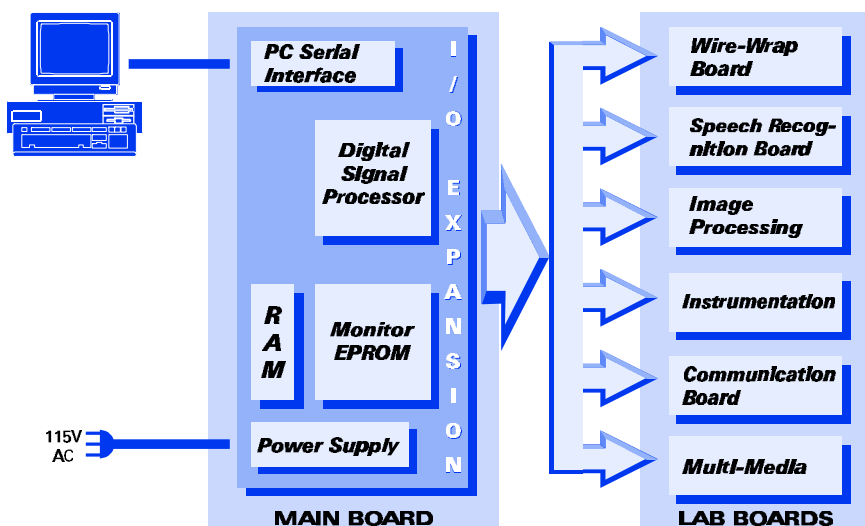


EZ-DSP TutorTM

"State-of-the-Art Teaching Tools"



Advanced
Microcomputer
Systems, Inc.

The EZ-DSP product is published and marketed by Advanced Microcomputer Systems, Inc., 1460 S.W. 3rd Street, Suite B-8, Pompano Beach, Florida, 33069 USA.

Specifications are subject to change without notice.

***When you have finished with the evaluation kit, please pass it on to a friend.
The software in the evaluation kit is not copy protected.***

To order call **(852) 2402 3200**

June 1996, 1st printing

Contents Copyright © 1996 by

Advanced Microcomputer Systems, Inc.

Features of the EZ-DSP Manager Software

The EZ-DSP Manager software is used to communicate (through a host computer or terminal's serial port) with the EZ-DSP CPU-52 microprocessor board (which utilizes a Texas Instruments TMS320BC52 microprocessor).

The EZ-DSP Manager software allows you to modify and display on-board user data/program memory, disassemble program memory (to the screen or dump to a file), assemble source code and download corresponding hex files, execute user programs, set breakpoints and display "watch" variables, modify and display CPU register contents and display the contents of user memory "graphically" on the screen.

Running the EZ-DSP Manager in the demo mode will allow you to evaluate the features and the ease of operation of using this "menu driven" monitor program. Unlike other "monitor" programs that require you to memorize and type numerous commands, the EZ-DSP Manager software has all of the commands listed for you in categorized pull down menus.

Note: Certain options in the pull down menus may be "grayed out" and can not be selected when running in the demo mode.

System Requirements

In order to run the EZ-DSP Manager software, the following equipment is required:

- **IBM personal computer (or compatible)**
- **MS-DOS (version 3.0 or later)**
- **Minimum 640K RAM memory**
- **EGA or VGA graphics adapter card**
- **RS-232 serial interface port**
- **A mouse is suggested but is not required**
- **A hard disk is suggested but is not required**

Introduction to the EZ-DSP Manager and EZ-DSP CPU-52

The EZ-DSP Manager demo, will allow you to evaluate most of the features of the EZ-DSP Manager software. Some of the options in the pull down menus will be "grayed out" and will only function when the EZ-DSP CPU-52 microprocessor board is connected to the serial port of the host computer.

The TMS320BC52 CPU device is the newest member of the 'C5x generation of Texas Instruments TMS320 digital signal processors. The combination of advanced Harvard architecture, on-chip peripherals, on-chip memory and a highly specialized instruction set is the basis of the operational flexibility and speed of the 'C5x devices. They are designed to execute up to 40 MIPS (million instructions per second).

To demonstrate the capabilities of this CPU, the EZ-DSP CPU-52 microprocessor board was designed along with a monitor/debugging program. The EZ-DSP CPU-52 microprocessor board allows the user to debug user code under control of the monitor program. User code is assembled (into an Intel HEX format) using the EZ-DSP CPU Manager software and is then downloaded to the EZ-DSP CPU-52's RAM via a RS-232 serial port. The monitor program is then used to execute and debug the assembled user code.

Features of the EZ-DSP CPU-52 Microprocessor Board

The EZ-DSP CPU-52 provides a low cost method for debugging and evaluating the TMS320BC52 Texas Instruments microprocessor. The 'C52's advanced Harvard type architecture maximizes the processing power by maintaining two separate memory bus structures, program and data, for full-speed execution. Instructions support data transfers between the two spaces. This architecture permits coefficients stored in program memory to be read into the RAM, eliminating the need for a separate coefficient ROM. It also makes available immediate instructions and subroutines based on computed values. Control signals and instructions provide floating-point support, block memory transfers, communication to slower off-chip devices and multiprocessing implementations. The TMS320BC52 microprocessor that is used on the EZ-DSP CPU-52 microprocessor board also offers the following advantages:

Features of the EZ-DSP CPU-52 Microprocessor Board

- TMS320BC52 CPU in a 100 pin thin quad flat package.
- 25-, 35- and 50-ns Single-Cycle Instruction execution time with 3-V or 5-V operation.
- Single-Cycle 16 x 16-Bit Multiply/Add.
- 128K Words total Data/Program space.
- 4K x 16-Bit Single access on-chip program ROM.
- 1K x 16-Bit Dual access on-chip program/data ROM.
- Full-Duplex synchronous serial port for CODEC interface.
- Hardware or software wait state generation.
- On-chip timer for control operations.
- Repeat instructions for efficient use of program space.
- Multiply-by-two and divide-by-two clocking options.
- Block moves for data/program management.
- On-chip scan based emulation logic.
- Low power dissipation and power down modes:
 - 2.35 mA/MIP) at 5V, 40-MHz
 - 3mat 5V, 40-MHz (typical IDLE2)
 - 5 uA at 5V, clocks off (typical standby)

The EZ-DSP CPU-52 microprocessor board is shown below:

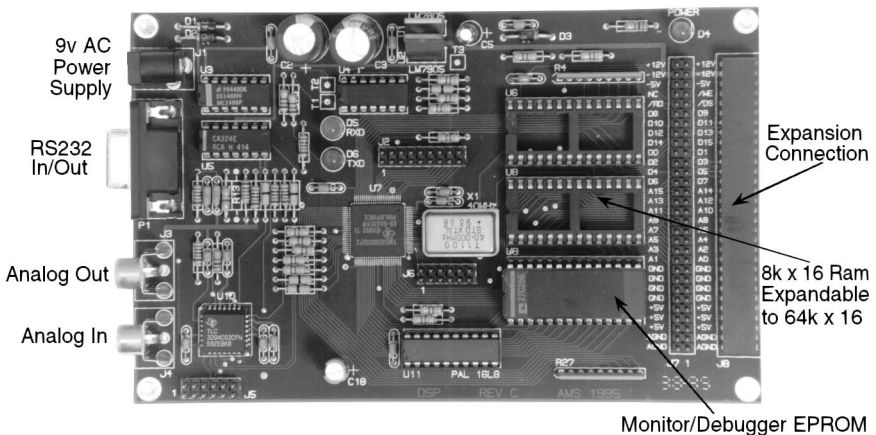


Figure 1: EZ-DSP CPU-52 Microprocessor Board.

An optional wire-wrap section can be provided with the EZ-DSP CPU-52 to provide an area for custom interfacing to the CPU. The wire-wrap section can be detached from the EZ-DSP CPU-52 so that a new wire-wrap section or one of the other EZ-DSP CPU-52 interfacing projects can be connected. This allows for the same EZ-DSP CPU-52 microprocessor board to be used over and over again on many different projects.

TMS320BC52 Memory

The 'C52 implements three separate address spaces for program memory, data memory and I/O. Each space accommodates a total of 64K 16-bit words. Within the 64K words of data space, the 256 to 32K words at the top of the address range can be defined to be external global memory in increments of powers of two, as specified by the contents of the GREG register. Access to global memory is arbitrated using the global memory bus request (/BR) signal.

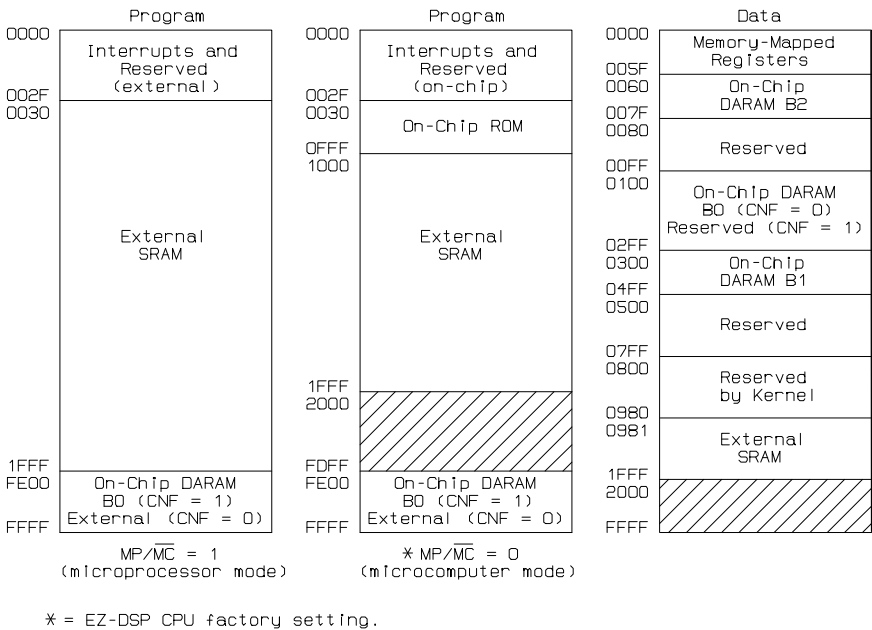


Figure 2: EZ-DSP CPU-52 Memory.

The 'C52 contains 4K words of mask programmable on-chip ROM located in program memory space. Users can arrange for this ROM to be programmed with contents unique to any particular application. The ROM is enabled or disabled by

the state of the MP/MC control input upon resetting the device or by manipulating the MP/MC bit in the PMST status register after reset. The ROM occupies the first 4K words of internal program space (0000h-0FFFh) when enabled; when disabled, these addresses are located in the device's external program memory space.

The 'C52 also provides a total of 1056 16-bit words of on-chip data RAM, divided into three separate blocks (B0, B1 and B2). Of the 1056 words, 544 words (blocks B1 and B2) are always data memory and 512 words (block B0) are programmable as either data or program memory. A data-memory size of 1056 words allows the 'C52 to handle a data array of 1024 words (512 words if on-chip RAM is used for program memory), while still leaving 32 locations for intermediate storage. When using block B0 as program memory, instructions can be downloaded from external program memory into on-chip RAM and then executed.

The CNFD (configure block B0 as data memory) CNFP (configure block B0 as program memory) instructions allow dynamic configuration of the memory maps through software. Regardless of the configuration, code can still be executed from external program memory.

When using on-chip program RAM, ROM, or high speed external program memory, the 'C52 runs at full speed without wait states. Internal RAM is dual access memory, allowing two accesses to be performed in one machine cycle. This, coupled with the parallel nature of the 'C5x architecture, allows for the device to perform three concurrent memory accesses in any given machine cycle. Externally, the READY line can be used to interface the 'C52 to slower, less expensive external memory. Downloading programs from slow off-chip memory to on-chip program RAM can then speed processing while cutting system costs.

On the 'C52, the first 96 (0000h-005Fh) data memory locations are allocated for memory mapped registers. This memory mapped register space contains various control and status registers including those for the CPU, serial port, timer and software wait-state generators. Additionally, the first 16 I/O port locations are also mapped into this data memory space, allowing them to be accessed either as data memory using single-word instructions or as I/O locations with two-word instructions, allowing access to the full 64K words of I/O space.

An optional boot loader (as in the EZ-DSP CPU-52) is available that can be included in the device's on-chip ROM. This boot loader can be used to transfer a program automatically from data memory or the serial port to anywhere in program memory. In data memory, the program can be located on any 1K word boundary and can be in either byte-wide or 16-bit word format. Once the code is transferred, the boot loader releases control to the program for execution.

The 'C52 has a mask programmable option that provides security protection for the contents of on-chip ROM. When this internal option bit is programmed, no

externally originating instruction can access the on-chip ROM. This feature can be used to provide security for proprietary algorithms.

EZ-DSP CPU-52's ADC and DAC Interface

The EZ-DSP CPU-52 microprocessor board provides an analog interface circuit (AIC) using the Texas Instruments TLC320AC02 device (28-pin PLCC package). This device is an audio-band processor that provides an analog-to-digital and digital-to-analog input/output interface system. This device integrates a band-pass switched capacitor antialiasing input filter, a 14-bit resolution analog-to-digital converter (ADC), a 14-bit resolution digital-to-analog converter (DAC), a low-pass switched capacitor output reconstruction filter, $(\sin x)/x$ compensation and a serial port for data and control transfers. Note: The maximum sampling rate for the ADC channel is 43.2 kHz and the maximum rate for the DAC channel is 25 kHz. Two RCA-type connectors have been provided on the EZ-DSP CPU-52 microprocessor board to provide easy access to the ADC and DAC interface. The ADC input connector is labeled J3 and the DAC output connector is labeled J4 on the EZ-DSP CPU-52 microprocessor board.

Demo Software Installation

To install the EZ-DSP Manager software DEMO onto your hard drive, insert the DEMO disk that was provided into your floppy drive. Change the current directory to your floppy drive by typing the following at the prompt:

A: (and press the <ENTER> key)

Note: If your floppy drive is Drive B, then substitute "B:" in the above command. Next, type "INSTALL" and press the <ENTER> key to begin the installation. You will be further instructed on what to do.

To run the demo software, make sure that the current directory is the directory where the demo manager software is located. Refer to the following sections for instructions on starting the demo.

Starting the EZ-DSP Manager Demo

Several files (including several sample assembly language routines) will be included on the EZ-DSP Manager demo disk. A description of each file is shown below for your convenience:

File Name	Description of file.
EZDSP.EXE	EZ-DSP Manager software.
C32C.EXE	TMS320C52 assembler (used internally by the EZ-DSP manager software).
LESSONx.ASM	Sample assembly language programs that correspond to the various chapters in workbook.
LESSx_x.ASM	Sample assembly language programs that correspond to the various chapters in workbook.
DTMF.ASM	Sample assembly language programs that generates DTMF tones (dials 555-1212).
TWID.ASM	"Twiddle" factors used for Lesson 9.
TONE_x.ASM	Contains data that corresponds to a DTMF tone. These can be used with the DTMF tone recognition lesson.
TONE_x.HEX	These are the assembled TONE_x.ASM files that can be downloaded to the EZ-DSP CPU-52 and displayed graphically on the screen.
S941.ASM	Contains data that corresponds to a 941 Hz sinusoidal signal.
S941.HEX	This is the assembled S941.ASM file that can be downloaded to the EZ-DSP CPU-52 and displayed graphically on the screen.
S1477.ASM	Contains data that corresponds to a 1477 Hz sinusoidal signal.
S1477.HEX	This is the assembled S1477.ASM file that can be downloaded to the EZ-DSP CPU-52 and displayed graphically on the screen.

If you wish to run the EZ-DSP Manager demo, you must first change the current directory to the directory where the demo has been installed. This can be done by typing the following at the prompt:

C: (and press the <ENTER> key)

CD\EZDSP (and press the <ENTER> key)

Note: "D:" may be substituted in the above command if you have installed the EZ-DSP Manager demo software on Drive D. After changing to the correct directory, type the following at the prompt:

EZDSP DEMO (and press the <ENTER> key)

and the software will be started in the demo mode. Note: If you are evaluating the EZ-DSP CPU-52 microprocessor board, type "EZDSP" instead of "EZDSP DEMO" to start-up the software in the "normal" mode.

How Does The EZ-DSP Manager Software Work?

The EZ-DSP Manager software makes it possible to communicate directly to the EZ-DSP CPU-52 (Texas Instruments' TMS320BC52 microprocessor) using any IBM compatible "host" computer or terminal that has at least one asynchronous serial port.

All of the commands that are issued to the EZ-DSP CPU-52 microprocessor board are selected from various pull down menus located across the top of the EZ-DSP Manager's presentation screen (Figure 3). Therefore, it is not necessary to memorize and type numerous commands in order to communicate to the CPU using this software.

The EZ-DSP Manager presentation screen will be displayed when the program is initially started and is shown below.

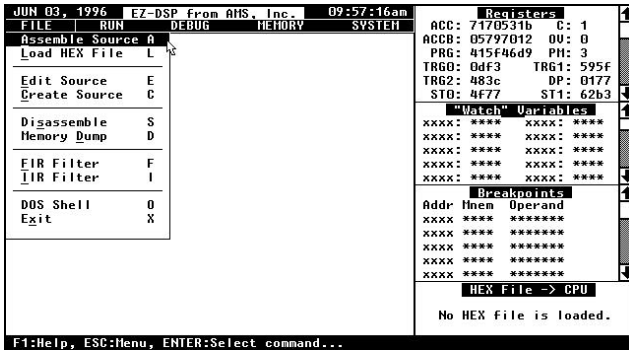


Figure 3: EZ-DSP Manager Presentation Screen.

The EZ-DSP Manager presentation screen is divided into eight areas.

- The "Title Bar", located at the top of the screen, displays the current date and time and also the title of this software.
- The "Action Bar", located just below the Title Bar, contains the various pull down menus where the commands that are issued to the EZ-DSP CPU-52 microprocessor board can be selected. The pull down menus are shown in the following figure:

FILE	RUN	DEBUG	MEMORY	SYSTEM
Assemble Source A Load HEX File L	Go G Continue C Step S Reset R	Modify Registers R Scroll Registers S Add Variable U Remove Variable E Clear All C Scroll Variables O	Memory Dump D Disassemble S Memory Plot P Memory Modify M Block Fill F Block Move U	Configure Port P Configure Colors C Change DIR D Scroll when Bump B About EZ-DSP A
Edit Source E Create Source C		Set Breakpoint B Remove Breakpoint M Clear All L Scroll Breakpoints A		
Disassemble S Memory Dump D				
FIR Filter F IIR Filter I				
DOS Shell O Exit X				

Figure 4: EZ-DSP Pull Down Menus.

One of the pull down menus will normally be pulled down. If a menu is not currently pulled down, you can do one of the following:

- * **Press the ESC key.**
- * **Point at the menu heading with the mouse.**
- * **Press the key combination ALT plus the first letter of the menu's heading.**

Notice that in all three cases a menu will be pulled down. To access an adjacent menu, simply press the right or left arrow keys on the keyboard or point at the menu heading with the mouse.

A menu bar is located within the pull down menu and is used to select one of the commands in the menu. The menu bar can be moved up or down by pressing the up or down arrow keys on the keyboard. Once the menu bar is positioned on the desired command, press the ENTER key to select that command. If you are using a mouse, point at the command and click the left mouse button to select the command. Listed to the right of each command in the pull down menus are "hot keys" which allow you to select a command (in the current pulled down menu) by simply pressing its hot key on the keyboard.

■ The **"Registers"** display window is located in the upper right corner of the EZ-DSP Manager presentation screen. This window is used to display the current EZ-DSP CPU-52 register contents. There are a total of forty-two registers available for the EZ-DSP CPU-52. To scroll the list of the registers, use the "scroll bar" located to the right of the Registers window.

■ The **"Watch Variables"** display window is located below the Registers display window. This window is used to monitor the contents of several memory locations at once while running/debugging the assembled code. There are a total of twenty-four watch variables that can be defined in the EZ-DSP Manager software. To scroll the list of the watch variables, use the "scroll bar" located to the right of the Watch Variables window.

■ The **"Breakpoints"** display window is located below the Watch Variables display window. This window is used to display the current user breakpoints that have been entered into the breakpoint address table. There are a total of sixteen breakpoints that can be defined in the EZ-DSP Manager software. To scroll the list of the breakpoints, use the "scroll bar" located to the right of the Breakpoints window.

■ The **"HEX File -> CPU"** display window is located below the Breakpoints display window. This window is used to display the name of the last assembled

320C5X hex file that has been downloaded into user memory on the EZ-DSP CPU-52 microprocessor board.

■ The **"Main Display"** window is the large area located in the left half of the EZ-DSP Manager presentation screen. This window is used to display the EZ-DSP CPU-52 memory contents, disassembled memory listing, graphical plot of user memory and to edit/create an assembly language source file.

■ The **"Status Line"** display window is located at the bottom of the EZ-DSP Manager presentation screen. This window is used to display the current command that has been selected and any options that may be available for that command.

Getting Help

On-line help is always available by pressing the F1 key and is context sensitive. That is, the help dialog box that appears will be concerning the current highlighted command. For example, highlight a command (using the menu bar) in one of the pull down menus and press the F1 key. A "help" dialog box will appear explaining how the command is to be used.

EZ-DSP Manager Commands and Tutorial

The File Menu

The FILE menu contains commands that will access "file type" operations. That is, these commands will read/write to disk memory of the "host" computer or terminal.

Assemble File

The "Assemble File" command in the FILE menu is used to assemble a Texas Instruments 320C5X assembly language source file into an Intel hex file. The assembled hex file can be later downloaded to the EZ-DSP CPU-52 microprocessor board and executed.

For example, select the Assemble File command from the FILE menu. Since we have not yet assembled a source file, the file name "noname.asm" will be present in the dialog box. Press the ESC key (or the right mouse button) to remove the existing name in the dialog box and press the ENTER key twice (or click the left mouse button twice) to display the available assembly language source files (*.asm).

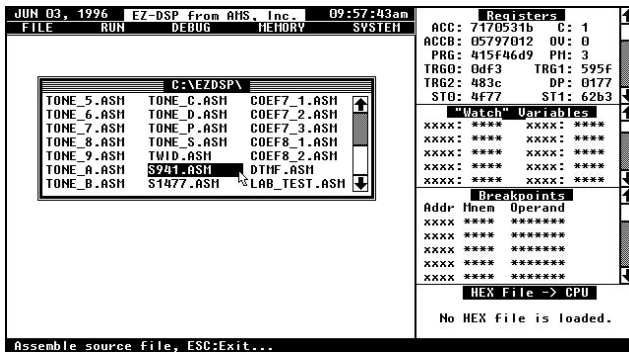


Figure 5: Assembly Language Source Files Listing.

Select the file "s941.asm" by pointing at the file name with the mouse and clicking the left mouse button. The following dialog box will be displayed showing the options that are available for the assembler. Note: The demo version of the EZ-DSP Manager software will not have the option to generate the assembled Intel hex file. However, the necessary "hex" files have been provided on the demo diskette in order to follow along with the tutorials.

The file, "s941.asm", contains data that corresponds to a 941 Hz sinusoidal analog signal that was sampled using ADC logic and stored digitally into memory. Press the OK button (Figure 6) to begin assembling the file. The assembly language source file will be assembled to the file "s941.hex".

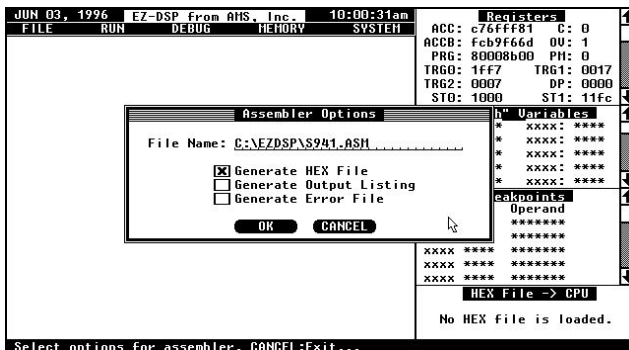


Figure 6: Assembler Options.

Load HEX File

The "Load HEX File" command in the FILE menu is used to download an assembled Texas Instruments 320C5X hex file into user memory located on the EZ-DSP CPU-52 microprocessor board.

Note: The Load HEX File command can also be selected by pointing at the "HEX File -> CPU" window (in the lower right corner of the screen) with the mouse and clicking the left mouse button.

For example, select the Load HEX File command from the FILE menu. Since we have not yet selected an assembled source file, the file name "noname.hex" will be present in the dialog box. Press the ESC key (or the right mouse button) to remove the existing name in the dialog box and press the ENTER key twice (or click the left mouse button twice) to display the available assembled Texas Instruments 320C5X hex files.

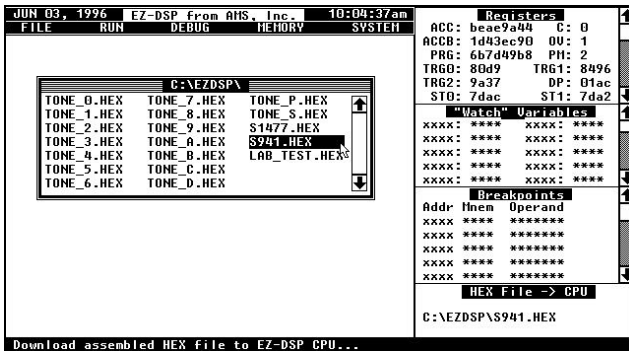


Figure 7: Assembled Hex Files Listing.

Select the hex file "s941.hex" by pointing at the file name with the mouse and clicking the left mouse button. Note: The file "s941.hex" was previously generated in the last section (Assemble File). The hex file will be downloaded into memory on the EZ-DSP CPU-52 and the message "Done..." will be displayed when the transfer has been completed. Also, the name of the last assembled hex file that has been downloaded into user memory on the EZ-DSP CPU-52 microprocessor board will be displayed in the lower right corner of the screen (in the "HEX File -> CPU" display window).

To view the "data" that has been downloaded into memory, select Memory Plot from the MEMORY menu. After selecting the Memory Plot command, you will be prompted to enter the starting hex address to begin the memory plot. Enter the hex address "0400" for this example and press the ENTER key (or click the left mouse button) to accept this number. The Main Display window will expand and a

plot of the contents of memory will be displayed. More information concerning the Memory Plot command will be explained later. When you have finished viewing the memory plot, press the ESC key (or the right mouse button).

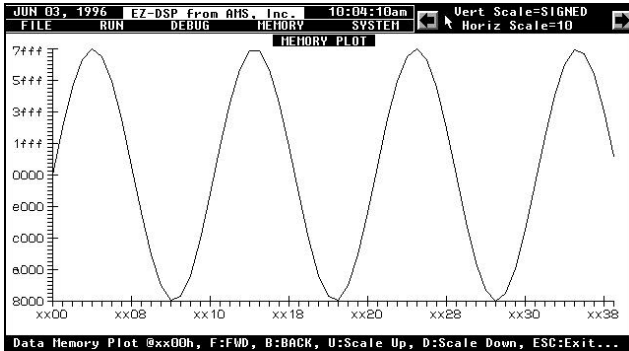


Figure 8: Memory Plot (S941.HEX).

Edit Source

The "Edit Source" command in the FILE menu is used to edit an existing assembly language source file (or any ASCII file).

For example, select the Edit Source command from the FILE menu. Since we have not yet selected a file to edit, the file name "noname.asm" will be present in the dialog box. Press the ESC key (or the right mouse button) to remove the existing name in the dialog box and press the ENTER key twice (or click the left mouse button twice) to display the available assembly language source files.

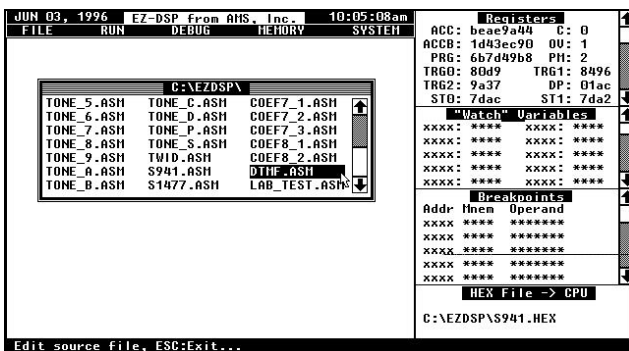


Figure 9: Assembly Language Source Files Listing.

Select the file "dtmf.asm" by pointing at the file name with the mouse and clicking the left mouse button. The Main Display window will now expand to display the assembly language source file. The editor commands can be displayed by pressing the F1 key. Press the F1 key now to display these commands. When you have finished viewing the editor commands, press the ESC key and you will return to the currently loaded assembly language source file to edit.

When you have finished viewing/editing the file, press the ESC key to exit the editor. If any changes were made to the file, you will be prompted to save these changes.

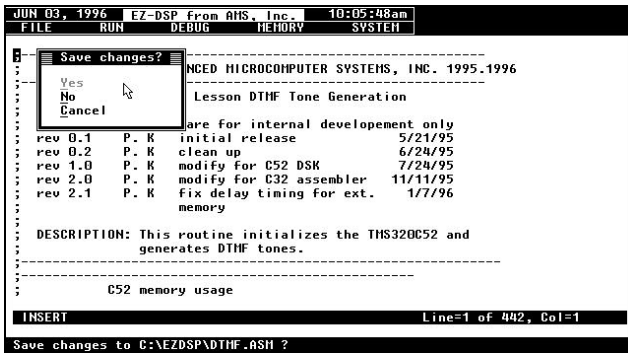


Figure 10: Edit Source File (DTMF.ASM).

Create Source

The "Create Source" command in the FILE menu is used to create a new assembly language source file. After selecting the Create Source command from the FILE menu, you will be prompted to enter the name that will be given to this new file. When you have finished editing the new source file, press the ESC key and you will be prompted to save this newly created file.

Disassemble

The "Disassemble" command in the FILE menu is used to write the disassembled contents of memory to a DOS file. When the Disassemble command is selected, a dialog box will appear allowing you to enter the name of the file (default file name = EZDSP.DMP) to receive the disassembled memory listing. Also, you must enter the starting and ending address range (of memory) to be disassembled. For example, select the Disassemble command from the FILE menu and the following dialog box will appear.



Figure 11: Disassemble Memory to a File.

The file name "ezdsp.dmp" will do for this example. Enter "FE00" for the starting address and enter "FE2F" for the ending address.

Note: When the EZ-DSP Manager is running in the demo mode, there are 256-bytes of memory (0000h-00FFh) available in order to make the commands in the pull down menus appear to function. When the EZ-DSP CPU-52 microprocessor board is connected to the host computer or terminal (via the RS-232 serial port), then there will be:

- 4048-bytes (0030h-0FFFh) on-chip ROM
- 4096-bytes (1000h-1FFFh) external (program memory) SRAM
- 512-bytes (FE00h-FFFFh) on-chip (program memory) DARAM B0
- 32-bytes (0060h-007Fh) on-chip (data memory) DARAM B2
- 512-bytes (0100h-02FFh) on-chip (data memory) DARAM B0
- 512-bytes (0300h-04FFh) on-chip (data memory) DARAM B1
- 5759-bytes (0981h-1FFFh) external (data memory) SRAM

available to the user.

Press the OK button to accept the parameters entered in the dialog box (Figure 11). We can use the EZ-DSP Manager's editor to view the results of the disassembled memory dump that was just generated. Select the Edit Source command from the FILE menu. The name of the last file that was edited, "dtmf.asm", will be displayed in the dialog box that appears. Press the ESC key (or the right mouse button) to remove this file name and enter "ezdsp.dmp". Now press the ENTER key (or click the left mouse button) and a similar disassembled program memory listing should be displayed.

EZ-DSP Disassembled Memory Dump
START_DISASSEMBLE_DUMP

```
fe00: LAR    AR0,0h
fe01: ADD    *0 -,fh,AR1
fe02: XPL    *BR0 -
fe03: LTS    *BR0 -,AR4
fe04: BANZD  7875h,*0+
fe06: CPL    #38c9h,*,AR5
fe08: SMMR   *BR0 -,#d960h,AR2
fe0a: SPLK   #8f8dh,*,AR7
fe0c: SAR    AR0,*0 -,AR1
fe0d: SAR    AR4,* -
fe0e: SACH   37h,2h
fe0f: ***UNKNOWN***
fe10: RETC   NEQ,OV,C,BIO
.
.
.
fe23: CPL    #7879h,* -
fe25: BANZD  74c7h,*0+
fe27: XPL    *,AR2
fe28: ADD    *BR0 -,fh,AR7
fe29: RETCD  LT,OV
fe2a: MPY    #101dh
fe2b: TBLW   37h
fe2c: MAR    30h
fe2d: SAR    AR0,19h
fe2e: SAR    AR7,*,AR6
fe2f: NORM   7ah
END_DISASSEMBLE_DUMP
```

Example 1: EZDSP.DMP (Disassembled Memory Listing).

When you have finished viewing the disassembled memory dump, press the ESC key to exit the editor.

Memory Dump

The "Memory Dump" command in the FILE menu is used to write the hexadecimal contents of memory to a DOS file.

When the Memory Dump command is selected, a dialog box will appear allowing you to enter the name of the file (default file name = EZDSP.HEX) to receive the hexadecimal memory listing. Also, select the memory type (Data or Program memory) to be extracted from the EZ-DSP CPU-52 microprocessor board and

enter the starting and ending address range (of memory) that will be included in this memory dump.

For example, select the Memory Dump command from the FILE menu and the following dialog box will appear.

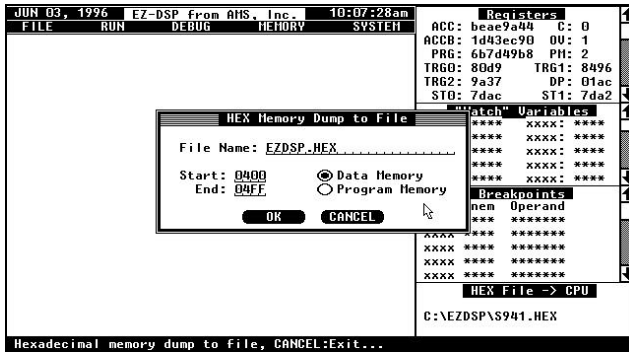


Figure 12: Hexadecimal Memory Dump to a File.

The file name "ezdsp.hex" will do for this example. Enter "0400" for the starting address and enter "04FF" for the ending address and select "data memory" for the memory type.

Press the OK button to accept the parameters entered in the dialog box (Figure 12). We can use the EZ-DSP Manager's editor to view the results of the memory dump that was just generated. Select the Edit Source command from the FILE menu. The name of the last file that was edited, "ezdsp.dmp", will be displayed in the dialog box that appears. Press the ESC key (or the right mouse button) to remove this file name and enter "ezdsp.hex". Now press the ENTER key (or click the left mouse button) and the following hexadecimal listing should be displayed.

```
EZ-DSP HEX "Data Memory" Dump
START_HEX_DUMP
0000: 0000 2fd9 58c2 74cc 7fe6 7875 5f8d 38c9
0008: 09ca d960 ae8f 8f8d 80d9 8496 9a37 be9b
0010: ec7b 1d2e 49a7 6b72 7da8 7da6 6b6c 499e
0018: 1d23 ec70 be92 9a31 8493 80db 8f92 ae97
0020: d96a 09d4 38d3 5f94 7879 7fe6 74c7 58ba
0028: 2fcf fff6 d01d a737 8b30 8019 878e a07a
0030: c740 f641 26aa 517a 7078 7f28 7b67 65c2
0038: 415b 137b e2c8 b650 9489 8256 825c 949a
0040: b66b e2e7 139a 4177 65d6 7b70 7f24 7069
0048: 5161 268c f621 c724 a065 8783 801a 8b3d
0050: a74e d03b 0015 2fec 58d1 74d5 7fe7 786e
0058: 5f7f 38b6 09b4 d94b ae7e 8f83 80d7 849b
0060: 9a44 beae ec90 1d43 49b8 6b7d 7dac 7da2
0068: 6b60 498d 1d0f ec5b be80 9a24 848d 80dd
0070: 8f9c aea8 d97e 09ea 38e6 5fa2 7880 7fe5
0078: 74bf 58aa 2fbb ffe0 d00a a727 8b27 8018
0080: 0000 0000 0000 0000 0000 0000 0000 0000
0088: 0000 0000 0000 0000 0000 0000 0000 0000
      .
      .
      .
00e0: 0000 0000 0000 0000 0000 0000 0000 0000
00e8: 0000 0000 0000 0000 0000 0000 0000 0000
00f0: 0000 0000 0000 0000 0000 0000 0000 0000
00f8: 0000 0000 0000 0000 0000 0000 0000 0000
END_HEX_DUMP
```

Example 2: EZDSP.HEX (Hexadecimal Memory Dump).

When you have finished viewing the hexadecimal memory dump, press the ESC key to exit the editor.

FIR Filter Design

This option is included only in the EZ-DSP Advanced version.

IIR Filter Design

This option is included only in the EZ-DSP Advanced version.

DOS Shell

Use this command to temporarily shell to DOS. While in the DOS shell, you may execute any of the DOS commands. However, depending on the amount of memory that is available, you may not be able to run or execute other programs while in this DOS shell. You will "shell" out into the same directory that the EZ-DSP Manager software is running. The prompt that is displayed will not show you the current directory that you are in. Instead, the following prompt will appear:

DOS>

To return to the EZ-DSP Manager, simply ENTER a "blank line" at the prompt.

Exit

In order to "exit" the EZ-DSP Manager software normally, select this command. After selecting this command, you will return to the DOS prompt.

Note: If you are evaluating the EZ-DSP CPU-52 microprocessor board and you are running the EZ-DSP Manager software in the "normal" mode, then any changes that have been made to the host serial port (i.e. baud rate, parity, word length and stop bits) using the Configure Port option will be saved to the configuration file (ezdsp.cfg). These serial port parameters will be used again the next time the program is started.

The Run Menu

The RUN menu contains commands that are used when executing user code on the EZ-DSP CPU-52 microprocessor board. Note: All of the commands in the RUN menu will be "grayed out" when running the EZ-DSP Manager software in the demo mode. The EZ-DSP CPU-52 hardware must be present in order to select any of the commands in the following section.

Go

The "Go" command in the RUN menu allows the user to initiate user program execution (free run in real time). The user must specify a starting address where execution is to begin. Program execution continues until a breakpoint is encountered or until the ESC key is pressed. If a valid breakpoint is not reached by the executed program, then you will need to press the ESC key in order to stop program execution.

Note: The contents of the Registers and Watch Variables windows are only updated after program execution has stopped.

Example: If you are evaluating the EZ-DSP CPU-52 microprocessor board and Keypad/Display Lab board, then you will want to first make sure that the 9 volt AC power supply is NOT connected to the EZ-DSP CPU-52 microprocessor board (connector J1) at this time.

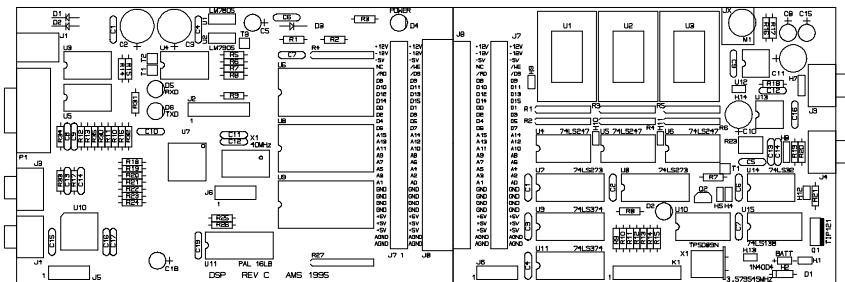
Connect the Keypad/Display Lab board to the EZ-DSP CPU-52 microprocessor board via the 60 pin right angle connectors (J8).

Connect header J6 of the EZ-DSP CPU-52 microprocessor board to J6 of the Keypad/Display Lab board with the supplied ten conductor flat ribbon cable. The ends of the ten conductor flat ribbon cable have a black plastic IDC (10-pin) connector. Pin number 1 of this ID connector is designated by an arrow on the connector itself. **IMPORTANT:** Make sure that pin number 1 of J6 on the EZ-DSP CPU-52 microprocessor board is connected to pin number 1 of J6 on the Keypad/Display Lab board.

Connect the supplied twenty conductor keypad cable to header K1 on the Keypad/Display Lab board. Make sure that pin number 1 of the ID (20-pin) keypad cable connector is oriented correctly to match pin number 1 of header K1.

Connect the 9 pin "DB type" male connector of the supplied RS-232 serial cable to connector P1 of the EZ-DSP CPU-52 microprocessor board and connect the other end (9 pin "DB type" female) to the serial port of the host computer or terminal.

Now you are ready to re-connect the 9 volt AC power supply to connector J1 of the EZ-DSP CPU-52 microprocessor board.



Start-up the EZ-DSP Manager software and assemble the source file "lab_test.asm" by selecting the Assemble Source command from the FILE menu. After the source file has been assembled, select Load HEX File from the FILE menu and select the assembled hex file "lab_test.hex" to be downloaded to the EZ-DSP CPU-52 microprocessor board.

Next, select Go from the RUN menu and enter "FE00" as the address of the first instruction to be executed. This will begin the execution of the program that we have downloaded to the EZ-DSP CPU-52. The "lab_test" program will scan the keypad (on the Keypad/Display Lab board) for user input. Pressing a key on the keypad will display the corresponding number on all three LED displays.

To stop the program execution, press the ESC key on the keyboard of the host computer or terminal. The EZ-DSP Manager software will sense that the program execution has stopped and will automatically exit from the Go command. After the program execution has been stopped, the EZ-DSP Manager will display a disassembled listing of the downloaded user code starting with the next instruction that was to be executed before the program was terminated.

To demonstrate the ability of the EZ-DSP CPU-52 to stop program execution at a breakpointed address, select the Set Breakpoint command from the DEBUG menu and enter the breakpoint address "FEBD". This address corresponds to an instruction (in the program that was downloaded to the EZ-DSP CPU-52) that senses when the number "0" is pressed on the keypad. Select Go once more from the RUN menu and enter "FE00" as the address of the first instruction to be executed. Press all of the numbers on the keypad (except for "0"). Notice that everything is functioning as before. Now press the "0" key and the EZ-DSP Manager software will sense that a breakpoint has been encountered and program execution will be stopped.

Proceed

The "Proceed" command in the RUN menu is used to proceed or continue program execution without having to remove assigned breakpoints. This command is used to bypass assigned breakpoints in a program executed by the GO command. The program will continue to "proceed" to the next assigned breakpoint. If a valid breakpoint is not reached by the executed program, then you will need to press the ESC key in order to stop program execution.

Note: The contents of the Registers and Watch Variables windows are only updated after program execution has stopped.

The EZ-DSP Manager software will sense when the next valid breakpoint is reached and will automatically exit from the Proceed command.

Step

The "Step" command in the RUN menu is used to execute each instruction one at a time. Execution starts at the current program counter (or PC-register) address. Each instruction that is executed will be displayed in the Main Display window. To execute the next instruction, simply press the PgDn key on the keyboard (or point at the down arrow button with the mouse and click the left mouse button). The contents in the Registers and Watch Variables display windows will be updated after each instruction that is executed.

To scroll the register or watch variable contents while stepping the program execution, use the "R" and "W" push buttons. For example, point at the push-button labeled "R" with the mouse and click the left mouse button. This will enable the mouse to move freely in the Registers window. While in the Registers window, you will be able to access the register's scroll bar in order to scroll the list of the register's contents. When you have finished viewing the register's contents, press the ESC key (or the right mouse button) to resume stepping the program execution.

To exit from the STEP command, press the ESC key (or the right mouse button) while in the Main display window.

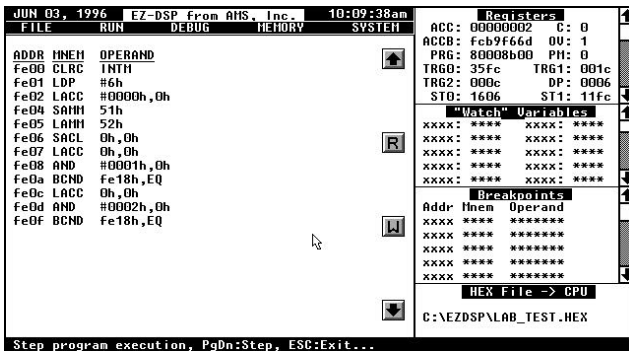


Figure 14: Step Program Execution.

Reset

The "Reset" command in the RUN menu is used to initiate a hardware reset on the EZ-DSP CPU-52 microprocessor board. If you find that the EZ-DSP Manager software is no longer communicating properly with the EZ-DSP CPU-52 microprocessor board, then select this command.

The Debug Menu

The DEBUG menu contains commands that are used for debugging/executing user code downloaded to the EZ-DSP CPU-52 microprocessor board.

Modify Registers

The "Modify Registers" command in the DEBUG menu is used to modify the EZ-DSP CPU-52 register contents. The EZ-DSP CPU-52 microprocessor board register contents are "mapped" to external "data memory". The addresses of these memory mapped registers are as follows:

Register	Memory Location	Register	Memory Location
ACC	0060h	AR0	0010h
ACCB	0062h	AR1	0011h
PRG	0064h	AR2	0012h
ST0	0066h	AR3	0013h
ST1	0067h	AR4	0014h
PC	0069h	AR5	0015h
		AR6	0016h
St0	0069h	AR7	0017h
St1	006ah		
St2	006bh	INDX	0018h
St3	006ch	BMAR	001fh
St4	006dh	DRR	0020h
St5	006eh	DXR	0021h
St6	006fh	SPCR	0022h
TIM	0074h	PRD	0025h
		TCR	0026h
IMR	0004h	CWSR	002ah
GRG	0005h		
IFR	0006h	C	Bit 9 of ST1 register
PMST	0007h	DP	Bits 0 to 8 of ST0 register
TRG0	000ch	OV	Bit 12 of ST0 register
TRG1	000dh	PM	Bits 0 and 1 of ST1 register
TRG2	000eh		
DBMR	000fh		

TABLE 1: EZ-DSP CPU-52 Memory Mapped Registers.

Note: The Modify Registers command can also be selected by pointing at the Registers window (in the upper right corner of the screen) with the mouse and clicking the left mouse button.

When this command is selected, a dialog box will appear displaying the current register contents. Use the up or down keyboard arrow keys to select (or highlight) one of the registers that you want to modify. Once the desired register is selected, you may begin typing to enter the new hexadecimal value for the register. If you are using a mouse, simply point at the register that you want to modify and click the left mouse button.

When you have finished modifying the registers, press the HOME key to select the OK button and then press ENTER. If you are using a mouse, point at the OK button and click the left mouse button. Once the OK button is selected, the register contents (displayed on the screen and on the EZ-DSP CPU-52 microprocessor board) will be updated.

If you do not want to save the changes that were made to the registers, then press the END key to select the CANCEL button and then press ENTER to discard the changes. If you are using a mouse, point at the CANCEL button and click the left mouse button.

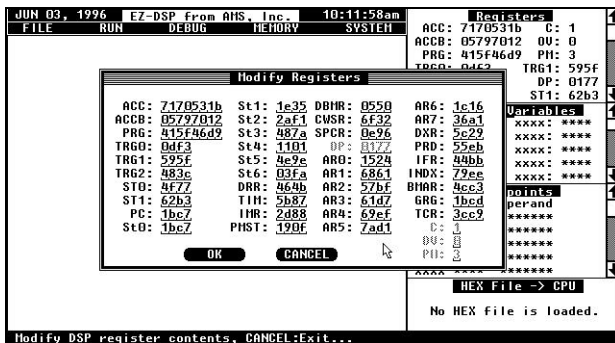


Figure 15: Modify Registers.

Scroll Registers

The "Scroll Registers" command in the DEBUG menu is used to scroll the list of the EZ-DSP CPU-52 register contents that are displayed in the Registers window. This command would be used when a mouse is not available. After selecting this command, the keyboard can be used to scroll the list of registers. Pressing the up or down arrows keys will scroll the display one position at a time. Using the PgUp and PgDn keys will scroll the registers "one page" at a time.

Add Variable

The "Add Variable" command in the DEBUG menu is used to add watch variable addresses into the watch variable's address table. A watch variable is simply a memory location on the EZ-DSP CPU-52 microprocessor board that you wish to monitor the contents of.

The EZ-DSP Manager software allows the user to enter up to twenty-four addresses (or memory locations) into the watch variable's address table. The current watch variable addresses are automatically saved when you exit the EZ-DSP Manager software (by selecting the Exit command in the FILE menu). This saves the user from having to re-enter the list of watch variable addresses each time the EZ-DSP Manager software is run.

To demonstrate this feature, select the Add Variable command from the DEBUG menu and you will be prompted to enter a hex address. Enter the following hex addresses: 040A, 040B, 040C, 040D, 040E and 040F. After you have entered the last hex address, press the ESC key (or the right mouse button) to exit the Add Variable command. Upon pressing the ESC key, you should notice that the Watch Variables window now displays the six addresses (along with the contents of each memory location) that were just entered.

Remove Variable

The "Remove Variable" command in the DEBUG menu is used to remove the address of a specific watch variable from the watch variable's address table. When prompted, enter the hex address of the watch variable that you want to remove.

Clear All Variables

The first "Clear All" command in the DEBUG menu is used to remove all of the watch variable addresses that have been entered into the watch variable's address table.

Scroll Variables

The "Scroll Variables" command in the DEBUG menu is used to scroll the list of the watch variable addresses and contents that are displayed in the Watch Variables window. This command would be used when a mouse is not available. After selecting this command, the keyboard can be used to scroll the list of watch variables. Pressing the up or down arrows keys will scroll the display one position at a time. Using the PgUp and PgDn keys will scroll the list "one page" at a time.

Set Breakpoint

The "Set Breakpoint" command in the DEBUG menu is used to place an address into the breakpoint address table. When prompted, enter the hexadecimal address

of where the breakpoint is to be inserted. A maximum of sixteen breakpoints may be set and will be displayed in the Breakpoints window.

Whenever the Go or Proceed commands are selected (from the RUN menu), then "breakpoints" are inserted into the user code at the address specified in the breakpoint address table. During user program execution, the program will stop execution immediately preceding the execution of any instruction's address that is in the breakpoint address table.

Breakpoints are implemented by placing a TRAP instruction at each address specified in the breakpoint address table. The TRAP service routine saves and displays the internal machine state, then restores the original opcodes at the breakpoint location before returning control back to the EZ-DSP Manager program.

Note: TRAP opcodes cannot be executed or breakpointed in user code because the monitor program uses the TRAP vector. Only RAM locations can be breakpointed. Branch on self instructions can not be breakpointed.

Remove Breakpoint

The "Remove Breakpoint" command in the DEBUG menu is used to remove one of the breakpoints that have been set in the breakpoint address table. When prompted, enter the address of the breakpoint that is to be removed from this list.

Clear All Breakpoints

The second "Clear All" command in the DEBUG menu is used to remove all of the breakpoints that have been set in the breakpoint address table.

Scroll Breakpoints

The "Scroll Breakpoints" command in the DEBUG menu is used to scroll the list of the breakpoint addresses that are displayed in the Breakpoints window. This command would be used when a mouse is not available. After selecting this command, the keyboard can be used to scroll the list of the breakpoint addresses. Pressing the up or down arrows keys will scroll the display one position at a time. Using the PgUp and PgDn keys will scroll the list "one page" at a time.

The Memory Menu

The MEMORY menu contains commands that are used for modifying/displaying user memory located on the EZ-DSP CPU-52 microprocessor board.

Memory Dump

The "Memory Dump" command in the MEMORY menu is used to display the hexadecimal contents of user memory (located on the EZ-DSP CPU-52 microprocessor board) to the screen.

For example, select the Memory Dump command from the MEMORY menu and you will be prompted to select the "type" of memory (Data or Program) and to enter the starting memory location to begin the hexadecimal memory dump. Select "data memory" and enter the hex address "0400" for this example. Next, point at the OK button with the mouse and click the left mouse button (or press the HOME key to select the OK button and then press the ENTER key) to accept these parameters.

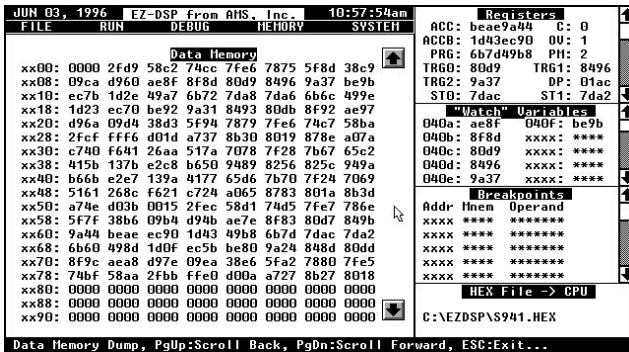


Figure 16: Hexadecimal Memory Dump.

Note: The memory addresses will appear as "XX00" only during the demo mode. This is done since there are only 256-bytes of memory (0000h-00FFh) available when running the EZ-DSP Manager in the demo mode.

Use the PgUp and PgDn keys to view the previous or next page of the hexadecimal memory dump (or point at the up and down arrow buttons with the mouse and click the left mouse button). When you have finished viewing the hexadecimal memory dump, press the ESC key (or the right mouse button).

Disassemble

The "Disassemble" command in the MEMORY menu is used to display a disassembled listing of the program code to the screen. To demonstrate this feature, select the Disassemble command from the MEMORY menu. After the Disassemble command is selected, you will be prompted to enter the address of the first line of code (or instruction) to be disassembled. Enter the hex address "FE00" for this example.

Use the PgDn and PgUp keys to scroll the disassembly listing forward and backwards (or point at the down and up arrow buttons with the mouse and click the left mouse button). Note: You will not be able to scroll backwards past the first instruction that is disassembled. Press the ESC key (or the right mouse button) to exit the disassemble command.

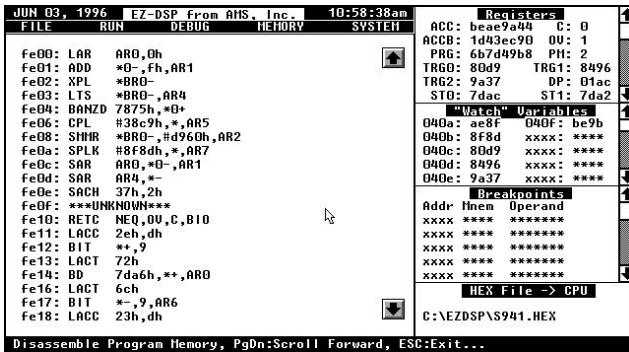


Figure 17: Disassembled Memory Dump.

Memory Plot

The "Memory Plot" command in the MEMORY menu allows the user to view the contents of memory "graphically" on the screen. This is especially useful when sampling an analog signal and storing the resulting data into RAM.

To demonstrate this feature, first select the Load HEX File command from the FILE menu. When prompted, enter the file name "s941.hex" and press ENTER (or the left mouse button) to load the hex file into memory.

Next, select the Memory Plot command from the MEMORY menu and you will be prompted to enter the starting address of memory to be plotted. Enter "0400" and press the ENTER key (or the left mouse button). The Main Display window will expand and a plot of the contents of memory will be displayed.

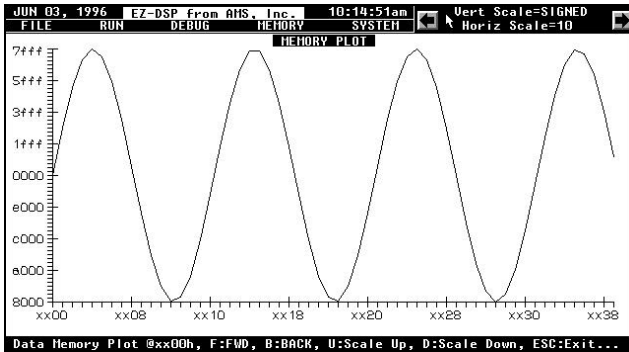


Figure 18: Memory Plot.

The vertical axis represents the amplitude (or hex value) of the data in memory. The vertical axis can be changed to plot signed or unsigned data by pressing the letter "V". The current vertical axis "type" will be displayed in the upper right corner of the screen.

The horizontal axis represents the memory address location, starting with the first address specified. The horizontal axis can be scaled UP (to display fewer bytes of data) or scaled DOWN (to display more bytes of data) by pressing the "U" and "D" keyboard keys respectively. The current horizontal axis "scale" will also be displayed in the upper right corner of the screen. Pressing the letter "B" and "F" on the keyboard (or pointing at the left and right arrow buttons with the mouse and clicking the left mouse button) will go backwards and forwards in memory.

Pressing the letter "A" on the keyboard will allow you to specify a new starting address to begin the memory plot.

When you have finished viewing the "graphical" plot of memory on the screen, press the ESC key (or the right mouse button) to exit the Memory Plot command.

Memory Modify

The "Memory Modify" command in the MEMORY menu allows the user to modify the contents in user memory beginning at the specified address. To demonstrate this feature, select the Memory Modify command from the MEMORY menu. When prompted, select the memory type as "data memory" and enter "0400" as the starting hexadecimal address of memory to begin modifying. Next, point at the OK button with the mouse and click the left mouse button (or press the HOME key to select the OK button and then press the ENTER key) to accept these parameters.

You will then be prompted to enter the new contents for this memory location. Enter "0000" when prompted. After entering "0000", you will be prompted to enter the contents for the next memory location. This time enter "1111". Continue to enter the following four hex values when prompted: "2222", "3333", "4444" and "5555".

Note: Pressing the ENTER key without entering a value will leave the current memory location unchanged and increment to the next memory location. To exit the Memory Modify command, simply press the ESC key (or the right mouse button).

After entering the last hex value ("5555"), press the ESC key to exit the Memory Modify command. To view the changes that have been made to the six memory locations (0400h-0405h), select the Memory Dump command from the MEMORY menu. When prompted, select the memory type as "data memory" and enter "0400" as the starting hex address to begin the memory dump.

The following figure shows this memory dump. Notice that the first six memory locations show the hex numbers that were just entered. When you have finished viewing the hexadecimal contents of memory, press the ESC key (or the right mouse button).

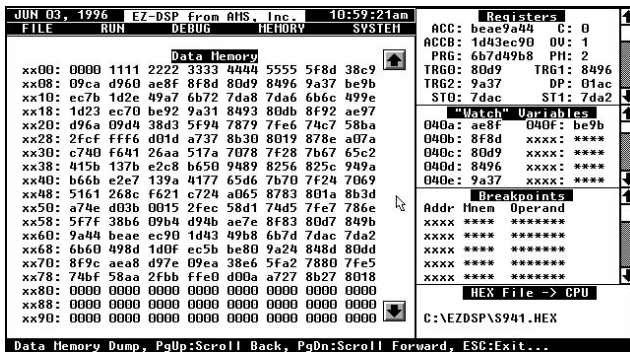


Figure 19: Memory Dump after Modifying Memory.

Block Fill

The "Block Fill" command in the MEMORY menu allows the user to repeat a specific pattern throughout a determined user memory range. To demonstrate this feature, select the Block Fill command from the MEMORY menu and the following dialog box will be displayed.

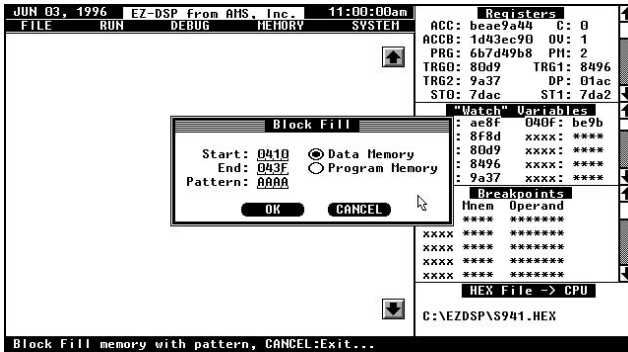


Figure 20: Block Fill Memory with Pattern.

Select the memory type as "data memory", enter "0410" for the starting address location, enter "043F" for the ending address location and enter "AAAA" for the pattern. Point at the OK button with the mouse and click the left mouse button to accept the parameters you have just entered.

To view the changes that have been made to the memory locations (0410h-043Fh), select the Memory Dump command from the MEMORY menu. Select the memory type as "data memory" and enter "0400" as the starting hex address to begin the memory dump. The following figure shows this memory dump. Notice that memory locations XX10h to XX3Fh are filled with the value AAAAh. When you have finished viewing the hexadecimal contents of memory, press the ESC key (or the right mouse button).

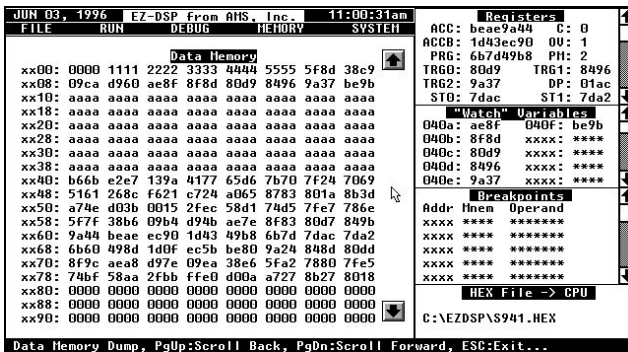


Figure 21: Memory Dump after Filling Block of Memory.

Block Move

The "Block Move" command in the MEMORY menu allows the user to copy or move a "block" of memory to a new memory location. To demonstrate this feature, select the Block Move command from the MEMORY menu and the following dialog box will be displayed.

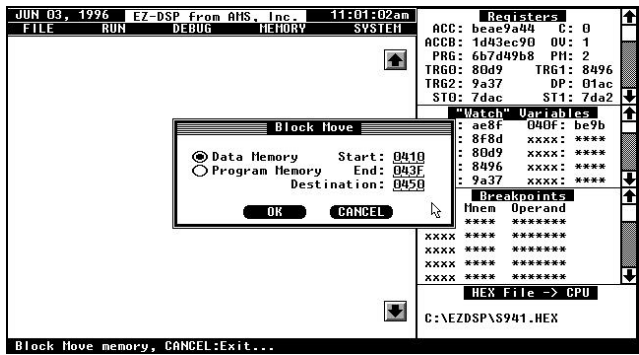


Figure 22: Block Move Memory.

Select the memory type as "data memory", enter "0410" for the starting address location and enter "043F" for the ending address location of the block of memory to move. Enter "0450" for the destination hex address to receive the block (48-bytes) of memory. Point at the OK button with the mouse and click the left mouse button to accept the parameters you have just entered.

To verify that the block of memory (0400h-043Fh) was copied to 0450h, select the Memory Dump command from the MEMORY menu. Select the memory type as "data memory" and enter "0400" as the starting hex address to begin the memory dump. The following figure shows this memory dump. Notice that memory locations XX50h to XX7Fh are filled with the value AAAAh. When you have finished viewing the hexadecimal contents of memory, press the ESC key (or the right mouse button).

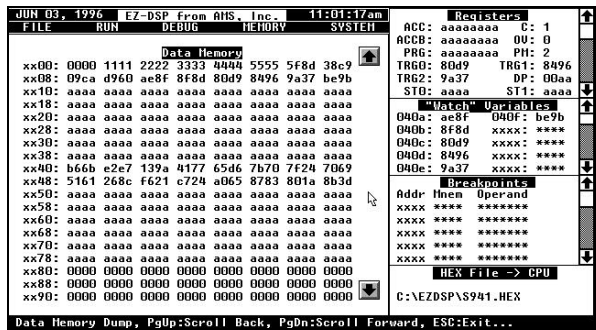


Figure 23: Memory Dump after Moving Block of Memory.

The System Menu

The SYSTEM menu contains commands that are used for configuring options that concern the operation of the EZ-DSP Manager software.

Configure Port

The "Configure Port" command in the SYSTEM menu is used to configure the COM port (or serial port) of the host computer or terminal for communicating with the EZ-DSP CPU-52 microprocessor board.

When the EZ-DSP CPU-52 is first powered-on, its serial port will be configured to 9600 baud, no parity, eight bit word length and one stop bit. The options that you select for the host serial port must match these settings (9600,N,8,1) exactly or you will not be able to communicate with the EZ-DSP CPU-52. Note: When the EZ-DSP Manager software is started (in the "normal" mode) for the first time, the Configure Port dialog box will automatically be displayed so that you can select the COM port that you will be using to communicate with the EZ-DSP CPU-52 microprocessor board.

When the "Configure Port" command is selected from the SYSTEM menu, a dialog box will appear allowing you to select the port address, baud rate, parity, word length and stop bit parameters. The options that are selected in the Configure Port dialog box will be saved to the EZ-DSP Manager's configuration file (ezdsp.cfg) when you exit the program normally. That is, by selecting Exit from the FILE menu. Once the serial port options have been selected, the Configure Port dialog box will not be displayed again when the EZ-DSP Manager software is re-started.

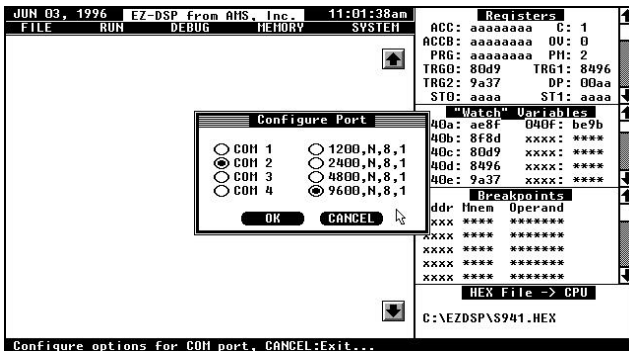


Figure 24: Configure Port Options.

Configure Colors

The "Configure Colors" command in the SYSTEM menu is used to configure the screen colors of the EZ-DSP Manager's pull down menus, dialog boxes and display windows. Changing the screen colors may require that you exit the software and re-start it for the changes to take affect. The screen color selections are saved to the configuration file "ezdsp.clr".



Figure 25: Configure Screen Colors.

Change DIRectory

The "Change DIR" command in the SYSTEM menu is used to change the default search path when the EZ-DSP Manager is searching for files. Note: The default search path is the directory where the EZ-DSP Manager program is currently running.

For example, select the Assemble File command from the FILE menu. Press the ESC key (or the right mouse button) to remove the existing file name in the dialog box. Next, press the ENTER key twice (or click the left mouse button twice) and a listing of the available assembly language source files (in the "c:\ezdsp" directory) will be displayed. The directory "c:\ezdsp" is the default search path.

If we wanted, we could move all of the assembly language source files (*.asm) into a directory called "c:\lessons". We would then use the Change DIR command to change the search path to "c:\lessons". Now when the Assemble File command is selected, the EZ-DSP Manager will search the directory "c:\lessons" for any source files.

Scroll When Bump

A check-mark next to this option in the SYSTEM menu will allow the user to scroll a Memory Dump display or a Disassembled memory dump by simply touching the top or bottom of the MAIN display window with the mouse. To

enable or disable this feature, simply select this option in the menu. This will toggle the check-mark (on and off) for this option.

About EZ-DSP

Selecting this command will display the version of the EZ-DSP Manager software that is currently running.

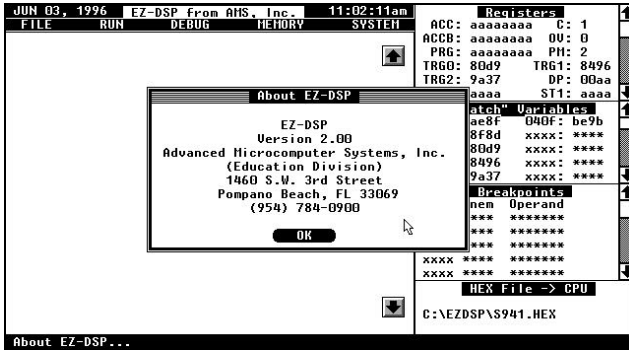
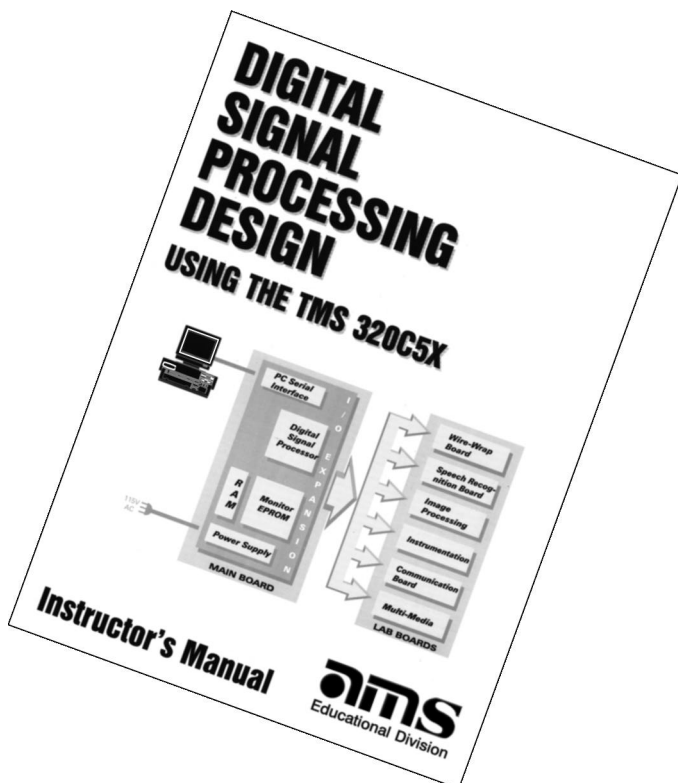


Figure 26: About EZ-DSP.

Features of the EZ-DSP Workbook



1 Introduction to DSP chips and DSP system design process

■ In this chapter, the student will learn about:

1. The basics of real time DSP and the system design alternatives.
2. The different types of DSP chips available and their architectural features.
3. The fixed point and floating point internal data representations.
4. The applications of each type of DSPs.
5. The choice of DSP subsystems and the process of designing a DSP based system.

2 How to set up and use the evaluation board

■ In this chapter, the student will learn about:

1. The overview of software and hardware development tools
2. The architecture and features of the evaluation board
3. How to set up the AMS Evaluation boards
4. Basic EVM commands for resets, help and memory modification
5. How to assemble, single-step and execute the given simple DSP program
6. How to use a host computer to download programs and display results

3 The DSP chip architecture

■ In this chapter, the student will learn about:

1. The internal structure of buses, ALU, MAC, memory and registers
2. The Interrupts, timers and DMA features
3. The basic system configuration and memory interfacing.

4 The DSP chip instruction set

■ In this chapter, the student will learn about:

1. The overview of instruction set and data structures
2. The working of instructions by monitoring the contents of registers and memory
3. How to write and test simple programs for data moving and double precision arithmetic
4. How to interpret fixed-point numbers as fractions and to do scaling
5. How to program for evaluating a sum-of-products without numerical overflow.

5 How the chip interfaces with the outside world

■ In this chapter, the student will learn about:

1. The features of I/O ports and I/O instructions of the chip
2. How to setup the Analog Interface module (AIM) and communicate with Evaluation Board via I/O ports
3. How to change the sampling rate by modifying the given AIM program segment
4. How to hookup tape/microphone, loudspeaker/oscilloscope through AIM
5. Buffering requirements and popular peripheral chips
6. How to acquire digitized music data from the tape and to play it on the speaker.

6 How to program the system for basic DSP operations

- In this chapter, the student will develop simple assembly programs or modify given program segments to perform the following basic DSP operations. The student will determine the memory requirements and the number of processor cycles required for each operation.
 1. Generate and play an Echo signal by adding to the music sample, a properly scaled and delayed sample.
 2. Transfer data blocks between PC host and DSP processor using DMA.
 3. Generate sinusoidal samples of specified frequency and output on the DAC by implementing a simple bi-quad equation.
 4. Convert PCM data to u-law logarithmic compressed data and back, by using the given.
 5. Generate samples of pseudo random binary signal (PRBS) by programming feedback shift register configuration and narrow-width pulse train of given period.

7 Simulation of hearing impairment using real-time FIR filtering

- A tape with specific test sounds is played to demonstrate how a person with hearing impairment actually hears. The digitized audio is played through three different FIR filters to simulate the perceived hearing of three hearing impairments. The coefficients of three 32 coefficient lowpass FIR filters with roll-off and cut-off characteristics corresponding to impairments of mild hearing, hard hearing and deaf hearing, are provided. Using PRBS as input and monitoring the filter output on a spectrum analyzer, the frequency response of the filters are studied.

8 Suppression of 60 Hz interference in biomedical signal using IIR band-reject filtering

- The 10-th order IIR filter is implemented as the cascade of five bi-quads and the prescaled coefficients of the bi-quads are provided. The contaminated analog signal is made available on an audio tape and is A/D converted at the given rate. Filter implementation consists of (1) loading scale factors, coefficients and data values of each bi-quad, (2) performing the bi-quad operation in sequence for all sections, and (3) repeating the bi-quad loop for each data sample. The reduction of 60 Hz at filter output is measured on a spectrum analyzer.

9 Low frequency real-time radix-2 FFT spectrum analyzer

- It serves as a low cost, flexible tool to analyze speech and baseband data signals. The 128 sample-block of digitized data is DMA transferred, multiplied with a given window function, and analyzed using a radix-2, 128-point DIT FFT algorithm, to display the signal power vs. frequency plot on the host PC. The developed Butterfly (BF) segment with coefficient LUT and the given bit-reversal data ordering program are synthesized in to the FFT implementation by programming the grouping of BFs and the FFT stages. The analyzer accuracy is tested by monitoring sinusoids, speech and biomedical signals and comparing with the measurements of lab spectrum analyzer.

10 Touch-tone (DTMF) phone dialer and decoder

- DTMF used in push button dialing, generates a pair of tones of specified duration for signaling each specific digit. Digital samples of the tone pair are generated by two bi-quad sections whose outputs are scaled, added and logarithmically compressed using LUT operation. 8 such bi-quad sections are needed to signal all codes on DTMF keypad. The decoding program logarithmically expands the received data (LUT operation) and then identifies the dialed digit by calculating the energy at each frequency by using 8 bi-quad Goertzel filters. The implementation is tested on a dialing sequence of several digits and the operation is monitored on a spectrum analyzer.

11 Word (digit) recognizer for voice activation

- Possible applications are in hands-free data entry and as aid to disabled. Lessons 10 and 11 can be combined to form a telephone management system. A simple speaker-dependent scheme is implemented for recognizing the utterance in a vocabulary of 10-15 words which includes digits and a few commands. The program detects the beginning and end points of the utterance, then measures features such as the rate of zero axis-crossings and speech energy in 7 overlapping frequency bands using 7 specified bi-quad IIR filters. The average value of each feature over subintervals, is compared with that of prestored templates and based on its similarity, the utterance is associated with words in vocabulary. The bi-quads can be replaced by the FFT program (developed in Lesson 9) for more accurate energy measurement.

12 Speech synthesizer based on Linear Predictive Coding

- Such a system can automatically announce measurements, warn emergency situations, and free eyes for other operations. The implemented small vocabulary consists of 40-50 words and is memory constrained. Each word is reconstructed by filtering a specific excitation using a specific lattice filter, whose parameters change every frame (10 msec). For each frame, the program uses 10 LPC (reflection) parameters, the pitch (if voiced) and the energy (gain), which are stored in LUTs. The lattice filter implementation is given in (L6.7). A properly parametered periodic pulse train (for voiced) or PRBS (for unvoiced) generated using (L6.6) serves as the excitation. The coefficients and excitation are updated every frame. By concatenating words, phrases and sentences can be formed. New words can be added to the vocabulary by

determining their parameters using LPC of the speech segment framewise.

Other AMS Educational Products

EZ-ROUTE Tutor™ Electronic CAD Courseware

Evaluation Package Includes:

- EZ-ROUTE™ Std Software Package
(Schematic Capture, Symbol Editor, PCB Layout)
- 300+ Page Student Courseware Book
- Instructor's Lab Manual

EZ-ROUTE Tutor™ Advanced Electronic CAD Courseware

Evaluation Package Includes:

- EZ-ROUTE™ Pro Software Package
(Schematic Capture, PCB Layout and Auto Router)
- 300+ Page Student Courseware Book
- Instructor's Lab Manual

EZ-MICRO Tutor™ Microprocessor Design Course

Evaluation Package Includes:

- EZ-MICRO™ Lab Software Package
- 68HC11 CPU Board
- Lab Project Board
- 300+ Page Student Courseware Book
- Instructor's Lab Manual

EZ-DSP Tutor™ Digital Signal Processing Course

Evaluation Package Includes:

- EZ-DSP™ Lab Software Package
- EZ-DSP™ CPU Board
- Lab Project Board
- 300+ Page Student Courseware Book
- Instructor's Lab Manual

Note: For pricing information please call (852) 2402 3200

