

BENEMÉRITA
UNIVERSIDAD AUTÓNOMA DE PUEBLA
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

SCIP
Sistema de Coprocesamiento de Información
para PC

Tesis Profesional
que para obtener el título de
Licenciado en Ciencias de la Computación
presenta

Rodolfo González González

Asesor
M.C. Rafael Rivera Rodríguez

Puebla, Pue.

1998.

*BENEMÉRITA
UNIVERSIDAD AUTÓNOMA DE PUEBLA
FACULTAD DE CIENCIAS DE LA
COMPUTACIÓN*

SCIP

Sistema de Coprocesamiento de Información
para PC

*Tesis Profesional
que para obtener el título de
Licenciado en Ciencias de la Computación
presenta*

Rodolfo González González

Asesor
M.C. Rafael Rivera Rodríguez

Puebla, Pue.

1998.

*Dedico esta tesis a mis padres, Concepción González Mendoza y Rodolfo
González García.*

A toda mi familia.

A mi asesor, M.C. Rafael Rivera Rodríguez, por su ayuda para la realización de esta tesis.

A los miembros de la Comisión Revisora de la Academia de Hardware para asignación de mención Cum Laude, y a los miembros del H. Jurado de Exámen Profesional.

Al M.C. Esteban Torres León y al Dr. Manuel Martín Ortíz, por la información y sugerencias que ayudaron a llevar a cabo el proyecto.

A mis profesores.

A todo el personal del Laboratorio de Hardware de la F.C.C - B.U.A.P.

*A mis amigos y compañeros, especialmente a Vanessa Hekel Enschedé,
Berenice Bocanegra, Diana Ramírez, Iris Molleda, Margarita Barajas,
Claudia Romero, Alberto Beltrán, Emilio Casco, Edmundo Martínez, Abel
Flores, Carlos Ayala Raggi, et. al.*

Ah!, On, Off, simplement!...

Jean-Michel Jarre.

CONTENIDO

Introducción y Objetivos.	9
1 Microprocesadores.	11
1.1 Introducción.	11
1.2 Generalidades.	11
1.2.1 Evolución del Microprocesador.	11
1.2.2 Controladores Dedicados.	13
1.2.3 CPU's de Propósito General.	13
1.3 Introducción a los microprocesadores Intel 8088 y 80x86. . . .	14
1.3.1 Coprocesadores matemáticos.	15
1.3.2 Arquitectura interna de los microprocesadores Intel 8088 y 8086.	15
1.3.3 Inicialización y reinicialización del procesador.	19
1.3.4 Descripción de las terminales del 8088 y 8086.	21
2 Arquitectura basada en 8088.	27
2.1 Introducción.	27
2.2 Organización de la Memoria para el 8088/8086.	27
2.3 Operación del bus.	28
2.3.1 Ciclo de lectura.	29
2.3.2 Ciclo de escritura.	30
2.4 Interrupciones.	31
2.4.1 Interrupciones por hardware.	32
2.4.2 Respuesta ante una interrupción.	33
2.5 Puertos de Entrada/Salida.	34
2.6 Bus del Sistema de la IBM PC.	37
2.6.1 Consideraciones especiales sobre poder.	39
2.7 Extensión de procesador numérico.	40
2.7.1 Ventajas.	40
2.7.2 Arquitectura Interna.	42

2.7.3	Manejo de Excepciones.	43
2.7.4	Configuración y operación.	44
3	Diseño y Descripción del Sistema.	49
3.1	Introducción.	49
3.2	Módulo de procesamiento.	50
3.3	Módulo de memoria, puertos y decodificación interna.	53
3.3.1	Memoria interna de SCIP.	53
3.3.2	Puertos internos.	56
3.4	Módulo de comunicación.	57
3.4.1	Técnicas de decodificación para selección de dispositivos.	57
3.4.2	Descripción del circuito de decodificación de SCIP.	59
3.4.3	Descripción del circuito de comunicación de SCIP.	60
3.5	Protocolo de comunicación entre la tarjeta SCIP y la computadora anfitriona	63
3.6	Consideraciones importantes sobre la programación.	64
4	Pruebas del Sistema y sus Resultados.	67
4.1	Introducción.	67
4.2	Pruebas del sistema.	67
4.3	Resultados de las pruebas.	73
	Conclusiones y Perspectivas.	75
	Bibliografía.	77
A	Manual del Usuario.	79
A.1	Introducción.	79
A.2	Manejo e instalación del hardware de SCIP.	79
A.2.1	Introducción al hardware de SCIP.	79
A.2.2	Precauciones.	82
A.2.3	Instalación.	82
A.2.4	Manejo del Hardware.	83
A.3	Programación de SCIP.	84
A.3.1	Localización del programa en memoria.	84
A.3.2	Programación de las memorias EPROM.	86

B	Especificaciones de los principales circuitos usados.	87
B.1	Introducción.	87
B.2	Intel 8088.	87
B.2.1	Introducción.	87
B.2.2	Microprocesador de 8 bits HMOS (8088/8088-2).	87
B.3	Intel 8087.	89
B.3.1	Introducción.	89
B.3.2	Coprocesador de Datos Numéricos 8087/8087-2/8087-1.	89
B.4	Intel 8288.	90
B.4.1	Introducción.	90
B.4.2	Controlador de Bus CHMOS 82C88.	90
B.5	Intel 8259.	92
B.5.1	Introducción.	92
B.5.2	Controlador de Periféricos Programable 8259A.	92
B.6	Intel 8284.	93
B.6.1	Introducción.	93
B.6.2	Generador de reloj y manejador, para CPU's 8086, 8284.	93
B.7	Intel 8255A.	93
B.7.1	Introducción.	93
B.7.2	Controlador programable de periféricos 8255.	94
C	Diagramas.	111

LISTA DE FIGURAS

1.1	Sistema 8086/8088 básico en modo mínimo.	16
1.2	Configuración “Modo Máximo” del 8088.	17
1.3	Diagrama de bloques del 8088 y 8086	18
1.4	Formato del registro de banderas del Intel 8088.	19
1.5	Diagrama de terminales del 8088 en Modo Máximo (Fuente: OrCad).	21
1.6	Diagrama de terminales del 8088 en Modo Mínimo (Fuente: OrCad).	22
1.7	Configuración de bus multiplexado.	24
1.8	Configuración de bus demultiplexado (arriba) y de Sistema con buffers completos usando el controlador de bus (abajo).	25
2.1	Diagrama de tiempo del sistema básico.	28
2.2	Secuencia de reconocimiento de interrupción.	33
2.3	Espacio de direcciones de puertos de entrada/salida	35
2.4	Espacio de direcciones de puertos de entrada/salida en la tar- jeta madre	35
2.5	Espacio de direcciones de puertos de entrada/salida para las ranuras de expansión	36
2.6	Diagrama de una ranura de expansión de la PC IBM (bus AT).	38
2.7	Diagrama funcional de bloques del 80x87.	43
2.8	Configuración de un sistema 8088/8087 u 8086/8087.	45
2.9	Diagrama de flujo de la operación del 8088 y 8087 sincronizados.	48
3.1	Diagrama en bloques de SCIP: (1) Sección preconstruida de decodificación, (2) Módulo de Procesamiento y Control de BUS, (3) Módulo de decodificación interna y buffers, (4) Módulo de temporización, (5) Módulo de comunicación, (6) Módulo de lógica de selección, (7) Módulo de memorias RAM, (ROM1) Memoria EPROM 1, (ROM2) Memoria EPROM 2.	50
3.2	Mapa de memoria de la tarjeta SCIP.	55
3.3	Controlador Programable de Periféricos 8255.	61

3.4	Controlador Programable de Periféricos 8255 en configuración de Modo 0, con la palabra de palabra de control 6, programando el Puerto A como salida y el Puerto B como entrada. .	62
A.1	Lado de componentes de SCIP.	80
A.2	Lado de alambramiento del prototipo de SCIP.	80
A.3	Diagrama en bloques de SCIP: (1) Sección preconstruida de decodificación, (2) Módulo de Procesamiento y Control de BUS, (3) Módulo de decodificación interna y buffers, (4) Módulo de temporización, (5) Módulo de comunicación, (6) Módulo de lógica de selección, (7) Módulo de memorias RAM, (ROM1) Memoria EPROM 1, (ROM2) Memoria EPROM 2.	81
A.4	Tarjeta de expansión SCIP montada sobre una tarjeta extensora del bus (no necesaria en la práctica) en un ensamble 386, durante las pruebas del sistema.	83

LISTA DE TABLAS

1.1	Algunos microprocesadores.	12
1.2	Algunos microprocesadores (<i>continuación</i>).	13
1.3	Arquitecturas.	14
1.4	Microprocesadores y Coprocesadores en el caso de computadoras basadas en Intel 8088/80x86.	15
1.5	Funciones de las banderas del registro de banderas del Intel 8088.	20
1.6	Terminales del Intel 8088.	23
2.1	Bits de estado $\overline{S_0}$ - $\overline{S_2}$ en modo máximo.	30
2.2	Bits de estado S_3 y S_4	30
2.3	Aumento de velocidad usando un 8087.	41
2.4	Comparación entre 8086/8087 (izq.) y 8086 emulando a 8087 (der.).	41
2.5	Tipos de datos del 8087.	42
2.6	Líneas de estado que determinan el tipo de ciclo de bus. . . .	46
3.1	Rangos de direcciones de los bloques de memoria de SCIP. . .	55
3.2	Entradas del 74LS139 para la selección de las memorias RAM	56
3.3	Entradas del 74LS139 para la selección de las memorias EPROM	56
3.4	Direcciones de los puertos para los dispositivos PIC y PPI. . .	57
3.5	Modo Básico de Operación del 8255.	63
3.6	Comandos básicos para el SCIP.	64
B.1	Rangos Máximos Absolutos.	88
B.2	Rangos Máximos Absolutos.	90
B.3	Comandos del 8288.	91
B.4	Rangos Máximos Absolutos.	91

Introducción y Objetivos.

Introducción.

El **Sistema de Coprocesamiento de Información para PC¹** (SCIP) consiste en una tarjeta de expansión de 8 bits para un bus AT de una PC-IBM o compatible cuya tarea es realizar el procesamiento de información al mismo tiempo que la CPU de la computadora anfitriona ejecuta otra tarea, llamándose a esto coprocesamiento. El sistema consiste básicamente de un procesador 8088, un coprocesador aritmético 8087, memoria EPROM con rutinas básicas, chips de memoria RAM estática de trabajo y una EPROM para rutinas del usuario, aunado a la lógica adicional de soporte de estos componentes.

La función de la tarjeta coprocesadora es la de realizar el coprocesamiento de información de manera que el procesador principal de la computadora delegue determinadas funciones en esta tarjeta. El sistema planteado es multipropósitos, y puede ser programado por el usuario.

El procesador principal verá al procesador de la tarjeta de expansión como un puerto de entrada/salida.

El proyecto surge ante la necesidad de realizar un procesamiento rápido de datos analógicos adquiridos por medio de algún dispositivo de conversión analógico-digital, pero fue extendido a un sistema de coprocesamiento de propósito general. El sistema de coprocesamiento es útil ya que permite que el procesador principal de la PC delegue determinadas operaciones a la tarjeta coprocesadora. En el sistema presentado no se desarrolla la fase de adquisición de datos, ya que esto no entró dentro de los objetivos del presente trabajo.

¹Computadora Personal, por sus siglas en inglés.

Objetivos Generales del Proyecto.

Se plantea como objetivo general el desarrollo de una tarjeta de expansión para una PC-IBM o compatible que realice coprocesamiento de información.

Objetivos Específicos del Proyecto.

Se plantearon los siguientes objetivos específicos:

1. El diseño y desarrollo de una tarjeta de expansión para computadoras PC-IBM y compatibles con bus AT que realice la ejecución de un programa al mismo tiempo que la CPU de la computadora anfitriona ejecuta otro programa.
2. La utilización del procesador Intel 8088 y su lógica de soporte para la realización del procesamiento, incluyendo memorias RAM y ROM.
3. La utilización del Controlador de Periféricos Programable 8255 para efectuar la comunicación bidireccional entre la tarjeta SCIP y la computadora anfitriona.
4. El desarrollo de programas básicos que permitan comunicar a la tarjeta con la computadora.

Capítulo 1.

Microprocesadores.

1.1 Introducción.

En este capítulo se hace una breve reseña histórica de los microprocesadores y coprocesadores y se presenta una introducción a los microprocesadores Intel 8088 y 8086.

1.2 Generalidades.

1.2.1 Evolución del Microprocesador.

El microprocesador es la parte fundamental de una microcomputadora. Un microprocesador es la *Unidad de Procesamiento Central* (*CPU* por sus siglas en inglés), que controla la operación de la computadora. Toma una instrucción codificada en binario de la memoria, decodifica las instrucciones en series de acciones simples, y maneja dichas acciones. La CPU contiene una *Unidad Aritmética y Lógica* (ALU) que realiza diversas operaciones aritméticas y lógicas sobre palabras binarias. La CPU también contiene un *apuntador a instrucciones* (IP o *instruction pointer*) usado para cargar la dirección de la siguiente instrucción o dato de la memoria, registros de propósito general, y circuitería para controlar las señales del bus [HaD1986].

Históricamente, el primer microprocesador fue el Intel 4004, aparecido en Noviembre de 1971. Este era un microprocesador de 4 bits ideado para ser utilizado en una calculadora [HaD1986]. Este procesador funcionaba a una frecuencia de 108Khz [BaJ1996].

Posteriormente, Intel produciría el microprocesador 8008, usado como controlador de terminales. El sucesor de este fue el Intel 8080, aparecido en 1974, que tenía un bus de datos de 8 bits, y un bus de direcciones de 16 bits, teniendo internamente 7 registros de 8 bits. Con este microprocesador

aparecieron los puertos de entrada y salida [BaJ1996].

Intel 4004, 4040
Intel 8008, 8080, 8085
Intel 8048, 8051, 8052
Intel 80x86, Pentium, AMD K5/K6, Cyrix M1, Nx586
Intel 80960
Intel 80860
Intel i432
Motorola MC14500B
Motorola 680x, 6809, Hitachi 6309
Motorola 680x0
Motorola 88000
Motorola DSP96002/DSP56000
AMD 29000
Zilog Z-80, Z-280
Zilog Z-8000, Z80000
Fairchild F8
Fairchild 9440

Tabla 1.1: Algunos microprocesadores.

Un avance sobre el Intel 8080 fue el 8085 (1976), el cual tiene dos instrucciones para deshabilitar o habilitar tres terminales de interrupción, y la terminal de entrada y salida serial. Estas instrucciones son RIM y SIM [CEK]. También tenía el generador de reloj y el controlador de bus integrados. Otras compañías fabricaron pronto microprocesadores de 8 bits, dando inicio la era de las microcomputadoras; entre estos se encuentran el Motorola MC6800, el MOS Technology 6502 (usado como CPU en la microcomputadora Apple II [HaD1986]) y el Zilog Z80 (utilizado como CPU en la microcomputadora Radio Shack TRS-80 [HaD1986]).

Posteriormente han aparecido muchos más microprocesadores, algunos de los cuáles se presentan en las Tablas 1.1 y 1.2 [BaJ1996], y sus usos se han ido extendiendo en otros campos, no solo como CPU's de microcomputadoras. Hall [HaD1986] define dos direcciones de evolución en los microprocesadores (ver Subsecciones 1.2.2 y 1.2.3). Así mismo han aparecido diversas arquitecturas de procesadores, algunas de las cuales se presentan en la Tabla 1.3 [BaJ1996].

Fairchild/Intergraph Clipper
National Semiconductor SC/MP (and COP)
National Semiconductor 320xx, Swordfish
TI 9900
TI TMS320Cx0
MIPS R2000, R3000, R4000, R5000, R8000, R10000
MOS Technologies 650x, Western Design Center 65816
Microchip Technology PIC 16x
RCA 1802
Ferranti F100-L
Western Digital MCP-1600
Signetics 2650
Siemens 80C166
MISC M17
Rekursiv
AT&T CRISP/Hobbit

Tabla 1.2: Algunos microprocesadores (*continuación*).

1.2.2 Controladores Dedicados.

Este tipo de dispositivos tienen aplicación en los más diversos campos, desde el control de hornos de microondas hasta sistemas de rehabilitación para personas discapacitadas. Entre los primeros microcontroladores aparecidos en el mercado están el Texas Instruments TMS-1000 y el Intel 8048. Actualmente existen microcontroladores como los Intel 8051, 8096 y los Motorola MC6801 y MC6811 [HaD1986].

1.2.3 CPU's de Propósito General.

Este es el campo de aplicación más extenso de los microprocesadores, los cuáles son usados en microcomputadoras más poderosas que la mayoría de las minicomputadoras anteriores a su aparición.

PDP-8/Intersil 6100
PDP-11
Data General NOVA/MN601, Eclipse
IBM/Motorola POWER/PowerPC
IBM 801, ROMP
IBM System/360/370/390
TRON
Hitachi SuperH
SPARC
HP PA-RISC
ARM
Patriot Scientific ShBoom
DEC VAX
DEC Alpha
CDC 6600/7600
Berkeley RISC

Tabla 1.3: Arquitecturas.

1.3 Introducción a los microprocesadores Intel 8088 y 80x86.

Los microprocesadores Intel fueron elegidos por IBM para sus microcomputadoras por razones de mercadotecnia y bajo costo comparativamente a otros dispositivos, así como por razones de derechos industriales [BaJ1996]. La primera computadora personal de la IBM aparecería el 12 de agosto de 1981, usando el chip Intel 8088 [QuH1992]. Esta máquina marcó el despegue mercantil de las computadoras multipropósito para el hogar y la oficina (*Computadoras Personales, o PC's*). Las *PC's* comenzaron a explorar la arquitectura abierta, con la posibilidad de añadir *tarjetas de expansión* en la tarjeta principal de la computadora, en ranuras (*slots*) provistas para tal fin en el *bus de expansión*. Actualmente estas tarjetas abarcan una amplia gama de aplicaciones, que va desde tarjetas de video, MODEM's¹ y tarjetas controladoras de diversos dispositivos periféricos, hasta tarjetas de digitalización o especializadas en propósitos científicos.

¹Dispositivos Moduladores-Demoduladores que permiten la comunicación telefónica mediante la computadora.

1.3.1 Coprocesadores matemáticos.

Frecuentemente se asocian a los chips Intel 80x86 circuitos integrados, llamados coprocesadores, cuyo propósito es llevar a cabo operaciones especializadas a alta velocidad, liberando a la CPU principal de determinada tarea [GoJ1991]. El caso más común es el de los coprocesadores de punto flotante, encargados de realizar operaciones de punto flotante a alta velocidad. Históricamente uno de los primeros coprocesadores matemáticos fue el AMD 9511.

Microprocesador	Coprocesador
8088	8087
8086	8087
80286	80287
80386	80387

Tabla 1.4: Microprocesadores y Coprocesadores en el caso de computadoras basadas en Intel 8088/80x86.

Los coprocesadores matemáticos utilizados dependen del microprocesador principal de la computadora. Estas dependencias se presentan en la Tabla 1.4 [GoJ1991].

El uso de los coprocesadores no se limita a las operaciones de punto flotante. Existen coprocesadores con aplicación en adaptadores gráficos (*coprocesadores gráficos*), como los coprocesadores gráficos 34010 o 34020 de Texas Instruments [QuJ1992].

Las especificaciones del coprocesador matemático 8087 se presentan brevemente en el Apéndice B y más ampliamente en [Int1989], mientras que en la Sección 2.7 se presenta una breve descripción del mismo.

1.3.2 Arquitectura interna de los microprocesadores Intel 8088 y 8086.

Las microcomputadoras IBM han estado basadas en los microprocesadores Intel 8088 y 80x86. Se les llama *compatibles con IBM* a las microcomputadoras con organización y arquitecturas compatibles con las de dicha empresa. En el Apéndice B se presenta una breve descripción del 8088 y su hoja técnica tomada de [Int1989].

Figura 1.1: Sistema 8086/8088 básico en modo mínimo.

Las primeras microcomputadoras de IBM utilizaban el microprocesador Intel 8088 y las *tarjetas madres* o *motherboards* de ambas (8088 y 8086) son similares [GoJ1991]. El esquema de conexión en modo mínimo se presenta en la Figura 1.1 y para el caso del modo máximo se presenta en la Figura 1.2 [HaD1986].

Esencialmente el microprocesador 8088 y el 8086 son iguales en lo referente a sus terminales. Ambos son microprocesadores con registros de 16 bits.

El Intel 8088 es un microprocesador de tecnología HMOS, en un circuito integrado de 40 terminales. Su conjunto de instrucciones es compatible con el 8086. La tarjeta principal de una computadora basada en 8088 u 8086 tiene un conector para el coprocesador matemático 8087. En la PC original hay 5 ranuras de expansión, número que crece en la XT a 8 (cabe señalar que en las microcomputadoras actuales el número de ranuras de expansión puede variar de un modelo a otro). El control de interrupciones lo lleva a cabo el Intel 8259A, un controlador de interrupciones programable. El Intel 8255A

Figura 1.2: Configuración “Modo Máximo” del 8088.

es la interfaz al teclado [GoJ1991]. El Intel 8272A es el circuito integrado utilizado como controlador de las unidades de discos flexibles. El Intel 8037A es el controlador de acceso directo a memoria (DMA) usado para manejar las transferencias en bloque de entrada y salida de datos, y el Intel 8253, que es un contador y temporizador usado como contador de eventos [GoJ1991]. El circuito encargado de generar los pulsos de reloj para el microprocesador y otros dispositivos es el 8284A.

La arquitectura interna del 8086/8088 en diagrama de bloques se puede ver en la Figura 1.3 [HaD1986].

El 8086 y el 8088 están divididos en dos partes funcionales independientes, las cuáles son la *unidad de interfaz de bus* (BIU) y la *unidad de ejecución* (EU).

Estos procesadores se basan en el diseño del 8080/8085. Tienen un conjunto de registros similares, pero de 16 bits. La BIU alimenta del flujo de instrucciones a la EU, a través de una cola precargada de 6 u 8 bytes, de manera que la toma de instrucciones y la ejecución son en cierta forma con-

Figura 1.3: Diagrama de bloques del 8088 y 8086

currentes. Las instrucciones del 8088/8086 varían de longitud de entre 1 y 4 bytes.

El procesador tiene 4 registros de propósito general (llamados AX, BX, CX, DX), también accesados como registros de 8 bits (AH—AL, BH—BL, CH—CL, DH—DL), y 4 registros índice de 16 bits (SI, DI, SP, BP), además de 4 registros de segmento (DS, CS, ES, SS), que permiten acceder al CPU hasta 1Mb de memoria. Así mismo tiene un *registro de banderas* (ver Figura 1.4), de 16 bits, siendo los 8 inferiores compatibles con el 8085. Solo 9 bits son utilizados, 6 de las cuales indican condiciones producidas por la ejecución de una instrucción, y 3 controlan alguna operación de la unidad de ejecución. Las banderas que componen a este registro se presentan en la Tabla 1.5.

El 8088 tiene puertos de entrada y salida de 64K 8-bits (o 32K 16-bits) y vectores de interrupción fijos.

El direccionamiento de memoria con los microprocesadores 8088/8086 se

Figura 1.4: Formato del registro de banderas del Intel 8088.

realiza sumando el contenido de los registros de segmento, una vez que se han corrido a la izquierda 4 bits, a la dirección del desplazamiento [BaJ1996].

1.3.3 Inicialización y reinicialización del procesador.

La inicialización o arranque del microprocesador 8088 es realizado mediante la puesta a un nivel lógico ALTO de la terminal RESET. Esta terminal debe estar en ALTO al menos cuatro ciclos de reloj. El procesador terminará sus operaciones en el borde de subida del RESET. En la transición de bajada de la señal de RESET el procesador activa una secuencia de reinicio interno de aproximadamente siete ciclos de reloj de duración. Después de esto, el procesador opera normalmente, comenzando con la instrucción localizada en la dirección absoluta FFFF0h. La entrada RESET es sincronizada internamente con el reloj del procesador. Durante la inicialización, la transición de

Bandera	Nombre	Bit	Función
CF	<i>Carry Flag</i>	0	Vale 1 si una operación de adición o substracción produjo acarreo o préstamo.
PF	<i>Parity Flag</i>	2	Vale 1 si el resultado de una operación de datos tiene un número par de bits con valor 1.
AF	<i>Auxiliary Carry Flag</i>	4	Se activa si se generó de un acarreo generado del cuarto bit de un byte.
ZF	<i>Zero Flag</i>	6	Se activa si el resultado de una operación es cero.
SF	<i>Sign Flag</i>	7	Se activa si el resultado de una operación con números signados es negativo.
OF	<i>Overflow Flag</i>	11	Se activa cuando el resultado de una operación es mayor que el máximo valor que se puede representar con el número de bits del operando destino.
TF	<i>Trap Flag</i>	8	Cuando se activa, el microprocesador ejecuta solo una instrucción a la vez.
IF	<i>Interrupt Flag</i>	9	Solo cuando este bit esta activo el 8088 atiende las interrupciones.
DF	<i>Direction Flag</i>	10	Cuando es activado causa que el contenido de los registros de índice disminuya después de cada operación de una cadena de caracteres.

Tabla 1.5: Funciones de las banderas del registro de banderas del Intel 8088.

ALTO a BAJO de la señal RESET debe ocurrir no antes de $50 \mu s$ después del encendido, para permitir la completa inicialización del 8088.

Una Interrupción no Enmascarable, o NMI (ver Subsubsección 2.4.1), ocurrida antes del segundo ciclo de reloj después del fin del RESET no será atendida. Si la NMI es generada después de ese ciclo, y durante la secuencia interna de inicialización, el procesador ejecutaría una instrucción antes de ejecutar la interrupción. Una petición de contención o toma (*hold*) inmediatamente después del RESET será atendida antes de la primera carga de instrucción.

Durante el RESET todas las salidas de 3 estados van al tercer estado (OFF) [Int1989].

1.3.4 Descripción de las terminales del 8088 y 8086.

Figura 1.5: Diagrama de terminales del 8088 en Modo Máximo (Fuente: OrCad).

El diagrama de terminales del 8088 en modo mínimo se presenta en la Figura 1.6, mientras que en la Figura 1.5 se presenta el diagrama del modo máximo. Nótese que estos diagramas no corresponden a todas las 40 terminales de los chips, sino solo a los más relevantes para la operación en cada modo.

Figura 1.6: Diagrama de terminales del 8088 en Modo Mínimo (Fuente: OrCad).

La Tabla 1.6 presenta una breve descripción de las terminales del 8088 en relación con el modo en que se opere (mínimo o máximo).

Los requerimientos para soportar sistemas 8088 mínimo y máximo son diferentes, por lo que no se pueden llevar a cabo con 40 terminales. Debido a esto, el 8088 está equipado con una terminal de selección (MN/\overline{MX} , terminal número 33) la cual define la configuración del sistema. La definición de determinado subconjunto de cambios en las terminales depende del estado de la terminal MN/\overline{MX} . Cuando MN/\overline{MX} está a GND (tierra), el 8088 define las terminales 24 a 31 y 34 en modo máximo. En este modo se requiere

Terminal No.	Modo Mínimo	Modo Máximo
9 - 6	AD7 - AD0	AD7-AD0
2-8, 39	A15-A8	A15-A8
35-38	A19/S6, A18/S5, A17/S4, A16/S3	A19/S6, A18/S5, A17/S4, A16/S3
32	\overline{RD}	\overline{RD}
22	READY	READY
18	INTR	INTR
23	\overline{TEST}	\overline{TEST}
17	NMI	NMI
21	RESET	RESET
19	CLK	CLK
40	V_{cc}	V_{cc}
1, 20	GND	GND
33	MN/\overline{MX}	MN/\overline{MX}
28	IO/\overline{M}	S0
29	\overline{WR}	LOCK
24	INTA	QS1
25	ALE	QS0
27	DT/R	S1
26	\overline{DEN}	S2
31, 30	HOLD, HOLDA	$\overline{RQ}/\overline{GT1}$, $\overline{RQ}/\overline{GT0}$
34	$\overline{SS0}$	-

Tabla 1.6: Terminales del Intel 8088.

que las señales de control del bus sean generadas por el circuito integrado 8288. Cuando la terminal MN/\overline{MX} esta a V_{cc} , el 8088 genera las señales de control de bus por sí mismo de las terminales 24 a 31 y 34.

Tanto el modo mínimo como el modo máximo del 8088 pueden ser usados con bus multiplexado o demultiplexado. La configuración de bus multiplexado (Figura 1.7) brinda al usuario con un sistema de chips mínimo, sin perder poder de procesamiento [Int1989].

El modo demultiplexado requiere un *latch*² para direccionamiento de 64K,

²Circuito que permite almacenar bits, usualmente también se les llama *registros*.

Figura 1.7: Configuración de bus multiplexado.

o dos *latches* para direccionar 1Mb completo. Se puede usar un tercer *latch* para establecer un *buffer*³ si la carga del bus de direcciones lo requiere; un *transceiver*⁴ puede ser usado también en este caso (ver Figura 1.8). Las señales \overline{DEN} y DT/\overline{R} del microprocesador son utilizadas para controlar el *transceiver* y la señal *ALE* es utilizada para habilitar el *latch* para que pueda retener las direcciones [Int1989].

El modo máximo emplea, como se dice antes, el controlador de bus 8288. El 8288 decodifica las líneas de estado $\overline{S0}$, $\overline{S1}$ y $\overline{S2}$, y genera todas las señales

³Memoria intermedia.

⁴Dispositivo que permite comunicar o asilar dos líneas de transmisión de datos, o buses.

Figura 1.8: Configuración de bus demultiplexado (arriba) y de Sistema con buffers completos usando el controlador de bus (abajo).

de control de bus. El 8288 proporciona un mejor control del bus [Int1989] y además libera terminales del 8088 para aumentar las capacidades de éste, tales como permitir la conexión de coprocesadores (como el 8087) al bus local, y configuraciones de bus remoto.

El 8088 requiere que el reloj CLK sea controlado por un generador de reloj externo controlado por un cristal (el generador de reloj y manejador 8284) para sincronizar sus operaciones internas. Las frecuencias del reloj para el 8088 pueden ser de 5 Mhz para el 8088 y de 8 Mhz para el 8088-2.

Modo Mínimo.

En modo mínimo (ver Figura 1.1), las terminales al bus están multiplexados, para poder colocar al 8088 en un circuito integrado de 40 terminales. Utiliza las terminales del número 9 al 16 para el bus de datos (8 bits) y direcciones y del 2 al 8 y del 35 al 39 para el bus de direcciones únicamente. Esto da los 8 bits de datos y 20 de direcciones.

Cabe aclarar que el 8088 es operado en modo mínimo cuando es el único microprocesador en los buses del sistema.

Modo Máximo.

El modo máximo es activado colocando la terminal MN/\overline{MX} en bajo, como se menciona en la Subsección 1.3.4.

Capítulo 2.

Arquitectura basada en el Intel 8088.

2.1 Introducción.

En este capítulo se describen: la operación del microprocesador Intel 8088, las interrupciones, los puertos, y la operación del coprocesador numérico Intel 8087.

2.2 Organización de la Memoria para el 8088/8086.

El procesador 8088 provee direcciones de memoria de 20 bits, divididas en dos componentes, el *segmento* y el *desplazamiento*. La memoria está organizada como un arreglo lineal de hasta 1 Mb. (2^{20}), direccionados de 00000h hasta FFFFFh. La memoria está dividida lógicamente en segmentos de 64Kb. La dirección de inicio de un segmento define su localización. La dirección de un segmento puede estar contenida en uno de los cuatro registros de segmento que tiene el 8088 (ver Subsección 1.3.2). Se utilizan palabras de 16 bits para definir la localización de los segmentos, y para acceder a las localidades de memoria dentro de un segmento, se utiliza otro parámetro, llamado *desplazamiento*. La dirección completa en el espacio de memoria de 1 Mb. se obtiene combinando las direcciones del segmento y del desplazamiento. La combinación se realiza primero haciendo un corrimiento de 4 bits a la izquierda de la dirección contenida en el registro de segmento, introduciendo ceros por la derecha en los bits menos significativos, y luego se suma al resultado la dirección del desplazamiento, obteniéndose la dirección de 20 bits.

Todas las referencias a memoria son hechas relativas a las direcciones base contenidas en los registros de segmento. Toda la información en un segmento comparte los mismos atributos lógicos.

Se pueden colocar operandos de 16 bits en direcciones pares o nones. La BIU automáticamente ejecuta dos ciclos de lectura o escritura para operandos de 16 bits.

Las localidades de memoria de la dirección FFFF0h a la FFFFFh están reservadas para diversas operaciones, como el salto a la rutina de inicialización del sistema. Después de un RESET, el CPU comenzará la ejecución en la localidad FFFF0h, donde debe estar dicho salto. Las localidades de 00000h a 003FFh están reservadas para el vector de interrupciones (ver Sección 2.4). En dichas localidades se colocan apuntadores de 4 bytes que contienen la dirección del segmento y del desplazamiento de 256 posibles rutinas de servicio de interrupciones. Los apuntadores deben ser colocados en aquellas localidades antes de que se ejecute una interrupción. Esto normalmente se hace durante el arranque de la computadora [GoJ1991].

2.3 Operación del bus.

El diagrama de tiempo de la operación básica del bus se presenta en la Figura 2.1.

Figura 2.1: Diagrama de tiempo del sistema básico.

Cada ciclo del bus del procesador consiste de por lo menos cuatro ciclos del reloj (CLK), en la parte superior de la Figura 2.1. Estos son llamados T_1 , T_2 , T_3 y T_4 . La dirección es mandada al bus de direcciones por el procesador durante T_1 y la transferencia de datos se realiza durante T_3 y T_4 , mientras que T_2 es usado para cambiar la dirección del bus durante las operaciones de lectura [Int1989]. Al tiempo que tarda el procesador en cargar y ejecutar una instrucción se le denomina *ciclo de instrucción*. Al tiempo que tarda una operación del bus se le llama *ciclo de máquina*, y este está compuesto de grupos de *estados*, los cuáles son ciclos de reloj de la señal generada por el 8284. Dichos estados se miden del borde de bajada de un pulso de reloj al borde de bajada del siguiente [HaD1986], o bien, del borde de subida de un pulso de reloj al borde de subida del otro.

En caso de que se produzca una indicación de “NOT READY” por parte del dispositivo al que se está direccionando, se insertan estados de espera llamados T_w entre T_3 y T_4 . Estos estados de espera tienen la duración de un ciclo de CLK cada uno. También se puede dar el caso de que ocurran periodos de inactividad (estados “idle”) de CLK entre ciclos del bus local [Int1989].

2.3.1 Ciclo de lectura.

Durante T_1 de un ciclo de máquina de lectura el microprocesador coloca la señal de $\text{IO}/\overline{\text{M}}$ al nivel lógico que corresponda a la operación realizada: ALTO para una operación de entrada/salida de un puerto, o BAJA en caso de lectura de memoria [HaD1986]. En T_1 de cualquier ciclo de bus, la señal de ALE (*Address Latch Enable*) es emitida, ya sea por el microprocesador o por el controlador de bus 8288, dependiendo del modo en que esté trabajando el 8088 (modo mínimo o máximo). La señal ALE es conectada a la entrada de habilitación (STB) de los *latches*, como se muestra en la Figura 1.1. Esto se hace ya que en ese momento el bus en sus líneas AD0-AD7 contiene una parte de la dirección, la cual debe estar en los *latches* para que pueda ser mantenida por el dispositivo direccionado, mientras se coloca el dato en las antes citadas líneas. Las entradas de datos de los *latches* están conectadas a las líneas AD0-AD7, A8-A19 y $\overline{\text{BHE}}$, que es la terminal de habilitación del bus alto (*bus high enable*, ver Apéndice B).

En modo máximo, los bits de estado $\overline{\text{S0}}$, $\overline{\text{S1}}$ y $\overline{\text{S2}}$ son usados por el controlador de bus 8288 para identificar el tipo de operación de bus a realizar (ver Tabla 2.1).

Los bits de estado S3-S6 son multiplexados con los bits de dirección más

$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	Características.
0(BAJO)	0	0	Reconocimiento de Interrupción.
0	0	1	Lectura de puerto.
0	1	0	Escritura a puerto.
0	1	1	Alto (<i>Halt</i>).
1(Alto)	0	0	Cargado de instrucción.
1	0	1	Lectura de datos de memoria.
1	1	0	Escritura de datos a memoria.
1	1	1	Pasivo (no hay actividad en el bus).

Tabla 2.1: Bits de estado $\overline{S_0}$ - $\overline{S_2}$ en modo máximo.

altos y permanecen válidos de T_2 a T_4 . S_3 y S_4 indican cuál registro de segmento fue usado para este ciclo de bus para formar la dirección (Ver Tabla 2.2).

S_4	S_3	Características.
0(BAJO)	0	Segmento extra.
0	1	Pila.
1 (ALTO)	0	Código o ninguno.
1	1	Datos.

Tabla 2.2: Bits de estado S_3 y S_4 .

Una vez que la dirección esta en los *latches*, las líneas AD0-AD7 pueden ser usadas para transferencia de datos de memoria o de un puerto. Al mismo tiempo el microprocesador activa la señal \overline{BHE} y la información de A8-A19 y envía información de estado en esas líneas.

En este punto se llega a T_2 y se puede llevar a cabo una operación de lectura, por lo que \overline{RD} es colocado en bajo, habilitando la salida del dispositivo de memoria o el puerto direccionado, colocándose los datos en el bus (16 u 8 bits).

2.3.2 Ciclo de escritura.

El ciclo de escritura, como ambos ciclos del bus, consiste de cuatro ciclos de reloj (CLK) T_1 , T_2 , T_3 y T_4 . Durante T_1 en un ciclo de máquina de escritura M/\overline{IO} se coloca a un nivel lógico BAJO si la escritura se va a realizar a un

puerto, o a ALTO si la escritura se realizará a memoria. La línea *address latch enable* (ALE) se coloca en alto para habilitar los *latches* de direcciones (en caso de que el 8088 esté operando en modo mínimo). Las líneas AD0-A19 tienen en ese momento la dirección a donde se escribirá, y también es habilitado el $\overline{\text{BHE}}$ en BAJO.

Cabe señalar que en el caso de una lectura o escritura a puerto, las líneas A16-A19 estarán siempre en un nivel lógico BAJO; esto se debe a que las direcciones de puerto solo constan de 16 bits [HaD1986].

Los *latches*, además de retener las direcciones, sirven como un *buffer* para las líneas de dirección. Una vez que la dirección está en los *latches*, el microprocesador coloca el dato deseado en el bus, activando la señal $\overline{\text{WR}}$ en BAJO, habilitando la memoria o el puerto a donde se escribirá el dato presente en el bus. Antes de esto se desactiva la señal ALE.

La entrada READY (listo) le indica al microprocesador cuando un dispositivo de hardware externo está disponible para cierta operación (de lectura o escritura). En este caso, si ésta entrada está en BAJO, antes o durante T_2 , el microprocesador debe insertar un estado de espera (WAIT) antes de T_3 . Si READY va a ALTO antes de que finalice el estado WAIT, pasa a T_4 cuando aquel acabe. En caso contrario, si READY es BAJO antes de que termine el estado WAIT, se inserta otro estado WAIT. El proceso se repite hasta que READY es puesto en ALTO [HaD1986]. Esta función es útil, ya que si un puerto o un dispositivo de memoria a donde se quiera escribir es muy lento, la inserción de estados WAIT permite a dicho puerto o memoria darle el tiempo suficiente para retener o presentar el dato.

2.4 Interrupciones.

Una *interrupción* es una condición que detiene la ejecución de un programa y coloca el apuntador de instrucciones (IP) apuntando a una localidad de memoria específica en donde se reanudará el procesamiento [GoJ1991] (en el caso del 8088/8086).

Una tabla de 256 elementos contiene apuntadores a las direcciones de los *programas de servicio de interrupción*. Esta tabla se ubica de la dirección 0 a la 3FFh, en un área reservada [Int1989]. Cada apuntador tiene 4 bytes de longitud, 16 bits para el segmento y 16 para el desplazamiento de la dirección de la rutina de servicio de interrupción. Un dispositivo (o programa) que origina una interrupción proporciona un número de 8 bits indicando el

tipo de ésta, con el cuál se obtiene la dirección del vector correspondiente. Esto se hace multiplicando el número de interrupción por 4, para obtener el desplazamiento del vector de interrupción.

En el caso del microprocesador Intel 8088 u 8086, las interrupciones pueden deberse a tres orígenes:

1. Una señal externa aplicada a la terminal NMI (*NonMaskable Interrupt*) o a la terminal INTR (*interrupt*) del microprocesador. En ambos casos se les llama *interrupciones de hardware* (ver Subsección 2.4.1) Esta interrupción es la número 2.
2. La ejecución de una instrucción de interrupción INT por parte de un programa. A esto se le llama *interrupción de software*.
3. Alguna condición producida en el microprocesador debida a la ejecución de una instrucción, como por ejemplo, una división entre cero [HaD1986]. A este tipo de *interrupciones condicionales* se les considera interrupciones por software. El número de esta interrupción es el 0.

2.4.1 Interrupciones por hardware.

Las interrupciones por hardware se dividen en *enmascarables* y *no enmascarables*.

Interrupciones no enmascarables.

El microprocesador 8088 tiene una terminal para interrupción no enmascarable, la cuál tiene prioridad más alta que la terminal de petición de interrupción (INTR). Se usa comúnmente para activar una rutina para casos de falla de poder. La terminal NMI es activada en la transición de BAJO a ALTO [Int1989].

Interrupción enmascarable.

El 8088 tiene una sola terminal de entrada de petición de interrupción (INTR) que puede ser enmascarada por software internamente, limpiando el bit de bandera de interrupción (IF). Esta señal es disparada en nivel. Es sincronizada internamente durante cada ciclo de reloj, en el borde de subida de

la señal CLK. Para poder responder a ella, INTR debe estar en un nivel lógico ALTO durante el periodo de reloj que precede al final de la instrucción actual o el final de un movimiento, en el caso de una instrucción de tipo bloque. Durante la secuencia de respuesta a la interrupción, las interrupciones son deshabilitadas.

Figura 2.2: Secuencia de reconocimiento de interrupción.

Durante la secuencia de respuesta (ver Figura 2.2), el microprocesador ejecuta dos ciclos de reconocimiento de interrupción sucesivos. El 8088 emite la señal de LOCK (solo en modo máximo) de T_2 del primer ciclo de bus, hasta T_2 del segundo. una petición de contención o toma (*hold*) del bus local no será atendida sino hasta el final del segundo ciclo de bus. En el segundo ciclo de bus se toma un byte del sistema de interrupción externo (por ejemplo, del controlador de interrupciones 8259) el cuál identifica el tipo de la interrupción. Este byte es multiplicado por cuatro y usado como un apuntador al vector de interrupción, como se mencionó anteriormente.

2.4.2 Respuesta ante una interrupción.

Al final de cada ciclo de instrucción el microprocesador revisa si hay peticiones de interrupción. Si es así, el microprocesador realiza las siguientes

acciones:

1. Disminuye el apuntador de la pila en dos y pone el contenido del registro de banderas en el *stack* (pila).
2. Deshabilita la entrada INTR colocando en 0 la bandera de interrupción del registro de banderas.
3. Limpia la bandera de *trap* en el registro de banderas (ver Sección 1.3.2).
4. Disminuye el apuntador de la pila en dos y guarda en la pila el contenido del registro de segmento de código.
5. Disminuye el apuntador de la pila en dos y guarda en la pila el IP (apuntador a instrucción) actual.
6. Realiza un salto lejano indirecto al inicio de la rutina de interrupción [HaD1986], tomando la dirección del vector de interrupción correspondiente al número de interrupción solicitada.

2.5 Puertos de Entrada/Salida.

Las operaciones de entrada y salida permiten a la computadora obtener y enviar datos.

En el 8088 las operaciones de entrada y salida pueden direccionar hasta un máximo de 64k en registros. Las direcciones de entrada y salida aparecen en el formato de direcciones de memoria en las líneas del bus de direcciones A_0 - A_{15} . Las líneas restantes (A_{19} - A_{16}) están a un nivel lógico BAJO en operaciones de E/S (I/O). Esto define un espacio de direcciones tal como se muestra en la Figura 2.3.

Las instrucciones de entrada y salida variables usan el registro DX del procesador como apuntador, teniendo una capacidad de direccionamiento completa, mientras que las instrucciones de entrada y salida directas direccionan uno o dos de las 256 localidades de bytes de E/S en la página 0 del espacio de direccionamiento de E/S. Los puertos de E/S son direccionados de la misma forma que las localidades de memoria [Int1989].

En la arquitectura basada en el Intel 8088 solo se utilizan 10 bits para direccionar puertos o dispositivos de entrada y salida. Estas líneas son A_0 - A_9 .

Figura 2.3: Espacio de direcciones de puertos de entrada/salida

Figura 2.4: Espacio de direcciones de puertos de entrada/salida en la tarjeta madre

Figura 2.5: Espacio de direcciones de puertos de entrada/salida para las ranuras de expansión

Como se menciona arriba, la Figura 2.3 presenta la distribución del espacio de direcciones de puertos de entrada y salida (E/S). El bloque 1, que comprende el rango 0000H-01FFH reside en la tarjeta madre y se utiliza para direccionar los dispositivos de soporte de la CPU (ver Figura 2.4). El bloque 2, que comprende el rango 0200H-03FFH se utiliza para direccionar los puertos de entrada y salida, decodificando las líneas del bus de la PC.

Este rango es utilizado para direccionar las tarjetas de expansión que residen en las ranuras de expansión de la tarjeta madre (ver Figura 2.5). El bloque 3 (0400H-FFFFH) no es utilizado.

Las formas de selección de dispositivos se discuten en la Subsección 3.4.1.

2.6 Bus del Sistema de la IBM PC.

El bus del sistema de la IBM PC y compatibles es una extensión del bus del microprocesador 8088/8086. El bus del sistema es demultiplexado y proporciona señales adicionales para soporte de Acceso Directo a Memoria (DMA), interrupciones, etc. Todas las señales tienen nivel TTL y hay terminales de GND (tierra) y voltaje. La Figura 2.6 presenta la configuración de las ranuras de expansión de la PC AT, el cual tiene 62 terminales, las cuales son descritas a continuación [DoT]:

A0 - A19 Estas terminales corresponden a las direcciones de memoria y E/S (entrada/salida).

D0 - D7 Estas terminales conforman el bus de datos bidireccional. Durante el ciclo de escritura del 8088 el procesador coloca datos en el bus de datos antes del borde hacia arriba de la señal $\overline{\text{IOW}}$ o $\overline{\text{MEMW}}$, la cual pondrá los datos en el puerto de salida o memoria. Durante el ciclo de lectura, el puerto de entrada o la memoria deben poner los datos antes del borde ascendente de la señal $\overline{\text{IOR}}$ o $\overline{\text{MEMR}}$, las cuales ponen los datos en el 8088.

$\overline{\text{MEMR}}$, $\overline{\text{MEMW}}$, $\overline{\text{IOR}}$, $\overline{\text{IOW}}$ Estas señales son activas en bajo y controlan las operaciones de lectura y escritura antes mencionadas. Son generadas por el 8088/8086 o bien por el controlador de DMA.

ALE Esta señal es el Address Latch Enable e indica el comienzo de un ciclo de bus del 8088.

AEN Esta señal es el Address Enable y es enviada por el controlador de DMA para indicar que un ciclo de DMA esta activo.

OSC, CLOCK OSC es un reloj del sistema de alta velocidad con un periodo de 70 ns (14.31818 MHz) y un ciclo del 50%. La señal de CLOCK dura un tercio de la frecuencia del oscilador (4.77 Mhz), es decir, tiene un periodo de 210 ns y un ciclo del 33%.

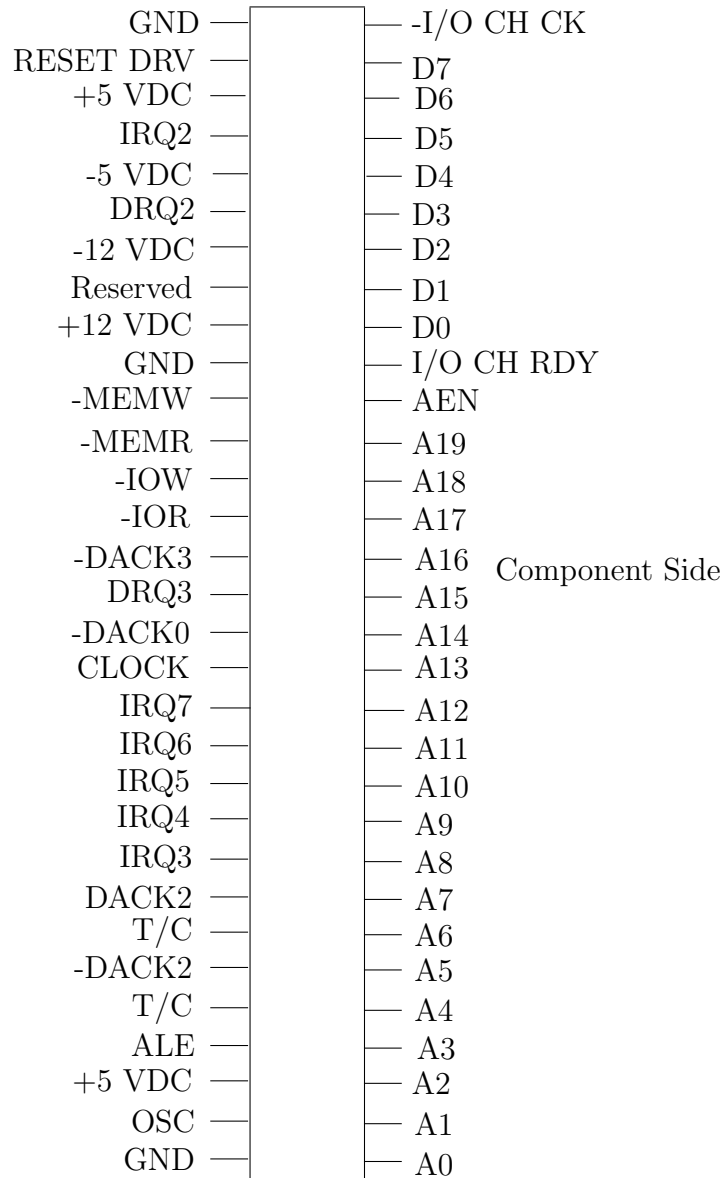


Figura 2.6: Diagrama de una ranura de expansión de la PC IBM (bus AT).

IRQ2 - IRQ7 Los dispositivos de entrada/salida usan estas seis líneas de entrada para generar las peticiones de interrupción hacia el 8088. La prioridad es mayor en IRQ2 y descende hasta IRQ7. Cabe señalar que la línea $\overline{\text{INTA}}$ (Interrupt Acknowledge) no esta disponible en el bus

del sistema, por lo que el reconocimiento de interrupción es realizado generalmente por un puerto de E/S usando una instrucción OUT que es ejecutada por la rutina de servicio de interrupción.

I/O CH RDY Esta es la señal *I/O channel ready* (canal de E/S listo). Esta es una señal de entrada que es usada para generar estados de espera, lo que extiende la longitud de los ciclos de bus del procesador, en el caso de tener que trabajar con dispositivos lentos de entrada/salida o de memoria.

I/O CH CK La señal *I/O channel check* sirve para informar al procesador que hay un error de paridad en una memoria o dispositivo de E/S.

RESET DRV La señal *reset drive* es usada para reiniciar o inicializar el sistema después del encendido o en caso de que el suministro de energía caiga por debajo de los niveles de operación válidos. Esta señal esta sincronizada con el borde de bajada de OSC.

DRQ1 - DRQ3 Las señales de entrada *DMA request* (petición de DMA) son canales asíncronos que son usados para realizar peticiones de acceso directo a memoria o para refrescar memorias dinámicas (DACK0).

T/C La línea *terminal count* envía un pulso cuando el conteo final para el canal de DMA es alcanzado.

2.6.1 Consideraciones especiales sobre poder.

Existen tres limitantes principales en lo que se refiere al suministro de energía a una tarjeta de expansión [DoT]:

1. **Limitantes de poder:** La fuente típica de una PC proporciona una corriente de 4 A a las tarjetas en los *slots* o ranuras de expansión, por lo que si todas las ranuras tienen tarjetas, entonces la corriente se reduce a 1 A o menos.
2. **Desacoplamiento de poder:** Es necesario colocar capacitores para desacoplar la entrada de la fuente de poder de +5 V, y así mismo se deben usar capacitores en el diseño. Es recomendado usar capacitores cerámicos o de tantalio en el rango de 10 a 100 nF para aplicaciones con poco consumo y alta frecuencia, y en caso de diseños que hagan

un gran consumo de corriente, usar capacitores cerámicos o de tantalio de 10 a 50 μF . En todos los casos los capacitores son colocados usualmente entre las terminales de tierra y voltaje de dispositivos con gran consumo de corriente (CI's TTL, bus drivers, transceivers, CI's LSI o de gran escala de integración como microprocesadores, etc.). Además las líneas de tierra y voltaje en la tarjeta de expansión deben colocarse paralelamente.

3. **Carga y capacidad del bus:** Para evitar problemas ocasionados por sobrecargar al bus se recomienda que:

- No se coloquen circuitos LSI NMOS directamente al bus del sistema, ya que no pueden tolerar bajas de poder en mismo.
- No se coloquen más de dos LSI TTL conectados a una señal del bus.
- No se coloquen señales del bus a distancias largas dentro de la tarjeta, ya que esto ocasionaría capacitancias indeseadas que podrían retrasar o alterar tales señales.

2.7 Extensión de procesador numérico.

El circuito integrado Intel 8087 es el coprocesador numérico utilizado con los microprocesadores 8088/8086. Este circuito brinda muchas ventajas en cuanto a desempeño a los sistemas que lo utilizan.

2.7.1 Ventajas.

El coprocesador numérico Intel 8087 proporciona un aumento en la velocidad para la realización de operaciones numéricas. El manual de Intel ([Int1989]) señala que dicho aumento es del orden de 100 veces. Godfrey ([GoJ1991]) presenta la tabla comparativa (Tabla 2.3) que ilustra el aumento de velocidad relativa al 8086 de varias operaciones usando el 8087.

La Tabla 2.4 ([Int1989]) presenta los tiempos de ejecución de diversas instrucciones numéricas. Se comparan los tiempos de un par 8086/8087 y un 8086 con emulación de 8087 por software. Los tiempos de ejecución se presentan en microsegundos y son aproximados.

Tipo de instrucción	Aumento relativo de velocidad respecto al 8086
Multiplicación de simple precisión	84
Multiplicación de doble precisión	78
Suma	94
División de simple precisión	82
Comparación	144
Carga de simple precisión	189
Almacenamiento de simple precisión	67
Raíz cuadrada ¹	544
Tangente ²	144
Exponenciación ³	171

Tabla 2.3: Aumento de velocidad usando un 8087.

Inst. de punto flotante	8086/8087 (8Mhz)	Emulación
Sumar/Restar	10.6	1000
Multiplicar (Precisión sencilla)	11.9	1000
Multiplicar (Precisión extendida)	16.9	1312
Dividir	24.4	2000
Comparar	-5.6	812
Carga (Doble precisión)	-6.3	1062
Almacenar (Doble precisión)	13.1	750
Raíz cuadrada	22.5	12250
Tangente	56.3	8125
Exponenciación	62.5	10687

Tabla 2.4: Comparación entre 8086/8087 (izq.) y 8086 emulando a 8087 (der.).

Además de representar un beneficio en cuanto a velocidad, el uso del 8087, y de coprocesadores numéricos en general, también representa un beneficio en capacidad. En el caso del 8087, su rango dinámico interno es de 10^{4932} ⁴,

⁴Se refiere al **real temporal**, el cuál consiste de 64 bits de mantisa, 15 bits de exponente corrido y 1 bit de signo. Se dice que el exponente es corrido dado que a un exponente dado se le debe sumar el número representado por los bits de exponente (0011111111111111 = 3FFFH) al exponente original [HaD1986], pp. 382-384.

gracias a sus registros internos de 80 bits. El 8087 proporciona los tipos de datos mostrados en la Tabla 2.5.

Formatos de Datos	Rango	Precisión
Palabra entera	10^4	16 bits
Entero corto	10^9	32 bits
Entero largo	10^{18}	64 bits
BCD empaquetado	10^{18}	18 dígitos
Real corto	$10^{\pm 38}$	24 bits
Real largo	$10^{\pm 308}$	53 bits
Real temporal	$10^{\pm 4932}$	64 bits

Tabla 2.5: Tipos de datos del 8087.

2.7.2 Arquitectura Interna.

El 8087 tiene 8 registros internos (ST0 - ST7) de 80 bits de capacidad, además de 1 registro de palabra de estado, un registro de palabra de control, un registro para etiquetar y cierto espacio para apuntadores a excepciones.

El 8087 esta internamente compuesto por dos unidades: de Control (CU) y de Ejecución Numérica (NEU). La primera (CU) mantiene la sincronización entre el 8087 y el 8088/8086 anfitrión. La Unidad de Control “monitorea” la señal de estado ($\overline{S0}$ - $\overline{S2}$, S6) del 8088 para determinar si una instrucción del 8087 es cargada por la CPU. La CU del 8087 mantiene una cola de instrucciones idéntica a la cola ubicada en la CPU anfitriona. La longitud de la cola es determinada automáticamente por la CU del 8087 después del *reset*, por medio de un chequeo en la línea $\overline{BHE}/S7$. La sincronización en la carga y decodificación de instrucciones la realiza la CU monitoreando las líneas de estado de la cola (QS0, QS1).

La NEU tiene una “ruta de datos” de 84 bits de longitud, divididos de la siguiente forma: 68 bits fraccionarios, 15 bits de exponente y un bit de signo. Cuando la NEU comienza la ejecución de la instrucción, se activa la señal de BUSY del 8087, la cual puede ser usada junto con la instrucción WAIT de la CPU para hacer una resincronización de la CPU anfitriona y el 8087 al completar esta la ejecución de esta instrucción (ver Sección 2.7.4). En la Figura 2.7 se presenta el diagrama de bloques del 8087.

Figura 2.7: Diagrama funcional de bloques del 80x87.

2.7.3 Manejo de Excepciones.

Se le llama *excepción* a una condición de error que se produce al momento de ser ejecutada una instrucción en el coprocesador 8087.

Las excepciones detectadas por el 8087 son 6 y se clasifican según su causa en:

1. Operación inválida - incluye: desbordamiento de pila, *stack underflow*, forma numérica indeterminada, o NAN (es decir, un valor que no representa a un número).
2. *Overflow* - ocasionado por un número de más bits que el tamaño del formato especificado.
3. Divisor cero.
4. *Underflow* - ocasionado cuando el resultado de una operación no es cero, pero es demasiado pequeño para caber en el tamaño especificado.

5. Operando denormalizado - esta excepción ocurre cuando al menos uno de los operandos esta *denormalizado*, es decir, que el operando tiene un exponente pequeño pero no es cero.
6. Resultado inexacto - se ocasiona cuando el resultado no se puede representar exactamente en el formato especificado, lo que afecta el redondeo del resultado según el modo, y la colocación de la bandera para esta excepción.

Las excepciones causan una interrupción en caso de que éstas estén habilitadas, y el bit de la correspondiente excepción en la palabra de control esta desenmascarado. Si no lo están, el 8087 continua la ejecución sin importar si la CPU anfitriona limpia la excepción. Se puede enmascarar una excepción en particular, en cuyo caso el 8087 coloca la excepción en el registro de estado y procede a ejecutar una función de manejo de interrupciones interno, con lo que se puede continuar el procesamiento.

2.7.4 Configuración y operación.

El 8087 ejecuta instrucciones como coprocesador de una CPU en modo máximo, conectándose en paralelo a la CPU [Int1989], como se ilustra en la Figura 2.8.

Las líneas de estado S_0 - S_2 y QS_0 - QS_1 habilitan al 8087 para que monitoree y decodifique instrucciones en sincronización con el 8088. Una vez iniciado, el 8087 puede procesar en paralelo e independientemente de la CPU anfitriona. La señal BUSY del 8087 sirve para sincronizar la CPU y el coprocesador, informándole a aquél que el 8087 está ejecutando una instrucción. La terminal WAIT del 8088 verifica aquella señal para asegurarse de que el 8087 está listo para ejecutar instrucciones subsecuentes. El 8087 puede interrumpir a la CPU cuando detecta una excepción. La línea de petición de interrupción del 8087 se rutea al 8088/8086 mediante un PIC 8259 (“Controlador de Interrupciones Programable”). El coprocesador utiliza las líneas $\overline{RQ}/\overline{GT0}$ del 8088 para obtener el control del bus local para transferencia de datos [Int1989].

La estructura, operación y temporizado del bus del 8087 son idénticos a las de los procesadores 8086/8088 en modo máximo. Las direcciones son multiplexadas con los datos en las líneas AD0-AD15, mientras que las líneas

Figura 2.8: Configuración de un sistema 8088/8087 u 8086/8087.

A16-A19 son multiplexadas con las líneas de estado S3-S6. De estas líneas de estado, S3, S4 y S6 está siempre en ALTO y la línea S5 está siempre en BAJO durante los ciclos donde el 8087 maneja el bus. S6 es monitoreada cuando el 8087 se encuentra monitoreando los ciclos de bus de la CPU (modo pasivo), con el objeto de diferenciar la actividad del 8088 de la de entrada/salida local del coprocesador, o de cualquier otro maestro del bus local. Finalmente, S7 tiene siempre el mismo valor que $\overline{\text{BHE}}$.

El tipo de ciclo de bus es determinado por el controlador de bus 8288 usando las líneas $\overline{\text{S0-S2}}$ según la Tabla 2.6 ([Int1989]).

Las instrucciones del 8087 aparecen como instrucciones ESCAPE a la CPU. Dichas instrucciones son decodificadas y ejecutadas por la CPU y el 8087 al mismo tiempo. Si la instrucción leída no es una del 8087, este actúa como si hubiera sido una instrucción NOP. La ejecución de una instrucción del conjunto de instrucciones del 8087 se realiza cuando la CPU anfitriona ejecuta la instrucción ESCAPE. Cabe señalar que toda instrucción del 8087 tiene el código 11011 en los bits mas significativos de su primer byte de código.

S2	S1	S0	
0	×	×	No usado
1	0	0	No usado
1	0	1	Lectura de datos de memoria
1	1	0	Escritura de datos a memoria
1	1	1	Pasivo (no hay ciclo de bus)

Tabla 2.6: Líneas de estado que determinan el tipo de ciclo de bus.

Intel señala que la instrucción puede o no identificar a un operando de memoria, pero que la CPU “distingue” entre instrucciones de ESCAPE que hacen referencia a memoria e instrucciones que no hacen tal cosa. Si la instrucción se refiere a un operando de memoria, la CPU calcula la dirección del operando usando alguno de los modos de direccionamiento y luego realiza una simulación de lectura de la palabra en tal localidad, mediante un ciclo de lectura normal, salvo que la CPU ignora los datos, los cuales son leídos del bus de datos por el 8087; y si la instrucción ESCAPE no tiene una referencia a memoria, la CPU procede a cargar la siguiente instrucción.

Los modos de referencia de memoria del 8087 son tres:

1. No hay referencia a memoria.
2. Carga de operando de memoria al 8087.
3. Almacenamiento de operando del 8087 a memoria

En caso de que el 8087 necesite leer o escribir en memoria un dato de más de una palabra (hasta 80 bits), el 8088 envía la dirección de la primera palabra de datos en el bus de direcciones, y la señal de control adecuada, según se trate de una operación de lectura o escritura. El 8087 lee el dato del bus de datos o lo escribe a memoria, y toma la dirección física (20 bits) que envió el 8088, con el fin de tomar el control del bus para realizar la transferencia del resto de las palabras entre él y la memoria. Esto lo hace poniendo en la terminal $\overline{RQ}/\overline{GT0}$ del 8088 un pulso BAJO, con lo que el 8088 envía un pulso BAJO a la terminal $\overline{RQ}/\overline{GT0}$ del 8087 y pone sus buses en estado de alta impedancia. El 8087 incrementa entonces la dirección de 20 bits que tomó durante la primera transferencia, y la coloca en el bus de direcciones, posibilitando la lectura o escritura de la siguiente palabra. Este proceso es continuado hasta que el 8087 termina de leer o de escribir las

palabras de los datos. Una vez concluida esta operación, el 8087 envía un pulso bajo a la terminal $\overline{RQ}/\overline{GT0}$ del 8088 para que este pueda tomar el control de los buses nuevamente [HaD1986].

El 8087 debe asegurarse de que el 8088 no ejecute la siguiente instrucción antes de haber acabado él mismo de ejecutar la instrucción en curso. Esto se puede presentar (A) cuando el 8088 necesita datos producidos por el 8087 para ejecutar la siguiente instrucción, o (B) cuando se ejecutan varias instrucciones del 8087 en secuencia.

En el primer caso, lo que se hace es conectar la terminal de salida BUSY del 8087 a la terminal \overline{TEST} del 8088 y colocar una instrucción WAIT en el programa. Esta ocasiona que el 8088 verifique la terminal \overline{TEST} y entre en un ciclo (“loop”) interno en el que se verifica repetidamente \overline{TEST} hasta que esta se hace ALTO, lo que indica que el 8087 ha completado la ejecución de la instrucción.

En el segundo caso, el 8087 solo puede ejecutar una instrucción a la vez, así que se debe asegurar que el 8087 a completado la ejecución de una instrucción antes de leer la siguiente de memoria. Esto se realiza utilizando las terminales BUSY del 8087 y \overline{TEST} del 8088 y la instrucción FWAIT (WAIT) del 8088, que se coloca después de cada instrucción del 8087 para asegurar que el 8088 quede en un ciclo interno verificando \overline{TEST} una vez que esta ha sido puesta en BAJO por la terminal BUSY del 8087 (al igual que en el primer caso).

La Figura 2.9 muestra un diagrama de flujo simplificado de la operación del 8087 y el 8088 (se omiten de dicho diagrama los casos de excepción).

Figura 2.9: Diagrama de flujo de la operación del 8088 y 8087 sincronizados.

Capítulo 3.

Diseño y Descripción del Sistema.

3.1 Introducción.

Este capítulo presenta la descripción del diseño de la tarjeta SCIP. La tarjeta SCIP fue ensamblada usando bases para alambramiento¹ sobre una tarjeta para prototipos de 16 bits JDR-PR10 de JDR Microdevices, la cual se monta, para propósitos de prueba y mediciones, sobre una tarjeta de extensión EXT-80286. La tarjeta JDR-PR10 tiene integrado un diseño para la decodificación y generación de las líneas de selección de la tarjeta, y “buffers” para los buses de dirección, datos y otras líneas provenientes de conectores de expansión de la PC. Sin embargo, dado que el diseño de JDR utiliza PAL’s (Arreglos Lógicos Programables) en la lógica de decodificación, los cuales no se pudieron obtener, se realizó dicha lógica independientemente.

El diseño se divide en 3 módulos principales:

1. Módulo de procesamiento.
2. Módulo de memoria.
3. Módulo de comunicación entre la tarjeta SCIP y la computadora.

El diagrama por bloques se presenta en la Figura 3.1.

En éste capítulo se hace referencia a las hojas de diagramas presentadas en el Apéndice C.

¹Estas bases para alambramiento o wire-wrap proporcionan la ventaja de que no es necesario soldar postes extra (como sería el caso si se usaran sockets para C.I.’s), ahorran espacio y son fáciles de alambrear o desalambrear los circuitos ensamblados con ellas. Sin embargo elevan el costo del diseño, ya que son de 3 a 5 veces mas caras que un socket normal para integrados.

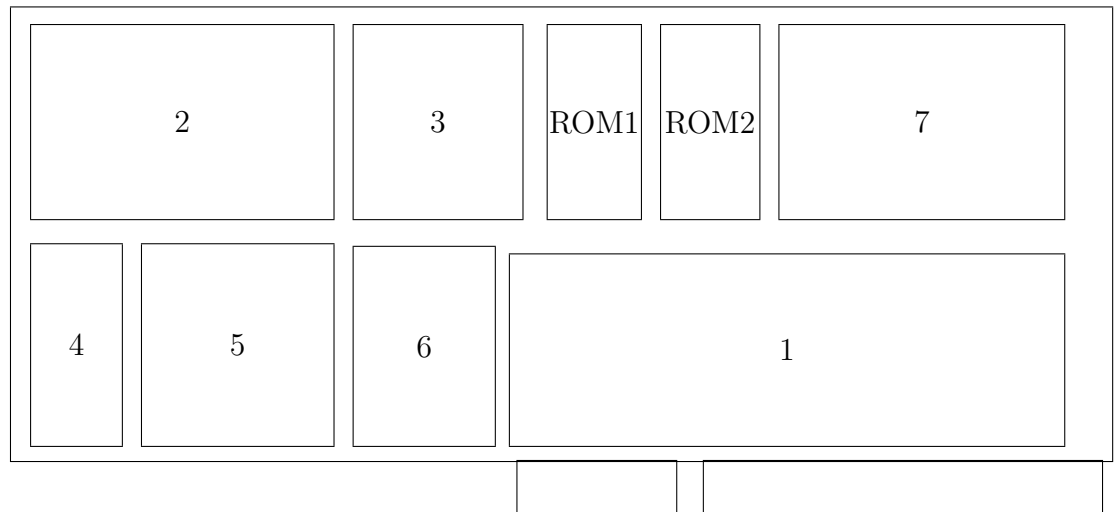


Figura 3.1: Diagrama en bloques de SCIP: (1) Sección preconstruida de decodificación, (2) Módulo de Procesamiento y Control de BUS, (3) Módulo de decodificación interna y buffers, (4) Módulo de temporización, (5) Módulo de comunicación, (6) Módulo de lógica de selección, (7) Módulo de memorias RAM, (ROM1) Memoria EPROM 1, (ROM2) Memoria EPROM 2.

3.2 Módulo de procesamiento.

La función del módulo de procesamiento es la de ejecutar el programa de usuario que reside en una memoria EPROM dentro de la tarjeta SCIP.

Este módulo consta de las siguientes partes principales:

- Un microprocesador Intel 8088.
- Un coprocesador matemático Intel 8087.
- Un temporizador Intel 8284.
- Un controlador de interrupciones programable 8259.
- Un controlador de bus 8288.
- Latches para las líneas de datos/direcciones.
- Circuito para generar el RESET de este módulo.

Esta parte del diseño se puede ver en las hojas de diagrama números 4 (“Módulo de generación de líneas de control/IRQ’s”) y 1 (“Módulo de Procesamiento”) del Apéndice C.

Como se ve en la hoja 1, el microprocesador Intel 8088 esta conectado en modo máximo (el cual se define en la Subsubsección 1.3.4, la Subsección 1.3.2 y la Subsección 1.3.4), para poder soportar la utilización del coprocesador matemático Intel 8087. La conexión de los buses de datos y direcciones de ambos se realiza en paralelo (ver Subsección 2.7.4). Así mismo la conexión de las líneas de estado S1-S3 de ambos circuitos se realiza en paralelo, para poder decodificar el estado del bus en un momento dado. Estas líneas de estado se conectan a las entradas correspondientes del 8288 (terminales 3, 18 y 19), el cuál se encarga de la generación de las líneas de control del bus ($\overline{\text{MRD}}$, $\overline{\text{MWT}}$, $\overline{\text{AWM}}$, $\overline{\text{IOR}}$, $\overline{\text{IOW}}$, $\overline{\text{AIOW}}$, $\text{DT}/\overline{\text{R}}$, DEN , $\text{MC}/\overline{\text{PD}}$, y ALE). Las líneas de control que son usadas en SCIP son ALE , $\text{DT}/\overline{\text{R}}$, $\overline{\text{MRD}}$, y $\overline{\text{MWT}}$ para la selección de memoria local, y $\overline{\text{IOR}}$ y $\overline{\text{IOW}}$ para la selección de puertos. Así mismo, se utilizan para controlar el latch 74LS373 y los transceivers 74LS245 (U4 y U5/U6 respectivamente, en la hoja 1 del circuito), para mantener el tráfico en el bus de datos y de direcciones de acuerdo a las especificaciones dadas anteriormente (ver Sección 2.3).

La temporización del sistema se realiza mediante un generador de reloj 8284 (U1 en la hoja 4), con un cristal de 14.31818 Mhz, frecuencia que es dividida por 3 para la adecuada operación del 8088, 8087 y del 8288, la cuál es de $4.74 \approx 5$ Mhz. Este mismo circuito integrado sirve para generar la señal de RESET para el microprocesador 8088 (ver Subsección 1.3.3). El circuito utilizado para la generación de la señal de RESET consta de una resistencia de $640 \text{ K}\Omega$, un capacitor de $360 \mu\text{F}$ y un interruptor de botón. Al presionarse el botón se pone en BAJO la terminal RES del 8284, con lo que éste genera el pulso de RESET para el 8088 (y en consecuencia para el 8087, 8255 y el 8288). Cabe señalar que la duración de la señal de RESET es de importancia, debido a que debe de estar presente al menos durante 4 ciclos de reloj para que el procesador pueda iniciar su operación normalmente [Int1989].

En el diseño realizado es necesario el uso de las líneas AEN2-AEN1 y RDY2-RDY1 ya que estas se necesitan para configuraciones con múltiples maestros de bus; debido a esto, AEN1-AEN2 se colocan en BAJO y RDY1-RDY2 se colocan en ALTO. La terminal $\text{F}/\overline{\text{C}}$ se coloca a BAJO para habilitar la generación de la señal del reloj por medio del cristal.

La terminal $\overline{\text{TEST}}$ del 8088 se conecta a la terminal BUSY del 8087, para hacer posible la sincronización de ambos dispositivos, como se señala en la

Subsección 2.7.4. Con el mismo propósito de sincronización se conecta la terminal RQ/ $\overline{\text{GT1}}$ (no. 30) del 8088 con la terminal RQ/ $\overline{\text{GT0}}$ (no. 31) del 8087; en este caso, estas líneas controlan las peticiones o concesiones del bus, para que en caso de que el 8087 necesite tomar el control del bus, se coordine con el 8088.

Como se menciona arriba, el control del bus del sistema interno de la tarjeta, al estar el 8088 conectado en modo máximo, se lleva a cabo con un Controlador de Bus 8288. Las líneas de estado del 8088 (S0-S2) indican el modo de operación del bus (ver Subsección 1.3.4). Estas líneas se conectan a las líneas S0-S2 del 8288, el cual, a través de sus líneas de control determina la operación actualmente llevada a cabo por el bus.

El manejo de las excepciones que puedan ocurrir en el 8087 (ver Subsección 2.7.3) se lleva a cabo mediante un PIC (Controlador de Interrupciones Programable) 8259 (U2 en la hoja 4 del diagrama). La terminal INT del 8087 se conecta a la terminal IR0 del 8259. Esto se hace con el fin de generar una petición de interrupción. Esta se mantiene activa hasta que el 8288 genera una señal de aceptación de interrupción ($\overline{\text{INTA}}$). La terminal INTR del 8088 se conecta al 8259 para en caso de producirse una petición válida de interrupción, el 8259 interrumpa a la CPU 8088. Las líneas del bus de datos de la tarjeta (D0-D7) se conectan a las terminales del bus de datos del 8259 (D0-D7), las cuales son de tres estados, para que por ellas se transmita la información de control, estado y vector de interrupción. Este bus es bidireccional, y proviene del transceiver 74LS245 (U5 en la hoja 1 del diagrama). En el caso de SCIP, el 8259 trabaja como único PIC del sistema, por lo que las líneas CAS0-CAS1 se mantienen desconectadas, y la terminal 16 (SP/ $\overline{\text{EN}}$) se pone a un nivel ALTO.

Durante la inicialización del sistema, el software del BIOS de SCIP realiza la programación del 8259, mediante el código que se presenta a continuación:

```

MOV  DX,3C0h      ; Primer puerto (ICW 1)
MOV  AL,17h       ; Modo de disparo por borde, intervalo de 4,
                  ; sencillo, se requiere IC4
OUT  DX,AL        ; 00010111

MOV  DX,3C1h      ; Segundo puerto (ICW 2)
MOV  AL,0h        ; Inicio del vector de interrupciones
OUT  DX,AL        ; 00000000

```

```
MOV  DX,3C1h      ; Segundo puerto (ICW 3)
MOV  AL,0h        ; Esta palabra es ignorada, ya que
                  ; solo hay un PIC
OUT  DX,AL        ; 00000000

MOV  DX,3C1h      ; Segundo puerto (ICW 4)
MOV  AL,5h        ; Modo no especial completamente anidado
                  ; modo sin buffers, EOI normal,
                  ; modo para 8086/8088
OUT  DX,AL        ; 00000101
```

El programa escribe a los puertos 3C0h y 3C1h en donde se encuentra el 8259 (ver Subsección 3.3.2), y mediante cuatro Palabras de Control para Inicialización (ICW1-ICW4) lo programa:

1. La primera palabra programa el modo de disparo por borde, un intervalo de 4 bytes entre los vectores de interrupción, único PIC del sistema, y palabra de control 4 necesaria.
2. La segunda palabra de control indica al 8259 que el vector de interrupciones inicia en la dirección 0 absoluta.
3. La tercera palabra de control es ignorada, puesto que el 8259 es el único PIC del sistema.
4. La cuarta palabra indica al PIC que trabaje en modo no especial completamente anidado, en modo “sin buffers”, con EOI (Fin De Interrupción) normal, y en modo 8088/8086.

3.3 Módulo de memoria, puertos y decodificación interna.

3.3.1 Memoria interna de SCIP.

La función del módulo de memoria es el almacenar las rutinas de inicialización de la tarjeta, el programa de usuario que residirá en la tarjeta, y proporcionar un área de trabajo en RAM para tal programa.

Este módulo se presenta en las hojas de diagrama 5 y 6 del Apéndice C, y consta de:

- Dos EPROM² NEC 27512 de 64Kb \times 8 bits cada una.
- Cuatro RAM³ estáticas⁴ NEC PD43256A de 32Kb \times 8 bits cada una.
- Decodificadores/demultiplexores 2 a 4 74LS139.
- Compuertas OR 74LS32.
- Decodificadores 1 de 8 74LS138.
- Compuertas AND 74LS08.
- Compuerta AND 74LS10.
- Compuerta NOT 74LS04.

En la tarjeta SCIP se tienen 128 Kb de memoria ROM y 128 Kb de memoria RAM. 64 Kb de memoria ROM son destinados a las rutinas de inicialización de la tarjeta, y el circuito integrado que las contiene permanece en ella siempre (bloque 6, ver Tabla 3.1). Los otros 64 Kb de ROM están destinados al programa de aplicación específico del usuario que reside en la tarjeta, y el circuito integrado puede ser intercambiado para colocar otro programa dependiendo de las necesidades del usuario (bloque 5, ver Tabla 3.1). El circuito integrado de memoria EPROM que permanece en la tarjeta ocupa el rango de direcciones más alto (F0000-FFFFFF hexadecimal, es decir, el bloque de memoria 6, ver Tabla 3.1) ya que dadas la característica del reset e inicialización del 8088, este salta a la dirección FFFF0 hexadecimal (ver Subsección 1.3.3). Por tanto, el bloque 5 (ver Tabla 3.1) de memoria es la EPROM intercambiable que contendrá el programa del usuario.

El mapa de la memoria interna de SCIP se presenta en la Figura 3.2.

²Memoria de Solo Lectura Eléctricamente Programables y Borrables

³Memoria de Acceso Aleatorio

⁴La razón de la utilización de memorias RAM estáticas en lugar de usar memorias dinámicas o es que para estas no se requiere lógica adicional para el refresco de las memorias, y dado que el espacio en la tarjeta para prototipos JDR-PR10 usada es reducido, se prefirió utilizar memorias estáticas. Sin embargo, el costo de estas elevó el costo global del sistema.

Figura 3.2: Mapa de memoria de la tarjeta SCIP.

La Tabla 3.1 presenta los rangos de direcciones utilizados para la memoria de la tarjeta SCIP y la respectiva dirección de 20 bits como se presenta en el bus de direcciones del sistema. En binario se presenta solo la dirección de inicio de cada bloque de memoria.

$A_{19} - A_{16}$	$A_{15} - A_{12}$	$A_{11} - A_8$	$A_7 - A_4$	$A_3 - A_0$	Inicio	Fin	Bloque
0000	0000	0000	0000	0000	00000	07FFF	1
0000	1000	0000	0000	0000	08000	0FFFF	2
0001	0000	0000	0000	0000	10000	17FFF	3
0001	1000	0000	0000	0000	18000	1FFFF	4
1110	0000	0000	0000	0000	E0000	FFFFFF	5
1111	0000	0000	0000	0000	F0000	FFFFFF	6

Tabla 3.1: Rangos de direcciones de los bloques de memoria de SCIP.

La decodificación de las direcciones para la selección de los circuitos in-

tegrados de memoria se lleva a cabo usando decodificadores 74LS139⁵.

Para la selección de los circuitos de memoria RAM se conectan las líneas A15 y A16 del bus de direcciones de SCIP a las entradas A0 y A1, respectivamente, de un 74LS139, como se muestra en la Tabla 3.2. La entrada de habilitación (\overline{E}) de dicho 74LS139 se conecta a la línea A19, ya que en el rango de direcciones de memoria que ocupa la memoria RAM la línea A19 del bus de direcciones de SCIP siempre esta en BAJO (ver Tabla 3.1).

A15 \rightarrow A0	A16 \rightarrow A1	Línea habilitada en el 74LS139
0	0	$\overline{O0}$
1	0	$\overline{O1}$
0	1	$\overline{O2}$
1	1	$\overline{O3}$

Tabla 3.2: Entradas del 74LS139 para la selección de las memorias RAM

Para seleccionar los circuitos de memoria EPROM se utilizan las líneas A15 y A16 del bus de direcciones de SCIP como entrada para A0 y A1 respectivamente en un 74LS139, como se muestra en la Tabla 3.3. La entrada de habilitación del 74LS139 se conecta a la salida de una compuerta 74LS04 la cual invierte la línea A19 del bus de SCIP ya que esta siempre esta en ALTO en el rango de direcciones que comprende las dos memorias EPROM (ver Tabla 3.1).

A15 \rightarrow A0	A16 \rightarrow A1	Línea habilitada en el 74LS139
0	0	$\overline{O0}$
0	1	$\overline{O2}$

Tabla 3.3: Entradas del 74LS139 para la selección de las memorias EPROM

3.3.2 Puertos internos.

La lógica de decodificación presentada en la hoja de diagramas número 5 permite seleccionar hasta 8 puertos dentro del sistema SCIP⁶. Para esta parte del circuito se utilizan los decodificadores U5 y U6 (ambos 74LS138). En U5

⁵El 74LS139 es un decodificador/demultiplexor dual 1 de 4.

⁶Este número se podría aumentar con lógica adicional, usando A3=1 y otro 74LS138, por ejemplo.

se genera la señal Y7 en caso de que la dirección presente en las líneas A4 - A9 del bus de direcciones interno sea $3Cx$. Y7 entonces habilita al decodificador U6 el cuál decodifica las líneas A0, A1 y A2 para la selección de puertos (A3=0).

Los puertos internos de SCIP decodificados por U6 son 2 para el 8259 y 4 para el 8255, como se muestra en la Tabla 3.4.

A9 - A0	Hex.	Disp.
1111000000	3C0	8259
1111000001	3C1	8259
1111000100	3C4	8255
1111000101	3C5	8255
1111000110	3C6	8255
1111000111	3C7	8255

Tabla 3.4: Direcciones de los puertos para los dispositivos PIC y PPI.

Estos puertos se utilizan para programar y acceder tanto al PIC 8259 como al PPI 8255. Es necesario contar con tantos puertos debido a que ambos dispositivos lo requieren así (ver Apéndice B).

3.4 Módulo de comunicación.

La función de este módulo es realizar la comunicación entre la tarjeta SCIP y la computadora en ambos sentidos.

Este módulo se divide en dos partes:

1. Decodificación de líneas de dirección para selección de la tarjeta.
2. Puertos de comunicación PC \Longleftrightarrow SCIP.

3.4.1 Técnicas de decodificación para selección de dispositivos.

Para la decodificación y sensado de muchos dispositivos periféricos y de entrada y salida en la PC se utilizan los puertos (ver Sección 2.5). Para habilitar un dispositivo colocado en una ranura de expansión se necesita decodificar de alguna manera las líneas de dirección del bus de la PC. Esto se puede llenar a cabo de las siguientes formas:

1. Mapeo de memoria.
2. Puertos de entrada/salida.

El primer método consiste en ver cada periférico como una dirección única en la memoria de la computadora, siendo posible con esto el uso de todas las instrucciones que sirven para acceder a memoria.

El segundo método consiste en ver al periférico como un dispositivo de entrada/salida, independiente de la memoria. Con esto se limita el acceso al dispositivo a las instrucciones IN y OUT.

Las formas para seleccionar un dispositivo son:

1. Lógica aleatoria.
2. Decodificadores.
3. Comparadores.
4. Memorias ROM programables (PROM's) de mapeo.

El método de decodificación consiste en utilizar un circuito integrado decodificador tal como el 74LS138⁷ para demultiplexar ciertas líneas de dirección del bus y así generar la señal de habilitación de la tarjeta o dispositivo colocado en la ranura de expansión. Esto tiene la desventaja de que la dirección de la tarjeta se fija desde su diseño y podría entrar en conflicto con otro dispositivo.

El método de comparadores utiliza circuitos integrados de comparación, tales como el 74LS85⁸ para comparar una dirección fija (que puede ser modificada mediante interruptores⁹ o *jumppers* en la tarjeta) contra ciertas líneas del bus de direcciones. De esta manera, si la dirección presente en el bus es igual a la fijada en la tarjeta, se genera la señal de habilitación de la ésta. Esto da la ventaja de que no se restringe la habilitación de la tarjeta a una dirección permanente, sino que se puede modificar, en caso de que existiera conflicto con algún otro dispositivo.

Los métodos primero (lógica aleatoria) y último (memorias PROM de mapeo) no se consideraron en el diseño, ya que complican un tanto la lógica

⁷Decodificador/demultiplexor 1 de 8.

⁸Comparador de magnitud de 4 bits.

⁹Del tipo *dip switch*.

de decodificación, al requerir muchos circuitos integrados. Sin embargo, cabe señalar que la técnica de mapeo por memorias PROM tiene la ventaja de que un dispositivo puede ser accesado en cualquier dirección, ya que dicha selección se lleva a cabo utilizando direcciones almacenadas en la PROM para mapear direcciones secuenciales dentro de localidades de memoria diferentes. Estas direcciones se pueden cambiar con solo cambiar la memoria PROM [QuH1992].

3.4.2 Descripción del circuito de decodificación de SCIP.

Esta parte del circuito (la sección de decodificación) se presenta en la hoja 2 de diagramas del Apéndice C.

Se utiliza un dip switch de 8 bits (U5) para fijar la dirección base de la tarjeta (300h), y mediante un comparador¹⁰ 74LS684 (U2) se efectúa la comparación entre la dirección fijada por el dip switch y los bits presentes en las líneas A4 - A5 del bus de direcciones de la PC. Si ambos valores son iguales, la señal $\overline{\text{PORTSEL}}$ es activada y usa como parte de la habilitación de un decodificador 74LS138 (U3). Si al mismo tiempo no está presente la señal AEN del bus de la PC¹¹, el estado de las líneas $\overline{\text{IOR}}$ e $\overline{\text{IOW}}$ sirven para generar la señal de selección de la tarjeta $\overline{\text{BOARDSEL}}$, la cual se activa si Y5 o Y6 de U3 están en BAJO (es decir, si hay acceso a un puerto por parte de la CPU de la PC). Esto se logra mediante un AND en U4A (74LS08).

La señal de $\overline{\text{BOARDSEL}}$ habilita entonces el decodificador U1 (74LS138), el cuál sirve para generar, en conjunción con las líneas de dirección A0, A1 y A2 del bus de la PC, hasta 8 señales para direccionar puertos en la tarjeta SCIP. Las salidas Y0 y Y1 ($\overline{\text{DATAREAD}}$ y $\overline{\text{WRITEDATA}}$ respectivamente, correspondientes a los puertos 300h y 301h de la computadora anfitriona) proporcionan las señales para habilitar la lectura o escritura de datos. Dado que A3 se usa igualmente como habilitador de U1 (cuando A3=0), se puede colocar otro 74LS138 para tener hasta 16 puertos dentro de SCIP directamente (se podría usar lógica adicional para direccionar más puertos).

Para mantener la sincronización entre SCIP y la PC se usa un circuito compuesto por los flip-flops 74LS74 U8A, U7A y U9A, los cuales, mediante

¹⁰La ventaja de usar el comparador es que la dirección base de la tarjeta puede ser cambiada fácilmente.

¹¹Esta señal es activa en ALTO cuando hay acceso directo a memoria.

las señales de RESETDRV y CLOCK provenientes del bus de la PC, por cada selección de la tarjeta generan un pulso en la señal $\overline{\text{IOCHRDY}}$ con duración de dos ciclos de reloj, lo cual genera un retardo en del bus de la PC anfitriona (Ver Sección 2.6).

3.4.3 Descripción del circuito de comunicación de SCIP.

La parte del circuito de SCIP que realiza la comunicación entre la PC y la tarjeta se muestra en la hoja de diagramas número 3 del Apéndice C.

El circuito integrado que es el encargado principal de la comunicación es el controlador programable de periféricos 8255 (ver Apéndice B y Figura 3.3). Este es un dispositivo muy versátil ya que puede ser programado en tres modos principales de operación (Modos 0, 1 y 2), los cuales a su vez pueden ser combinados. Cualquiera de estos modos pueden ser seleccionados mediante software en cualquier momento. Cuenta con 3 puertos de 8 bits cada uno (A, B y C), los cuales se seleccionan mediante las líneas A0 y A1 del bus interno de direcciones. El Puerto A junto con la parte superior del Puerto C (C7 - C4) conforman el Grupo A, mientras que el Grupo B esta compuesto por el Puerto B y la parte inferior del Puerto C (C3 - C0).

La parte del diseño a continuación descrita se puede apreciar en la hoja de diagramas número 5. En dicho diagrama, U6 (un 74LS138) mediante un AND de las líneas Y4, Y5, Y6 y Y7 genera la señal $\overline{\text{PPISEL}}$ la cual se conecta a la terminal $\overline{\text{CS}}$ del 8255 para habilitarlo. Las señales generadas por el 8288 para el control de entrada y salida ($\overline{\text{IOR}}$ y $\overline{\text{IOW}}$) son utilizadas para controlar la entrada y salida del 8255. Las líneas del bus de datos interno de SCIP están conectadas a las terminales 27 a 34 (D7 - D0) del 8255, mientras que las terminales correspondientes al Puerto A del PPI estan conectadas a un transceiver 74LS245 (U4) el cual es usado para comunicarse con el bus del sistema anfitrión. La dirección de este transceiver es controlada con la línea $\overline{\text{DATAREAD}}$ generada por la lógica de decodificación presentada en la hoja 2 de diagramas, y que corresponde al puerto de lectura (puerto 300h de la PC), mientras que la habilitación de dicho transceiver se lleva a cabo con la línea $\overline{\text{BOARDSEL}}$ también generada internamente (vease Sección 3.4). Esto posibilita la lectura de datos de 8 bits desde la computadora anfitriona.

Para el caso de escritura de un byte a SCIP, se conectó al puerto B del

Figura 3.3: Controlador Programable de Periféricos 8255.

8255 (terminales 18 a 25) un 74LS373¹² cuyas entradas estan conectadas al bus de datos de la computadora anfitriona. La terminal de habilitación de este latch esta conectada a la línea $\overline{\text{WRITEDATA}}$ correspondiente al puerto 301h de la PC (ver Subsección 3.4.2) y la terminal de habilitación de salidas esta a tierra. Esto posibilita la escritura de un byte a SCIP por parte de la PC anfitriona.

Se usa la señal de $\overline{\text{MASTERRESET}}$ generada por el 8284 para reinicializar al 8255. Cuando el 8255 recibe dicha señal (es decir, cuando la terminal 35 es puesta en ALTO), todos los puertos son colocados en el modo de entrada (todas las líneas estaran en estado de alta impedancia) [Int1990]. Este dispositivo puede permanecer en tal estado si no se programa.

El 8255, para los propósitos de este trabajo, se programó en Modo 0. Este modo brinda operaciones de entrada y salida simples, para cada uno de los tres puertos del 8255. No existen señales de “*handshaking*”, sino que los datos son simplemente leídos o escritos mediante puertos. Este modo consiste en lo siguiente:

¹²Latch transparente de 8 bits con salidas de 3 estados.

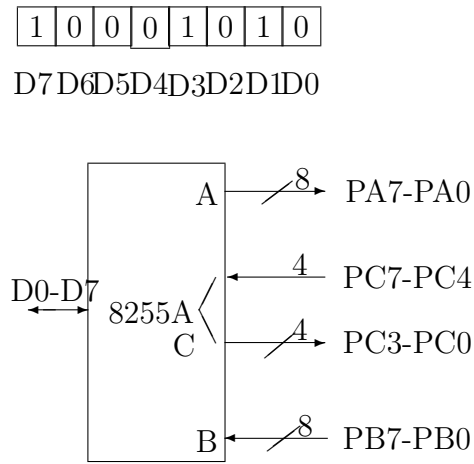


Figura 3.4: Controlador Programable de Periféricos 8255 en configuración de Modo 0, con la palabra de palabra de control 6, programando el Puerto A como salida y el Puerto B como entrada.

- Dos puertos de 8 bits y dos de 4 bits.
- Cualquier puerto puede ser entrada o salida.
- Las entradas no tienen latches.
- Las salidas tienen latches.
- Existen 16 posibles combinaciones de entrada/salida en este modo.

En la Tabla 3.5 se muestra la operación básica del 8255 [Int1990].

Cuando se escribe un dato al 8255 a través del bus de datos, y $A1 = A0 = 1$, entonces se transfiere una palabra de control al 8255, programándolo (ver Tabla 3.5).

En las pruebas realizadas se programó al 8255 utilizando la palabra de control 6 (ver Apéndice B), de tal forma que el Puerto A trabaja como salida y el Puerto B como entrada, además de que el Puerto C se divide en 2 partes: PC7-PC4 funcionando como entrada y PC3-PC0 como salida. El código que realiza esta programación es el siguiente:

A1	A0	RD	WR	CS	Op. de Entrada (Lectura)
0	0	0	1	0	Puerto A \Rightarrow Bus de Datos
0	1	0	1	0	Puerto B \Rightarrow Bus de Datos
1	0	0	1	0	Puerto C \Rightarrow Bus de Datos
					Op. de Salida (Escritura)
0	0	1	0	0	Bus de Datos \Rightarrow Puerto A
0	1	1	0	0	Bus de Datos \Rightarrow Puerto B
1	0	1	0	0	Bus de Datos \Rightarrow Puerto C
1	1	1	0	0	Bus de Datos \Rightarrow Control
					Deshabilitación
X	X	X	X	1	Bus de Datos \Rightarrow 3er. Estado
1	1	0	1	0	Condición Ilegal
X	X	1	1	0	Bus de Datos \Rightarrow 3er. Estado

Tabla 3.5: Modo Básico de Operación del 8255.

```

MOV DX,3C7h      ; Direccion del pto. de control del 8255
                  ; (A0=A1=1)
MOV AL,8Ah       ; Modo 0, Pto. A = Salida, Pto. B = Entrada,
                  ; PC03 = Salida, PC47 = Entrada
OUT DX,AL        ; Programar el 8255

```

De esta forma el 8255 sirve para comunicar a la tarjeta con la computadora, viendo a esta como a un periférico de SCIP.

3.5 Protocolo de comunicación entre la tarjeta SCIP y la computadora anfitriona

Dado que la tarjeta SCIP es una tarjeta coprocesadora de propósito general, a cuyo diseño básico se le pueden agregar otros componenets, el protocolo de comandos de comunicación entre la tarjeta y la PC puede variar dependiendo de la función de SCIP. Un protocolo básico propuesto para llevar a cabo la comunicación se presenta a continuación.

Dado que el Puerto B del PPI 8255 se programa en Modo 0 (ver Subsección 3.4.3), este es usado por la computadora anfitriona para escribir comandos a la tarjeta, mientras que el Puerto A es utilizado para transferencia

Dec.	Binario	Propósito
1	00000001	Iniciar programa de usuario ubicado en la EPROM de SCIP.
2	00000010	Iniciar transferencia de datos de la PC a SCIP (los siguientes 2 bytes en el Puerto B indican cuantos bytes serán transferidos).
3	00000011	En espera de transferencia de datos de SCIP a la PC anfitriona.
4	00000100	PC anfitriona lista para transferencia de datos de la tarjeta a la PC (los primeros 2 bytes en el Puerto A le indican a la PC cuantos bytes se transferiran).

Tabla 3.6: Comandos básicos para el SCIP.

de comandos y datos hacia aquella. Los comandos básicos de comunicación para SCIP se presentan en la Tabla 3.6.

Este conjunto de comandos básicos se puede ampliar dependiendo de la aplicación y de los programas que la tarjeta ejecute.

A su vez, la tarjeta SCIP puede indicar su estado ya sea a través del Puerto A o de PC0-PC2, o bien mediante el uso de las señales de *handshaking* del 8255, en caso de que se programe en Modo 1 o Modo 2.

3.6 Consideraciones importantes sobre la programación.

Para la realización de los programas que se colocan en SCIP es preciso señalar los siguientes aspectos, poniendo como ejemplo el siguiente fragmento que inicializa un programa en SCIP:

```

1      ORG 0FFFF0h
2
3      START:
4      MOV AX,0F000h
5      MOV CS,AX
6      JMP PROG
```

- Localización del programa en memoria: El programa inicial de SCIP siempre debe comenzar en la dirección FFFF0h absoluta, es decir, en el segmento y desplazamiento F000:FFF0 hex., ya que el procesador salta a esa dirección al inicializarse. Esto se realiza con la pseudoinstrucción **ORG** de ensamblador, como se ve en la línea 1 del código arriba mostrado.
- Inicialización del segmento de código: En las líneas 4 y 5 del código arriba presentado se puede ver la manera en como inicializar el segmento de código del programa de inicio de SCIP. Se tiene que poner un F000h en el registro CS ya que de otra forma el programa se “pierde” al realizar el salto al cuerpo del programa de inicialización (línea 6 del listado).

Así mismo el cuerpo del programa de inicialización debe ir de preferencia en la dirección 100h del segmento F000, realizando esto como se muestra en la línea 4 del siguiente listado parcial:

```
1  CODIGO SEGMENT 'Code'
2      ASSUME CS:CODIGO, DS:CODIGO, ES:NOTHING, SS:NOTHING
3
4      ORG 0F0100h
5
6  PROG PROC NEAR
7      ; el cuerpo principal se coloca aqui...
```

Esto es porque generalmente el ensamblador rechaza direcciones de inicio de un programa ubicadas en un desplazamiento menor que 100h.

Capítulo 4.

Pruebas del Sistema y sus Resultados.

4.1 Introducción.

En éste capítulo se presentan las pruebas efectuadas al sistema, y sus resultados. El software de prueba desarrollado es básico y solo fue creado con fines de demostración del hardware.

4.2 Pruebas del sistema.

Se realizaron las siguientes pruebas en el sistema, para demostrar la capacidad de ejecutar otro programa al mismo tiempo que la PC anfitriona, y comunicar a SCIP con dicha computadora:

1. Comunicación unidireccional de SCIP a la PC anfitriona.
2. Comunicación bidireccional sincronizada por software.

Para estos programas se utilizo Turbo Assembler y Turbo C, ambos de Borland Int.

Comunicación unidireccional de SCIP a la PC anfitriona.

Esta prueba consistió en que la tarjeta SCIP enviara datos a la computadora anfitriona mediante el 8255 programado en Modo 0 (Ver Subsección 3.4.3), y que un programa en aquella computadora los recibiera y desplegara en pantalla.

El siguiente código en ensamblador del 8088 fue programado y grabado en la memoria EPROM correspondiente al bloque 6 de memoria de SCIP (Ver Tabla 3.1):

```
1  .MODEL SMALL
2
3  CODIGO SEGMENT 'Code'
4      ASSUME CS:CODIGO, DS:CODIGO, ES:NOTHING, SS:NOTHING
5
6      ORG 0F0100h
7
8  PROG PROC NEAR
9      XOR AX,AX
10     MOV DX,3C7h    ; Direccion del pto. de control del 8255
11     MOV AL,8Ah     ; Modo 0, A=Salida, B=Entrada,
12                     ; PC03 = Salida, PC47 = Entrada
13     OUT DX,AL      ; Programar 8255
14
15     MOV DX,3C4h    ; Data Bus -> Port A
16     MOV AL,55h     ; Valor a enviar a la PC anfitriona
17
18     WRITE:         ; Ciclo infinito de escritura
19     OUT DX,AL
20     JMP WRITE
21
22     ORG 0FFFF0h    ; Inicio del programa F000:FFF0
23
24     START:         ; Inicializacion del CS
25     MOV AX,0F000h
26     MOV CS,AX
27     JMP PROG       ; Salto a la rutina principal
28 PROG ENDP
29 CODIGO ENDS
30     END PROG
```

En este listado se aprecia el bloque de inicialización descrito en la Sección 3.6, en las líneas 23 a 28.

En las líneas 8 a 21 del listado se muestra en código principal del programa, el cual realiza lo siguiente:

1. Programar el 8255 en Modo 0 (como se describe Subsección 3.4.3), en las líneas 9 a 14.

2. Enviar en un ciclo infinito el valor 55h al puerto 3C4h (puerto de salida de SCIP, ver Tabla 3.5), en las líneas 16 a 21.

En programa que se ejecutó en la computadora anfitriona fue realizado en lenguaje C, y su listado se presenta a continuación:

```
1  #include <stdio.h>
2  #include <dos.h>
3
4  int data;
5
6  void main(void)
7  {
8      do {
9          data = inportb(0x300); /* Leer byte del pto. */
10         printf("%c",data);      /* Escribir byte a pantalla */
11     } while (1);
12 }
```

Lo que realiza este programa es un ciclo infinito (líneas 8 a 11) en donde se lee un byte del puerto 300h (puerto de lectura de SCIP por parte de la PC anfitriona) e imprimirlo a pantalla (líneas 9 y 10). Entonces en pantalla se desplegará el valor enviado ('U' = 55h).

Comunicación bidireccional sincronizada por software.

Esta prueba consistió en que los caracteres tecleados por el usuario en la computadora anfitriona fueran enviados a la tarjeta SCIP, la cual los reenviara a aquella computadora, y fueran desplegados en pantalla, todo esto sincronizando las transferencias con caracteres de control, como se explica a continuación.

```
1  #include <stdio.h>
2  #include <dos.h>
3  #include <conio.h>
4
5  int data;
6  char c = '\0';
7
```

```
8  void main(void)
9  {
10     clrscr();
11
12     c = getch();           /* Leer caracter */
13     while (c != 'T') {     /* Termina cuando se lee T */
14         outportb(0x301,'T'); /* Enviar caracter de inicio */
15
16         do {
17             data = inportb(0x300); /* Esperar por caracter de reconoci- */
18         } while (data != 'R'); /* miento enviado por SCIP (R) */
19
20
21         outportb(0x301,c);     /* Enviar a SCIP caracter leído */
22         delay(1);
23         data = inportb(0x300); /* Leer caracter enviado por SCIP */
24
25         printf("%c",data);     /* Imprimir a pantalla el caracter */
26         c = getch();           /* Leer nuevo caracter de teclado */
27     }
28 }
```

El programa cuyo listado se presenta arriba esta realizado en lenguaje C y se ejecuta en la PC anfitriona. Lo que realiza es lo siguiente:

1. Leer un carácter¹ del teclado (línea 12).
2. Si el carácter leído es una T, entonces termina el programa (línea 13).
3. Enviar a SCIP por el puerto 301h el carácter T que indica “Inicio de Transmisión” (línea 14).
4. Esperar en un ciclo a que SCIP envíe en el puerto 300h el carácter R que indica “Transmisión Reconocida” (líneas 16 a 18).
5. Enviar a SCIP el carácter leído por el teclado (línea 21).
6. Leer de SCIP el carácter que la tarjeta envía, que es el mismo que el programa en la computadora anfitriona le envió (línea 23).

¹Un carácter corresponde a un byte (8 bits).

7. Imprimir en pantalla el carácter leído (línea 25).
8. Leer nuevo carácter del teclado y repetir el ciclo desde el paso 2 (línea 26).

Como se dijo en la Subsección 3.4.2 los puertos utilizados son el 300h y 301h (entrada y salida, respectivamente). Los caracteres usados para enviar mensajes son en este ejemplo T y R, pero pudieran haberse utilizado otros cualesquiera.

El programa en ensamblador ejecutado por SCIP se presenta a continuación:

```
1  .MODEL SMALL
2
3  CODIGO SEGMENT 'Code'
4      ASSUME CS:CODIGO, DS:CODIGO, ES:NOTHING, SS:NOTHING
5
6      ORG 0F0100h
7
8  PROG PROC NEAR
9      XOR AX,AX
10     XOR BX,BX
11
12     MOV DX,3C7h ; Direccion del pto. de control del 8255 (A0=A1=1)
13     MOV AL,8Ah  ; Modo 0, Pto. A = Salida, Pto. B = Entrada,
14                     ; PC03 = Salida, PC47 = Entrada
15     OUT DX,AL   ; Programar 8255
16
17     CICLO:
18         MOV DX,3C5h ; Port B -> Data Bus
19
20     LEE1:        ; Esperar comando de inicio de transmision (T)
21         IN  AL,DX
22         CMP AL,'T'
23         JNE LEE1
24
25         MOV AL,'R' ; Enviar al anfitrión un comando de reconocimiento (R)
26         MOV DX,3C4h ; Data Bus -> Port A
```

```
27      OUT DX,AL
28
29      MOV DX,3C5h   ; Port B -> Data Bus
30
31      LEE2:          ; Esperar primer caracter del anfitrión
32      IN  AL,DX
33      CMP AL,'T'
34      JE  LEE2
35
36      MOV DX,3C4h   ; Data Bus -> Port A
37      OUT DX,AL     ; Enviar caracter recibido de regreso a anfitrión
38
39      JMP CICLO     ; Reiniciar el ciclo
40
41      ORG 0FFFF0h
42
43      START:         ; Inicializar CS y saltar al inicio del programa
44      MOV AX,0F000h
45      MOV CS,AX
46      JMP PROG
47      PROG ENDP
48      CODIGO ENDS
49      END  PROG
```

Este programa realiza lo siguiente, en un ciclo infinito:

1. En las líneas 12 a 15 se programa al 8255 para trabajar en Modo 0.
2. Esperar en un ciclo leyendo al puerto 3C5h (Puerto B a Bus de Datos) a que se reciba el carácter de Inicio de Transmisión (T) (líneas 18 a 23).
3. Enviar un carácter R para indicar a la computadora anfitriona que se recibió el carácter T (líneas 25 a 27).
4. Esperar en un ciclo a que llegue el carácter de la computadora anfitriona (líneas 29 a 34).
5. Enviar a la computadora anfitriona el carácter recibido (líneas 36 y 37).

En ambos casos el programa en ejecución en SCIP puede reiniciarse aplicando la señal de RESET mediante el botón ubicado en la tarjeta.

4.3 Resultados de las pruebas.

Los programas de prueba desarrollados y presentados en el documento permiten que la computadora y SCIP se comuniquen sincronizadamente, y que un programa se este ejecutando en SCIP al mismo tiempo que otro se ejecuta en la computadora anfitriona. Los resultados fueron considerados satisfactorios.

Conclusiones y Perspectivas.

Conclusiones.

El “Sistema de Coprocesamiento de Información para PC”, resultado de este trabajo, permite efectuar la comunicación entre dos procesadores, y la ejecución de un programa en la tarjeta de expansión, enviando los resultados al procesador de la computadora. El uso del Controlador Programable de Periféricos 8255 permite que la tarjeta de expansión vea a la PC como a un periférico.

Se desarrolló software básico para demostrar la comunicación bidireccional entre la tarjeta SCIP y la computadora anfitriona, y al mismo tiempo la ejecución de programas distintos en ambos procesadores.

Perspectivas

El diseño presentado puede considerarse básico, y podrá ser personalizado por un usuario, con el uso de algún tipo de interfaz (serial, paralela u otra), o el uso de lógica adicional, para agregarle elementos externos, y expandir sus aplicaciones a áreas como adquisición de datos, o procesamiento de sonido o gráficos.

Dadas las limitaciones de los recursos disponibles para el desarrollo del sistema, la circuitería utilizada en el mismo podría considerarse obsoleta, habiendo ya procesadores mas modernos o especializados en ciertas tareas. Como mejoras al sistema se puede contemplar la substitución de los circuitos integrados utilizados (el procesador y toda su lógica de soporte) por otros más modernos, así como el uso de memoria de otro tipo, que permita expandir la capacidad de almacenamiento del sistema. Sin embargo, el 8088 era la única opción disponible que permitía simplificar la lógica del sistema, ya que si se hiciera uso de otro tipo de procesadores (de la familia Intel) se requeriría el uso de otros dispositivos y técnicas.

Así mismo, el software para una aplicación específica del sistema debe ser desarrollado. En el sistema actual, este software propiamente no existe, y solo se contemplaron pequeños programas de prueba para efectuar la comunicación bidireccional entre la computadora anfitriona y SCIP, y un software básico grabado en una memoria EPROM de la tarjeta, para programar los dispositivos que así lo requieren, y efectuar la comunicación mencionada.

BIBLIOGRAFÍA

- [DoT] Doulmas, Thomas A., An Jiu, Yorkey, Thomas J., *Interfacing to the IBM PC Bus*.
- [HaD1986] Hall, Douglas V., *Microprocessors and Interfacing. Programming and Hardware*, McGraw-Hill Book Co, 1986.
- [GoJ1991] Godfrey, J. Terry, *Lenguaje Ensamblador para Microcomputadoras IBM*, Primera Edición, Prentice Hall Hispanoamericana, 1991.
- [Int1989] Intel Incorporated, *Microprocessors Handbook*, Intel Inc., 1989.
- [Int1990] Intel Incorporated, *Peripherals Handbook*, Intel Inc., 1990.
- [NEC1990] National Electronic Co., *Memories*, NEC, S.F.
- [Mot1992] Motorola Incorporated, *FAST and LS TTL Data*, Segunda Edición, Motorola Inc., 1992.
- [QuH1992] Quesada Pérez, Hugo M., *SADYR, Sistema de Adquisición de Datos y Reproducción de Muestras. Tesis Profesional para obtener el Título de Licenciado en Computación.*, F.C.F.M., 1992.
- [RiR1994] Rivera R., Rafael, *Computadoras Personales*, B.U.A.P., 1994.
- [QuJ1992] Quain, John R., “Nuevas Rutas del Video”, *PC Magazine en Español*, Volúmen 4 Número 2, Febrero 1993, p. 59.
- [BaJ1996] Bayco, John, bayko@cs.uregina.ca, *Great Microprocessors of the Past and Present*, <http://www.cs.uregina.ca/%7Ebayco/cpu.html>, Noviembre 1996.
- [Str1996] Stratcom, *Bus Architectures*, <http://www.stratcom.com/bus.html>, 1996.
- [CEK] CEKIT, *Curso Práctico de circuitos digitales y microprocesadores*.

[JDR1988] JDR Microdevices, *User's Manual - 16 bit prototype board*, JDR Microdevices, 1988.

Apéndice A.

Manual del Usuario.

A.1 Introducción.

En este manual del usuario de SCIP se presentan los procedimientos para:

1. Manejo e instalación del hardware de SCIP.
2. Realización de software para SCIP.

A.2 Manejo e instalación del hardware de SCIP.

En esta sección se detallan los procedimientos para la instalación y manejo de la tarjeta SCIP.

A.2.1 Introducción al hardware de SCIP.

SCIP es un sistema de hardware que consiste de una tarjeta de expansión para bus ISA AT que puede ser utilizada en computadoras basadas en el Intel 286 o superior. El prototipo de esta tarjeta está ensamblado en una placa para wire-wrap de 16 bits JDR-PR10 de JDR Microdevices.

En la Figura A.1 se puede ver una vista del lado de componentes de SCIP, mientras que en la Figura A.2 se aprecia el lado de alambramiento del sistema.

En la Figura A.3 se muestra un esquema de las partes constituyentes de SCIP. La descripción de estas partes es la siguiente¹:

- 1 - Sección preconstruida de decodificación.** De esta sección solo se utilizaron los transceivers para el bus de direcciones.

¹Los números se refieren a los que identifican cada bloque en la figura.

Figura A.1: Lado de componentes de SCIP.

Figura A.2: Lado de alambramiento del prototipo de SCIP.

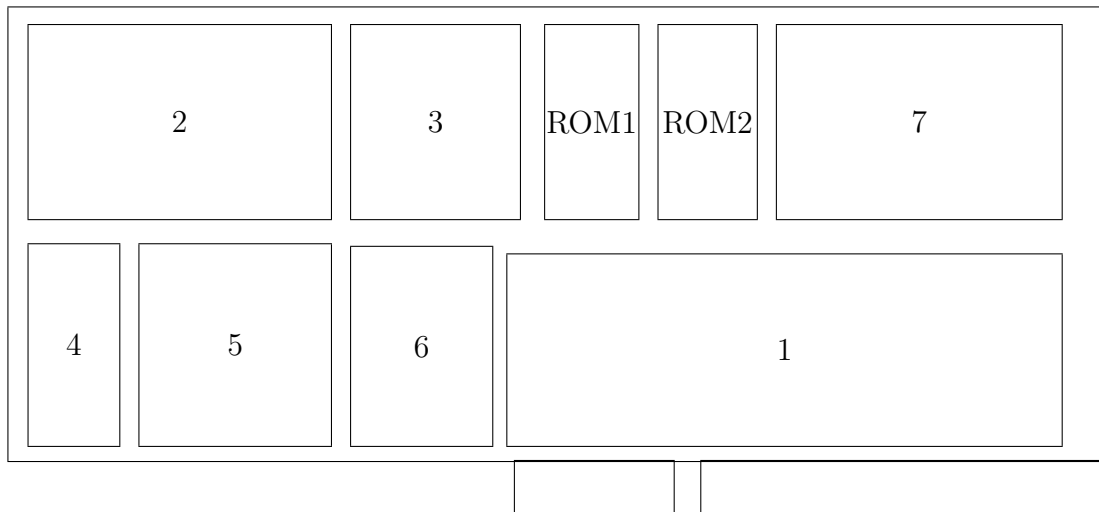


Figura A.3: Diagrama en bloques de SCIP: (1) Sección preconstruida de decodificación, (2) Módulo de Procesamiento y Control de BUS, (3) Módulo de decodificación interna y buffers, (4) Módulo de temporización, (5) Módulo de comunicación, (6) Módulo de lógica de selección, (7) Módulo de memorias RAM, (ROM1) Memoria EPROM 1, (ROM2) Memoria EPROM 2.

- 2 - Módulo de Procesamiento y Control de BUS.** Esta sección contiene al procesador 8088 y el coprocesador 8087, así como el controlador del bus 8288, y se encarga del procesamiento.
- 3 - Módulo de decodificación interna y buffers.** Esta sección contiene la lógica de decodificación de puertos y memoria interna así como buffers y transceivers para las memorias.
- 4 - Módulo de temporización.** Esta sección contiene la lógica de temporización y reinicialización de la tarjeta.
- 5 - Módulo de comunicación.** Esta sección contiene al Controlador Programable de Periféricos 8255 así como buffers y transceivers para la comunicación entre el bus de expansión de la computadora anfitriona y el bus de SCIP.
- 6 - Módulo de lógica de selección.** Esta sección se encarga de decodificar las direcciones y señales del bus de la computadora anfitriona para

realizar la selección de la tarjeta SCIP y los diversos puertos que la conforman.

7 - Módulo de memorias RAM. Esta sección contiene a las cuatro memorias RAM estáticas de 32Kb.

ROM1 - Memoria EPROM 1. En esta memoria esta la dirección de arranque del procesador 8088 (F000:FFF0 hex), por lo que en esta memoria se debe colocar el programa de inicio de la tarjeta, en la dirección adecuada, como se describe en la Sección A.3.

ROM2 - Memoria EPROM 2. Esta es la segunda memoria EPROM, y puede contener el programa de usuario, en caso de que la ROM1 solo contenga la rutina de inicialización.

A.2.2 Precauciones.

Dado que se utilizan circuitos MOS y CMOS en la tarjeta SCIP, los cuales son altamente sensibles a descargas de electricidad estática, se recomienda manejar la tarjeta con precaución.

- La computadora debe estar apagada al conectar o desconectar la tarjeta de expansión.
- No retire la tarjeta de su empaque antiestático hasta que se vaya a utilizar.
- Procure evitar tocar las terminales de las bases e wire-wrap de la tarjeta prototipo, así como los componentes.
- Descarguese de electricidad estática tocando una parte no pintada del chasis de la computadora donde vaya a instalar la tarjeta, o cualquier otra pieza metálica que este aterrizada.

A.2.3 Instalación.

Para instalar la tarjeta SCIP en la computadora anfitriona, siga estos pasos:

1. Desconecte la computadora de la corriente.
2. Abra el gabinete de la computadora.

3. Busque una ranura de expansión ISA libre.
4. Inserte firmemente la tarjeta SCIP en dicha ranura.

Una vez hecho esto podrá encender la computadora.

Figura A.4: Tarjeta de expansión SCIP montada sobre una tarjeta extensora del bus (no necesaria en la práctica) en un ensamble 386, durante las pruebas del sistema.

En la Figura A.4 se presenta una fotografía de la tarjeta de expansión montada en una computadora. En ese caso se utilizó una tarjeta extensora del bus, para que se pudiera apreciar mejor, pero en la práctica no es necesario.

A.2.4 Manejo del Hardware.

La tarjeta SCIP solo tiene un control operable directamente. Este es el botón de *RESET* ubicado en la parte inferior izquierda de la tarjeta. Este funciona con un “push button”, el cual al presionarse hace que el procesador de SCIP reinicie y se vuelva a comenzar el programa ubicado en la memoria EPROM.

A.3 Programación de SCIP.

SCIP tiene dos memorias EPROM de 64Kb y 4 memorias RAM de 32Kb. El programa de inicio de SCIP debe colocarse en la EPROM 1 (ver Subsección A.2.1), ya que en esta memoria se mapea la dirección de arranque del procesador 8088, la cual es F000:FFF0 hexadecimal.

En el Capítulo 3 se presenta la descripción completa de los bloques de memoria y puertos utilizados por el sistema.

Los programas de SCIP se pueden hacer en cualquier lenguaje que genere código para el 8088, pero se recomienda el uso de ensamblador, ya que permite la generación de código compacto y es más versátil para el manejo de los registros de segmento y del hardware en general.

A.3.1 Localización del programa en memoria.

Para la realización de los programas que se colocan en SCIP es preciso señalar los siguientes aspectos, poniendo como ejemplo el siguiente fragmento que inicializa un programa en SCIP:

```
1      ORG 0FFFF0h
2
3      START:
4      MOV AX,0F000h
5      MOV CS,AX
6      JMP PROG
```

- Localización del programa en memoria: El programa inicial de SCIP siempre debe comenzar en la dirección FFFF0h absoluta, es decir, en el segmento F000h y el desplazamiento FFF0h, ya que el procesador salta a esa dirección al inicializarse. Esto se realiza con la pseudoinstrucción **ORG** de ensamblador, como se ve en la línea 1 del código arriba mostrado.
- Inicialización del segmento de código: En las líneas 4 y 5 del código arriba presentado se puede ver la manera en como inicializar el segmento de código del programa de inicio de SCIP. Se tiene que poner un F000h en el registro CS ya que de otra forma el programa se “pierde”²

²Se presentó éste problema durante las pruebas del sistema, pero se resolvió con lo que se explica aquí.

al realizar el salto al cuerpo del programa de inicialización (línea 6 del listado).

Así mismo el cuerpo del programa de inicialización debe ir de preferencia en la dirección 100h del segmento F000h, realizando esto como se muestra en la línea 4 del siguiente listado parcial:

```
1  CODIGO SEGMENT 'Code'
2      ASSUME CS:CODIGO, DS:CODIGO, ES:NOTHING, SS:NOTHING
3
4      ORG 0F0100h
5
6  PROG PROC NEAR
7      ; el cuerpo principal se coloca aqui...
```

Esto es porque generalmente el ensamblador rechaza direcciones de inicio de un programa ubicadas en un desplazamiento menor que 100h, dado que los programas para producir código binario lo generan para máquinas IBM compatibles.

Un programa en ensamblador usado en SCIP tiene la siguiente estructura:

```
CODIGO SEGMENT 'Code'
    ASSUME CS:CODIGO, DS:CODIGO, ES:NOTHING, SS:NOTHING

    ORG 0F0100h

PROG PROC NEAR

    ; el cuerpo principal se coloca aqui...

    ORG 0FFFF0h

START:
; Inicializar CS y saltar al inicio del programa
    MOV AX,0F000h
    MOV CS,AX
    JMP PROG
PROG ENDP
CODIGO ENDS
    END PROG
```

Una vez que finalice el programa principal, se podría poner un salto hacia el inicio del programa nuevamente, o un ciclo infinito que hiciera que la ejecución no saltará a direcciones erróneas. Esto dependería de cada aplicación.

Lo que conforme el programa puede ser cualquier instrucción del 8088, siempre y cuando:

1. No se usen interrupciones de MS-DOS o del BIOS de una PC.
2. No se usen instrucciones de entrada o salida (a pantalla, por ejemplo).

A.3.2 Programación de las memorias EPROM.

Las memorias EPROM utilizadas en el sistema son las Texas Instruments 27C512. Estas memorias pueden ser borradas con luz ultravioleta. El tiempo de borrado generalmente no excede los 5 minutos, aunque en caso de ser memorias viejas, pueden requerirse algunos minutos más.

Estas memorias se pueden programar electrónicamente mediante un programador de memorias. Para las pruebas del sistema se utilizó un programador AllMax, pero en cada caso se debe consultar la documentación del programador utilizado para determinar el procedimiento de programación. Generalmente el formato aceptado por el programador será Intel Hexadecimal o "Binary".

Una vez programadas las memorias, éstas deben ser colocadas en las bases correspondientes a su número: la EPROM con la rutina inicial en la primera base (ROM1, ver Figura A.3) y la segunda memoria en la base ROM2. La muesca en las bases de wire-wrap del prototipo indican la posición de la terminal 1 de los circuitos integrados. Además como guía para la colocación se puede ver la posición de las terminales de tierra y voltaje en la parte inferior (alambrado) de la tarjeta prototipo.

Apéndice B.

Especificaciones de los principales circuitos usados.

B.1 Introducción.

En este apéndice se presentan las especificaciones de los principales circuitos integrados utilizados en el proyecto de tesis. Al final del apéndice se presentan las secciones principales de las fichas técnicas¹ de estos circuitos, tomadas de [Int1989] y [Int1990]. Para mayor información sobre estos circuitos, consúltense los manuales citados.

B.2 Intel 8088.

B.2.1 Introducción.

En esta sección se presenta una breve descripción de las especificaciones del procesador Intel 8088.

B.2.2 Microprocesador de 8 bits HMOS (8088/8088-2).

Este microprocesador ofrece las siguientes características:

- Interfaz de bus de 8 bits.
- Arquitectura interna de 16 bits.
- Capacidad de direccionamiento directo de 1 megabyte de memoria.

¹Data Sheets.

- Compatibilidad de software directa con el CPU 8086.
- Conjunto de registros de 14 palabras de 16 bits con operaciones simétricas.
- Modos de direccionamiento de 24 operandos.
- Operaciones de tamaño byte, word y bloque.
- Aritmética con y sin signo de 8 y 16 bits, en binario o decimal, incluyendo multiplicación y división.
- Dos velocidades de reloj: 5 Mhz para el 8088 y 8 Mhz para el 8088-2
- Disponible en EXPRESS
 - Rango de temperatura estandar.
 - Rango de temperatura extendido.

El procesador Intel 8088 es un microprocesador de alto desempeño desarrollado en N-canal, reducción de carga, tecnología de compuerta de silicón (HMOS), y empacado en un Cerdip de 40 terminales. El procesador tiene atributos tanto de microprocesadores de 8 como de 16 bits. Es directamente compatible con el software del 8086 y el hardware y periféricos del 8080/8085.

La Tabla B.1 presenta los valores máximos y mínimos de temperatura y voltaje.

Temperatura Ambiente	0C a + 70C
Temperatura del Empaque (Plastico)	0C a +95C
Temperatura del Empaque (CERDIP)	0C a +70C
Temperatura de almacenamiento	-65C a +150C
Voltaje en cualquier terminal respecto a GND	-1.0 a +7V
Disipación de poder	2.5 Watts

Tabla B.1: Rangos Máximos Absolutos.

B.3 Intel 8087.

B.3.1 Introducción.

En esta sección se da una breve descripción de las especificaciones del coprocesador matemático Intel 8087.

B.3.2 Coprocesador de Datos Numéricos 8087/8087-2/8087-1.

Este coprocesador presenta las siguientes características:

1. Coprocesador de datos numéricos de alto desempeño.
2. Añade instrucciones de aritmética, trigonometría, exponenciación y logaritmos al conjunto de instrucciones estándar del 8088/8086 para todos los tipos de datos.
3. El CPU junto con el 8087 soportan 7 tipos de datos: enteros de 16, 32 y 64 bits, flotantes de 32, 64 y 80 bits y operandos de 18 dígitos BCD.
4. Compatible con el estándar de punto flotante IEEE 754.
5. Disponibilidad en 5 Mhz (8087), 8 Mhz (8087-2) y 10 Mhz (8087-1).
6. Añade una pila de registros individualmente direccionables de 8x80 bits a la arquitectura del 8086/8088 y 80186/80188.
7. Proporciona siete funciones de manejo de excepciones.
8. Sistema de interfaz compatible con MULTIBUS.

El coprocesador de datos numéricos 8087 proporciona las instrucciones y los tipos de datos necesarios para aplicaciones numéricas de alto rendimiento, proporcionando un mejor desempeño en 100 veces que la CPU sola. El 8087 está desarrollado en N-canal, con reducción de carga, tecnología de compuerta de silicón (HMOS III), y está contenido en un empaque de 40 terminales. Son añadidas 68 instrucciones de procesamiento numérico al conjunto de instrucciones del 80886/8088/80186/80188, así como ocho registros de 80 bits son añadidos al conjunto de registros. El 8087 es compatible con el Estandar de Punto Flotante IEEE 754.

Temperatura ambiente	0C a 70C
Temperatura de almacenamiento	-65C a +150C
Voltaje en cualquier pin respecto a GND	-1.0V a +7V
Disipación de poder	3.0 Watts

Tabla B.2: Rangos Máximos Absolutos.

La Tabla B.2 presenta los valores máximos y mínimos de temperatura y voltaje.

B.4 Intel 8288.

B.4.1 Introducción.

En esta sección se da una breve descripción de las especificaciones del controlador de bus 8288.

B.4.2 Controlador de Bus CHMOS 82C88.

Este controlador proporciona las siguientes características:

1. Compatibilidad de terminales con el 8288 bipolar.
2. Proporciona soporte para el 8088/86, 80C88/86.
3. Operación a bajo poder:
 - $I_{ccs} = 100\mu A$
 - $I_{cc} = 10\text{ mA}$
4. Proporciona comandos avanzados para buses multi-master.
5. Manejadores de salida de comandos de 3 estados.
6. Alta capacidad de transmisión.
7. Configurable para ser usado con un bus de entrada/salida.
8. Requiere una sola fuente de suministro de 5V.

9. Operación a 8 Mhz (8288-2).

El Intel 82C88-2 es una versión de alto rendimiento del controlador de bus bipolar 8288. El 8288-2 proporciona generación de de comando y control de temporización, para los sistemas con arquitectura 8086. El diseño de circuitos estáticos CHMOS asegura una operación de bajo poder. La capacidad de manejo del 8288 hace innecesarios a otros manejadores.

La Tabla B.3 presenta el significado de las líneas de estado del procesador con respecto a las funciones del 8288.

$\overline{S2}, \overline{S1}, \overline{S0}$	Estado del procesador	Comando del 82C88-2
0 0 0	Interrupt Acknowledge	\overline{INTA}
0 0 1	Read I/O Port	\overline{IORC}
0 1 0	Write I/O Port	$\overline{IOWC}, \overline{AIOW}$
0 1 1	Halt	Ninguna
1 0 0	Code Access	\overline{MRDC}
1 0 1	Read Memori	\overline{MRDC}
1 1 0	Write Memory	$\overline{MWTC}, \overline{AMWC}$
1 1 1	Pasivo	Ninguna

Tabla B.3: Comandos del 8288.

La Tabla B.4 presenta los valores máximos y mínimos de temperatura y voltaje.

Temperatura	0C a 70C
Temperatura de almacenamiento	-55C a +150C
Voltaje suministrado con respecto a GND	-0.5V a +7.0V
Todos los voltajes de entrada con respecto a GND	-0.5V a $V_{cc} + 0.5V$
Todos los voltajes de salida con respetco a GND	-0.5 a $V_{cc} + 0.5V$
Disipación de poder	0.7W

Tabla B.4: Rangos Máximos Absolutos.

B.5 Intel 8259.

B.5.1 Introducción.

En esta sección se da una breve descripción de las especificaciones del controlador de bus 8259.

B.5.2 Controlador de Periféricos Programable 8259A.

Este controlador proporciona las siguientes características:

1. Compatible con el 8086 y 8088.
2. Compatible con el MCS-80 y MCS-85.
3. Controlador con 8 niveles de prioridad.
4. Expandible a 64 niveles.
5. Modos de interrupción programables.
6. Capacidad de petición enmascarada individual.
7. Terminal de alimentación única de +5V (no requiere reloj).
8. Disponible en DIP de 28 terminales.

El Controlador Programable de Interrupciones Intel 8259A maneja hasta 8 interrupciones con prioridad vectorizadas para la CPU. Se puede conectar en cascada hasta tener 64 niveles de prioridad vectorizada de interrupción sin circuitería adicional. Está empacado en un DIP de 28 terminales, usa tecnología NMOS y requiere un único suministro de alimentación de +5V. La circuitería es estática, y no requiere entrada de reloj.

El 8259A está diseñado para minimizar el software y la sobrecarga en tiempo real al manejar interrupciones con prioridad multinivel. Tiene varios modos, permitiendo la optimización para una gran variedad de requerimientos del sistema.

El 8259A es completamente compatible con el Intel 8259. El software escrito originalmente para el 8259 operará al 8259A en todos los modos equivalentes.

B.6 Intel 8284.

B.6.1 Introducción.

En esta sección se da una breve descripción de las especificaciones del generador de reloj 8284.

B.6.2 Generador de reloj y manejador, para CPU's 8086, 8284.

Este generador proporciona las siguientes características:

1. Genera el reloj del sistema para el 8086/8088.
2. Usa un cristal o una señal TTL o fuente de frecuencia.
3. Alimentación única de +5V.
4. Empaque de 18 terminales.
5. Genera la salida de *reset* de una entrada con Schmitt Trigger.
6. Proporciona la señal “Local Ready” y sincronización de la señal *ready* en MULTIBUS.
7. Capacidad para sincronizar el reloj con otros 8284.

El 8284 es un generador bipolar de reloj y manejador diseñado para proveer señales de reloj para el CPU 8086 y 8088 y sus periféricos. También contiene lógica de READY para la operación con dos sistemas MULTIBUS y proporciona al 8086/8088 la sincronización y temporización READY requerida. La lógica de *reset* (reinicialización) con histerésis y sincronización también se proporciona.

B.7 Intel 8255A.

B.7.1 Introducción.

En esta sección se da una breve descripción de las especificaciones del controlador programable de periféricos 8255.

B.7.2 Controlador programable de periféricos 8255.

Este controlador brinda las siguientes características:

1. Compatible con MCS-85.
2. 24 terminales programables de entrada y salida.
3. Completamente compatible con TTL.
4. Completamente compatible con los microprocesadores de la familia Intel.
5. Características de temporización mejoradas.
6. Capacidad de activación/desactivación de bits directa.
7. Reduce la cuenta del empaque del sistema.
8. Capacidad mejorada de manejo de CD.
9. Empaque de 40 terminales DIP.

El Intel 8255A es un dispositivo de entrada y salida programable de propósito general diseñado para su uso con microprocesadores Intel. Tiene 24 terminales de entrada/salida (E/S) las cuales pueden ser individualmente programadas en dos grupos de 12 y usadas en 3 modos de operación. En el primer modo (Modo 0) cada grupo de 12 terminales de E/S pueden ser programados en conjuntos de 4 para funcionar como entrada o salida. En Modo 1, el segundo modo, cada grupo puede ser programado para tener 8 líneas de entrada o salida; de las 4 terminales sobrantes, 3 terminales son usadas para *handshaking* y señales de control de interrupción. El tercer modo de operación (Modo 2) es un modo de bus bidireccional el cual usa 8 líneas para un bus bidireccional, y 5 líneas, tomadas prestadas de otros grupo, para *handshaking*.

Apéndice C.

Diagramas.

En este apéndice se presentan los diagramas por módulos del sistema, así como el diagrama general.

Los diagramas fueron creados con OrCAD y están organizados como se describe a continuación:

Hoja 1 - Módulo de Procesamiento.

Hoja 2 - Módulo de Decodificación de la Tarjeta.

Hoja 3 - Módulo de Comunicación.

Hoja 4 - Módulo de Generación de Líneas de Control/IRQ's.

Hoja 5 - Módulo de Decodificación Interna.

Hoja 6 - Módulo de Memoria.

El diagrama general se presenta al final del apéndice.

