

UNIVERSIDAD NACIONAL DE INGENIERIA  
FACULTAD DE INGENIERIA ELECTRICA Y ELECTRONICA



*"Sistema Mínimo de Propósito  
General usando el  
Microprocesador Intel 8085"*

TITULACION POR EXAMEN PROFESIONAL  
Para Optar el Título Profesional de  
**INGENIERO ELECTRONICO**

*Mauricio Postigo Malaga*

*Promoción 1988 - 1*

**LIMA - PERU - 1995**

**A mi esposa e hija  
Norma y Alejandra**

## S U M A R I O

En los tiempos actuales encontramos muchos equipos que usan la tecnología de los microprocesadores para sus diseños por la flexibilidad y las grandes ventajas que nos brindan.

El objetivo del presente trabajo, es poner al alcance de los estudiantes e ingenieros electrónicos un módulo de propósito educativo; que les permita conocer, experimentar y diseñar sistemas basados en microprocesadores.

El módulo cuenta con el microprocesador Intel 8085, memoria RAM, EPROM, dispositivos de entrada salida, display, teclado, conversores análogo/digital y digital análogo, transmisión de datos seriales y un programa que facilita el uso del módulo. También nos da una buena base, para luego experimentar con los populares microcontroladores que integran toda esta arquitectura en un chip.

A diferencia de otros módulos; no solo nos enseña a entrenar en la programación; sino que orienta al usuario en todos los recursos; para que desarrolle sistemas basados en microprocesadores sencillos o complejos en un solo módulo.

El módulo fue usado para desarrollar varios proyectos universitarios, con resultados satisfactorios.

**" SISTEMA MINIMO DE PROPOSITO  
GENERAL USANDO EL MICROPROCESADOR  
INTEL 8085 "**

**E X T R A C T O**

**SISTEMA MINIMO DE PROPOSITO GENERAL USANDO EL  
MICROPROCESADOR INTEL 8085**

**MAURICIO POSTIGO MALAGA**

**PARA OPTAR EL TITULO PROFESIONAL DE INGENIERO  
ELECTRONICO**

**UNIVERSIDAD NACIONAL DE INGENIERIA**

**FACULTAD DE INGENIERIA ELECTRICA Y ELECTRONICA**

**LIMA-PERU**

**1995**

\*\*\*\*\*

**CAPITULO I**

Planteamiento del problema, se presentan los problemas que surgen al realizar un modulo de entrenamiento y se definen de objetivos del modulo.

**CAPITULO II :**

Solucion del problema, se describen los metodos de solucion, se definen las características del modulo de acuerdo al metodo elegido y se describe el diagrama de bloques que soluciona los problemas planteados.

**CAPITULO III :**

Desarrollo del sistema, descripción del microprocesador Intel 8085, desarrollo del diseño del bloque del microprocesador Intel 8085 y generación de los buses de

datos de direccion y de control, desarrollo del diseño del bloque de memoria, desarrollo del diseño del bloque de visualizacion de los buses, diseño del circuito de la funcion paso a paso, descripcion de los dispositivos perifericos, descripcion del PPI 8255, tipos de configuracion de puertos, diseño del bloque del teclado, descripcion del bloque de conversores analogo digital y digital analogo. Desarrollo del software del sistema, definicion del menu principal, descripcion del diseño del programa de los submenu A, B, C, D, E y F. Definicion del mapa de memoria y de puertos de entrada salida, posicion de memoria reservada por el modulo.

#### **CAPITULO IV :**

Instrucciones a seguir para la puesta en marcha del modulo.

#### **CAPITULO V**

Por las limitaciones de modulo se hace necesario desarrollar el modulo en dos etapas siguientes: la interconexion del modulo con la PC y su interaccion en tiempo real con la PC.

## INDICE

INTRODUCCION	01
CAPITULO I	
PLANTEAMIENTO DEL PROBLEMA	06
CAPITULO II	
SOLUCION DEL PROBLEMA	10
2.1 Tipeado del programa .....	11
2.2 Ejecución del programa en modo RUN O STEP .....	12
2.2.1 Modo RUN .....	12
2.2.2 Modo STEP .....	13
2.3 Conversión de datos analógico-digital y digital análogo .....	13
2.4 Sistema modular .....	14
2.4.1 Bloque del microprocesador Intel 8085 .....	14
2.4.2 Bloque de memoria .....	14
2.4.3 Bloque del sistema paso a paso .....	15
2.4.4 Bloque de transmisión-recepción de datos seriales	15
2.4.5 Bloque de interfase de entrada-salida	15
2.4.6 Bloque de conversión analógico-digital y digital- análogo .....	16
2.4.7 Bloque del teclado .....	16
2.4.8 Bloque de visualización	16
CAPITULO III	
DESARROLLO DEL SISTEMA	19
3.1 El microprocesador Intel 8085 .....	19

## VIII

3.1.1 Diagramas de patillas del 8085 .....	19
3.1.2 Arquitectura del 8085 .....	21
3.1.3 Modos de direccionamiento .....	24
3.2 Diseño del bloque del microprocesador Intel 8085.	25
3.2.1 Señales de reloj .....	26
3.2.2 Reseteado y puesta en marcha del procesador .....	27
3.2.3 Bus de direcciones y de datos .....	28
3.2.4 Bus de control .....	30
3.2.5 Señales de entrada de interrupción .....	31
3.2.6 Entrada serie y salida serie .....	34
3.3 Diseño de bloque de memoria .....	35
3.3.1 Construcción de la unidad de la memoria .....	36
3.3.2 Construcción de bloques de memoria de 8 KB .....	37
3.4 Bloque de visualización de los buses .....	38
3.5 Bloque del sistema paso a paso del 8085.....	39
3.5.1 La función READY .....	40
3.5.2 Circuito básico del modo paso a paso .....	40
3.6 Bloque de dispositivos entrada salida .....	42
3.6.1 Bloque de interfase de entrada salida .....	42
3.6.2 Bloque del teclado .....	47
3.6.3 Bloque de conversión de datos análogo digital y digital análogo .....	49
3.7 Desarrollo del software .....	50
3.7.1 Menú principal .....	51
3.7.2 Submenú de dirección de inicio de programa .....	
3.7.3 Submenú de tipeado de programa .....	
3.7.4 Submenú de lectura de programa .....	56
3.7.5 Submenú de lectura de registros .....	58

3.7.6 Submenú de ejecución de programa .....	60
3.7.7 Submenú de borrado de programa .....	60
<b>CAPITULO IV</b>	
<b>PUESTA EN MARCHA DEL MODULO .....</b>	<b>84</b>
<b>CAPITULO V</b>	
<b>DESARROLLO FUTURO .....</b>	<b>88</b>
5.1 Interconexión del sistema mínimo con la PC .....	88
5.2 Interacción del sistema con la PC en tiempo real..	89
<b>CONCLUSIONES .....</b>	<b>90</b>
<b>APENDICE A</b>	
Data Sheet del microprocesador Intel 8085 .....	96
<b>APENDICE B</b>	
Listado de programa del modulo .....	104
<b>BIBLIOGRAFIA .....</b>	<b>117</b>

## INTRODUCCION

Desde hace ya una década y media los equipos de diferentes ramas vienen usando la tecnología de los sistemas basados en microprocesadores, pero solo como equipos terminados de aplicación específica; en la que el usuario solo se limita a instalarlo y conectarlo. El sistema mínimo de propósito general usando el microprocesador Intel 8085 diseñado en el presente trabajo tiene como objetivo principal dar a conocer la tecnología de los sistemas basados en microprocesadores, por lo tanto es un módulo educativo que servirá para entrenar a los ingenieros y/o estudiantes en el diseño de sistemas basados en microprocesadores, además de que podrá realizar sus propios diseños usando esta tecnología.

Los sistemas basados en microprocesadores minimizan el hardware y además le dan mucha flexibilidad al diseño para que puedan ser mejorados o modificados para su desarrollo futuro.

Fue utilizado el microprocesador Intel 8085 por ser uno de los procesadores de 8 bits más generales, fácil de usar y de bajo costo, así mismo integra varias funciones en su chip ahorrando hardware. Su implementación en su sistema mínimo es muy sencillo ya que necesita solo tres chip como lo veremos mas adelante en el desarrollo del sistema,

además es fabricado por la compañía Intel de reconocida trayectoria en microprocesadores y microcontroladores asegurando que el conocimiento del 8085 nos va a servir para conocer otros dispositivos de la misma marca.

Con la aparición de los microcontroladores, que son sistemas mínimos de microprocesador en una sola pastilla, es decir que internamente cuentan con un microprocesador, una memoria EPROM, una memoria RAM, líneas de entrada/salida y temporizadores, el presente trabajo se hace aún más importante, ya que se debe conocer bastante bien los sistemas mínimos discretos para luego usar los microcontroladores. No se pretende competir con ellos, sino por el contrario, es una herramienta preliminar para conocer, experimentar y aplicarlo. Entonces el diseñador en la primera etapa para todas las pruebas lo haría en el sistema con el 8085 discreto, con chips de memoria y dispositivos entrada/salida en chips aparte; una vez corregido y depurado pasaría a la segunda etapa usando ya el microcontrolador, es decir nuestro sistema nos va a servir para desarrollos futuros con los microcontroladores. La mayoría de los microcontroladores usan microprocesadores de 8 bits; al igual que el microprocesador Intel 8085 y su arquitectura interna e instrucciones son parecidas, como por ejemplo los microcontroladores de la familia INTEL MCS-48XX (8021, 8022, 8048, 8748, 8355, 8755) o la familia INTEL MCS-51 (8031, 8032, 8052) estas familias conservan al microprocesador de 8 bits y lo que varía en cada modelo son las capacidades de las memorias RAM, EPROM, cantidad de

registros y de puertos de entrada salida; además adiciona temporizadores o conversores analógico digital.

Hay otros módulos de entrenamiento como los que usan el microprocesador de ZigLog Z-80, que están mas orientados a entrenar al estudiante en cuanto a la programación, este módulo diseñado se orienta mas a entrenar al estudiante en el diseño de sistemas basados en microprocesador para la solución de un problema. El módulo tiene un arquitectura muy abierta, para que el usuario use solo lo que necesite y cuente con lo que necesite.

El usuario del módulo podrá realizar las siguientes tareas:

- Típear su programa, correrlo y comprobar si funciona correctamente.
- Poder ubicar errores del programa y depurarlo, ya que el módulo cuenta con la opción de correr el programa paso a paso así como la visualización de registros del microprocesador Intel 8085.
- Disponer de entradas y salidas analógicas de 0 a 5 voltios, ya que se cuenta con conversores analógico-digitales y digital-análogo.
- Disponer de entrada y salida de datos seriales en señal de 20 mA o TTY intercambiable con un switch.
- Disponer hasta 64 KB de memoria y poder intercambiar entre memorias EPROM y RAM.
- Depurar su hardware y software para su posterior implementación usando microcontroladores.
- Disponer de 6 puertos de entrada salida para propósito

general.

- Acondicionar el sistema para su aplicación específica por su característica "transparente", modular y flexible.

El módulo se diseñó de tal forma que tenga como característica principal su "transparencia". Defino la característica "transparente" como un módulo que el usuario lo pueda usar como si hubiera implementado su propio circuito; usando todas las características del microprocesador, el hardware que requiere y acondicionarlo a su aplicación específica con mínimas limitaciones; y que el software del sistema no altere el software del usuario y por el contrario le haga las cosas más fáciles. También es un módulo que se acondiciona a cualquier aplicación, sencilla o compleja.

La limitación del presente trabajo está en correr el programa con la opción paso a paso que la realizamos por hardware y cada paso es un ciclo de máquina y no una instrucción. Esta limitación la superamos parcialmente introduciendo un programa de visualización de registros para ver los resultados después de una o varias instrucciones, también tiene limitación en la frecuencia de las señales analógicas que va a convertir en digital que es de 700 Hz. Cuando transmitimos o recibimos datos en forma serial lo podemos hacer hasta 1000 bps.

En un desarrollo futuro como segunda etapa será el conectarlo con la PC y poder realizar un software amigable,

tipear el programa en nemonico y también interactuar en tiempo real el microprocesador Intel 8085 con la PC.

La metodología usada en el presente trabajo fue siguiendo los siguientes pasos.

- 1.- Estudio del microprocesador Intel 8085.
- 2.- Definición de objetivos y características del sistema.
- 3.- Diseño e implementación del sistema mínimo del 8085.
- 4.- Pruebas de laboratorio del sistema.
- 5.- Elaboración del software del sistema.
- 6.- Pruebas de laboratorio con el software del sistema.
- 7.- Depuramiento y puesta a punto.
- 8.- Comprobación del sistema en aplicaciones específicas.
- 9.- Correcciones y mejoras finales.
- 10.- Definir los desarrollos futuros del sistema.

Entonces el presente trabajo tiene como propósito poner al alcance de los diseñadores un módulo de entrenamiento en el diseño e implementación de sistemas basados en microprocesadores usando el microprocesador Intel 8085, es decir que servirá como módulo educativo, para la realización de sistemas basados en microprocesadores.

## C A P I T U L O I PLANEAMIENTO DEL PROBLEMA

En la introducción hablamos sobre la necesidad de usar sistemas basados en microprocesadores; para dar solución a problemas de diferentes ramas, así como las ventajas que estos sistemas ofrecen.

Surgida esta necesidad, el estudiante debe familiarizarse con los sistemas basados en microprocesadores; por tanto debemos de contar con un módulo de entrenamiento para este fin.

Ahora, para implementar un sistema basado en microprocesador tenemos dos alternativas:

- La primera, es usar un sistema con un microprocesador de 8 bits como el 8085 o similar. Los sistemas con procesador de 16 bits se hacen muy complicados para realizar su implementación, por la cantidad de chips que lo acompañan y la complejidad de ellos, un microprocesador de 8 bits es más manejable y con el microprocesador Intel 8085 aún mas.

- La segunda alternativa, es usar microcontroladores de 8 bits como son los de la familia de INTEL MCS-48 y MCS-51 o microcontroladores de Motorola de la serie MC68XXX que son de propósito mas específico.

Si no se ésta bien familiarizado con los

microcontroladores, el estudiante o ingeniero deberá conocer y usar primero un sistema discreto con microprocesadores; con este conocimiento recién podrá experimentar, conocer y si queremos ampliar el sistema con los microcontroladores.

El objetivo de este trabajo es poder elegir la primera alternativa, además de poder obtener conocimiento y experiencia para la segunda.

Los problemas que surgen al realizar un módulo de entrenamiento con un sistema mínimo de microprocesadores son los siguientes:

- Cada vez que queremos probar un programa hay que grabar una EPROM.
- Una vez quemada la EPROM con el programa, no funciona el sistema, esto debido a errores del programa o porque esta mal grabada la EPROM.
- No funciona el sistema, y no se sabe si es el hardware o el software.
- No podemos ubicar ni corregir fácilmente el problema, ya que si es el programa debemos corregir y volver a quemar la EPROM.
- No podemos depurar o mejorar el programa en poco tiempo.
- Las pruebas iniciales se harán con riesgo de quemar algún componente; al no conocer el comportamiento del sistema y las señales de entrada.
- No podemos comprobar los resultados parciales.

- Cuando usamos retardos de tiempo no podemos hacer pruebas ya que hay que quemar nuevamente el programa.
- No podemos transmitir o recibir datos seriales, para interconexión de dispositivos externos.
- No sabemos exactamente cuanto de memoria necesitamos para determinar el mapa de memoria más óptimo.
- No podemos acondicionar el módulo para diferentes aplicaciones por ser muy rígido, además de obligarnos a variar sustancialmente nuestro diseño.
- No podemos visualizar las señales de control.
- No podemos comprobar los ciclos de máquina de cada instrucción.
- El módulo usa funciones y patillas del microprocesador que luego el usuario no podrá disponer para su aplicación.
- El software del sistema interfiere con el software del usuario.
- No podemos entrar señales analógicas, provenientes de sensores y/o transductores.

Por lo tanto, necesitamos de un módulo basado en microprocesador con mínimas limitaciones, que sea transparente, flexible, modular y fácil de acondicionar.

También deberá contar con conversor análogo-digital, transmitir y recibir datos seriales, permitir al usuario contar con todas las características y potencia del microprocesador y lo podrá usar en aplicaciones sencillas o complejas.

Para probar un programa o una parte de el, deberá cargarlo en memoria RAM y comprobar sus resultados de esta manera podrá corregir y depurar el programa.

En conclusión, un módulo que sea como un probador dentro de algunas limitaciones, en donde podamos implementar fácilmente nuestra aplicación.

## C A P I T U L O II SOLUCION DEL PROBLEMA

Para solucionar los problemas planteados en el capítulo I y cumplir con las características descritas el módulo diseñado debe realizar las siguientes tareas

- 1.- El módulo diseñado deberá permitir al usuario tiplar su programa y grabarlo en la memoria RAM.
- 2.- Poder correr el programa en dos modos:  
En modo RUN, es decir, todo el programa viendo su resultado final, y en modo STEP (paso a paso), para poder verificar su funcionamiento, encontrar los errores y corregirlos.
- 3.- Disponer de conversores análogo-digital y digital análogo.
- 4.- Verificar los tiempos de retardo y poderlos corregir al ver su comportamiento con los dispositivos de entrada salida.
- 5.- El módulo será diseñado en forma modular, para darle flexibilidad y transparencia, y poder ser usado en diferentes aplicaciones.
- 6.- Poder monitorear todas las señales del bus de control.
- 7.- El módulo podrá ser reinicializado (reseteado).
- 8.- El módulo mediante un programa hecho por el usuario podrá transmitir y recibir datos en forma serial.

- 9.- El usuario dispone de seis puertos de entrada-salida de propósito general.
- 10.- El módulo servirá para las pruebas iniciales y depuramiento de un sistema con microprocesadores.
- 11.- Disponer de cinco interrupciones hardware para diferentes prioridades.
- 12.- Borrar porciones de memoria RAM.
- 13.- Visualizar los registros internos del microprocesador Intel 8085.
- 14.- Leer posiciones de memoria.

A continuación, explicaremos brevemente como el sistema puede realizar estas tareas.

### 2.1 Teclado del programa

El módulo cuenta con un teclado de 16 teclas, con los números hexagesimales y 3 teclas de propósito especial. Mediante el integrado 74922 que es un decodificador de 16 a 4, conectado al bus de direcciones y datos mediante un dispositivo periférico PPI (Programable Peripheral Interfase).

Por medio de este teclado, el usuario dispondrá de 6 funciones que le permitirán: elegir la dirección de memoria donde copiará el programa, tiplar el programa, ejecutar el programa, corregir el programa, visualizar los datos finales de los registros, borrar porciones de memoria, y leer posiciones de memoria.

El usuario entonces podrá tiplar su programa en la

dirección de memoria RAM que decida; tipleará los dos números hexagesimales de datos que corresponden a la dirección.

El usuario podrá visualizar las direcciones y datos que esta tipleando mediante un display de seis dígitos, cuatro de ellos mostrarán las direcciones y dos los datos. Este display esta conectado al dispositivo periférico PPI de propósito especial y un pequeño programa para realizar la visualización.

## 2.2 . Ejecución del programa en modo RUN o STEP

Una vez tipleado el programa el usuario podrá ejecutar el programa en dos modos: en modo RUN o en modo STEP. El módulo dispone de un switch para poder cambiar de modo.

### 2.2.1 Modo RUN

En este modo el programa se ejecutará completamente y dará un resultado final que podrá ser verificado en los registros internos del microprocesador Intel 8085 y/o en los dispositivos de salida, también los visualizamos en el display. Al activar la función RUN, el programa correrá desde la dirección de memoria elegida por el usuario al tiplear el programa; ya que el sistema carga el registro PC (Contador del Programa) con esta dirección.

### 2.2.2 Modo STEP

En este modo el usuario podrá ejecutar su programa paso a paso, para la verificación y detección de errores del programa. El microprocesador Intel 8085 dispone de hardware para la realización de este modo usando las patillas READY y HOLD. Cada paso es un ciclo de máquina, de esta manera el módulo también servirá para mostrar al usuario el funcionamiento de las instrucciones en cada ciclo de máquina. Cabe aclarar que en este modo cada paso es un ciclo de máquina y no una instrucción, el módulo dispone de una subrutina de visualización de registros que podemos hacerla correr después del programa, o parte de él o después de cada instrucción, pero con ciertas limitaciones.

### 2.3. Conversión de datos analógicos-digital y digital-análogo

El módulo dispone de un Conversor Analógico Digital (ADC) y un Conversor Digital Análogo (DAC); que están conectados a dos puertos del dispositivo periférico

(PPI). El ADC usado es el ADC-0808 que dispone de 8 entradas analógicas de hasta 5 voltios, un bus de control y datos compatibles con los buses del microprocesador; elegimos usarlo a la salida de un PPI para darle mayor flexibilidad al módulo de tal forma que el usuario vía programa pueda usar estos conversores direccionando simplemente los puertos.

El módulo dispone también de un DAC que puede ser multiplexado para tener varias salidas dependiendo de las necesidades del usuario.

## 2.4 Sistema modular

Para que el sistema sea flexible de ser usado en aplicaciones de diferente complejidad; el diseño del sistema debe ser modular, es decir debe acomodarse a cualquier configuración de esta forma modificamos sólo un bloque. Los bloques que forman parte del sistema se describen en la Figura 2.1, dichos bloques son los siguientes:

### 2.4.1 Bloque del microprocesador Intel 8085

En este bloque esta el microprocesador Intel 8085 con todas las conexiones necesarias de sus patillas y además los tres buses: el bus de direcciones, bus de datos y bus de control. Los tres buses están disponibles al usuario.

### 2.4.2 Bloque de memoria

El sistema tiene la capacidad de direccionar 64KB de memoria, que están distribuidas en 8 bancos de 8 KB cada uno. A su vez estos bancos de 8 KB se distribuyen en 4 bancos de 2KB cada uno, estos bancos pueden ser intercambiables entre memoria EEPROM y RAM; solo cambiando las pastillas correspondientes y un switch de control ( tienen que ser memorias RAM y EEPROM

compatibles en patillaje como la 2016 y 2716). De acuerdo a la aplicación se puede implementar la memoria requerida.

#### 2.4.3 Bloque del sistema paso a paso

En este bloque esta el circuito que hace posible que el programa pueda ser ejecutado paso a paso (modo STEP). En cada ciclo de máquina el procesador se mantiene en estado de espera, hasta que, con un pulsador pasemos al siguiente ciclo de máquina. De esta manera cada vez que se presiona el pulsador STEP corremos un ciclo de máquina, con esta característica podemos analizar las instrucciones que tienen diferentes cantidades de ciclos de máquina.

#### 2.4.4 Bloque de transmisión-recepción de datos seriales

En este bloque el usuario puede recibir o transmitir datos seriales. El usuario dispone de una salida serial y una entrada serie, estas envían y reciben señales en dos formas, una TTY o de 20 mA y otra V24; que son intercambiables mediante un switch.

#### 2.4.5 Bloque de interfase de entrada-salida

En este Bloque están las interfases a periféricos de entrada-salida, consta de tres Programable Peripheral Interfase (PPI) conectados al bus de datos, al bus de control y al decodificador de direcciones. Cada PPI

dispone de tres puertos de entrada-salida que pueden ser programados en tres modos como lo detallaremos en el siguiente capítulo. Un PPI es de propósito especial para las funciones propias del sistema y esta direccionado como memoria; los PPI restantes son de propósito general y están disponibles al usuario; uno direccionado como puerto y el otro como memoria.

#### 2.4.6 Bloque de conversión análogo-digital y digital-análogo

Los conversores ADC y DAC están disponibles al usuario como dispositivo periférico de entrada salida a través de un PPI, se eligió esta forma para darle mayor flexibilidad al sistema. Los datos son enviados o recibidos en un puerto a través de un programa que lo hace el usuario.

#### 2.4.7 Bloque del teclado

En este bloque esta la circuitería necesaria para el teclado, dispone del decodificador de 16 teclas 74922, unas puertas lógicas y los switchs del teclado. Las salidas del 74922 están conectadas al PPI de propósito especial usando cuatro líneas para el dato y una de control, que avisa que se ha presionado una tecla. Las tres líneas restantes son las teclas especiales. El sistema contiene un programa de menús para elegir opciones mediante el teclado.

#### **2.4.8 Bloque de visualización**

En este bloque están los display de 7 segmentos para visualizar las direcciones y datos. Los buses de direcciones y datos están en código hexagesimal y van a ser visualizados mediante la conexión de los display a un puerto del PPI de propósito especial y un pequeño programa. El desarrollo del diseño de cada módulo será visto con detalle en el Capítulo III.

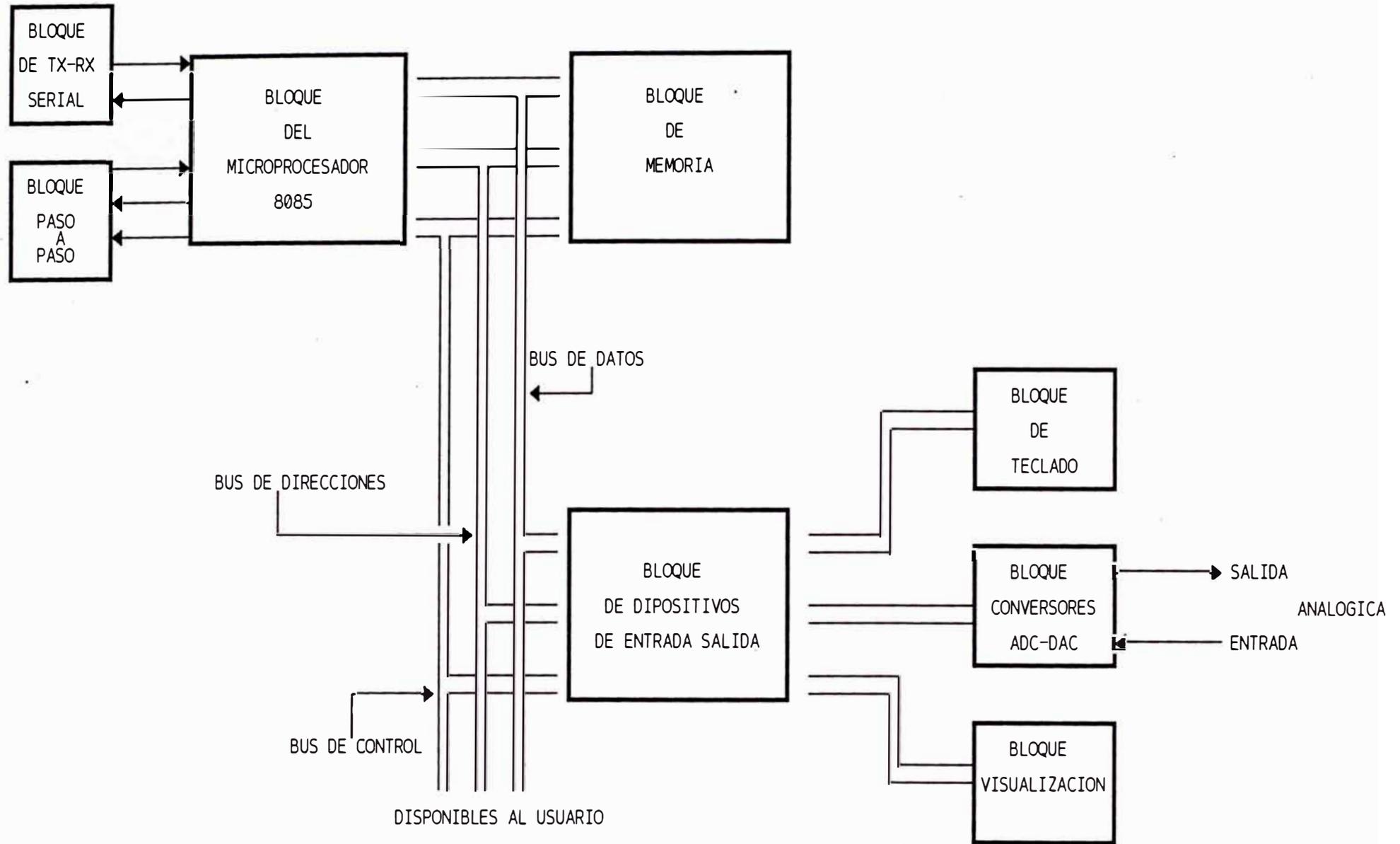


FIGURA 2.1 : DIAGRAMA DE BLOQUES DEL MODULO

## C A P I T U L O   I I I DESARROLLO DEL SISTEMA

### 3.1 El microprocesador Intel 8085

EL MICROPROCESADOR Intel 8085 es una versión mejorada del 8080, maneja palabras de datos de 8 bits, tiene 16 líneas de dirección y un puntero de pila de 16 bits. Integra el reloj, control del sistema y prioridad de las interrupciones en el microprocesador, reduciendo así el número de circuitos integrados utilizados. El 8085 opera con una sencilla fuente de alimentación de 5 voltios. El microprocesador 8085 utiliza las mismas instrucciones que el 8080, haciendo así sus programas compatibles, el 8085 tiene dos instrucciones más para aprovechar las características del hardware. En un sistema mínimo como el 8085, solamente es necesario tres circuitos integrados: el 8085, una unidad de memoria (ROM/RAM) y un dispositivo de entrada/salida.

#### 3.1.1 Diagramas de patillas del 8085

El microprocesador INTEL 8085, tiene un encapsulamiento de doble línea (DIP), de cuarenta patillas. El diagrama de patillas se muestra en el apéndice A. Hay una versión ligeramente revisada del microprocesador original 8085 que es el 8085A. Versiones posteriores incluye dispositivos de mas alta

velocidad como el 8085A-2, 8085AH, 8085AH-2 y el 8085AH-1. La descripción del diagrama de patillas es el siguiente:

AD0-AD7	Bus de direcciones/ datos son patillas Bidireccionales, y tres estados.
A8-A15	Bus de direcciones, son salidas de tres estados.
ALE	Habilitación del Latch de direcciones, como las 8 direcciones meno significativas están multiplexadas con los 8 bits de datos, esta patilla nos indicará cuando son datos, y cuando Direcciones. ALE es una patilla de Control.
RD	Control de Lectura, es una salida de tres estados.
WR	Control de escritura, es una salida de tres estados.
IO/M	Nos indica si hay salida/entrada de puertos o memoria, es una salida de tres estados.
S0, S1	Indicadores de estado de Bus, son salidas.
READY	Petición de estado de espera, es entrada.
SID	Entrada de datos serie, es una entrada.
SOD	Salida de datos serie, es una salida.
HOLD	Petición de mantenimiento, es una entrada.

HLDA	Conocimiento de mantenimiento, es una salida
INTR	Petición de interrupción, es una entrada
TRAP	Petición de interrupción no enmascarable, es una entrada.
RST 5.5,	
RST 6.5,	
RST 7.5	Peticiones de interrupciones Hardware vectorizadas, son entradas.
INTA	Conocimiento de Interrupción, es una salida.
RESET IN	Reinicialización del sistema, es una entrada.
RESET OUT	Reinicialización de los periféricos, es activa cuando se ha producido un reset.
X1, X2	Conexiones al cristal o RC.
CKL	Señal del reloj, es una salida.
Vcc, Vss	Alimentación tierra.

### 3.1.2 Arquitectura del 8085

La arquitectura interna del microprocesador Intel 8085 se reproduce en la Figura 3.2. El 8085 tiene un contador de programa y un Latch de direcciones de 16 bits que alimenta al bus de direcciones dedicado ( A15-A8 ) y al bus de direcciones/datos de doble propósito ( AD7-AD0 ). Los datos Paralelos entran y dejan el microprocesador vía el bus de direcciones/datos multiplexado. El bus de

direcciones/datos transmite una dirección cuando la línea de control ALE esta en ALTA y de datos cuando la línea ALE esta en BAJA. Las entradas o salidas de datos a la unidad se realizan a través del bus de datos interno de 8 bits. Los datos pueden fluir desde el bus de datos interno al acumulador de 8 bits o a los registros temporales, señalizadores, registro de instrucción, unidad de control de interrupción, unidad de control de E/S serie. Las entradas y salidas serie (SID y SOD) se muestran en la parte superior derecha. Las entradas de interrupción (INTR, RST 5.5, RST 6.5, RST 7.5 y TRAP) y la salida INTA (conocimiento de interrupción) se muestran en la parte superior izquierda. La Unidad Aritmética Lógica (ALU) esta alimentada por dos registros de 8 bits (acumulador y registro temporal). Los Flip-Flops señalizadores tienen cinco indicadores de status. El registro de instrucción alimenta al decodificador de instrucciones el cual interpreta la instrucción actual y determina el microprograma a seguir. El decodificador de instrucciones entonces instruye a las secciones de control y temporización sobre la secuencia de eventos que deben realizarse. La sección de temporización y control coordina las acciones de procesador y los periféricos.

El microprocesador Intel 8085 tiene 8 registros direccionables de propósito general de 8 bits. Seis de estos registros pueden utilizarse como registros de 8

bits o como registro pares de 16 bits. Los registros del microprocesador Intel 8085 son:

- El Acumulador (A), foco de todas las operaciones y es de 8 bits.

Los registros de propósito general BC, DE y HL pueden ser utilizados como seis registros de 8 bits o como tres registros de 16 bits dependiendo de la instrucción que se vaya a ejecutar. El registro HL (llamado Puntero de datos por Intel) puede ser utilizado para apuntar direcciones. Pocas instrucciones utilizan a los registros BC y DE como punteros de dirección, ya que normalmente se utilizan como registros de propósito general.

El Contador de Programa (PC), siempre señala la posición de memoria de la siguiente instrucción que se va a ejecutar, siempre contiene una dirección de 16 bits.

- El Puntero de Pila (SP), es un puntero de dirección de propósito especial que siempre apunta a la parte superior de la pila de la RAM. Es un registro de 16 bits.

El registro de señalizadores, contiene 5 señalizadores de 1 bit que contienen información del estado de la CPU.

Estos señalizadores son utilizados por instrucciones de bifurcación condicional llamada y vuelta de subrutina. El registro de señalizadores esta compuesta por los siguientes bits:

B7 = Señalizador de signo (S)  
B6 = Señalizador de cero (Z)  
B5 = No usado  
B4 = Señalizador de arrastre auxiliar (AC)  
B3 = No usado.  
B2 = Señalizador de Paridad (P).  
B1 = No usado.  
B0 = Señalizador de arrastre (CY).

### 3.1.3 Modos de direccionamiento

El microprocesador Intel 8085 para sus instrucciones utiliza cinco modos de direccionamiento, estos son :

- Direccionamiento Implicado

Son cuando el direccionamiento de la instrucción es inherente a la función de la instrucción, son instrucciones de un Byte.

- Direccionamiento de registro

Son cuando en las instrucciones se especifican la fuente del operando que es un registro, y la operación que va a realizar, son instrucciones de un byte.

- Direccionamiento inmediato

Las instrucciones que utilizan direccionamiento inmediato tienen los datos inmediatamente a continuación del código de operación en la memoria de programa, son instrucciones de dos bytes y cuando cargamos registros pares son de tres bytes.

- Direccionamiento directo

Son instrucciones de tres bytes cuyo formato es el siguiente El Byte 1 contiene el código de op para la

instrucción de direccionamiento directo. El byte 2 contiene el byte de orden inferior de la dirección del operando. El byte 3 contiene el byte de orden superior de la dirección del operando.

- Direccionamiento indirecto de registro

Son las instrucciones que referencian la memoria utilizando el contenido de un registro para señalar la dirección del operando.

- Modos de direccionamiento combinados

Es la combinación de algunos de los modos anteriores, como por ejemplo, la instrucción CALL combina el direccionamiento directo e indirecto de registro.

### 3.2 Diseño del bloque del microprocesador Intel 8085

El bloque del microprocesador Intel 8085, es el conjunto de dispositivos (chips, resistencias, capacitores, etc) externos a la pastilla del microprocesador, que son necesarios para hacer funcionar al 8085. Para empezar a diseñar este módulo debemos conocer como es que trabaja el 8085 internamente. La velocidad con que trabaja el procesador depende de la frecuencia del reloj; el programa es cargado en memoria en forma binaria. El 8085 entiende 93 diferentes instrucciones que pueden ser de 1, 2 ó 3 bytes. Estas instrucciones son procesadas de la forma siguiente:

1.- Transfiere el primer byte al registro de instrucción (la dirección de memoria es dada por el PC).

2.- Decodifica la instrucción.

3.- El 8085 pregunta si es una instrucción de un byte, si es afirmativo pasa al paso 6, y si es negativa pasa al paso siguiente.

4.- El 8085 pregunta si es una instrucción de 2 bytes, si es afirmativa incrementa el PC y se va al ciclo de fetch de búsqueda del operando. Si es negativa se va al paso 5.

5.- Incrementa el PC y se va al ciclo fetch de búsqueda del primer operando. Incrementa el PC y se va al ciclo fetch de búsqueda del segundo operando.

6.- Incrementa el PC y luego ejecuta la instrucción. Los pasos 1 al 5 constituyen la fase fetch de la instrucción y el paso 6 la fase de ejecución.

### 3.2.1 Señales de reloj

El microprocesador Intel 8085 tiene internamente un clock oscillator; la frecuencia es derivada de un cristal de cuarzo conectado externamente de 4 MHz. Esta entra a un divisor y un trigger smith para cuadrar la señal del reloj. Todos los procesos de la CPU son corridos con un reloj de 2 MHz., el cual actúa como reloj del sistema. El microprocesador Intel 8085 dispone de una salida (CLK-OUT) de este reloj para conexión a los periféricos que lo requieran. Sólo es necesario conectar el cristal a las patillas 1 y 2 del 8085 y de la patilla 2 un capacitor a tierra. En la figura 3.3 muestra el diagrama de tiempos del ciclo de



Para el reseteo automático, cuando el switch de la fuente es prendido el capacitor C3 se carga vía R18. El 8085 se mantiene en HOLD hasta que el voltaje del capacitor alcance el voltaje threshold (cerca de 2.4 Voltios) del Schmitt trigger del 8085. Este corto retardo es necesario para permitir que la fuente se estabilice y estar seguro de una óptima operación. El reseteo, durante una operación se realiza pulsando el switch S2. Cuando S2 es presionado el capacitor se descarga a través de R14. Mientras S2 se mantenga presionado el CPU no opera, cuando se suelta S2, C3 es recargado vía R18 y la CPU vuelve a operar desde la dirección 0000H.

### 3.2.3 Bus de direcciones y datos

Cada salida del CPU 8085 sólo da niveles de entrada TTL, por lo tanto no es suficiente para sistemas CMOS. Por ejemplo, las salidas de dirección de memoria están conectadas en paralelo a todas las memorias. Esto significa que es necesario colocar amplificadores entre la CPU y las memorias-dispositivos E/S. Estos amplificadores incrementan la corriente de salida que permiten conectar a varias entradas; al mismo tiempo protegen al CPU de cortocircuitos producidos en el bus. Estos amplificadores llamados drives o buffers usaremos en el bus de direcciones y de datos.

Debido a que el CPU 8085 tiene multiplexadas las líneas de datos con las de direcciones (AD0 -AD7) es necesario colocar en estas patillas dispositivos

Latches (arreglo de 8 flip-flop), cuyo control es la línea ALE del 8085. Para el bus de direcciones usaremos el Latch tres estados y buffers 74LS374, uno para las líneas AD0-AD7 y otro para A8-A15 como se muestra en la figura 3.5 que es el diagrama completo del bloque del microprocesador. Cuando la entrada HOLD del 8085 es activada (nivel alto) significa que otro dispositivo externo (como por ejemplo el DMA) quiere utilizar los buses de direcciones y datos. Después de recibir una entrada HOLD la CPU completará las transferencias actuales de datos en los buses, entonces las patillas de direcciones, datos, RD, WR, e IO/M del 8085 se ponen en alta impedancia para no interferir en las transferencias de datos. Una salida HDLA indica al periférico que se ha recibido una petición de mantenimiento (HOLD) y que el CPU abandonará el control de los buses en el siguiente ciclo de reloj. Entonces usaremos la salida HDLA para conectarla a la entrada OE (output enable) de los 74LS374, que cuando es activa (nivel ALTO) sus salidas se van alta impedancia (tres estados). Cada vez que se quiera transferir las entradas del 74LS374 hacia la salida lo hacemos mediante su patilla CK (Flanco de subida). La patilla ALE del 8085 la conectamos mediante un inversor a la entrada CK del 74LS374.

Ya hemos transferido las direcciones a los integrados IC9 e IC10 controlados por la señal ALE, la CPU usa las 8 direcciones mas bajas como líneas de datos.

Entonces mediante esta patillas podemos enviar salidas de direcciones como también entradas y salidas de datos. Esto significa que debemos colocar un dispositivo que reconozca entre dirección y dato y además que sea bidireccional en el caso que sea dato. El dispositivo que cumple estas tareas es el 74LS245 que es un buffers tres estados bidireccional. Las líneas de datos AD0-AD7 que están conectadas al IC9 también las conectamos a las entradas del 74LS245 que es el IC8 como se muestra en la figura 3.5. La entrada OE del 74LS245 la conectamos a HLDA del 8085. La entrada DIR (patilla 1) del 74LS245 la conectamos a la salida RD del 8085 vía un inversor, para que cuando sea 1 la dirección sea del bus al CPU. La tabla de 74LS245 es la siguiente:

OE	DIR	EFEECTO
0	0	transfiere de CPU al bus
0	1	transfiere del bus al CPU
1	x	alta impedancia (tres estados)

#### 3.2.4 Bus de control

Generación de señales MEMR, MEMW, IOR y IOW.

Las señales MEMR, MEMW son las que van a controlar las operaciones de lectura escritura de la memoria, y las señales IOR y IOW controlan las operaciones de lectura escritura de los dispositivos de entrada/salida (PPI). Estas señales son parte del bus de control que van a ser generadas por las señales IO/M, WR y RD del CPU



podemos parar el programa que esta realizando para irse a otro programa y una vez terminado este regresar al programa donde estaba. Aquí unos ejemplos que usan interrupciones:

- Periféricos como impresoras, teclados, disqueteras, etc.

- El operador puede parar el programa

Las fallas en el proceso de control o máquinas pueden ser corregidos.

Una interrupción causa los siguientes efectos en el microprocesador

1.- La instrucción actual es completada.

2.- La dirección de la próxima instrucción es cargada en memoria (pila), para que el programa pueda continuar posteriormente.

3.- Es generada la dirección específica de la interrupción y el CPU salta a esta dirección y empieza a correr el programa de servicio de interrupción.

4.- Retorna a la dirección guardada en memoria del programa original.

El microprocesador Intel 8085 dispone de 5 entradas de interrupción con diferentes prioridades y direcciones, estas interrupciones son las siguientes:



### 3.2.6 Entrada serie salida serie

El microprocesador Intel 8085 puede recibir y enviar datos en forma serial sin necesidad de mucha circuitería usando dos instrucciones proporcionadas para este propósito. El 8085 dispone de un salida SOD para sacar los datos serialmente. También dispone de la entrada SID para recibir datos serialmente y ser cargados en el acumulador. Muchos dispositivos periféricos como terminales de video y teletipos se conectan serialmente con el microcomputador, de esta forma el 8085 tiene esta ventaja de poderse conectar fácilmente a estos dispositivos.

Para poder interactuar con los dispositivos entrada/salida, son requeridos niveles especiales en las líneas de transmisión. Estos niveles son de dos tipos: niveles de corriente y niveles de voltaje. Cada uno de estos tipos necesitan una interfase. Los niveles de corriente usan los teletipos y necesitan una interfase 20 mA current loop. Estos trabajan con niveles siguientes:

Logic 1 = circuito abierto (no corriente)

Logic 0 = corriente de 20 mA.

Los niveles de voltaje como el V24 (usado por terminales de video). Estos trabajan con los niveles siguientes:

Logic 1 = voltaje entre -3v y -25 voltios

Logic 0 = voltaje entre +3v y +25 voltios

Como la CPU 8085 trabaja con fuente de alimentación de 5 voltios es necesario convertir a señales de 20 mA o V24 mediante interfases. En la figura 3.5 también se muestran estas interfases con un switch P0 para poder cambiar cualquiera de estas interfases. Para el V24 disponemos de los chip MC1488 y MC1489. Para 20 mA un sencillo arreglo de transistores que convierten señales de voltaje a corriente y viceversa.

### 3.3 Diseño del bloque de memoria

Las memorias son importantes componentes en un sistema con microprocesador porque en ellas están almacenadas las instrucciones que serán procesadas por la CPU. La CPU necesita de diferentes tipos de memoria para ejecutar un programa. Estos tipos de memoria son:

Memoria de escritura/lectura llamadas RAM, volátiles, es decir pierden los datos al quitarles la alimentación.

- Memoria de sólo lectura llamadas ROM, no volátiles.

Memorias de sólo lectura programables borrables llamadas EEPROM. Borrables con luz ultravioleta y reprogramables eléctricamente, no volátiles.

- Memorias de sólo lectura programables, borrables, llamadas EEPROM. Borrables y programables

eléctricamente, estas son las mas usadas actualmente, no volátiles.

Toda la información en memoria es almacenada en forma binaria. La palabra de dato es de 8 bits y es cargada en una posición de memoria, la cual consiste en 8 celdas de memoria. El microprocesador Intel 8085 dispone de 16 líneas de dirección por tanto, puede direccionar 65,536 posiciones de memoria, es decir 64KB ( 1KB=1024). Las líneas del bus de control MEMR y MEMW son las que deciden si la palabra dato es escrita en una posición de memoria (MEMW=BAJO), o si la palabra dato es leída de una posición de memoria (MEMR=BAJO). Estas señales son activas en nivel BAJO.

### 3.3.1 Construcción de la unidad de memoria

Como el microprocesador Intel 8085 puede direccionar 64KB, que es una capacidad considerable tratándose de aplicaciones de control, salvo cuando necesitamos grabar datos leídos de algún sensor. Una solución de distribuir los 64 KB en memorias ROM, EEPROM y RAM es usar 8 bloques de 8 KB. La ROM es usada para el programa BIOS, es decir el entorno, la EEPROM donde se va a cargar la APLICACIÓN y la RAM para almacenamiento temporal de datos procesados para su ejecución a través de los dispositivos de entrada salida.

Cada bloque de 8KB consiste en  $2^{13}$  localizaciones de memoria, es decir 8,192 direcciones. Para poder direccionar 8KB necesitamos 13 líneas de dirección, de

A0 - A12 que son conectadas a todos los bloques de memoria de 8KB del bus de direcciones. Para poder seleccionar cada bloque hacemos uso de un decodificador cuyas entradas serán A13, A14 y A15, que podrá seleccionar 8 bloques, que es lo que necesitamos y así completamos las 16 líneas de direcciones del bus. Las líneas del bus del control MEMR y MEMW son conectadas a las RAM y solo MEMR a las ROM y EPROM.

### 3.3.2 Construcción de bloques de memoria de 8KB

Como la mayoría de memorias disponibles en el mercado son de 2KB, diseñaremos los bloque de memoria de 8KB a partir de memorias comerciales de 2KB. Por tanto necesitamos 4 chip de memoria de 2KB, la figura 3.7 muestra este diseño. Para direccionar 2 KB ( $2^{11}$  posiciones) de memoria necesitamos 11 líneas de dirección A0-A10 y como son 4 chips conectamos un decodificador de 2 a 4, las direcciones A11 y A12 del bus de direcciones las conectamos a la entrada del decodificador y las otras (A0-A10) a los pines de dirección de las memorias. Las salidas de los decodificadores se conectan a cada entrada CS (Chip Select) de cada memoria.

El diseño de los bloques de memoria debe ser modular, es decir la tarjeta de memoria montada con sockets, de tal forma que de acuerdo a las necesidades podemos intercambiar las EEPROM con RAM; usando chips compatibles en patillas e instalando un switch en la



programa y testearlo. Este módulo muestra (visualiza) el bus del sistema, que consiste en el bus de direcciones, bus de datos y bus de control.

El bus de direcciones, tiene 16 líneas que son displayados en 4 display en forma hexagesimal. El bus de datos, tiene 8 líneas que son displayados en 2 display en forma hexagesimal. El bus de control, tiene 4 líneas que son displayados directamente con Leds.

En la figura 3.9 se muestra el diagrama de bloques del módulo de visualización, donde se muestra como fácilmente podemos mostrar los buses. El bloque de "display multiplexer" es usado para economizar hardware usando un decodificador. El método utilizado, es hacer el multiplexado mediante programa, así logramos mayor transparencia al módulo, aunque usamos mas hardware se tuvo algunas dificultades al tratar de visualizar directamente los buses que no fueron superadas óptimamente por esto se decidió hacerlo con un programa que cuenta con la subrutina display para hacer la multiplexión por software.

### 3.5 Bloque del sistema paso a paso del 8085

Para probar un programa, una herramienta muy útil, es hacer correr el programa paso a paso, para poder corregir los errores. Hay dos métodos de operar en un sistema paso a paso:

- Que cada paso sea una instrucción.

- Que cada paso sea un ciclo de máquina.

En el sistema mínimo diseñado, se usó el segundo método, haciendo uso de dos líneas del CPU 8085: La entrada READY y la salida ALE.

### 3.5.1 La función READY

Debemos conocer que hace esta patilla para poderla utilizarla, entonces describiremos su funcionamiento. En el segundo ciclo de reloj T2 (figura 3.3.), la CPU pregunta el estado de la entrada READY. Un nivel BAJO causa que el CPU inserte "ciclos de espera" (wait cycles) hasta que esta señal se vaya a nivel ALTO. Entonces los ciclos de lectura y escritura en proceso, no serán terminados hasta desactivar esta función. Las líneas de dirección, datos y control están disponibles de ser displayados durante el ciclo de espera.

### 3.5.2 Circuito básico del modo paso a paso

En la figura 3.6 se muestra un simple circuito que realiza la operación de paso a paso. Si está en modo RUN (corrido del programa), la patilla READY siempre está en nivel ALTO a través de la resistencia R que está conectada a 5 voltios, por tanto no hay ciclos de espera, y el programa corre normalmente. El flip flop es reseteado con cada pulso de ALE, pero como la salida Q no está conectada por el switch, la entrada READY sigue en ALTO.

Si esta en modo STOP (paso a paso), la entrada READY va ser gobernada por la salida Q del flip flop porque el switch conecta hacia la patilla READY. Con cada pulso de ALE va a resetear el flip-flop y poner a nivel BAJO a READY, por tanto el procesador esta en estado de espera hasta que switch STEP sea presionado, cuando esto ocurre, hay un flanco de subida en el reloj del flip-flop y la salida Q=ALTO y READY=ALTO, el procesador retoma sus tareas hasta que el siguiente pulso de ALE, y así se repite el proceso. Entonces como en cada ciclo de máquina empieza con un pulso de ALE, y con este circuito de paso a paso, cada vez que hay un pulso de ALE pasa el CPU al estado de espera, podemos concluir el programa va a correr paso a paso, donde cada paso corresponde a un ciclo de máquina.

Se presenta un problema, el microprocesador Intel 8085 no acepta la entrada HOLD (petición de mantenimiento), mientras la entrada READY se mantenga en BAJO. Es decir el procesador no puede dejar el control de los buses mientras se mantenga en estado de espera. Esto para una aplicación determinada que haga uso de Acceso Directo a Memoria (DMA), no podríamos chequear correctamente el sistema. Este problema lo corregimos conectando la línea HOLD que proviene del DMA a la entrada S (set) del flip-flop de la figura 3.6 y de este punto conectar a través de un inversor a la entrada HOLD del 8085. Entonces cada vez que aparezca una petición de mantenimiento (HOLD), el procesador

deja el estado de espera y es aceptada la petición de mantenimiento.

### 3.6 Bloque de los dispositivos de entrada salida

El microprocesador para poder recibir o entregar datos al exterior necesita de dispositivos que nos permitan realizar esta tarea. Estos dispositivos preparan la señal para que pueda ser procesada.

#### 3.6.1 Bloque de interfase de entrada salida

Para conectar el sistema del microprocesador Intel 8085 con el exterior necesitamos unas interfases que nos permitan comunicarnos con los dispositivos periféricos como el teclado, los ADC, los DAC, display. Generalmente una interfase es una frontera entre dos o más dispositivos que comparten información; otras consideraciones relativas a las interfases incluyen la sincronización, dirección de transmisión de datos, ajuste de niveles de señal.

Una operación de entrada salida es el acto de transferir datos desde un dispositivo periférico seleccionado al microprocesador. El microprocesador es el foco de todas las operaciones, por tanto una entrada significa que el dato fluye hacia el microprocesador mientras que una salida, es cuando el dato fluye del microprocesador. Aquellas posiciones donde el dato entra o sale se denominan puertos de entrada o salida.

El microprocesador Intel 8085 utiliza dos instrucciones para transferir datos de los puertos de entrada/salida. La instrucción de salida esta representada por el nemonico OUT, mientras que la de entrada se usa el nemonico IN, estas instrucciones son de dos bytes, el primer byte es el Código de Operación y el segundo la dirección del puerto, esto quiere decir que se puede seleccionar 256 direcciones de puerto (8 bits). Cuando se utiliza la instrucción OUT, se utiliza una señal especial de escritura de entrada/salida IOW, la instrucción IN también requiere el uso de una señal de control llamada lectura de entrada/salida IOR. Ambas señales son activas en nivel bajo, la generación de estas señales se vio en la sección 3.2.4 , como parte de las señales del bus de control. Las transferencias de datos que utilizan las instrucciones IN y OUT se denominan E/S controladas por programa. Las instrucciones de programa están controlando las transferencia de datos durante las operaciones IN y OUT. Las transferencias de datos pueden ser iniciadas por un dispositivo periférico, utilizando interrupciones.

Existen dos técnicas de E/S controladas por programa:

La técnica "A" es la que usa las instrucciones IN y OUT.

Y la técnica "B" trata a las posiciones de entrada y salida como direcciones regulares de memoria. Esta

técnica se denomina E/S de mapa de memoria o direccionados como memoria, con esta técnica se utilizan instrucciones regulares de acceso a memoria como LOAD o STAGE. Entonces en esta técnica las líneas de dirección deben ser decodificadas y utilizadas para seleccionar la dirección exacta de un puerto de entrada o salida. Las señales habituales de control de escritura WR y lectura RD son usadas para entrar o sacar datos.

En el sistema diseñado, cuenta con tres dispositivos de entrada/salida: Uno de propósito especial que lo usaremos en los módulos del teclado y visualización, que esta direccionado como memoria (técnica B) en la de las direcciones 0800H hasta la 0803H; y dos de propósito general, para que sea usado por el usuario, estos dispositivos están direccionados, uno con la técnica "A" en la dirección de puerto 00H y el otro con la "B" en la dirección dispuesta por el usuario. Para que el usuario pueda practicar estas dos técnicas, uno de ellos esta conectado a los conversores de señales analógicas.

Las interfases utilizadas son el Programable Peripheral Interfase PPI 8255, que consisten en 3 puertos de entrada/salida de 8 bits programable, y van a conectar al sistema del microprocesador Intel 8085 con el teclado, los ADC, DAC, y otros componentes.

Describiremos brevemente la interfaz paralelo programable PPI 8255. Es un circuito integrado de 40

patillas, cuya adecuada programación resuelve la adaptación del sistema de buses de los microprocesadores de INTEL a periféricos que trabajan con información en paralelo con palabras de 8, 16 y 24 bits. El PPI consta de 3 puertos de entrada/salida en paralelo, a las que se denomina A, B y C. Las ocho líneas de la puerta C se dividen en 2 grupos de cuatro, que pueden trabajar combinadamente con los puertos A y B. Cada puerto tiene sus características específicas, lo que proporciona una gran flexibilidad a este dispositivo. Internamente el 8255 dispone de cuatro registros de los cuales tres se dedican a almacenar información que entra o sale por los puertos y el otro a realizar las funciones de control. La programación desde la CPU de este último registro, sirve para configurar los puertos y el funcionamiento general del PPI.

El PPI admite tres modos de funcionamiento que son:

**Modo 0** Es la forma mas sencilla y básica de comportamiento de los puertos de E/S. Los 3 puertos pueden trabajar como entrada o salida, pero sin posibilidad de disponer de líneas auxiliares de control para soportar diálogos con periféricos.

- **Modo 1** En este caso es posible que los puertos A y B puedan dialogar con los periféricos, actuando las líneas de el puerto C como soporte de las señales de diálogo.

- **Modo 2** Se permite trabajar al puerto A de forma bidireccional, o sea, que pueda actuar como entrada y salida de información, utilizando ciertas líneas del puerto C para establecer el diálogo con los periféricos. En este modo, la puerta B puede actuar en modo 0 o modo 1. Para la programación del registro de control se escribe una palabra en este registro, esta palabra de control determinan las siguientes funciones:

<b>BIT</b>	<b>PUERTO</b>	<b>NIVEL</b>	<b>FUNCION</b>
Bit D7		1 0	Para configurar modo del PPI puesta a 1 ó 0 de bit de PC
Bit D6	PA	0 0	modo 0
D5		0 1	modo 1
Bit D4	PA	1 0	entrada salida
Bit D3	PCH (PC4-PC7)	1 0	entrada salida
Bit D2	(PB + PCL)	1 0	modo 0 modo 1
Bit D1	PB	1 0	entrada salida
Bit D0	PCL (PC0-PC3)	1 0	entrada salida

WR: es el pin 36, es la señal que indica un ciclo de escritura sobre el PPI, se activa con nivel bajo.

RD: es el pin 5, es la señal que indica un ciclo de lectura sobre el PPI, se activa con nivel bajo.

CS: es el pin 6, y es la línea destinada para la selección de chip. Es necesario que este activa para que funcione el PPI, se activa con nivel bajo.

RESET: es el pin 35, la activación de esta señal significa la inicialización del PPI y la puesta a cero de los bits del Registro de Control, configurando como entradas a todas las líneas de las puertas.

A0-A1: son los pines 9-8, seleccionan cual de los registro del PPI es al que se procede a leer o a escribir. Dichos registros son PA, PB, PC, y el Registro de Control.

Las conexiones del PPI con el microprocesador se muestran en la figura 3.10.

El pin CS selecciona el chip cuando es decodificada esta dirección, además se conectan las señales del bus de control según como sea direccionado el PPI (técnica "A" o "B").

### 3.6.2 Bloque del teclado

Para poder ingresar el programa al sistema usamos un teclado de 16 teclas hexagesimales y 3 de propósito especial. Las 16 salidas de la teclas de los números hexagesimales son conectadas al decodificador de 16 a 4 líneas 74922. Las 4 salidas decodificadas del 74922



se conectan a los bits menos significativos del puerto A del PPI de propósito especial. El bit 4 de este puerto es conectado a la patilla del 74922 que es ALTA cuando se ha presionado una tecla. El bit 5 , 6 y 7 del puerto A se conectan a las teclas especiales ENTER1, SALIDA y AVANCE respectivamente. Estas conexiones se hacen directamente ya que la protección contra rebotes la hacemos por software.

El circuito es mostrado en la figura 3.11-A.

### 3.6.3 Bloque de conversión de datos análogo digital y digital análogo

El módulo dispone de un conversor análogo digital que pueden ser multiplexadas hasta ocho señales, que es el ADC 0808 y una Conversor Digital Análogo. Las operaciones con estos conversores lo hacemos a través de un PPI de propósito general y el software es dado por el usuario, es decir solo se dispone de la implementación hardware del módulo de conversión, y dejarlo listo para que con las adaptaciones a la aplicación específica funcione, también contiene circuitos de protección contra sobretensiones.

La entrada del puerto A del PPI es conectado a las patillas de datos del ADC, los 4 bit mas significativos del puerto C (CH) se conectan a las patillas de control de ADC.

La entrada del puerto B del PPI es conectado a las patillas de datos del DAC y los bits menos

significativos del puerto C (CL) a las patillas de control del DAC.

Estas conexiones se usando el ADC 0804 que solo dispone de una entrada analógica es mostrada en la Figura 3.11-B

El sistema tiene la opción de deshabilitar este módulo y usar el PPI para otro propósito.

### 3.7 Desarrollo del software

El software principalmente esta dedicado a las funciones de teclado. El teclado consiste en 19 teclas, 16 de los números hexagesimales y 3 teclas de función especial, que son ENTER1, SALIDA y AVANCE.

El módulo tiene un menú principal en donde podemos elegir 6 opciones, presionando las teclas A hasta la F. Estas opciones nos permiten realizar lo siguiente:

- Ingresar la dirección de memoria donde empezará el usuario su programa.
- Tiplar el programa desde la dirección elegida en memoria RAM.
- El usuario podrá leer la memoria RAM o ROM tiplando la dirección.
- Visualizar los 8 registros del microprocesador Intel 8085.
- Ejecutar el programa tiplado en memoria.
- Borrar porciones de memoria



Los 8 bits del puerto A están distribuidos de la siguiente forma:

- Los 4 primeros bits del 0 hasta el 3 son los datos de teclado.
- El bit 4 es conectado al pin del decodificador de teclado 74922 que se hace "1" cuando hay tecla presionada.
- El bit 5 es la tecla ENTER1.
- El bit 6 es la tecla SALIDA.
- El bit 7 es la tecla AVANCE.

Se pregunta si se presionó tecla a través del bit 4 del puerto A, si es 1 hay tecla presionada, si no vuelve a preguntar. Si hay tecla presionada lee el puerto A en sus 4 bits menos significativos, luego desplazamos estos 4 bits a las posiciones mas significativas mediante la subrutina RLC (desplazamiento a la izquierda en 4 posiciones) cargamos el registro H con 07 y L con el Acumulador, después el Contador de Programa PC es cargado con el contenido de HL.

De acuerdo a las direcciones especificadas por HL se van a las direcciones de las subrutinas de los 6 submenús:

Dirección 07A0 Submenú A      Dirección de Inicio de  
Programa



entramos el puerto A y verificamos el bit 5, que es "1" si la tecla ENTER1 es presionada (sólo se presiona la tecla ENTER1 para aceptar la dirección, no es necesario tipear 4 números, podemos tipear 3, 2 ó 1 número y luego presionar ENTER1 y salta al programa del menú principal), si es "0", leemos el puerto A en el bit 4 que es "1" si hay tecla presionada, si no sigue en este bucle hasta que se presione una tecla. Si hay tecla presionada leemos en puerto A, si es la primera tecla presionada, borra los datos de las siguientes 4 direcciones; el acumulador es guardado en memoria en la dirección especificada en el registro HL (que ha sido cargado en el menú principal con 0806H) luego incrementamos el registro HL y el lazo continua hasta tipear las 4 teclas y/o presionar la tecla ENTER1 para aceptar la dirección. Como cada dirección contiene solo 4 bit de datos, pasamos a una subrutina para pasarlos a 8 bits, es decir, como las teclas tipeadas se han guardado en 4 posiciones de memoria con solo 4 bits de datos, los pasamos a 2 bytes a partir de la dirección 0810H. Luego saltamos al programa del menú principal.

### 3.7.3 Submenú de tipeado de programa

Para ingresar a este submenú, presionados la tecla B en el Menú Principal, esta opción nos permite tipear nuestro programa en memoria RAM desde la dirección ingresada en el Submenú A. El usuario entonces tipeara

dos números hexagesimales que corresponden al dato de la dirección visualizada en el display y luego presionar la tecla ENTER1 para aceptar el dato y visualizar la próxima dirección, y así continuar hasta terminar su programa. Después del último dato de su programa deberá tipear C3, XX, YY en las tres siguientes direcciones (XXYY es la dirección de la rutina de visualización de registros) Si queremos visualizar los registros después de algunas instrucciones debemos tipear C3, XX, YY inmediatamente después de la instrucción. Usamos una tecla SALIDA para salir del submenú tipeado de programa y entrar al menú principal.

El diagrama de flujo del submenú de tipeado de programa se muestra en la figura 3.14. Entonces para ingresar los dos números del dato hacemos un lazo de dos vueltas, primero preguntamos si la tecla SALIDA ha sido presionada a través del bit 6 del puerto A, si es "1", si se presionó y por tanto se salta incondicionalmente al menú principal. Si es "0", no se presionó; pasamos a preguntar si la tecla ENTER1 ha sido presionada a través del bit 5 del puerto A, solo presionamos esta tecla si ya tipeamos los dos datos y lo aceptamos, si no es presionada pasa a preguntar si hay tecla presionada a través del bit 4 del puerto A, si no la hay vuelve a realizar el lazo, si hay tecla presionada PA4=1 se decrementa el número de vueltas a través del registro B y almacena este dato en la

dirección de memoria visualizada e incrementa el registro HL y vuelve a realizar el lazo, al tipear el segundo dato, el programa se queda en un bucle infinito hasta que se presione la tecla ENTER1 para aceptar el dato, y se vuelve a realizar el lazo, luego de tipear todo el programa el usuario presionará la tecla SALIDA para volver al menú principal.

#### 3.7.4 Submenú lectura de programa

Presionando la tecla C del menú principal ingresamos al modo de lectura del programa; donde el usuario podrá leer el contenido de la memoria ROM o RAM; tipeando la dirección y presionando la tecla AVANCE se visualizará el dato de la dirección siguiente. Las direcciones y los datos son mostradas en el display, el diagrama de flujo es detallado en la figura 3.15.

Primero hacemos un lazo de 4 vueltas, para ingresar los 4 números de la dirección a ver a través del registro B que lo inicializamos a 5 (número de dígitos + 1). En este lazo preguntamos si la tecla SALIDA ha sido presionada (a través del bit 6 del puerto A) si es así, se salta incondicionalmente al programa principal, si no es presionada, pasa a preguntar si la tecla ENTER1 se ha presionado (que debe presionarse sólo cuando se ha terminado de tipear la dirección a visualizar) si es SI, pasa a la subrutina AVANCE, si es NO, pregunta si hay tecla presionada; si es NO, vuelve a realizar el lazo en forma indefinida y si es

SI, ingresa la tecla presionada a través de los 4 bits menos significativos del puerto A, pasa a la subrutina de DISPLAY.

Luego incrementa el registro B (número de vueltas del Lazo) guarda el acumulador (que contiene el número tpiado) en la dirección especificada por el registro HL se incrementa el registro HL y vuelve a realizar el lazo. Recordemos que HL fue cargado en el menú Principal con 0830H.

Entonces cada vez que presionamos una tecla se visualizará el número tpiado.

La subrutina DISPLAY se encarga de visualizar a través del display, como son 6 display de 7 segmentos, la técnica de multiplexado permite reducir el número de decodificadores a solo 1 el diagrama de flujo se muestra en la figura 3.19.

En el byte de memoria, los 4 bits menos significativos se han guardado el número tpiado (de 0 a F) y en los cuatro más significativos lo usamos para la técnica de multiplexado como muestra la figura 3.9. Como son 6 display usamos 3 bits que van a ser conectados a través del puerto B del PPI de propósito especial, al decodificador, luego en el programa estos 3 bits varían de 000 a 111, para que se prenda un display a la vez. La subrutina AVANCE carga los números de las teclas tpiadas (desde la dirección 0830H) a las posiciones 0837H y 0838H, ya que las anteriores eran usadas para displayar, los datos de

estas direcciones son cargadas al registro HL. Luego pregunta si la tecla SALIDA se ha presionado, si es así salta incondicionalmente al submenú de lectura , si es NO pasa a la rutina display, luego pregunta si la tecla AVANCE ha sido presionada, si es SI incrementa el registro HL y visualiza el dato luego vuelve al inicio de la subrutina y así en bucle indefinido hasta presionar la tecla SALIDA.

### 3.7.5 Submenú lectura de registros

Para ingresar al submenú de Lectura de Registros presionamos la tecla D del menú principal.

El usuario tiene la posibilidad de ver el contenido de los 8 registros del microprocesador Intel 8085, apenas ingresa al submenú, como son 6 display, cada display muestra un registro de la siguiente forma

Primer display      Registro A (Low)  
Segundo display :    Registro A (High)  
Tercer display      Registro B (Low)  
Cuarto display      Registro B (High)  
Quinto display      Registro C (Low)  
Sexto display        Registro C (High)

Para mostrar los demás registros, usamos el teclado, cada tecla presionada mostrará los tres siguientes registros. El programa guarda el contenido de todos los registros en memoria, apenas ingresamos a la

subrutina en las siguientes posiciones de memoria:

Posición de Memoria 0870H	Registro A
Posición de Memoria 0871H	Registro C
Posición de Memoria 0872H	Registro B
Posición de Memoria 0873H	Registro E
Posición de Memoria 0874H	Registro D
Posición de Memoria 0875H	Registro L
Posición de Memoria 0876H	Registro H
Posición de Memoria 0877H	Registro SP (Low)
Posición de Memoria 0878H	Registro SP (High)
Posición de Memoria 0879H	Registro de status

Luego pasa a la subrutina de multiplexaje para poder displayar los tres registros en el orden siguiente

Primero mostrará los registros A, C y B, presionamos una tecla y mostrará los registros E, D y L luego presionamos otra tecla y mostrará los registros H, SP(L) y SP(H), y por último al presionar una tecla mostrara el registro de status. Esto lo hacemos preguntando si hay tecla presionada a través del bit 4 del puerto A, si no la hay displaya lo que tenga en las direcciones especificadas y vuelve a preguntar, si hay tecla presionada desplaza en tres las posiciones a displayar y así hasta presionar tres veces cualquier tecla. Esta opción la tenemos para visualizar los registros y verificar resultados, cada vez que









































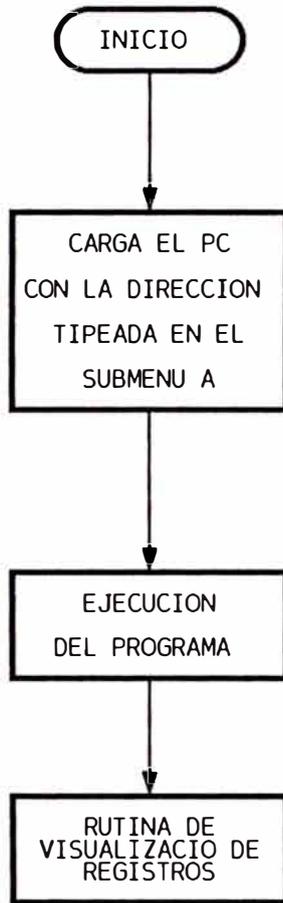


FIGURA 3.17 DIAGRAMA DE FLUJO DEL SUBMENU EJECUCION DE PROGRAMA





## C A P I T U L O   I V PUESTA EN MARCHA DEL MODULO

El módulo diseñado entonces, nos permite entrenar en sistemas basados en microprocesadores hacer pruebas y si los resultados son los esperados se implementa el circuito. Para poner en marcha a módulo debemos seguir los siguientes pasos

- 1.- Para una aplicación determinada el usuario deberá tener la solución del problema en un circuito usando microprocesadores.
- 2.- Si usa el microprocesador Intel 8085 pasa al paso tres, de lo contrario modificar su circuito si es necesario, para usar el microprocesador 8085.
- 3.- Diseñar el programa usando las instrucciones del microprocesador Intel 8085, en código fuente (nemonico) y en código objeto (binario), lo más ordenado posible.
- 4.- Conectar al módulo los dispositivos externos como señales de sensores, transductores o dispositivos finales de salida. Teniendo en cuenta que las características eléctricas sean compatibles con el módulo.
- 5.- Prender el módulo conectando la fuente de alimentación de 5 voltios; el módulo responderá colocándose en el





(señalizadores).

Si presionamos nuevamente otra tecla se repetirá la secuencia visualizando A, C y B. Finalmente presionamos la tecla SALIDA para regresar al menú principal.

- 11.- Hacer las pruebas respectivas en de campo y si los resultados son los esperados, quemar la EPROM junto con el programa, con direcciones bajas y llevar a circuito impreso el sistema mínimo con las especificaciones de la aplicación.

No es necesario seguir los pasos anteriores; si la aplicación lo requiere podemos entrar a cualquier submenú en cualquier orden porque el módulo así lo permite.

También podemos probar porciones del programa del usuario como por ejemplo, subrutinas si el programa es muy grande.

El módulo se uso para realizar un termómetro digital y control de temperatura ON/OFF con microprocesador, un secuenciador de luces y un órgano electrónico

## CAPITULO V DESARROLLO FUTURO

El sistema realizado en el presente trabajo por las limitaciones descritas en el capítulo anterior puede desarrollarse en dos etapas siguientes

- Interconexión del sistema mínimo con la PC.
- Interacción del sistema mínimo con la PC en tiempo real.

### 5.1 Interconexión del sistema mínimo con la PC

El sistema mínimo de propósito general con el microprocesador Intel 8085, es un sistema básico que puede mejorarse si se conecta con la PC. Como primera etapa es conectar mediante una interfase de adquisición de datos conectado al slot de la PC de tal forma que mediante un sencillo programa en la PC podríamos tipiar el programa en nemonico, listarlo, verificar si esta bien tipiado para luego con una función grabarlo en disquete al mismo tiempo que grabamos el programa en la memoria del sistema mínimo.

El programa de la PC convierte los nemonicos en códigos de operación del microprocesador Intel 8085 con una simple tabla de equivalencias; este programa puede ser realizado con lenguajes de alto nivel que son populares actualmente (Clipper, Pascal, C.).

## 5.2 Interacción del sistema mínimo con la PC en tiempo real

Como segunda etapa para desarrollar el sistema; es interactuar con la PC en tiempo real, es decir que lo podamos controlar y verificar su funcionamiento con programas más sofisticados.

Con los recursos que nos brinda la PC como su fácil programación con lenguajes de alto nivel, su alta capacidad de memoria, velocidad, compatibilidad, unidades de almacenamiento, sus periféricos como impresoras, podemos llegar a tener un sistema muy completo.

Podremos tener una pantalla muy amigable fácil de usar, con detección de errores, medición de retardos, visualización de los registros internos del microprocesadores en tiempo real y usar todos los recursos que dispone la PC.

## CONCLUSIONES

- 1.- El módulo diseñado en el presente trabajo, encuentra su mayor aplicación como módulo de enseñanza y entrenamiento para conocer y aplicar sistemas basados en el microprocesador Intel 8085. También el módulo será usado para probar un diseño basado en microprocesador antes de su implementación final, ya que con el módulo se puede tiliar y correr el programa en memoria RAM, realizar pruebas, depurar el programa para que al final cuando se tenga los resultados que se quiere recién implementar el sistema final que va a ser puesto en funcionamiento en planta. No intenta suplir a los microcontroladores, sino por el contrario es una ayuda fundamental para ganar experiencia en sistemas discretos y luego realizarlo con microcontroladores.
  
- 2.- El microprocesador Intel 8085 es uno de los microprocesadores de 8 bits mas sencillos de implementar y es ideal para aplicaciones sencillas usando microprocesador de tal forma que muchos circuitos implementados con circuitos digitales pueden ser reemplazados por un sistema con el 8085 con todas las ventajas que nos proporciona el microprocesador.

- 3.- El módulo diseñado es bastante modular y flexible para poder usarlo en cualquier aplicación; la característica "transparente" que se fijó como objetivo, fue cumplido.
- 4.- Los módulos de memoria han sido distribuidos en 4 bloques de 8KB y estos a su vez en 4 bloques de 2KB de los cuales pueden intercambiarse como memoria RAM o EPROM con tan solo mover unos switch; esto da una característica muy modular al sistema.
- 5.- Todas las señales de control pueden ser visualizadas y usadas directamente por el usuario de tal forma que podemos hacer crecer a este sistema para aplicaciones mas complejas.
- 6.- Mediante un programa realizado por el usuario el sistema podrá recibir y transmitir datos en forma serial ya que el hardware esta implementado en el sistema y se aprovecha la opción interna que tiene el microprocesador Intel 8085; entonces puede realizarse aplicaciones con este tipo de transmisión de señales. Cabe resaltar que se realizaron varias pruebas y su uso es un poco limitado ya que no podemos calcular exactamente el tiempo de transmisión/recepción y se hace dificultoso la sincronización con frecuencias mayores a 1000Hz.  
El módulo o básicamente el microprocesador Intel 8085,

esto hace que el módulo no sea tan transparente, ya que insertaríamos varias instrucciones que no son parte del programa que está corriendo el usuario.

- 8.- El módulo dispone de tres dispositivos entrada/salida PPI, dos de ellos direccionados como memoria y uno como puerto para que el usuario pueda entrenar en estos dos modos de direccionamiento de un dispositivo E/S. Uno de estos PPI es de propósito especial ya que enteramente es usado por el módulo y los dos restantes son disponibles para el usuario; esta característica hace al módulo apropiado para la enseñanza.
  
- 9.- El módulo dispone de un bloque de conversión analógica digital y viceversa, este bloque se conecta después de un PPI, para darle la característica modular, si el usuario requiere de usar señales analógicas lo realiza direccionando los puertos, también debido a que estas conversiones se realizan con diferentes velocidades de muestreo no se realizó una subrutina para esto, sino se deja solo el hardware para que el usuario lo use con su propio programa. Como estos programas usan varias instrucciones y por tanto usan tiempo; tenemos la limitación de velocidad de muestreo al realizar las pruebas se llegó a muestrear señales de hasta 700 Hz con distorsiones aceptables a parte que los conversores usados son de 8 bits.

- 10.- Para el módulo del teclado se uso el método de "pooling", este método consiste en preguntar en cada intervalo de tiempo, si hay tecla presionada para poder procesarla. También se pudo usar el método que usa una interrupción para el teclado, pero como uno de los objetivos es hacer el módulo lo mas transparente posible y este método requiere una interrupción y ya no la puede disponer el usuario. Con el método usado todas las interrupciones son disponibles para el usuario. El software del módulo esta dedicado casi totalmente a las opciones de teclado.
- 11.- El programa puede depurarse y hacerlo mas pequeño ya que usan rutinas parecidas en cada submenú; debido a que se requiere hacer al módulo transparente como si el usuario tenga su aplicación como la tendría en su propio circuito. Se hace un poco largo y repetitivo, pero con la posibilidad de poderlo mejorar el módulo permite esto cuando hay que cambiar la memoria EPROM con el nuevo programa. Es decir que el sistema se le puede colocar cualquier programa siempre y cuando se tenga en cuenta solo la dirección de los puertos.
- 12.- Hay dos métodos para la visualización de los buses de dirección y datos, el de visualizarlo directamente de los buses usando el multiplexaje de display en hardware y el otro método de hacer el multiplexaje por software usando instrucciones del microprocesador



salto a la subrutina de visualización, entonces inmediatamente puede visualizar los registros usados. También podemos solucionar este problema realizando un programa mas complejo.

15.- Si el programa es muy grande es tedioso tipiar el programa, entonces podemos probar por partes el programa que es una limitación ya que en cada sesión debemos tipiar nuevamente el programa.

16.- En el desarrollo futuro planteamos el conectarlo con la PC y usar sus recursos de almacenamiento, es decir tipiar el programa y grabarlo en disquete entonces en cada sesión sólo se lee el disquete.

**APENDICE A**

**DATA SHEET DEL MICROPROCESADOR INTEL 8085**















**APENDICE B**  
**LISTADO DE PROGRAMA DEL MODULO**

















## Subrutina DIREC-DI

```

MOV      A, L
ANI      0FH
STA      0832H
MOV      A, L
ANI      F0H
CALL     RRC
STA      0833
MOV      A, H
ANI      0FH
STA      0834H
MOV      A, H
ANI      F0H
CALL     RRC
STA      0835H
RET

```

## SUBMENU D

Ver registros actuales

```

          0385H
          CALL     REG. TRANS. (03C5H)

INPA54:   LDA      0800 H
(038AH)

          ANI      20H
          CPI      20H
          JNZ      INPA44
          CALL     DELAY 0,5 seg.
          JMP      INPA4 (RETORNO PRINCIPAL)
          007C

INPA44:   LDA      0800H
(039AH)

          ANI      10H
          CALL     DISPLAY-POS
          (0100H)
          CPI      10
          JNZ      INPA54
          LDA      0800H
          ANI      0FH
          ORI      70H
          AOI      06H
          MOV      L, A
          MUI      H, 08H
          SH LD    0806H
          CALL     DELAY 0,5 seg.
          JMP      INPA54

```

Subrutina DISPLAY REG.  
(03C5H)

```

LDA      0840H
ANI      0FH
STA      0870H

```

LDA	0840H
ANI	F0H
CALL	RRC (0320)
STA	0871H
LDA	0841H
ANI	0FH
STA	0872H
LDA	0841H
ANI	F0H
CALL	RRC
STA	0873H
LDA	0842H
ANI	0FH
STA	0874H
LDA	0842H
ANI	F0H
CALL	RRC
STA	0875H
LDA	0843H
ANI	0FH
STA	0876H
LDA	0843H
ANI	F0H
CALL	RRC
STA	0877
LDA	0844H
ANI	0FH
STA	0878H
LDA	0844H
ANI	F0H
CALL	RRC
STA	0878H
LDA	0844H
ANI	F0H
CALL	RRC
STA	0879
LDA	0845H
ANI	0FH
STA	087AH
LDA	0845H
ANI	F0H
CALL	RRC
STA	087BH
LDA	0846H
ANI	0FH
STA	087CH
LDA	0846H
ANI	F0H
CALL	RRC
STA	0870H
LDA	0847H
ANI	0FH
STA	087EH
LDA	0847H
ANI	F0H

CALL	RRC
STA	087FH
LDA	4748H
ANI	0F
STA	0880A
LDA	0848H
ANI	F0H
CALL	RRC
STA	0881H
LDA	0849H
ANI	0FH
STA	0882H
LDA	0849H
ANI	F0H
CALL	RRC
STA	0883H
LDA	084AH
ANI	0FH
STA	0884H
LDA	084A
ANI	F0H
CALL	RRC
STA	0885H
LDA	084BH
ANI	F0H
CALL	RRC
STA	0887H

**SUBMENU F**  
**Borra RAM**

A	LHLD	08B7H (POSICION DIRECCION INICIAL BORRAR)
A	XCHG	
	1HLD	08A7H (POSICION DIRECCION INICIAL BORRAR)
LABEL-4:	MOV	A, H
	CMP	D
	JZ	LABEL-1
	JM	LABEL-2
	JMP	LABEL-3
LABEL-1:	MOV	A, L
	CMP	E
	JM	LABEL-2
	JZ	LABEL-3
LABEL-2	MUI	M, 00H
	INX	H
	JMP	LABEL-4
LABEL-3	JMP	RETORNO (007CH) PRINCIPAL

## RUTINAS INTERMEDIARIAS DE OPCIONES

0780H

```
LXI     H, 08A0H
SHLD   0806H
CALL   DELAY-0,5 seg.
LXI     H, 08A2
JMP    MENU 1 0140H
```

0790H

```
LXI     H, 08B0H
SHLD   0806H
CALL   0115H
LXI     H, 08B2H
JMP    MENU 1 0140H
```

07A0B

```
LXI     H, 0810H
SHLD   0806H
CALL   0115H
LXI     H, 0812H
JMP    0140H Opcion 1
```

0780H

```
LXI     H, 0820H
SHLD   0806H
CALL   0115H
JMP    01CAH Opcion 2
```

07C0H

```
LXI     H, 0830H
SHLD   0806H
CALL   0115H
LXI     H, 0832H
JMP    028AH Opcion 3
```

07D0

```
LXI     H, 0870H
SHLD   0806H
CALL   0115H
JMP    0385H Opcion 4
```

07E0H

```
LHLD   0817H
PCHL   Opcion (RUN)
```

07E0H

```
JMP    Borra RAM 6
        (04F5H)
```

## BIBLIOGRAFIA

1.- Manuel Torres Portero :

MICROPROCESADORES Y MICROCONTROLADORES APLICADOS A LA INDUSTRIA  
Edicion 1989.  
Editorial Paraninfo.

2.- Roger L. Tokhein

FUNDAMENTOS DE LOS MICROPROCESADORES  
Segunda Edicion, 1990  
Editorial Mc Graw Hill

3.- Helbert Taub y Donald Shilling

ELECTRONICA DIGITAL INTEGRADA  
Edicion 1980  
Editorial Marcombo