

TUES
1504
P426
1993
E-2

UNIVERSIDAD DE EL SALVADOR

FACULTAD DE INGENIERIA Y ARQUITECTURA
ESCUELA DE INGENIERIA ELECTRICA

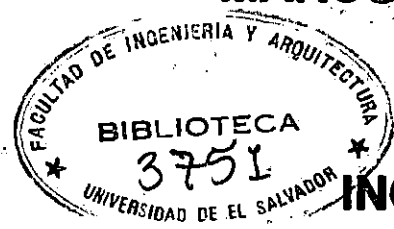


TRABAJO DE GRADUACION:

"DISEÑO Y CONSTRUCCION DE UN SISTEMA DIDACTICO MINIMO Z80 CONTROLADO POR LA AT LEMMON 286S"

PRESENTADO POR:

**JUAN CARLOS PERALTA ORELLANA
MARCOS TULIO PORTILLO GUEVARA**



PARA OPTAR AL TITULO DE:

15101355
15101355

INGENIERO ELECTRICISTA

JUNIO DE 1993

SAN SALVADOR, EL SALVADOR, CENTRO AMERICA

Recibido 29/10/93

UNIVERSIDAD DE EL SALVADOR

RECTOR : DR. FABIO CASTILLO FIGUEROA

SECRETARIO GENERAL : LIC. MIRNA ANTONIETA PERLA DE ANAYA

FACULTAD DE INGENIERIA Y ARQUITECTURA

DECANO: ING. JUAN JESUS SANCHEZ SALAZAR

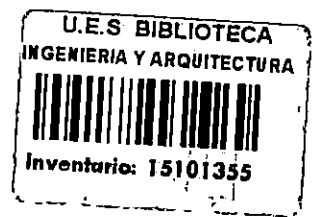
SECRETARIO : ING. JOSE RIGOBERTO MURILLO CAMPOS

ESCUELA DE INGENIERIA ELECTRICA

DIRECTOR: ING. RICARDO ERNESTO CORTEZ

SECRETARIO: ING. JORGE GALDAMEZ





UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERIA Y ARQUITECTURA
ESCUELA DE INGENIERIA ELECTRICA

Trabajo de Graduación previo al grado de:

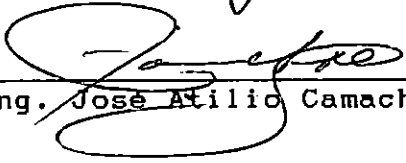
INGENIERO ELECTRICISTA

Presentado por:

Juan Carlos Peralta Orellana
Marcos Tulio Portillo Guevara

Trabajo de Graduación aprobado por:

Coordinador: 
Ing. Ricardo Ernesto Cortez

Asesor: 
Ing. José Atilio Camacho

ESCUELA DE INGENIERIA ELECTRICA
FACULTAD DE INGENIERIA
Y ARQUITECTURA
Universidad de El Salvador

TRABAJO DEDICADO A:

DIOS TODOPODEROSO: Por iluminar y guiar mi camino.

MIS PADRES: Por brindarme su incondicional apoyo moral y espiritual en todo momento.

MI HERMANA: Rosella Aída Portillo, por apoyarme en los momentos difíciles y ser excelente hermana.

MIS SERES QUERIDOS: Por que de una u otra forma me alentaron y apoyaron en el transcurso de coronar mi carrera.

A ELLOS DEDICO ESTE TRIUNFO

Marcos Tulio Portillo Guevara

ACTA DE CONSTANCIA DE NOTA Y DEFENSA FINAL

25 Mayo

3

En esta fecha, de _____ de 199_____,
en el local de Sala de Lectura de la Escuela de Ingeniería Eléctrica
a las 14:00 horas, con la presencia de las siguientes autoridades de la
Escuela de Ingeniería Eléctrica de la Universidad de El Salvador

- 1- Ing. Ricardo E. Cortez
Director de la E.I.E.
- 2- Ing. Jorge A. Galdámez
Secretario de la E.I.E.
- 3- _____

[Handwritten signature]
[Handwritten signature]



Y con el Honorable Jurado de evaluación integrado por las personas
siguientes:

- 1- Ing. José Heriberto Suria *[Signature]*
- 2- Ing. José Atilio Camacho *[Signature]*
- 3- Ing. Héctor Pompilio Escobar *[Signature]*
- 4- _____
- 5- _____

Se efectuó la defensa final reglamentaria del Trabajo de
Graduación: "DISEÑO Y CONSTRUCCION DE UN SISTEMA DIDACTICO MINIMO Z80 CONTROLADO POR LA
AT LEMMON 286S"

a cargo del (los) Br(es): Marcos Tulio Portillo Guevara
Juan Carlos Peralta Orellana

Habiendo obtenido el presente trabajo una nota final, global de 8.3

Ocho punto tres

PREFACIO

El objetivo fundamental de este trabajo es el diseñar y construir un dispositivo que esté basado en microprocesador y que pueda proporcionar un mecanismo para la enseñanza de sistemas autosostenidos en el area de instrumentación electrónica en la Escuela de Ingeniería Eléctrica de la Universidad de El Salvador.

Para ello se plantearon los siguientes objetivos:

1. Proveer a la Escuela de Ingeniería Eléctrica de la Universidad de El Salvador con una herramienta de hardware elemental para la enseñanza de sistemas programables a nivel básico.
2. Promover el diseño de sistema autosostenidos con microprocesadores en el area de Instrumentación de la Escuela.

Estos objetivos han sido cumplidos. El primer objetivo se cumplió construyendo una CPU elemental basada en el Z80 mas una interfase de comunicación protocolaria asincrona que permite la comunicación CPU Z80 PC AT LEMMON.

El hardware por si solo no trabaja en sistemas con microprocesadores, por tanto se creo un programa monitor para que ejecute diversos comandos y atrape la información que se manda desde la PC.

Por otra parte desde la PC se procesa la informacion que se manda al sistema a partir de un programa de computador, permitiendo mandar así información coherente que interpreta el módulo.

El sistema puede aumentar sus capacidades y dependerá de los usuarios lograr su evolución dado que puede modificarse para efectuar tareas diferentes a la cual fue concebido.

Debido a lo anterior, se puede afirmar que el sistema es versatil, de bajo costo y tiene proyección de desarrollo. Creemos que para la instrumentacion electrónica representa una buena alternativa para el aprendizaje de sistemas basados en microprocesadores.

RESUMEN DEL TRABAJO

La implementación de este dispositivo puede resumirse de la siguiente manera: Comunicación de la información desde la PC y el procesamiento de la información transmitida en base a los comandos que se le manden.

En el capítulo I, se hace un breve estudio de los sistemas con microprocesadores y se define un procedimiento de diseño para estos sistemas autosostenidos. Se explica también los diferentes recursos que se necesitan para su materialización.

En el capítulo II, se detallan las partes fundamentales de los sistemas con microprocesadores y se describe su funcionamiento y de lo necesario que son en estos sistemas. Llegándose así a un primer diseño. Este diseño abrió el camino para poder construir el equipo actual.

El capítulo III establece los circuitos finales que se han implementado y se describen los elementos que los conforman, se establece además la forma de comunicación entre la PC y el módulo que es una de las partes más importantes de este dispositivo.

El capítulo IV describe el sistema de software diseñado para poder efectuar las tareas para cumplir el objetivo que se nos encomendó. Allí se detalla la forma en que se transmite la información, así como del formato con el cual se manda la información.

Este diseño es un prototipo, que fue concebido para ser utilizado con la PC AT LEMMON 296S o IBM. Los criterios empleados son obviamente la economía y ha resultado tener una eficiencia muy buena, permitiendo aumentar sus capacidades en el futuro.

Los usos que puede tener el equipo son diversos, ellos van desde control de motores por microprocesador, convertirse en controlador programable, etc. De ahí se manifiesta su versatilidad para diferentes usos.

TABLA DE CONTENIDOS

Capítulo	Página
I. SISTEMAS BASADOS EN MICROPROCESADOR Y PROCESO DE DISEÑO PARA SU IMPLEMENTACION.....	1
1.1 Los microprocesadores y las microcomputadoras	1
1.2 Aplicaciones de los microprocesadores.....	2
1.2.1 Aplicaciones industriales.....	2
1.2.2 Aplicaciones comerciales.....	3
1.2.3 Aplicaciones orientadas al consumidor....	3
1.3 Arquitectura de computadores y diseño de sistemas	4
1.3.1 Principios de microcomputadoras.....	4
1.3.2 Arquitectura de la CPU.....	5
1.3.3 La memoria de sólo lectura (ROM).....	7
1.3.4 La memoria de lectura/escritura.....	7
1.3.5 Entradas y salidas digitales y analógicas	7
1.3.6 Controlador de interrupciones.....	8
1.4 Ventajas y limitantes del computador.....	8
1.5 Estados en el diseño de sistemas basados en microprocesadores.....	10
1.5.1 Estados de análisis y diseño.....	10
1.5.2 Estado del diseño de hardware.....	11
1.5.3 Estado del diseño del software.....	12
1.5.4 Estado del desarrollo del programa y pruebas.....	13
1.5.4.1 Programación modular.....	13
1.5.4.2 Soportes para el desarrollo del sistema.....	13

1.5.4.3 Desarrollo del programa.....	14
1.6 Descripción formal de las especificaciones del diseño y construcción	15
1.6.1 Objetivos.....	16
1.6.2 Especificaciones para materializar el módulo.....	16
1.7 Conclusiones y recomendaciones.....	17
Referencias Bibliográficas.....	18
II. COMPONENTES BASICOS DE UN SISTEMA CON MICROPROCESADOR.....	19
2.1 Componentes minimos de un sistema generalizado.....	19
2.1.1 Fuente de alimentación.....	19
2.1.2 Reloj del sistema.....	20
2.1.3 Compuertas lógicas y decodificadores...	?
2.1.4 Latches.....	?
2.1.5 Buffers.....	27
2.1.6 Memoria.....	29
2.1.7 Sistema minimo basado en microprocesador	32
2.1.8 Computadora en una tarjeta.....	33
2.2 Sistema minimo basado en el Z-80.....	34
2.2.1 Componentes del sistema básico inicial..	34
2.2.2 Reloj del sistema Z80 minimo	35
2.2.3 Circuito de reset del Z-80.....	36
2.2.4 Amplificación de los buses del sistema minimo Z-80.....	37
2.2.5 Decodificación de memoria y puertos de E/S.....	38
2.2.6 Memoria del sistema Z-80 minimo.....	40

2.2.7 Puertos de E/S del sistema Z-80 minimo..	41
2.3 Conclusiones y recomendaciones.....	44
Referencias bibliográficas.....	45
III. DISEÑO Y CONSTRUCCION DEL SISTEMA DIDACTICO MINIMO Z-80.....	46
3.1 Módulo principal del sistema.....	43
3.1.1 Componentes del módulo principal del sistema didáctico.....	47
3.2 Decodificador de direcciones para entrada salida.....	54
3.3 Diseño y construcción de la interfase de la interfase de comunicación en la CPU Z80..	53
3.3.1 Comunicación entre dispositivos de velocidad media.....	57
3.3.2 Procedimiento de comunicación.....	57
3.3.3 Descripción de pines del 8255.....	58
3.3.4 Descripción de las señales del conector de la CPU con el Z80.....	59
3.3.5 Diseño de la interfase paralelo para comunicación Z80-PC.....	60
3.4 El desplegador del sistema basado en el Z80.	65
3.4.1 Descripción del TIL 311.....	65
3.5 Conexión del circuito PIO y CTC.....	67
3.5.1 Descripción de patillas del circuito PIO	67
3.6 Conclusiones y recomendaciones.....	70
Referencias bibliográficas.....	71
IV. OPERACION Y APLICACIONES DEL SISTEMA.....	72
4.1 Operación del sistema didáctico mínimo Z-80.	72
4.1.1 Examinar memoria.....	73

4.1.2	Cambiar contenido de memoria.....	73
4.1.3	Introducción de datos.....	74
4.1.4	Ejecución de un programa.....	74
4.1.5	Examinar registros.....	75
4.1.6	Alterar el contenido de registros.....	75
4.1.7	Cargar un programa a partir de la PC....	76
4.1.8	Comando de trazado de registros.....	76
4.1.9	Establecer/Borrar un punto de ruptura...	77
4.1.10	Habilitar/ Deshabilitar Breakpoints....	78
4.2	Posibles aplicaciones del Sistema didáctico.	78
4.3	Conclusiones y recomendaciones.....	79
	Referencias bibliográficas.....	80
	CONCLUSIONES Y RECOMENDACIONES GENERALES.....	81
ANEXO A1	SOFTWARE Y DIAGRAMAS DEL CIRCUITO PARA LA IMPLEMENTACION DE UN SISTEMA MINIMO INICIAL BASADO EN EL Z80.....	82
ANEXO A2	CIRCUITOS DEL SISTEMA DIDACTICO MINIMO BASADO EN EL Z80 Y ALGORITMOS DEL SOFTWARE IMPLEMENTADO.....	85

CAPITULO I

- SISTEMAS BASADOS EN MICROPROCESADORES Y PROCESO DE DISEÑO PARA SU IMPLEMENTACION

Introducción

Las microcomputadoras han influenciado profundamente el avance y vistosidad de muchas disciplinas; especialmente la de INSTRUMENTACION Y CONTROL. Su implementación no solo ha acarreado la simplificación del hardware y mejoramiento de la confiabilidad; sino también permitió la optimización de su funcionamiento y las capacidades de diagnóstico poderosos. Lo que no hubiese sido posible por hardware dedicado.

Las características de las computadoras cada vez son mejores en términos de arquitectura, velocidad de cálculo, tamaño y otros. Y esta tendencia continuará.

Las computadoras han sido ahora universalmente aceptadas como instrumentos de gran ayuda y por tanto el espectro de aplicaciones crecerá continuamente. El objetivo de este capítulo, es el de presentar los conceptos relacionados con los sistemas basados en microprocesadores tales como la arquitectura de computadores y del microprocesador, y fundamentalmente establecer los estados en el diseño de sistemas con microprocesador.

1.1 Los microprocesadores y las microcomputadoras

Cualquier sistema de computadora, desde el más simple controlador hasta el probador más sofisticado, consiste de algunos subsistemas bien definidos. Entre ellos se tienen:

- a. La CPU
- b. El subsistema de memoria
- c. Interfase de entrada/salida

La Fig. 1.1 muestra el diagrama del sistema de microcomputador fundamental. Los subsistemas de la CPU ejecutan las funciones clásicas, tales como las funciones aritméticas, lógicas y de control. Los subsistemas de memoria contienen los programas y los datos sobre los cuales la CPU opera. Las interfases de entrada/salida representan la comunicación entre la computadora y el mundo exterior. En este contexto queda bien definida la concepción del microprocesador y de la microcomputadora.

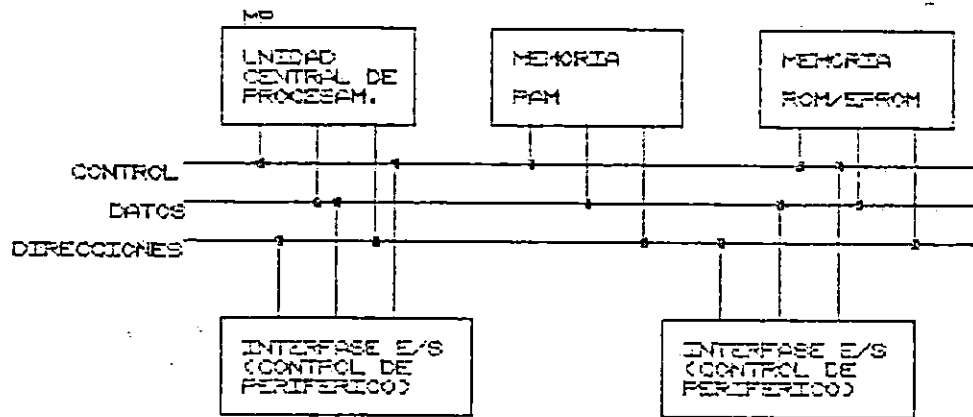


FIG. 1.1 DIAGRAMA DE BLOQUES DEL MICROCOMPUTADOR

1.2 Aplicaciones de los microprocesadores

La evolución del microprocesador es ciertamente una revolución en las aplicaciones. Para el año 2000, de 5 a 10 billones de microprocesadores y microcomputadoras estarán en servicio(alrededor de 1 por cada persona sobre la tierra). En la mayoría de los casos, las personas interactuando con el microprocesador o microcomputador no sabrá que encierra una computadora, simplemente sabrá que tiene un instrumento útil a la mano.

Las aplicaciones de los microprocesadores caen en tres amplias categorías: industrial, comercial y del consumidor. Las aplicaciones industriales incluye aquellas que emplean microprocesadores en el diseño o en procesos de manufactura y control. Las aplicaciones comerciales son las orientadas a proporcionar nuevos servicios. Las aplicaciones del consumidor incluye aquellas que agrega capacidades a productos que ya posee el consumidor.

1.2.1 Aplicaciones industriales

Los microprocesadores están teniendo un gran impacto sobre aplicaciones industriales, incluyendo las áreas de experimentación, control, instrumentación, adquisición de datos y la robótica.

Experimentación. Los sistemas de prueba basado en microprocesadores, pueden ser usados para identificar automáticamente componente ,subsistemas y fallas de sistemas.

Son usados también para mejorar la calidad y reducir los costos de fabricación. La llegada de microprocesadores ha abierto diversos caminos para los probadores dedicados, lo cual justifica económicamente a la industria.

Control de Procesos y Adquisición de Datos: Estas áreas difieren de lo anterior en que ellas deben operar en tiempo real. Esto quiere decir que el microprocesador tiene que ser suficientemente rápido para adaptarse al proceso de control o al sistema adquisitor.

Ciertamente con la disminución de los costos, se ha hecho factible de sistemas enteramente dedicados; además, los sistemas son más flexibles, ya que con solo cambiar algunas líneas de software puede alterarse el tiempo de muestreo, especificar nuevos formatos de datos, etc.

Instrumentación: Al igual que en las áreas anteriores, el microprocesador ha impactado en el área de instrumentación. Agregando nuevas capacidades que van desde mejores presentadores hasta la autocalibración del instrumento.

Otras aplicaciones son control de máquinas, robótica y también ha incursionado en el área de educación proporcionando instrumentos dedicados.

1.2.2 Aplicaciones comerciales

Las aplicaciones comerciales de los microprocesadores incluyen sus usos en las comunicaciones, aplicaciones médicas y aplicaciones en los negocios. La aplicación más significativa en las comunicaciones es la conversión analógica/digital en telefonía.

Los microprocesadores están siendo usados en aplicaciones médicas. Debido a aspectos legales y técnicos su uso en diagnósticos, monitoreo o tratamiento, los microprocesadores están siendo usados sobre una base experimental. Otras aplicaciones comerciales incluyen su uso en sistemas de seguridad, sistemas de control ambiental, controladores de tráfico, etc.

1.2.3 Aplicaciones orientadas al consumidor.

Nuevas funciones están siendo agregadas a productos estándar y nuevas categorías de productos están siendo posibles. El caso popular es el de la computadora, donde ha habido una explosión de productos accesorios, tales como impresores, plotters, set de productos orientados al aficionado. Otras aplicaciones son los sistemas de seguridad inalámbrico, calendarios/agendas electrónicos.

Se puede concluir que el impacto de los microprocesadores en la vida humana es bastante grande, de tal manera que es inmesurable el impacto social.

1.3 Arquitectura de computadoras y diseño de sistemas

Debe tenerse presente que el objetivo de este trabajo no consiste en diseñar formalmente una computadora, sin embargo es necesario establecer un marco de referencia para el instrumento, ya que el objetivo de este trabajo es el de diseñar y construir un sistema mínimo basado en microprocesador, el Z80, y por tanto tiene en su esencia grandes similitudes con los computadores, por lo cual se dedicarán las siguientes secciones a explicar los conceptos relacionados con las computadoras.

1.3.1 Principios de microcomputadoras

Aunque puntualmente el concepto de microcomputadora no está bien definido debido a que con el cambio de la tecnología, su tamaño y sus capacidades han cambiado, es aceptable decir que la microcomputadora es la responsable de las operaciones de cálculo y toma de decisiones dentro de un sistema. El corazón de una microcomputadora es la unidad de procesamiento central (CPU) o microprocesador. Además de la CPU, una microcomputadora tiene los siguientes elementos (aquí serán discutidos brevemente):

- * El generador de reloj.
- * Manejadores del bus de datos y direcciones (opcional).
- * Memoria de solo lectura (ROM).
- * Memoria de lectura/escritura (RAM).
- * Controlador de interrupciones.
- * Entrada/salida digital.
- * Entrada/salida analógica.

Un diagrama típico de bloques de una microcomputadora es mostrado en la Fig. 1.2. La CPU es unida a los otros elementos a través del bus de direcciones, el bus de datos y el bus de control. El ancho del bus de datos normalmente determina el tamaño de la palabra de una computadora. El bus de direcciones especifica una localidad particular de memoria, el bus de control determina la operación (Lectura o Escritura), y el bus de datos porta los datos o instrucciones desde una posición de memoria. Algunas líneas del bus de direcciones pueden ser empleadas para generar "chip select" (habilitadores) de elementos periféricos.

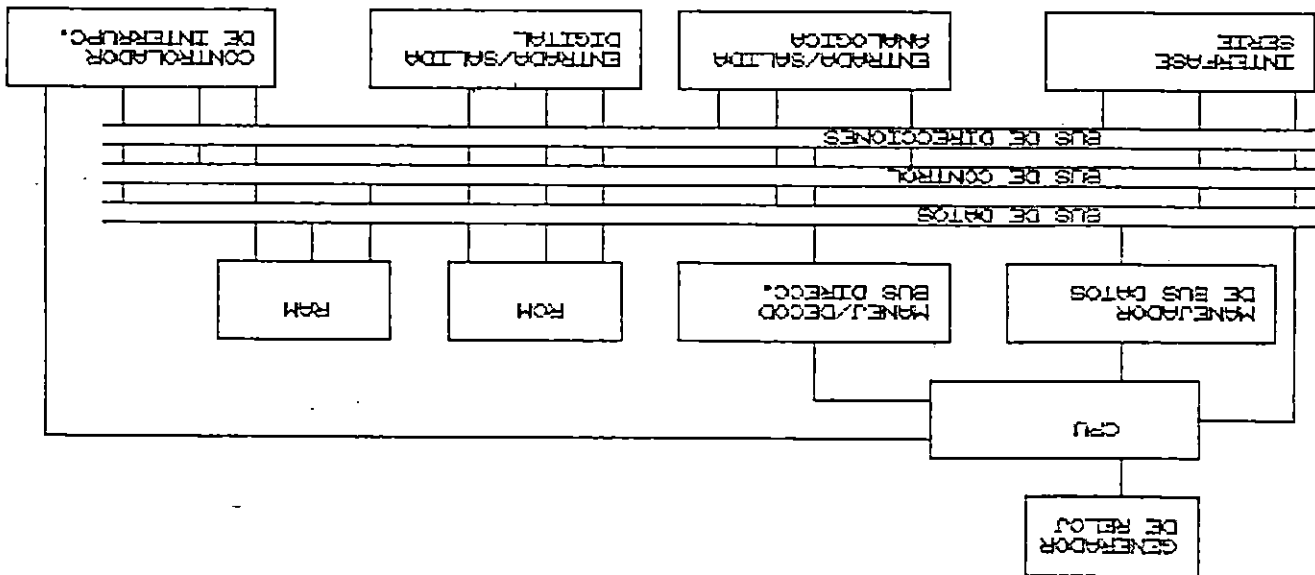
Los componentes funcionales básicos de una CPU son mostrados en la Fig. 1.3. El acumulador registra los contenidos que serán manipulados por la unidad aritmética y lógica (ALU).

1.3.2 Arquitectura de la CPU.

Por otra parte, el programa de una microcomputadora consiste de instrucciones las cuales son usualmente almacenadas en ROM. Los datos a ser procesados por las instrucciones son almacenados en RAM o accedidos a través de entradas analógicas/digitales. Los datos procesados pueden ser almacenados en memoria o mandados al mundo exterior a través de las salidas analógicas/digitales. La CPU va a traer las instrucciones en secuencia, las decodifica y ejecuta las operaciones aritméticas y lógicas. El controlador de interrupciones interrumpe la operación normal del procesador y lo dirige a ejecutar una rutina que necesite urgente atención.

Los buses pueden utilizar manejadores en el caso de que haya mucha carga. El generador del reloj proporciona los pulsos de reloj a la CPU tal que todas las operaciones del microprocesador ocurren en una forma sincronizada. Todos los dispositivos de memoria y periféricos unidos CPU deberían tener un tiempo de acceso compatible tal que las operaciones de lectura/escritura puedan ser ejecutadas exitosamente en el tiempo especificado.

FIG. 1.2 DIAGRAMA DE BLOQUES DE UNA MICROCOMPUTADORA



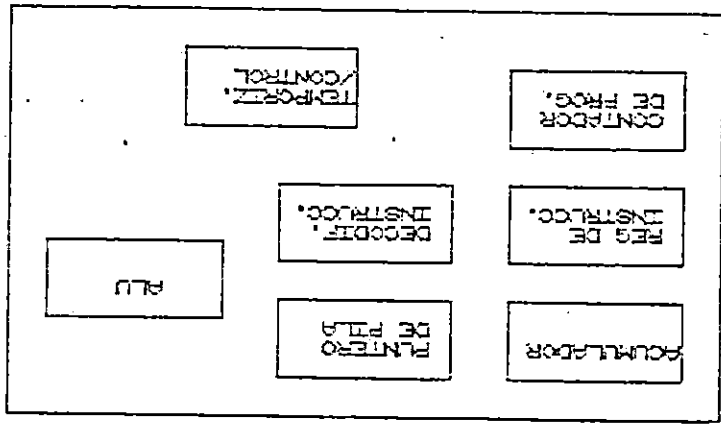
En una llamada a subrutina, el PC es incrementado y su contenido es almacenado en una área reservada de memoria conocida como "stack". El stack pointer (SP) mantiene la dirección de la entrada más reciente. Al efectuarse un retorno después de la ejecución de una subrutina, el PC empieza con la dirección recuperada del SP.

Esta regla es violada solo cuando la CPU ejecuta una instrucción de salto o ejecuta una llamada a subrutina. direcciones numéricamente adyacentes puedan ser encontradas. que trae una nueva instrucción tal que las instrucciones en instrucción a ser ejecutada. La CPU incrementa el PC cada vez El contador de programa (PC) almacena la dirección de la

La ALU, como su nombre lo indica, ejecuta operaciones aritméticas y lógicas sobre datos binarios. La mayoría de las ALU's proporcionan funciones internas, tales como suma, resta, operaciones booleanas y capacidades de desplazamiento. La ALU genera indicadores (banderas) tales como: bit de acarreo, cero, signo y paridad como resultado de manipulaciones aritméticas y lógicas.

La CPU podría contener registros adicionales de propósito general los cuales incrementan la eficiencia en cuanto a la manipulación de los datos y mejoran la velocidad de procesamiento.

FIG. 1.3 ARQUITECTURA BASICA DE LA CPU



Una instrucción típica podría ser sumar el contenido de algún registro al contenido del acumulador y almacenar el resultado en el acumulador.

Una microcomputadora puede dar servicio a un número ilimitado de subrutinas de una manera anidada. El registro de instrucciones (IR) almacena la instrucción traída por la CPU. El decodificador de instrucciones traduce el código de la instrucción para ejecutar la tarea específica. Las unidades de tiempo y control permiten la operación de la CPU en una manera ordenada.

1.3.3 La memoria de sólo lectura (ROM)

Como se mencionó previamente, una ROM almacena el programa monitor de una microcomputadora. El programa puede contener datos importantes o coeficientes que no pueden alterarse. Como su nombre lo indica, las posiciones de memoria solo pueden leerse. La memoria no es volátil porque la falla de la fuente no destruye el contenido de memoria. Un programa, el cual no necesita alteración es almacenado en ROM. En el estado inicial del desarrollo, un programa puede necesitar alteración; entonces, debe ser almacenado en una memoria EPROM.

1.3.4 La Memoria de Lectura/escritura (RAM).

Una RAM normalmente almacena datos que son generados por la ejecución del programa. Ambas, la ROM (como la EEPROM) y la RAM tienen iguales capacidades de acceso, pero la RAM es generalmente usada para operación de lectura/escritura. Una RAM tiene almacenamiento volátil.

La memoria RAM puede ser estática o dinámica. Una RAM estática usa flip-flops como dispositivos de almacenamiento, los cuales tienen alta disipación de potencia, y entonces proporciona almacenamiento a baja densidad. Una RAM dinámica usa un pequeño capacitor como dispositivo de almacenamiento de alta densidad, pero necesita refrescamiento periódico para mantener la información.

1.3.5 Entradas y salidas digitales y analógicas

Una microcomputadora se comunica con el mundo externo a través de dispositivos de E/S. Las señales de E/S digitales pueden tomar diversos tamaños (byte, nibble) dependiendo de la aplicación particular.

Las señales de entrada/salida digitales pueden ser controladas de igual manera como se maneja la información de la memoria o por simples instrucciones de entrada/salida (IN/OUT).

Los puertos de entrada/salida normalmente tienen operaciones específicas; por ejemplo, el interfase de comunicación serie transmite o recibe datos en forma serie, pero internamente, la microcomputadora maneja los datos en forma paralelo.

Son ejemplos comunes de dispositivos periféricos: la terminal de video, los manejadores de disco, impresores ... etc. Por otra parte, las señales analógicas de un sistema físico son mandadas al computador a través de convertidores analógico/digital (A/D) o digital/analógico (D/A). Un dispositivo de entrada/salida analógica puede ser bipolar o unipolar, y el tamaño digital de la información define la precisión o resolución de la señal. El tiempo de conversión de un convertidor A/D debería ser corto comparado con el tiempo de muestreo. En un adquisitor de datos analógico, múltiples señales analógicas son accesibles al microcomputador a través de multiplexores de señales analógicas y un simple convertidor A/D.

1.3.6 Controlador de interrupciones

El controlador de interrupciones manipula las solicitudes de los dispositivos externos. Cuando la CPU recibe una señal de interrupción, suspende la secuencia normal de operación, ejecuta la subrutina de atención de interrupciones y luego retorna al programa principal. Si un evento no es crítico, puede detectarse por el método "polling" (de escrutinio), mientras la CPU periódicamente muestrea las señales lógicas y prueba sus transiciones. Una interrupción puede iniciarse por software o por hardware (internamente); o por un sistema externo. Cuando una señal de interrupción se recibe la CPU completa la actual instrucción, guarda el estado actual (próxima dirección, registros internos) y atiende a la rutina de servicio de interrupciones. Un controlador de interrupciones puede manipular un número de señales de interrupción sobre la base de una prioridad predefinida en una manera anidada. Las señales de interrupción puede ser mascarable o no mascarable, dependiendo de la prioridad de la interrupción.

1.4 VENTAJAS Y LIMITANTES DEL MICROCOMPUTADOR.

La microcomputadora, o en general el diseño de un sistema digital, tiene sus méritos y limitantes, las cuales pueden resumirse así:

A. Costo del hardware bajo

La simplificación del hardware y la correspondiente reducción del costo comparado con los sistemas analógicos es la principal ventaja de los sistemas digitales. Esta tendencia puede observarse con la explosión del mercado de las

computadoras. Circuitos VLSI de consumo con integración de hardware de control para aplicaciones específicas en gran volumen de producción pueden ser muy económicos. Menor peso y tamaño con consumo de potencia reducido son ventajas adicionales.

B. Confiabilidad mejorada

La confiabilidad del sistema puede ser mejorada considerablemente por los sistemas digitales. La confiabilidad de un circuito LSI o VLSI es considerablemente mayor que los sistemas con gran número de componentes electrónicos. La experiencia pasada ha mostrado que los computadores tienen mayor confiabilidad que otros sistemas electrónicos. La confiabilidad puede ser mejorada usando componentes con especificaciones militares.

C. Disminución de los problemas por interferencia electromagnética (EMI)

La integración de alto nivel de una computadora evita el acoplamiento de grandes voltajes y transitorios de corriente en sistemas de potencia. Solo la protección nominal del hardware es suficiente. Los acoplamientos de ruido a través de fuentes y señales de entrada pueden minimizarse proporcionando un buen filtrado.

D. No fugas e invarianza de parámetros

El procesamiento de señales digitales elimina fugas de señales y los efectos por variación de parámetros, los cuales prevalecen en sistemas analógicos. Los cálculos digitales son 100% precisos, y los problemas de sobreflujo son evitados seleccionando escalas apropiadas.

E. Software y hardware universal

El hardware universal puede ser diseñado para cierta clase de sistemas, donde el software puede ser modificado para satisfacer cierta clase de especificaciones. Por ejemplo, un manejador puede tener el mismo hardware pero diferentes especificaciones de funcionamiento pueden ser incorporados por modificaciones al software.

F. Diagnósticos

Una ventaja adicional de los sistemas basados en microprocesadores es que un poderoso software puede ser diseñado y que puede ser usado por personal técnico. Además adquirentes de datos, monitores de señales y desplegados pueden proporcionarse.

G. Cálculos relativamente lentos

Cuando se implementan funciones con una microcomputadora, las capacidades son más lentas que hardware analógico dedicado. La razón es que la microcomputadora calcula una función de forma serie, mientras hardware dedicado procesa las señales de forma paralela. Si una microcomputadora maneja varias

tareas en forma multiplexada, es decir, el tiempo tiene que compartirse; y esto disminuye la velocidad de ejecución. Grandes retardos causan deterioro de funcionamiento y en control de sistemas de potencia puede producir inestabilidad. Sin embargo, las microcomputadoras están incrementando su velocidad de cálculo. Esta tendencia continuará en el futuro.

H. Cuantización del error

Las señales en sistemas físicos son analógicos en naturaleza, y cuando estos son conectados con la microcomputadora por convertidores A/D y D/A, existe un error en la conversión. La distorsión de la señal debido a la cuantización mas los retardos de las señales (samples) puede ser evidente. El error en la cuantización puede minimizarse aumentando el tamaño de la palabra de la computadora.

I. Desarrollo del software puede ser caro

El desarrollo del diseño y depuración del software, especialmente en lenguaje ensamblador puede consumir bastante tiempo; por ejemplo, el tiempo que se invierte en un programa en lenguaje C, puede multiplicarse por tres cuando se desarrolle en ensamblador. Y así puede ser caro. El costo adicional puede ser justificado desde el punto de vista de las ventajas totales. El costo incremental para grandes volúmenes de producción podría no ser notable.

1.5 Estados en el diseño de sistemas basados en microprocesadores

El diseño del sistema empieza una vez las especificaciones del funcionamiento están disponibles en detalle. Generalmente, involucra los siguientes estados:

- * Análisis y diseño teórico del sistema.
- * Simulación por computadora del sistema (si es necesario).
- * Construcción del hardware.
- * Implementación del software.
- * Integración, depuración y pruebas del sistema.

La Fig. 1.4 muestra los estados de diseño en la forma de flujograma.

1.5.1 Estados de análisis y diseño

En primer lugar, la estructura del sistema es formulada en base a las especificaciones de funcionamiento del sistema. Todas las funciones son entonces identificadas, y así es seguida por análisis detallado y diseño de cada función. Como ejemplo de estas funciones están el monitoreo de datos, protección, diagnóstico, etc.

Las especificaciones del sistema puede necesitar que se redefinan después de un análisis del sistema. Para un sistema complejo, la simulación digital puede ser necesaria. El estado de análisis y diseño es terminado por algún esquema detallado para el desarrollo de todas las funciones.

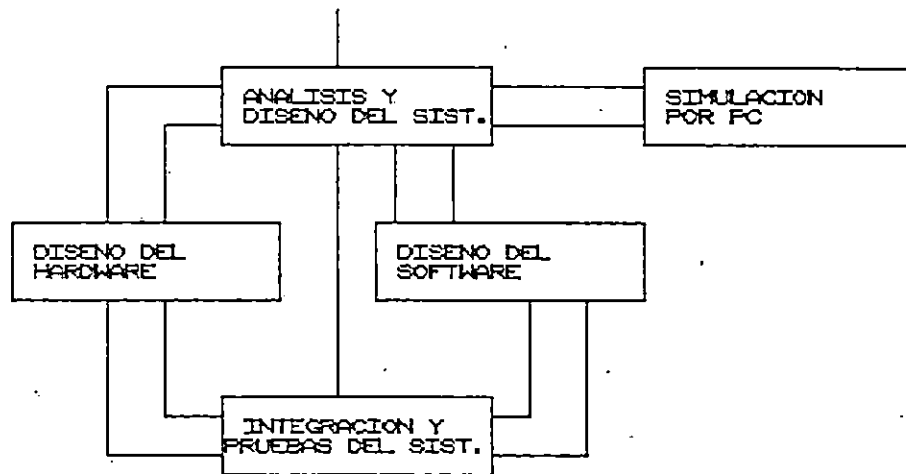


FIG 1.4 ESTADOS DEL DISEÑO BASADO EN MICROPROCESADOR

1.5.2 Estado del diseño de hardware

El hardware del sistema consiste de hardware de microcomputador y de periféricos dedicados. Una vez más, el hardware de microcomputador consiste de un microconprocesador (CPU) y circuitos periféricos. Las limitantes del hardware queda supeditado por la velocidad de cálculo de la CPU. Un adecuado microprocesador puede ser elegido después de que las funciones del sistema hayan sido analizadas.

Algunos de los criterios de selección son el tamaño de palabra, velocidad de ejecución, capacidades directas de multiplicación/división, disponibilidad para ambiente industrial/militar, existencia de software para desarrollo del sistema a partir de computadoras personales.

El tamaño de las memorias ROM y RAM debería diseñarse conservativamente y son determinadas por el tamaño del programa y los datos. Un programa basado en lenguaje de ensamble necesita de algunos kbytes de memoria EPROM; en vez de ROM permite un flexible desarrollo de programas en el estado inicial. Por otra parte la E/S del sistema debería ser definida.

Para funciones de alta velocidad, una microcomputadora puede

ser auxiliado por hardware dedicado, tales como contadores, procesadores de E/S, etc. Las técnicas de comunicación deben ser consideradas; varias técnicas de comunicación son: transferencia serie, E/S paralelo con handshake, transferencia DMA, transferencia FIFO y multibus con lógica de control de bus.

1.5.3 Estado de diseño del software

Después de identificar las funciones del sistema, sus algoritmos deben ser descritos en detalle. Un algoritmo es un método mediante el cual una acción es implementada en esencia digitalmente en una computadora. Normalmente, el algoritmo es descrito en forma de flujograma, el cual es luego traducido en un programa.

El software consiste de un set de módulos de programas los cuales son ejecutados en tiempo real en forma secuencial. El diseño de software involucra un número de pasos que pueden describirse como sigue:

i. Selección del lenguaje

Existen funciones del sistema cuya implementación son muy críticas en relación al tiempo. Entonces, estas son generalmente implementadas en ensamblador. El lenguaje ensamblador es lo más cercano al lenguaje de máquina; de ahí que un programa escrito en assembler tiene una rápida ejecución y ocupa poca cantidad de memoria.

Por otro lado, el desarrollo de un programa en lenguaje de ensamble consume considerable cantidad de tiempo y es tedioso. Un lenguaje de alto nivel, tal como C, Pascal, Basic etc. puede elegirse pensando en que el desarrollo de programas consume menos tiempo. Sin embargo, un programa en lenguaje de alto nivel es lento en ejecución y ocupa mas memoria.

Es posible mezclar lenguaje ensamblador y lenguaje de alto nivel en un programa dependiendo de las restricciones del tiempo de ejecución y el tamaño de la memoria.

En tal caso, segmentos de programas en diferentes lenguajes pueden ser separadamente compilados y ensamblados, y luego enlazados y localizados para constituir un código objeto total.

ii. Estructura del software

Un primer paso en el diseño del software es identificar las funciones del microcomputador. Los correspondientes módulos de software deben ser diseñados. Todas las funciones de software son así agrupadas en varias tareas, de tal manera que se emplee la modularidad y la técnica de programación TOP-DOWN.

1.5.4 Estado del desarrollo del programa y pruebas:

El desarrollo del programa empieza luego que los algoritmos de las funciones han sido desarrolladas y descritas en forma detallada (flujogramas, pseudocódigos). Los programas pueden ser desarrollados en lenguaje de alto nivel, lenguaje ensamblador o por una mezcla de ambos como previamente se ha dicho.

1.5.4.1 Programación modular

Los programas del sistema son a menudo demasiados largos y complejos para ser desarrollados como una simple unidad. Por otra parte, es conveniente dividir el programa en pequeñas unidades funcionales y luego desarrollar un programa de cada unidad en forma modular.

Los programas modulares son más fáciles de codificar, depurar y alterar que los programas no modulares, y tales programas pueden ser desarrollados independientemente y en paralelo. La programación modular es similar al diseño de hardware electrónico que contiene numerosos circuitos funcionales. Cada módulo de programa puede ser considerado como una "caja negra" con entradas y salidas especificadas, y estas cajas negras pueden ser luego unidas para completar el programa total.

1.5.4.2 Soportes para el desarrollo del sistema

Estas consisten de las herramientas para implementar el software. Se trata de un sistema operativo residente el cual permite el desarrollo de programas fuentes, manejo y manipulación de archivos, traducción de programas fuentes en programas objetos, y luego enlazar y colocar los programas objetos en el código objeto absoluto.

El desarrollo del software puede también ser posible con el auxilio de las siguientes unidades:

- * Una microcomputadora con ROM, RAM, monitor y teclado.
- * Una unidad de disk drive mas impresor.
- * Un programador de EPROM.
- * Una tarjeta de comunicación serie o paralelo

La Fig. 1.5 muestra un diagrama de bloques del sistema de soporte incluyendo las unidades perifericas.

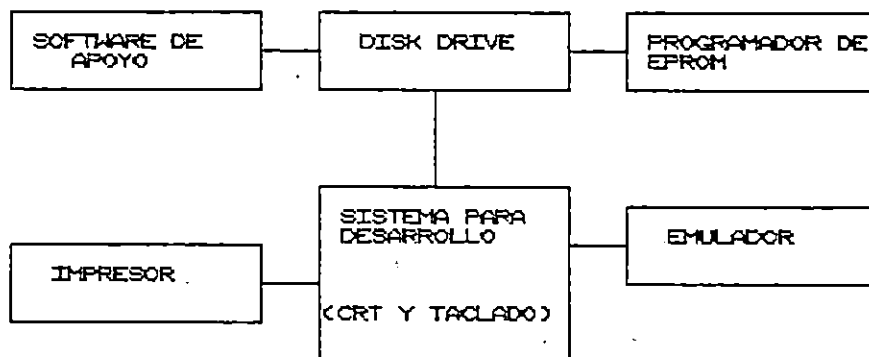


FIG. 1.5 SOPORTE PARA DESARROLLO DE SISTEMAS BASADOS EN MICROPROCESADOR

1.5.4.3 Desarrollo del programa

El flujograma para el desarrollo de programas es mostrado en la Fig. 1.6. Una vez los módulos de los programas se han escrito, estos son guardados en discos a partir de editores de texto. El editor permite corregir los archivos dependiendo de las necesidades. El ensamblador, luego, traduce el código fuente en código objeto.

El programa en lenguaje ensamblador tiene tres partes: instrucciones de máquina, directivas y controles del programa ensamblador. Una instrucción de máquina es un código de máquina que puede ser ejecutado por la máquina. Las directivas del ensamblador son usadas para definir la estructura del programa y los símbolos. Los controles del ensamblador fijan los modos del programa ensamblador y dirige el flujo del ensamblador.

La invocación del ensamblador indicará los errores de sintaxis en el programa fuente. Después de corregir los errores el ensamblador producirá un archivo objeto y un listado mostrando los resultados del ensamble. El archivo objeto puede ser relocable o absoluto. El módulo objeto contiene instrucciones del lenguaje y datos que pueden ser cargados a la memoria del computador para la ejecución. Los módulos objetos relocables son entonces procesados a través del enlazador.

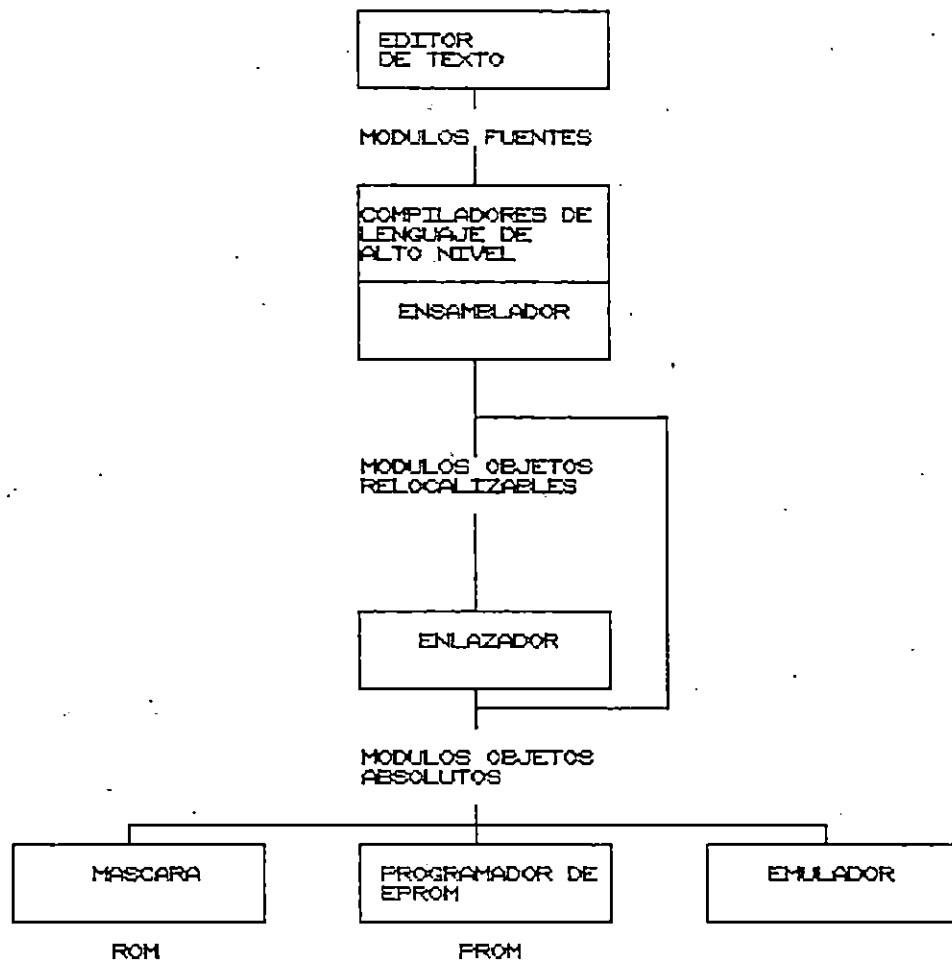


FIG 1.6 FLUJOGRAMA PARA DESARROLLO DE PROGRAMAS

1.6 Descripción formal de las especificaciones del diseño y construcción

Tal como se establece en la sección 1.5 de este capítulo, el primer paso de diseño en sistemas basados en microprocesadores, es identificar claramente las funciones del sistema que se desea diseñar. Entonces el objetivo de la presente sección será la de definir las especificaciones que debe cumplir el diseño y construcción de un sistema didáctico mínimo z80 controlado por la AT LEMMON 286S.

1.6.1 Objetivos

En las secciones anteriores se ha dado un bosquejo del ambiente de los microprocesadores. En esta sección se procederá a establecer las especificaciones del diseño para tener una visión clara de las funciones que realizará y empezar de esta manera la materialización del equipo. Los objetivos que deberá cumplir son los siguientes:

1. Proveer a la Escuela de Ingeniería Eléctrica de la Universidad de El Salvador con una herramienta de hardware elemental para la enseñanza de sistemas programables a nivel básico.
2. Promover el diseño de sistemas autosostenidos con microprocesadores en el área de instrumentación de la Escuela de Ingeniería Eléctrica.

1.6.2 Especificaciones para materializar el Módulo

Los siguientes numerales establecen puntualmente todos los requisitos primordiales para identificar las funciones del módulo:

1. Diseñar y construir el módulo mínimo fundamental del sistema, el cual incluirá por lo menos los componentes siguientes:

- Un microprocesador Z80.
- Una memoria ROM.
- Una memoria RAM.
- Un PIO y un CTC (circuitos de soporte).
- Seis desplegados de 7 segmentos de 1" para datos y contenidos de registros y direcciones
- Circuitos de soporte.

2. Diseño de un programa para comunicarse con el módulo a través de la interfase paralelo basada en el IC-8255A (existente). Tanto el módulo como el programa deben distinguir si lo que se envía es un dato, un comando o dirección. El programa debe ser capaz de aceptar los nemónicos y operandos, traducirlos a hexadecimal y enviarlos un byte tras otro al módulo. En el caso de comandos, el módulo debe interpretarlos apropiadamente realizando la función requerida por medio del programa monitor residente. Los programas de usuario se almacenarán en la RAM del módulo.

3. Los comandos mínimos que el sistema deberá ejecutar son:
RUN, SET MEMORY, DISPLAY MEMORY, SET REG, SET BREAK POINT,
DISPLAY REG, TRACE, PROGRAMAR

4. El módulo quedará en circuito impreso, con puntos accesibles para la visualización de señales principales con el osciloscopio.

5. Incluir una guía de usuario con ejemplos de uso.

1.7 Conclusiones y recomendaciones.

1. Independiente del área de aplicación tecnológica, un buen laboratorio es esencial para producir buenos resultados. Esto exige una cantidad de personas idóneas para el buen funcionamiento del sistema educativo y así es necesario proyectar los trabajos de investigación en el área de instrumentación de la Escuela de Ingeniería Eléctrica.

2. La creación de sistemas basados en computadores permite que los estudiantes tengan contacto con sistemas electrónicos autosostenidos y facilita poner en práctica los conocimientos para resolver problemas reales con diversos conocimientos.

3. Económicamente los sistemas basados en microprocesador han disminuido grandemente y se estima que en el futuro cercano los costos seguirán disminuyendo de tal manera que las personas estarán interactuando con tales sistemas mas frecuentemente.

4. La investigación orientada a la creación de dispositivos para la enseñanza académica es importante porque genera nuevos mecanismos de aprendizaje y crea alternativas de instrumentación económicas para la Escuela de Ingeniería Eléctrica.

5. Las etapas de diseño de sistemas basados en microprocesador están bien definidas, sin embargo existen algunas limitantes en cuanto a la materialización de estos equipos debido a la inexistencias de técnicas propias en su construcción. Recordemos que los circuitos que se emplean son contruidos por tecnología de integración a gran escala.

6. El diseño e implementación del dispositivo por su esencia tendrá diversas aplicaciones y por tanto es un instrumento de gran versatilidad.

7. Por orden de prioridades se han considerado en cada etapa del diseño : el factor económico, simplicidad y sencillez, factibilidad y otros; como guías en las decisiones de lo que se debe incluir o no.

Referencias Bibliográficas

1. A. S. Tanenbaum, Arquitectura de Computadoras, Prentice Hall, 1984.
2. B. K. Bose(ed.), Microcomputer Control of Power Electronics and Drives. New York: IEEE Press, 1987.
3. K. J. Astrom and M. Lundh, Lund Control Program Combines Theory with Hands On Experience, IEEE Control Syst. Mag., vol. 12, no. 3, pp. 22-30, 1992.
4. Tom Hogan, C versus Assembly-C plus Assembly, BYTE, vol. 11, no. 11, pp. 267-284, 1986.

CAPITULO II

COMPONENTES BASICOS DE UN SISTEMA CON MICROPROCESADOR

Introducción

Un sistema basado en microprocesador por muy mínimo que sea requiere, además de este, de otros elementos de apoyo para poder funcionar y desempeñar alguna tarea útil. En general estos elementos adicionales son: fuente(s) de alimentación, reloj(es), compuertas lógicas, latches, buffers y memoria externa.

La configuración básica, sin embargo, puede alejarse del caso generalizado dependiendo de las particularidades de cada microprocesador; ya que mientras algunos incluyen ya varios de los elementos mencionados, otros necesitan, además, de algunos otros dispositivos especializados a fin de poder ejecutar algún programa. En éste capítulo se presentarán con algún detalle éstos elementos; inicialmente tratando de ser lo más general posible, para finalizar con la presentación de un sistema mínimo z-80, el cual se convertirá en la base de todo el trabajo posterior.

2.1 Componentes mínimos de un sistema generalizado

A pesar de todas las diferencias que puedan presentar las distintas familias de microprocesadores con que se cuenta en la actualidad, es completamente factible el poder identificar cuales son los componentes comunes a todos ellos. Estos componentes se presentan en diversas formas en cada microprocesador, pero en esencia están ahí para realizar una función específica que les es común a todos ellos. A continuación se describen los elementos que seguramente se encontrarán en cualquier sistema basado en microprocesador.

2.1.1 Fuente de alimentación.

Aunque a menudo se pasa por alto, la fuente o fuentes de alimentación constituyen una parte esencial de cualquier sistema basado en microprocesador. Del correcto funcionamiento de la fuente dependerá si el sistema será capaz de realizar el trabajo que se le encomienda o por el contrario presentará un comportamiento errático. Peor aún, una fuente defectuosa o inapropiada puede provocar daños graves a todo el sistema.

El voltaje o voltajes requeridos por un sistema de este tipo deben mantenerse normalmente dentro del 5% del valor nominal exigido. Una manera sencilla de obtener una fuente con estas características es usar circuitos integrados reguladores de voltaje, tales como la serie 78xx (voltajes positivos) y la serie 79xx (voltajes negativos).

A diferencia de su antecesor (Intel 8080), el Z-80 requiere de una sola fuente de voltaje (+5 Voltios $\pm 5\%$). Esta es una ventaja ya que fuentes de este tipo son más comunes de encontrar que una fuente apropiada para un sistema con el 8080 (+5V, -5V y +12V).

2.1.2 Reloj del sistema.

Las señales de reloj en un sistema basado en microprocesador son las formas de onda que se utilizan para sincronizar la operación del sistema. Algunos microprocesadores tienen un oscilador interno para generar la señal de reloj, con lo cual solo requieren componentes pasivos externos. Otros, en cambio, necesitan de una circuitería un poco más complicada para generar las formas de onda de reloj.

El 8080, por ejemplo, es del tipo de microprocesador que necesita dos señales de reloj generadas externamente ϕ_1 y ϕ_2 , las cuales deben alternar entre 0 y +12 Voltios y que además no deben traslaparse.

Aún cuando es posible generar estas dos señales de reloj a partir de componentes discretos (inicialmente así se hacía), una mejor alternativa es utilizar circuitos integrados especializados.

Intel introdujo el chip 8224, con lo cual se redujo el número de elementos y se simplificó el problema de la interfase de reloj, ya que de esta manera solo se tiene que conectar el cristal al 8224, el 8224 al 8080, y toda la interfase de reloj está completa.

El Z-80 por otra parte, tiene requisitos de reloj totalmente distintos al 8080. Requiere de una señal de reloj de una sola fase generada externamente y que alterne de 0 a 5 voltios, es decir una señal compatible TTL. La frecuencia de esta señal puede ir de 500 KHz hasta 6 MHz dependiendo de la versión del microprocesador.

Aunque la frecuencia mínima puede ser menor a los 500 KHz, el fabricante recomienda respetar este límite a fin de garantizar un funcionamiento confiable. Se puede trabajar a menores frecuencias únicamente para fines de prueba en las etapas de desarrollo de un sistema.

Figura 2.2 Circuito de reloj de frecuencia variable.

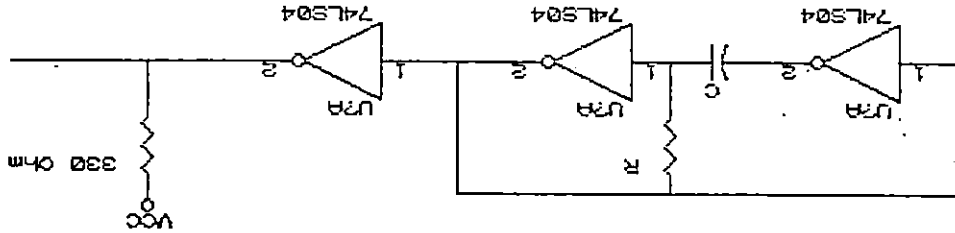
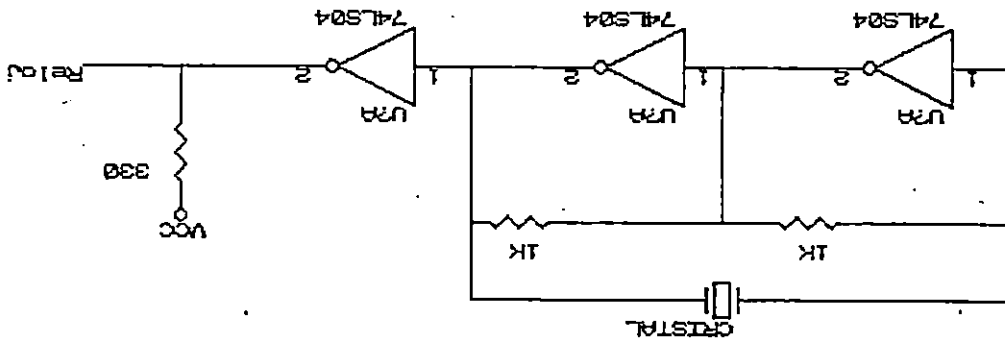


Figura 2.1 Circuito de reloj con cristal



Como puede verse, el circuito de generación de señal de reloj del Z-80 es de los más simples y menos exigentes entre los microprocesadores. Para sistemas con el Z80 que no estén corriendo a su máxima velocidad, un oscilador RC simple puede ser usado. Cuando el microprocesador opera cerca de su máxima frecuencia de trabajo, un oscilador de cristal es lo más recomendable debido a que la temporización del sistema no tolerará las irregularidades que una red RC generaría. En las figuras 2.1 y 2.2 se presentan dos alternativas de reloj para el Z-80.

El primer circuito es preferible cuando se desea mantener un tiempo de ejecución constante. Esto se hace indispensable cuando el sistema debe desempeñarse como temporizador/controlador de sucesos.

El segundo circuito tiene su mayor utilidad en las etapas de desarrollo del sistema, ya que permite modificar la velocidad del reloj; con lo cual el usuario puede ser capaz de disminuirla hasta un punto que le permita monitorear directamente el funcionamiento del microprocesador (observar los buses de datos y direcciones y las señales de control).

Lo anterior también puede lograrse cuando el reloj está a máxima velocidad, pero para ello se debe agregar circuitería adicional y tratar con las señales de control, específicamente con las señales M1 y WAIT. Sin embargo por lo general se hará de esta manera cuando ya se tiene al sistema trabajando normalmente y lo que se desea es depurar algún programa del usuario y no el hardware.

2.1.3 Compuertas lógicas y decodificadores

El microprocesador lee las instrucciones y los datos de la memoria, y usa el bus de direcciones para seleccionar las posiciones de memoria. La mayoría de los microprocesadores utilizados en los microcomputadores personales tienen 16 líneas de dirección en el bus de direcciones y, como los circuitos lógicos ordinarios, cada una de estas líneas puede estar en un 1 lógico ó en un 0 lógico. Estas líneas de dirección se llaman A0, A1, A2, etc., hasta A15.

Las compuertas lógicas se utilizan en los sistemas con microprocesadores para decodificar direcciones y para decodificar señales de control.

Aunque la familia lógica TTL es la más utilizada en sistemas con microprocesador, la familia lógica CMOS está siendo utilizada cada vez con mayor frecuencia, sobre todo en situaciones en las que el consumo de potencia es mucho más importante que la velocidad.

Además, los circuitos CMOS presentan algunas ventajas respecto a TTL: mayor margen de ruido, son más silenciosos; o sea, que a diferencia de los TTL, estos no tienden a generar ruido cuando cambian de estado.

En la actualidad se dispone de las serie CMOS 74HC y 74HCT, las cuales son equivalentes pin a pin con sus contrapartes TTL y que además tienen mejores velocidades que los de la serie CMOS standard (serie 4000).

Para mostrar como se utilizan las compuertas en los sistemas con microprocesador, considerese el caso del Z-80, el cual utiliza las líneas de control M1\ y IORQ\ para indicar que está listo para responder a una interrupción.

A fin de generar una señal alta activa de reconocimiento de interrupción (INTA), a partir de las señales disponibles en el bus de control del Z-80 puede utilizarse una compuerta NOR de dos entradas, de manera que INTA estará a nivel alto únicamente cuando tanto M1\ como IORQ\ estén en un nivel bajo.

Otro ejemplo del uso de compuertas en un sistema Z-80 se da cuando se quiere acceder algún puerto o memoria. Si se quisiera escribir en el puerto FF, las compuertas a utilizar deben combinarse de tal manera que obtengan información tanto del bus de direcciones como de las líneas de control WR\ y IORQ\. En la figura 2.3 se muestra la forma en que las compuertas lógicas pueden ser usadas para este fin.

Aún cuando el uso de compuertas cumple perfectamente el cometido de generar señales de selección o de control, a menudo en sistemas con varios dispositivos de salida o varios bloques de memoria no es muy conveniente quedarse solo con ellas, ya que de esta manera se hace excesivamente grande el número de IC's que se utilizaran únicamente para la decodificación. Como es lógico, entre más componentes se tengan en un sistema se hace más difícil su construcción y su depuración.

Una mejor manera de obtener las señales de selección de dispositivos de E/S y memoria es utilizando circuitos integrados decodificadores, los cuales pueden sustituir muchas compuertas.

A manera de ejemplo, supóngase que se requiere dividir la memoria del sistema Z-80 en bloques de 4 KBytes. Una forma de lograrlo sería generando una señal de selección para cada bloque a partir de compuertas. Para el bloque más bajo podría utilizarse el arreglo de la figura 2.4a, mientras que para el segundo bloque se tendría el arreglo de la figura 2.4b.

Si se continúa de esta forma se tendrían 16 circuitos de selección para poder direccionar adecuadamente cada bloque de 4K. Aunque en el papel los circuitos parezcan sencillos, esto se ve completamente distinto al momento de alambrear o diseñar el circuito impreso.

En cambio si se utilizara un decodificador de 4 a 16 estaríamos obteniendo todas las señales de selección a partir de un solo IC, lo cual simplifica enormemente todo el trabajo.

Figura 2.3 Ejemplo de decodificación usando compuertas lógicas

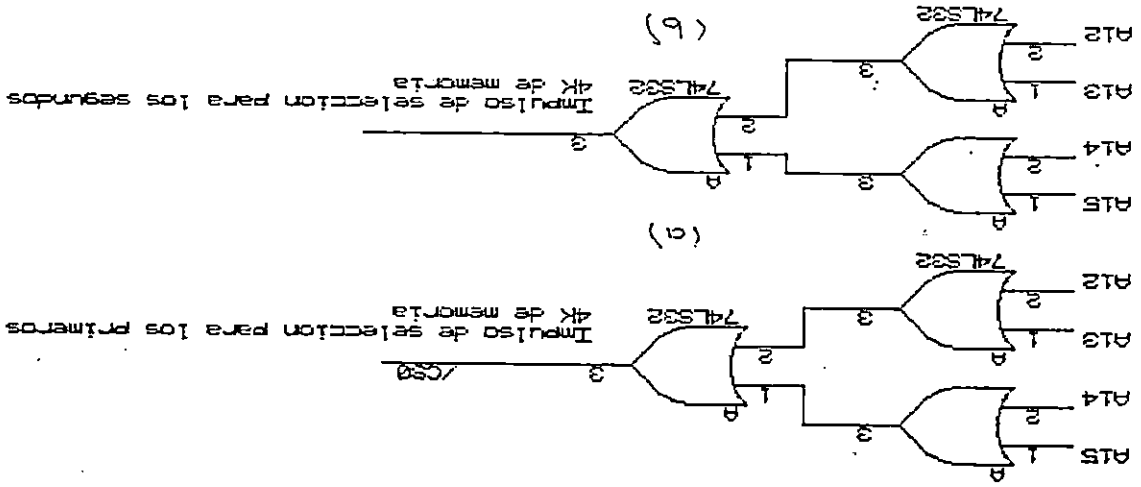
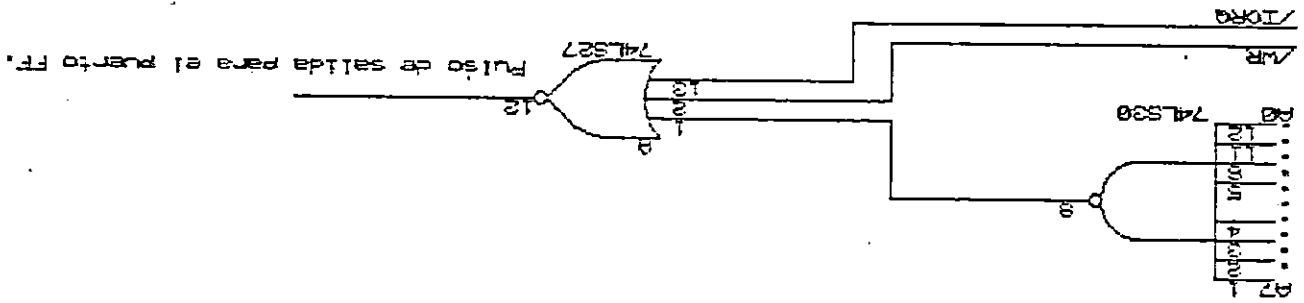


Figura 2.4 Utilización de Compuertas para seleccionar bloques de memoria de 4K.

Figura 2.8 Uso del latch 8228 como puerto de salida

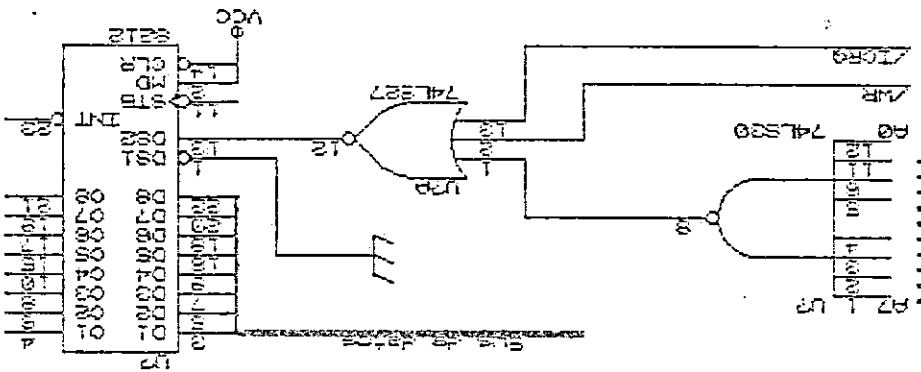
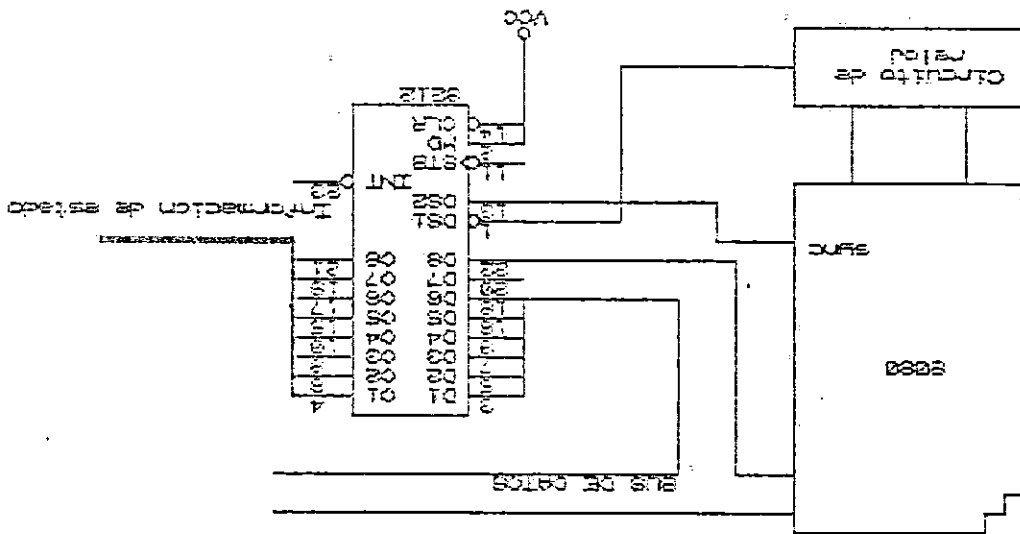


Figura 2.8 Latch 8228 utilizado para retener la información de estado del 8255.



2.1.4 Latches.

Muchas veces en los sistemas con microprocesador, la información que aparece en el bus del sistema necesita ser retenida para un posterior uso.

Algunos microprocesadores, tales como el 8080 ó el 8086/8088 necesitan imperativamente de latches ya sea para mantener información de estado (Status Latch en el 8080) o para retener la dirección en un bus de datos y direcciones multiplexado (8088).

En el caso del 8080, el latch de estado mantiene la información que el 8080 coloca en el bus de datos al comienzo de cada ciclo de máquina. Cuando el reloj ϕ y la línea de control SYNC están ambas en un 1 lógico significa que hay nueva información de estado en el bus de datos. Un latch 8212 puede utilizarse para retener esta información, tal como puede observarse en la figura 2.5.

Con el Z-80, la utilización de los latches no es tan imperativa, ya que todas las señales tienen asignado su pin respectivo, es decir que no existe la multiplexión de pines.

Los latches pueden usarse en un sistema Z-80 cuando se quiera un puerto de salida simple, de manera que la información enviada sea mantenida aún después de que haya sido colocada en el bus de datos. En la figura 2.6 se presenta un esquema de un puerto de salida utilizando un latch.

La serie de circuitos integrados 74LSxx ofrecen varios flips-flops y latches octales adecuados para usarse en puertos de salida simples en sistemas con microprocesador. Probablemente, el más útil de ellos para esta aplicación sea el flip-flop tipo D 74LS273. Los detalles de este IC están en la figura 2.7



Figura 2.7 Patillajes de los IC 74LS373 y 74LS273

En su aplicación, como puertos de salida, los terminales de datos (D0--D7) son las entradas y se conectan al bus de datos del microprocesador. Los terminales Q son las salidas y dan los niveles normales de salida TTL. Las entradas de pulso de reloj (CP) es el equivalente a la entrada de activación negativa del chip de un dispositivo puerto de entrada, y este es proporcionado por el circuito de decodificación de las líneas de dirección y control. Esta señal retiene cada salida Q al nivel lógico que hubiera en su entrada de datos en un instante particular.

Las salidas quedan retenidas (latched) en esos estados hasta que un nuevo conjunto de datos se escriban por la CPU en el puerto. Con los interfases adecuados, el puerto puede manejar lámparas, motores, relés, etc.

El terminal master reset (MR), en la patilla 1, no se suele necesitar en esta aplicación, y simplemente se conecta a la fuente de tensión positiva.

Otro dispositivo que puede ser útil es el latch octal transparente 74LS373. Cuando está en modo "transparente", sencillamente se comporta como ocho buffers, con los niveles lógicos de entrada transmitidos hacia las salidas. Cuando está en modo retención (latch), las salidas permanecen en los estados lógicos que tenían cuando se recibió la señal de retención. En la figura 2.7 se muestra el patillaje del 74LS373.

Cuando se usa como puerto de salida, la patilla 1 (OE) se conecta a la fuente de tensión negativa; el pulso de control del decodificador se conecta a la entrada de decodificación (patilla 11). Las entradas se pasan hacia las salidas cuando la patilla 11 está alta, y el dispositivo retiene cuando este terminal está bajo. Por tanto se requiere un pulso de retención (latching pulse) positivo del circuito de decodificación.

2.1.5 Buffers.

Los microprocesadores al igual que todo dispositivo electrónico, tiene especificados ciertos límites de operación, ya sea de voltajes máximos y mínimos, temperaturas extremas, disipación máxima, etc.

En cuanto a la corriente que puede manejar en cada terminal, el Z-80, por ejemplo, está especificado para absorber un máximo de 1.8mA o una carga TTL estandar en cada uno de sus pines. En caso de cargas LSTTL, el Z-80 puede manejar hasta cuatro directamente en cada pin. Este máximo especificado puede presentar de alguna manera una limitante al diseño.

Sin embargo ello no constituye un problema grave si el diseñador especifica memorias de baja potencia. Desafortunadamente el hecho de que sean dispositivos que consumen poca potencia significa generalmente que son más caros que los dispositivos estandar.

De hecho se podría comenzar a diseñar un circuito tomando en consideración cada uAmperio a fin de no sobrepasar el máximo manejable por el dispositivo en cuestión. Sin embargo si el circuito construido se encuentra en el límite de tolerancia sería muy arriesgado utilizar algún equipo de medición (sonda lógica, frecuencímetro, voltímetro, etc.), ya que la carga representada por estos sumada a la de los otros dispositivos conectados a una línea particular, sobrepasará la capacidad de esa línea.

Una manera mejor de superar la limitación de excitación de las líneas de un microprocesador es utilizar "buffers", con lo cual se puede llegar a un punto en el cual la carga ya no sea un factor que limite excesivamente al diseño del sistema.

Un buffer es un amplificador que puede utilizarse para incrementar la capacidad de manejo de corriente en una línea del microprocesador. Los buffers con salidas de tres estados también pueden utilizarse para proporcionar aislamiento eléctrico entre partes del sistema.

Un uso importante de un buffer es en un puerto de entrada. En esta aplicación, el buffer sirve para aislar los datos de entrada del bus de datos del microprocesador. Compuertas lógicas pueden ser usadas para detectar una petición de entrada de datos desde un puerto específico y generar la señal de activación de ese puerto.

En el caso del Z-80, la señal IORQ\ y la señal RD\ van a cero lógico para indicar una petición de entrada. Cuando estas dos señales están en 0 lógico, la dirección del puerto de entrada seleccionado se encuentra en los 8 bits de menor orden del bus direcciones.

Las líneas bidireccionales también pueden beneficiarse de los buffers con manejadores bidireccionales. Un manejador bidireccional puede formarse a partir de dos buffers de tres estados conectados apropiadamente a fin de asegurar que sólo uno de los buffers estará habilitado en un tiempo dado.

El 74LS244 es probablemente el que se usa mas a menudo en las aplicaciones de amplificación de buses de datos, de direcciones o de control unidireccional de 8 bits. Los ocho bits del bus del microprocesador se acoplan a las ocho entradas del 74LS244.

Hay dos entradas negativas de activación del chip (chip enable) en este dispositivo (patillas 1 y 19), con CE1 controlando los buffers 0,2,4 y 6; mientras que CE2 controla los buffers 1,3,5 y 7.

Cuando se usa para amplificación de un bus, todos los buffers de ese bus necesitarán operar al mismo tiempo, y las patillas 1 y 19 se conectarían, por tanto, juntas. Nótese que es una señal negativa la que activa los buffers, por lo que es una señal de activación baja lo que debe proporcionar el circuito decodificador al dispositivo. El detalle del patillaje del 74LS244 se presenta en la Fig. 2.8.

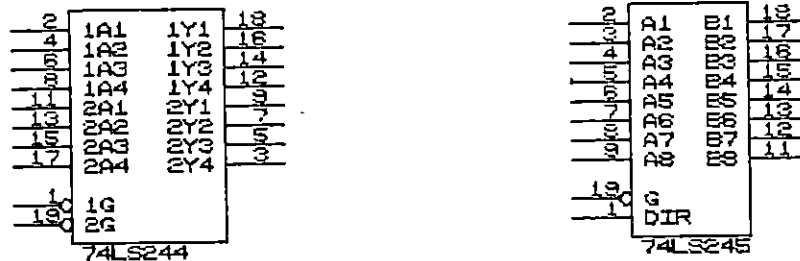


Figura 2.8 Patillaje de los IC 74LS244 y 74LS245

Una alternativa útil al 74LS244 es el transceptor octal triestado 74LS245. La figura 2.8 también presenta el patillaje de este dispositivo. Este difiere de un buffer triestado en que la señal puede pasar por el dispositivo en ambas direcciones.

Con la entrada enviar/recibir (S/R) a nivel alto, la entrada está en el lado "A", y la salida se toma del lado "B". Hay una entrada de activación negativa en la patilla 19, y sea cual fuere el lado del dispositivo puesto como salida, pasa al tercer estado lógico, a menos que se le aplique a esta la señal de activación adecuada. Como puede concluirse, un 74LS245 puede sustituir ventajosamente a 2 IC 8212 en un bus bidireccional.

En la figura 2.9 puede observarse como se usaran estos dispositivos como amplificadores de los buses de datos y dirección en un microprocesador.

2.1.6 Memoria.

Esta es quizás la parte más importante de cualquier sistema de computadora. Tanto las instrucciones de programación como los datos deben ser almacenados en el tiempo adecuado de tal forma que la computadora pueda realizar su función.

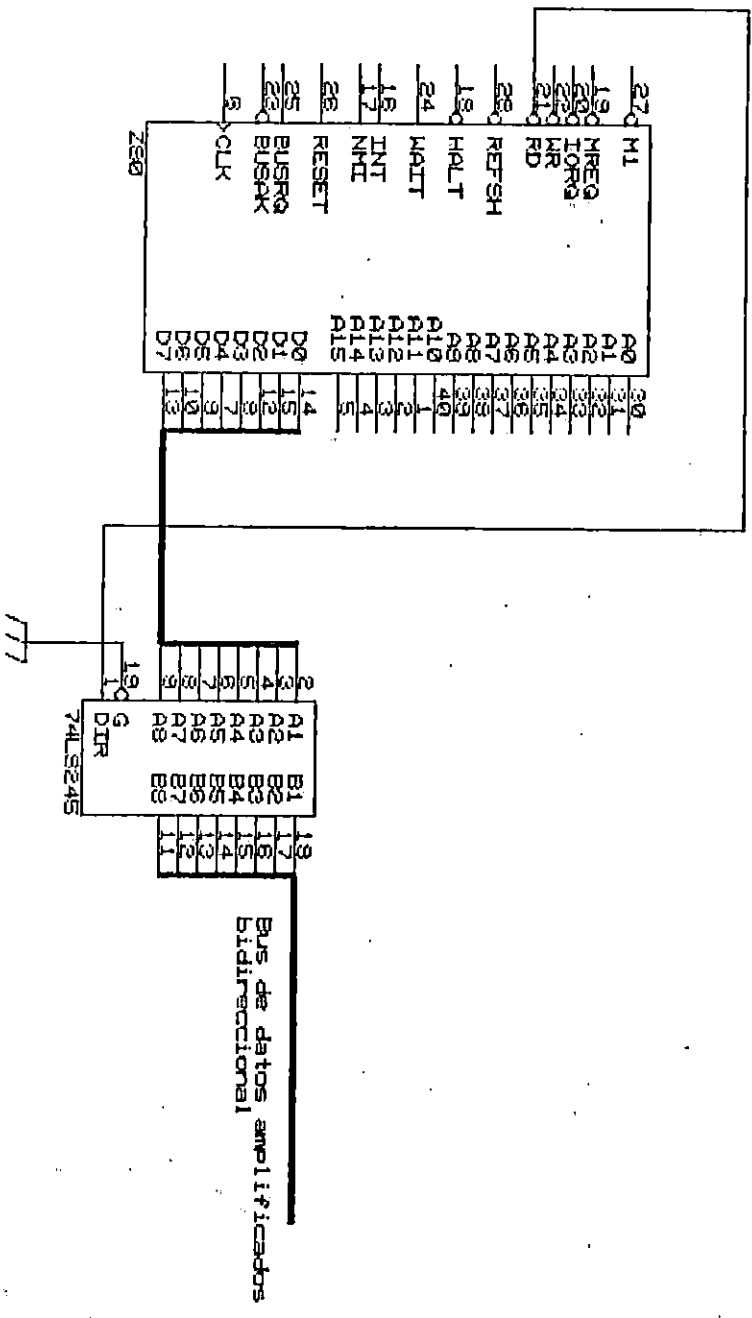


Figura 2.9 Amplificación de buses usando el 74LS245

Aún cuando los microprocesadores tienen una cierta cantidad de registros de almacenamiento, estos pueden ser utilizados solamente en la manipulación temporal de datos y no pueden almacenar instrucciones de programas.

Las instrucciones de programación deben ser almacenadas en elementos de memoria exterior, aunque en algunos casos, toda la memoria necesaria puede estar incluida en el chip del microprocesador, pero en la mayoría de los casos se requiere de memoria externa al microprocesador.

Aunque al hablar de memoria externa pueden abarcarse muchas formas de almacenamiento que pueden ir desde las memorias de ultra-alta velocidad como las utilizadas en los caches de memoria hasta las formas de almacenamiento lentas como los cassetes o las tarjetas perforadas; cuando se habla de sistemas con microprocesador por lo general se está refiriendo a formas de memoria que se encuentran ubicadas en medio de esos dos extremos de velocidad y de densidad de almacenamiento.

En el ambiente de los microprocesadores se trabaja principalmente con dos niveles de memoria; memoria de corto plazo de alta velocidad (RAM MOS, nucleo magnético, ROM MOS para almacenamiento permanente) y Memoria de almacenamiento masivo de mediano plazo (floppy disk, memoria de burbuja, memoria de tambor, CD ROM). Para un sistema minimo lo que interesa es la memoria de trabajo de corto plazo. Las características importantes de estas unidades de almacenamiento incluyen tiempos de acceso rápido y capacidad de acceso aleatorio.

Dado que las memorias de núcleo magnético se han ido rezagando, en la actualidad las memorias semiconductoras se usan casi invariablemente en todas las μ Computadoras.

Las memorias semiconductoras se dividen en dos clases principales: ROM (Memoria de solo lectura) y RWM (Memoria de lectura y escritura). La ROM se utiliza para almacenar datos específicos, pasos permanentes de programación o datos. Los contenidos de estas posiciones de memoria son considerados permanentes y no pueden cambiarse fácilmente.

La memoria de lectura/escritura, por otra parte, es utilizada para almacenar datos que cambian cuando la computadora está funcionando. Estos datos pueden ser resultados generados por un programa, o bien pueden ser parte de las instrucciones de un programa del usuario.

Aunque ambas son direccionables aleatoriamente, la costumbre ha hecho que a la memoria de lectura/escritura se le conozca como memoria RAM (Memoria de Acceso Aleatorio).

Por su parte, la RAM se divide en dos categorías: RAM estática y RAM dinámica. La RAM estática tiene la característica de que conserva la información durante el tiempo que se mantenga la alimentación al circuito; en cambio las memorias dinámicas requieren que la información esté siendo regenerada periódicamente. La compensación a este inconveniente es que debido a su estructura más simple es posible obtener mayores densidades de almacenamiento, menor disipación de potencia y mayor velocidad.

Las formas de regenerar la información en una memoria dinámica pueden ser a veces complicadas, ya que por lo general los microprocesadores no proporcionan ningún soporte especial para este tipo de memorias. El Z-80 es uno de los pocos micros que contienen la circuitería interna que facilita de alguna manera el uso de memorias dinámicas, aunque siempre requerirá de circuitería adicional para que el sistema funcione correctamente.

El uso de memoria dinámica se justifica únicamente cuando se necesite hacer uso de grandes cantidades de memoria. En sistemas pequeños, el uso de memoria dinámica constituye más un problema que una ventaja. Es por ello que estos utilizan casi invariablemente memoria estática para el almacenamiento de datos cambiantes.

2.1.7 Sistema mínimo basado en microprocesador

Como ya se dijo al inicio de este capítulo, cada microprocesador tiene ciertos requerimientos mínimos para constituir un sistema funcional completo.

Consideramos un sistema mínimo compuesto por un puerto de entrada, un puerto de salida y memoria ROM (la memoria RAM no siempre se utiliza, y de hecho existen muchos sistemas que no lo incluyen).

Para configurar un sistema mínimo a partir del 8080 se necesitaría de un chip de memoria ROM, dos chips que sirvan como puertos, un latch de estado, tres fuentes de alimentación y dos señales de reloj generadas externamente.

Un sistema basado en el Z-80 tiene menores requerimientos: un chip de memoria ROM, dos chips de puerto (si se desea tener uno de entrada y uno de salida), una fuente de alimentación y una señal de reloj generada externamente.

Ambos sistemas deben incluir también algunas compuertas lógicas para fines de selección de dispositivos.

Es evidente que un sistema alrededor del Z-80 es más sencillo y económico que uno basado en el 8080. Esta es la razón por la cual muchos de los sistemas y programas que originalmente fueron creados para el 8080; al surgir el Z-80 fueron adaptados a éste, dada la compatibilidad existente a nivel de código de máquina y con la ventaja adicional que el Z-80 incluía nuevas y poderosas instrucciones.

Un sistema mínimo basado en un microprocesador del tipo "single-chip computer", tal como el INTEL 8748 sería aún más simple y fácil de construir, ya que incluye 1 kbyte de PROM dentro del chip. Además este microprocesador incorpora dos puertos bidireccionales; con lo cual prácticamente solo hay que agregar la alimentación y el circuito de reloj para obtener un sistema funcional mínimo.

Claramente, el 8748 sería elegido indudablemente para la construcción de pequeños sistemas. Sin embargo tiene la desventaja de no poderse ampliar para sistemas mayores con suficiente facilidad comparado como sucedería con un sistema 8080 ó Z-80.

De los tres microprocesador mencionados, el Z-80 es el más conveniente, dado que la construcción de un sistema mínimo es menos compleja que con el 8080 y además, a diferencia del 8748, su expansión para formar sistemas más grandes no es muy difícil.

2.1.8 Computadora en una tarjeta

Se les conoce de esta manera a los sistemas basados en microprocesador que incluya toda la circuitería de soporte en una sola tableta de circuito impreso y que por consiguiente puedan ponerse a trabajar casi inmediatamente.

La gran ventaja de construir u obtener un sistema de este tipo es que una vez disponible, este puede servir como la base de un sistema más complejo en lugar de comenzar siempre a partir del microprocesador. De esta manera el diseñador se libera en gran medida de los detalles del diseño de hardware del microprocesador y le permite concentrarse inmediatamente en los problemas del diseño del sistema mayor.

Aunque la computadora de una tableta puede utilizarse como un sistema autónomo, es de recalcar que los buses de datos, dirección y control son llevados a conectores terminales de la tarjeta, de manera que las capacidades del sistema pueden ser expandidas enormemente agregando tarjetas con memoria adicional, tarjetas de interfase de dispositivos de E/S, etc. a medida que se vayan necesitando.

2.2 Sistema mínimo basado en el Z-80

Un sistema mínimo construido a partir del Z-80 presenta ventajas en comparación con sistemas basados en otros microprocesadores de su generación .

Aparte de requerir pocos componentes adicionales para configurar un sistema mínimo, el Z-80 permite que éste pueda ser ampliado enormemente de acuerdo a las necesidades que vayan presentando.

En esta parte se presenta el diseño de un sistema mínimo funcional, el cual constituirá el núcleo alrededor del cual se irán agregando las otras partes que complementarán el sistema final que se busca tener al concluir este trabajo de graduación.

Una ventaja al proceder de esta manera es que una vez se tenga este núcleo funcional, el mismo servirá para depurar las otras partes que se agregen mediante la utilización de algunas rutinas en software .

Debe tenerse siempre en cuenta que dado que el primer objetivo es obtener un circuito que funcione, en esta parte se privilegiarán la simplicidad y la disponibilidad de componentes sobre la elegancia o la sofisticación del diseño. Una vez se tenga algo que trabaje correctamente puede comenzarse a realizar mejoras .

2.2.1 Componentes del sistema básico inicial

Anteriormente se mencionó que un sistema Z-80 mínimo requiere de los siguientes componentes :

- CPU Z-80
- Memoria ROM
- 1 chip de puerto de entrada
- 1 chip de puerto de salida
- Una fuente de alimentación
- Una señal de reloj generada externamente

Todos estos elementos se encontrarán en el sistema que se construirá; aunque en el caso de la fuente de alimentación, ésta no se construirá sino que se aprovechará la alimentación proveniente de la computadora que trabajará en conjunto con el sistema Z-80.

A diferencia del sistema mínimo mencionado, el sistema Z-80 básico inicial incluirá algunos otros elementos destinados a facilitar y permitir su expansión posterior. Con esto se está hablando específicamente de buffers y circuitos de

decodificación. Además, en lugar de utilizar chips de puertos de entrada y de salida separados, se empleará inicialmente un chip de entrada/salida programable (PPI).

2.2.2 Reloj del sistema Z-80 mínimo

El circuito que se presentó en la figura 2.1, puede utilizarse perfectamente en nuestro caso. Lo único que faltaría por determinar en ese circuito sería la frecuencia del cristal la cual quedará determinada en su límite superior por la máxima frecuencia que pueda soportar el microprocesador o las memorias que se utilicen. Para nuestro caso, y dado que se utilizará un Z-80A, se empleará un oscilador, de 4 MHz.

A veces es conveniente disponer en las primeras etapas del desarrollo de un sistema de este tipo, de alguna forma de controlar directamente el funcionamiento del CPU mediante la disminución de la velocidad del reloj. Para conseguir lo anterior se puede utilizar inicialmente el circuito generador de reloj de la figura 2.2. Otra forma de lograr este tipo de control es mediante la utilización de un generador de señales de frecuencia variable y que sea capaz de proporcionar señales compatibles TTL.

Esta última opción es quizá la más conveniente, dado que de ésta manera se consigue un mayor control de la frecuencia y también se garantiza que la forma de la señal no distorsione excesivamente, a diferencia de lo que ocurre con el circuito de la Fig. 2.2 el cual en la práctica tiene un rango de acción bastante limitado.

Una manera un tanto diferente de controlar la velocidad de ejecución de un programa es utilizando la entrada WAIT del Z-80. La señal de reloj se mantiene invariable (por lo general a su nivel máximo) y la señal WAIT se lleva a su nivel lógico "0", con cual se suspende la ejecución del programa y el microprocesador se mantiene indefinidamente en el ciclo M1 (búsqueda de instrucción). Este estado se mantendrá hasta que la señal WAIT pase a un nivel "1" lógico.

Para poder utilizar las señales M1 y WAIT en el control del microprocesador se necesita construir un circuito adicional. Es por esta razón que por el momento su uso constituirá más una complicación que una ayuda para la obtención del sistema mínimo funcional.

En conclusión, inicialmente se tomará la señal de reloj a partir del generador de señales, y una vez se esté en operación normal se utilizará el circuito de la fig. 2.1.

El circuito de reloj es particularmente sensible, por lo cual se recomienda mantener las patillas de los elementos lo más cortas posible para evitar la creación de cargas inductivas y capacitivas que puedan inutilizar el circuito debido a ruido excesivo.

2.2.3 Circuito de reset del Z-80

Este circuito es muchas veces ignorado en los documentos que tratan acerca del diseño de sistemas con microprocesadores. Sin embargo tiene una gran importancia, la cual se manifiesta tanto al momento de energizar el sistema (función de inicialización) como al momento de la ejecución de programas (que no se estén ejecutando de la forma esperada (reinicialización)).

La función de inicialización es indispensable, ya que el microprocesador exige que el terminal RESET se mantenga en nivel bajo por lo menos 50 μ segundos después que se hayan alcanzado los +5V en los terminales de alimentación de los IC del sistema.

Dado que no existe ninguna garantía de que dos fuentes de alimentación tengan el mismo tiempo de subida, el diseñar un circuito de inicialización que solo cumpla con los requisitos mínimos es completamente insensato sobre todo teniendo en cuenta que no existe ningún tipo de limitación respecto al tamaño del pulso de reset.

Es por ello que el circuito de reset a utilizar en este sistema tendrá un pulso con una duración de 50 μ seg, lo cual es 1000 veces más largo de lo absolutamente necesario. Aunque cuando esto puede parecer excesivo visto desde la perspectiva del microprocesador, 50 μ seg es en realidad un tiempo despreciable desde la perspectiva humana.

En la figura 2.10 se muestra el circuito de reset a emplear. Este circuito incorpora tanto el reset manual como el reset automático. El reset manual lo proporciona el pushbutton, el cual al ser presionado aterrizza el pin RESET del Z-80.

La longitud del pulso de reset quedará determinada por el producto RC. Escogiendo $C=4.7\mu F$ y $R=10K$ Ohms, se tiene un pulso de alrededor de 50 μ seg. El diodo presente en el circuito funciona como una válvula de seguridad que permite la descarga rápida del capacitor al desconectar la alimentación. Los inversores con disparador Schmitt aumentan la confiabilidad del diseño y es por eso que se incluyen en el circuito.

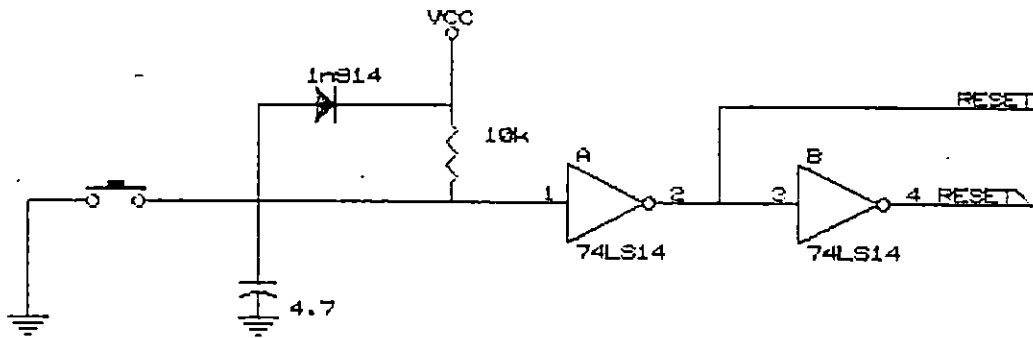


FIG. 2.10 CIRCUITO GENERADOR DE SENAL RESET DEL SISTEMA Z80

2.2.4 Amplificación de los buses del sistema mínimo Z-80

La razón de utilizar los buffers ya fué ampliamente discutida en la sección 2.1.5. Básicamente puede decirse que los buffers permiten amplificar las señales provenientes del Z-80. Además se consigue un alto grado de aislamiento entre el Z-80 y las otras partes del sistema, con lo cual se protege al microprocesador.

El resultado es que las salidas amplificadas pueden manejar 15 unidades de carga TTL o 60 cargas de baja potencia Schottky cuando están habilitadas.

La figura 2.11 muestra la configuración del bus de direcciones amplificado usando dos IC 74LS245. Las características generales de este IC ya fueron mencionadas en la sección 2.1.5. Es de aclarar la razón por la cual a pesar de que el bus de direcciones es unidireccional se esté utilizando un IC bidireccional. Esto se debe a que el 74LS245 (a diferencia del 74LS244) tiene la ventaja de que todos los pines de entrada se encuentran en un lado del IC y todos los pines de salida se encuentran en el otro lado.

Esto facilita enormemente el cableado del prototipo y lo hace más claro al momento de revisar. También facilita a la hora de realizar el circuito impreso.

El circuito de la figura 2.11 ya tiene previsión para operaciones con DMA, ya que mediante la utilización de la señal BUSAK\ del Z-80 puede hacerse que los IC 74LS245 pasen al estado de alta impedancia, con lo cual otro dispositivo puede tomar control del bus de direcciones del sistema. Las

salidas de control también pueden amplificarse mediante el uso de los buffers. Tanto el bus de direcciones como el de datos y las salidas de control pasarán al estado de alta impedancia al pasar BUSACK\ al nivel 0 lógico.

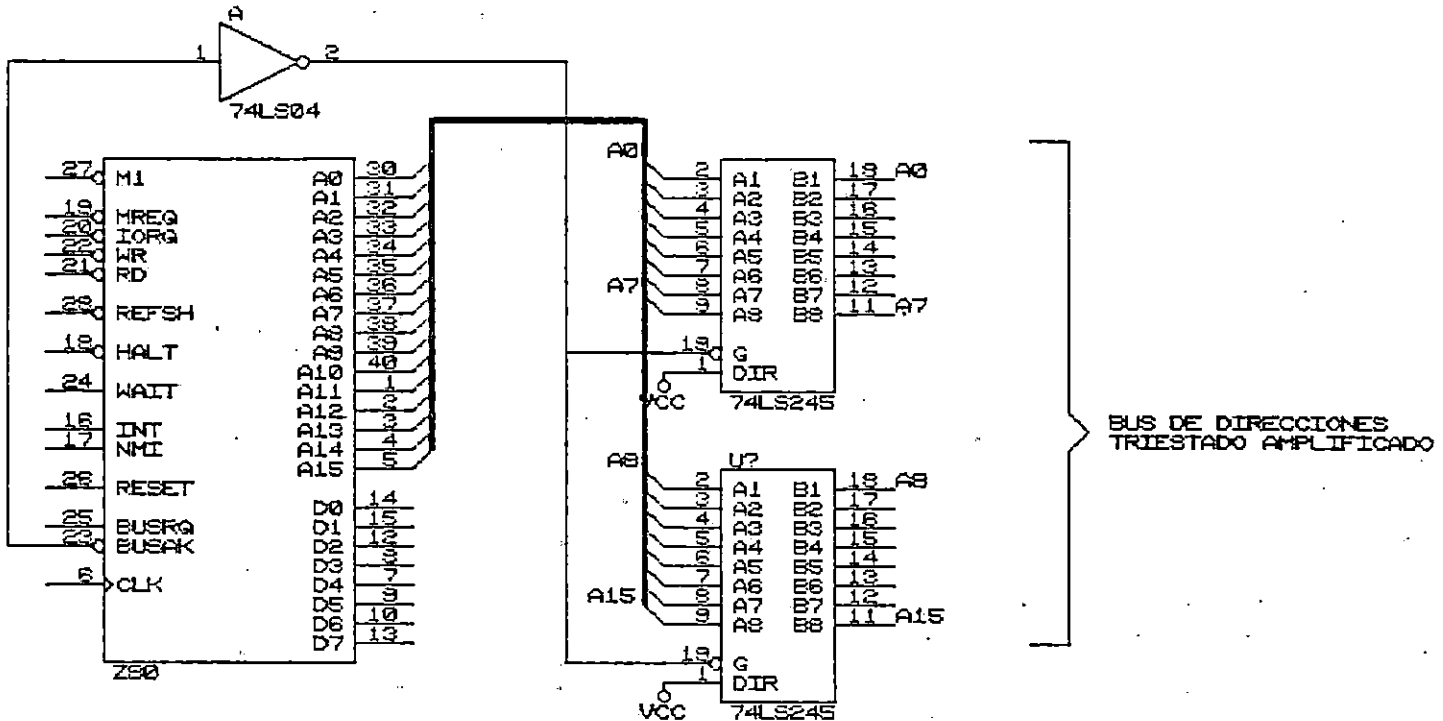


FIG. 2.11 BUS DE DIRECCIONES PROVISTO DE BUFFERS

2.2.5 Decodificación de Memoria y Puertos de E/S.

Para que el Z-80 pueda enviar o recibir datos desde un dispositivo o memoria específico es necesario que se disponga de alguna forma de seleccionar cada uno de ellos por separado mediante la generación de impulsos de selección.

Dado que el Z-80 hace una distinción entre lo que son los dispositivos de entrada y salida y lo que es la memoria; la generación de impulsos de selección por estas dos categorías es un tanto diferente.

En el Z-80 los impulsos de selección de dispositivos están asociados a la señal IOREQ\, mientras que los impulsos de selección de dirección están asociados a la señal MREQ\. Cuando se quiere generar estos impulsos de selección, se puede seguir dos estrategias: Decodificación ambigua (parcial) y Decodificación completa (absoluta).

Con la decodificación ambigua el circuito de decodificación se simplifica, ya que se utilizan menos líneas de dirección, quedando las otras como estados "no importa". Esto provoca que el dispositivo o memoria será activado por más de un estado del bus de direcciones, lo que se conoce como "imágenes" del dispositivo o memoria, lo cual no es un problema para sistemas que no van a utilizar las áreas de memoria que en un momento dado funcionan como "imágenes".

La decodificación absoluta, por otra parte, garantiza que un dispositivo o posición de memoria será accesado por una única dirección colocada en el bus de direcciones. El precio a pagar por esta decodificación completa es que se necesitarán más compuertas o demultiplexores para generar los impulsos de selección. La decodificación completa a menudo no se necesita en muchos sistemas industriales dedicados ya que por lo general estos no hacen uso de todo el mapa de direcciones.

Para el sistema Z-80 que se propone en esta primera etapa tampoco es necesario lograr una decodificación absoluta, ya que el objetivo primario es conseguir un sistema que funcione correctamente y con el menor número de componentes.

Como se observa en el circuito de la figura 2.12, la decodificación tanto para la memoria como para los dispositivos de E/S se consigue utilizando un solo IC. Este circuito está diseñado para controlar 8K de memoria (dividido en 4 bloques de 2K) y 16 puertos (divididos en bloques de 4 puertos). Estas divisiones se han hecho pensando en utilizar las EPROM 2716 y las RAM 6116, las cuales son de 2K x 8 bits. Además se ha pensado en utilizar los dispositivos de entrada/salida PPI y PIO, los cuales ocupan cada uno 4 direcciones del mapa de E/S.

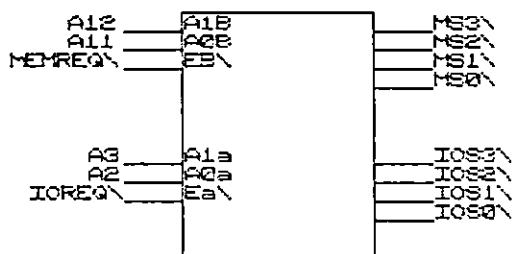
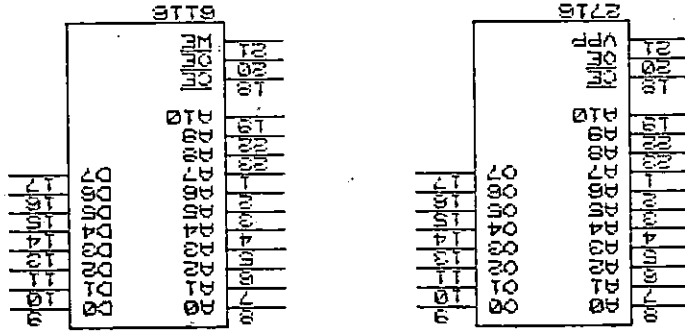


FIG. 2.12 CIRCUITO GENERADOR DE PULSOS DE SELECCIÓN PARA LA MEMORIA Y LOS PUERTOS DE E/S

La razón por la cual se utiliza una EPROM de 2K (2716) en lugar de una de 1K (2708) es que esta última tiene la desventaja de requerir de tres fuentes de alimentación (+5, -5 y +12 V) y el sistema a construir solo dispondrá de una

FIG. 2.13 PARTILAJE DE LA ROM Y LA RAM A UTILIZAR



Por el momento, el sistema dispondrá de 2K de memoria ROM mediante el uso de una EPROM 2716 y, aunque no es tan necesario por ahora, 2K de RAM proporcionados por una RAM estática 6116. En la figura 2.13 se ve el patillaje de las memorias a utilizar en el sistema.

La mayoría de sistemas pequeños como el que se quiere construir no necesitan gigantes memorias para poder desempeñar su trabajo satisfactoriamente, aunque esto no significa que no sea posible aumentar considerablemente la memoria en caso de que surja la necesidad en un futuro.

Sin importar que tan complejo o sencillo sea el sistema que se está desarrollando, éste es un elemento indispensable. La razón de ello es que un microprocesador sencillamente no hará nada útil si carece de un lugar donde almacenar datos y programas.

2.2.6 Memoria del sistema Z-80 mínimo.

Estas señales (READ y WR) se utilizan mas tarde para determinar si lo que se hace es una operación de lectura o de escritura, pero esto se realiza ya en el dispositivo seleccionado y no en el circuito de selección.

Obsérvese que la selección de los dispositivos de E/S y de la memoria no se ve influenciada en este circuito por el tipo de memoria (RAM o ROM) o por el tipo de puerto (entrada o salida), ya que las señales READ y WR no intervienen en esta etapa de la decodificación.

fuente de alimentación. Una desventaja aun más importante es que la programadora de EPROMS con que se dispone no acepta la 2708.

Las conexiones de la EPROM y la RAM al sistema son bastante directas, ya que el circuito de decodificación fué pensado para estas memorias en particular. La EPROM se colocará de manera que ocupe el primer bloque de 2K, mientras que la RAM estará en el cuarto bloque de 2 K del mapa de memoria.

En la figura 2.14 se presenta la forma en que quedan conectadas estas memorias al sistema. Los pines A0 a A10 de la EPROM 2716 y de la RAM 6116 se conectan a las líneas respectivas del bus de direcciones. Los pines D0 a D7 de la EPROM e, I/O0 a I/O7 de la RAM se conectan a las líneas respectivas del bus de datos del sistema.

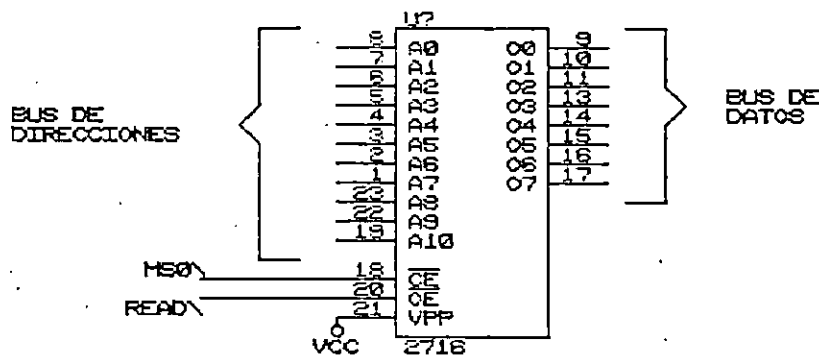


FIG. 2.14 CONEXION DE LA ROM 2716 AL SISTEMA Z80

Los pines 20 de ambos IC se conectan a la línea READ del bus de control, mientras que los pines 21 se conectan a la línea WRITE del bus de control en el caso del 6116, mientras que en el caso del 2716 se conecta a +5 V. El pin # 18 del 2716 se conecta a la línea MS del circuito de decodificación, mientras que el pin respectivo del 6116 se conecta a la línea MS del circuito de decodificación.

3

2.2.7 Puertos de E/S del sistema Z-80 mínimo

Un puerto de salida puede conseguirse con un latch; y un puerto de entrada puede ser tan simple como un buffer triestado. Al hacer las cosas de esta manera puede complicar bastante la lógica de direccionamiento y también hacer al puerto muy dependiente de la temporización.

Además, esto significaría que un puerto de salida siempre sería un puerto de salida, mientras que un puerto de entrada sería siempre un puerto de entrada. Aunque esto no parezca un problema, lo que sucede es que se pierde flexibilidad en el sistema. Lo más recomendable es construir los puertos utilizando dispositivos especializados tales como el PPI, PIO, PIA, etc.

El PPI y el PIO son idóneos para un sistema basado en el Z-80, ya que el primero (PPI) fue diseñado para trabajar con el microprocesador 8080, el cual tiene muchas cosas en común con el Z-80. El PIO fue creado especialmente para trabajar con el Z-80.

En general, el PIO y el PPI presentan las mismas características, ya que ambos poseen puertos paralelos de 8 bits. El PPI, sin embargo, por tener 3 puertos, carece de algunas de las características del PIO, con sus dos puertos.

En esta etapa del diseño, las características adicionales del PIO no son de mucha importancia. Además, el PPI se encuentra más disponible y es más barato que el PIO. El PPI (Programmable Peripheral Interface) es un IC de 40 pines y posee cuatro registros. Los tres primeros registros pueden ser individualmente programados para ser casi cualquier tipo de puerto de E/S. El cuarto puerto es el que hace que este dispositivo sea tan flexible y el candidato ideal para la E/S en un sistema basado en microprocesador. En la figura 2.15 se muestra el patillaje del PPI 8255.

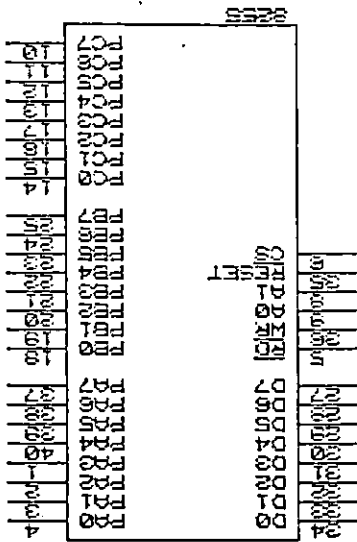


FIG. 2.15 PATILLAJE DEL PPI 8255

Al escribir distintos valores en el puerto de comando (puerto cuatro), se pueden definir las funciones de los otros tres puertos. Esto incluye configurarlos ya sea como entradas o como salidas para transferencia simple de datos, ó como puertos bidireccionales para transferencia de datos con protocolo. Dado que las características de los puertos dependen de la palabra de control cargada en el registro de comando, el 8255 puede ser dinámicamente configurado por software. Con esto, el funcionamiento del 8255 dentro del sistema puede cambiarse en cualquier momento, con solo escribir en el registro de comando la palabra que define la configuración que se desea.

El 8255 tiene tres modos básicos de operación y en cada uno de estos modos existen muchas opciones que determinan las características para cada uno de los puertos, pero dado que por ahora lo que interesa es tener puertos de E/S simples, no se continuarán detallando las posibilidades de este IC. Se utilizará al 8255 en su forma más sencilla, es decir como puertos de E/S para transferencia simple de datos.

El 8255 tiene cuatro puertos, así que el primer paso es decidir en qué direcciones del mapa de E/S se ubicarán estos puertos. Dado que no existe ninguna ventaja o desventaja, por el momento ocuparán los puertos #0 al #3, siendo los tres primeros los puertos programables de propósito general; mientras que el puerto #3 corresponderá al registro de comando.

Del circuito de selección de memoria y dispositivos, se recordará que la sección de dispositivos fué diseñada pensando en dispositivos como el 8255 (con cuatro puertos), de manera que al igual que la memoria, la conexión del PPI al sistema es bastante directa.

Las patillas RD y WR del PPI corresponden a las señales RD y WR del bus de control del sistema; mientras que la patilla CS del PPI se conectará a la señal IOS del circuito de

decodificación. Los pines A1 y A0 del PPI se conectan a las líneas A1 y A0 del bus de direcciones del sistema; mientras que las patillas D0 a D7 del PPI se conectan a las líneas respectivas del bus de datos del sistema. El pin RESET del PPI se conecta a la línea RESET del sistema.

Suponiendo que configuremos al PPI con el puerto A como entrada y los puertos B y C como salidas, se podría tener el arreglo mostrado en la figura 2.16.

El sistema mínimo inicial está totalmente detallado en el ANEXO A1. Será bastante útil para aquellos que se deseen crear su propio sistema.

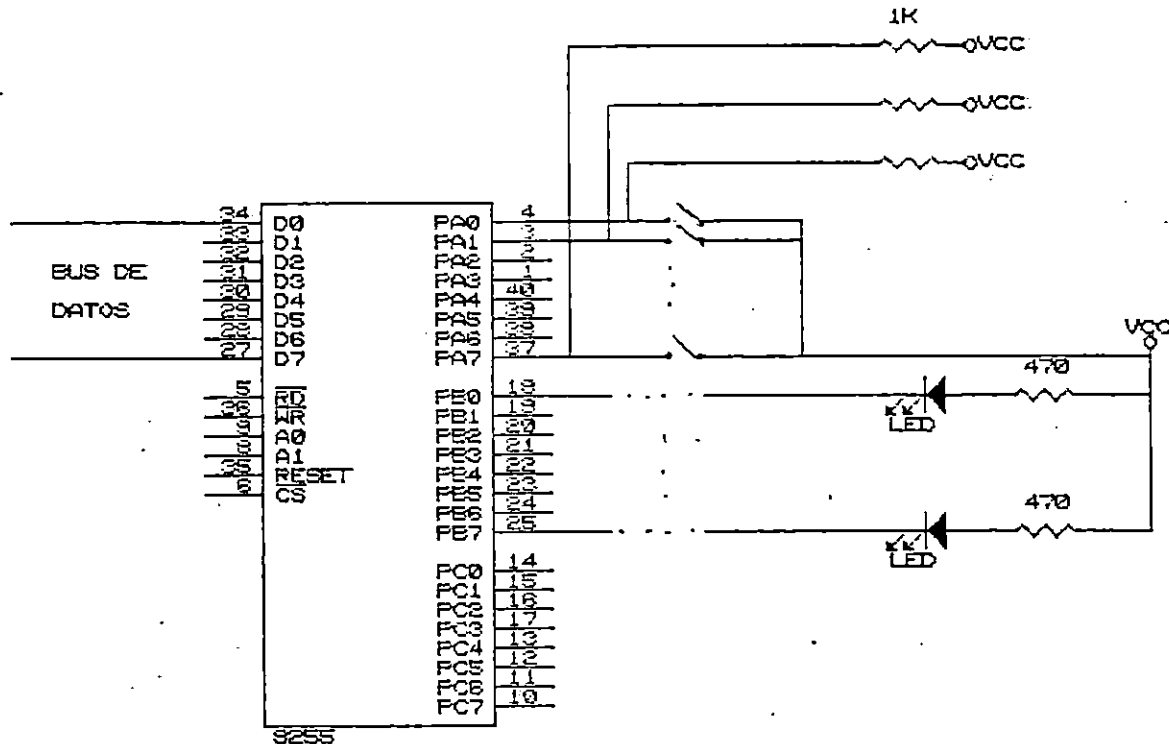


FIG 2.16 PPI 8255 UTILIZANDO EL PUERTO A Y B
A ES EL PUERTO DE ENTRADA Y B ES EL PUERTO DE SALIDA

2.3 Conclusiones y recomendaciones

1. Definitivamente en sistemas basados en microprocesadores la CPU Z80 es de los mas faciles de utilizar, ya que no se requiere de muchos dispositivos perifericos para lograr obtener un sistema funcionando.
2. El costo de los elementos de soporte para el Z80 es de los mas baratos existentes en el mercado y la versatilidad que adquiere un sistema basado en el Z80 es bastante grande.
3. La Escuela de Ingenieria Electrica cuenta con gran variedad de circuitos integrados, para diversas aplicaciones con microprocesadores. Se recomienda la implementacion de trabajos que involucren la aplicacion de estos ICs, puesto que es la unica forma de valorar su verdadero potencial.
4. El seguir un proceso de diseño bien definido es bastante saludable ya que permite encontrar el camino hacia el objetivo que se persigue de forma ordenada.

Referencias Bibliográficas

1. A. A. Bruce, Microcomputer Interfacing, New Jersey: Prentice Hall, 1980.
2. Grossblatt, R. The 8088 Project Book, Blue Ridge Summit, Pa., TAB BOOKS, 1989.
3. G. Harry, Introduction to Microprocessor System Design, Stanford: Stanford University, 1978.
4. Intel Corporation, Microprocessor and Peripheral Handbook 8086, 8088, 80286 & 80386, Santa Clara, CA.: Intel Corporation, 1987.
5. L. Austin, Z. Rodnay. Microprocessor Interfacing Techniques, U.S.A: SYBEX, SECOND EDITION, 1977
6. R. A. Penfold, Técnicas y proyectos de interfases, Madrid, España: EDICIONES ANAYA MULTIMEDIA, 1986.

CAPITULO III

DISEÑO Y CONSTRUCCION DEL SISTEMA DIDACTICO MINIMO Z-80

Introducción

Aún cuando en el capítulo II se llegó a obtener un sistema mínimo funcional, éste se queda muy limitado para los propósitos didácticos que han sido planteados el presente trabajo de graduación, ya que un sistema de ese tipo debe cumplir con algunos requisitos particulares, tales como la capacidad de poder cargar y ejecutar un programa en memoria RAM, poder monitorear el estado en que se encuentra un programa que se esté ejecutando (ya sea mediante un display propio o mediante la conexión a una videoterminal), etc.

Para poder cumplir con tales requisitos, el sistema mínimo funcional debe sufrir cambios y ampliaciones. Estas modificaciones permitirán obtener un sistema bastante flexible, de manera que pueda servir tanto para el desarrollo de programas en lenguaje ensamblador Z-80 como para el desarrollo de otros proyectos de hardware. En ésta etapa del diseño, los parámetros determinantes han sido la simplicidad del diseño y la flexibilidad del sistema resultante.

A fin de facilitar la construcción y depuración del sistema, éste ha sido diseñado en forma de módulos. El módulo principal contiene los elementos mínimos descritos en el capítulo II; es decir : CPU, memoria, decodificación no absoluta de memoria, reset, reloj y buffers. Además, en éste modulo se incluyen algunas modificaciones que contribuyen a darle la flexibilidad deseable en un sistema didáctico.

Al finalizar éste capítulo se tendrá completamente especificado todo lo que tiene que ver con el hardware del sistema.

3.1 Modulo Principal del Sistema

En éste módulo principal del sistema se localizan los elementos indispensables para el funcionamiento del sistema, de tal manera que por sí solo puede perfectamente ejecutar un programa, ya sea en RAM ó en ROM, toda vez que se haya colocado previamente en esas memorias. En el caso de la ROM no parece haber ningún problema, ya que se supone que ésta debe haber sido programada previamente en una programadora de EPROM y luego llevada al sistema didáctico Z-80. El caso de la RAM es un tanto diferente, ya que cada vez que se

desenergize el sistema se perderá lo que se haya almacenado en ella. Aparéentemente para poder trabajar en la RAM no es suficiente tener únicamente el módulo principal del sistema, pues primero ésta debe tener un programa almacenado, y usualmente los programas en RAM son introducidos por el usuario una vez el sistema está energizado y funcionando correctamente. Sin embargo aún sin la interacción con el usuario, es posible utilizar la RAM situada en el módulo principal. Esto es posible de dos formas:

1.- Que el programa en ROM utilice la RAM como un área de almacenamiento de datos y resultados de sus cálculos.

2.- Que el programa en ROM cargue la RAM con todo o parte de su contenido, de manera que en un momento dado únicamente la RAM se encuentre controlando el sistema. Algunos sistemas comerciales hacen algo parecido debido a que las memorias RAM son usualmente más rápidas que las memorias ROM.

Con lo anterior se quiere hacer ver que aunque el módulo principal contiene todo lo necesario para funcionar correctamente, un sistema que no interactúe con su entorno no tiene ninguna utilidad práctica, por lo cual para realizar algo concreto con él debemos acoplarle los otros módulos.

3.1.1 Componentes del módulo principal del sistema didáctico

En el módulo principal pueden identificarse varios subsistemas constitutivos. Algunos de estos subsistemas son idénticos a los descritos en el capítulo II cuando se presentó el sistema mínimo Z-80. Concretamente el subsistema de reset y de amplificación de buses han permanecido invariantes, por lo cual no se hace necesario volver a describirlos. En cambio otros subsistemas si han evolucionado, principalmente en aras de la flexibilidad y de permitir posibles expansiones al sistema.

El subsistema de memoria se ha ampliado para acomodar 4K de memoria ROM y 4K de memoria RAM. En memoria ROM se colocará el programa monitor y algunas rutinas de uso general, a semejanza de las rutinas del BIOS de las PC. También se ha pensado que en el futuro en la ROM se puede localizar un programa ensamblador ó posiblemente un intérprete de BASIC o de un lenguaje propio diseñado para el sistema didáctico Z80.

El subsistema de decodificación de memoria también se ha modificado, de manera que ahora se tiene una decodificación absoluta. De esta forma no se interfiere con otras posiciones de memoria que puedan utilizarse en futuras ampliaciones.

La decodificación de entrada/salida no se ha incluido en el módulo principal, ya que en éste no se tiene ningún dispositivo de E/S. Esta decodificación se hará en el módulo de entrada/salida.

Un componente que no se encontraba en el sistema mínimo funcional es el que ha sido denominado "cambio RAM/ROM". Este circuito le brinda una enorme flexibilidad al sistema cuando se esté trabajando con interrupciones, ya que permite que los parámetros que controlan el manejo de las interrupciones se encuentren en memoria RAM y no en memoria ROM como sería lo usual. De ésta manera el usuario puede determinar completamente el comportamiento de las interrupciones y no verse sometido a la forma en que el diseñador del sistema lo haya creído conveniente al momento de escribir el programa monitor.

Por último tenemos el subsistema de generación de señal de reloj. En éste circuito se han dado dos cambios importantes. Uno de ellos ha simplificado la circuitería asociada, mientras que el otro ha contribuido a la flexibilidad del sistema, al permitir cambiar la velocidad del sistema de una manera precisa y sencilla sin complicar considerablemente la circuitería.

1.- Subsistema de memoria del módulo principal

La memoria incluida en el módulo principal tiene un tamaño total de 8 Kbytes, distribuidos en 4K de ROM y 4K de RAM de la siguiente manera:

Tabla 3.1 Mapa de Memoria del módulo principal

<u>MEMORIA</u>	<u>INICIO</u>	<u>FINAL</u>	<u>FUNCION</u>
RAM1	0000	07FF	DATOS DEL SISTEMA Y PROGRAMAS DEL USUARIO.
ROM1	0800	0FFF	PROGRAMA MONITOR.
ROM2	1000	17FF	EXPANSION FUTURA.
RAM2	1800	1FFF	PROGRAMAS DEL USUARIO.

Es de hacer notar una característica importante del mapa de memoria que se muestra en la Tabla 3.1, y es el hecho de que la posición 0000h se encuentra mapeada en memoria RAM y no en ROM como normalmente debería ser en un sistema Z-80.

Para los que están familiarizados con el Z-80 les resultará un poco extraño que se haya obrado de ésta manera, ya que cada vez que se arranca ó se le da un reset al Z-80, éste espera encontrar su primer código de instrucción en la posición de memoria 0000h, la cual obligatoriamente debe ser memoria no volátil, pues de lo contrario el Z-80 encontraría un código distinto cada vez que se diera la reinicialización y empezaría a ejecutar instrucciones al azar, con lo cual se perdería por completo el control sobre el sistema.

Sin embargo, a pesar de que la posición 0000h se encuentra en memoria RAM, el sistema no se descontrola ya que se ha utilizado un artificio para "engañar" al micro. Con esto se quiere decir que desde la perspectiva del micro al darse el reset se accesa la posición 0000h, pero en la realidad se hace que se accese la posición 0800h, la cual sí se encuentra en memoria ROM y es el inicio del programa monitor.

La forma en que se logra esto quedará más clara cuando se examine el circuito de "cambio RAM/ROM". Por el momento quizá lo importante sea aclarar el porqué de ésta complicación.

El Z-80 posee un manejo de interrupciones bastante versátil, ya que puede trabajar en 3 modos en lo que respecta a interrupciones por hardware. Además posee una línea de interrupción no-mascarable, la cual al darse obliga al micro a saltar a la posición 66h. También posee interrupciones por software, las cuales tienen su dirección de inicio en la parte baja del mapa de memoria.

Como puede verse, algunas posiciones de memoria bajas son usadas por los distintos modos de interrupción del Z-80. Si el programa monitor se hubiese colocado en la parte más baja de la memoria, esto implicaría que de alguna manera el manejo de las interrupciones se haría un tanto rígido para el usuario, dado que posiciones claves para las interrupciones estarían en ROM.

Si en cambio, las posiciones bajas de memoria son RAM, esto le permite al usuario que él decida cómo quiere manejar las interrupciones según las posibilidades que le brinda el Z-80.

2.- Circuito de cambio RAM/ROM

Este circuito ha sido denominado de esta manera debido a que permite que en el momento de energizar o de reinicializar el sistema, la búsqueda del primer código de operación se haga en ROM y no en la RAM como lo determina el mapa de memoria del módulo principal. En la figura 3.1 se muestra un esquema de este circuito.

Al darse un reset o una energización, los flip-flops ponen sus salidas Q en estado alto y por consiguiente sus salidas Q en estado bajo. Esto hace que en CSRAM1\ se tenga un nivel alto a pesar que CS1\ se encuentre en estado bajo (pulso de selección), con lo cual se evita que la RAM1 sea seleccionada en el arranque.

Por otra parte CSROM1\ se pone en estado bajo a pesar que CS2\ se encuentra en estado alto (no hay pulso de selección en CS2\), lo que resulta en que la ROM1 sea la memoria seleccionada al momento del Reset. Cuando el microprocesador va a buscar su primer código de operación, o sea cuando se da el primer pulso de la señal M1\, el estado del segundo flip-flop no se ve afectado, con lo cual efectivamente el micro toma ese primer opcode de la ROM1.

Sin embargo cuando se de la búsqueda de la segunda instrucción, el mapa de memoria volverá a concordar con el mapa de memoria visto desde la perspectiva del micro, tal como si no existiera el circuito de cambio RAM/ROM. Esto significa que para que el cambio RAM/ROM funcione tal como se espera, la primera instrucción que debe leer el micro es un salto a una posición en memoria ROM. Para el caso del programa monitor del sistema didáctico Z-80, la secuencia de instrucciones inicial es:

```

0800 C3 03 08          JP      ORIGIN
0803 21 FC 03          LD      HL,3FCH
0806 F9              LD      SP,HL

```

Las líneas de selección CS1\ y CS2\ provienen del circuito de decodificación absoluta de memoria, el cual consta de un 74LS138 y dos compuertas OR 74LS32. Con este circuito de decodificación se seleccionan áreas de memoria de 2K Bytes cada una. Un esquema del mismo puede verse en la figura 3.2.

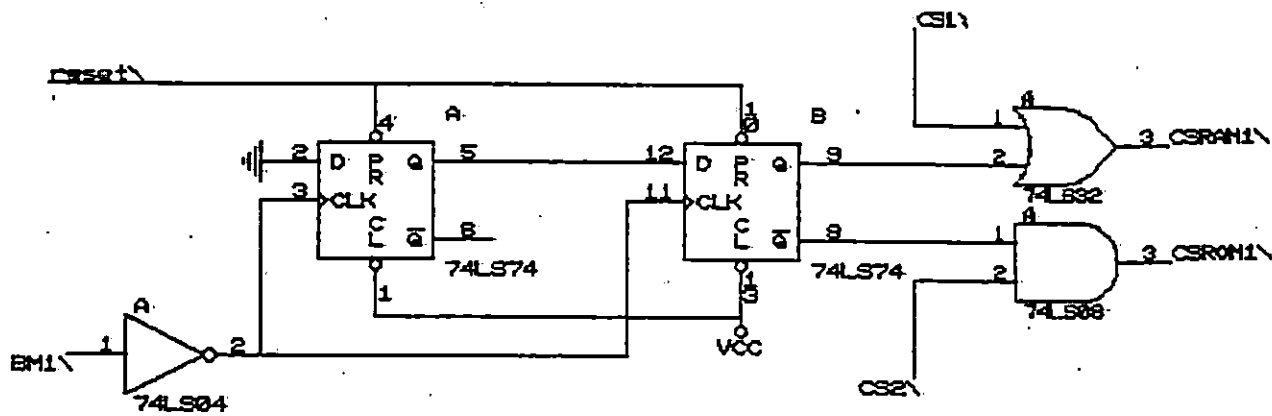


FIG. 3.1 DECODIFICADOR RAM/ROM

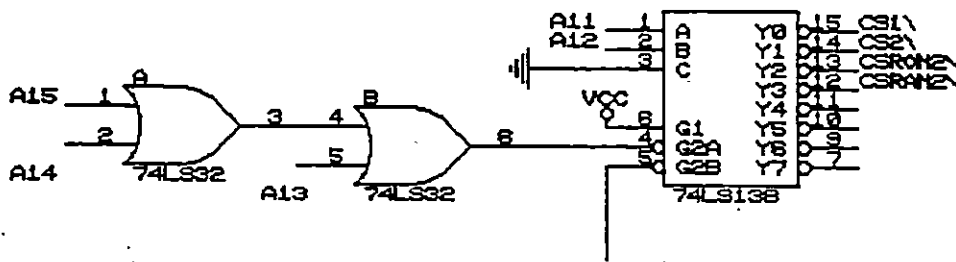


FIG. 3.2 DECODIFICADOR DE MEMORIA

3.- Circuito generador de señal de reloj.

El circuito de reloj del sistema Z-80 didáctico ha tenido algunos cambios en relación al del sistema mínimo funcional, ya que ahora se utiliza un oscilador de 8Mhz, en conjunto con un contador 74293, el cual sirve para dividir la frecuencia de 8Mhz entre 2, entre 4, entre 8 y entre 16. De esta manera es posible hacer que el sistema corra a 4 distintas frecuencias, lo cual puede ser muy útil para interfazarlo con dispositivos de distintas velocidades. Para cambiar la velocidad de operación únicamente hay que cambiar un jumper a una de las cuatro posiciones posibles. Un esquema del circuito de reloj se presenta en la figura 3.3.

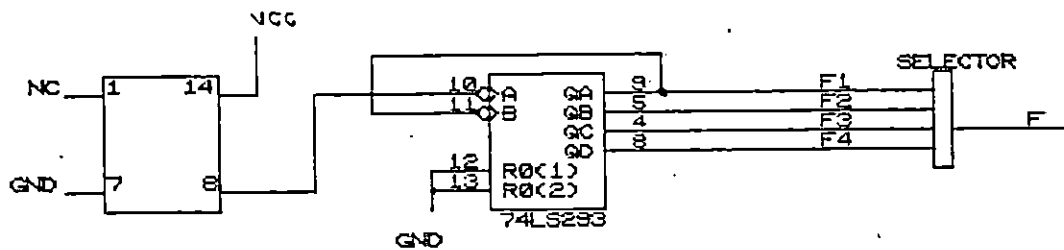


FIG. 3.3 GENERADOR DE RELOJ

4.- Conector de expansión del módulo principal.

Como se dijo al principio de este capítulo, el módulo principal difícilmente pueda tener una utilidad práctica por sí sólo. Debido a ello es necesario proporcionar el medio por el cual pueda conectarse a los otros módulos de manera que en conjunto puedan desempeñar la tarea para la cual ha sido concebido el sistema total. Es por esto que en el módulo principal se ha dejado un conector de 40 pines, el cual es un reflejo de los pines del Z-80. Mediante este conector se ha hecho la conexión al módulo de entrada/salida; y es de esta forma como se ha logrado cumplir en buena parte los objetivos generales del trabajo. En las secciones 3.3.4 y 3.3.5 se define cada línea del conector y servirá por tanto de explicación de los pines del Z80.

5.- Circuito de generación de interrupción no mascarable.

Aunque este circuito no se incluyó en el módulo principal debido a que la densidad del circuito impreso no lo permitió, lo ideal hubiera sido que se incluyera dentro de éste.

La generación de la NMI es básica para poder implementar la ejecución paso a paso y los break points, por lo cual es indispensable en el sistema didáctico.

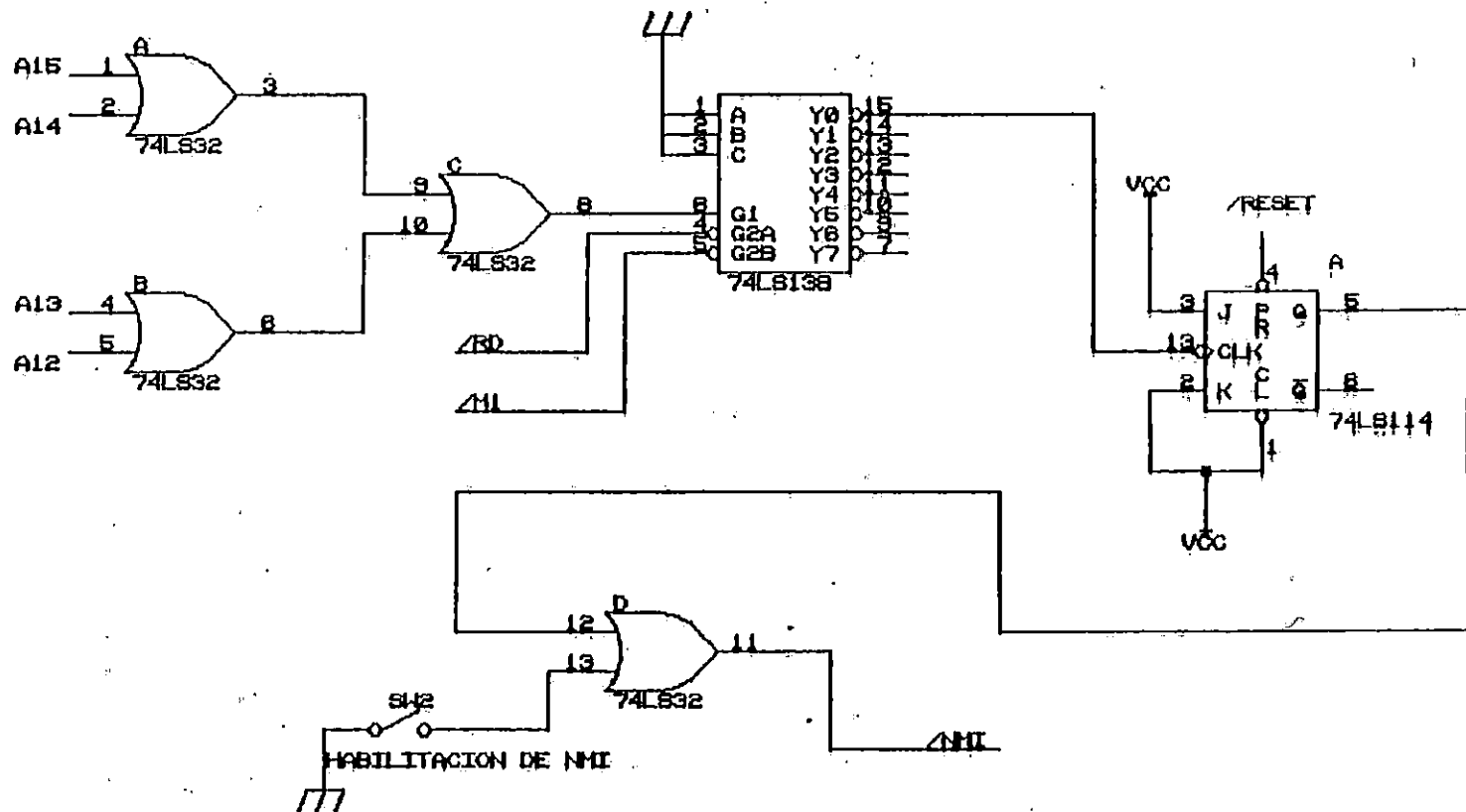


Fig. 3.4 Circuito generador de NMI

TRABAJO DE GRADUACION		
DISEÑO Y CONSTRUCCION DE UN SISTEMA DIDACTICO MINIMO BASADO EN EL Z-80.		
Title		
CIRCUITO DE GENERACION DE NMI.		
Size	Document Number	REV
A	FROZ80-p00.1	001
Date:	June 17, 1993	Sheet 1 of 1

Un esquema del circuito de generación de NMI se muestra en la figura 3.4.

Este circuito junto al circuito de cambio RAM/ROM son los más interesantes del sistema, ya que interactúan con las señales de control del micro de una manera que a primera vista puede no ser muy aparente. Para poder llegar a él se tuvieron que analizar detenidamente los ciclos de máquina de la operación involucrada. En el caso de la generación de la NMI, el ciclo de atención a una interrupción no mascarable presenta ciertas peculiaridades que debieron tenerse muy en cuenta pues de lo contrario podría no generarse la interrupción, o por el contrario se podría caer en una infinidad de interrupciones anidadas que entre otras cosas puede conducir a un rebalse del stack. La principal dificultad encontrada consistió en definir cuándo y como inhabilitar la generación de la NMI una vez que se estaba en el modo de trazado (paso a paso).

La interrupción funciona de la siguiente manera:

Para comenzar, el programa que va a ser trazado debe estar cargado en la memoria disponible exclusivamente al usuario (arriba de la ROM). Esto implica que alguno de los 4 bits más altos de la dirección no son ceros, de tal manera que a la salida de la compuerta 7432 se tiene un nivel alto. Además al darse el ciclo de búsqueda de código de operación, las señales $M1\backslash$ y $RD\backslash$ pasan al estado bajo. Estas tres condiciones habilitan al IC 74138, el cual a su vez envía un pulso negativo al flip-flop JK 74114 que se encuentra conectado como un toggle switch. Una vez que el micro detecta que se ha dado una petición de NMI, inicia un ciclo de atención a la NMI, el cual comienza generando de nuevo las señales correspondientes a un ciclo de búsqueda de opcode pero sin leer el bus de datos. Esta simulación de ciclo de búsqueda de opcode es la que ha sido aprovechada para desarmar el dispositivo de generación de NMI, pues de lo contrario el flip-flop 74114 se quedaría clavado en estado bajo y por lo tanto se generarían muchísimas interrupciones, perdiéndose todo control sobre el sistema.

3.2 Decodificador de direcciones para entrada/salida

Para utilizar un dispositivo relacionado con microprocesador es necesario asignarle una dirección de entrada ó salida para evitar que existan dos dispositivos al mismo tiempo quieran compartir los recursos que les proporciona el microprocesador, por tanto hay que hacer una planificación respecto del direccionamiento de los dispositivos que se emplearán.

En ese diseño el objetivo era capturar una misma clase de datos por lo cual se empleó el modo más sencillo de operación del 8255 y efectivamente era el más idóneo.

Debido a que, se debía utilizar el puerto de salida paralelo construido en otro trabajo de graduación de la Escuela de Ingeniería Eléctrica, se pensó en utilizar igualmente la interfase programable empleada en ese diseño. El 8255 de Intel.

El problema que se plantea es como la CPU 280 podría coherentemente recibir la información que sea proporcionada por el puerto de entrada/salida desde la PC ó una compatible y efectuar las acciones que sean definidas en función de esa información ?

El objetivo de esta sección es el diseño y construcción de una interfase que sea capaz de interpretar la información que se proporciona por la AT LEMON 286S.

3.3 Diseño y construcción de la interfase de comunicación en la CPU 280

Los circuitos empleados son el 74LS138 que es un decodificador de tres a ocho y el 74LS139 que son dos decodificadores, integrados en el chip, de dos a cuatro.

El decodificador de direcciones aparece en la Fig. 3.5, donde se establecen claramente las direcciones mencionadas anteriormente para acceder cada dispositivo de entrada y salida.

Por otra parte se activan en estas operaciones las señales IORÉQ, que especifica claramente que la operación que se efectúa es de entrada o salida externa. La señal RD\ define si se leerá del dispositivo y la señal WR\ define si se escribirá al dispositivo.

Dado que para entrada/salida el microprocesador puede direccionar hasta 256 localidades este usa los ocho bits más bajos del bus de direcciones.

- Interfase de comunicación (00-03)hex
 - Desplegadores de información (04-06)hex
 - 280 PIO (08-0B)hex
 - 280 CTC (0C-0F)hex

El mapa de direcciones de entrada/salida que se planteó es el siguiente:

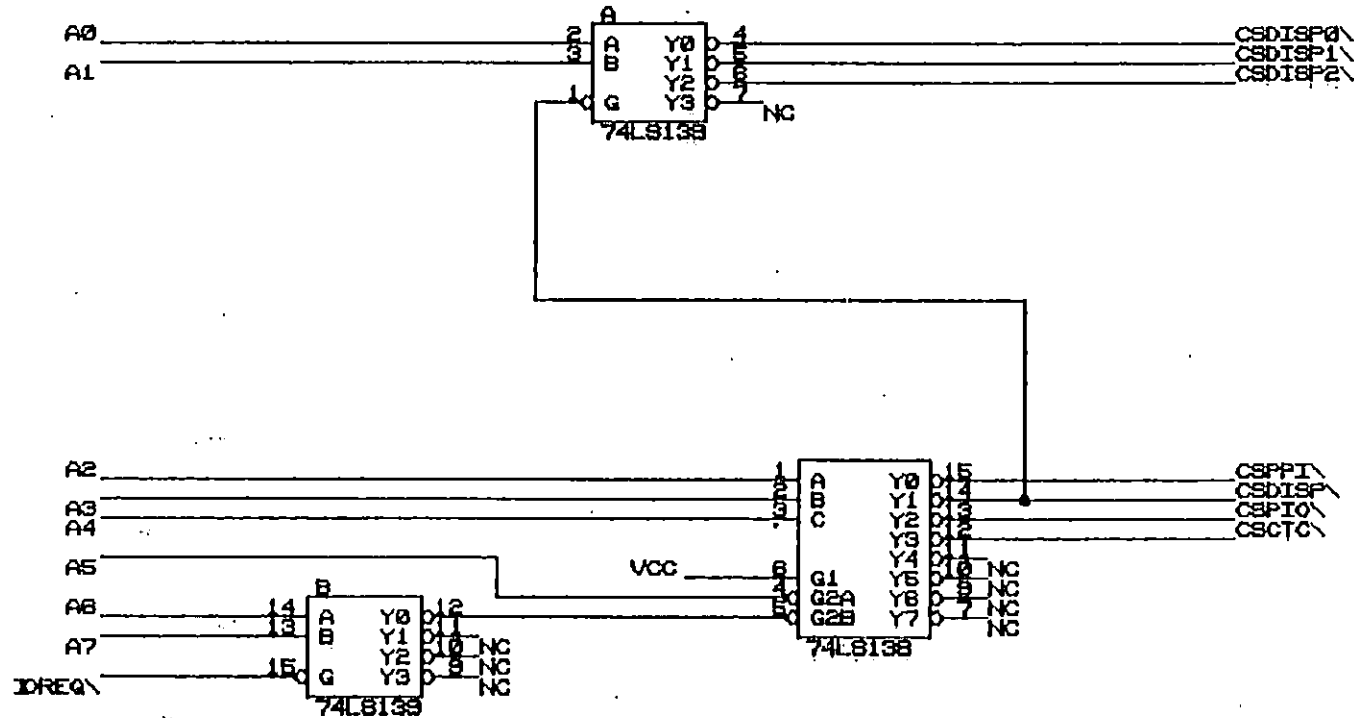


Fig. 3.5 Decodificador de entrada/salida

TRABAJO DE GRADUACION DISEÑO Y CONSTRUCCION DE UN SISTEMA DIDACTICO MINIMO BASADO EN EL Z-80.		
Title		
CIRCUITO DECODIFICADOR DE PERIFERICOS		
Size	Document Number	REV
A	PROZ80-000.2	001
Date:	June 18, 1993	Sheet 1 of 1

Hay que definir cual es el modo de operación más conveniente para los propósitos actuales, ya que se necesita interpretar tres clases de información, que son: datos, direcciones y comandos. Esto en virtud que la información que se transmitirá en su mayoría será código de máquina útil para el Z80. Este código es constituido por datos y direcciones, puesto que los programas en lenguaje de máquina están conformados por esa clase de información.

Por otro lado, recordemos que en el capítulo I se definió que el módulo que se construyera debería interpretar diferentes comandos para manejar los registros internos del microprocesador así como de memoria. Estos comandos exigen que se identifique las clases de informaciones ya mencionadas.

3.3.1 Comunicación entre dispositivos de velocidad media.

Hay que notar que en este caso existen condiciones específicas que están definidas por la naturaleza propia de los dispositivos a comunicar, las cuales son:

- * En esencia la comunicación que se establecerá será prácticamente entre computadores.
- * Las velocidades de operación entre la CPU Z80 y la PC son diferentes.

En función de estas condiciones es imperativo que existan señales de control para comunicar datos y así ambos dispositivos estarán habilitados en determinar en que momento están listos para recibir o mandar la información.

Los dispositivos de velocidad media deben ser sincronizados de alguna manera al reloj de tales dispositivos. Las CPUs no pueden simplemente tratarse como si ellas tomaran datos siempre o pudieran recibir datos en cualquier tiempo. Lo cierto es que cada dispositivo debe ser hábil para determinar cuando tiene nuevo dato de entrada o está listo para recibir datos de salida.

Debe también tener una forma de decirle al dispositivo que nuevos datos de salida están disponibles o que el dato de entrada previo ha sido aceptado.

3.3.2 Procedimiento de comunicación

El procedimiento estándar asincrónico es el diálogo. Aquí el emisor indica la disponibilidad del dato al receptor y transfiriere el dato; el receptor completa el diálogo reconociendo la recepción del dato.

El receptor puede controlar la situación inicialmente pidiendo el dato o indicando su disponibilidad para aceptar datos; el emisor luego manda el dato y completa el diálogo indicando que el dato está disponible.

En ambos casos, el emisor sabe que la transferencia ha sido completada exitosamente y el receptor sabe cuando el dato esta disponible. El proceso de diálogo puede operar a cualquier velocidad, ya que el emisor y el receptor controlan la secuencia de los eventos.

Lo discutido anteriormente afirma que el modo uno ó el modo dos del PPI serian los útiles para establecer la comunicación efectiva , ya que en estos modos el PPI 8255 proporciona las señales adecuadas de control y poder sostener el diálogo.

3.3.3 Descripción de pines del 8255

Los pines 1 a 4 representan el nibble menos significativo del puerto A, estan representados por PA3, PA2, PA1 y PA0 respectivamente.

El pin 5 es RD\, señal baja activa que realiza una operación de lectura de algún puerto

El pin 6 es el CS\, señal baja activa que habilita el funcionamiento del 8255.

El pin 7 es GND, la tierra del sistema.

Los pines 8 y 9 son A1 y A0. Estas lineas se encargan de seleccionar los diferentes registros del 8255:

A1A0 = 00 accesa al puerto A
A1A0 = 01 accesa al puerto B
A1A0 = 10 accesa al puerto C
A1A0 = 11 accesa al registro de configuración

Los pines 10 al 13 son PC7, PC6, PC5 y PC4 respectivamente. El nibble más significativo del puerto C.

Los pines del 18 al 25 son PBO a PB7, el puerto B

El pin 35 es el RESET, alto en esta señal inicializa al 8255 colocando todos sus puertos como entrada.

El pin 36 es WR\, señal que es baja cuando se realiza una operación de escritura en algún puerto.

Los pines 37 al 40 son PA7, PA6, PA5, PA4, respectivamente.

3.3.4 Descripción de las señales del conector de la CPU con el Z80

A0 - A15 (bus de direcciones) son salidas alto activos de tres estados, constituyen direcciones de 16 bits. Estos pines proporcionan las direcciones para decodificar dispositivos de memoria y E/S.

D0 - D7 son los ocho datos proporcionados por el Z80, son entrada/salida de tres estados, alto activo. Estos pines representan el bus de datos que son utilizados por las memorias y los dispositivos de entrada/salida.

M1\ es salida baja activa, indica que el ciclo de máquina actual es el ciclo de búsqueda del código de operación en la ejecución de una instrucción. También se produce con IOREQ\ para indicar el reconocimiento de un ciclo de instrucción.

MREQ\ es salida de tres estados, activa a nivel bajo, representa la señal de petición de memoria para efectuar una operación de lectura o escritura.

IOREQ\ es salida baja activa de tres estados indica que el byte bajo del bus de direcciones mantiene una dirección válida de E/S. También se genera con una señal M1\ cuando el Z80 reconoce una interrupción.

RD\ es salida baja activa de tres estados, RD\ indica que la CPU desea leer datos de la memoria o de un dispositivo de entrada/salida.

WR\ salida baja activa de tres estados, indica que el bus de datos de la CPU mantiene un dato válido para ser mandado a memoria o a un dispositivo de entrada salida.

RFSH\ es de nivel bajo activo cuando la parte baja del bus de direcciones contiene una dirección de refresco. Es útil para las memorias dinámicas.

HALT\ señal baja activa que indica que el Z80 ejecutó una instrucción HALT.

WAIT\ entrada activa a nivel bajo que indica a la CPU que el elemento direccionado, ya sea memoria o puerto de E/S no está apto para una transferencia de datos.

INT\ es entrada baja activa, al llegar una señal a este pin (de interrupción), se atenderá una petición al final de la instrucción que se está ejecutando.

NMI\ es la entrada de interrupción de más alta prioridad, siempre es reconocida al final de la instrucción que se está ejecutando.

RESET\ es entrada de nivel bajo que fuerza al contador del programa a cero.

BUSRQ\ entrada que sirve para indicar al Z80 que debe de poner en estado de alta impedancia el bus de datos, el bus de direcciones y las señales de control.

BUSAK\ salida baja activa que indica que se ha colocado el bus de datos, el bus de direcciones y las señales de control en estado de alta impedancia.

φ es el reloj a nivel de TTL.

3.3.5 Diseño de la interfase paralelo para comunicación Z80-PC

La interfase esta conformada por tres elementos: decodificador, one shot y el 8255.

El decodificador permite habilitar al 8255 para que ejecute las tareas de entrada salida. Los circuitos empleados son el 741s138 y el 741s139 que son dos decodificadores ampliamente utilizados.

Las líneas del conector (ver fig. 3.6) que intervienen en la decodificación son la A2, A3, A4, A5, A6 y la A7 más la señal IOREQ\ que define al 8255 como dispositivo de entrada/salida.

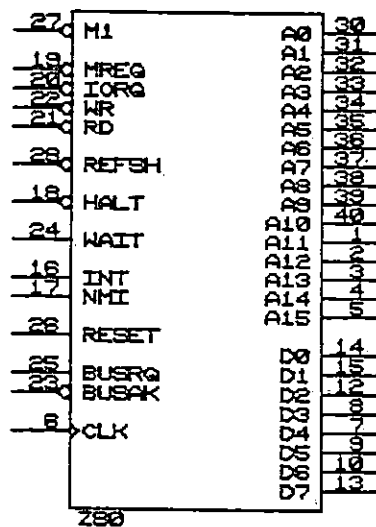


Fig. 3.6 Conector de expansion de la CPU Z80

Las líneas de dirección A0 y A1 direccionan los puertos del 8255. Al observar detenidamente el decodificador se notará que el PPI responderá a las siguientes direcciones:

- 00 Hex : puerto A
- 01 Hex : puerto B
- 02 Hex : puerto C
- 03 Hex : registro del byte de control

Las líneas RD\ (Z80) y WR\ (Z80) le indican al puerto el flujo de la información, es decir, si se ejecutará una operación de lectura y escritura en el PPI.

El RESET garantiza la inicialización del 8255 en el encendido de la máquina.

El puerto que recibirá la información mandada desde la PC será el puerto A, y las señales de protocolo son proporcionadas por lo siguientes pines del puerto C:

- PC0 y PC1 son empleados para definir el tipo de la información, codificada de la siguiente manera

PC1 PC0 = 00 no información útil
PC1 PC0 = 01 dirección
PC1 PC0 = 02 dato
PC1 PC0 = 03 comando

- PC4 y PC5 definen las señales de protocolo y establecen el diálogo con la PC.

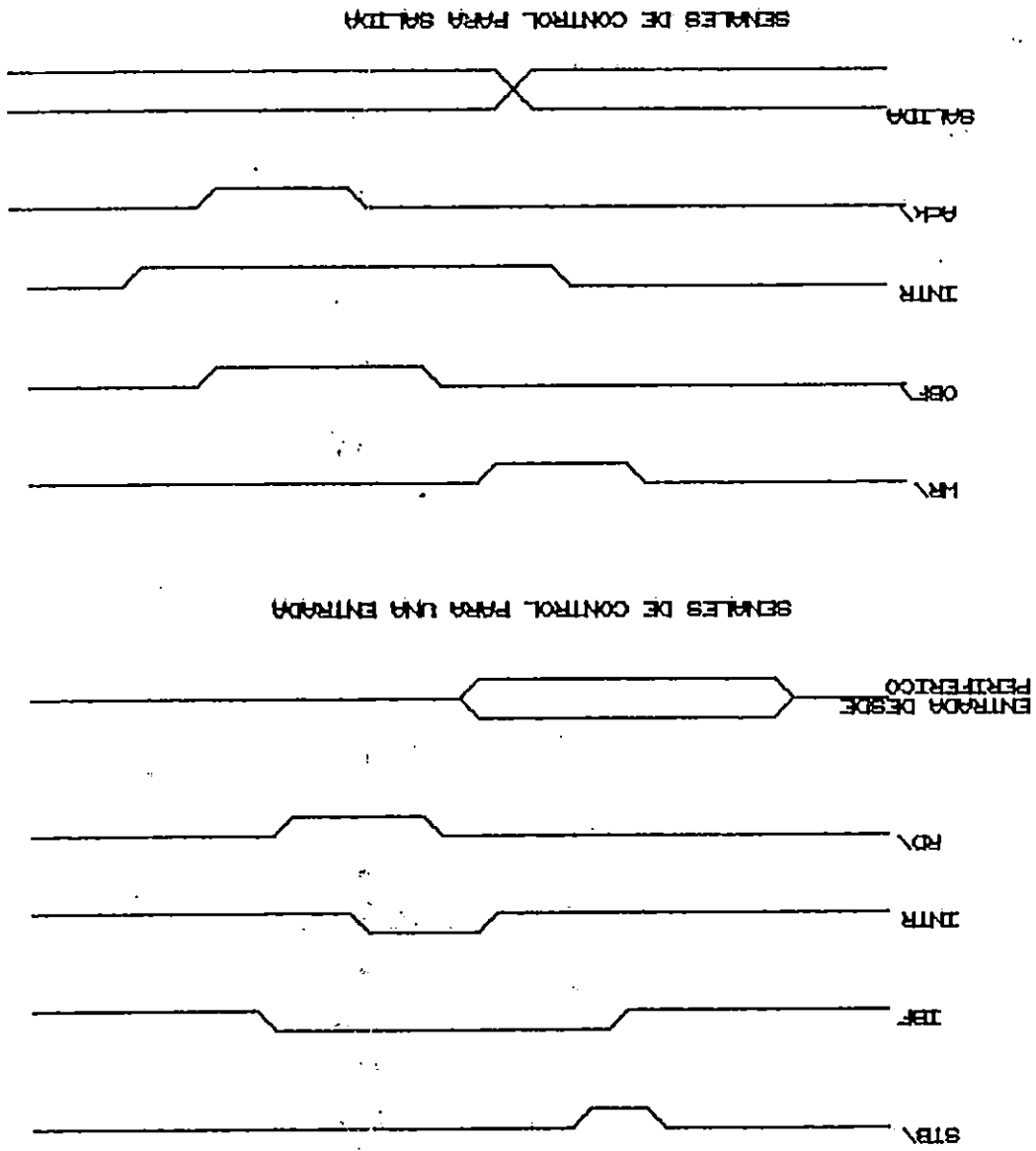
Cuando la PC desea mandar información al sistema con el Z80, se escribe al puerto A, del puerto de salida de la PC y se genera una señal OBF\ (PC7 del puerto de E/S de la PC), esta señal llega al 241s221, que es el elemento de un disparo empleado, generando un pulso. Este pulso llega al 8255 del sistema con el Z80 como un aviso de la existencia de información disponible (en el PC4 lado del Z80)

El 8255 del lado del Z80 recibe esta información y genera la señal IBF\ (PC5) para indicar que el dato ha sido cargado al latch de de entrada. Esta señal pasa a nivel cero cuando el Z80 hace una operación de lectura al puerto A.

Al pasar a nivel bajo se genera un disparo en el 741s221 y se genera la señal equivalente al ACK\ llegando al PC6 del PPI del lado de la PC. Indicándole a la PC que esta habilitada para mandar nuevamente datos. Las señales idóneas aparecen en la Fig. 3.7.

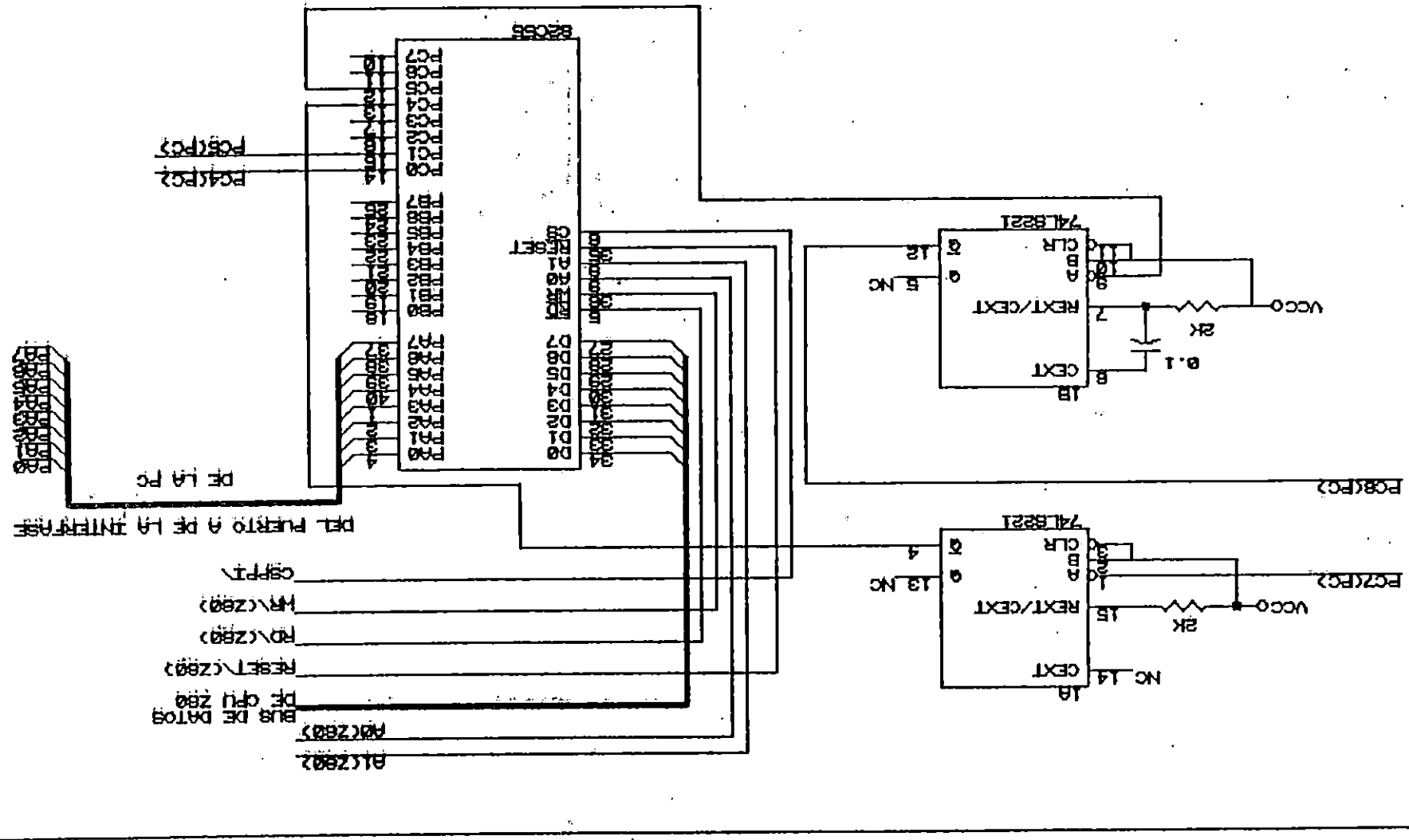
El diagrama del circuito de comunicación recién explicado puede verse en la Fig. 3.8. También se ha agregado nuevamente por comodidad, el decodificador de periféricos en la Fig. 3.9.

FIG. 3.7 Señales de protocolo para entrada/salida



TRABAJO DE GRADUACION
DISEÑO Y CONSTRUCCION DE UN SISTEMA
DIDACTICO MINIMO BASADO EN EL Z80
Titulo
CIRCUITO DE INTERFASE DE COMUNICACION PC/Z80
State Document Number
PROZ80-000-3
REV
Date: June 18, 1983 Sheet 3 of 3

Fig. 3.8 Interfase de comunicacion Z80-PC



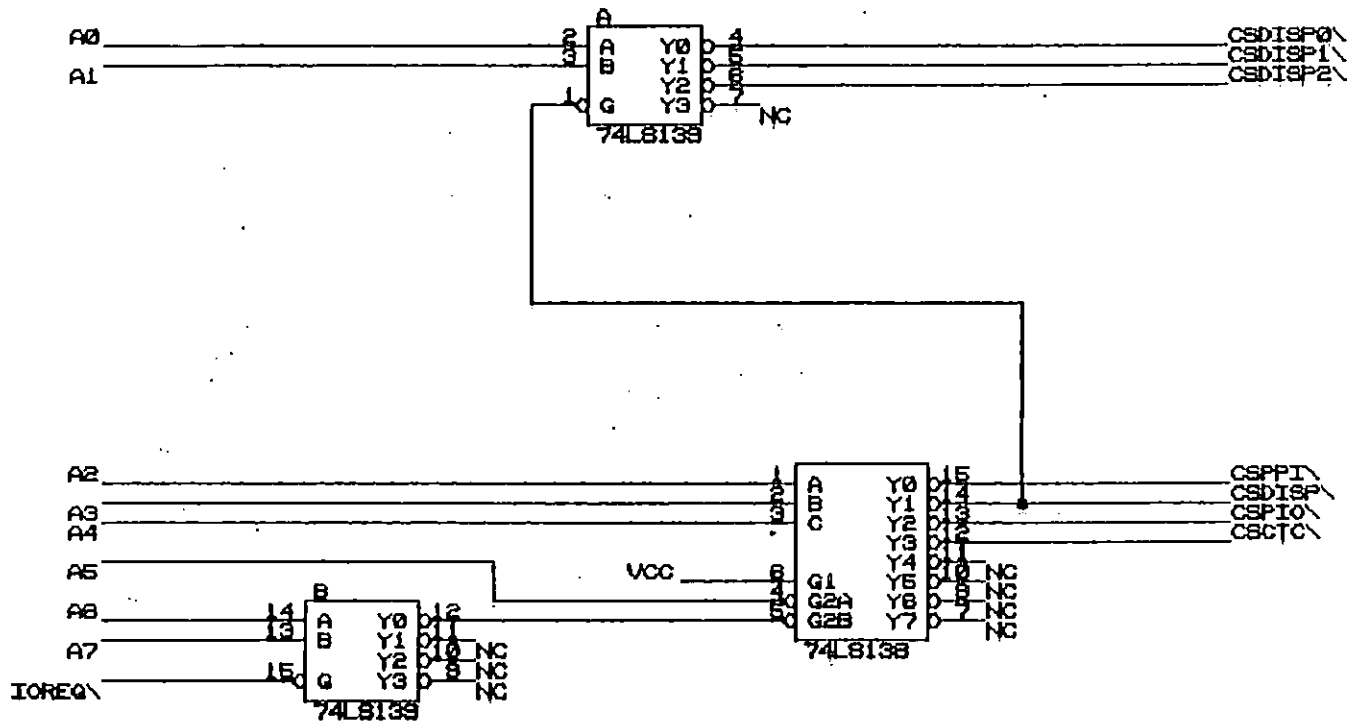


Fig. 3.8 Decodificador de entrada/ salida

TRABAJO DE GRADUACION		
DISEÑO Y CONSTRUCCION DE UN SISTEMA DIDACTICO MINIMO BASADO EN EL Z-80.		
Title		
CIRCUITO DECODIFICADOR DE PERIFERICOS		
Size	Document Number	REV
A	PROZAD-000.2	001
Date:	June 18, 1993	Sheet 1 of

3.4 El desplegador del sistema basado en el Z80

El desplegador de datos y direcciones es un circuito sencillo que esta conformado de TILS 311, Fig 3.10. Estos son desplegados con la capacidad de usar directamente el bus de datos del sistema con microprocesador para representar los datos que se necesiten en virtud de un latch de entrada que puede ser habilitado con una simple intrucción de salida.

Los cuatro desplegados de la parte superior del diagrama que aparece en la pagina siguiente se emplearan para representar los datos y direcciones en el caso cuando se desea ver posiciones de memoria y los de la parte inferior para representar los datos. La disposición física de estos desplegados es en una misma fila.

Cuando se deseen ver registros, el contenido del registro aparecerá en los desplegados donde se visualizan las direcciones y donde se visualizan los datos el código del registro que se esta accedendo. Lo mismo sucede para el caso de fijar el contenido del registro.

Las señales CSDISPO\, CSDISP1\ y CSDISP2\ seleccionan cual par de TIL desplegarán la información.

Del decodificador de direcciones(ver decodificador de E/S), se puede notar que las direcciones empleadas para su decodificación son :

- 04 Hex : par de desplegados de la izquierda
- 05 Hex : par de desplegados del centro
- 06 Hex : par de desplegados de la derecha

3.4.1 Descripción del TIL 311

El pin 1 y el pin 14 son las alimentaciones de la parte logica y de polarización de los LED del TIL.

Los pines 3, 2, 13 y 12 representan las entradas para desplegar el numero hexadecimal que se desea observar, manteniendo el peso binario respectivamente.

El Pin cuatro es entrada para desplega un punta a la izquierda del numero desplegado, este no es usado en nuestra aplicación.

El pin 5 corresponde a la entrada que hace posible ue se atrape el dato presente en los pines 3, 2, 13 y 12.

El pin 7 es la tierra del sistema.

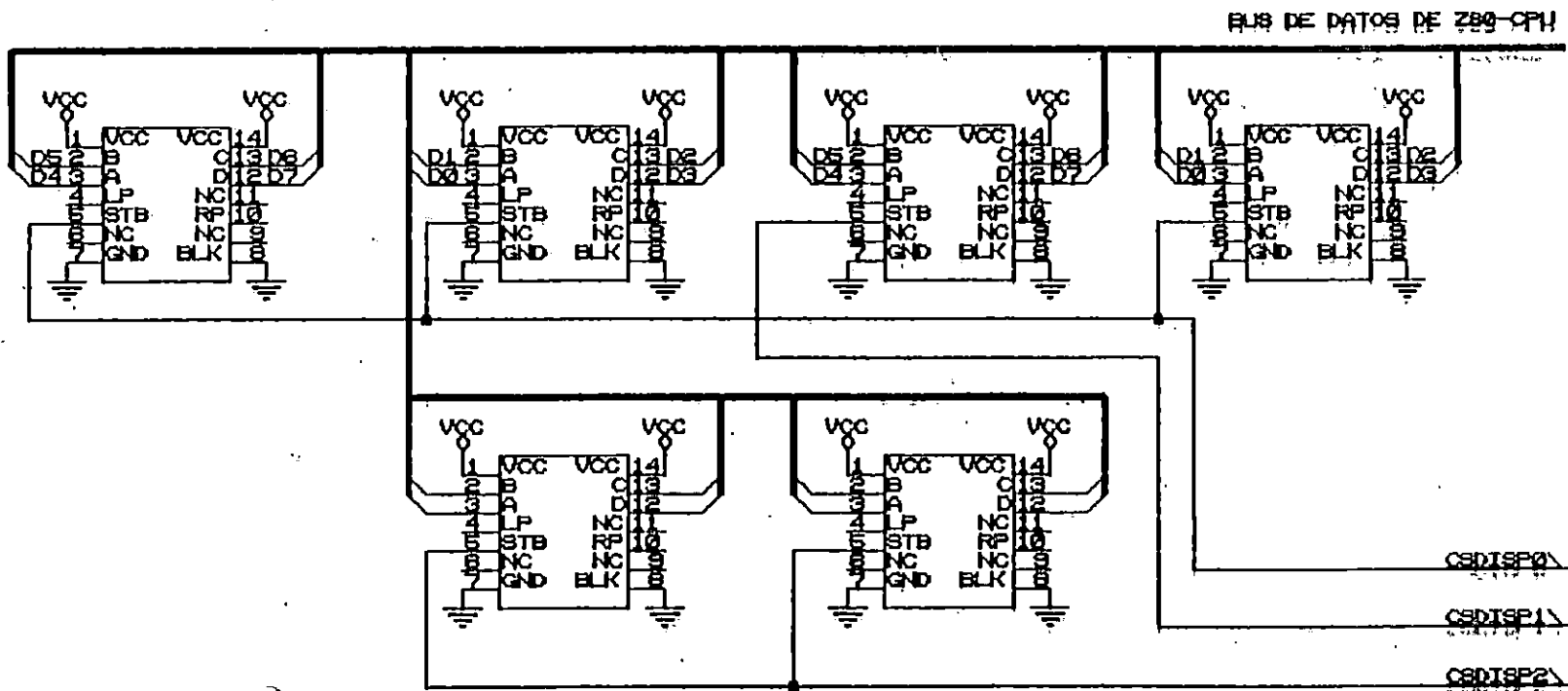


Fig. 3.10 Desplegador del sistema

TRABAJO DE GRADUACION DISEÑO Y CONSTRUCCION DE UN SISTEMA DIDACTICO MINIMO BASADO EN EL Z80		
Title		
CIRCUITO DESPLEGADOR		
Size	Document Number	REV
A	PROZ80-000-4	001
Date:	June 18, 1983	Sheet 1 of 1

El pin 8 es entrada para poner a cero el desplegador

Los pines 6, 9 y 11 no se conectan.

3.5 Conexión del circuito PIO y CTC

El circuito PIO del Z80, es un circuito de entrada/salida paralelo, el circuito PIO esta diseñado para proporcionar dos puertas programables y compatibles a nivel TTL, para la transferencia de datos paralelo entre la CPU Z80 y los dispositivos perifericos.

El PIO proporciona las siguientes posibilidades necesitandose una pequeña circuitería externa:

- Dos puertas independientes de 8 bits en paralelo, bidireccionales para la interfase de perifericos, con control de la transferencia de datos del tipo dialogo.

- Entrada/salida mandada por interrupción.

- Cuatro modos de funcionamiento seleccionables mediante software:

- Modo 0 - salida de byte
- Modo 1 - entrada de byte
- Modo 2 - byte bidireccional(para puerto A)
- Modo 3 - Control de Bit

Cada modo de funcionamiento utiliza handshaking, controla do por interrupción.

- Logica de interrupción de prioridad en cadena serie.

- Todas las entradas y salidas son completamente compatibles con TTL.

3.5.1 Descripción de patillas del circuito PIO

La Fig. 3.11 es un diagrama de la configuración del circuito PIO. La descripción de cada patilla es la siguiente:

D7-D0 representa el bus de datos el cual es usado para transferir todos los datos y mandos entre la CPU y el PIO. D0 es el bit menos significativo del bus.

B/A es entrada activa a nivel alto, selecciona el puerto B o el puerto A. A menudo el bit de dirección A0 de la CPU es utilizado para la función de selección.

C/D esta patilla define el tipo de transferencia de datos a ser realizada entre la CPU y el PIO. Un nivel alto en este pin durante la acción de escritura de la CPU al PIO provoca que el bus de datos sea interpretado como mando. A menudo el bit de dirección A1 es conectado a este pin.

CE\ es el habilitador del Chip. Esta señal es generalmente una decodificación del bus de direcciones.

ϕ es el reloj del sistema de una sola fase.

M1\ Señal del ciclo de maquina de la CPU, es empleada como pulso de sincronización para controlar varias funciones internas del PIO.

IOREQ\ es la señal de petición de entrada salida, el flujo de la información dependera de las señales RD\ y WR\. Si esta señal esta activa al mismo tiempo que M1\ significa que la CPU reconoce una interrupción y el PIO colocara inmediatamente un vector de interrupciones.

RD\ pin empleado para leer datos del PIO

IEI esta señal es empleada para formar una cadena de prioridades de interrupción cuando mas de un dispositivo mandado por interrupción esta siendo utilizado. Un nivel alto en esta patilla indica que ningun otro dispositivo de mayor prioridad está siendo atendido por la CPU.

IEO esta es la otra señal necesaria para formar el esquema de prioridades serie. Está a nivel alto si la CPU no está sirviendo a una interrupción desde este PIO.

INT\ cuando es activa el PIO esta solicitando una interrupción a la CPU.

A0-A7 son los pines del puerto A

A STB\ señal para ser utilizada en comunicación de protocolo. Tiene diferentes significados de acuerdo al modo de operación del puerto A.

A RDY tambien es señal empleada en comunicación asincrona y depende del modo de operación del puerto su significado.

B0-B7 representa los bits del puerto B. B0 es el menos significativo.

B STB\ Impulso de sincronización para el puerto B.

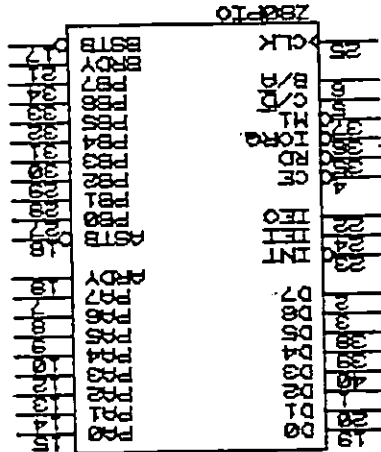
1. El método de comunicación dialogada es de las mejores formas de comunicación para esta aplicación, ya que permite a los sistemas tomar cierta independencia y efectuar otras tareas y atender la petición en el momento que se desee.

2. Las velocidades de operación de cada sistema no afecta en la comunicación defitiva, ya es el protocolo asincrono el que se encarga de hacer coherencia en la transmisión de los datos.

3.6 Conclusiones Y Recomendaciones

El circuito construido del sistema mínimo con el Z80 puede verse al final en el ANEXO A2.

Fig. 3.11 Descripción de pines del PIO



B-RDY: Impulso que establece si el puerto B está preparado.

El esquema de la siguiente página muestra la conexión del sistema con el módulo que se ha construido. Están conectados de tal manera que el PIO tenga alta prioridad de interrupciones ya que IEI del PIO está conectado a VCC.

El PIO está conectado de tal manera que responda a las direcciones 08-0B hex y el CTC para las direcciones 0C-0F.

3. La necesidad de codificar la información que se transmite es obvia y ha resultado de gran eficiencia, ya que se pueden implementar algoritmos de corrección de errores que detectan si se ha perdido información en la transmisión.

4. Los dispositivos periféricos utilizados son bastante sencillos sin embargo se puede utilizar el sistema para manejar periféricos mas complejos.

5. Para tener control de interrupciones se diseño un circuito que desvia al microprocesador a ejecutar su primera instrucción en la dirección 800 hex, ya que las localidades de interrupciones están en los primeros 256 bytes, y el microprocesador por defecto empieza a ejecutar en la posición de memoria 00H.

6. Se implementó un circuito de reloj para poder variarle las velocidades de operación de el microprocesador con el objetivo de correr programas mas lentamente y poder visualizar resultados.

7. Se deja un espacio de 2 Kbytes de memoria para poder hacer ampliaciones a las capacidades del módulo y proporciona la facultad para hacerlo en un futuro mas poderoso. Por ejemplo llegar a crear una computadora propias de la Escuela de Ingeniería Eléctrica o crear microcontroladores dedicados, ya sus aplicaciones pueden ser diversas.

Referencias bibliográficas:

1. A. A. Bruce, Microcomputer Interfacing, Prentice Hall, New Jersey, 1980
2. C. N. Joseph, A. Nichols elizabeth, R. R. Peter, Microprocesador Z80, Programacion e Interfases, Mexico: Alfaomega-Marcombo, 1984.
3. C. Steve, Construya una microcomputadora basado en el Z80Mc Graw Hill.
4. G, Harry, Introduccion to Microprocessor System Design. Department of Electrical Engineering, Stanford University, 1978.
5. G. Robert, The 8088 Project Book Blue Ridge Summit CA, TAB BOOKS, 1989.
6. L. M. Christofer, Introduccion al Microprocesador 8086/8088, Mc Graw Hill :Mexico, 1982.
7. Lesea Austin, Microprocessor Interfacing Technique, U.S.A: SYBEX, 1977.
8. P. S. Marco, Diseño y construccion de un dispositivo analizador de distorsion armonica en redes de baja tension, Tesis para optar al grado de ingeniero electricista. Biblioteca de la Escuela de Ingenieria Electrica, Universidad de El Salvador. Octubre de 1991.

CAPITULO IV

OPERACION Y APLICACIONES DEL SISTEMA

Introducción

El sistema didáctico mínimo Z-80 construido ha sido diseñado para contribuir de alguna manera al desarrollo de otros sistemas basados en el Z-80 e incluso en otros microprocesadores. Tal como su nombre lo indica, éste sistema incluye los componentes mínimos requeridos para conseguir una herramienta de desarrollo de sistema. La utilización del sistema es bastante simple, sobre todo para aquellas personas habituadas a trabajar con lenguajes de ensamble, depuradores, enlazadores, etc.

Además de los fines estrictamente didácticos, un sistema de este tipo puede ser fácilmente adaptado para realizar tareas de control, monitoreo, análisis de señales, etc. De hecho, una vez depurado completamente, bastaría agregar uno o dos dispositivos externos y hacer un programa apropiado para dedicarse a una nueva aplicación. Las posibilidades son muchas y muy variadas.

4.1 Operación del sistema didáctico mínimo Z-80

En esta sección se presentan los usos básicos del SDM_Z80. Se describirá a continuación la utilización de los distintos comandos:

- 1.- Examinar el contenido de una posición de memoria.
- 2.- Cambiar el contenido de una posición de memoria.
- 3.- Introducir datos en memoria.
4. Ejecutar un programa en lenguaje de máquina localizado en memoria.
- 5.- Examinar el contenido de un registro del Z-80.
- 6.- Cambiar el contenido de un registro del Z-80.
- 7.- Cargar un programa a partir de la PC.
- 8.- Correr un programa en modo Paso a Paso (Trace).
- 9.- Establecer/Borrar un punto de ruptura (breakpoint)

10.- Desactivar/activar puntos de ruptura.

Cada uno de estos comandos será examinado detalladamente a continuación:

4.1.1 Examinar memoria.

Se puede examinar el contenido de la memoria del sistema didáctico Z-80 escribiendo M y luego introduciendo la dirección que se quiere examinar (en formato hexadecimal).

Por ejemplo si se quiere ver el contenido de la posición de memoria 0400h habría que escribir lo siguiente:

```
M 04 00
```

Inmediatamente aparecerá en los visualizadores la posición de memoria que se está examinando así como su contenido. El par de visualizadores más a la izquierda representa la parte alta de la dirección, el siguiente par representa la parte baja de la dirección, y el par de visualizadores más a la derecha representa el contenido de esa posición de memoria.

Para ver la posición de memoria siguiente únicamente hay que presionar la tecla S. Para ver la posición de memoria anterior presionar la tecla A.

Para salir del comando Ver Memoria, únicamente hay que presionar la tecla Q, con lo cual se regresa al módulo de procesamiento de comandos.

4.1.2 Cambiar contenido de memoria

Para cambiar el contenido de una posición de memoria después de haberla examinado se escribe / y luego se introduce el nuevo contenido (en hexadecimal).

Por ejemplo, al examinar el contenido de la posición 400h se escribiría:

```
M 04 00
```

Suponiendo que ésta contenga el valor 6F y se quiere que contenga el valor AB, entonces se escribiría /AB. Al momento de hacer el cambio en la PC, éste cambio debe reflejarse en el visualizador del módulo Z-80

Para que el sistema Z-80 didáctico ejecute un programa, todo lo que hay que hacer es decirle donde comenzar y luego utilizar el comando G. El programa monitor ejecuta un programa haciendo una llamada a la posición de inicio especificada. Es responsabilidad del usuario determinar qué es lo que hará el Z-80 luego de ejecutar su programa. Lo más

4.1.4 Ejecución de un programa

M 00 42

Para observar el resultado luego de la ejecución del programa, todo lo que hay que hacer es escribir :

1. Escribir M 00 40 , luego la tecla RETURN
2. Escribir /AB (introducimos el valor AB en la posición 40)
3. Escribir S para observar el valor de la posición 41
4. Escribir /CD (introducimos el valor CD en la posición 41)

Para introducir los datos en las posiciones 40 y 41 se seguirían los siguientes pasos:
Este programa necesita datos en las posiciones de memoria 40 y 41. El resultado es guardado en la posición de memoria 42.

```
LD A,(40)
AND (41)
LD (42),A
RET
```

Supóngase que se tiene en memoria el siguiente programa:

Antes de indicarle al módulo Z-80 que corra un programa, se necesita de una forma de introducir datos y observar como esos datos afectan los resultados de la corrida del programa. Esto es muy sencillo: dado que podemos utilizar los procedimientos descritos para examinar y cambiar el contenido de posiciones de memoria.

4.1.5 Introducción de datos.

recomendable es que el usuario termine su programa con una instrucción RET. De esta manera se garantizará que al terminar la ejecución del programa se retorne al procesador de comandos. Sin embargo si al ejecutar un programa, todo indica que ha habido algún problema, siempre queda la posibilidad de usar el botón de reset.

Para indicarle al módulo Z-80 que se quiere establecer una posición de inicio (por ejemplo la posición 800h) se escribe:

```
I 0800          seguido por la tecla RETURN
```

Luego se le indica que ejecute el programa localizado a partir de la posición antes definida.

4.1.5 Examinar registros

Para examinar el contenido de un registro se hará de una forma parecida a examinar una posición de memoria, sólo que en este caso con el comando R :

```
R HL
```

Con éste comando estaríamos viendo el contenido del par de registros HL. El valor de estos registros se presentará en los desplegados destinados a las direcciones, siguiendo la misma convención que con la memoria. El par de visualizadores destinados a los datos presentará en este caso un código que se le ha asignado a cada par de registros.

4.1.6 Alterar el contenido de registros

Para cambiar el contenido de un registro se sigue el mismo procedimiento que para cambiar el contenido de una posición de memoria.

Primero se usa el comando de visualizar registro, y una vez visualizado se puede alterar escribiendo / y luego el nuevo valor del registro

Por ejemplo si se quiere cambiar el valor de H:

```
R HL
```

Suponiendo que en el visualizador aparece que HL tiene un valor de 03FF. Si se quiere que H tenía un valor de 07 entonces se escribe:

4.1.7 Cargar un programa partir de la PC .

Para cargar un programa a partir de la PC se utiliza el comando L. Al darle éste comando, la PC nos pedirá el nombre del programa que se desea enviar al módulo. Este programa debe encontrarse en formato INTEL HEX para que pueda ser enviado. Una vez enviado, entraremos de nuevo al procesador de comandos.

L

"Nombre del Archivo a Transferir?" Prueba.hex

A continuación se mostrará en la pantalla del computador cada dato que vaya siendo enviado, y por último estaremos de regreso en el procesador de comandos. Se puede verificar que la comunicación se ha dado si el dato que aparece en el desplegador del sistema Z-80 es el mismo que el último dato enviado.

4.1.8 Comando de trazado de registro.

Con éste comando le indicamos al módulo Z-80 cuál es el registro que se quiere trazar cuando se corre un programa en el modo de paso a paso. En el modo paso a paso, la ejecución del programa se detiene después de cada instrucción, para examinar si se está haciendo un trazado de registro o si se ha llegado a un breakpoint.

Si por ejemplo se quiere examinar como se va modificando el registro A con la ejecución de cada instrucción, se escribiría el siguiente comando:

T A

Al dar el comando de ejecución, el módulo Z-80 ejecutará una instrucción y luego presentará el valor del registro A en el visualizador correspondiente. Para que se ejecute la siguiente instrucción hay que escribir S. Nuévemente el módulo Z-80 presentará el valor del registro A y quedará a la espera de la siguiente orden. Se puede continuar de esta manera hasta que el programa que se está trazando devuelva el control al programa monitor (cuando se encuentre la instrucción RET final) ó se puede retornar en cualquier momento al programa monitor escribiendo Q.

Si además del registro A se quiere ver como van quedando los otros registros; lo que hay que hacer es escribir SR en lugar de S. De ésta manera el siguiente valor que se presente en los desplegados corresponderá al valor del registro que le sigue a A según el orden que se les ha establecido en el comando VER REGISTRO. Se puede continuar escribiendo SR para ir viendo los valores de los otros registros. Se puede saber cuál registro se está viendo observando el par de desplegados que indican el número del registro. Para trazar la siguiente instrucción hay que escribir S.

Si no se quiere ir trazando de instrucción en instrucción sino que se quiere ver como queda el registro luego de varias instrucciones, lo que hay que hacer es agregarle al comando el número de instrucciones que se quiere ejecutar antes de que presente el valor del registro que está siendo trazado.

Por ejemplo si se quiere conocer el valor del registro B luego de que se ejecuten 9 instrucciones, el comando a escribir sería:

T A 9

4.1.9 Establecer/Borrar un punto de ruptura

Otra manera de depurar un programa es estableciendo puntos de ruptura. Cuando el sistema se encuentre en el modo de paso a paso y los puntos de ruptura se hayan habilitado, el procesador parará de ejecutar el programa del usuario cada vez que se encuentre en el rango de memoria depurable (arriba de los 4K bytes del mapa de memoria). La parada del programa se debe a que se genera una interrupción no mascarable, la cual lleva el flujo de la ejecución a la rutina de servicio de la NMI. Esta rutina verifica inicialmente si los puntos de ruptura han sido habilitados. Si ese es el caso, cada dirección de un breakpoint activo se compara con la dirección de la siguiente instrucción que se va a ejecutar. Si la dirección de uno de los breakpoints activos concuerda con la dirección de la instrucción a ser ejecutada, la ejecución del programa se detiene y se transfiere el control al programa monitor.

Para activar/desactivar cualquiera de los cuatro puntos de ruptura permitidos (breakpoint 0 a breakpoint 3) se utiliza el comando B.

Por ejemplo para un breakpoint en la dirección 1000h se escribiría:

B 0 10 00

Con este comando se está indicando que se quiere poner el breakpoint 0 en la dirección 1000h.

Para desactivar el breakpoint 0 únicamente se escribiría 0 en la dirección correspondiente a ese breakpoint, o sea:

B 0 0

4.1.10 Habilitar/Deshabilitar Breakpoints

El comando TB sirve para activar o desactivar los puntos de ruptura. Cuando los breakpoints se encuentran habilitados y el sistema se encuentra en el modo paso a paso, los breakpoints en un programa son revisados por el programa monitor luego de ejecutar cada instrucción del usuario. La forma normal de trabajar con este comando es asignar inicialmente las direcciones de ruptura mediante el comando B y luego habilitar los puntos de ruptura mediante el comando TB cuando el usuario quiere que sean chequeados.

El comando TB también permite que los puntos de ruptura sean deshabilitados temporalmente sin necesidad de tener que reestablecerlos posteriormente.

Para saber si en un momento dado los puntos de ruptura están habilitados o deshabilitados, lo que se hace es ejecutar el comando TB, y como resultado nos indicará si se encuentra en ON ó en OFF. Por ejemplo si al ejecutar el comando TB nos da por resultado un OFF, ésto indica que los breakpoints se encontraban habilitados antes de que diéramos el comando TB.

4.2 Posibles aplicaciones del Sistema didáctico mínimo Z-80.

Además de servir como un instrumento educativo y para propósitos de desarrollo de sistemas basados en el Z-80, existen muchas otras posibilidades que podrían ser exploradas y explotadas ya sea con el sistema mínimo actual o con uno mejorado.

Entre las posibles aplicaciones que podrían intentarse se pueden mencionar:

- 1.- Controlador de instrumental médico.
- 2.- Probador de circuitos integrados.
- 3.- Monitoreo de líneas de potencia.
- 4.- Sistema de adquisición de datos biomédicos y procesador de señales.
- 5.- Controlador de motores o transformadores

- 6.- Controlador de sistemas de energía solar.
- 7.- Controlador de procesos industriales.
- 8.- Controlador de sistemas de seguridad residenciales o industriales.
- 9.- Controlador de sistemas de climatización.
- 10.- Simulador de termocuplas.
- 11.- Controlador de temperatura de hornos industriales

En fin, las posibilidades son casi infinitas. De hecho puede utilizarse en casi cualquier aplicación en la que pueda usarse un computador, toda vez que no se requiera de una gran velocidad ni de un manejo intensivo de grandes cantidades de información. Podría utilizarse en aquellos casos en que la utilización de un computador completo resulte demasiado onerosa, ya sea porque un computador estaría siendo subutilizado o porque la actividad requiere que el computador le esté dedicando demasiado de su tiempo y no pueda dedicarse eficientemente a otras actividades que le son asignadas.

Por otra parte hay que recalcar que independientemente de la diversidad de aplicaciones en las que se puede utilizar, en su forma actual el sistema es barato y en efectivo representa una alternativa para resolver problemas diversos. Si bien es cierto que existen dispositivos similares, estos son caros. Ciertamente el dispositivo resultó ser muy versátil y económico.

4.3 Conclusiones y recomendaciones.

1. El sistema desarrollado puede ser de mucha utilidad tanto a nivel didáctico como en otras aplicaciones.
2. Para fines didácticos, los comandos son suficientes, ya que permiten la depuración y la creación de programas de una manera bastante sencilla.
3. La enorme ventaja de los sistemas basados microprocesadores es que pueden fácilmente adaptarse para realizar un trabajo distinto al que originalmente se cocibió.
4. Sería recomendable incluir un pequeño programa ensamblador en la ROM de expansión a fin de facilitar la prueba de pequeñas rutinas.
5. Otros elementos que aumentarían la flexibilidad del sistema serían la inclusión de un teclado y de un visualizador de caracteres ASCII, lo cual permitiría la creación de programas más interactivos.

Referencias bibliográficas

1. Rockwell International, AIM Microcomputer User's Guide, USA: Rockwell International, 1979.
2. C. Steve, Construya una microcomputadora basada en el Z80, México: Mc Graw-Hill, 1979.

Conclusiones y recomendaciones generales

1. El diseño e implementación del dispositivo por su esencia tendrá diversas aplicaciones y por tanto es un instrumento de gran versatilidad.
2. Económicamente los sistemas basados en microprocesadores ha disminuido enormemente y se estima que en el futuro cercano los costos seguirán disminuyendo de tal manera que las personas estarán interactuando con tales sistemas mas frecuentemente. De ahí la necesidad de conocerlos.
3. La investigación orientada a la creación de dispositivos para la enseñanza académica es importante porque genera nuevos mecanismos de aprendizaje y crea alternativas de instrumentación económicas para la Escuela de Ingeniería Eléctrica de la Universidad.
4. El seguir un proceso de diseño bien definido es bastante saludable ya que permite encontrar el camino, hacia el objetivo que se persigue, de forma ordenada.
5. En sistemas de comunicación, el método de comunicación dialogada es de las mejores formas, ya que permite a los sistemas tomar cierta independencia y efectuar otras en el momento que se desee.
6. Se recomienda adaptarle un teclado y un desplegador de caracteres ASCII, a fin de crear un sistema mas interactivo.
7. El diseño de circuitos impresos para este tipo de sistemas es sumamente complicado, por lo tanto se recomienda hacer esfuerzos reales para mejorar la técnica de circuitos impresos. En estos momentos resulta beneficioso usar el híbrido circuito impreso y wire wrap.

ANEXO A1

SOFTWARE Y DIAGRAMAS DEL CIRCUITO PARA LA IMPLEMENTACION DE
UN SISTEMA MINIMO INICIAL BASADO EN EL Z80.

A1.1 Programa para el circuito básico de sistema con Z80.

Para probar el circuito básico de un sistema con el Z80 se diseñó un programa simple de entrada salida externo. El objetivo de tal programa es el de entrar el dato presente en el puerto A del PPI y después de cierto tiempo presentarlo en el puerto B del mismo PPI.

En primer lugar se presenta el algoritmo del programa diseñado, para posteriormente presentar el código fuente del programa en lenguaje assembly del Z80. Se agrega el mapeo de los elementos empleados (memoria y puerto de E/S).

Finalmente se presenta el circuito totalmente detallado del sistema para ser implementado por aquellos que estén interesados en empezar a crear su propio sistema.

A1.2 Algoritmo del programa empleado por el sistema con el Z80.

El algoritmo que se presenta consiste en una entrada salida fundamental, ya que el PPI es el único dispositivo de entrada/salida externa en el sistema.

Los pasos del algoritmo son los siguientes:

1. Definir la configuración del PPI.
(Configurar el puerto A como entrada y el puerto B como salida)
2. Leer el dato presente en el puerto A.
3. Si el dato leído es igual a cero, parar.
4. Si es diferente de cero, sacar el dato por el puerto B.
5. Generar un retardo.
6. Terminar.

A1.3 Código fuente del programa de e/s fundamental del sistema con el Z80.

El siguiente listado representa el código fuente del programa de e/s implementado en memoria EPROM, para probar el sistema básico con el Z80:

```
;DEFINICION DE CONSTANTES
BYTCON EQU 03H
CONFIG EQU 90H
PORTA EQU 00H
PORTB EQU 01H
```

```

;CONFIGURACION DEL PUERTO DE E/S(PPI)
LD C,BYTCON
OUT (C),CONFIG; PUERTO A: ENTRADA Y B: SALIDA
;
;ENTRADA DE DATOS DESDE EL PUERTO A
INPA: LD C,PORTA
IN A,(C);ENTRAR EL DATO DESDE EL PUERTO A
;
;
CP 00H
JP Z,PARAR
;
;
;SACAR DATO EN EL PUERTO B
LD C,PORTB
OUT (C),A;MOSTRAR DATO
;
;
;RETARDO
LD A,FFH
LAZO: LD C,FFH
LAZO1: DEC C
JR NZ,LAZO1
DEC A
JR NZ,LAZO
JP INPA
;
;
PARAR: HALT

```

A1.4 Mapa de memoria y circuito del sistema con el Z80

Dado que los dispositivos conectados al Z80 solo son: una memoria RAM, y una EPROM, más un circuito de entrada/salida; el mapa de memoria es sencillo, el cual se muestra a continuación:

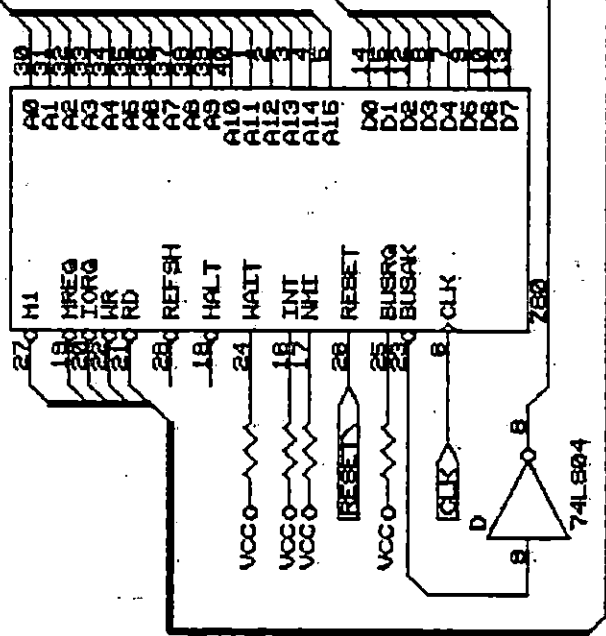
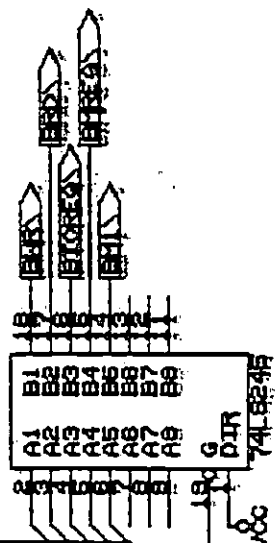
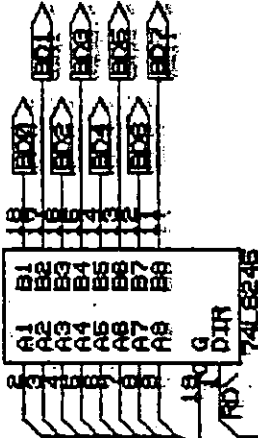
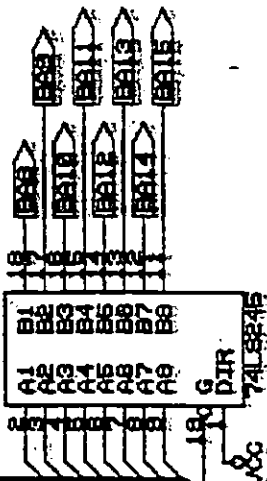
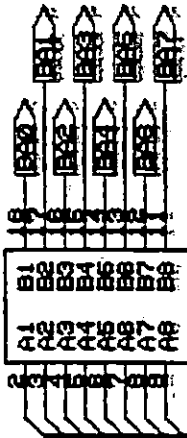
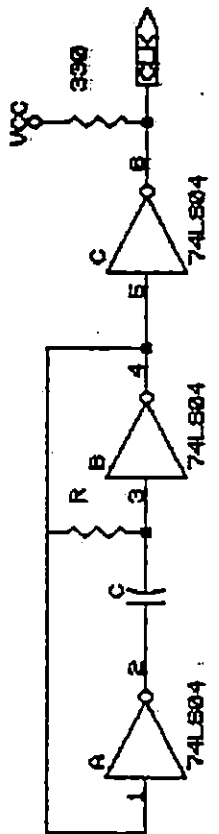
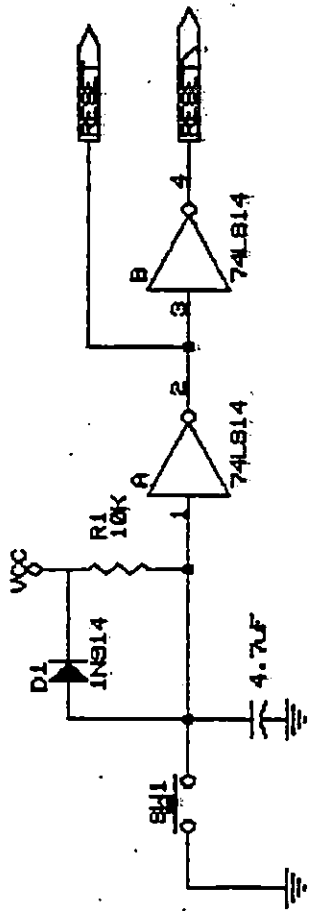
- Mapa de memoria

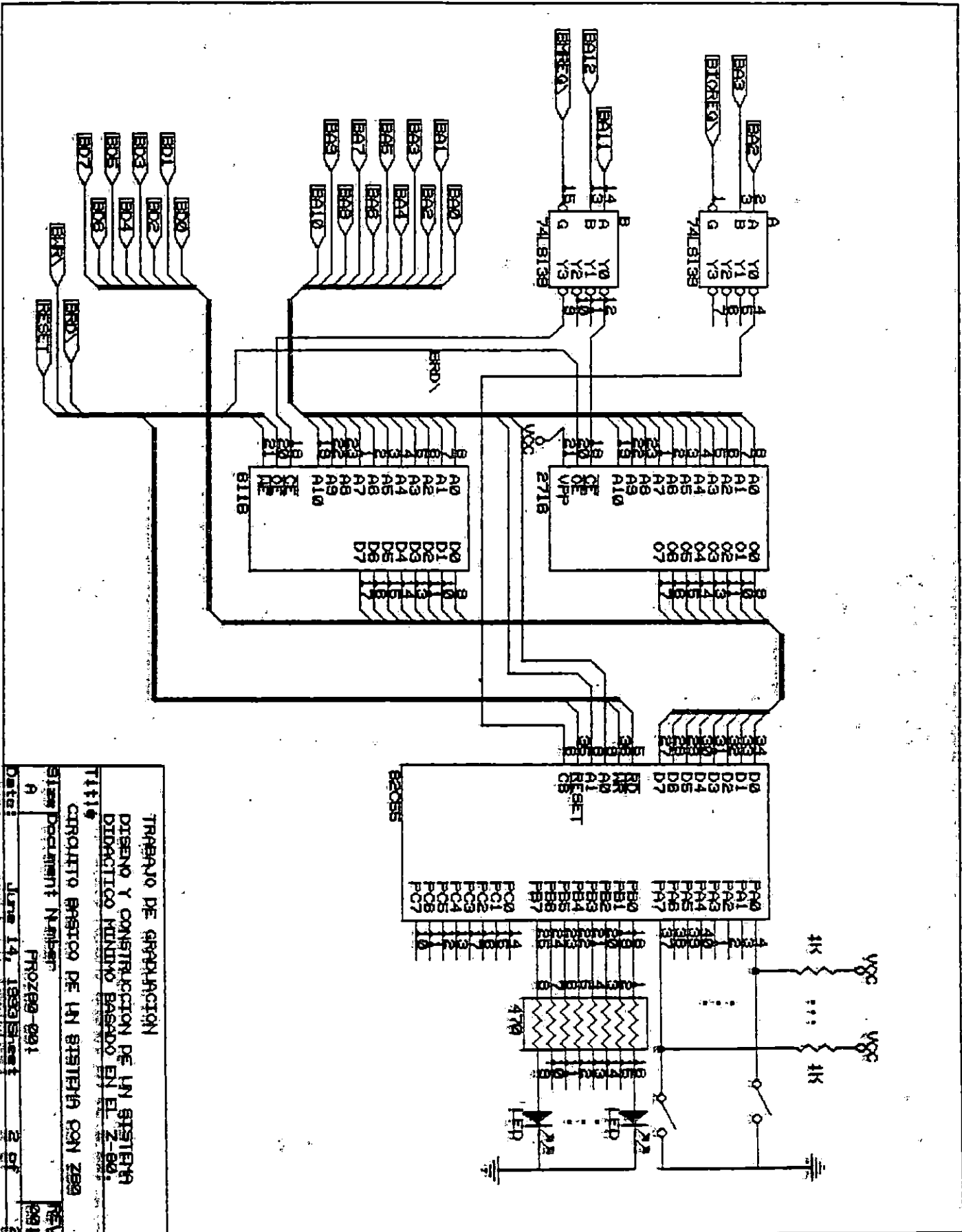
Direcciones	Dispositivo
0000 - 07FFH	EPROM
1800 - 1FFFH	RAM

-Mapeo de entrada/salida

Direcciones	Puerto del PPI
00H	P. A (PORTA)
01H	P. B (PORTB)
02H	P. C
03H	Registro de control (BYTCON)

El diagrama de circuito aparece a continuación, notar que los dispositivos responderán a otras direcciones!!.





TRABAJO DE APLICACION

DISEÑO Y CONSTRUCCION DE UN SISTEMA DIDACTICO MINIMO BASADO EN EL Z-80.

CIRCUITO BASICO DE UN SISTEMA CON Z80

Sheet Document Number

Proj: 09-091

Date: June 14, 1983

ANEXO A2

CIRCUITOS DEL SISTEMA DIDACTICO MINIMO BASADO EN EL Z80 Y
ALGORITMOS DEL SOFTWARE IMPLEMENTADO

A2.1 Algoritmos del software del sistema:

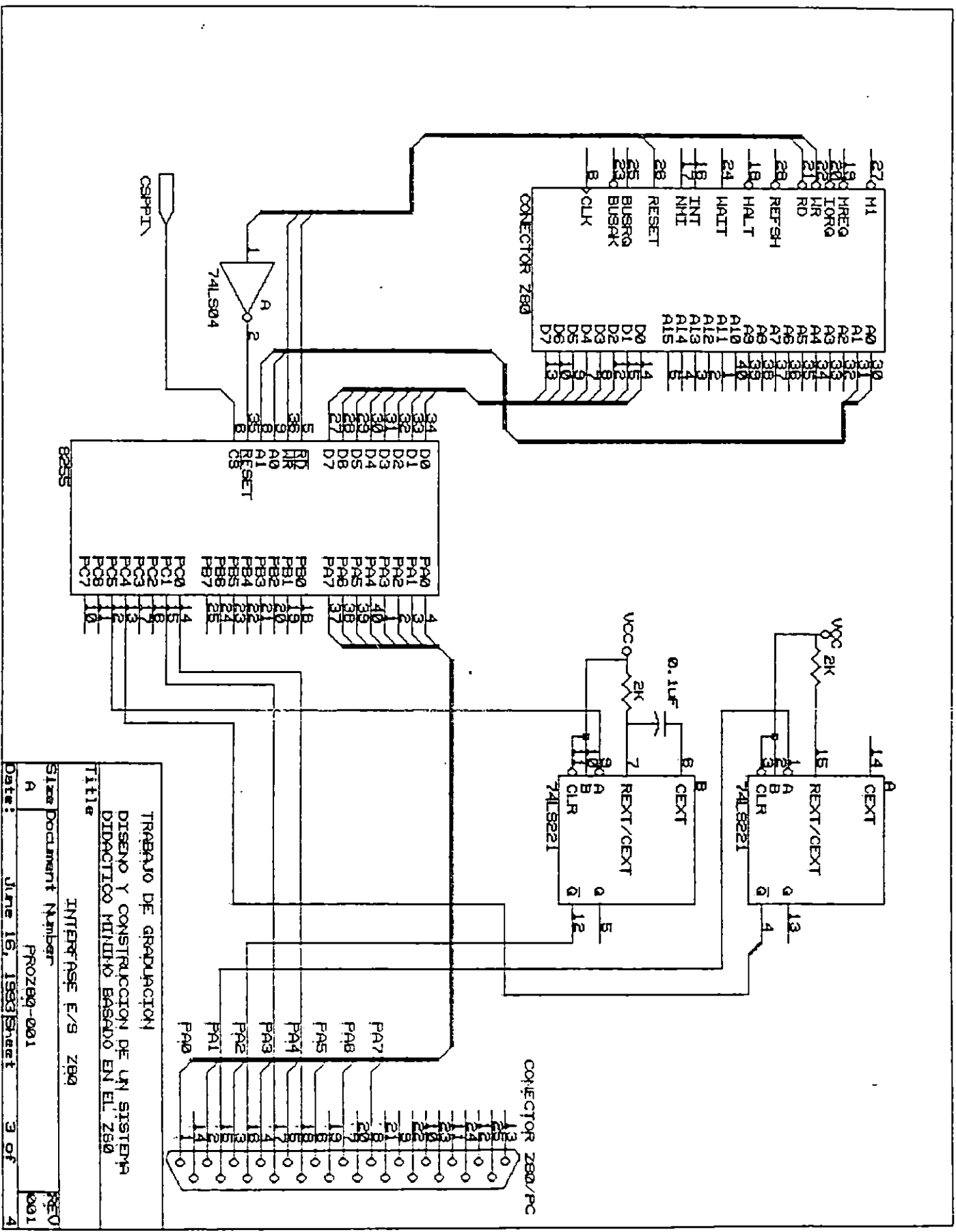
Básicamente el software del sistema consiste de dos partes, el software que estará residente en ROM en el módulo y el interfase de usuario desde la computadora. A continuación se describen los algoritmos de tales programas.

A2.2 Algoritmo de la interfase de usuario.

Antes de utilizar esta interfase es necesario haber creado el código fuente en lenguaje assembly del Z80 desde un editor de texto. El flujograma completo del proceso previo al uso de la interfase aparece diagramado en la sección 1.5 del capítulo I, en la Fig. 1.6.

El algoritmo de la interfase de usuario es el siguiente:

1. Desplegar presentación del software.
2. Desplegar menú de las siguientes opciones:
ARCHIVOS, EDITAR, RUN y DEBUG.
3. Seleccionar opción
4. En caso de seleccionar:
 - 4.1 ARCHIVOS
 - 4.1.1 Desplegar menú de las siguientes subopciones:
Cargar, Directorio, Ver, Sistema O. y Salir
 - 4.1.2 Seleccionar subopción
 - 4.1.3 En caso de seleccionar:
 - 4.1.3.1 Cargar: - Entrar nombre de archivo
- Enviar a memoria RAM/Z80
- Regresar a paso 2
 - 4.1.3.2 Directorio - Mostrar directorio
- Retornar a paso 2
 - 4.1.3.3 Ver: - Editar archivo cargado
- Retornar a paso 2
 - 4.1.3.4 Sistema O.: - Salir temporalmente al ambiente del DOS
- Retornar al paso 2
 - 4.1.3.5 Salir: - Terminar y salir al DOS
 - 4.2 EDITAR
 - 4.2.1 Desplegar menú de las siguientes subopciones:
Registros y Banderas
 - 4.2.2 Seleccionar subopción
 - 4.2.3 En caso de seleccionar:
 - 4.2.3.1 Registros: - Visualizar los registros internos del Z80.
- Retornar al DOS.
 - 4.2.3.2 Banderas: - Visualiza el estado de las banderas del Z80



TRABAJO DE GRADUACION
 DISEÑO Y CONSTRUCCION DE UN SISTEMA
 DIDACTICO MINIMO BASADO EN EL Z80

INTERRUPTOR E/S Z80

Size Document Number PROZ80-001
 Date: June 16, 1983 Sheet 3 of 4

- Retornar al paso 2

4.3 RUN

4.3.1 Desplegar menú de las subopciones siguientes:
Trazar y/ Correr

4.3.2 Seleccionar subopción

4.3.3 En caso de seleccionar:

4.3.3.1 Trazar: - Correr paso a paso el programa.
desplegando el estado del
Z80

- Retornar al paso 2.

4.3.3.2 Correr: - Correr el programa residente
en RAM del Z80.

- Retornar al paso 2.

4.4 DEBUG:

4.4.1 Desplegar en la PC menú de la siguiente
subopción: Breakpoints

4.4.2 Seleccionar opción

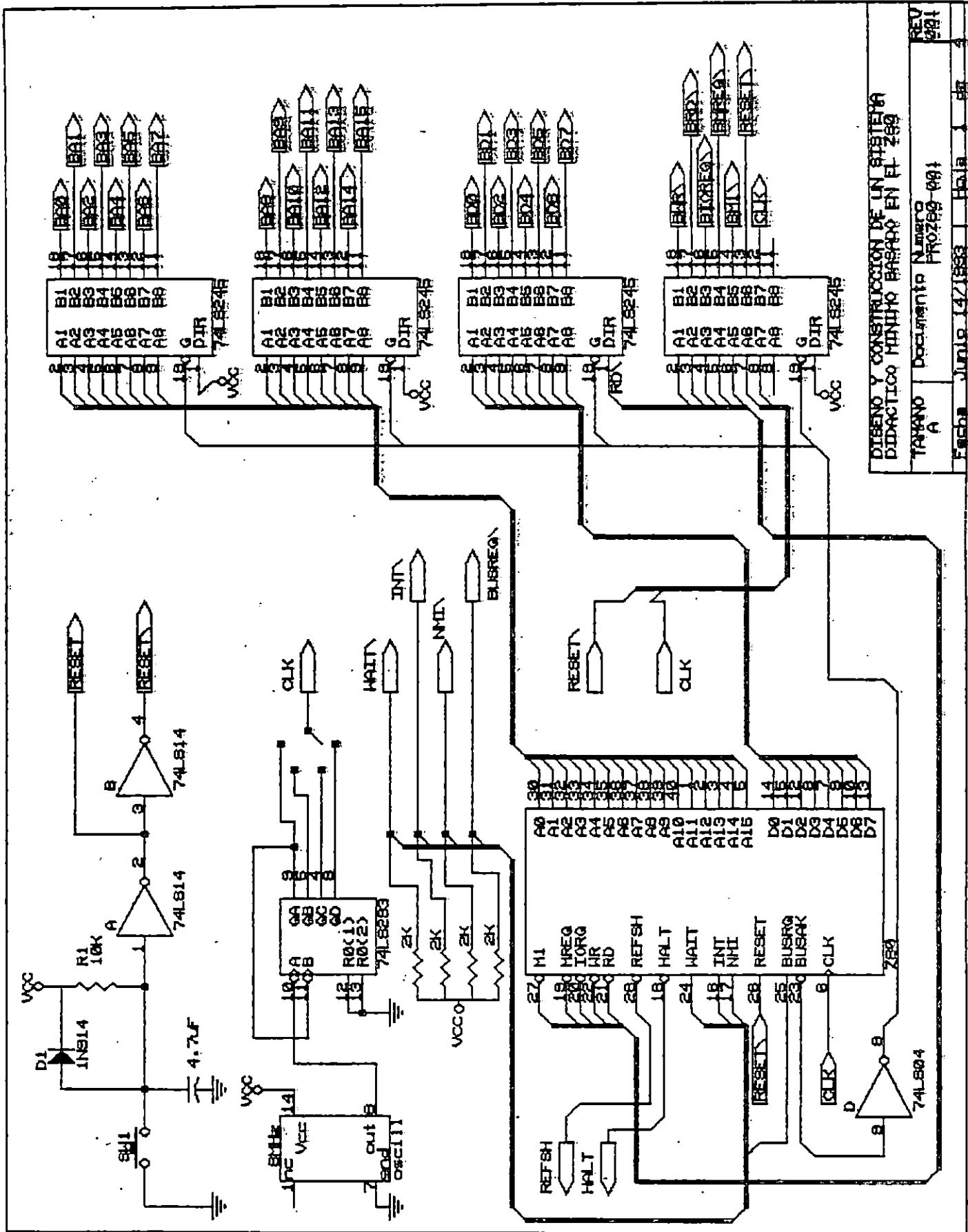
4.4.3 Fijar puntos de ruptura que defina el usuario

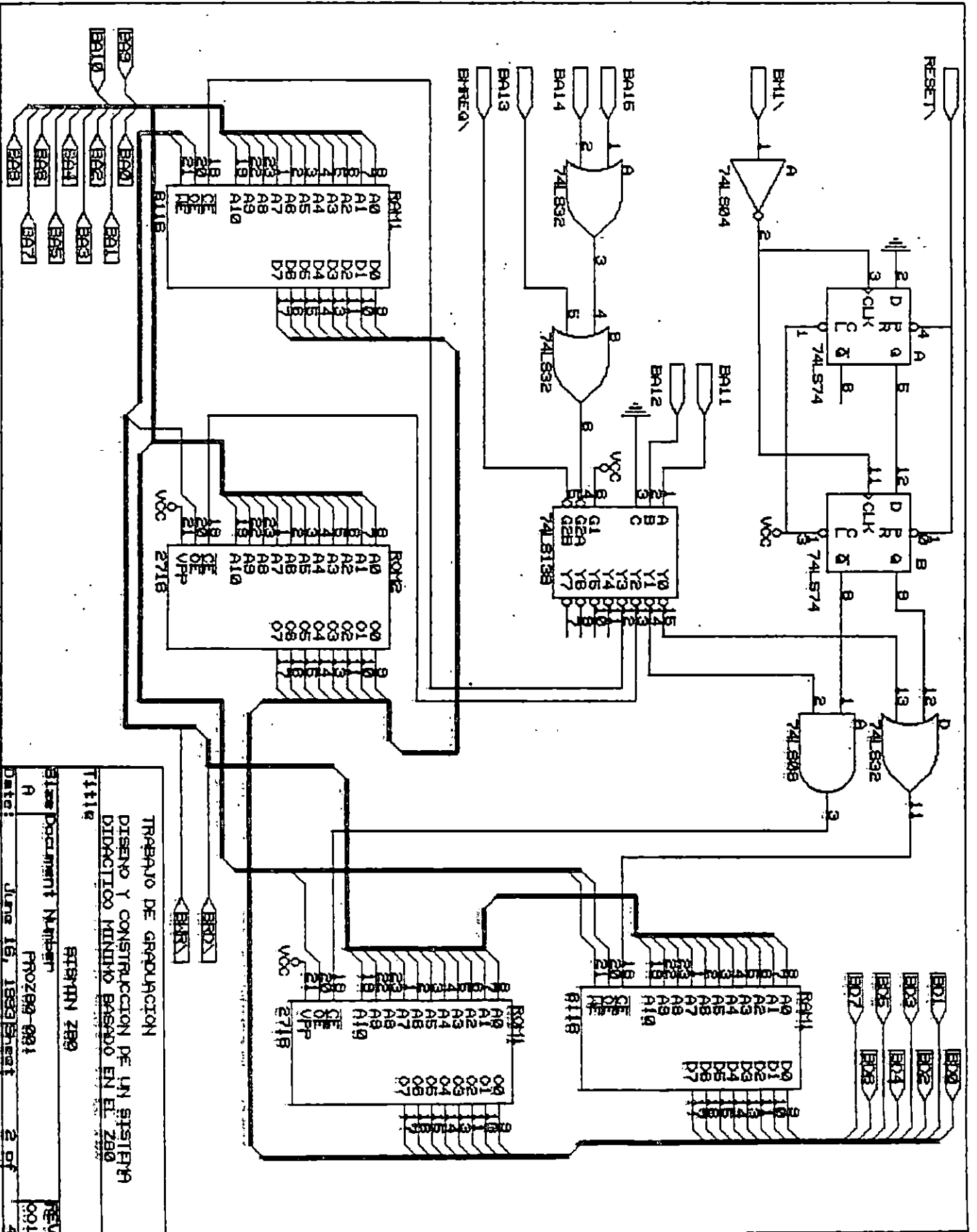
4.4.4 Retornar al paso 2

A2.3 Algoritmo del software residente en el módulo

El siguiente algoritmo define basicamente al procesador de comandos implementado en memoria ROM del sistema. El programa en esencia consiste de las rutinas de comunicación y el interpretador de la información transmitida desde la PC.

1. Definir constantes empleadas
2. Configurar puerto en modo diálogo
3. Ejecutar rutina de comunicación
4. Si en el paso 3 se detecta información útil continuar, sinó regresar al paso 3
5. Interpretar información recibida en paso 3
6. Ejecutar tarea interpretada en paso 5





TAREA DE GRUPO

DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DIDACTICO MINIMO BASADO EN EL Z80

1111

STB-THN Z80

PROZ-89-081