Z80 Development System



Zilog

4/23/76 **Product Specification**

The Z-80 Development System is a turn-key unit designed to support all activities associated with the creation of microprocessor hardware and software. The system includes two floppy disks with a sophisticated file maintenance system. With this capability, the user can quickly retrieve, manipulate and store large files of data to minimize software development time. The system also includes an advanced real time debug module that connects directly to the user's system, thus providing a simultaneous hardware and software debug capability.

System Features

- Turn-key system including:
 - Z80-CPU with 4K bytes dedicated ROM memory
 - RS-232 or current-loop serial interface
 - 16K Bytes of read/write memory expandable to 60K bytes
 - Programmable hardware breakpoint module
 - Programmable real-time event storage module
 - In-circuit emulation bus to connect system to user's equipment
 - 2 floppy disk drives and controller
 - Full software including:
 - ROM based operating system
 - ROM based debug package
 - Editor
 - Assembler
 - File maintenance
 - Optional Universal Parallel I/O card for interface to printers, PROM programmers, etc.

System Description

The heart of the development system is the powerful Z-80 single chip microprocessor which is ideally suited to the multi-task operational requirements of a development system. A single Z80-CPU is shared between both the user's hardware (User Mode) and the System resident monitor (Monitor Mode).

In the Monitor Mode the System performs as a stand-alone development tool allowing software programs to be entered into RAM memory, edited, assembled, filed on disk for future use and loaded for execution. This entire process is quickly performed through simple commands from the user's terminal.

In the User Mode, the system memory and peripheral elements are dedicated to the user's own system. The system peripherals use I/O port numbers $E0_H$ through FF_H these port numbers are reserved for the system. In User Mode, a RAM resident user's program is executed in real time. The use of RAM memory for the program eliminates costly and timeconsuming PROM programming in the early phases of software development. The in-circuit emulation bus allows the user to connect his own peripheral devices or memory to the system and use them in conjunction with the system elements.

A major feature of the Z-80 is its powerful debug module. This module allows selected User Mode system transactions to be stored in real-time into a special memory. The user can also specify that various types of system transactions can suspend user operation and cause the system to reenter the Monitor Mode. A complete record of the 256 transactions that were recorded in the independent memory just prior to suspension can then be conveniently displayed on the system terminal or listed on a line printer. This ability to preserve real-time event sequences and then review selected events in detail, permits the user to accomplish product design and hardware/ software debugging in the shortest time possible.

Hardware Module Description

Processor Module

The Processor Module is a single card containing all elements necessary to function as a stand-alone computer. A serial asynchronous I/O port is provided for operation of a teletype or CRT terminal. The card also contains 3K bytes of ROM and 1K byte of RAM in which resides the operating system, peripheral drivers, bootstrap loader and debug software. The peripheral driver routines can be accessed by the user.

Real-Time Debug Module

The Zilog Development System realtime debugging capability enables the user to easily locate and correct any hardware or software design errors. With this module, the user monitors the operation of his software in real-time and sets hardware breakpoints to stop the program on any data, address or control bit pattern. Once stopped, the system returns to the Monitor Mode where the ROM resident debug software allows the user to display the contents of any internal CPU register, or memory location, or to change any register or memory location prior to continuing the program from that point.

The real-time debug module consists of a Real-Time Storage PCB and Breakpoint PCB.

The Real-Time Storage board contains a 256 x 32 storage array. This array stores up to 256 events. The 32 stored bits include: **1**6-bit address bus, **8**-bit data bus **7**-bit control bus.

The last bit is used as a marker to identify the first transaction that is stored when the user's program begins execution. The debug software package allows the user to specify the type of transactions that are to be stored in the memory. Any combination of the following transactions can be stored:

Memory Reads Memory Writes

I/O Port Reads I/O Port Writes.

After the system returns to Monitor Mode from User Mode the contents of storage array can be printed on the user's terminal in a concise form so that he can analyze how he got to the current point in his program.

The Breakpoint card monitors the system bus and halts execution of a user's program if a user specified transaction occurs. The user may specify that a break should occur on any combination of the following transactions: ■ Memory Read ■ Memory Write ■ I/O Port Read ■ I/O Port Write. In addition he may specify that the selected transaction have: ■ A specified 16-bit address ■ Any specific bit pattern on the data bus. Thus the user can specify complex events such as writing a "1" on bit 6 of I/O port number F6_H.



System Memory

The System uses standard, 4K dynamic RAM circuits configured in 4K byte increments up to a total of 60K bytes as required by the user. The standard system requires only 16K bytes but additional memory can be easily added if large programs are to be executed in User Mode. System memory is shared between the Monitor Mode and the User Mode.

In Monitor Mode programs are entered, edited, assembled and loaded directly into RAM memory for immediate execution without the additional cost and time delays associated with programming PROM's. In the User Mode, RAM memory contains the user's software and the user has complete control over the system peripherals and CPU. The monitor does not use the first 60K byte memory locations while in User Mode. These locations are totally dedicated to the user. Any combination of external ROM, RAM or PROM can be used in place of, or in combination with the standard system RAM through the In-circuit Emulation Bus.

Floppy Disk Controller

This single PCB interfaces two floppy disk drives in support of the Z-80 disk operating software. The ROM based monitor software contains the floppy disk software driver and bootstrap loader which loads the file maintenance software, system routines and user programs to be executed.

Universal Parallel I/O Module (Optional)

This module contains four Zilog parallel I/O controllers (Z80-PIO) which can control a wide range of parallel interface peripherals. The card is universal in that the system software can configure the Z80-PIO's with any set of bidirectional data transfer or any combination of status and control lines to match the requirements of various peripherals. It is used as an interface to optional peripherals such as line printers, paper tape punches and readers or PROM programmers.

In-Circuit Emulation Interface

This card contains all elements necessary to share the System between the User and Monitor Modes. In addition, a standard hardware interconnection cable is also provided for simple interface between the user's hardware and the System. This port includes: **1**6-bit address bus **8**-bit data bus **1**All CPU control signals **5**ystem Clock External Memory enable. All lines are fully buffered and provide TTL compatible signal levels for connection to any external user peripheral device, CPU control logic, memory system or even the user's own unique CPU card.



Chassis Description

Spare slots in the system chassis are provided for additional user cards and I/O connectors. Seven slots are available; two for user cards; two for I/O connectors; and three for additional memory.

Software Description

The Z-80 Operating System

The Z-80 floppy disk based Development System is accessed through OS Z-80, which resides in one page (4K) of dedicated non-volatile memory. OS Z-80 controls the switching between User Mode and Monitor Mode, and provides a hierarchy of command levels. The monitor software also resides in the non-volatile 4K dedicated memory while the editor and assembler are stored on the floppy disk and are called into the general purpose system memory when required.

A cold start is initiated when power is turned on. The user then types "S" on his terminal and the system automatically determines the terminal's speed and adjusts itself accordingly. The terminal then responds by printing OS>. Four commands can be issued at this level: Debug, Edit, Assemble or File. When in the debug level, user programs can be executed with a simple GO command. The user's program will then continue to execute until a break is encountered.

Data and commands are entered into the system by the user in free-form format. That is, data and command fields are separated by any number of

PC

4

INITIA



Operating System

commas or spaces. In addition, commands may be fully spelled out or abbreviated. There are two special characters that can be used at any time:

@ = delete last character

! = delete entire line

Debug Level

The Z80 Debug software has a repertoire of fourteen instructions, designed to give the user facilities in controlling, analyzing, and debugging his own programs which reside in up to 60K bytes of system memory. The debug commands:

BASE specifies the numerical base in which the user chooses to enter memory addresses and data. The base may be hexadecimal or decimal. When the base is unspecified, the system uses hexadecimal notation at this level.

BREAK sets an automatic hardware breakpoint into the real-time debug module. This break can be on a memory read, memory write, I/O port read, or I/O port write. Addresses, data and data masks can also be specified. A break from the user program can also be caused by pressing the Monitor button on the front panel. In either case, a break causes the state of the user's CPU to be stored so that



tem Flow of Control

execution can be resumed later Control returns to the debug level.

COMPARE allows the user to compare two memory blocks of any size and list any locations that are not identical.

DISPLAY prints the contents of all CPU registers in a concise format. Alternately any individual register or any contiguous block of memory may be displayed.

GO begins execution of the user's program. Execution can begin at any specified address, or it can continue from a previous breakpoint. A hardware or manual break is required to return control back to the debug level.

HISTORY is normally issued after a break from a user program. This instruction lists on the terminal the state of the address, data and control busses of the CPU during the execution of up to 255 bus transactions that occurred in the users program just prior to a break.

LOAD transfers assembled programs into system memory, ready to be executed by the GO command.

MOVE allows the user to transfer a block of memory of any size from any location to any other location.

PULSE is identical to Break except that a pulse on a special connector is provided each time the specified condition occurs and the program continues to execute. Pulse can be used to synchronize an oscilloscope display.

SAVE stores the RAM image of linked programs and subroutines on the user's disk.

SET stores data entered from the terminal into specified registers or memory locations.

STEP executes one instruction and then prints the contents of the CPU registers.

TRACE specifies if memory read, memory write, port read and/or port write conditions are to be stored in the Real-Time Debug Module during execution of the user's program.

QUIT returns control to the OS level.

Editor Level

The Z-80 text editor is called from the OS level by the command:

EDIT filename filetype

The EDITOR transfers the user's file from disk to a work space in memory.

The EDITOR works on a pointer concept where a pointer is moved by the user to access any desired line. The user can modify this file by any of the following commands:

AGAIN repeats the previous command

BOTTOM moves the line pointer to the bottom line.

CHANGE locates any specified character string and replaces it with any new character string as entered from the terminal. The user can specify how many occurrences per line as well as the maximum number of lines to change.

DELETE removes a specified number of lines from the file.

FILE writes the file from the work space on to the user's disk and returns to the OS level.

GET loads a block of text temporarily stored on the disk back into the work space to the user's disk and returns

GOTO places the pointer at a specified line number.

INSERT transfers new text from the terminal to the location of the pointer in the work space.

LINENO prints out the line number of the current pointer position.

LOCATE locates any specified string in the text and places the pointer at the beginning of the first line in which it occurs.

MACRO concatenates a string of editor commands, coupled with an & symbol. MACRO is executed with the XECUTE command.

NEXT moves the pointer down by a specified number of lines in the work space.

PRINT outputs the specified number of lines to the terminal beginning from the current pointer position.

PUT writes a specified block of text on the disk for temporary storage. The text is recovered with the GET command.

QUIT returns control to the OS level without filing the text on the disk.

REPLACE replaces the current line of text with new text from the terminal.

SAVE files the contents of the work space on the disk and returns to the EDITOR.

TOP moves the line pointer to the zero position in front of the first line.

(Software Description continued)

UP moves the pointer up a specified number of lines in the work space.

XECUTE executes the Macro command.

Assembler Level

The Z-80 resident assembler is a counterpart of the Z-80 cross-assembler and processes the same source programs.

The assembler is entered from the OS level by the command

ASSEMBLE filename filetype

The source program is written in assembly language. Free-form format of the instructions is permitted within the following rules:

label: OP code operand-1 operand-2; comment

Any number of commas or blanks may be used to delimit the terms. Some instructions possess no or only one operand. Labels are terminated by a colon unless they begin in column 1, in which case the colon is optional. OP codes cannot begin in column 1. Comments begin with a semicolon and can begin in any column.

File Maintenance Level

The file maintenance system has a repertoire of twelve commands that allow the user to easily manipulate large files of any type. These commands include:

APPEND one file to another file.

COMBINE creates a new file from any number of existing files.

COPY copies the entire contents of one diskette to another diskette

COMPACT removes any unused information in the directory.

DUMP performs a hexidecimal dump of a file to the terminal.

ERASE eliminates a file from the disk.

FORMAT initializes a diskette for system operation

LIST prints the directory of all files on the user's disk.

NAME changes the name associated with a file.

PRINT lists a file on the terminal.

QUIT causes the system to re-enter OS.

STAT gives the number of sectors that are unused on the disk.

Logical Device Assignments

All system I/O activity is oriented to six logical devices with the following default assignments:

Logical Device	Default Assignment
 Console Input Console Output System Input System Output Utility Input 	Console terminal Console terminal Floppy disk Floppy disk Console terminal
6) Utility Output	Console terminal

Logical devices 1 and 2 are used to receive console commands and communicate their direct results.

Logical devices 3 and 4 are used to input user files and for output of system operations such as editing and assembling.

Logical devices 5 and 6 are used for special I/O activity such as reading and punching paper tapes, and printing user files.

Default assignments can be overridden by the command:

ASSIGN device parm

Where "device" is one of the following six logical device abbreviations

CONIN
 CONOUT
 SYSIN
 SYSOUT
 UTLIN
 UTLOUT

and "parm" is either TTY or ZDOS for the standard device assignment of the console terminal or floppy disk, respectively, or the actual address of a special I/O routine for manipulating a custom I/O device.

This command can be used to interface I/O devices not supported on the Z-80 Development System.

For example,

ASSIGN CONOUT 1740

will direct all console output to address 1740. By loading the I/O handler for a special display device at this address, the results of user commands will appear on this device.



Dual Floppy Disc Subsystem

Specifications

CPU

Standard Z-80 CPU

Memory

3K bytes ROM/1K bytes static RAM dedicated to system monitor 16 bytes general purpose RAM expandable to 60K bytes

System Clock

Crystal controlled at 2.0 MHz or optional 2.5 MHz

I/O Channels

Standard Interface to Disk Unit, Serial Data Terminal and In-Circuit Emulator. Optional Interface for PROM Programmer and Line Printer. Two spare connectors for other user designated system peripherals.

Standard Equipment

Z-80 CPU Card with 4K bytes of ROM/RAM Monitor and debug Software 16K Bytes of RAM Realtime Debug Module (Storage module and breakpoint) Dual Floppy Disk Subsystem In-Circuit Emulator RS-232 or Current Loop Asynchronous Terminal Interface Z-80 Resident Assembler, Editor, Disk Operating System and File Maintenance System Full Documentation

Optional Modules

8 port parallel I/O Modules with Interrupt RAM, 16K byte Memory Modules Module Extender Drawer Slides and Extenders for Rack Mounting

AC Power Requirement

50/60 Hz, 115 VAC, 200 Watts Optional 230 VAC Power Supply

Environmental Characteristics

Operating Temperature: 0° to 50°C

Physical Characteristics

Two separate chassis. One contains the disk drives and disk power supplies, while the other contains all other elements

Approximate weights and dimensions apply to both chassis: Size: 19"W x 9"H x 15"D Weight: 35 lbs.

Electrical

Integral Power Supplies provide all necessary voltages, plus 4 amps of $+5V \pm 5\%$ is available in the CPU chassis for user cards



170 State Street / Los Altos, California 94022 / (415) 941-5055 TWX 910-370-7955

Printed in U.S.A.