

SISTEMA INTEGRADO PARA DESARROLLO DE SOFTWARE
PARA EL MICROPROCESADOR Z80
S I D E S - Z 8 0

MONTALVO, LUIS ING.
VALENCIA, GALO ING.
ESCUELA POLITECNICA NACIONAL

RESUMEN

Se presenta una descripción de las características del Sistema Integrado para Desarrollo de Software para el Microprocesador Z80 diseñado en el Área de Sistemas Digitales de la Facultad de Ingeniería Eléctrica de la Escuela Politécnica Nacional. El sistema se halla constituido por el administrador central implementado en lenguaje C; editor de textos, calculadora y tabla ASCII elementos del Sidekick; Meta-assembler Cross-16; Simulador para el microprocesador Z80 AVSIMZ80 y Programador de memorias PROM-2000.

ABSTRACT

A description of the characteristics of an Integrated System for Software Development for the Z-80 Microprocessor, designed at the Digital Systems Area of the Electrical Engineering Department of the National Polytechnique University, is presented. The system is composed by the central manager coded in C; the word processor, calculator and ASCII table from Sidekick; the Cross-16 Meta-assembler; the AVSIMZ80 simulator and the PROM-2000 memory programmer.

INTRODUCCION

El presente trabajo es el resultado de la labor colectiva llevada a cabo durante los últimos tres años por aquellos profesores que han tenido a cargo las materias de Microcomputadores I y Laboratorio de Microcomputadores I en la Facultad de Ingeniería Eléctrica de la Escuela Politécnica Nacional. Tanto el esquema de las prácticas de laboratorio como las herramientas de hardware y software empleadas en las mismas, han sido ensayados por varios semestres y continuamente modificados, acogiendo las valiosas sugerencias brindadas por los estudiantes, principales beneficiarios de este trabajo.

Antecedentes Históricos:

La enseñanza de microcomputadores en la Facultad de Ingeniería Eléctrica se inicia con el estudio de microprocesadores de 8 bits y toma como referencia principal al microprocesador Zilog Z-80, debido a su amplia utilización. Hasta agosto de 1986 las prácticas de laboratorio se las realizaba utilizando el equipo de desarrollo Micro-Professor I fabricado por Multitech, el cual presenta varias limitaciones propias de equipos didácticos de bajo costo.

Con el advenimiento de los computadores personales aparece en el mercado una serie de herramientas de Software que permiten utilizar el computador personal para el desarrollo de software de otros microprocesadores diferentes a la CPU del sistema. El área de Sistemas Digitales de la Facultad de Ingeniería Eléctrica se propuso introducir tales herramientas en la enseñanza de microprocesadores.

Entre los elementos necesarios para llevar adelante este proceso se encuentran un procesador de textos, un cross-assembler, un simulador, un programador de memorias y un módulo de hardware básico. En el mercado no existe un solo proveedor para todos estos módulos por lo cual fue necesario familiarizar a los estudiantes con los distintos paquetes de los distintos fabricantes, cada uno de ellos con sus peculiaridades propias.

La utilización de los computadores personales si bien había ampliado el área de estudio también conllevaba un estudio preliminar de los módulos independientes lo cual se traducía en un menor aprovechamiento de los recursos debido a que los estudiantes no tenía conocimientos previos de computadores personales, diluyéndose el objetivo principal de la materia, que es el estudio del microprocesador Z80.

Objetivos:

El Área se propuso desarrollar el sistema integrado para desarrollo de software para el microprocesador Z80, el cual es materia del presente artículo, con las siguientes características: integrado, rápido, general y fácil de utilizar; que permita elevar la eficiencia y productividad de los alumnos en el estudio del microprocesador.

DESCRIPCION

La Fig. 1.1 presenta un diagrama de bloques del hardware básico necesario para el presente sistema.

Los computadores personales disponibles en la Facultad de Ingeniería Eléctrica son: Computador SANYO MBC-885 y Computador SANYO PLUS cuyas características se presentan en la Tabla 1.1.

El proceso de desarrollo de los programas es el que se muestra en la Fig. 1.2. Todos los paquetes de software son controlados por el Sistema Operativo MS-DOS a través del administrador central implementado.

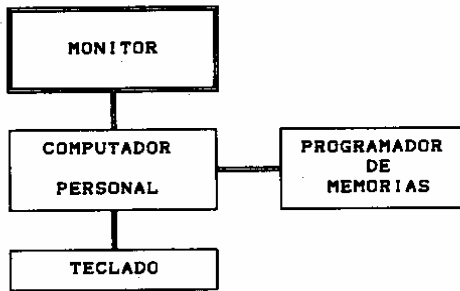


Fig. 1.1 Diagrama de Bloques del Hardware

Procesador	0000 (4.7 - 8 Mhz)
Memoria	640K
Almacenamiento masivo	2 unidades de disco flexible de doble lado/doble densidad de 360K.
Pórticos de E/S	1 RS232-C. 1 pórtico paralelo Centronics. 1 pórtico para video.
Ranuras de expansión	8 o 3
Monitor	Video Compuesto Monocromático

T. 1.1 Características Computadores SANYO

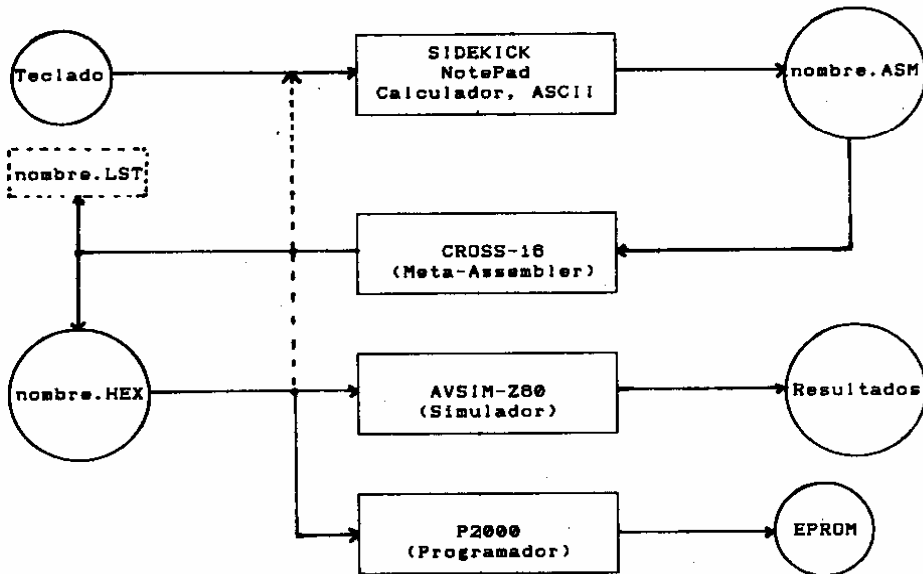


Fig. 1.2 Proceso de Desarrollo de Programas.

Sidekick. Es un paquete integrado de Software denominado "Desktop Organizer", el mismo que contiene varios utilitarios que una vez instalados quedan residentes en la memoria del computador y por lo tanto son accesibles en cualquier instante. Los utilitarios que se utilizan son:

- 1.- NotePad (Editor de texto) empleado para crear los archivos fuente escritos en lenguaje assembly del microprocesador Zilog Z80. El programa fuente es ingresado a través del teclado.
- 2.- Calculadora que permite realizar operaciones básicas en binario, decimal y hexadecimal.
- 3.- Tabla de equivalencias del código ASCII.

Cross-16 Meta-Assembler. Un programa en lenguaje assembly es un archivo de texto (fuente) en donde los mnemónicos del fabri-

cante son utilizados para representar directamente los códigos binarios que el computador realmente ejecuta.

Un assembler es un programa que convierte éstos mnemónicos a sus correspondientes códigos binarios. El término simple "Assembler" usualmente se refiere a un Assembler Residente, el cual ensambla programas para el mismo procesador en el cual se ejecuta el "Assembler". Un Cross-Assembler, ensambla programas para un solo procesador distinto a aquel en el que se ejecuta el "Assembler".

CROSS-16 es un programa Meta-Assembler que permite ensamblar programas para varios procesadores distintos. El programa lee desde disco el archivo fuente codificado en lenguaje assembly junto con una tabla de instrucciones del assembler y genera un archivo de salida que contiene el lenguaje de máquina hexadecimal ejecutable. En el laboratorio de microcomputadores I se utiliza la tabla de instrucciones del Assembler del microprocesador Zilog Z80.

Operación del Assembler.- CROSS-16 puede ejecutarse utilizando el siguiente comando:

```
C16 nom.ASM [-L nom.LST] [-H nom.HEX]
```

Donde los paréntesis cuadrados [] indican items opcionales.

Declaración de la Tabla de la CPU.- La directiva CPU indica a CROSS-16 la tabla de instrucciones del procesador que debe cargarse durante el ensamblado, que en el caso del laboratorio de microcomputadores es "Z80.TBL".

CPU "Z80.TBL"

Avsimz80. Es un programa diseñado para simular en memoria del computador personal, el hardware del microprocesador Z80 incluyendo todos los registros, puertos, memoria y periféricos. El simulador provee una "CPU visual" completa en la cual se puede observar la ejecución del código y su efecto en los registros, puertos y direcciones de memoria. Sus características son:

- 1.- Simulación completa de la CPU, incluyendo el hardware del temporizador, interrupciones y puertos.
- 2.- Despliegue visual del status de todos los registros, banderas, puertos y áreas de memoria seleccionables por el usuario, durante la ejecución del programa.
- 3.- Capacidades de depuración simbólica completas, incluyendo puntos de parada: fijos, dinámicos y condicionales, puntos de paso, ejecución paso a paso y ejecución paso a paso reversa.
- 4.- Cross-Assembler incremental que permite implementar parches de código y probarlos inmediatamente sin necesidad de abandonar el simulador.
- 5.- Despliegue interactivo de toda la pantalla. Cualquier registro, puerto o dirección de memoria puede ser modificado directamente.
- 3.- Simulación de entrada/salida directa cambiando los valores de los puertos.
- 7.- "Archivos de Comando" que permiten memorizar secuencias de teclas para su utilización en sesiones posteriores.

Operación del Simulador.- AVSIMZ80 puede ejecutarse utilizando el siguiente comando:

```
AVSIMZ80 [switches_opc] [argumentos_opc]
```

Los switches opcionales permiten optimizar la operación de AVSIMZ80 en varios computadores compatibles con el IBM-PC. Los argumentos opcionales son una serie de comandos para el AVSIMZ80 que el simulador realizará cuando se ejecuta el programa.

P2000. Es un programador de EPROMS que contiene un interfaz para conectar a un puerto de expansión de un PC compatible con IBM. El utilitario consta de tres partes: Hardware (que se conecta al puerto), Software (que controla las operaciones

del P2000) y el módulo receptor de la EPROM. Permite programar todas las EPROMS normalizadas industrialmente de 16, 20 y 24 pins.

Ms-Dos. Microsoft-Disk Operating System, es el sistema operativo encargado de coordinar el flujo de información entre la pantalla, el teclado, la memoria y los discos de almacenamiento. El sistema operativo se encarga también de controlar otros dispositivos conectados al computador, tales como impresores y periféricos especializados.

SIDES-Z80

El Sistema Integrado para Desarrollo de Software para el microprocesador Z80, es un paquete de Software desarrollado en lenguaje C, que simplifica considerablemente las tareas de edición, ensamblado, depuración y simulación de programas codificados para el uP Z80, así como, la programación de memorias EPROM con el programa final para su utilización práctica.

El sistema ha sido integrado en base a los paquetes de Software previamente descritos y utilizando de los mismos únicamente aquellas partes que son de utilidad en el laboratorio de microcomputadores I.

El sistema puede ser utilizado en computadores IBM PC o compatibles, IBM PS; que dispongan de dos unidades de diskette o una unidad de diskette y disco duro, monitor monocromático o de color y 256 Kbytes de memoria RAM.

Aun cuando el sistema es totalmente general, en el laboratorio de microcomputadores I se utiliza una versión compatible con las características de configuración de los computadores SANYO disponibles en la Facultad de Ingeniería Eléctrica, de tal manera de aprovechar al máximo las características particulares de dichos equipos.

El paquete de software se encuentra implementado en dos diskettes de 360 Kbytes, uno para los programas del sistema y otro para almacenamiento de los datos (programas de usuario).

En el caso de computadores que dispongan de unidades de diskette de 3 1/2" (720 Kbytes o 1.44 Mbytes), el sistema completo y datos se instala en un solo diskette.

El sistema se compone de los siguientes módulos de software:

- DOS (versión 3.1 o posterior)
- NotePad, Calculadora, Tabla ASCII (SIDE-KICK)
- CROSS-16
- AVSIM-Z80
- P2000
- FORMAT (DOS)
- MANIPULACION LIMITADA DE ARCHIVOS (DOS)
- SIDES

Descripción del Sides. El diskette del sistema debe ser colocado en la unidad "A" y el diskette de datos en la unidad "B" (computadores con dos unidades de diskette).

Al encender el equipo se crea un disco virtual (siempre y cuando el computador disponga de 512 Kb de memoria o más) en el cual se cargan los programas utilizados con el objeto de aprovechar su alta velocidad de ejecución, se ejecuta el Sidekick quedando residente en memoria y se ejecuta el programa SIDES.EXE

Se despliega una pantalla de presentación, en la cual se deben actualizar la fecha y hora del sistema y una ventana de ayuda. Luego se ingresa al menu principal en el cual se presentan las diferentes opciones.

Hardware. Adicionalmente se ha implementado un prototipo experimental que permita realizar pequeñas prácticas de laboratorio en las cuales el estudiante entre en contacto con el hardware del microprocesador Z80.

El diagrama de bloques del sistema de entrada/salida del prototipo experimental se presenta en la Fig. 1.4.

La Fig. 1.3 muestra el mapa de memoria del prototipo experimental.

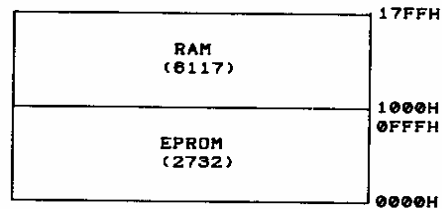


Fig. 1.4. Mapa de memoria del Prototipo

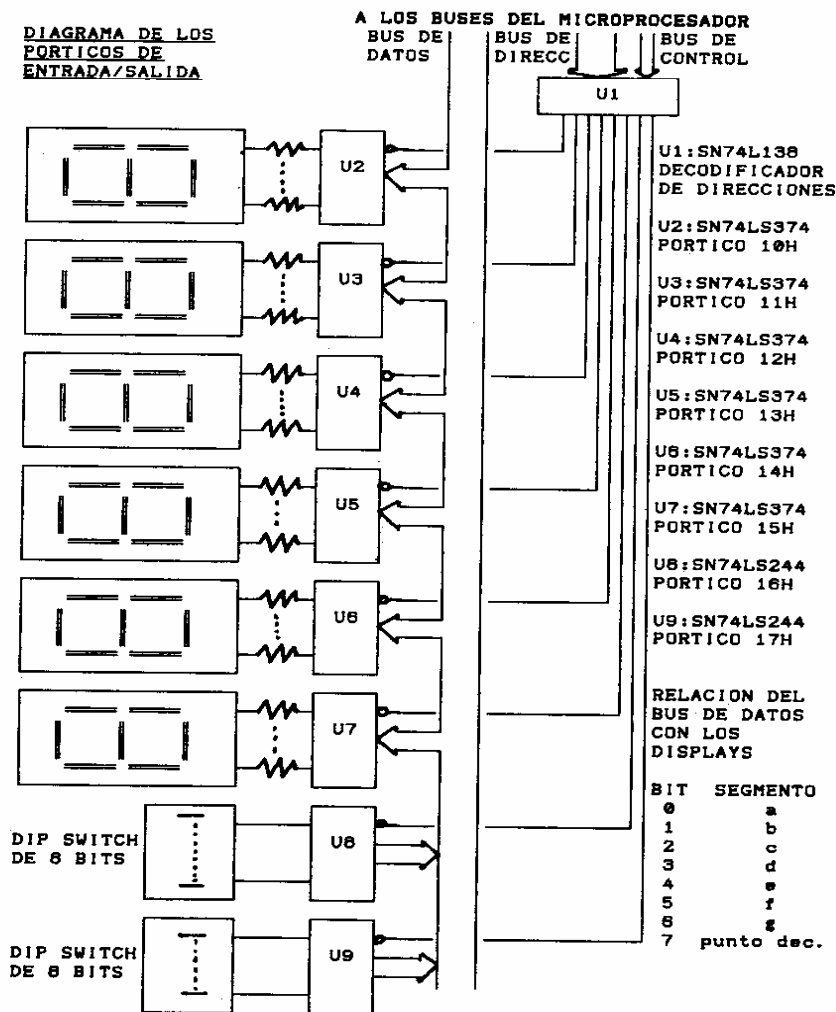


Fig. 1.4 Sist. Entrada/Salida del Prototipo Experimental

El prototipo consta de 6 displays de 7 segmentos de ánodo común y dos DIP switches (8 elementos cada switch). Cada display se encuentra conectado al bus de datos del microprocesador a través de un pórtilo de salida implementado en base a retenedores con salidas tres estados (SN74LS374) y cada DIP switch se halla conectado al mismo bus a través de un puerto de entrada implementado en base a buffers tres estados (SN74LS244). Los puertos de entrada han sido asignados a direcciones 16 y 17 y los puertos de salida las direcciones comprendidas entre 10 y 15.

Con el propósito de ilustrar la utilización del sistema implementado se presenta a continuación el proceso completo de desarrollo de un programa que permite mostrar en los displays del prototipo el mensaje "Lab uC".

- 1.- La pantalla de ingreso al sistema al encender el equipo se presenta en la Fig. 1.5 y la pantalla del menú principal que maneja todos los módulos del sistema en la Fig. 1.6 respectivamente.

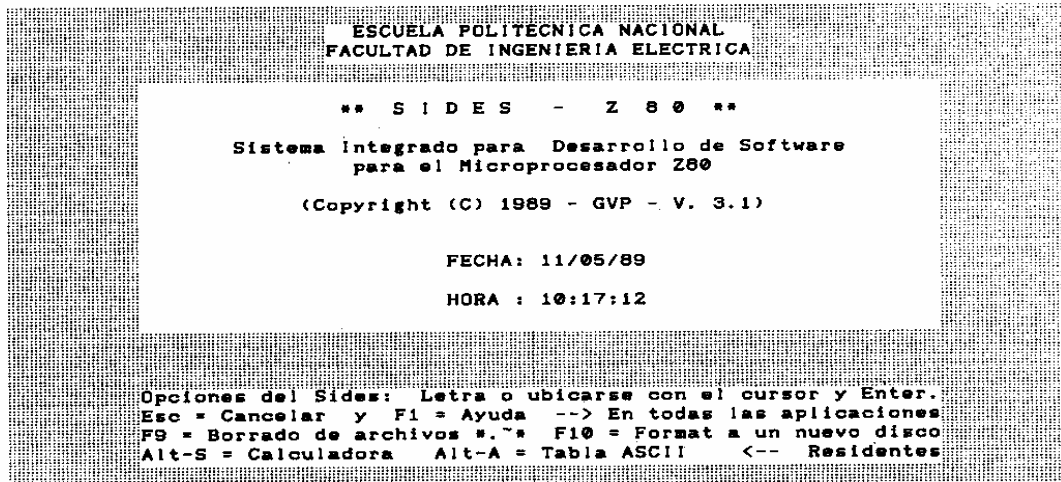


Fig. 1.5 Pantalla de Presentación del Sistema

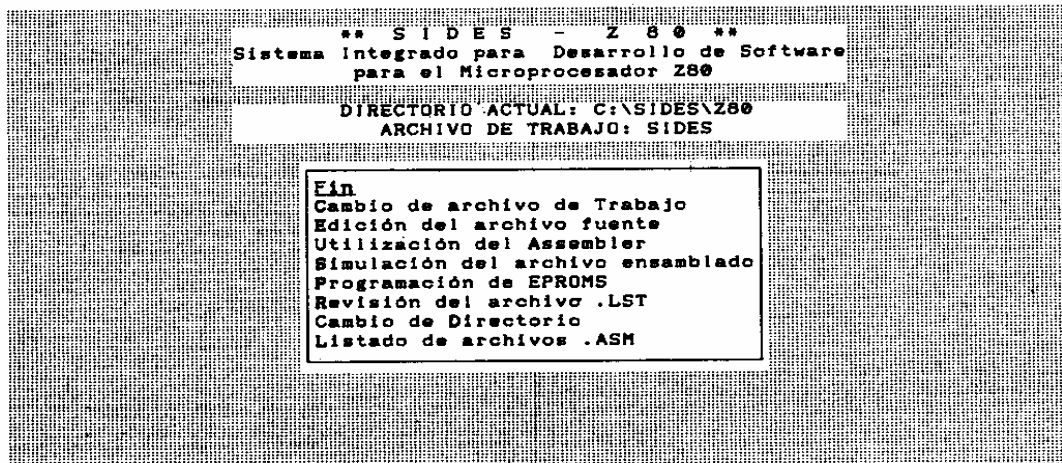


Fig. 1.6 Menú Principal

- 2.- Escritura del archivo fuente utilizando el NotePad del Sidekick. 3.- Ensamblado del archivo de trabajo

```

CPU "C:Z80.TBL"
;ET OPC OPER COMENTARIOS
ORG 07F0H ;Origen tabla de datos
DFL 87A083FFH ;Lab
DFL 0C1C90000H;UC0 (0 = fin tabla)
ORG 0000H ;Origen
LD HL,07F0H ;Puntero a la tabla
LD C,#15H
LZ: LD A,(HL) ;Cargar dato en A
OR A ;Fin tabla ?
JR Z, FN ;Si, fin
OUT (C),A ;Sacar dato
INC HL ;Modificar punteros
DEC C
JR LZ
FN: JR * ;Lazo sin fin
END

```

Ensamblado del archivo de trabajo

Cross-16 Meta-Assembler PC/MS-DOS
Version 1.12
Copyright (C) 1986 Universal
Cross-Assemblers

Starting pass number 1,
Starting pass number 2
P 0000F 18FE FIN: JR
Warning -- Phase Error
Starting pass number 3
End of Assembly -- No Errors

```

LABEL OPERATION Z80 AVSIM Z80 Simulator/Debugger Vi.30
0000H LD HL,07F0H CPU REGISTERS FLAGS SCL SPD DSP SKP CURSOR
0003H LD C,15H C Accumulator Z PV S N H OFF HI ON OFF MENU
0005H LD A,(HL) 0 00000000:00: 1 1 0 0 0 Cycles:
0007H OR A addr data ALT REGISTERS
0007H JR Z,000FH PC:000F » 18 FE FF FF FF FF FF FF AF:0000 DE:0000
0009H OUT (C),A SP:0000 » 21 F0 07 0E 15 7E B7 28 BC:0000 HL:0000
000BH INC HL 06 ED 79 23 0D 18 F6 18 06 ED 79 23 0D 18 F6 18
000CH DEC C BC:000F » F6 18 FE FF FF FF FF FF FF +↑:
000DH JR 0005H DE:0000 » 21 F0 07 0E 15 7E B7 28 != #1 ( PINS
000FH JR * HL:07F0 » 83 FF C1 C9 00 00 FF FF #1-1 ( Int 1
0011H RST 56 IX:0000 » 21 F0 07 0E 15 7E B7 28 != #1 ( NMI 1
0012H RST 56 IY:0000 » 21 F0 07 0E 15 7E B7 28 != #1 ( IFF1 0
0013H RST 56 Memory Space
0014H RST 56 07F0 87 86 83 FF C1 C9 00 00 cea Lp I/O Address
0015H RST 56 07F8 FF FF FF FF FF FF FF FF #1-1 (
0016H RST 56 0800 FF FF FF FF FF FF FF FF #1-1 (
0017H RST 56 0808 FF FF FF FF FF FF FF FF #1-1 (
0018H RST 56 I/O Space
0019H RST 56 0000 00 00 00 00 00 00 00 00 #1-1 (
001AH RST 56 0008 00 00 00 00 00 00 00 00 #1-1 (
001BH RST 56 0010 C9 C1 FF 83 88 87 00 00 #1-1 (
001CH RST 56 0018 00 00 00 00 00 00 00 00 #1-1 (

```

Fig. 1.7 Pantalla de Simulador AVSIMZ80

- 4.- La pantalla del simulador con el archivo fuente ensamblado, cargado y ejecutado se presenta en la Fig. 1.7 5.- El menú principal del módulo de programación de la memoria es el que se muestra en la Fig. 1.8

```

MAIN MENU
Enter Command

[F1] Help [T] Specify PROM Type
[F2] Set Options [P] Program PROM
[D] Directory [R] Read PROM
[L] Load Disk File [C] Check PROM
[W] Write Disk File [V] Verify PROM
[X] Transfer Disk File [ESC] Exit to System

[S] Show Memory Data
[F] Fill Memory Data
[M] Move Memory Data
[B] Patch Memory Data
[K] Checksum

DEVICE: 2732 (21v) ALGO: Slow SPLIT:Off
UNIVERSAL PROM PROGRAMMER AMS, Inc.

```

Fig. 1.7 Menú Principal del PROM 2000

6.- El programa grabado en una memoria Intel de 32 Kb se probó satisfactoriamente en el prototipo experimental.

RESULTADOS

Luego de cuatro semestres consecutivos a lo largo de los cuales el paquete ha ido siendo modificado de acuerdo con las sugerencias de los estudiantes y profesores que lo han utilizado se ha llegado a la versión actual del sistema la cual está siendo usada con resultados satisfactorios en el Laboratorio de Microcomputadores I, a tal punto que en el presente semestre las prácticas de laboratorio se realizan con la guía de un solo instructor con doce estudiantes por sesión distribuidos en grupos de dos en seis computadores, a diferencia de los semestres anteriores en los cuales era necesaria la participación de dos profesores.

La utilización del paquete ha trascendido el campo didáctico y al momento es utilizado para desarrollo de software en aplicaciones prácticas, debido a su facilidad de utilización, rapidez, versatilidad y generalidad.

PROYECCIONES

Los resultados alentadores que se han obtenido han conducido al desarrollo de similares sistemas para otros microprocesadores y microcontroladores populares en el mercado como son: Intel 8086, Intel 8048, Intel 8051.

CONCLUSIONES

Se ha implementado un sistema integrado para desarrollo de software para microprocesadores, el cual contiene todas las herramientas necesarias para realizar todo el proceso de diseño de sistemas basados en microprocesadores. El sistema ha llenado el vacío existente en el campo de software integrado para la enseñanza de microprocesadores, como una herramienta útil tanto en la enseñanza cuanto en el desarrollo de aplicaciones prácticas.

REFERENCIAS

- [1] PETERSON JAMES, Computer Organization And Assembly Language Programming, Academic Press 1978.
- [2] MICROSOFT, AT&T Personal Computer 6300 MS-DOS, Versión 2.11, Agora Resource, Inc., Lexington, 1984.

- [3] UNIVERSAL CROSS-ASSEMBLERS, Cross-16 Meta-Assembler User's Guide, Bedford, Nova Scotia, 1986.
- [4] AVOCET SYSTEMS INC., AVSIM Z80 User's Manual, Avocet Systems, Inc., Rockport, Maine USA., 1986.
- [5] ADVANCED MICROCOMPUTER SYSTEMS, PROM 2000-1 Programmer User's Manual, Advanced Microcomputer Systems, Inc., 1986.
- [6] ZILOG, Components Data Book, Zilog, Inc., Cupertino, California, 1985.
- [7] MONTALVO LUIS, VALENCIA GALD, Hojas Guías para Laboratorio de Microcomputadoras I, EPN, Quito, 1987-1989.
- [8] TORRES EDGAR, Cross-Assemblers en el desarrollo de Sistema basados en Microprocesadores. Estudio, implementación, y ejemplos de aplicación para el Z80/Z80A, Jornadas en Ingeniería Eléctrica y Electrónica, VOL #5, EPN, Quito, Junio 1984.
- [9] BORLAND, SideKick User's Manual, Borland, inc., 1985

BIOGRAFIAS

MONTALVO, LUIS.- Nació en Quito el 3 de Agosto de 1955. Obtuvo el título de Ingeniero en Electrónica y Telecomunicaciones en la Escuela Politécnica Nacional (1982). Becado por la Comisión Fulbright y auspiciado por la Escuela Politécnica Nacional obtuvo el título de MSc. en Computer Engineering en Ohio University, Athens, Ohio (1986). Fue asistente de cátedra en el laboratorio de Robótica en el departamento de Ingeniería Eléctrica y Computación en Ohio University (1984 - 1986). Desde 1981 hasta la fecha trabaja en el área de Sistemas Digitales en el Departamento de Electrónica y Telecomunicaciones de la E.P.N. Sus principales áreas de interés son: Microprocesadores, Diseño VLSI y Procesamiento de Imágenes.

VALENCIA, GALD.- Nació en Quito el 3 de Diciembre de 1960. Obtuvo el título de Ingeniero en Electrónica y Telecomunicaciones en la Escuela Politécnica Nacional (1984). Desde 1982 hasta la fecha trabaja en el área de Sistemas Digitales en el Departamento de Electrónica y Telecomunicaciones de la E.P.N. Sus principales áreas de interés son: Hardware y Software de Computadores Personales, Microprocesadores, Comunicaciones Digitales.