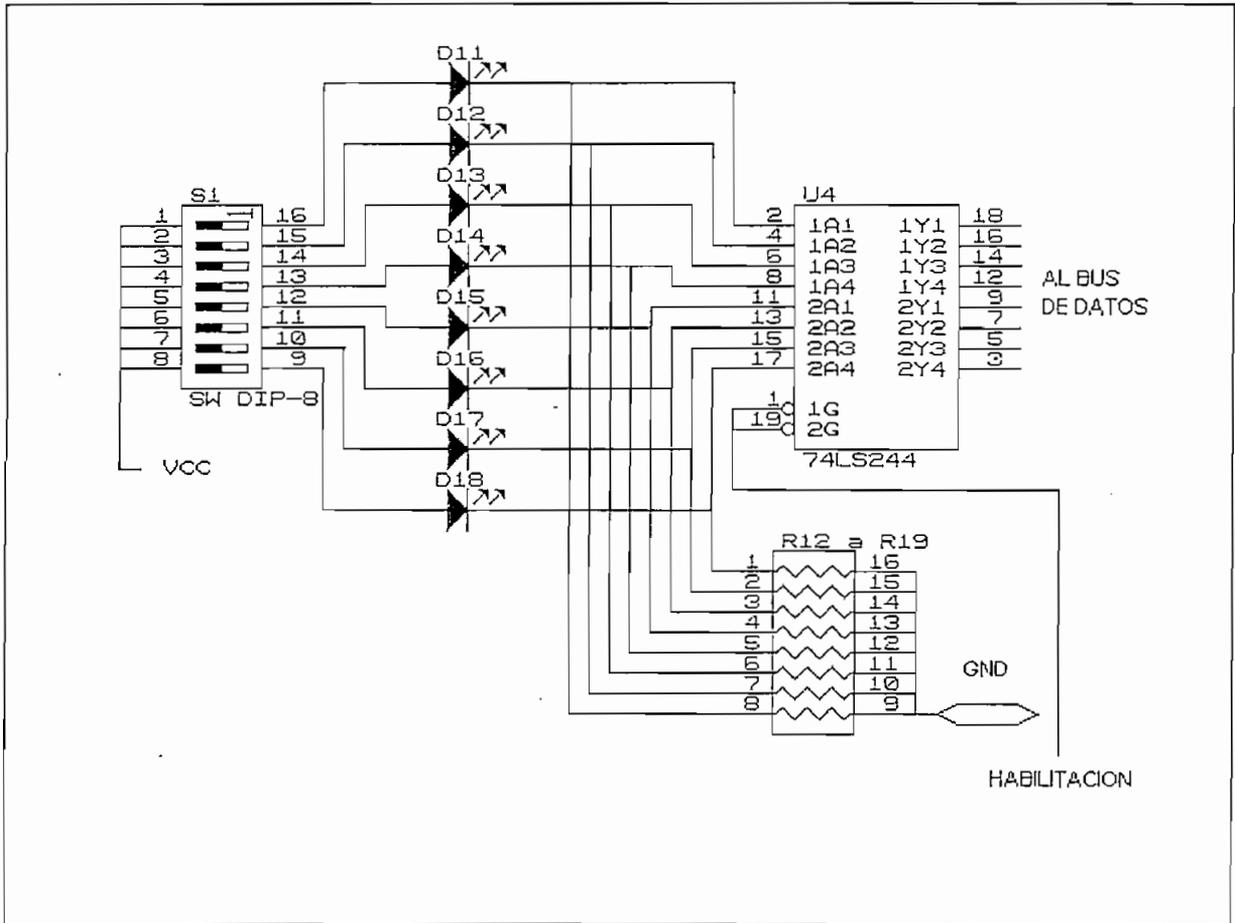


FIGURA N° 3.7

CIRCUITO DE ENTRADAS DIGITALES

3.6.2 CIRCUITO DE SALIDAS DIGITALES.

Este circuito está provisto de 8 salidas para conectores banana para obtener información en formato TTL desde el bus de datos del microcontrolador. La salida de datos se lo realiza a través del LATCH DM74LS373 que permite retener la información aplicada a este elemento. El circuito es manejado como localidad de memoria externa (\$6800-\$6C00). Este circuito es exclusivamente de salida, es decir solo escritura.

La figura No 3.8 muestra el circuito de salida de información.

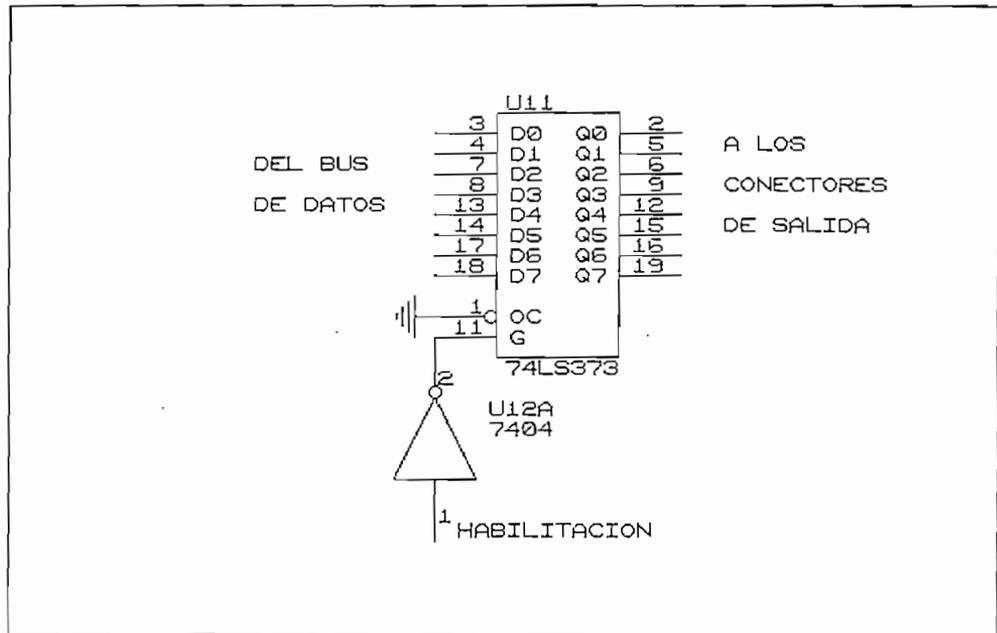


FIGURA No 3.8

CIRCUITO DE SALIDAS DIGITALES

Adicionalmente a este circuito se tiene el pórtico C conectado a ocho leds los cuales sirven para indicar el valor del dato que se ha escrito a este pórtico.

3.7.- DISEÑO DE LA TARJETA V&W – IN/OUT

La tarjeta V&W – IN/OUT ha sido construida con el criterio de presentar al usuario del módulo V&W – HC11 un acceso fácil y ordenado a todos los periféricos de entrada y salida, así como también a las líneas de entrada y salida de datos. Todas las entradas se encuentran al lado izquierdo de la tarjeta, mientras que las salidas se encuentran al lado derecho. Los conectores

para la interconexión con la tarjeta "M68HC11EVB" se encuentran en la parte superior e inferior, presentándose así uniformidad total en el diseño de la tarjeta.

Los diagramas del circuito impreso, y la disposición de los elementos en la tarjeta se lo puede ver en el apéndice D, mientras que las conexiones eléctricas de los elementos en la tarjeta lo encontramos en el apéndice E.

Dado que las dimensiones de la tarjeta son demasiado grandes, para evitar la alteración de los niveles lógicos TTL debido a las interferencias electromagnéticas y al fenómeno de Diafonía (que se presenta en las pistas de grandes longitudes dispuestas demasiado juntas en la tarjeta de circuito impreso, se ha colocado un sistema de filtrado a la entrada del display alfanumérico que es el caso más crítico, de modo que se filtren todas las señales mayores a 2 MHz que es la frecuencia de trabajo E-clock del sistema.

Para determinar con exactitud el valor del capacitor C conectado entre cada una de las líneas de entrada al display alfanumérico y la línea de tierra (Ver figura No. 3.8) se necesita el valor de las resistencias de las pistas del circuito impreso involucradas, lo cual es muy difícil de determinar y no es motivo de análisis en este trabajo, por lo que se recurre a realizar pruebas con el equipo, con lo que se consigue filtrar el ruido colocando un capacitor C igual a $0.01[\mu\text{F}]$ en cada línea.

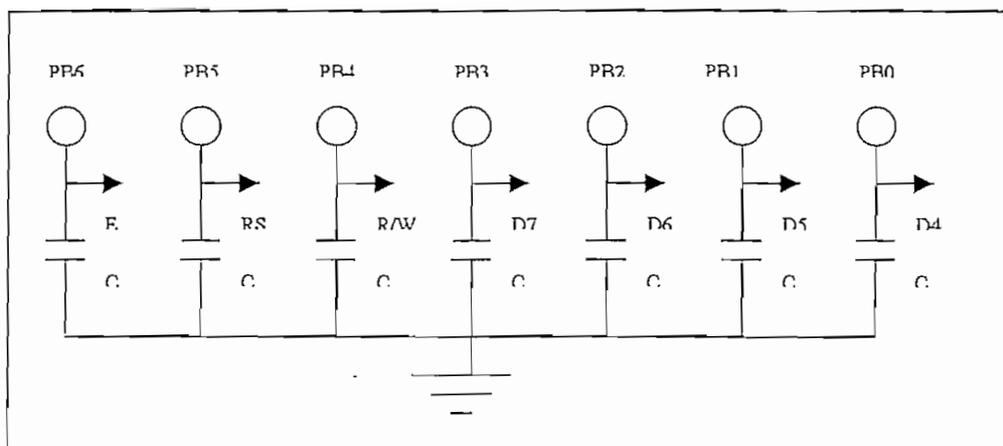


FIGURA No. 3.8

FILTRO DE ENTRADA A DISPLAY ALFANUMERICO

3.8.- ESPECIFICACIONES DEL MODULO V&W – HC11

3.8.1.- DESCRIPCION GENERAL

Este módulo ha sido construido de modo que presente todas las facilidades para el usuario, en cuanto a conexiones externas para entrada y salida de datos y también presenta uniformidad en su presentación de salida. En la parte superior se coloca una placa acrílica, en la cual constan las identificaciones de todos y cada uno de los periféricos, así como también de las entradas y salidas. Dentro del mismo se ha interconectado de manera compacta y tratando de ocupar el menor espacio posible, tres tarjetas como son la tarjeta principal 'M68HC11EVB', la tarjeta de entradas y salidas V&W – IN/OUT, y la tarjeta de fuente la cual entrega la energía necesaria a todo el sistema.

El equipo posee dispositivos de visualización numérico y alfanumérico, como también un teclado matricial y teclas para el ingreso de datos. Además se dispone de dos puertos de

comunicación serial lo cual permite simplificar el trabajo. El p3rtico terminal (P2) se utiliza para cargar los programas en memoria RAM y el p3rtico host (P3) para la comunicaci3n de los programas en lenguaje Quick-Basic y ensamblador correspondientes a las pr3cticas demostrativas descritas en el cap3tulo cuatro.

Para la reinicializaci3n del sistema se cuenta con un sistema de RESET, cuya acci3n se genera a trav3s del pulsador de color rojo (Pulsador de RESET)

3.8.2 ESPECIFICACIONES DEL EQUIPO.

Microcontrolador: MC68HC11A1FN

Reloj del Sistema: El cristal de 8 MHz con el que cuenta el m3dulo, permite tener un reloj del sistema de 2 MHz..

Convertidores: - Sistema de conversi3n an3logo digital, el cual permite el ingreso multiplexado en el tiempo de ocho canales de entrada.

- Dos sistemas de conversi3n digital an3logo basados en el circuito integrado DAC0830LCN conectados de modo que trabajen en el modo de conversi3n continua.

Memorias: - EEPROM de 512 Bytes

- EPROM de 8 Kbytes donde se encuentra grabado el programa Monitor
- RAM interna de 256 Bytes y externa de 8 Kbytes.
- ROM deshabilitada internamente.

Interfaces: - Dos interfaces de comunicaci3n serial RS-232.

- Un interfaz de comunicaci3n serial s3ncronica (SPI).

Pórticos: - Pórtico PA, conectado para trabajar con el sistema de temporización y sus interrupciones.

- Pórticos PB y PC utilizados como buses de datos y direcciones para el sistema, pero recuperados por el amplificador de pórticos MC68HC24FN (PRU), el pórtico B es utilizado para manejar el display alfanumérico, y el pórtico C es utilizado para encendido y apagado de leds, de acuerdo al estado del pórtico.

-Pórtico D, utilizado para transmisión serial sincrónica y asincrónica, y también como líneas de entrada y salida de propósito general.

-Pórtico E, utilizado como entradas de datos analógicos y digitales.

3.8.3.- FUENTE DE ALIMENTACION

Para la alimentación de las tarjetas se cuenta con una fuente del tipo conmutada de DC la cual provee niveles de voltaje (+12 V, -12 V, +5V), necesarios para la polarización y referencias de los elementos de las mismas.

El voltaje de entrada a la tarjeta de la fuente de alimentación puede ser conmutado por un switch colocado en la estructura metálica del gabinete para permitir el ingreso de voltajes de 220 o 110 VAC.

3.8.4.- OTRAS CARACTERISTICAS.

- Caja metálica con ventilador y aberturas para disipación de calor
- Tapa de acrílico para identificación de entradas, salidas y periféricos.

- Conectores: cable plano de 60 hilos con sus respectivos conectores hembra a los extremos, cable plano de 28 hilos conectado a un adaptador de socket al un extremo y un conector hembra al otro extremo.
- Dimensiones: Largo = 31 cm.
Ancho = 22 cm.
Altura = 12 cm.
- Rango de Temperatura: 0 a 55 °C.

CAPITULO No. 4

DESARROLLO DE PRACTICAS

PRACTICA No. 1

TEMA: FAMILIARIZACION CON EL EQUIPO

1.- **Objetivo.-** Proporcionar los conocimientos necesarios a cerca del módulo "V&W-HC11", para que el estudiante adquiera destreza y habilidad con el equipo para su fácil manejo en futuras prácticas.

2.- Teoría.

2.1 .- Descripción del Hardware

Existen en el modulo "V&W_HC11" dos tarjetas interconectadas entre sí:

- a.- "M68HC11EVB Evaluation Board"
- b.- V&W-IN/OUT Tarjeta de entradas y salidas

(Información en detalle a cerca de estas tarjetas se las encuentra en los capítulos No. 2 y No. 3 respectivamente.)

En la figura No. P1 se puede observar la conexión entre el módulo "V&W-HC11" y un computador personal

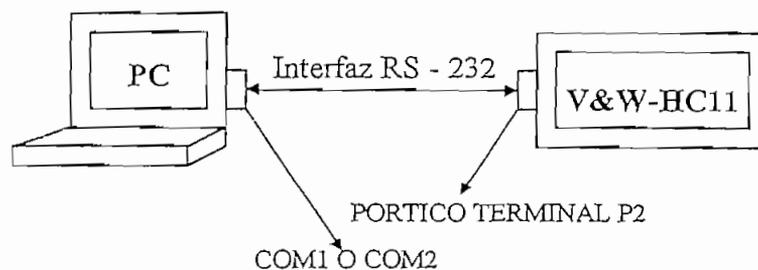


FIGURA No. P1

2.2 .- MAPEO DE MEMORIA

El módulo "V&W-HC11" tiene capacidad de manejar un total de 64 Kbytes de memoria, la cual se encuentra distribuida de la siguiente manera:

RAM DE USUARIO	\$0000 - \$0035
STACK POINTER	\$0036 - \$004A
VARIABLES DEL PROGRAMA MONITOR	\$004B - \$00C3
TABLA DE VECTORES DE SALTO	\$00C4 - \$00FF
PRU Y REGISTROS DE DATOS Y CONTROL	\$1000 - \$17FF
NO USADO	\$1800 - \$3FFF
FLIP-FLOP DECODIFICADOR	\$4000 - \$5FFF
PERIFERICOS DE ENTRADA – SALIDA	\$6000 - \$7FFF
NOT USED	\$8000 - \$97FF
TERMINAL ACIA	\$9800 - \$9FFF
NO USADO	\$A000 - \$B5FF
EEPROM	\$B600 - \$B7FF
NO USADO	\$B800 - \$BFFF
RAM DE USUARIO	\$C000 - \$DFFF
EPROM PARA EL PROGRAMA MONITOR	\$E000 - \$FFFF

2.3.- Ensamblaje y Depuración de Programas

Existen dos métodos de ensamblar un programa:

a.- Ensamblaje de código usando el ensamblador en línea del programa monitor BUFFALO para lo cual se dispone de la RAM de usuario (\$C000 - \$DFFF) y el programa de interfaz Kermit para establecer comunicación entre el Módulo y un Computador personal.

Los pasos a seguirse son los siguientes:

- 1) C:\HC11\KERMIT ↵ (ejecución del programa Kermit)
- 2) KERMIT-MS>SET PORT 1 (2) ↵ (establece el pÓrtico de comunicaci3n)
- 3) KERMIT-MS>SET BAUD 9600 ↵ (establece velocidad de transmisi3n)
- 4) KERMIT-MS>CONNECT ↵ (inicia comunicaci3n PC - BUFFALO)
- 5) ↵ (aparece indicador > del programa monitor BUFFALO)
- 6) > ASM C000 ↵ (instrucci3n para ingresar al modo ensamblador en lÍnea)
- 7) > INSTRUCCIONES DEL PROGRAMA ↵
- 8) >CTRL A (salir del modo ensamblador)
- 9) >G C000 ↵ (ejecuci3n del programa desde la direcci3n especificada. Para restablecer comunicaci3n PC - M3dulo se debe presionar el bot3n de RESET)
- 10) > CTRL] C (salir al programa Kermit)
- 11) KERMIT-MS>EXIT (abandonar el programa Kermit)

b.- Ensamblaje de c3digo en alg3n editor de texto y descarga serial al m3dulo vÍa conector P2 (TERMINAL I/O PORT)

Los pasos a seguirse son los siguientes:

- 1) Creamos el programa en cualquier editor de texto
- 2) C:\HC11>AS11 C:\PATH\nombre.EXT ↵ (Ensamblaje del programa. Al ejecutar este comando se crea el archivo nombre. S19 en el directorio actual de trabajo)
- 3) C:\HC11>AS11 C:\PATH\nombre.EXT ↵ -L> C:\PATH\nombre.LST (Crea el archivo de listado de programa nombre.LST)
- 4) C:\HC11\KERMIT ↵ (ejecuci3n del programa Kermit)
- 5) KERMIT-MS>SET PORT 1 (2) ↵ (establece el pÓrtico de comunicaci3n)
- 6) KERMIT-MS>SET BAUD 9600 ↵ (establece velocidad de transmisi3n)
- 7) KERMIT-MS>CONNECT ↵ (inicia comunicaci3n PC - BUFFALO)
- 8) ↵ (aparece indicador > del programa monitor BUFFALO)
- 9) LOAD T (habilita al m3dulo para la descarga del archivo nombre.S19 desde el PC. El programa no responde con ning3n mensaje)
- 10) CTRL] C (retorno temporal al programa Kermit)

- 11) KERMIT-MS>TYPE C:\PATH\nombre.S19 >COM1 (Ó COM2)␣ (descarga del archivo ensamblado)
- 12) KERMIT-MS>CONNECT␣ (retorna al programa monitor BUFFALO para la depuración del programa descargado)
- 13) >G C000␣ (ejecución del programa desde la dirección especificada. Para restablecer comunicación PC - Módulo se debe presionar el botón de RESET)
- 14) > CTRL] C (salir al programa Kermit)
- 15) KERMIT-MS>EXIT (abandonar el programa Kermit)

3.- TRABAJO PREPARATORIO

- 3.1.- Consultar el funcionamiento de las tarjetas del módulo “V&W-HC11”
- 3.2.- Consultar las instrucciones del microcontrolador MC68HC11

4.- PARTE PRACTICA

- 4.1.- Siguiendo los pasos indicados en el numeral 2.3.a desarrollar los siguientes programas de aplicación:
 - 4.1.1.- Realizar un programa que realice el encendido secuencial de los LEDS, uno a la vez y en forma repetitiva.
 - 4.1.2.- Realizar un programa que reproduzca en los LEDS los estados de los DIP_SWITCH conectados al pÓrtico C (dirección \$1003)
- 4.2.- Siguiendo los pasos realizados en el numeral 2.3.b desarrollar los siguientes programas de aplicación:
 - 4.2.1.- Escribir un programa que realice un contador que a partir de cero cuente indefinidamente de uno en uno y en forma ascendente. El valor del contador deberá aparecer en el display numérico.
NOTA.- El microcontrolador MC68HC11 trata al display numérico como un conjunto de 4 localidades de memoria RAM ubicadas a partir de la localidad \$6000.
 - 4.2.2.- Realizar un programa que luego de un retardo de 4 segundos muestre en el display numérico el valor hexadecimal del estado de los DIP_SWITCH.

PRACTICA # 2

TEMA : UTILIZACIÓN DEL PROGRAMA MONITOR BUFFALO

1.- **Objetivo.**- Presentar al estudiante los diferentes recursos y facilidades que dispone el programa monitor BUFFALO .

2.- Teoría.

El programa monitor BUFFALO (Bit User Fast Friendly Aid to Logical Operation) se encuentra ubicado en la memoria EPROM externa al microcontrolador MC68HC11A1 , en las localidades (\$E000-\$FFFF).

Este programa contiene varias subrutinas para realizar tareas de entrada y salida de datos, las cuales serán para el estudiante totalmente transparentes. Sin embargo mayor información sobre estas se encuentra se encuentra en el cap.3 del manual de usuario "M68HC11EVB Evaluation Board".

Comandos del Programa Monitor.

ASM <dirección> .- Permite ingresar al modo ensamblador/editor interactivo, <dirección> es la dirección de inicio de la operación de ensamblado (sino se especifica la dirección, la operación de ensamblado se la realiza en la RAM interna.).

BF <dirección1> <dirección2> <dato> .- Permite llenar un bloque de memoria con un byte específico.<dirección1> dirección de inicio del bloque, <dirección2> dirección de fin de bloque, <dato> valor Hexadecimal a ser llenado.

BR <dirección> .- Permite ubicar puntos de parada en la ejecución de un programa .

BULK .- Permite borrar todo el contenido de la memoria EEPROM interna del microcontrolador (\$B600-\$B7FF).

CALL <dirección>.- Permite ejecutar una subrutina , <dirección> es la dirección donde inicia la subrutina .

G <dirección> .- Permite la ejecución de un programa , <dirección> es la dirección donde inicia la ejecución del programa.

HELP .- Despliega los comandos del programa monitor.

LOAD T .- Permite descargar archivos en formato S-Records (archivos ensamblados con formato nombre.S19), vía el pórtico terminal (P2).

MD <dirección1> <dirección2> .- Despliega en la pantalla el contenido de un bloque de memoria. <dirección1> dirección de inicio del bloque de memoria <dirección2> dirección final del bloque.

MM <dirección> .- Permite examinar y modificar el contenido de localidades de memoria <dirección> Es la localidad de memoria donde se inicia la modificación del contenido de la misma. **Tener precaución al modificar los registros internos del microcontrolador.**

MOVE <dirección1> <dirección2> <destino>.- Permite copiar los contenidos de un bloque de memoria a otro bloque de memoria . Este comando es útil para programar la memoria EEPROM interna del microcontrolador.<dirección1> dirección de inicio del bloque <dirección2> dirección final del bloque <destino> dirección de inicio donde se mueve el bloque.

P .- Permite continuar la ejecución de un programa luego de un comando BR (Break point).

RM.- Permite modificar el contenido de los registros: Contador de Programa (P) , Registro Índice (Y), Registro Índice (X), Acumulador (A), Acumulador (D) y Stack Pointer.

T <n>.- Permite la ejecución de un programa paso-paso <n> es el número en Hexadecimal de instrucciones a ejecutar.

3.- Trabajo preparatorio.

Consultar como el programa monitor BUFFALO realiza la asignación de RAM, ROM, registros internos y en qué localidades de memoria quedan éstos ubicados.

4.- Parte Práctica.

4.1.- Se realizará un ejemplo de como cargar un programa en la memoria EEPROM interna (\$B600-\$BFFF) del microcontrolador. El programa debe realizar lo mismo que se especifica en el numeral 4.2.b. Para lo cual se sugiere seguir los siguientes pasos:

a.- Cargar el archivo PRACTI21.S19 al módulo “V&W-HC11”.

b.- Ingresar al programa monitor BUFFALO y cambiar en las instrucciones de salto los valores de \$C0 por B6 utilizando el comando MM (modificar memoria).

c.- Utilizando el comando MOVE cargar el programa a la memoria EEPROM interna del microcontrolador.

d.- Para ejecutar el programa mantener presionado el botón RESET y cambiar la posición del jumper J4 .

4.2.- Se realizará un programa que cumpla con lo siguiente:

a.- Al presionar la tecla 1 se debe realizar el encendido secuencial de los LEDS (numeral 4.1.1 de la práctica anterior) .

b.- Al presionar la tecla 2 se debe reproducir en los LEDES el estado de los DIP-SWITCHS (numeral 4.1.2 de la práctica anterior).

c.- Al presionar la tecla 3 se debe implementar un contador que a partir de cero cuente indefinidamente de 1 en 1 y en forma ascendente. El valor del contador deberá aparecer en el display (numeral 4.2.1 de la práctica anterior).

d.- Al presionar la tecla 4 implementar un retardo de dos segundos, luego del cual se presente en display el número indicado por el estado de los DIP-SWITCHS.

PRACTICA No. 3

TEMA: INTERRUPCIONES, CONTADORES Y TEMPORIZADORES

1.- **Objetivo.**- Familiarizar al estudiante con las distintas opciones tanto de timers como de interrupciones que presenta el microcontrolador MC68HC11

2.- Teoría.

2.1 Interrupciones.

El microcontrolador MC68HC11 cuenta con veinte diferentes interrupciones, y cualquiera de ellas puede ser definida como la interrupción de más alta prioridad (cargar el registro HPRIO para definir la interrupción de más alta prioridad).

VECTOR DE INTERRIIPCION	LOCALIDAD DE MEMORIA
SERIAL COMMUNICATIONS INTERFACE (SCI)	S 00C4 - S00C6
SERIAL PERIPHERAL INTERFACE (SPI)	S 00C7 - S00C9
PULSE ACCUMULADOR INPUT EDGE	S 00CA - S00CC
PULSE ACCUMULATOR OVERFLOW	S 00CD - S00CF
TIMER OVERFLOW	S00D0 - S00D2
TIMER OUPUT COMPARE 5	S00D3 - S00D5
TIMER OUPUT COMPARE 4	S00D6 - S00D8
TIMER OUPUT COMPARE 3	S00D9 - S00DB
TIMER OUPUT COMPARE 2	S00DC - S00DE
TIMER OUPUT COMPARE 1	S00DF - S00E1
TIMER INPUT COMPARE 3	S00E2 - S00E4
TIMER INPUT COMPARE 2	S00E5 - S00E7
TIMER INPUT COMPARE 1	S00E8 - S00EA
REAL TIME INTERRUPT	S00EB - S00ED
IRQ	S00EE - S00F0
XIRQ	S00F1 - S00F3
SOFTWARE INTERRUPT (SWI)	S00F4 - S00F6
ILLEGAL OPCODE	S00F7 - S00F9
COMPUTER OPERATING PROPERLY (COP)	S00FA - S00FC
CLOCK MONITOR	S00FD - S00FF

FIGURA No.P3

La tabla de vectores de interrupción (ver figura No. P3) se encuentra ubicada en las localidades (\$00C4 a \$00FF), y a cada vector se le asigna un campo de tres bytes los cuales son usados para realizar un salto a la subrutina de servicio de atención a la interrupción respectiva.

2.2 CONTADORES Y TEMPORIZADORES (TIMER/COUNTERS)

El sistema de temporización (timer) se basa en un contador de 16 bits de corrido libre, el cual es manejado por la señal de reloj E del microcontrolador ($E = 2\text{Mhz.}$ para un cristal de 8 Mhz.). El sistema cuenta con acciones de entrada y salida.

Las acciones de entrada (input captures) graban el valor del contador de corrido libre en los registros de captura de entrada TICx cuando se detecta un cambio en el respectiva línea de entrada (pines 0,1 y 2 del pórtico A).

Las acciones de salida (ouput compares) generan una acción de salida en los pines 3, 4, 5, 6 y 7 del pórtico A, cuando el valor del respectivo registro TOCx es igual al valor del registro del contador de corrido libre TCNT.

Además el sistema cuenta con un acumulador de pulsos, el cual es un contador de 8 bits y puede trabajar en los modos de contador de eventos externos o acumulador de tiempo. En el modo de contador de eventos externos, el contador de 8 bits es manejado por la acción del pin externo PAI (pin 7 del pórtico A), la máxima frecuencia que se puede trabajar en este modo es de $E/2$ (1 Mhz). En el modo acumulador de tiempo, el contador de 8 bits está incrementando libremente su valor a una frecuencia de $E/64$ (31.25 Khz) pero solo mientras haya una señal de entrada en el pin PAI.

Todas las acciones tanto de temporizadores como de contadores cuentan con su respectiva acción de interrupción. Estas interrupciones cuentan con sus respectivos registros de control, los cuales sirven para habilitar y deshabilitar a las mismas.

3.- Trabajo Preparatorio.

Consultar las características de los registros de control tanto de interrupciones como de temporizadores y contadores.

4.- Parte Práctica.

Se implementará un sistema que cumpla con lo siguiente:

- Utilizando un optoacoplador y una rueda con 16 huecos se desea poder medir la velocidad de giro del motor en un rango de 0 a 4000 RPM con una resolución de 10 RPM. Para el efecto la señal externa generada por el optoacoplador se conecta al pin de entrada IC3 del módulo "V&W-HC11". La base de tiempo será generada utilizando una interrupción generada por la acción de un "output compare".

- El valor de la velocidad deberá ser codificado a BCD y mostrado en el display de cristal líquido.

PRACTICA No. 4

TEMA: INTERFAZ DE COMUNICACIÓN SERIAL RS-232

1.- Objetivo.

Familiarizar al estudiante con el uso del p rtico de comunicaci n serial SCI del microcontrolador MC68HC11 y el manejo del p rtico serial de un computador personal utilizando el programa Quick Basic.

2.- Teor a.

El SCI es una interface de comunicaci n serial Full-D plex el cual trabaja con formato est ndar NRZ (un bit de inicio, 8 o 9 bits de datos, y un bit de parada) y una variedad de baud rates. El receptor y el transmisor del SCI son funcionalmente independientes, pero usan el mismo formato de datos y baud rates.

Hay cinco registros usados para definir las condiciones de trabajo del SCI:

1) SCDR (Registro de Datos).- Usado para el almacenamiento de los datos recibidos o transmitidos por el MCU.

2) SCR1 (Registro 1 de Control de Comunicaci n Serial).- En el cual se define la longitud del dato y el m todo utilizado para el reconocimiento de inicio de recepci n.

3) SCR2 (Registro 2 de Control de Comunicaci n Serial).- Habilitan o deshabitan las funciones individuales de transmisi n serial del SCI.

4) SCSR (Registro de Estado de Comunicaci n Serial).- Usado para definir las habilitaciones para la generaci n de las diferentes interrupciones de transmisi n serial.

5) BAUD (Registro de Selecci n de Velocidad de Comunicaci n).- Usado para definir diferentes valores de velocidad para transmisi n y recepci n de datos.

*Para mayor informaci n acerca del SCI y de los registros de datos y control del mismo, referirse al libro "M68HC11 Reference Manual" de Motorola.

3.- Trabajo Preparatorio.

Consultar la configuración de los diferentes registros que definen la comunicación serial SCI del microcontrolador MC68HC11.

Consultar los comandos del Quick Basic que permiten manejar el puerto de comunicación serial del computador.

4.- Parte Práctica.

4.1.-Implementando el mismo circuito de la práctica anterior, añadir al programa las instrucciones y rutinas que permitan realizar el enlace con el computador utilizando el pórtico Host del módulo "V&W-HC11" y que cumpla con las siguientes condiciones:

- Transmitir al PC el valor de la velocidad.
- Velocidad de transmisión Serial: 9600 bps
- Formato de Transmisión: 1 bit de inicio + 8 bits de datos y uno de parada.

4.2.- Elaborar un programa en Quick Basic que permita establecer la comunicación serial entre el PC y el microcontrolador y muestre en pantalla el valor de la velocidad de rotación del motor.

PRACTICA No. 5

TEMA: MANEJO DE DISPLAY ALFANUMERICO

1.- Objetivo.

Familiarizar al estudiante con el manejo del display alfanumérico LTN211.

2.- Teoria.

El display alfanumérico nos da la posibilidad de visualizar 32 caracteres en 2 filas de 16 caracteres, incluye decodificador y circuitos adicionales para ser accedidos por medio del microcontrolador.

Las características del display LTN211 son:

- Ingreso de datos en formato de cuatro u ocho bits, con tres señales de control.
- Selección de ingreso de datos de izquierda a derecha o viceversa.
- Control de cursor
- Set de caracteres ASCII extendido
- Registro interno de 80 localidades
- Control de nivel de contraste.

Para facilitar el manejo de este display se han elaborado las siguientes subrutinas:

INITDISC: Inicializa la transferencia de cuatro bits, habilitación de las dos filas, encendido del cursor e incrementación del cursor.

CLEAR: Permite el borrado total del display

MCURSOR: Permite posicionar el cursor en cualquier posición.

SENDM: Permite escribir uno o varios caracteres a partir de la posición indicada por el cursor.

PRINTD: Permite escribir un número de cero a nueve en la posición especificada por el cursor.

3.- Trabajo Preparatorio.

Consultar el manual de usuario del display LTN211 y analizar el funcionamiento del mismo.

4.- Parte Práctica.

Utilizando las rutinas de manejo del display, desarrollar un programa en lenguaje ensamblador que permita mostrar en el display alfanumérico el contenido de un registro bajo el siguiente formato:

DATO = NNN

Donde NNN puede variar entre cero y doscientos cincuenta y cinco en pasos de uno o de cinco de acuerdo a la tecla presionada.

- Tecla 1: Incrementa en uno
- Tecla 2: Decrementa en uno
- Tecla 3: Incrementa en cinco
- Tecla 4: Decrementa en cinco

PRACTICA No. 6

TEMA: UTILIZACION DE CONVERSORES ANALOGO DIGITALES

1.- Objetivo.

Familiarizar al estudiante con la utilización de las entradas de conversión análogo-digitales del Microcontrolador MC68HC11.

2.- Teoria.

El MC68HC11 incluye 8 canales de conversión análogo-digital de entradas multiplexadas, con un sistema de muestreo y retención para minimizar los errores de conversión causados por los cambios rápidos de las señales de entrada. Dos líneas, VRL y VRH son dadas para las entradas del voltaje de referencia.. Esos pines pueden ser conectados a una fuente de energía separada o aislada para asegurar mayor precisión del conversor A/D, sin embargo en el módulo V&W – HC11, estas líneas están conectadas a GND y VCC respectivamente. Cada conversión se realiza en un tiempo de 32 ciclos de reloj (E-clock).

Un voltaje de entrada igual a VRL es convertido a \$00 y voltaje de entrada igual a VRH es convertido a \$FF. Un multiplexor permite al conversor A/D seleccionar una de las 16 señales análogas, 8 de las cuales corresponden a las líneas de entrada del pòrtico E del microcontrolador.

Los registros usados por el conversor son:

ADCTL .- Utilizado para el control del conversor A/D.

ADR1,ADR2,ADR3 y ADR4.- Utilizado para el almacenamiento de los resultados de las conversiones.

3.- Trabajo preparatorio.

Consultar la configuración de los diferentes registros que definen la conversión análogo-digital del microcontrolador MC68HC11.

Consultar los comandos del Quick Basic que permiten manejar gráficos.

4.- Parte Práctica.

Realizar un programa para el microcontrolador y otro para el Quick Basic que permitan implementar un sistema de adquisición de datos que cumplan las siguientes funciones:

- Medir la velocidad del Motomatic utilizando el taco generador acoplado al motor. El microcontrolador recibirá el dato correspondiente a la velocidad a través del pin PE1 del pòrtico E (entrada conversor A/D) y deberá mostrar en la pantalla del display alfanumérico la velocidad en el siguiente formato:

VEL = NNNN (rpm); NNNN (0-4000)

- El teclado matricial del equipo permitirá cambiar las unidades:

Tecla 1: rpm

Tecla 2: rad/seg.

- En el computador se recibe cada medición y se grafica en pantalla los datos a medida que van llegando.

PRACTICA No. 7

TEMA: SINTESIS DE SEÑALES DIGITALES.

1.- Objetivo.

Familiarizar al estudiante con la utilización de conversores y las síntesis de señales analógicas a partir de técnicas digitales.

2.- Teoría.

Los circuitos de la figura número P7 nos permite realizar de manera simple las síntesis de una forma de onda deseada. En el primer circuito la generación de la señal analoga se la realiza utilizando el conversor D/A (DAC0830LCN), mientras que en el segundo circuito se utiliza una salida digital del módulo V&W-HC11 para generar la señal PWM (modulación por ancho de pulso), y a continuación se pone un filtro analogo.

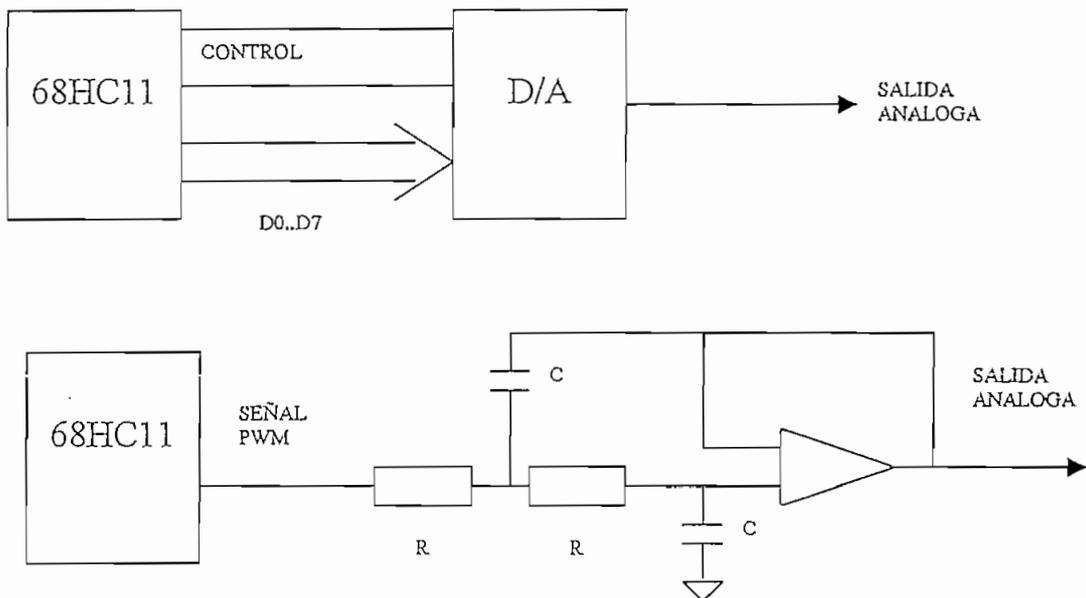


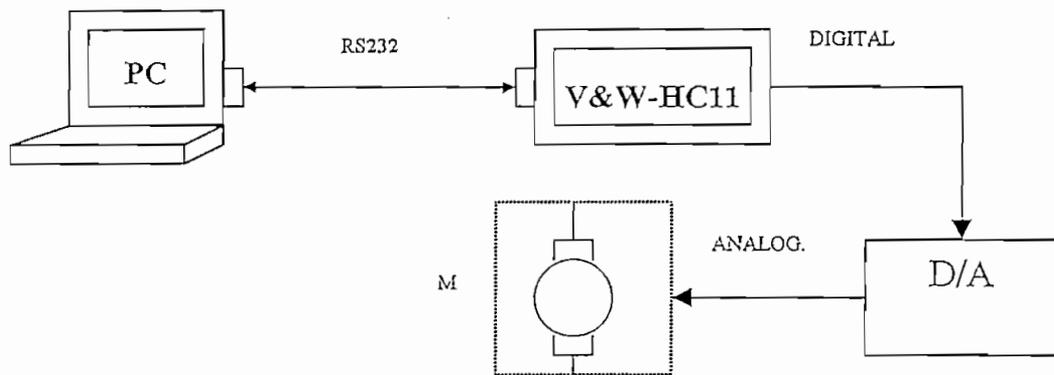
FIGURA P.7

3.- Trabajo preparatorio.

- Consultar las características del convertor D/A (DAC0830).
- Diseñar el filtro de salida para el circuito número dos de la figura P7.

4.- Parte Práctica.

Implementar el siguiente sistema:



Realizar los programas para que se cumpla lo siguiente:

Utilizando las teclas del PC se desea variar la amplitud de la señal análoga con el fin de poder variar la velocidad del motor.

- Utilizando como interface el convertor D/A (DAC0830).
- Utilizando una salida digital del módulo V&W-HC11 (manejado como localidad de memoria externa \$6800) para generar una señal PWM, en el cual se acoplará el filtro de salida.

PRACTICA No. 8

TEMA: IMPLEMENTACION DIGITAL DE ALGORITMOS DE CONTROL.

1.- Objetivo.

Familiarizar al estudiante con las técnicas de implementación digital de los algoritmos de control mas comunmente utilizados en los sistemas de control y aplicarlos a control en lazo cerrado de la velocidad de un motor de corriente continua (Motomatic).

2.- Teoría.

Actualmente los microcontroladores, sistemas digitales y transductores han incursionado exitosamente en el campo de los sistemas de control, brindando mayor flexibilidad y versatilidad en el diseño de sistemas de control.

Con el procesamiento digital de señales de un sistema se ha logrado implementar con relativa sencillez algoritmos de control muy complejos que permite optimizar el comportamiento del mismo, e incorporar en el controlador tareas de visualización de las variables de interés de la planta y el auto - ajuste de los parámetros del controlador.

En la actualidad se conoce técnicas de control muy avanzadas como son: control supervisor, el control adaptivo , control predictivo, control difuso, etc . Sin embargo en esta práctica se implementará en forma digital uno de los algoritmos de control más sencillos y más comunes como es el caso del controlador Proporcional – Integral – Derivativo (PID).

El control Proporcional – Integral – Derivativo es el esquema más conocido en el control de variables continuas que requieren de rápida respuesta, un pequeño sobreimpulso y un buen margen de estabilidad. En el dominio del tiempo, la relación que define la acción del control PID es:

$$e(t) = V_{ref} - V_{in}$$

$$V_{o (PID)} = K_p [e(t) + K_i \int e(t) dt + K_d \cdot de(t)/dt]$$

3.- Trabajo preparatorio.

- Deducir la expresión discreta (ecuación de diferencias) correspondientes al controlador PID .

4.- Parte Práctica.

Se desea implementar un sistema que realice las siguientes funciones:

- A través del puerto de comunicación serial del computador se envía al microcontrolador los códigos necesarios para que este realice la medición de una señal analógica conectada al conversor A/D pin PE1. Esta señal es el voltaje de realimentación acondicionado, proveniente de un taco generador y corresponde a la velocidad del motor del Motomatic. De igual manera el computador consulta al microcontrolador la referencia de velocidad (set point).
- La referencia de velocidad podrá ser variada mediante las teclas del teclado matricial:
 - Tecla 1: incrementa set point en 5
 - Tecla 2: incrementa set point en 10
 - Tecla 3: decrementa set point en 5
 - Tecla 4: decrementa set point en 10
- El computador recibe las señales de set point y de realimentación, y obtiene el error. Esta señal de error es la entrada de la ecuación de diferencias correspondiente al algoritmo del controlador PID. El resultado del cálculo del algoritmo del controlador deberá ser apropiadamente limitado y escalado para que su rango de variación esté siempre entre 00 y 255 (FFH).
- El computador envía al microcontrolador el resultado del PID, el cual enviará este valor al conversor D/A. El voltaje analógico del conversor D/A será conectado al amplificador de potencia a fin de que esta señal sea la que imponga el voltaje de armadura al motor para mantener la velocidad real igual al set point.
- Permanentemente en la pantalla del computador se graficará la señal de realimentación proveniente del taco generador a fin de poder visualizar el comportamiento de la señal de la velocidad ante variaciones del set point o ante perturbaciones en el motor. Simultáneamente en el display numérico se mostrará el dato de velocidad.

PRACTICA No. 9

TEMA: IMPLEMENTACION DE FILTROS DIGITALES.

1.- Objetivo.

Familiarizar al estudiante con las técnicas de implementación de filtros digitales

Utilizar el pórtico paralelo del computador en la adquisición de señales analógicas, para poder lograr mayores velocidades de transmisión de datos.

2.- Teoría.

En muchas aplicaciones de procesamiento digital de señales resulta ventajoso la utilización de filtros digitales en lugar de filtros analógicos en vista que con la opción digital se consigue mejores características en cuanto a exactitud en magnitud y fase y la eliminación total de los posibles corrimientos de voltaje, deriva térmica y problemas de ruido típicamente asociados a un filtro analógico.

Es común que la señal obtenida de un transductor aparezca con niveles altos de ruido eléctrico, interferencias, etc. que impiden la utilización directa como señal de realimentación en un sistema de control. Para superar éste problema tradicionalmente han existido dos alternativas; la utilización de transductores más sofisticados y costosos o bien la incorporación de circuitos de acondicionamiento de señal. La tendencia tecnológica actual esta orientada a reducir los costos de instrumentación mediante la utilización de transductores de bajo costo y a las señales de salida de éstos, realizar el procesamiento digital para el acondicionamiento y depuración con algoritmos y métodos numéricos sofisticados pero factibles de implementar en procesadores digitales de bajo costo.

El término filtro digital hace referencia al procesamiento computacional o algoritmo mediante el cual una señal digital o una secuencia de valores numéricos que actúan como entrada es transformada en una segunda secuencia de valores numéricos denominada secuencia digital de salida. Los filtros digitales actúan sobre señales discretas que se obtienen por muestreo en el tiempo, seleccionando para el efecto una frecuencia muestreadora apropiada a través de un conversor A/D. La señal de salida puede volver a ser convertida en una señal analógica mediante la utilización de un converso D/A.

A más de las ventajas antes señaladas, la implementación digital de filtros permite la alteración de los parámetros del filtro con bastante facilidad. Dado que la implementación digital está basada en ecuaciones de diferencias, la modificación del filtro y de sus características se reduce a cambiar las constantes de la ecuación correspondiente.

Los tipos de filtros más utilizados son los filtros pasa-bajos, pasa-altos, pasa-banda y rechaza-banda. Estos filtros son utilizados en gran cantidad de aplicaciones ya que posibilitan el remover ruido de una señal, eliminar la distorsión de una señal, la separación de señales cuando son transmitidas en conjunto, la demodulación de señales, etc.

Para diseñar un filtro digital se deben seguir los siguientes pasos:

- a) Aproximación: es el proceso de obtener una función de transferencia que satisfaga un conjunto de satisfacciones deseadas, las cuales pueden incluir la respuesta en el dominio del tiempo, la respuesta de frecuencia o una combinación entre ambas características.
- b) Realización: Consiste en la conversión de la función de transferencia en redes digitales y la definición de un algoritmo digital factible de implementar.
- c) Estudio de errores aritméticos: que implica un análisis de los efectos que los errores aritméticos de truncación pueden producir en la respuesta del filtro.
- d) Implementación: que es la tarea concerniente a la definición del hardware real, la circuitería y la codificación de los algoritmos.

3.- Trabajo Preparatorio.

Utilizando el método de ZOH obtener la ecuación de diferencias correspondiente a un filtro pasa bajos de segundo orden cuyos parámetros de operación sean:

Frecuencia de corte: f_c .

Ganancia en D.C.: 1,0

Período de muestreo T_m .

4.- Parte Práctica.

Escribir un programa en Quick-Basic para poder realizar las tareas que a continuación se detallan:

- El computador a través del puerto paralelo realiza la medición de una señal proveniente de un generador externo aplicada a una entrada de conversión A/D

del módulo V&W – HC11 y sacada hacia el computador a través de las salidas digitales.

- La señal externa es sinusoidal, con valor medio 2.5 voltios y amplitud menor a 5 voltios pico-pico. La frecuencia de ésta señal es aproximadamente 5 Hz.
- El computador recibe el valor instantáneo de la señal externa e interpreta los valores superiores a 7FH (2.5 V) como positivos y los valores menores a 7FH como negativos.
- El computador obtiene y grafica el valor absoluto de la señal de entrada, simulando la función de rectificación y, utiliza ésta señal rectificada como la variable de entrada para la evaluación de la ecuación de diferencias correspondiente a un filtro digital pasa-bajos de segundo orden que garantice un factor de rizado en la salida de 10%.
- El resultado de pasar la señal externa rectificada a través del filtro digital implementado en Quick-Basic deberá ser graficado en la pantalla del computador. En la pantalla del computador se debe poder apreciar simultáneamente las formas de onda tanto de la señal de entrada rectificada como de la señal filtrada.

Los programas desarrollados en lenguaje ensamblador y en Quick Basic para la resolución de los problemas planteados en todas las prácticas se encuentran en el apéndice B.

CAPITULO No. 5

CONCLUSIONES Y RECOMENDACIONES

- El módulo V&W-HC11 cumple con los objetivos propuestos, es decir ser un módulo didáctico para probar programas realizados en lenguaje ensamblador del microcontrolador MC68HC11 y permitir visualizar sus resultados en los diferentes periféricos de salida, además incluye componentes que permiten el ingreso de información en formato digital como también análogo.

- Para desarrollar esta Tesis se construyó la tarjeta de propósito general (“V&W-IN/OUT”) de entrada y salida de datos, que facilita la comunicación entre el usuario y el equipo. Además ésta se acopla a las condiciones de funcionamiento impuestas por la tarjeta M68HC11EVB.

- Las pruebas de funcionamiento de módulo V&W-HC11 fueron realizadas durante el desarrollo de las prácticas para el Laboratorio de Control con Microprocesadores, en las cuales se pudo comprobar el potencial de este microcontrolador así como la versatilidad que presenta el módulo para su manejo, por ser un equipo integrado por 3 tarjetas de propósito general lo cual implica pocas conexiones que reducen las fallas por cableado.

- Los microcontroladores de la familia M68HC11 de Motorola presentan múltiples ventajas para la implementación de Sistemas de Control (Adquisición de Datos, Implementación de Algoritmos de Control, Procesamiento Digital de Señales, etc.), debido a su rapidez en la ejecución de instrucciones (un ciclo de reloj igual a un ciclo de máquina), gran número de instrucciones para facilitar la implementación de programas (ciento ochenta), múltiples pórtricos de entrada - salida analógicas y digitales, transmisión serial, registros de control, sistemas de temporización y contadores de eventos externos.

- El programa monitor presenta la ventaja de permitir el ensamblaje de los programas utilizando dos métodos, el primero es usando el ensamblador en línea del programa monitor BUFFALO el cual automáticamente detecta los errores en cada línea de comando, mientras que el segundo método consiste en ensamblar los programas en un computador utilizando el ensamblador AS11 y luego descargarlos utilizando el pórtrico terminal del módulo V&W-HC11.

- El módulo V&W-HC11 es un dispositivo que ha sido diseñado para trabajar en forma paralela al MICROLAB en las prácticas de Control con Microprocesadores, con lo cual se pretende que el estudiante vaya familiarizándose con las funciones que puede realizar este módulo, pues también se ha realizado la implementación de todos los algoritmos para resolver los problemas planteados en las prácticas en lenguaje ensamblador del microcontrolador M68HC11.

El módulo V&W – HC11 presenta todas las facilidades para implementar un sistema de control sobre un sistema real, ya que se lo puede usar en fase de pruebas para depurar el

algoritmo de control, y una vez finalizado y probado el programa se lo graba en la memoria EEPROM, se cambia el jumper J4, se resetea el módulo y ahora el programa que correrá será el correspondiente al algoritmo de control desarrollado y grabado en la memoria EEPROM.

Los conectores correspondientes al pórtilo D del módulo V&W – HC11 permiten desarrollar sistemas de comunicación serial tanto sincrónica como asincrónica con otros módulos de similares características en comunicaciones, lo que nos permitiría implementar redes de equipos trabajando en los modos simplex, half – duplex y full – duplex tanto para pruebas de laboratorio como en aplicaciones para sistemas reales.

Todos los programas implementados para resolver los problemas planteados en las diferentes prácticas enumeradas en el capítulo cuatro se han realizado tratando de optimizar en cuanto al número de instrucciones y al tiempo total de ejecución del programa con el propósito de ocupar en los programas la menor cantidad de memoria y que éstos se ejecuten en el menor tiempo posible.

- Debido a la amplia difusión de los microcontroladores en los últimos años, se recomienda incluir en la materia Control con Microprocesadores un capítulo en el cual se describan todas las características, ventajas y desventajas de las familias de microcontroladores que en la actualidad están repuntando, como por ejemplo la familia de los M68HC11 de la casa MOTOROLA, la familia de los PIC de la casa MICROCHIP , la familia de los DS5000 de la casa DALLAS entre otros.

BIBLIOGRAFIA:

- M68HC11 REFERENCE MANUAL, Motorola, Tercera Edición, 1991, USA.
- M68HC11A8 TECHNICAL DATA, Motorola, Primera Edición, 1991, USA.
- M68HC11A8 PROGRAMMING REFERENCE GUIDE, Motorola, Cuarta Edición, 1990, USA.
- M68HC11EVB EVALUATION BOARD USER'S MANUAL, Motorola, Primera Edición, 1986, USA.
- INTERFERENCIAS ELECTROMAGNETICAS EN SISTEMAS ELECTRONICOS, Editorial Marcombo, España, 1992.
- QUICK BASIC, Microsoft Corporation, USA, 1987.
- ENGINEERING SOLUTION ON A CHIP FROM INTERSIL, Intersil, New York, 1982.
- INTRODUCCION A LOS MICROCONTROLADORES, Gonzáles Vásques, Editorial McGraw-Hill, Madrid, 1992.
- LINEAR DATA BOOK, National Semiconductor, California, 1992.
- INGENIERIA DE CONTROL MODERNO, Ogata Katsuhiko, Editorial Prentice/Hall, 1974.

APENDICE A

CARACTERISTICAS ELECTRICAS

DEL MC68HC11

En este apéndice se resumen las principales características eléctricas del elemento principal del módulo V&W – HC11, como es el microcontrolador MC68HC11.

ELECTRICAL CHARACTERISTICS

Table A-1. Maximum Rating

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	- 0.3 to + 7.0	V
Input Voltage	V_{in}	- 0.3 to + 7.0	V
Operating Temperature Range MC68HC11A8 MC68HC11A8C MC68HC11A8V MC68HC11A8M MC68L11A8	T_A	T_L to T_H 0 to 70 - 40 to 85 - 40 to 105 - 40 to 125 - 20 to 70	°C
Storage Temperature Range	T_{stg}	- 55 to 150	°C
Current Drain per Pin* Excluding V_{DD} , V_{SS} , V_{IH} , and V_{IL}	I_D	25	mA

*One pin at a time, observing maximum power dissipation limits.

Internal circuitry protects the inputs against damage caused by high static voltages or electric fields; however, normal precautions are necessary to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. Extended operation at the maximum ratings can adversely affect device reliability. Tying unused inputs to an appropriate logic voltage level (either GND or V_{DD}) enhances reliability of operation.

Table A-2. Thermal Characteristics

Characteristic	Symbol	Value	Unit
Average Junction Temperature	T_J	$T_A + (P_D \times \Theta_{JA})$	$^{\circ}\text{C}$
Ambient Temperature	T_A	User-determined	$^{\circ}\text{C}$
Package Thermal Resistance (Junction-to-Ambient) 52-Pin Plastic Quad Pack (PLCC) 48-Pin Plastic Dual In-Line Package (DIP)	Θ_{JA}	50 40	$^{\circ}\text{C/W}$
Total Power Dissipation	P_D	$\frac{P_{INT} + P_{IO}}{K + (T_J + 273^{\circ}\text{C})}$ (Note 1)	W
Device Internal Power Dissipation	P_{INT}	$I_{DD} \times V_{DD}$	W
I/O Pin Power Dissipation	P_{IO} (Note 2)	User-determined	W
A Constant	K	$\frac{P_D \times (T_A + 273^{\circ}\text{C}) + \Theta_{JA} \times P_{D2}}$ (Note 3)	W \cdot $^{\circ}\text{C}$

NOTES:

1. This is an approximate value, neglecting P_{VO} .
2. For most applications $P_{VO} = P_{INT}$ and can be neglected.
3. K is a constant pertaining to the device. Solve for K with a known T_A and a measured P_D (at equilibrium). Use this value of K to solve for P_D and T_J iteratively for any value of T_A .

V_{DD}
 Out
 I_{Lea}
 Out
 I_{Lea}
 Out
 I_{Lea}
 Inpt
 Inpt
 I/O
 V_{Z1}
 Inpt
 V_{Z1}
 V_{Z1}
 RAH
 RAH
 Total
 RUN
 WA
 STC
 Inpt
 Pow

NOTE
 1.
 2.
 3.

Table A-3. DC Electrical Characteristics

V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H, unless otherwise noted

Characteristics	Symbol	Min	Max	Unit
Output Voltage (Note 1) All Outputs except XTAL All Outputs Except XTAL, RESET, and MODA I _{Load} = ± 10.0 μA	V _{OL} V _{OH}	— V _{DD} - 0.1	0.1 —	V
Output High Voltage (Note 1) I _{Load} = -0.8 mA, V _{DD} = 4.5 V	V _{OH}	V _{DD} - 0.8	—	V
Output Low Voltage I _{Load} = 1.5 mA	V _{OL}	—	0.4	V
Input High Voltage	V _{IH}	0.7 x V _{DD} 0.8 x V _{DD}	V _{DD} + 0.3 V _{DD} + 0.3	V
Input Low Voltage	V _{IL}	V _{SS} - 0.3	0.2 x V _{DD}	V
I/O Ports, Three-State Leakage V _{in} = V _{IH} or V _{IL}	I _{OZ}	—	±10	μA
Input Leakage Current (Note 2) V _{in} = V _{DD} or V _{SS} V _{in} = V _{DD} or V _{SS}	I _{in}	— —	±1 ±10	μA
RAM Standby Voltage	V _{SB}	4.0	V _{DD}	V
RAM Standby Current	I _{SB}	—	10	μA
Total Supply Current (Note 3)				
RUN:	I _{DD}			mA
Single-Chip Mode	dc - 2 MHz 3 MHz	— —	15 27	
Expanded Multiplexed Mode	dc - 2 MHz 3 MHz	— —	27 35	
WAIT:	W _{IDD}			mA
All Peripheral Functions Shut Down				
Single-Chip Mode	dc - 2 MHz 3 MHz	— —	6 15	
Expanded Multiplexed Mode	dc - 2 MHz 3 MHz	— —	10 20	
STOP:	S _{IDD}			μA
No Clocks, Single-Chip Mode	dc - 2 MHz 3 MHz	— —	50 150	
Input Capacitance	C _{in}	— —	8 12	pF
Power Dissipation	P _O			mW
Single-Chip Mode	2 MHz	—	85	
Expanded Multiplexed Mode		—	150	
Single-Chip Mode	3 MHz	—	150	
Expanded Multiplexed Mode		—	195	

NOTES:

- V_{OH} specification for RESET and MODA is not applicable because they are open-drain pins. V_{OH} specification not applicable to ports C and D in wired-OR mode.
- Refer to A/D specification for leakage current for port E.
- EXTAL is driven with a square wave, and
 I_{cy} = 500 ns for 2 MHz rating;
 I_{cy} = 333 ns for 3 MHz rating.
 V_{IL} ≤ 0.2 V
 V_{IH} ≥ V_{DD} - 0.2 V
 No dc loads.

APENDICE B

LISTA DE INSTRUCCIONES DEL PROGRAMA

MONITOR

En éste Apéndice se lista las instrucciones de inicialización del programa monitor BUFFALO. El listado total de todo el programa monitor lo encontramos en el libro "M68HC11EVb Evaluation Board User Manual".

```
*****
*   EQUATES   *
*****
RAMBS      EQU   $0000      start of ram
REGBS      EQU   $1000      start of registers
ROMBS      EQU   $E000      start of rom
STREE      EQU   $B600      start of eeprom
ENDEE      EQU   $B7FF      end of eeprom
PORTE      EQU   REGBS+$0A   port e
CFORC      EQU   REGBS+$0B   force output compare
TCNT       EQU   REGBS+$0E   timer count
TOC5       EQU   REGBS+$1E   oc5 reg
TCTL1      EQU   REGBS+$20   timer control 1
TMSK1      EQU   REGBS+$22   timer mask 1
TFLG1      EQU   REGBS+$23   timer flag 1
TMSK2      EQU   REGBS+$24   timer mask 2
BAUD       EQU   REGBS+$2B   sci baud reg
SCCR1      EQU   REGBS+$2C   sci control1 reg
SCCR2      EQU   REGBS+$2D   sci control2 reg
SCSR       EQU   REGBS+$2E   sci status reg
SCDAT      EQU   REGBS+$2F   sci data reg
BPROT      EQU   REGBS+$35   block protect reg
OPTION     EQU   REGBS+$39   option reg
COPRS      EQU   REGBS+$3A   cop reset reg
PPROG      EQU   REGBS+$3B   ee prog reg
HPRIO      EQU   REGBS+$3C   hprio reg
CONFIG     EQU   REGBS+$3F   config register
DFLOP      EQU   $4000      evb d flip flop
DUART      EQU   $D000      duart address
PORTA      EQU   DUART
PORTB      EQU   DUART+8
ACIA       EQU   $9800      acia address
PROMPT     EQU   '>'
BUFFLNG    EQU   35
CTLA       EQU   $01      exit host or assembler
CTLB       EQU   $02      send break to host
CTLW       EQU   $17      wait
CTLX       EQU   $18      abort
DEL        EQU   $7F      abort
EOT        EQU   $04      end of text/table
SWI        EQU   $3F
```

```
*****
*   RAM       *
*****
      ORG   $33
*** Buffalo ram space ***
      RMB   20      user stack area
USTACK    RMB   30      monitor stack area
STACK     RMB    1
```

REGS	RMB	9	user's pc, y, x, a, b, c
SP	RMB	2	user's sp
INBUFF	RMB	BUFFLNG	input buffer
ENDBUFF	EQU	*	
COMBUFF	RMB	8	command buffer
SHFTREG	RMB	2	input shift register
BRKTABL	RMB	8	breakpoint table
AUTOLF	RMB	1	auto lf flag for i/o
IODEV	RMB	1	0=sci, 1=acia, 2=duartA, 3=duartB
EXTDEV	RMB	1	0=none, 1=acia, 2=duart,
HOSTDEV	RMB	1	0=sci, 1=acia, 3=duartB
COUNT	RMB	1	# characters read
CHRCNT	RMB	1	# characters output on current line
PTRMEM	RMB	2	current memory location

*** Buffalo variables - used by: ***

PTR0	RMB	2	main, readbuff, incbuff, AS
PTR1	RMB	2	main, BR, DU, MO, AS, EX
PTR2	RMB	2	EX, DU, MO, AS
PTR3	RMB	2	EX, HO, MO, AS
PTR4	RMB	2	EX, AS
PTR5	RMB	2	EX, AS, BOOT
PTR6	RMB	2	EX, AS, BOOT
PTR7	RMB	2	EX, AS
PTR8	RMB	2	AS
TMP1	RMB	1	main, hexbin, buffarg, termarg
TMP2	RMB	1	GO, HO, AS, LOAD
TMP3	RMB	1	AS, LOAD
TMP4	RMB	1	TR, HO, ME, AS, LOAD

*** Vector jump table ***

JSCI	RMB	3
JSPI	RMB	3
JPAIE	RMB	3
JPAO	RMB	3
JTOF	RMB	3
JTOC5	RMB	3
JTOC4	RMB	3
JTOC3	RMB	3
JTOC2	RMB	3
JTOC1	RMB	3
JTIC3	RMB	3
JTIC2	RMB	3
JTIC1	RMB	3
JRTI	RMB	3
JIRQ	RMB	3
JXIRQ	RMB	3
JSWI	RMB	3
JILLOP	RMB	3
JCOP	RMB	3
JCLM	RMB	3

```

*****
*
* ROM starts here *
*
*****

```

```

ORG ROMBS

```

```

*****
** BUFFALO - This is where Buffalo starts
** out of reset. All initialization is done
** here including determination of where the
** user terminal is (SCI,ACIA, or DUART).
*****

```

```

BUFFALO LDX #PORTE
        BRCLR 0,X,$01 BUFISIT if bit 0 of port e is 1
        JMP $B600 then jump to the start of EEPROM
BUFISIT LDAA #$93
        STAA OPTION          adpu, dly, irqe, cop
        LDAA #$00
        STAA TMSK2          timer pre = %1 for trace
        LDAA #$00
        STAA BPROT          clear 'E9 eeprom block protect
        LDS #STACK          monitor stack pointer
        JSR VECINIT
        LDX #USTACK
        STX SP              default user stack
        LDAA TCTL1
        ORAA #$03
        STAA TCTL1          force oc5 pin high for trace
        LDAA #$D0
        STAA REGS+8          default user ccr
        LDD #$3F0D          initial command is ?
        STD INBUFF
        JSR BPCLR           clear breakpoints
        CLR AUTOLF
        INC AUTOLF          auto cr/lf = on

```

```

* Determine type of external comm device - none, or acia *

```

```

        CLR EXTDEV          default is none
        LDAA HPRIO
        ANDA #$20
        BEQ BUFF2          jump if single chip mode
        LDAA #$03          see if external acia exists
        STAA ACIA          master reset
        LDAA ACIA
        ANDA #$7F          mask irq bit from status register
        BNE BUFF1          jump if status reg not 0
        LDAA #$12
        STAA ACIA          turn on acia
        LDAA ACIA
        ANDA #$02
        BEQ BUFF1          jump if tdre not set
        LDAA #$01
        STAA EXTDEV          external device is acia
        BRA BUFF2

```

```

BUFF1 EQU *              see if duart exists
        LDAA DUART+$0C     read IRQ vector register

```

```

CMPA  #$0F    should be out of reset
BNE   BUFF2
LDAA  #$AA
STAA  DUART+$0C    write irq vector register
LDAA  DUART+$0C    read irq vector register
CMPA  #$AA
BNE   BUFF2
LDAA  #$02
STAA  EXTDEV      external device is duart A

```

* Find terminal port - SCI or external. *

```

BUFF2  CLR  IODEV
      JSR  TARGCO    disconnect sci for evb board
      JSR  SIGNON   initialize sci
      LDAA EXTDEV
      BEQ  BUFF3    jump if no external device
      STAA IODEV
      JSR  SIGNON   initialize external device
BUFF3  CLR  IODEV
      JSR  INPUT    get input from sci port
      CMPA #$0D
      BEQ  BUFF4    jump if cr - sci is terminal port
      LDAA EXTDEV
      BEQ  BUFF3    jump if no external device
      STAA IODEV
      JSR  INPUT    get input from external device
      CMPA #$0D
      BEQ  BUFF4    jump if cr - terminal found ext
      BRA  BUFF3

SIGNON  JSR  INIT    initialize device
      LDX  #MSG1    buffalo message
      JSR  OUTSTRG
      RTS

```

* Determine where host port should be. *

```

BUFF4  CLR  HOSTDEV    default - host = sci port
      LDAA IODEV
      CMPA #$01
      BEQ  BUFF5    default host if term = acia
      LDAA #$03
      STAA HOSTDEV    else host is duart port b
BUFF5  EQU  *

```

APENDICE C

ARCHIVOS "S-RECORDS"

Los archivos con formato S – records fueron creados con el propósito de codificar programas o archivos de datos en un formato de tipo para transportación entre sistemas de computadoras. Este proceso de transmisión puede por lo tanto ser monitoreado, y los archivos con formato s – records pueden ser fácilmente editados.

Cuando observamos un archivo con formato s – records vemos esencialmente cadenas de caracteres formados por varios campos, los cuales identifican el tipo de s – record, su longitud, dirección de memoria, el código o dato y un checksum. Cada byte de datos es codificado como un número hexadecimal de dos caracteres: el primer caracter representa los cuatro bits de orden alto, y el segundo los cuatro bits de orden bajo del byte.

Los cinco campos que componen un archivo s – record son:

TIPO	LONGITUD	DIRECCION	DATO	CHECKSUM
------	----------	-----------	------	----------

Donde los campos son compuestos así:

CAMPO	CARACTERES ENVIADOS	CONTENIDO
TIPO	2	Tipo de archivo s – record: S0, S1, etc.
LONGITUD	2	Par de caracteres que cuentan el número de bytes del s- record, excluido el tipo y la longitud.
DIRECCION	4,6,8	2, 3 o 4 bytes de dirección en la cual el dato va a ser cargado en la memoria.
DATO		De 0 a n bytes de código ejecutable, datos cargados en memoria o información descriptiva.
CHECK SUM	2	Byte menos significativo de los complementos de uno de la suma de los valores representados por los pares de caracteres que componen los campos de longitud, dirección y dato.

Un ejemplo de un archivo s – record es:

S00600004844521B

En el cual S0 es el tipo de archivo s – record, 06 es el número de datos enviados, 0000 es la dirección a la cual se envían los datos, 48, 44 y 52 son los caracteres ASCII de los datos enviados (H, D y R), y 1B es el resultado del checksum.

S-RECORD INFORMATION

INTRODUCTION

The Motorola S-record format was devised for the purpose of encoding programs or data files in a *printable* format for transportation between computer systems. This transportation process can therefore be monitored and the S-records can be easily edited.

S-RECORD CONTENT

When observed, S-records are essentially character strings made of several fields which identify the record type, record length, memory address, code/data, and checksum. Each byte of binary data is encoded as a 2-character hexadecimal number: the first character representing the high-order 4 bits, and the second the low-order 4 bits of the byte.

Five fields which comprise an S-record are shown below:

TYPE	RECORD LENGTH	ADDRESS	CODE/DATA	CHECKSUM
------	---------------	---------	-----------	----------

where the fields are composed as follows:

FIELD	PRINTABLE CHARACTERS	CONTENTS
Type	2	S-record type — S0, S1, etc.
Record Length	2	Character pair count in the record, excluding the type and record length.
Address	4, 6, or 8	2-, 3-, or 4-byte address at which the data field is to be loaded into memory.
Code/data	0-2n	From 0 to n bytes of executable code, memory loadable data, or descriptive information. For compatibility with teletypewriters, some programs may limit the number of bytes to as few as 28 (56 printable characters in the S-record).
Checksum	2	Least significant byte of the one's complement of the sum of the values represented by the pairs of characters making up the record length, address, and the code/data fields.

Eight types of S-records have been defined to accommodate the several needs of the encoding, transportation, and decoding functions. The various Motorola upload, download, and other record transportation control programs, as well as cross assemblers, linkers, and other file-creating or debugging programs, utilize only those S-records which serve the purpose of the program. For specific information on which S-records are supported by a particular program, the user manual for that program must be consulted.

NOTE

The EVB monitor supports only the S1 and S9 records. All data before the first S1 record is ignored. Thereafter, all records must be S1 type until the S9 record terminates data transfer.

An S-record format may contain the following record types:

- S0 Header record for each block of S-records. The code/data field may contain any descriptive information identifying the following block of S-records. The address field is normally zeroes.
- S1 Code/data record and the 2-byte address at which the code/data is to reside.
- S2-S8 Not applicable to EVB.
- S9 Termination record for a block of S1 records. Address field may optionally contain the 2-byte address of the instruction to which control is to be passed. If not specified, the first entry point specification encountered in the input will be used. There is no code/data field.

NOTE

The LOAD command reads the S9 record but ignores the start address, if any. Control is returned to BUFFALO.

Only one termination record is used for each block of S-records. Normally, only one header record is used, although it is possible for multiple header records to occur.

S-RECORD CREATION

S-record format programs may be produced by several dump utilities, debuggers, or several cross assemblers or cross linkers. Several programs are available for downloading a file in S-record format from a host system to an 8-bit or 16-bit microprocessor-based system.

S-RECORD EXAMPLE

Shown below is a typical S-record format, as printed or displayed:

```
S00600004844521B
S1130000285F245F2212226A000424290008237C2A
S11300100002000800082629001853812341001813
S113002041E900084E42234300182342000824A952
S107003000144ED492
S9030000FC
```

The above format consists of an S0 header record, four S1 code/data records, and an S9 termination record.

The S0 header record is comprised of the following character pairs:

S0	S-record type S0, indicating a header record.
06	Hexadecimal 06 (decimal 6), indicating six character pairs (or ASCII bytes) follow.
00 00	Four-character 2-byte address field, zeroes.
48 44 52	ASCII H, D, and R — "HDR".
1B	Checksum of S0 record.

The first S1 code/data record is explained as follows:

S1	S-record type S1, indicating a code/data record to be loaded/verified at a 2-byte address.
13	Hexadecimal 13 (decimal 19), indicating 19 character pairs, representing 19 bytes of binary data, follow.
00	Four-character 2-byte address field; hexadecimal address 0000, indicates location where the following data is to be loaded.

The next 16 character pairs are the ASCII bytes of the actual program code/data. In this assembly language example, the hexadecimal opcodes of the program are written in sequence in the code/data fields of the S1 records:

OPCODE	INSTRUCTION
28 5F	BHCC S0161
24 5F	BCC S0163
22 12	BHI S0118
22 6A	BHI S0172
00 04 24	BRSET 0,\$04,\$012F
29 00	BHCS S010D
08 23 7C	BRSET 4,\$23,\$018C

(Balance of this code is continued in the code/data fields of the remaining S1 records, and stored in memory location 0010, etc..)

2A Checksum of the first S1 record.

The second and third S1 code/data records each also contain S13 (19) character pairs and are coded with checksums 13 and 52, respectively. The fourth S1 code/data record contains 07 character pairs and has a checksum of 92.

The S9 termination record is explained as follows:

- S9 S-record type S9, indicating a termination record.
- 03 Hexadecimal 03, indicating three character pairs (3 bytes) follow.
- 00 Four-character 2-byte address field, zeroes.
- 00
- FC Checksum of S9 record.

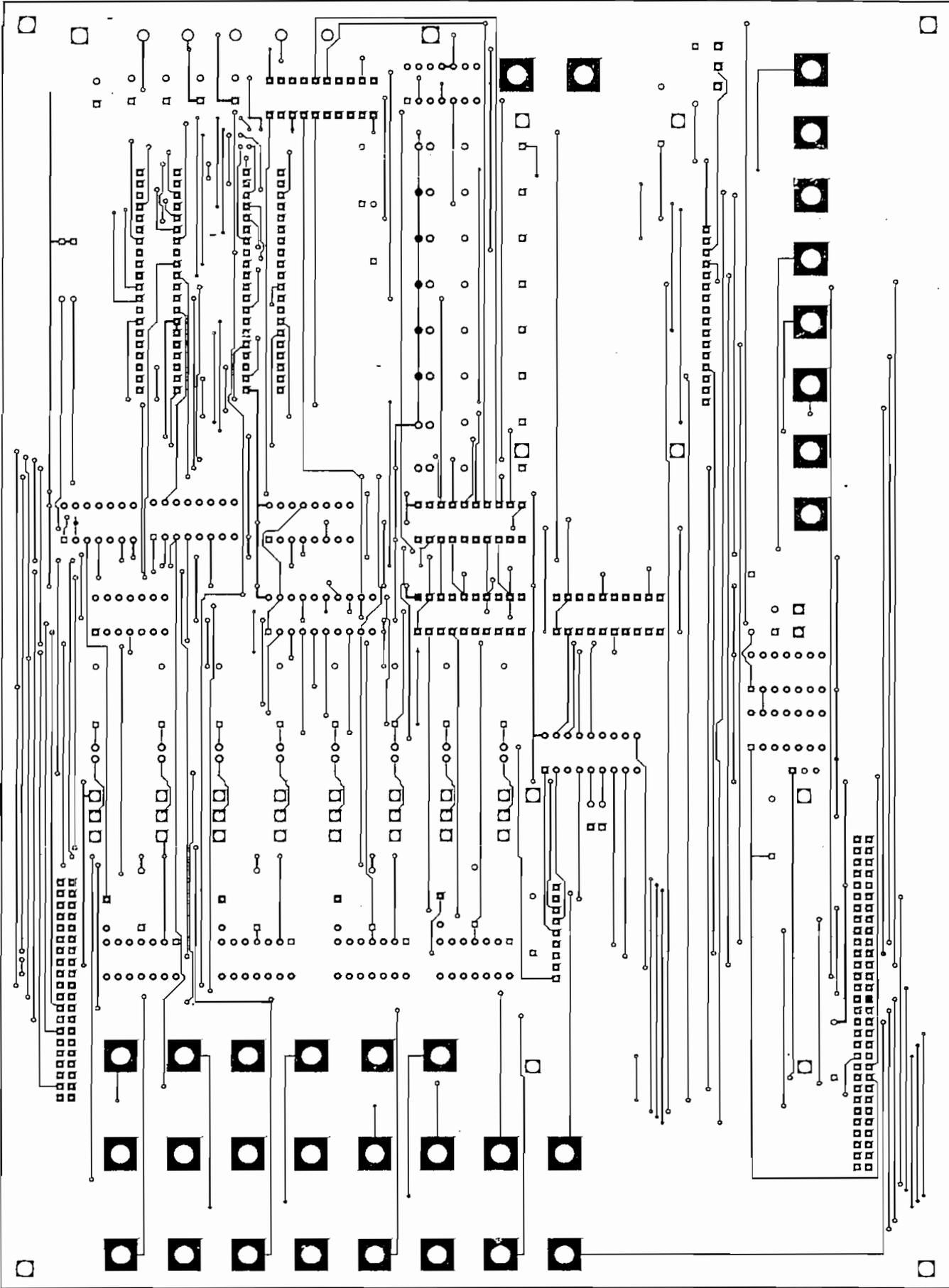
Each printable character in an S-record is encoded in hexadecimal (ASCII in this example) representation of the binary bits which are actually transmitted. For example, the first S1 record above is sent as shown below.

TYPE	LENGTH	ADDRESS	CODE/DATA	CHECKSUM
S	1	0 0 0 0	2 8 5 F ...	2 A
3	3 1	3 1 3 3	3 0 3 0 3 0 3 0	3 2 3 8 3 5 4 6 ...
		0011 0011 0001 0011 0001 0011 0011 0000 0011 0000 0011 0000 0011 0000 0011 0010 0011 1000 0011 0101 0100 0110 ...		0011 0010 0100 0001

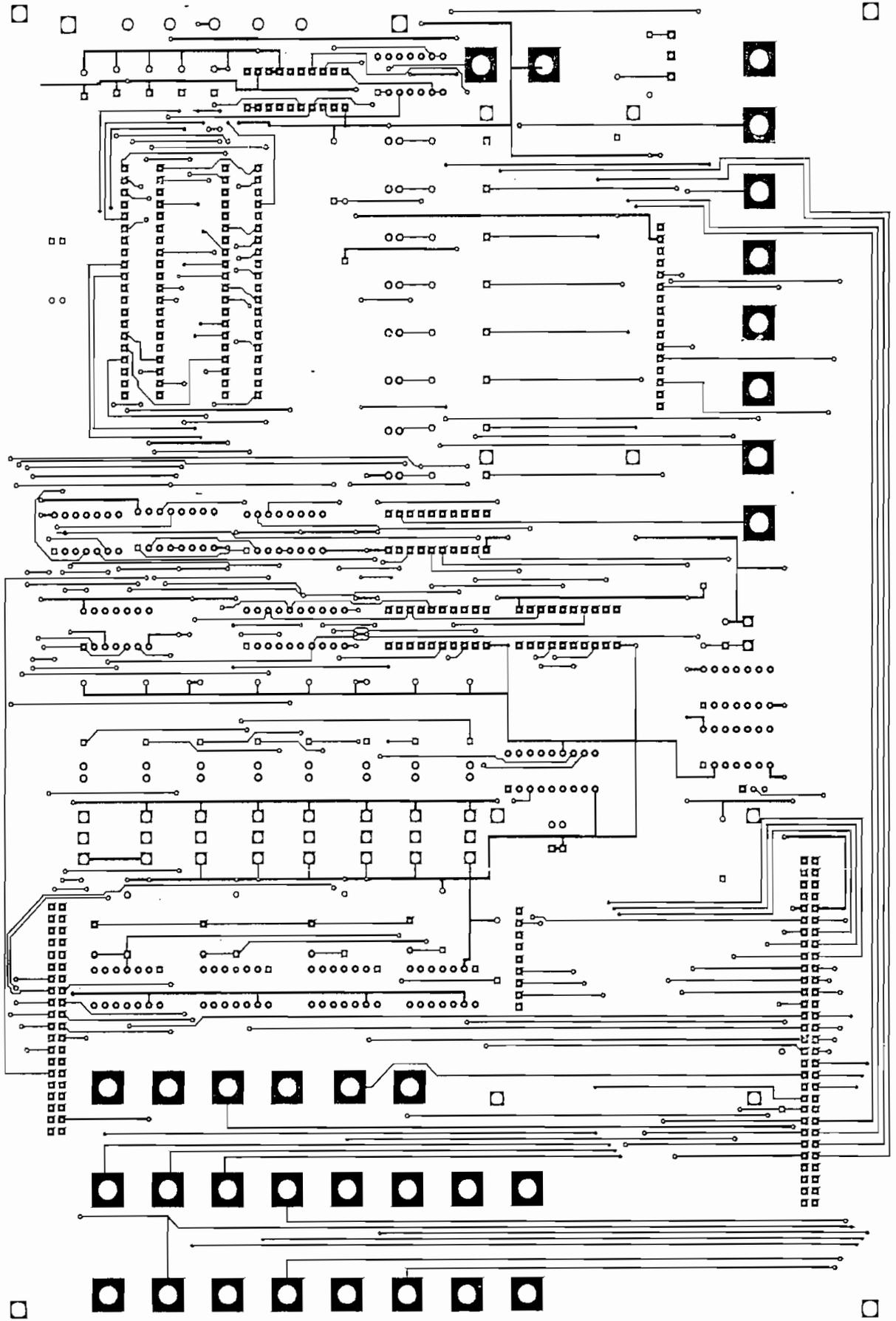
APENDICE D

CIRCUITO IMPRESO DE LA TARJETA V&W - IN/OUT

CIRCUITO IMPRESO (LADO SUPERIOR)



CIRCUITO IMPRESO (LADO INFERIOR)



APENDICE E

DIAGRAMA DE TARJETAS

