TEK 8500 MODULAR MDL SERIES

FLEXIBLE SOLUTIONS
FOR A FUTURE OF CHANGE

The 8500 Series Technical Catalog

Tektronix®
COMMITTED TO EXCELLENCE

Table of Contents

*8550 support only

Mainframes

8500 MDL Modular Family
8550 Microcomputer Development Lab
8560 Multi-User Software Development Unit
8540 Integration Unit

8500 Series Modular MDL Family

Three Basic Design Environments

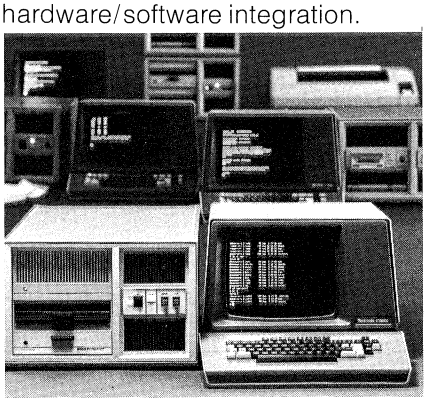
Microcomputer design is currently performed in three basic work environments. These are the single-user environment, the multiple-user environment, and the host computer environment. The Tektronix 8500 Series of Microcomputer Development Labs offers a full complement of development tools for all three microcomputer design environments. In each case, the most advanced technology available has been incorporated into an 8500 Series microcomputer development system that gives maximum performance in its intended setting. In this manner, Tektronix provides unmatched support for both microcomputer software development and hardware/software integration, no matter where it occurs.

The 8550 MDL for the single user environment

The 8550 MDL is a self-contained microcomputer development lab that gives complete support to one user at a time. Important features of the 8550 MDL include a flexible filing system that easily adapts to any type of software project, high level language and assembly support with symbolic debugging for both 8-bit and 16-bit chips, real-time emulation to install developed software in the prototype hardware, and a powerful set of real-time debugging tools, including the Trigger Trace Analyzer for 8-bit and 16-bit designs, and the Real-Time Prototype Analyzer for 8-bit designs.

The 8560 MDL for the multi-user environment

The 8560 MDL is a full microcomputer software development lab that supports up to eight separate workstations. It's done through the 8560's powerful TNIX* operating system, a derivation of Bell Lab's UNIX**Operating System Version 7 that has been tailored specifically for team-oriented microcomputer development work. It includes a fast, flexible filing system for effective data base management, advanced command capability for streamlining software development tasks, and several optional utility packages such as text processing. The 8560 uses a powerful 16-bit CPU to control system resources. Software work stations consist of CRT terminals; and the 8540 Integration Unit, which has full emulation and debugging capability, or the 8550 can be used for hardware/software integration.



The 8540 Integration Unit for the host computer environment

In the host computer environment, a general purpose computer is used to support microcomputer-based design projects. The host computer's timesharing capability allows a number of designers at CRT terminals to simultaneously use the computer's facilities. Each designer has access to microcomputer development tools such as editors, assemblers, and compilers. The 8540 Integration Unit extends the host computer's microcomputer development support to include the task of hardware/software integration. While the 8560 or other host computer supports the software development task, the 8540 is used to integrate, test and debug the prototype hardware. The 8540 uses Real-Time Emulation, a fast and efficient integration technique that uses an emulator processor identical in function to the one targeted for the prototype. For added real-time debugging, a 62-channel Trigger Trace Analyzer is available as an option.

*TNIX is a trademark of Tektronix
**UNIX is a trademark of Bell Laboratories

8550 Microcomputer Development Lab

The 8550 MDL is a completely self-contained microcomputer development system that gives full support to both software development and hardware/software integration. For software development, the 8550 provides a wide range of sophisticated tools for fast, effective code generation. These include many 8-bit and 16-bit assemblers that support symbolic debugging, produce relocatable code and have powerful macro capabilities. High level language support includes Pascal compilers, and MDL/μ compilers, Tektronix enhanced form of BASIC with many microcomputer-oriented extensions. Also included are both a line-oriented editor and an Advanced CRT-oriented Editor.

All software tools and files are managed by a highly flexible filing system that allows nesting to any level a project requires. Included are a linker and library generator to assist in modular code development. The 8550's two 1Mbyte floppy discs provide more than adequate mass storage.

For hardware/software integration, the 8550 offers debugging tools that cover the full range of 8500 Series support, both 8-bit and 16-bit. All use Real-Time Emulation, a superior debugging technique that permits code execution in real-time, with no waits or stretched clock pulses added. Symbolic debugging allow easily identified labels to be substituted for numerical addresses. Real-time debugging is available through the Real-time Prototype Analyzer (8-bit chips), and Trigger Trace Analyzer (16-bit and 8-bit chips) with state-of-the-art triggering capability.

Program Entry and Editing

Program entry and editing is accomplished via the standard line-oriented editor or the optional Advanced CRT-oriented Editor, so both line and screen-oriented editing are allowed.

Assembly and HLL Support

The 8550's assembler packages provide full code development support for a wide range of 8-bit and 16-bit chips. Powerful macros allow the designer to access frequently used sets of code by name reference. The linker, working with the relocating features of the assembler, links and locates multiple code segments into a complete executable program. Additionally, a conditional assembly feature allows the designer to customize the final program by testing conditions to determine which of certain code segments are to be assembled into the final program. Extensive English language diagnostics provide easy-to-understand error messages and locate the line in which the error has occurred. When assembly is completed, the assembled code is stored on disc in a binary format file.

For selected chips, high level language support is available to increase your development power. One language offered is MDL/μ, a BASIC-like language created specifically for microcomputer design. Another is Pascal, with its structured approach that readily adapts to microcomputer software development. For both languages, Tektronix has added a full complement of extensions that allow code manipulation at the microprocessor level, including access to interrupt vectors, individual memory locations and I/O ports.

Program Development and Data File Management

The 8550's DOS/50 operating system helps the designer manage all phases of program development and debugging. DOS/50 supervises general I/O, file creation and maintenance, program assembly and compilation, program execution, monitoring and debugging. Data management is simplified through a versatile file management system, which allows the user to specify one main system directory, one root directory for each disc, and any number of sub-directories under the root directory. Data files may be created and entered directly into the root directory. As files are accumulated, the user may organize them into specific groups, each under its own specific directory. This allows the user to logically organize his files according to specific project needs.

Advanced Debugging Tools

The 8550 MDL also offers complete in-circuit emulation and hardware testing capabilities. With the appropriate prototype control probe for the target microprocessor, the user can transfer control from the 8550 to the prototype one block at a time, debugging at every stage. The Real-Time Prototype Analyzer option provides an invaluable tool for verifying and correcting execution of the program in real time. Its hardware trace function captures bus information from the program as it executes. This information can then be displayed in a trace format, or used to trigger a break in execution, time a program segment, or signal an external device such as a logic analyzer.

For 16-bit and 8-bit designs, the Trigger Trace Analyzer uses a high speed trace buffer capable of working with bus cycle speeds up to 8 MHz. The buffer is 62 bits wide to capture 16 data bits, 24 address bits, 14 emulator-dependent bits, and 8-bits from external hardware. Up to four independent events can be combined in both logical and sequential combinations to form a breakpoint or data storage trigger. The events themselves can be defined from a combination of bus data, pass delay counters, and an external qualifier. Each event can also initiate its own external output to trigger other instruments such as oscilloscopes and logic analyzers.

The Trigger Trace Analyzer allows a powerful and diverse set of debugging operations. For instance, the real-time length of an interrupt service routine can be measured while simultaneously counting the number of assertions from a second interrupt source. Or the maximum depth of a stack during program execution can be determined. To optimize code performance, the read or writes within any given address range can be isolated and counted, thus identifying heavily used areas to be recoded in assembly language.

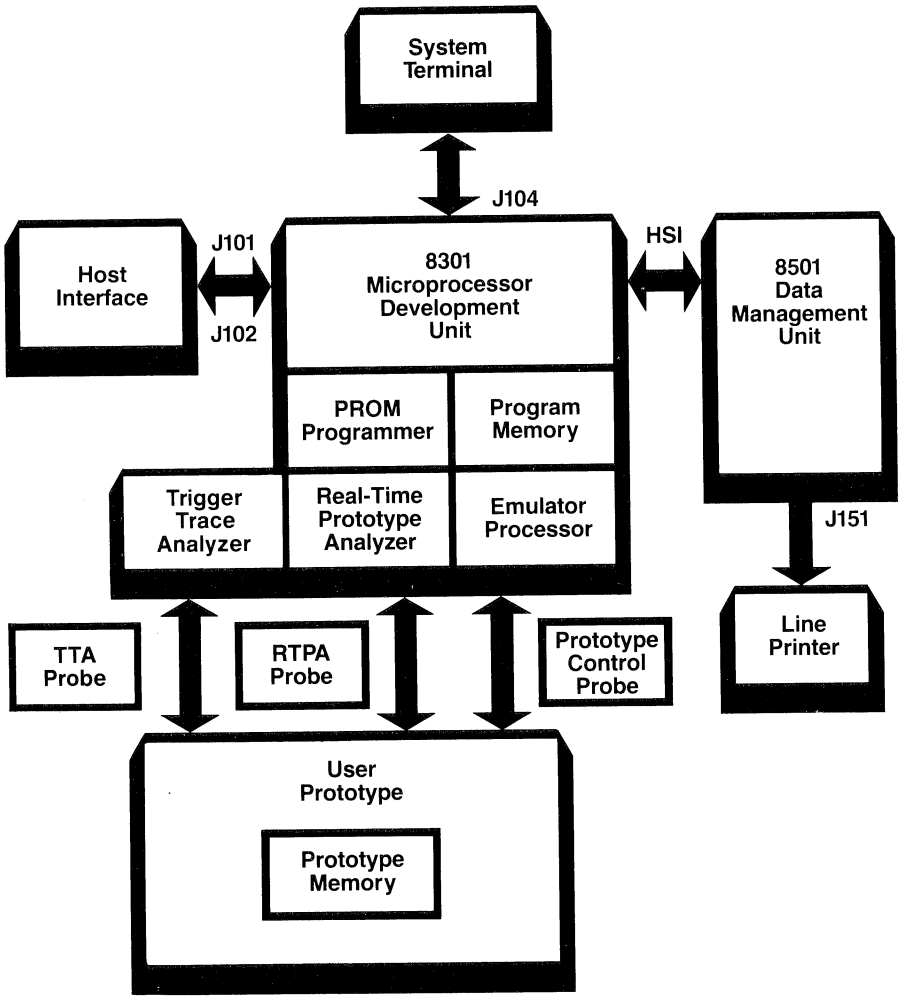
Real Time Emulation — Three Emulation Modes

After an error-free assembly listing has been obtained, the resulting object code may be executed in system emulation mode 0 on the optional emulator processor. The emulator processor is identical in function to the microprocessor that will finally be installed in the user's prototype. Execution is performed under control of the debug system; during execution, program steps can be traced, software breakpoints can be set, and memory can be examined and changed as required. Should an error be discovered, that portion of the program can be corrected at the source level using the text editor. It can then be reassembled and executed again. This procedure continues until the program is correct.

After the software has been debugged, it may be exercised on the prototype circuitry in the partial emulation mode (mode 1). During partial emulation, control may be released from the 8550 to the prototype in stages. The development software runs using 8550 memory space and prototype I/O and clock. The 8550 memory mapping feature allows code to be gradually mapped over to the prototype's memory in manageable blocks. Throughout partial emulation, the user has access to prototype circuitry through the debugging system, which enables him, as before, to trace, set breakpoints, examine and change memory and register contents.

In full emulation (mode 2) the program is run on the prototype, but program execution is still under the complete control of the debug system. All I/O and timing functions are directed to the prototype; all memory has been mapped over to the prototype; and only the prototype control probe is still in place, emulating the target microprocessor. Although the prototype is effectively free-standing, the user may still direct program activity from the 8550.

8550 Parts and Functions
Fig. 1



8550 CHARACTERISTICS

8301 MICROPROCESSOR DEVELOPMENT UNIT

Physical

Height	11 in	(280 mm)
Width	17 in	(430 mm)
Length	23 in	(585 mm)
Net Weight	60 lbs	(27 kg)

ENVIRONMENTAL

Operating Temperature	32°F to 122°F (0°C to 50°C)
Humidity	90% @ 86°F to 140°F (30°C to 60°C)
Altitude	
Operating	0 to 15,000 ft (4,500 m)
Storage	0 to 50,000 ft (15,000 m)

POWER REQUIREMENTS

115 V ac (90 V ac-132 V ac) @ 48 to 66 Hz.
230 V ac (180 V ac-250 V) @ 48 to 66 Hz.

Outputs

5.2 V dc + 1%/-2% @ 35.0A
+ 12 V dc + 0/-5% @ 1.7A
−12 V dc + 0/-5% @ 1.7A

8501 DATA MANAGEMENT UNIT

Physical

Height	10.5 in	(267 mm)
Width	16.8 in	(424 mm)
Length	23.5 in	(597 mm)
Net Weight	55 lb	(25 kg)

ENVIRONMENTAL

Operating Temperature	50°F to 104°F (10°C to 40°C)
Humidity	20% to 80% relative noncondensing
Altitude	
Operating	0 to 8,000 ft (2500 m) Derate max operating temp. by 1°C for each 300 m above 2400 m.
Storage	0 to 50,000 ft (15,000 m)

POWER REQUIREMENTS

115 V ac (90-127 V RMS) @ 50 Hz ±1% or 60 Hz ±1%. 230 V ac (180-250 V RMS) @ 50 Hz ±1% or 60 Hz ±1%.

Outputs

24 V dc ±5% @ 2A
12 V dc ±3% @ 4A
−12 V dc ±5% @ 540 mA
5 V dc ±5% @ 20 A
15 V dc ±10% @ 20 mA

Output Ripple

24 V dc	100 mV (p-p)
±12 V dc	120 mV (p-p)
15 V dc	50 mV (p-p)
15 V-dc	100 mV (p-p)

Overload Protection

Automatic current limit foldback.

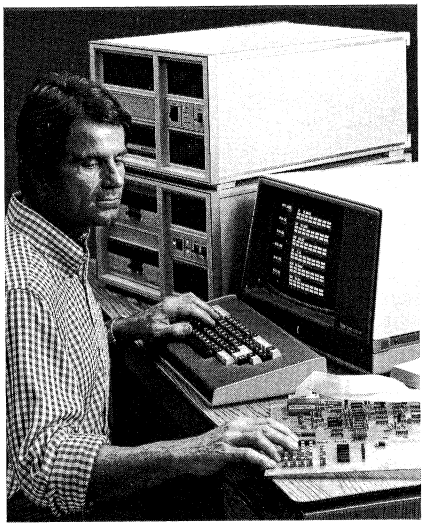
FLEX DISC CHARACTERISTICS

Encoding — IBM compatible single or double density. Format must qualify as follows: MFM sectors — 256 bytes. FM sectors — 128 bytes.

Diskette Type — Single or double sided, soft sectored.

Capacity —

Double sided, double density
1,021,696 bytes.
Single sided, double density
509,184 bytes.
Single sided, single density
256,256 bytes.



8560 Multi-User Software Development Unit

The Tektronix 8560 is a multi-user development system that, when used in conjunction with an 8540 Integration Unit, covers the entire microcomputer design process, from software development through hardware/software integration. At the same time, it allows maximum design flexibility by supporting a broad range of chips at both the 8-bit and 16-bit levels. The 8560's multi-user capability offers numerous advantages to the design team, such as lower cost per user, shared software and hardware resources, unified project management and enhanced security.

The heart of the 8560 is a minicomputer running under TNIX,* a derivation of UNIX** the powerful Operating System designed by Bell Laboratories. The CPU is an LSI 11/23 that operates in conjunction with two I/O processors for high performance throughput. A separate board is included to control two spooling line printers. System memory is 128K bytes (expandable to 256K bytes), and mass storage includes a 1 Mbyte floppy disc and a 35.6 Mbyte (formatted) Winchester hard disc. The I/O processors support baud rates up to 9600 bps, providing wide flexibility when interfacing with various terminals.

Software work stations each consist of a standard RS-232C CRT terminal, with the Tektronix CT8500 being recommended. In this manner, the system's software development capability can be expanded by simply adding terminals. Integration stations each consist of a Tektronix 8540 Integration Unit, which has built-in high-speed serial communications compatibility with the 8560 or an 8550 MDL.

For added flexibility, the 8560 can be configured with the 8540's and 8550's in several different communications modes. The 8540 can be connected in-line with a terminal to a single 8560 port, or without a terminal to a separate port. In either configuration, a user can control any 8540 in the system including several units at the same time. In addition, it is possible to use many of the 8560's powerful command resources when debugging with the 8540. This includes the ability to use the 8560's hardware and software for I/O simulation in the early states of debugging.

The 8560 will support up to two line printers, with the Tektronix 4643 being recommended. The line printers will both operate in the spooling mode, so as not to tie up the users' terminals while printing.

TNIX Operating System

Running under the TNIX operating system, the 8560 can include a wide variety of software design tools offered by Tektronix. These include macro assemblers and compilers*** for both 8-bit and 16-bit chips; loader/linkers to combine object files into executable object code; and several types of editors, including an advanced screen-oriented version. There is also a text processing package that can be a valuable aid to software documentation, and a number of utility packages to expand system capabilities in various ways.

The 8560's TNIX operating system was derived directly from Bell Laboratories' UNIX Operating System, an operating system that has gained widespread acceptance throughout the computer world. TNIX takes all the proven advantages of UNIX and applies them directly to the specific tasks required during microcomputer design.

TNIX includes a hierarchical filing system that greatly enhances the ease and speed of filing operations. It is organized with a "root" directory on top and descending levels of priority underneath. Each level may contain either files or directories that point to more files on a lower level.

TNIX also provides a sophisticated system of read/write protection that guards user material but retains the inter-file access needed to enhance productivity. Each user can specify read, write, or execute protection on any of three levels: user, user's group or public. This way, files can first be completely protected while under initial development, then released to those working on related sections of the program, and finally released to the entire project after debugging is completed. To enhance communication between users, TNIX includes a system of electronic mail that lets team members send messages to one another. It also allows direct communication between two users at separate terminals, plus the ability to broadcast a single message to all terminals. These various communications modes can be a valuable asset to both team members and team management.

Powerful command structures simplify software development and integration

TNIX uses an extremely flexible command structure, one that permits effective programming at the expert level as well as the beginner level. In all cases, TNIX commands let the user concentrate on software design instead of system manipulation. One simple, but very powerful TNIX command operation is pipelining. It permits the output of one command to act as the input to another. At an even higher level of refinement, TNIX permits the construction of intelligent command files that can be called by the user and executed as a single unit.

Other TNIX command features include full read-ahead which lets the user type in commands as fast as possible, without waiting for previous commands to finish. Another time saver is job prioritization, which lets the user concurrently run several jobs which may be at different levels of priority. For instance, a compilation could be detached and run in the background while the user went on to start editing a new source file. This feature saves time while making more intelligent use of the system's resources.

To optimize the user interface, TNIX includes a special amenity called GUIDE, which combines the best features of both menu-driven and command-driven systems. GUIDE lets the user select any system operation from a menu display, and then automatically completes the operation. At any time, the user can exit from GUIDE and escape back to TNIX and its user-driven command format. This way, the user can employ the menu as a learning tool, but is able to depart to the regular TNIX command format for more sophisticated operation.

*TNIX is a trademark of Tektronix
**UNIX is a trademark of Bell Laboratories
***Available 2nd quarter, 1982

Automatic module combination cuts complexity

During debugging, software modules that exhibit defects must have corrections entered into their original source code files. Before debugging can resume, all files affected by the changed source modules must be recompiled or reassembled into new, updated object code modules. Otherwise, new and old versions of the interdependent object modules will be mixed.

To streamline debugging, TNIX includes a special feature called 'make" which uses a control file to oversee the modular combination process and ensure that all modules are of the most recent version when combined for execution. "Make" can cut recompiling or reassembling time to an absolute minimum by performing these operations only on files that actually require it.

Advanced software development tools

The 8560 MDL system offers a large selection of software development tools that can be run under TNIX. These include a variety of 8-bit and 16-bit assemblers with features such as macro capability, conditional assembly and relocatable code. Also 16-bit compilers with features like structured constants, bit manipulation and re-entrant code.

Other specialized tools include a loader/linker, and a library generatory. Also two types of editors, including an advanced CRT-oriented editor that allows fast, efficient text manipulation at both the line and character level.

In addition, there are a number of 8560 system utility packages available that offer advanced support to microcomputer design projects. Among these is a text processing package that can be a valuable aid when documenting software, and native programming package for developing your own 8560 software.

Hardware/Software Integration

To handle hardware/software integration tasks, the 8560 is optimized to use the Tektronix 8540 Integration Unit as peripheral work station. Once code targeted for the prototype has been assembled or compiled and linked into executable object modules, it is loaded into the 8540's program memory via a high-speed interface. The code can now be gradually introduced to the hardware using Real-Time Emulation, a powerful debugging method that employs a processor identical in function to the one targeted for the prototype.

Data captured by the 8540 during debugging and testing can be displayed on the console terminal, the line printer, or routed to the 8560 for further processing.

Real-Time Emulation takes place in three progressive modes, all under the control of the 8540's debug software. During the first mode, all code is executed out of the 8540's program memory, with I/O simulated by software insertions that use 8560 peripherals or files for input and output. In this manner, prototype software debugging can begin even before the hardware becomes available. During the second mode, I/O and clock functions are transferred from the 8540 to the prototype, and code can be mapped over to the prototype memory in manageable blocks as it is debugged. A control probe now connects the emulator processor to the vacant processor socket on the prototype board. During the final mode, all code is installed in the prototype memory, as well as clock and I/O functions. Through the control probe, the 8540 can now exercise prototype hardware in the same manner that it will function when standing alone.

During all three modes of emulation, the 8540's powerful debug software can be applied. Breakpoints can be set using mnemonic symbols for key program locations. The status of processor registers can be examined on a cycle-by-cycle basis. All registers and memory locations can be examined and modified. And for detailed analysis of real-time execution on the prototype bus and selected hardware points, an optional Trigger Trace Analyzer is available with four powerful trigger channels that allow highly selective data acquisition.

For added power during debugging, the 8540's resident command set is also included as part of the TNIX command set. This allows the full range of TNIX command manipulation features to be applied to 8540-based operations. It is even possible to intermix conventional TNIX and 8540 commands in a single command structure.

8560 Physical, Environmental and Electrical Characteristics

PHYSICAL CHARACTERISTICS

Height		
H1	10.5 inches	(267 mm)
H2	11.1 inches	(282 mm)
Width	16.8 inches	(428 mm)
Length		
L1	23.3 inches	(593 mm)
L2	25.4 inches	(646 mm)
Weight	72.5 lbs.	(33 kg)

ENVIRONMENTAL CHARACTERISTICS

	Operating	Storage
1) Atmospheric		
Ambient temperature	10 to 40°C	−10 to 65°C
Relative humidity	20 to 80% (non-condensing)	10 to 80% (non-condensing)
Temp. rate of change	1°C/5 minutes max	24°C/hour max
2) Altitude		
Operating	0 to 8,000 feet (2 500 m). Derate max temp by 1°C for each 300 m above 2 400 m.	
3) Storage	0 to 40,000 ft (12 200 m).	
Static Discharge		
Operating	0 to 12.5 KV without effect on equipment operation. 0 to 12.5 KV without hard failures.	

ELECTRICAL CHARACTERISTICS

Power Requirements
- 115 V Nominal (90-132 VRMS)
- or
- 230 V Nominal (180-250 VRMS)
- Line Frequence Range: 48-66 Hz

Power Consumption
- 430 Watts Max.

8540 Integration Unit

The Tektronix 8540 Integration Unit is used with a 8560 Multi-user Software Development Unit or a General Host Computer, to integrate, test and debug microprocessor based software. The 8540 supports the hardware and software integration phase of a microprocessor/microcomputer based design. The Tektronix 8560 or other host computer supports the software development task.

Application

8540 support is employed when the software has been developed to the object code level and is ready to be tested with the hardware. The object code binary program is downloaded to the 8540 and stored in its resident program memory or, if available, in the prototype's own memory. If the prototype hardware is not available, the software can be executed on the 8540 emulator processor using Mode 0 emulation.

A wide variety of emulators are available for in-circuit testing and debugging. For in-circuit testing, the emulator probe replaces the microprocessor chip in the prototype. In emulation mode 1 or mode 2, the software can interact directly with the prototype hardware.

The designer controls and monitors the testing process via a CRT Terminal like the Tektronix CT8500.

Block Diagram

As shown in figure 2, a block diagram of the 8540, the complete operating system is contained in 125K byte PROM/ROM memory. The operating system is loaded from PROM and executed in the 64K-byte system memory. User symbol table information is also stored in system memory.

Part of the 8540 memory system also contains a non-volatile EEPROM buffer to store command strings and firmware update information. Complex and repeated command sequences can be permanently stored and then recalled by employing one user-definable string name. The EEPROM is also used to store update information for firmware maintenance.

The 8540 also includes 32K bytes of static program memory for use by the emulator processor. The static program memory can be expanded to 256K bytes and with the optional Memory Allocation Controller, segments of this memory can be mapped anywhere in the emulator processor's address range.

8540 System Configuration

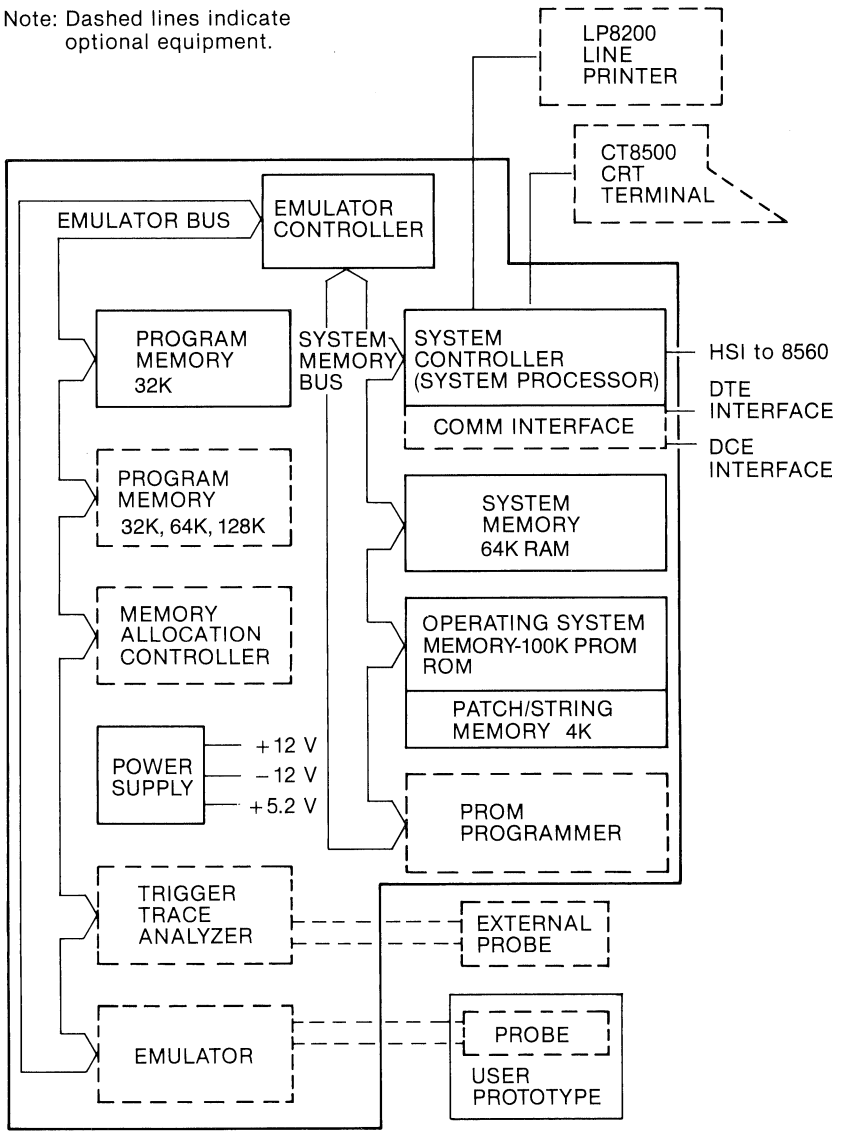
Local Interface Configuration. The 8540 Integration Unit connects to the 8560 Multi-user Software Development Unit via a high speed serial interface. Interconnecting cable lengths of up to 2,000 feet can be accommodated without performance degradation. A specialized interface protocol is used between the 8540 and the 8560 to eliminate any errors that might occur in the transmission process.

Remote Interface Configuration. For remote configurations (greater than 2,000 feet), the 8540 can be connected to the 8560 via RS-232 compatible data sets. Typical intermediate range data sets can be used over dedicated lines for distances of approximately 8 kilometers (5 miles) at data rates up to 9600 baud. For longer distances, data sets can also be used in conjunction with dial-up phone lines.

8540 Block Diagram

FIG. 2

Note: Dashed lines indicate optional equipment.



Operating Modes

The Tektronix CT8500 CRT Terminal (or a user supplied RS-232 terminal) is the user's interface with the 8540/8560 system. 8540 & 8560 commands that are entered from the terminal are routed first to the 8560. The 8560 processes each command line and routes 8540 commands to the appropriate 8540. This allows commands to be intermixed on a single line and also gives the user the ability to control any 8540 (or several 8540s) in the system. (The 8540 can be configured to operate through a separate 8560 port with no terminal attached.) A local mode is also available so user can communicate directly with the 8540.

8540 (With Option 1 Comm Interface) Host Computer Configuration

The 8540 Integration Unit is interfaced to a host computer via an RS-232 serial port. Data bytes are ASCII encoded and transmitted asynchronously at speeds up to 9600 baud between the 8540 and the host computer. For local interface configurations (up to 50 feet), the 8540 can be hardwired directly to the host computer using a standard RS-232 cable. For remote configurations (more than 50 feet) the 8540 interfaces to the host computer via a data set arrangement. Data sets can connect over moderate distances using dedicated lines or virtually unlimited distances with conventional phone lines.

The 8540 can communicate with a host computer in several modes including:

Local Mode. In this mode the terminal communicates directly with the 8540 operating system to control the emulation and debugging process.

Terminal Transparent Mode. In this mode the terminal communicates with a host computer using standard protocol; keyboard data is transmitted to the host and host data is displayed on the terminal's display screen.

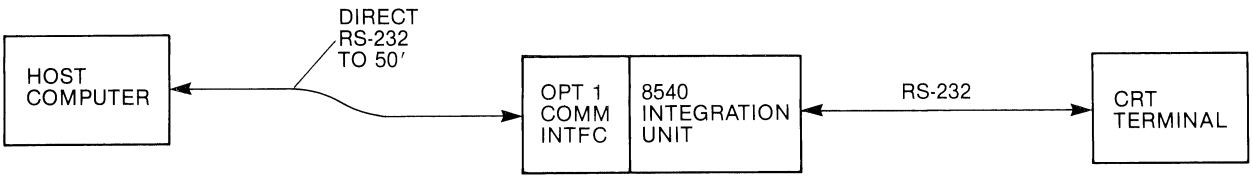
Formatted Upload/Download. In this mode microprocessor object code is transferred between the host computer and the 8540 program (emulation) memory. Binary data is encoded into Extended TEKHEX format and transmitted in message blocks. Checksum information is also included in the message blocks to allow detection of communication errors.

Details of TEKHEX encoding and protocol are included in the 8540 System Users Manual.

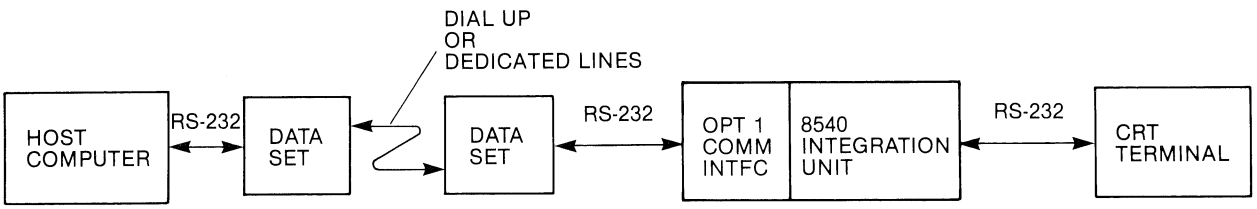
Emulation Support

When a program is ready for testing, the object module is downloaded from the host computer (or 8560) into 8540 emulator memory. When a symbol table is included with the object module, it is also downloaded and stored in a separate portion of system memory. Program symbols can then be used to reference memory locations instead of the absolute addresses.

The object code is now executed by the emulator processor under control of the 8540 debugging system. Trace displays to the screen (or line printer) show all pertinent information including instruction mnemonics, processor status and any program symbols. The user can set breakpoints to stop in critical sections or use the trace display to show selected instructions or address ranges.



DIRECT RS-232 CONFIGURATION



DATA SET CONFIGURATION

There are three progressive emulation modes available in the 8540. In the first (mode 0) the 8540 supplies all memory, clock, and interface signals to the emulator processor. No prototype hardware is required. In the second (mode 1) the clock and interface signals are provided by the prototype hardware. The emulator processor connects to the prototype microprocessor socket via the prototype control probe. The program under test still resides in the 8540 but as debugging progresses, it can be gradually transferred (mapped) over to the prototype's memory. In the third (mode 2) all of the program is transferred to prototype memory. The 8540 still maintains control of program execution through the debugging system. For more specific emulator support information, refer to the appropriate data sheet for each (supported) emulator.

Diagnostics

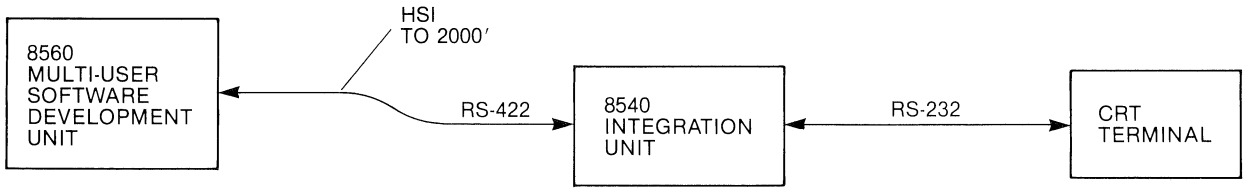
The 8540 has a complete diagnostic subsystem. On each power-up cycle, diagnostic firmware checks all available 8540 memory and verifies that all boards are functional.

Trigger Trace Analyzer (Optional)

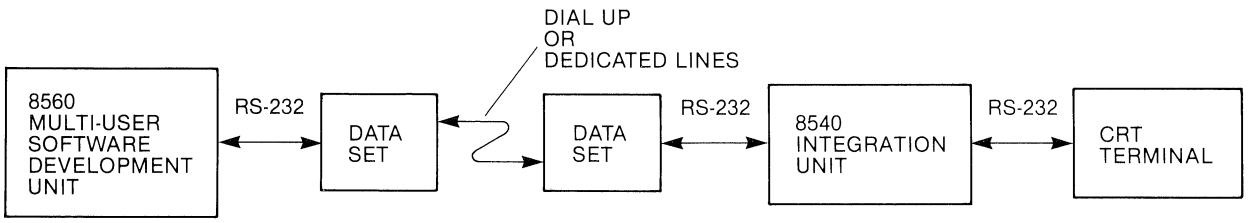
The Trigger Trace Analyzer (TTA), optional equipment for the 8540 is a powerful debugging tool. It monitors program flow in real time and allows sophisticated control and analysis of the emulator processor. The TTA stores program information in a high-speed 255-event buffer. Each event contains 62-bits of information: up to 24-bits of address, 16-bits of data, up to 14-bits of processor information, and 8-bits of external probe information. TTA information is displayed on the terminal in processor specific mnemonics along with all pertinent register and flag information.

PROM Programmer (Optional)

The PROM Programmer, optional equipment for the 8540 is a general purpose controller unit with plug-in adapter modules. Each module supports a group of similar programmable devices. Functions include: READ, WRITE and COMPARE.



HSI CONFIGURATION



DATA SET CONFIGURATION

8540 General Specifications

Operating System (Software): DOS/50 derivative (contained in 8k PROM/ROM's) i.e., 96k bytes, command interpreter and kernel 20k bytes diagnostics; 4k bytes patch and command storage (EEPROM)

Card Slot Usage

Standard Boards:
- system controller
- emulator controller
- system memory (64k RAM)
- program memory (32k RAM)
- system memory (224k PROM/ROM capacity)

Empty Board Slots:
- system side = 1 slot
- program side = 11 slots

Electrical Characteristics

Nominal Operating Voltage: 115 Vac and 230 Vac at 60 Hz

Power Requirements: 700 W max

Line Voltage Ranges: 90 to 132 Vac and 180 to 250 Vac

Line Frequency Range: 48 to 66 Hz

I/O Ports

I/O Ports: J100 (8560 Interface) J101 (remote DTE male); J102 (remote DCE female) J103 (Line Printer, Aux); J104 (Terminal)

I/O Port Data Rates:
- J100 (HSI) = 153.6k baud
- J101, J102, J103, J104 = 110/ext CLK, 150, 300, 600, 1200, 2400, 4800, 9600 baud

Reliability

MTBF: 6700 hr (calculated)
MTTR Board: 0.5 hr.
MTTR Component: 1.5 hr.

Standards Compliance: UL1244; CSA Bulletin 556B; IEC 348

Environmental Characteristics

Temperature Range
Operating: 0°C to 50°C (32°F to 122°F)
Storage: 55°C to 75°C (67°F to 167°F)

Altitude Range

Operating: Sea level to 4500 m (15,000 ft)
Storage: Sea level to 15,000 m (50,000 ft)

Humidity: 0 to 90% non-condensing (0 to 50°C)

Physical Characteristics

Cabinet Dimensions:
- Height = 280 mm (11 in)
- Width = 430 mm (17 in)
- Depth = 585 mm (23 in)

Cabinet Weight: 26 kg (57.5 lb)
Shipping Weight: 35 kg (77.5 lb)*

Cabinet Color: 3 tone; ivory/gray, smoke tan, earth brown

*Standard configuration only, no options.

RS-232 Communications Interface (Option 1)

Interface signals are routed to and from the 8540 system controller board via rear panel connectors J101 (male) and J102 (female). Table 2 lists the J101/J102 pin numbers and corresponding signal names and descriptions. Baud rate on transmitted data and received data is selectable from 110 to 9600 baud. All signals are RS-232-C compatible.

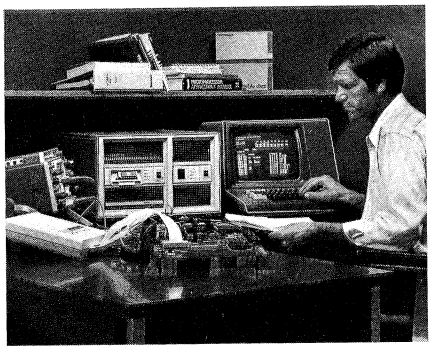


Table 2. Interface signals (Option 1, Comm Interface)

Signal Name	Circuit	J101 DTE	J102 DCE	Pin
GND	AA	Protective ground	Protective ground	1
Tx	BA	Out	In	2[a]
Rx	BB	In	Out	3[a]
RTS	CA	Out[b]	In[c]	4
CTS	CB	In[d]	Out[e]	5
DSR	CC	In[f]	Out	6
GND	AB	Signal ground	Signal ground	7
DCD	CF	In[g]	Out	8
DTR	CD	Out	In[h]	20

[a] Selectable, 110-9600 baud
[b] DTE 1, DTE 2 modes; goes high when data to send
[c] DCE mode; must be high before 8540 accepts data
[d] DTE 1 mode; must be high before 8540 sends data
[e] DCE mode; goes low when 8540 sends data
[f] DTE 2 mode; must be high before 8540 sends data
[g] DTE 1, DTE 2 mode; must be high before 8540 accepts data
[h] DCE mode; must be high before 8540 sends data

Integration Tools

Emulators & Probes
RTPA
TTA
MAC Board

Emulator Processor and Prototype Control Probe Support Packages

The 8500 Microprocessor Development Lab family support a wide variety of different microprocessors and microcomputers.

Emulators

Emulator packages may be ordered as systems options. These options provide the capabilities necessary to fully emulate the target microprocessor in a user's prototype system.

The emulator processor, which resides on a plug-in circuit module along with controlling logic circuitry, enables the user to execute and debug the program on a microprocessor identical to the one which will be used in the prototype, while giving him access up to 256K bytes of Microprocessor Lab program memory.

Software execution is performed under control of DOS-50 in the 8550 system, and under control of TNIX in the 8560 system. During software development, DOS-50 allows the creation of a tree-like file structure with subdirectories to whatever nesting level a project requires. When converting source code to object code, it provides complete supervision of assembly or compilation procedures. During hardware/software integration, DOS-50 handles prototype execution monitoring and debugging operations. It also takes care of general I/O, intersystem communication and PROM programming.

TNIX, the operating system used in the 8560 system uses timesharing to apportion system resources among up to eight workstations plus system utilities. A hierarchial-type filing system is used that groups files and directories logically. Each file carries a date/time attribute to help confirm that the proper version is being accessed. A security system allows password protection, multiple-user file access, and work copies according to current project needs.

Probes

The prototype control probe connects the emulator processor card to the prototype microprocessor, and allows a designer to transfer program control in three stages from the 8550 or 8560/8540 MDL to the prototype.

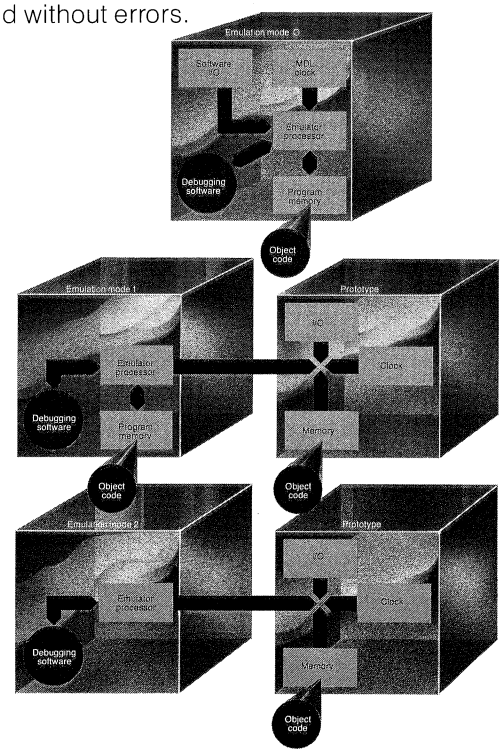
All emulation operations are controlled by the powerful Microprocessor Lab system software. The user is able to monitor program execution, set software breakpoints, examine and change memory and register contents. Debug trace information is displayed in a format unique to the microprocessor, with instruction fetches disassembled into mnemonics for easy interpretation.

Three Emulation Modes

Once an object code file has been created, the program may be exercised in system mode 0. During execution, program steps can be traced, hardware breakpoints can be set, and memory can be examined and changed as required. Should an error be discovered, that portion of the program can be corrected at the source level using the text editor. Object code can then be reassembled and reexecuted. This procedure continues until the developmental software program is complete and can be executed without errors.

After the developmental software has been debugged, it may be exercised on the prototype circuitry (mode 1). During partial emulation, control may be released in stages to the prototype. Developmental software is sequenced using the emulator memory space and the prototype I/O and clock. The memory mapping feature allows code to be gradually mapped over to the prototype's memory in blocks. Throughout partial emulation, the program designer has access to prototype circuitry. This access enables the examination and changing of memory and register contents, hardware tracing, and setting of breakpoints.

In full emulation, mode 2, the developmental software program is run on the prototype, but program execution is still under the complete control of the 8550 MDL or 8540 system. All I/O and timing functions are directed by the prototype and all memory has been mapped over to it. The Prototype Control Probe is connected and still emulating the prototype microprocessor. Although the prototype is effectively free-standing, the designer may still direct program activity from the console terminal.



8080A and 8085A Emulator Processors and Prototype Control Probes

8080/8080A and 8085/8085A refer to microprocessors manufactured by Intel Corporation. Tektronix, Inc. does not guarantee that other vendors' versions of the 8080/8085 will be compatible with the Tektronix Microprocessor Labs.

8080A Emulator Support Package Characteristics

PHYSICAL CHARACTERISTICS

Length: 6-foot cable from the Emulator Processor to the interface assembly. 1.5-foot cable from the interface assembly to the 40-pin plug.

Configuration, 6-foot cable: Two 40-conductor groundplane ribbon cables with chassis groundplane and signal paths.

Configuration, 1.5-foot cable: Two twisted pair of 40-conductor cables.

Termination, 6-foot cable: The interface assembly contains resistive termination and receivers for data, address and control from the Emulator Processor module. (The 1.5-foot cable is not terminated.)

AC CHARACTERISTICS

Emulation Clock	
Mode 1 or Mode 2 (user's clock) with 8085A prototype control probe.	6.25 MHz max; crystal, RC timing network or TTL input to X1.
Mode 0 (system clock)	6.25 MHz ± 0.01%.

8085A Emulator Support Package Characteristics

PHYSICAL CHARACTERISTICS

Length: 6-foot cable from the Emulator Processor to the interface assembly. 1-foot cable from the interface assembly to the 40-pin plug.

Configuration, 6-foot cable: Two 40-conductor ribbon cables with chassis groundplane and signal paths.

Configuration, 1-foot cable: Two twisted pair of 40-conductor cables.

Termination, 6-foot cable: The interface assembly contains receivers for data, address and control from the Emulator Processor module. (The 1-foot cable is not terminated.)

TIMING CHARACTERISTICS
Representative 8080A Emulator 8080A Microprocessor Timing Differences

Symbol	Parameter	Processor Min.	Processor Max.	Emulator[a] Min.	Emulator[a] Max.	Unit
tCY	Clock Period	.48	2.0	.48	2.0	μs
tr,tf	CLK Rise and Fall Time	0	50	0	50	ns
tDA	Address Output Delay from $\phi 2$		200		185	ns
tDD	Data Output Delay from $\phi 2$		220		220	
tDC	Signal Output Delay from $\phi 1$ or $\phi 2$ (SYNC, WR, WAIT, HLDA)		128		165	ns
tDF	DBIN Delay from $\phi 2$	25	140	40	180	ns
tDI	Delay for Input Bus to Enter Input Mode		tDF	48	205	ns
tDS1	Data Setup Time during $\phi 1$ and DBIN	30		69		ns
tDS2	Data Setup Time to $\phi 2$ During DBIN	150		189		ns
tDH	Data Hold Time from $\phi 2$ during DBIN	[b]		[b]		ns
tIE	INTE Output Delay from $\phi 2$		200		255	ns
RS	READY Setup Time during $\phi 2$	120		130		ns
tHS	HOLD Setup Time to $\phi 2$	140		180		ns
tIS	INT Setup Time during $\phi 2$ (during $\phi(1)$ in Halt Mode)	120		190		ns
tH	Hold Time from $\phi 2$ (READY, INT, HOLD)	0		[c]		
tFD	Delay to Float during Hold (Address and Data Bus)		120		[d]	ns
tAW	Address Stable Prior to WR	[e]		[f]		ns
tDW	Output Data Stable Prior to WR	[g]		[h]		ns
tWD	Output Data Stable from WR	[i]		[j]		ns
tWA	Address Stable from WR	[i]		[j]		ns
tHF	HLDA to Float Delay	[k]		[d]		ns
tWF	WR to Float Delay	[l]		[d]		ns
tAH	Address Hold Time after DBIN	−20		−50		ns

a. All emulator delays are given with reference to $\phi 1$ and $\phi 2$ as applied to the emulator CPU. The delay from the prototype $\phi 1$ and $\phi 2$ to the CPU $\phi 1$ and $\phi 2$ is 98 ns (maximum) and 34 ns (minimum).

b. Data input should be enabled with DBIN status. No bus conflict can then occur and data hold time is assured. tDH = 50 ns or tDF whichever is less.

c. READY = − 10 ns minimum INT = − 30 ns minimum HOLD = − 15 ns minimum

d. Emulator address and data bus float is based upon (HLDA x DBIN). Maximum float delay is 220 ns after the rising edge of $\phi 1$ for non-DBIN cycles and 210 ns after the rising edge of $\phi 2$ for DBIN cycles.

e. tAW = 2tCY − tD3 − trϕ2 − 140 ns

f. tAW = 2tCY − tD3 − trϕ2 − 130 ns

g. tDW = tCY − tD3 − trϕ2 − 170 ns

h. tDW = tCY − trϕ2 − 175 ns

i. If not HLDA, tWD = tWA = tD3 + trϕ2 + 10 ns if HLDA, tWD = tWA = tWF

j. If not HLDA, tWD = tWa = tCY − trϕ2 − 20 ns if HLDA, see note[d]

k. tHF = tD3 + tr$\phi$$ − 50 ns

l. tWF = tD3 + trϕ2 − 10 ns

Timing Assumptions

1. CPU timing reference Intel Component Data Catalog (1979).
2. All emulator timing is relative to clock at emulator CPU. See Note[a] for CLK delays for prototype.
3. Emulator delays are calculated from maximum values in the second edition of The TTL Data Book for Design Engineers, published by Texas Instruments, and minimum delays equal to 20 percent of maximum.

TIMING CHARACTERISTICS
Representative 8085 Emulator 8085 Microprocessor Timing Differences

Symbol	Parameter	Processor		Emulator[a]		Unit
		Min.	Max.	Min.	Max.	
t_r, t_t	CLK Rise and Fall Time		30		30[a]	ns
t_{XKR}	X1 Rise to CLK Rise	30	120	82	191	ns
t_{XKF}	X1Rise to CLK Fall	30	150	82	221	ns
t_{AC}	A8-15 Valid to Leading Edge of Control[a]	270		255		ns
t_{ACL}	A0-7 Valid to Leading Edge of Control	240		225		ns
t_{AD}	Valid Address to Valid Data In		575		540	ns
t_{AFR}	Address Float after Leading Edge of READ INTA		0	−85[b]		ns
t_{AL}	Address Valid before Trailing Edge of ALE	115		100		ns
t_{ALL}	A0-A7 Valid before Trailing Edge of ALE	90		75		ns
t_{ARY}	READY Valid from Address Valid		220		190	ns
t_{CA}	Address (A8-A15) Valid after Control	120		135		ns
t_{CL}	Trailing Edge of Control to Leading EDGE of ALE	50		40		ns
t_{DW}	Data Valid to Trailing Edge of WRITE	420		405		
t_{HABE}	HLDA to BUS ENABLE		210		55[c]	ns
t_{HABF}	Bus Float after HLDA		210		55	ns
t_{HACK}	HLDA Valid to Trailing Edge of CLK	110		80		ns
t_{HDH}	HOLD Hold Time	0		−30		ns
t_{HDS}	HOLD Setup Time to Trailing Edge of CLK	170		190		
t_{INH}	INTR Hold Time	0		[d]		ns
t_{INS}	INSTR, RST, and TRAP Setup time to Falling Edge of CLK	160		[d]		ns
t_{LA}	Address Hold Time after ALE	100		85		ns
t_{LC}	Trailing Edge of ALE to Leading Edge of Control	130		115		ns
t_{LCK}	ALE Low During CLK High	100		85		ns
t_{LDR}	ALE to Valid Data During Read		460		445	ns
t_{LDW}	ALE to Valid Data During Write		200		185	ns
t_{LRY}	ALE to READY Stable		110		95	ns
t_{RAE}	Trailing Edge of READ to Re-enabling of Address	150		135		ns
t_{RD}	READ (or INTA) to Valid Data		300		265	ns
t_{RDH}	Data Hold Time after READ INTA	0		−35		ns
t_{RYH}	READY Hold Time	0		−20		ns
t_{RYS}	READY Setup Time to Leading Edge of CLK	110		130		ns
t_{WD}	Data Valid after Trailing Edge of WRITE	100		85		ns
t_{WDL}	Leading Edge of WRITE to Data Valid		40		25	ns

[a]The 8085A probe cable uses an FET oscillator buffer in the probe tip. For a crystal applied to X1 and X2 the delay to the CPU is a maximum of 18 ns from the clock phase appearing at X2. For an external clock input at X2, the clock presented to the CPU will be inverted (as output at X2) and typically delayed 40 ns.

[b]Address float will occur for a maximum of 45 ns after ALE drops low.

[c]Although bus drivers to prototype will be turned on, CPU drive to drivers will be tri-stated for a maximum of 210 ns.

[d]Maximum input delays. INTR (20 ns)
RST (15 ns)
TRAP (60 ns)

Z80A Emulator Processor and Prototype Control Probe

Z80A is a trademark of Zilog Corp. Tektronix, Inc. does not guarantee that other vendors' versions will be compatible with Tektronix 8550 MDL or 8540 .

Z80A Emulator Support Package Characteristics

PHYSICAL CHARACTERISTICS

Length: 1.8 meter (6 feet) of cable from the Emulator Processor to the interface assembly; 305 millimeter (one foot) of cable from the interface assembly to the 40-pin plug.

Configuration, 1.8 meter cable: Two 40-conductor ribbon cables with chassis ground plane and signal paths.

Configuration, 305 millimeter cable: Two 40-conductor twisted pair cables.

Termination, 1.8 meter cable: The interface assembly contains receivers for data, address, and control from the Z80A emulator processor module. The 305 millimeter cable is not terminated.

TIMING CHARACTERISTICS
Representative Z80A Emulator
Z80A Microprocessor Timing Differences

Symbol	Parameter	Processor		Emulator		Unit
		Min.	Max.	Min.	Max.	
^tC	Clock Period	.25	[1]	.25	[1]	ns
$^tW\phi_H$	Clock Pulse Width, Clock High	110	*a	110	*a	ns
$^tW\phi_L$	Clock Pulse Width, Clock Low	110	2000	110	2000	ns
$^tr, f$	Clock Rise and Fall Time		30	*b	30	ns
$^tD_{AD}$	Address Output Delay		110		130	ns
$^tF_{AD}$	Delay to Float				85*c	ns
^tD_D	Data Output Delay		150		170	ns
^tF_D	Delay to Float During Write Cycle		90		90*f	ns
$^tS\phi_D$	Data Setup Time to Rising Edge of Clock During M1 Cycle	50		30		ns
$^tS\phi_D$	Data Setup Time to Falling Edge of Clock During M2 to M5	60		40		ns
$^tDL\phi_{MR}$	MREQ Delay from Falling Edge of Clock, MREQ Low	20	85	40	105	ns
$^tDH\phi_{MR}$	MREQ Delay From Rising Edge of Clock, MREQ High		85		105	ns
$^tDH\phi_{MR}$	MREQ Delay from Falling Edge of Clock, MREQ High		85		105	ns
$^tDL\phi_{IR}$	IORQ Delay from Rising Edge of Clock, IORQ Low		75		95	ns
$^tDL\phi_{IR}$	IORQ Delay from Falling Edge of Clock, IORQ Low		85		105*d	ns
$^tDH\phi_{IR}$	IORQ Delay from Rising Edge of Clock, IORQ High		85		105	ns
$^tDH\phi_{IR}$	IORQ Delay from Falling Edge of Clock, IORQ High		85		105	ns
$^tDL\phi_{RD}$	RD Delay from Rising Edge of Clock RD Low		85		105	ns
$^tDL\phi_{RD}$	RD Delay from Falling Edge of Clock, RD High		95		115	ns
$^tDH\phi_{RD}$	RD Delay from Rising Edge of Clock, RD High		85		105	ns
$^tDH\phi_{RD}$	RD Delay from Falling Edge of Clock, RD High		85		105	ns
$^tDL\phi_{WR}$	WR Delay from Rising Edge of Clock, WR Low		65		85	ns
$^tDL\phi_{WR}$	WR Delay from Falling Edge of Clock, WR Low		80		100	ns
$^tDH\phi_{WR}$	WR Delay from Falling Edge of Clock, WR High		80		100	ns
$^tDL_{M1}$	M1 Delay from Rising Edge of Clock, M1 Low		100		120	ns
$^tDH_{M1}$	M1 Delay from Rising Edge of Clock, M1 High		100		120	ns
$^tDL_{RF}$	RFSH Delay from Rising Edge of Clock, RFSH Low		130		150	ns
$^tDH_{RF}$	RFSH Delay from Rising Edge of Clock, RFSH High		120		140	ns
$^tS_{WT}$	WAIT Setup Time from Falling Edge of Clock	70		100*e		ns
$^tD_{HT}$	HALT Delay Time from Falling Edge of Clock		300		320	ns
$^tS_{IT}$	INT Setup Time to Rising Edge of Clock	80		110*e		ns

Z80A TIMING CHARACTERISTICS (cont.)

$^tS_{BQ}$	BUSRQ Setup Time to Rising Edge of Clock	50		70*e		ns
$^tDL_{BA}$	BUSAK Delay from Rising Edge of Clock, BUSAK Low		100		130	ns
$^tDH_{BA}$	BUSAK Delay from Falling Edge of Clock, BUSAK High		100		130	ns
$^tS_{RS}$	RESET Setup Time to Rising Edge of Clock	60		80*e		ns
^tF_C	Delay to/from Float (MREQ, IORQ, RD, and WR)		80		150	ns
^{t}mr	M1 Stable Prior to IORQ (Interrupt Ack.)	[2]		[2]*d		ns

[1] t(c) = t(w(ϕH)) + t(w(ϕL)) + t(r) + r(f)
[2] t(mr) = 2* t(c) + t(w(ϕH)) + t(f) − 65

Notes
*a. Although static by design, testing guarantees t(w(%H)) of 200 μs max.
*b. Clock delay from prototype to emulator CPU is 15 ns maximum.
*c. delay measured from BUSAK aserted at CPU.
*d. IORQ can be delayed a maximum of 153 ns during INTA cycles while the Debug TRACE MODE is active.
*e. timing reference to CPU clock — to reference to prototype, subtract propagation delay through buffers (15 ns maximum).
*f. Data will go to an indeterminate state, but will not tri-state unless BUSREQ is asserted.

Timing Assumptions:
* CPU timing reference: Mostek Microcomputer Z80 Data Book; Mostek Corporation, (1978).

6800/6802/6808 Emulator Processor and Prototype Control Probe

6800, 6802 and 6808 are registered trademarks of Motorola Corporation. Tektronix, Inc. does not guarantee that other vendors' versions will be compatible with Tektronix 8550 MDL or 8540.

6800/6802/6808 Emulator Support Package Characteristics

PHYSICAL CHARACTERISTICS

Length: 6 feet of cable from the Emulator Processor to the interface assembly. 1 foot of cable from the interface assembly to the 40-pin plug.

Cable configuration

6 foot: 2 40 conductor ribbon cables with chassis groundplane and signal paths.

1 foot: 2 twisted pair conductor cables made up of signal ground pairs.

TIMING CHARACTERISTICS
Representative 6800 Emulator — 6802 Microprocessor Timing Differences

Symbol	Parameter	Processor Min.	Processor Max.	Emulator[a] Min.	Emulator[a] Max.	Unit
t_{cyc}	Cycle Time	1	10	1	10	μs
t_{AD}	Address Delay		270		[b]	ns
t_{acc}	Peripheral Read Access Time		530		464	ns
t_{DSR}	Data Setup Time (Read)	100		182		ns
t_{H}	Input Data Hold Time	10		26		ns
t_{H}	Output Data Hold Time	30		−25		ns
t_{AH}	Address Hold Time (Address, R/W, VMA)	20		[c]		ns
t_{DDW}	Data Delay Time (Write)		225		[d]	ns
t_{PCS}	Processor Control Setup Time	200		[e]		ns
t_{PCr} t_{PCf}	Processor Control Rise and Fall Time		100		—	ns

[a]All emulator delays are given with reference to the E signal as presented to the user. The delay from the user clock to the E signal is 93 ns maximum and 19 ns minimum.

[b]Address = 328 ns maximum
R/W = 354 ns maximum
VMA = 321 ns maximum

[c]Address = −5 ns minimum
R/W = 0 ns minimum
VMA = −6 ns minimum

[d]Valid data on a write is based on the failing edge of E rather than on the rising edge. Data is valid from the emulator 500 ns (maximum) after the falling edge of E.

[e]NMI = 279 ns minimum
IRQ = 284 ns minimum
RESET = 284 ns minimum

Timing Assumptions. CPU timing reference. The Complete Microcomputer Data Library; published by Motorola Corporation, 1979.

Representative 6800 Emulator — 6800 Microprocessor Timing Differences

Symbol	Parameter	Processor Min.	Processor Max.	Emulator[a] Min.[d]	Emulator[a] Max.[d]	Unit
t_{cyc}	Cycle Time	1	10	1	10	μs
$PW_{\phi H}$	Clock Pulse Width	400	9500	400	9500	ns
t_{AD}	Address Delay		270		[e]	ns
t_{acc}	Peripheral Read Access Time		530		425	ns
t_{DSR}	Data Setup Time (Read)	100		140		ns
t_{HD}	Address Hold Time					
	(address)	30		40		ns
	(R/W)	30		45		ns
	(VMA)	30		40		ns
t_{EH}	Enable High Time for DBE Input	450		—		ns
t_{DDW}	Data Delay Time (WRITE)		225		482[f]	ns
t_{PCS}	Processor Control Setup Time	200		[b]		ns
t_{PCr}	Processor Control Rise and		100		—	ns
t_{PCf}	Fall Time					
t_{BA}	Bus Available Delay		250		305	ns
t_{TSD}	Three-State Delay		270	[c]	[c]	ns
t_{DBE}	Data Bus Enable Down Time During $\phi 1$ Up Time	150		—		ns
t_{DBEr}	Data Bus Enable Rise and		25		—	ns
t_{DBEf}	Fall Time					

6809 Emulator Processor and 6809 Prototype Control Probe

6809 is a registered trademark of Motorola Corporation. Tektronix, Inc. does not guarantee that other vendors' versions will be compatible with Tektronix 8550 MDL or 8540.

6809 Emulator Support Package Characteristics

PHYSICAL CHARACTERISTICS

Length: 6 feet of cable from the Emulator Processor to the interface assembly. 1 foot of cable from the interface assembly to the 40-pin plug.

Cable configuration

6 foot: Two 40-conductor ribbon cables with chassis ground plane and signal paths.

1 foot: Two 22-conductor teflon cables with alternating grounds.

Termination

6 foot: The interface assembly contains receivers for data, address, and control from the 6809 emulator processor module.

1 foot: The probe assembly contains an oscillator circuit to drive and buffer the 6809 clock input pins. Input lines are not terminated. All output or bidirectional lines are series terminated with 100 ohms.

TIMING CHARACTERISTICS
Representative 6809 Microprocessor — 6809 Emulator (Combined with 6809 Prototype Control Probe) Timing Differences

Symbol	Parameter	Processor		Emulator		Unit
		Min.	Max.	Min.	Max.	
^tCYC	Cycle time	0.5	10	0.5	10	μs
^tUT	Total up time	480		480		ns
^tACC	Peripheral read access time	320		278		ns
^tDSR	Data setup time (read)	40		80		ns
^tDHR	Input data hold time	10		−6		ns
^tDHW	Output data hold time	30		18		ns
^tAH	Address hold time	20		8		ns
^tAD	Address delay time		110		122	ns
^tDDW	Data delay time (write)		145		157	ns
^tAVS	E(low) to Q(high) time		125		125	ns
^tAQ	Address valid to Q(high)		15	3		ns
^tPWEL	Processor clock low	210		210		ns
^tPWEH	Processor clock high	220		220		ns
^tPCSM	MRDY (H) setup time	125		169		ns
^tPCS	Interrupts setup time[a]	110		154		ns
^tPCSH	HALT (L) setup time	110		154		ns
^tPCSR	RESET (L) setup time	110		180		ns
^tPCSD	DMA/BREQ (L) setup time	125		169		ns
^tRC	Crystal oscillator start time	100		[b]		ns
^tEr, ^tEf	E rise and fall time	5	20	[c]	[c]	ns
^tPCr, ^tpcf	Processor control rise and fall time		100		[d]	ns
^tQR, ^tQf	Q rise and fall time	5	20	[c]	[c]	ns
^tPWQH	Q clock high time	220		220		ns

Notes:
[a]NMI (L) setup time to E(rising) is 56 ns
[b]Oscillator must be operating before the emulator can run
[c]Outputs are driven by 74LS244's, series terminated with 100-ohm resistors
[d]Rise and fall times are TTL compatible

3870/72/74/76/F8 Emulator Processor and Prototype Control Probes

3870, 3872, 3874 and 3876 are registered trademarks of Mostek. F8 is a registered trademark of Fairchild. Tektronix, Inc. does not guarantee that other vendors' versions will be compatible with Tektronix 8550 MDL.

3870/72/74/76/F8 Emulator Support Package Characteristics

PHYSICAL CHARACTERISTICS

Length: 6-foot cable from the Emulator Processor to the interface assembly. 1-foot cable from the interface assembly to the 40-pin plug.

Configuration

6-foot cable: Two 40-conductor ribbon cables with chassis ground-plane and signal paths.

1.5-foot cable: Two twisted pair of 40-conductor cables.

Termination, 6-foot cable: The interface assembly contains receivers for data, address and control from the 3870, 3872, 3874, 3876, and F8 Emulator Processor module. (The 1-foot cable is not terminated.)

TIMING CHARACTERISTICS
Representative 3870, 3872, F8 Emulator 3870, 3872 Microcomputer Timing Differences

Symbol	Parameter	Processor Min.	Processor Max.	Emulator Min.	Emulator Max.	Units
	3870/3872					
$t_{o(INT)}$	Time Base Period, internal oscillator	250	1000	250[a]	1000	ns
$t_{o(EX)}$	Time Base Period, all external modes	250	1000	250	1000	ns
$t_{EX(H)}$	External Clock Pulse Width High	90	700	40[a]		ns
$t_{EX(L)}$	External Clock Pulse Width Low	100	700	40	ns	
$t_{dI/O}$	Output delay from internal WRITE Clock	0	1000	25[b]	1525[b]	ns
$t_{sI/O}$	Input Setup time to WRITE Clock	1000		340[b]	ns	
$t_{110\text{-}s}$	Output valid to STROBE Delay	$3t\phi - 1000$ $3t\phi - 250$		$3t\phi - 970$ $3t\phi + 280$		ns
t_{RH}	RESET Hold Time, Low	$6t\phi + 750$		$6t\phi + 1040$		ns
t_{EH}	EXT INT Hold Time, Active	$6t\phi + 750$		$6t\phi + 770$		ns
	F8					
P_X	External Input Period	0.5	10	0.5[c]	10	ns
P^W_X	External Pulse Width	200	$P\phi$-200	200	$P\phi$-200	ns
t_{x1}	Ext. to ϕ — to — Delay	20	110	25[f]	140	ns
t_{x2}	Ext. to ϕ + to + Delay	20	125	25[f]	150	ns
t_{d3}	WRITE to ROMC Delay	80	550	105	630	ns
t_{d5}	WRITE to INT REQ + Delay		430		490[d]	ns
t_{d6}	WRITE to INT REQ + Delay		1.65		1.70	ns
t_{SX}	EXT RES set-up time	1.0		1.1		ns
t_{SU}	I/O set up time	300		360		ns
t_H	I/O hold time	50		35		ns
t_{db0}	WRITE to data bus High Impedance		500		683[e]	ns
t_{db3}	Data Bus Set-up	200		235[d]		ns
t_{db4}	Data Bus Set-up	300		335[d]		ns
t_{db5}	Data Bus Set-up	500		535[d]		ns
t_{db6}	Data Bus Set-up	300		335[d]		ns

[a] The 3870/3872 probe cable uses a JFET oscillator buffer in the probe tip. The probe does not support the RC oscillator option, but you may use the mode O emulator clock source in modes 1 and 2 by using jumper P7805 on the probe's driver/receiver board. The external clock is divided by 2 before being sent to the 3850 CPU.

[b] Delay referenced to WRITE signal leaving 3850 CPU (based on CPU I/O times).

[c] The F8 probe cable uses a JFET oscillator buffer in the probe tip. For a crystal applied to XTLY and XTLX, the delay to the CPU is a maximum of 48 ns (referenced to the XTLY crystal drive line). For an external clock input at XTLY, the clock presented to the CPU will be inverted and typically delayed 60 ns. Note that Mostek does not support the RC time mode. Only the crystal and external modes are supported by the emulator.

[d] Table value calculation is referenced to WRITE signal delivered to prototype.

[e] The data bus from CPU to prototype is buffered through a LS245 whose tri-state controls are derived from the CPU's ROMC lines. The F8 prototype control probe will float its data bus under the following conditions:

- Emulator is in mode 0.
- Emulator is in mode 1 and in a read state.
- Emulator is examining CPU registers and is not accessing the prototype circuit.

[f] All emulator timing is relative to the clock presented to the CPU. Time delays in note c should be added to values listed when referencing prototype clock source.

- CPU timing reference. Mostek Microcomputer 3870/F8 Data Book; Mostek Corporation (1979).
- Emulator delays are calculated from maximum values in the second edition of the T.I. TTL Data Book. Minimum values are those referenced in the data book or, in the event that no data value is available, are calculated by (½ • typical) if typical/max < = 67% or by (2 • typical) – max) if typical/max > 67%. Note that table values are calculated maximum (rounded upwards toward the nearest 5 ns), and typical performance should not reach these worst case limits.

1802 Emulator Processor and Prototype Control Probe

1802 is a registered trademark of Rockwell, Inc. Tektronix, Inc. does not guarantee that other vendors' versions will be compatible with the Tektronix 8550 MDL.

1802 Emulator Support Package Characteristics

PHYSICAL CHARACTERISTICS

Length: 6-foot cable from the Emulator Processor to the interface assembly. 1.5-foot cable from the interface assembly to the 40-pin plug.

Configuration, 6-foot cable: Two 40-conductor ribbon cables with chassis groundplane and signal paths.

Configuration, 1.5-foot cable: Two twisted pair of 40-conductor cables.

Termination, 6-foot cable: The interface assembly contains resistive termination for data, and control from the Emulator Processor module. (The 1.5-foot cable is not terminated.)

TIMING CHARACTERISTICS
Representative 1802 Emulator — 1802 Microprocessor Timing Differences

Parameter	V_{CC}	V_{DD}	Processor Maximum	Emulator Maximum	Units
EF 1-4 Setup	5	5	0	355	ns
	5	10	0	355	ns
	10	10	0	155	ns
EF 1-4 Hold	5	5	250	130	ns
	5	10	200	130	ns
	10	10	125	95	ns
Minimum Pulse Width, t_{WL}					
CLEAR Pulse Width	5	5	600	600	ns
	5	10	400	400	ns
	10	10	300	300	ns
CLOCK Pulse Width, t_{WL}	5	5	200	200	ns
	5	10	160	160	ns
	10	10	100	100	ns

a. The emulator CLOCK to (CPU DATA to BUS) characteristic is based on the leading (falling) edge of T4 rather than the rising edge of T1. The delay from the leading edge of T4 to valid data is 428 ns (max) at V_{CC} = 5 V, and 303 ns (max) at V_{CC} = 10 V.

b. The emulator cannot sample the WAIT signal on the falling edge of every clock pulse. Instead, the emulator samples the WAIT line on the leading (falling) edges of T5, T6, and T7. The delay from the time user WAIT is applied until the time it is sampled is 230 ns (max) at V_{CC} = 5 V and 155 ns (max at V_{CC} = 10 V. The delay from the time that WAIT is sampled until the time it is applied to the CPU is 183 ns (max) at V_{CC} = 5 V and 158 ns (max) at V_{CC} = 10 V.

Timing Assumptions:

1. Timing references. The TTL Data Book for Design Engineers: Texas Instruments, Inc. (1976). CMOS Databook; National Semiconductor Corporation, (1978). RCA Microprocessor Products, File Number 1023; RCA (1979).
2. All emulator delays are given with reference to CLOCK as applied to the emulator CPU. The delay from the user CLOCK to the CPU CLOCK is 342 ns (max). 68 ns (min) at V_{CC} = 5 V and 202 ns (max) 40 ns (min) at V_{CC} = 10 V.
3. USER CLOCK delay to USER XTAL is 92 ns (max) at V_{CC} = 5 V and 62 ns (max) at V_{CC} = 10 V.
4. Emulator delays may be less than processor delays because the emulator CPU is a specially selected part, and is operated with V_{DD} = 10 V in all circumstances.
5. The emulator probe draws no V_{DD} current from the user prototype. It's V_{CC} current draw is less than 2.5 mA.

TIMING CHARACTERISTICS
Representative 1802 Emulator — 1802 Microprocessor Timing Differences

Parameter	V_{CC}	V_{DD}	Processor Maximum	Emulator Maximum	Units
Propagation Delay Time, t_{PLH}, t_{PLH}					
Clock to TPA, TPB	5	5	450	350	ns
	5	10	350	350	ns
	10	10	200	195	ns
Clock-to-Memory High Address Byte	5	5	1350	530	ns
	5	10	750	530	ns
	10	10	600	380	ns
Clock-to-Memory Low Address Byte	5	5	500	350	ns
	5	10	375	350	ns
	10	10	250	195	ns
Clock to MRD, t_{PLH}	5	5	450	440	ns
	5	10	350	440	ns
	10	10	200	215	ns
Clock to MRD, t_{PHL}	5	5	450	440	ns
	5	10	350	440	ns
	10	10	200	215	ns
Clock to MWR, t_{PLH}, t_{PHL}	5	5	450	440	ns
	5	10	350	440	ns
	10	10	200	215	ns
Clock to (CPU DATA to BUS	5	5	650		ns
	5	10	450	a	ns
	10	10	300		ns
Clock to State Code (SCO)	5	5	750	420	ns
	5	10	450	420	ns
	10	10	300	270	ns
Clock to State Code (SC1)	5	5	750	610	ns
	5	10	450	610	ns
	10	10	300	340	ns
Clock to Q	5	5	550	350	ns
	5	10	400	350	ns
	10	10	250	195	ns
Clock to N(O-2), t_{PHL}	5	5	800	610	ns
	5	10	500	610	ns
	10	10	350	340	ns
Minimum Setup and Hold Times, t_{SU}, t_H					
Data Setup	5	5	0	140	ns
	5	10	10	240	ns
	10	10	20	130	ns
Data Hold	5	5	300	155	ns
	5	10	200	155	ns
	10	10	150	120	ns
DMA Setup	5	5	0	255	ns
	5	10	0	255	ns
	10	10	0	105	ns
DMA Hold	5	5	250	150	ns
	5	10	200	150	ns
	10	10	125	105	ns
Interrupt Setup	5	5	0	270	ns
	5	10	0	270	ns
	10	10	0	170	ns
Interrupt Hold	5	5	250	150	ns
	5	10	200	150	ns
	10	10	125	95	ns
WAIT Setup	5	5	0		ns
	5	10	25	b	ns
	10	10	50		ns

8048/35/39/39-6/49/41A/21/22/8748/8749

8048, 8035, 8039, 8039-6, 8049, 8041A, 8021, 8022, 8748 and 8749 are registered trademarks of Intel Corporation. Tektronix, Inc. does not guarantee that other vendors' versions will be compatible with Tektronix 8550 MDL or 8540 .

8048 / 35 / 39 / 39-6 / 49 / 41A / 21 / 22 / 8748 / 8749 Package Characteristics

PHYSICAL CHARACTERISTICS

Length: 6-foot cable from the Emulator Processor to the interface assembly. 1.5-foot cable from the interface assembly to the 40-pin plug.

Configuration

6-foot cable: Two 40-conductor ribbon cables with chassis ground-plane and signal paths.

1.5-foot cable: Two twisted pair of 40-conductor cables.

Termination, 6-foot cable: The interface assembly contains receivers and drivers for data, address and control on both the emulator and probe. All lines have resistive series termination at the drivers.

Maximum Allowable Clock Frequencies
8048/8021/8041A/8022 Emulator Processor

Emulation Mode	Device Being Emulated			
	8049,8039	8048,8748 8035,8039-6	8041A,8741A	8021,8022
	Maximum allowable clock frequency			
Mode 1 — Mapped to Program Memory	11.0 MHz	6.0 MHz	6.0 MHz	3.58 MHz
Mode 1 — Mapped to Prototype Memory Mode 2 — Prototype Memory only	11.0 MHz	6.0 MHz	External (prototype) memory is not addressable by these microcomputers	

DELAY CHARACTERISTICS
Probe/Prototype Interface Delays for the 8022

	Signal	^tPLH Maximum	^tPHL Maximum	Units
ALE		34	46	ns
P00-P07	CPU to prototype	87	91	ns
P00-P07	prototype to CPU	1.3	1.3	μs
P10-P17	t_1 — CPU to prototype	2	2	ns
	t_2 — Prototype to CPU	2	2	ns
P20-P23[ab]	t_3 — MOVD P2.A	18	24	ns
	t_4 — MOVD A, P2	18	24	ns
	t_5 — IN A, P2	18	24	ns
PROG		18	24	ns
T0		24	24	ns
T1		182	182	ns
AN0, AN1[c]		444	444	ns
XTAL1		33	45	ns
RESET		229	229	ns

[a]Inputs must be present until read by an input instruction (Intel Specification)
[b]For OUTL, ORL, and ANL instructions, data will be valid before ALE following the next instruction fetch
[c]Input capacitance: 37 pf maximum.

DELAY CHARACTERISTICS
Probe/Prototype Interface Delays for the 8041A

Signal	^tPHL (ns) Maximum	^tPLH (ns) Maximum
SYNC	20	20
PROG	20	20
T1		39
P10-P17	2	2
T0	45	34
P20-P27	a	a
SS	76	76
RESET	230	230

[a]OUT, ANL, and ORL information is valid before ALE following the next instruction fetch.

DELAY CHARACTERISTICS
Probe/Prototype Interface Delays for the 8048/8021

Signal		^tPHL (ns) Maximum	^tPHL (ns) Maximum
ALE		20	20
PSEN		30	30
RD, WR		35	31
PROG		20	20
DB0-DB7[a]	t_1 — fetch cycle	90	90
(P00-P07)	t_2 — execute cycle	38	38
Prototype to CPU			
DB0-DB7	t_3 — address out	38	38
(P00-P07)	t_4 — external data out	38	38
CPU to Prototype	t_5 — OUTL, ANL, ORL data out	b	b
P10-P17, P24-P27		2	2
P20-P23		c	c
T0 out/in		15	15
T1			182
INT		32	32
RST (^{1}PHL for 8021, ^{1}PLH for 8048)		284	284
SS		32	32
CLK		79	79

[a] t(RD) = t(1.2) + t(prototype memory access)

[b] OUTL, ANL, and ORL bus I/O information is latched during S1 of the second cycle of the instruction.

[c] When external program memory is enabled (by a DIP switch in the prototype control probe), the maximum address delay is 32 ns. When external program memoyr is disabled. OUTL, ANL, and ORL information is latched during the S3 cycle following the next instruction fetch.

TIMING CHARACTERISTICS
Representative 8041A Probe/8041A Microcomputer Timing Comparison

Symbol	Parameter	Processor Min.	Processor Max.	Emulator Min.	Emulator Max.	Units
DBB READ						
^tAR	CS, A_0 Setup to RD fall	0		0		ns
^tRA	CS, A_0 Hold After RD rise	0		0		ns
^tRR	RD Pulse Width	250		250		ns
^tAD	CS, A_0 to Data Out Delay		225[a]		95	ns
^tRD	RD fall ot Data Out Delay		225[a]		95	ns
^tDF	RD rise to Data Float Delay		100		75	ns
^tCY	Cycle Time (Except 8741A-8)[b]	2.5	15	2.5	15	μs
^tCY	Cycle Time (8741A-8)[c]	4.17	15	4.17	15	μs
DBB WRITE						
^tAW	CS, A_0 Setup to WR fall	0		0		ns
^tWA	CS, A_0 Hold After WR fall	0		24		ns
^tWW	WR Pulse Width	250		250		ns
^tDW	Data Setup to WR rise	150		150		ns
^tWD	Data Hold After WR rise	0		70		ns
DMA						
^tACC	DACK fall to WR or RD	0		54		ns
^tCAC	RD or WR to DACK rise	0		71		ns
^tACD	DACK fall to Data Valid		225[a]		225	ns
^tCRQ	RD or WR to DRQ Cleared		200		200	ns

[a] C_L = 150 pF

[b] 6.0 MHz Xtal

[c] 3.6 MHz Xtal

6500/1 Emulator Processor and Prototype Control Probe

6500/1 is a trademark of Rockwell International Corporation. Tektronix, Inc. does not guarantee that other vendors' versions will be compatible with the Tektronix 8550 MDL.

6500/1 Emulator Support Package Characteristics

PHYSICAL CHARACTERISTICS

Length: 1.8 m (6 ft.) of cable from the emulator processor to the interface assembly. 45 cm (1.5 ft.) of cable from the interface assembly to the 40-pin plug.

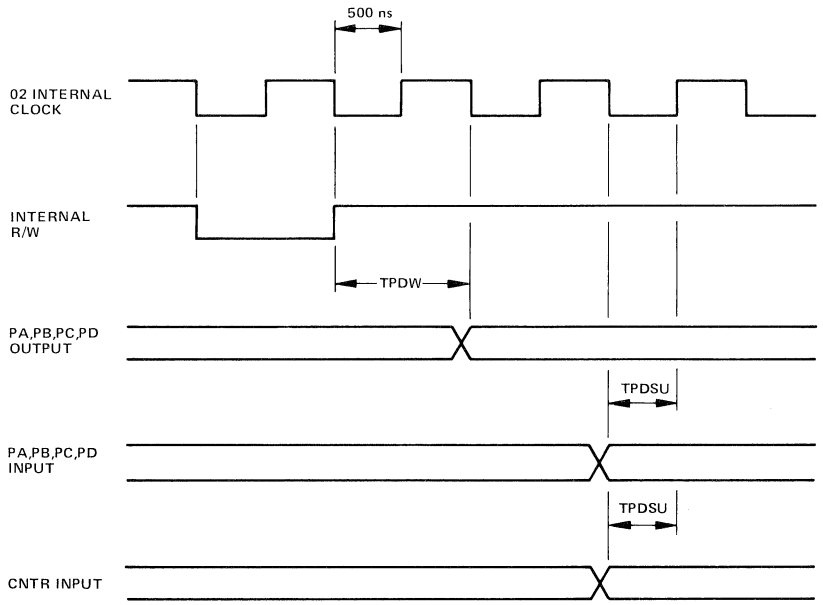
Cable Configuration

1.8 m (6 ft.) Two 40 conductor ribbon cables with chassis groundplane and signal paths.

45 cm (1.5 ft.) Two laminated 40 conductor cables made up of signal-ground pairs.

6500/1 TIMING CHARACTERISTICS AND EMULATION INTERFACE DELAYS

6500/1 TIMING CHARACTERISTICS AND EMULATION INTERFACE DELAYS



TPDW = 1 μs MIN
TPDSU = 400 ns MIN

TYP DELAYS IN ns ADDED TO 6500/1E BY EMULATOR WITH 6.8K PULL UP RESISTORS

		OUTPUT DRIVING	INPUT RECEIVING
PA0–PA7	RISING EDGE	1 CLK CYCLE +300	100
	FALLING EDGE	1 CLK CYCLE +300	100
PB0–PB7, PC0–PC7, PD0–PD7	RISING EDGE	300	*
	FALLING EDGE	30	*
CNTR	RISING EDGE	100	100
	FALLING EDGE	20	20

*GATED IN ONLY DURING A READ INSTRUCTION FROM 81, 82, 83

Z8001/Z8002 Emulator Processor and Prototype Control Probes

Z8001 and Z8002 are trademarks of Zilog Corp. Tektronix, Inc. does not guarantee that other vendors' versions will be compatible with Tektronix 8550 MDL or 8540.

Z8000A Emulator Support Package Characteristics

PHYSICAL CHARACTERISTICS

Length: 6 feet of cable from the Emulator Processor to the interface assembly; 1.5 feet of cable from the interface assembly to the 40- or 48-pin plug.

Cable configuration

6 foot: Two transmission line cables, each 40 signal lines and 80 signal return lines terminated to 40-pin connector.

1.5 foot for Z8001: Two transmission line cables, each 40 signal lines and 80 signal return lines terminated to 40-pin connector.

1.5 foot for Z8002: Two transmission line cables, each 26 signal lines and 52 signal return lines terminated to 26-pin connector.

Termination

6 foot: The interface assembly contains receivers for data, address, and control from the Z8000 emulator processor module.

1.5 foot: Address and data lines terminated with Schottky diodes. Control lines driven by pod are source terminated. Input lines to the emulator are received with PNP inputs. Clock, Wait, Stop, and Reset inputs are buffered in the probe to maintain signal quality.

TIMING CHARACTERISTICS
Representative Z8000 Emulator
Z8000A Microprocessor* Timing Differences

Symbol	Parameter	Processor		Emulator		Unit
		Min.	Max.	Min.	Max.	
t_{cC}	Clock Cycle Time	165	2000	165	2000	ns
t_{wCh}	Clock Width (High)	70	2000	70	2000	ns
t_{wCl}	Clock Width (Low)	70	2000	70	2000	ns
t_{fC}	Clock Fall Time		10		10	ns
t_{rC}	Clock Rise Time		10		10	ns
$t_{dC(SNv)}$	Clock ↑ to Segment Number Valid (50 pF load)		110		133	ns
$t_{dC(SNn)}$	Clock ↑ to Segment Number Not Valid	10		22		ns
$t_{dC(Bz)}$	Clock ↑ to Bus Float		55		133	ns
$t_{dC(A)}$	Clock ↑ to Address Valid		75		101	ns
$t_{dC(Az)}$	Clock ↑ to Address Float[a]		55		103	ns
$t_{dA(DI)}$	Address Valid to Data in Required Valid[b]	305		275		ns
$t_{sDI(C)}$	Data in to Clock ↓ Setup Time	20		29		ns
$t_{dDS(A)}$	DS ↑ to Address Active[b]	40		48		ns
$t_{dC(DO)}$	Clock ↑ to Data Out Valid		75		101	ns
$t_{hDI(DS)}$	Data in to DS ↑ Hold Time	0		0		ns
$t_{dDO(DS)}$	Data Out Valid to DS ↑ Delay[b]	195		187		ns
$t_{dA(MR)}$	Address Valid to MREQ ↓ Delay[b]	(35)		27		ns
$t_{dC(MR)}$	Clock ↓ to MREQ ↓ Delay		70		97	ns
t_{wMRh}	MREQ Width (High)[b]	135		126		ns
$t_{dMR(A)}$	MREQ ↓ to Address Not Active[ab]	35		67		ns
$t_{dDO(DSW)}$	Data Out Valid to DS ↓ (Write) Delay[b]	35		27		ns
$t_{dMR(DI)}$	MREQ ↓ to Data in Required Valid[b]	225		194		ns
$t_{dC(MR)}$	Clock ↓ MREQ ↑ Delay		60		87	ns
$t_{dC(ASf)}$	Clock ↑ to AS ↓ Delay		60		87	ns
$t_{dA(AS)}$	Address Valid to AS ↑ Delay[b]	35		27		ns
$t_{dC(ASr)}$	Clock ↓ to AS ↑ Delay		80		108	ns
$t_{dAS(DI)}$	AS ↑ to Data in Required Valid[b]	215		184		ns
$t_{dDS(AS)}$	DS ↑ to AS ↓ Delay[b]	35		26		ns
t_{wAS}	AS Width (Low)[b]	55		45		ns
$t_{dAS(A)}$	AS ↑ to Address Not Active Delay[b]	30		20		ns
$t_{dAz(DSR)}$	Address Float to DS (Read) ↓ Delay	0		−28		ns
$t_{dAS(DSR)}$	AS ↑ to DS (Read) ↓ Delay[b]	35		25		ns
$t_{dDSR(DI)}$	DS (Read) ↓ to Data in Required Valid[b]	130		100		ns
$t_{dC(DSr)}$	Clock ↓ to DS ↑ Delay		65		92	ns
$t_{dDS(DO)}$	DS ↑ to Data Out and STATUS Not Valid[b]	45		36		ns
$t_{dA(DSR)}$	Address Valid to DS (Read) ↓ Delay[b]	110		102		ns
$t_{dC(DSR)}$	Clock ↑ to DS (Read) ↓ Delay		85		112	ns
t_{wDSR}	DS (Read) Width (Low)[b]	185		176		ns
$t_{dC(DSW)}$	Clock ↓ to DS (Write) ↑ Delay		80		107	ns
t_{wDSW}	DS (Write) Width (Low)[b]	110		101		ns
$t_{dDSI(DI)}$	DS (Input) ↓ to Data in Required Valid[b]	200		170		ns

Clock-Cycle-Time-Dependent Characteristics

Symbol	Z8001/Z8002 Emulator Equation
$^tdA(DI)$	$2^tcC + ^twCh - 125$ ns
$^tdDS(A)$	$^twC1 - 22$ ns
$^tdDO(DS)$	$^tcC + ^twCh - 48$ ns
$^tdA(MR)$	$^twCh - 43$ ns
^twMRh	$^tc - 39$ ns
$^tdMR(A)$	$^twCl - 3$ ns
$^tdDO(DSW)$	$^twCh - 48$ ns
$^tdMR(DI)$	$2^tcC - 136$ ns
$^tdA(AS)$	$^twCh - 48$ ns
$^tdAS(DI)$	$2^tcC - 146$ ns
$^tdDS(AS)$	$^twCl - 44$ ns
^twAS	$^twCh - 25$ ns
$^tdAS(A)$	$^twCl - 50$ ns
$^tdAS(DSR)$	$^twCl - 45$ ns
$^tdDSR(DI)$	$^tcC + ^twCh - 135$ ns
$^tdDS(DO)$	$^twCl - 34$ ns
$^tdA(DSR)$	$^tcC - 63$ ns
^twDSR	$^tcC + ^twCh - 59$ ns
^twDSW	$^tcC - 64$ ns
$^tdDSI(DI)$	$2^tcC - 160$ ns
^twDS	$2^tcC - 84$ ns
$^tdAS(DSA)$	$4^tcC + ^twCl - 50$ ns
$^tdDSA(DI)$	$2^tcC + ^twCh - 135$ ns
$^tdS(AS)$	$^twCh - 49$ ns

Z8000 TIMING CHARACTERISTICS (cont.)

$^tdC(DSI)$	Clock ↓ to DS (I/O) ↓ Delay		100		127	ns
^twDS	DS (I/O) Width (Low)[b]	255		246		ns
$^tdAS(DSA)$	AS ↑ to DS (Acknowledge) ↓ Delay[b]	680		680		ns
$^tdC(DSA)$	Clock ↑ to DS (Acknowledge) ↓ Delay		85		112	ns
$^tdDSA(DI)$	DS (Ack.) ↓ to Data in Required Delay[b]	295		265		ns
$^tdC(S)$	Clock ↑ to Status Valid Delay		85		118	ns
$^tdS(AS)$	Status Valid to AS ↑ Delay	30		21		ns
$^tsR(C)$	RESET to Clock ↑ Setup Time[b]	70		95		ns
$^tnR(C)$	RESET to Clock ↑ Hold Time	0		0		ns
^twNMI	NMI Width (Low)	70		35		ns
$^tsNMI(C)$	NMI to Clock ↑ Setup Time	70		258		ns
$^tsVI(C)$	VI, NVI to Clock ↑ Setup Time	50		65		ns
$^tnVI(C)$	VI, NVI to Clock ↑ Hold Time	0		2		ns
$^tsSGT(C)$	SEGT to Clock ↑ Setup Time	55		92		ns
$^tnSGT(C)$	SEGT to Clock ↑ Hold Time	0		0		ns
$^tsMI(C)$	MI to Clock ↑ Setup Time	110		125		ns
$^tnMI(C)$	MI to Clock ↑ Hold Time	0		2		ns
$^tdC(MO)$	Clock ↑ to MO Delay		85		121	ns
$^tsSTP(C)$	STOP to Clock ↓ Setup Time	70		75		ns
$^tnSTP(C)$	STOP to Clock ↓ Hold Time	0		5		ns
$^tsWT(C)$	WAIT to Clock ↓ Setup Time	30		49		ns
$^thWT(C)$	WAIT to Clock ↓ Hold Time	10		13		ns
$^tbBRO(C)$	BUSRQ to Clock ↑ Setup Time	80		125		ns
$^thBRQ(C)$	BUSRQ to Clock ↑ Hold Time	10		4		ns
$^tdC(BAKr)$	Clock ↑ to BUSAK ↑ Delay		75		111	ns
$^tdC(BAKf)$	Clock ↑ to BUSAK ↓ Delay		75		111	ns

[a] Read cycle only. Line remains active throughout write cycle.
[b] Clock-cycle-time-dependent characteristics.
*Operating a 6 MHz

TMS 9900 Emulator Processor and Prototype Control Probe

TMS 9900 is a registered trademark of Texas Instruments. Tektronix, Inc. does not guarantee that other vendors' versions will be compatible with the Tektronix 8550 MDL.

TMS9900 Emulator Support Package Characteristics

PHYSICAL CHARACTERISTICS

Length: 6-foot cable from the Emulator Processor to the interface assembly. 9.5 inch cable from the interface assembly to the 64-pin plug.

Configuration

6-foot cable: Two 40-conductor ribbon cables with chassis ground plane and signal paths.

9.5 inch cable: Two 64 conductor twisted pair of cables.

Termination, 6-foot cable: The interface assembly contains receivers for address lines from the Emulator Processor module. (The 9.5 inch cable is not terminated.)

TIMING CHARACTERISTICS
Representative 9900 Emulator 9900 Microprocessor Timing Differences

Symbol	Parameter	Processor		Emulator		Unit
		Min.	Max.	Min.	Max.	
$^tC\phi$	Clock Cycle Time	300	500	300	500	ns
$^tr\phi$	Clock Rise Time	5				ns
$^tf\phi$	Clock Fall Time	10				ns
$^tw\phi$	Clock Pulse Width, High Level	40	100	40	100	ns
$^ts\phi$	Clock Spacing, Time between any Two Adjacent Pulses	0		0		ns
$^td\phi$	Time between rising edge Valid any two adjacent pulses	73		73		ns
^{t}SU	Data or Control Setup Time Before Clock 1					
	INTREQ	30		65		ns
	READY	30		65		ns
	IC0-IC3	30		55		ns
	CRUIN	30		55		ns
	LOAD	30		70		ns
	HOLD	30		65		ns
	RESET	30		170		ns
	DATA	30		50		ns
^{t}n	Data Hold time After Clock 1					
	INTREQ	10		0		ns
	READY	10		0		ns
	IC0-IC3	10		5		ns
	CRUIN	10		5		ns
	LOAD	10		0		ns
	HOLD	10		[a]		ns
	RESET	10		−40		ns
	DATA	10		5		ns
$^tPLH_B$	All Other Outputs					
or	ADDRESS		40		60	ns
$^tPHL_B$	IAQ		40		60	ns
	HOLDA		40		90	ns
	CRUOUT		40		65	ns
	DATA		40		320	ns
$^tPLH_C$	Propagation Delay					
or	CRUCLK		30		50	ns
$^tPHL_B$	WE		30		65	ns
	MEMEN		30		65	ns
	WAIT		30		50	ns
	DBIN		30		70	ns

Notes

[a]The user prototype must keep HOLD asserted until HOLDA is received.

Timing Assumptions:

1. CPU timing reference: Texas Instruments 9900 Family Systems Design and Data Book, Texas Instruments, Inc. (1978). Corporation (1978).
2. All emulator delays are given with reference to ϕ1-ϕ4 as applicable to the emulator CPU. The maximum delay from the user ϕ1-ϕ4 to the CPU ϕ1-ϕ4 is 58 ns. The minimum delay from the user ϕ1-ϕ4 to the CPU ϕ1-ϕ4 is 16 ns.
3. The emulator probe draws no V_{BB}, V_{CC}, or V_{CC}, current from the user prototype.

68000 Emulator Processor and Prototype Control Probes

68000 is a trademark of Motorola Corp. Tektronix, Inc. does not guarantee that other vendors' versions will be compatible with Tektronix 8550 MDL or 8540

68000 Emulator Support Package Characteristics

PHYSICAL CHARACTERISTICS

Length: 6 feet of cable from the Emulator Processor to the interface assembly; 1.5 feet of cable from the interface assembly to the 64-pin plug.

Cable configuration

6 foot: Three transmission line cables; Each with 36 signal lines and 72 signal group return lines terminated to a 40-pin connector.

1.5 foot

68000: Two transmission line cables; each with 36 signal lines and 72 signal group return lines terminated to a 40-pin connector.

Termination

6 foot: The interface assembly contains receivers for data, address, and control from the 68000 Emulator Processor Module.

1.5 foot: Address and data lines terminated with Schottky diodes. Control lines driven by pod are source terminated. Input lines to the emulator are received with PNP inputs. Clock, and DTACK inputs are buffered in the probe to maintain signal quality.

68000 Microprocessor/68000 Emulator Timing Differences*

Number	Characteristic	Symbol	Processor		Emulator		Unit
			Min.	Max.	Min.	Max.	
1	Clock Period	tCYC	125	800	125	500	ns
2	Clock Width Low	tCL	86	250	55	250	ns
3	Clock Width High	tCH	86	250	55	250	ns
4	Clock Fall Time	tCI	—	10	—	10	ns
5	Clock Rise Time	tCI	—	10	—	10	ns
6	Clock Low to Address	tCLAV	—	70	—	92	ns
6A	Clock High to FC Valid	tCHFCV	—	70	—	92	ns
7A	Clock High to Address High Impedance (Max)	tCHAZ	—	80	—	114	ns
7B	Clock High to Data High Impedance (Max)	tCHAZ	—	80	—	130	ns
8	Clock High to Address/FC Invalid (Minimum)	tCHAZn	0	—	14	—	ns
9[1]	Clock High to AS DS Low (Maximum)	tCHSL	—	80	—	86	ns
10	Clock High to AS DS Low (Minimum)	tCHSLn	0	—	14	—	ns
11[2]	Address to AS DS (Read) Low/AS Write	tAVSL	30	—	24	—	ns
11A[2]	FC Valid AS DS (Read) Low/AS Write	tFCVSL	80	—	44	—	ns
12[1]	Clock Low to AS DS High	tCLSH	—	70	—	96	ns
13[2]	AS DS High to Address/FC Invalid	tSHAZ	30	—	20	—	ns
14[2,5]	AS DS Width Low (Read)/ AS Write	tSL	240	—	240	—	ns
14A[2]	DS Width Low (Write)	—	115	—	115	—	ns
15[2]	AS DS Width High	tSH	150	—	150	—	ns
16	Clock High to AS DS High Impedance	tCHSZ	—	80	—	93	ns
17[2]	AS DS High to R/W High	tSHRH	40	—	40	—	ns
18[1]	Clock High to R/W High (Maximum)	tCHRHx	—	70	—	96	ns
19	Clock High to R/W High (Minimum)	tCHRHn	0	—	14	—	ns
20[1]	Clock High to R/W Low	tCHRL	—	70	—	96	ns
21[2]	Address Valid to R/W Low	tAVRL	20	—	14	—	ns
21A[2]	FC Valid to R/W Low	tFCVRL	80	—	54	—	ns
22[2]	R/W Low to DS Low (Write)	tRLSL	80	—	80	—	ns
23	Clock Low to Data Out Valid	tCLDO	—	70	—	95	ns
25[2]	DS High to Data Out Invalid	tSHDO	30	—	20	—	ns
26[2]	Data Out Valid to DS Low (Write)	tDOSL	30	—	20	—	ns
27[6]	Data in to Clock Low (Setup Time)	tDICL	15	—	17	—	ns
28[2]	AS DS High to DTACK High	tSHDAH	0	120	0	94	ns
29	DS High to Data Invalid (Hold Time)	tSHDI	0	—	0	—	ns
30	AS, DS High to BERR High	tSHBEH	0	—	0	—	ns
31[2,6]	DTACK Low to Data in (Setup Time)	tDALDI	—	90	—	88	ns
32	HALT and RESET Input Transition Time	tRHrf	0	200	0	200	ns
33	Clock High to BG Low	tCHGL	—	70	—	93	ns
34	Clock High to BG High	tCHGH	—	70	—	93	ns
35[7]	BH Low to BG Low	tBRLGL	1.5	3.0	1.5	3	Clk Por
36[7]	BR High to BG High	tBRHGH	1.5	3.0	1.5	3	Clk Por

68000 TIMING CHARACTERISTICS (cont.)

37	BGACK Low to BG High	tGALGH	1.5	3.0	1.5	3	Clk Por
38	BG Low to Bus High Impedance (With AS High)	tGLZ	—	80	—	148	ns
39	BG Width High	tGH	1.5	—	1.5	—	Clk Por
46	BGACK Width	tBGL	1.5	—	1.5	—	Clk Por
47[6, 8]	Asynchronous Input Setup Time	tASI	20	—	20	—	ns
48	BERR Low to DTACK Low (Note 3)	tBELDAL	50	—	50	—	ns
53	Data Hold from Clock High	tCHDO	0	—	0	—	ns
56	R/W to Data Bus Impedance Change	tRLDO	30	—	20	—	ns
56	High/RESET Pulse Width (Note 4)	tHRPW	10	—	10	—	Clk Por

*Operating at 8 MHz.

1. For a leading capacitance of less than or equal to 50 picofareds, subtract 5 nanoseconds from the values given in these columns.
2. Actual value depends on clock period (see Table 2).
3. If 047 is satisfied for both DTACK and BERR, 048 may be 0 ns.
4. After Vcc has been applied for 100 ns.
5. For T6E, BF4, and R9n mask sets 014 and 014A are one clock clock period less than the given number.
6. If the asynchronous setup time (047) requirements are satisfied, the DTACK low-to-data setup time (031) requirement can be ignored. The data must only satisfy the data-in to clock-low setup time (027) for the following cycle.
7. For the Probe-tip add 20 nanoseconds to the clock periods listed.
8. VPA — 50 nanoseconds for the Probe-tip.

68000 Clock-Cycle-Time Dependent Characteristics

Symbol	Equation	
t6P	t6 + 22 ns	
t7P	t12 − + 106 ns	(for ADDRESS)
	t12 − + 122 ns	(for DATA)
t8P	t8 + 14 ns	
t9P	t9 + 26 ns	
t10P	t10 + 14 ns	
t11P	t11 − 6 ns	
t11AE	t11A − 6 ns	
t12P	t12 + 26 ns	
t13P	t13 − 10 ns	
t16P	t12 − + 85 ns	
t18P	t18 + 26 ns	
t19P	t19 + 14 ns	
t20P	t20 + 26 ns	
t21P	t21 − 6 ns	
t21AP	t21A − 6 ns	
t23P	t23 + 25 ns	
t25P	t25 − 10 ns	
t26P	t26 − 10 Ns	
t27P	t27 + 2 ns	
t28P MAX	t28 MAX − 26 ns	
t33P	t33 + 23 ns	
t34P	t34 + 23 ns	
t35P MIN	1.5T + 20 ns	
t35P MAX	3T + 20 ns	
t36P MIN	1.5T + 20 ns	
t36P MAX	3T + 20 ns	
t38P	t38 + 68 ns	
t47P	t47	For GACK and IP63-2
	t47 + 30 ns	For VPA

$\frac{T}{2}$ $\frac{T}{2}$ $\frac{T}{2}$

8086/8088 Emulator Processor and Prototype Control Probes

8086 and 8088 are trademarks of Intel Corp. Tektronix, Inc. does not guarantee that other vendors' versions will be compatible with Tektronix 8550 MDL or 8540.

8086/8088 Emulator Support Package Characteristics

PHYSICAL CHARACTERISTICS

Length: 6 feet of cable from the Emulator Processor to the interface assembly; 1.5 feet of cable from the interface assembly to the 40-pin plug.

Cable configuration

6 foot: Two transmission line cables, each with 36 signal lines and 72 signal return lines terminated to 40-pin connector.

1.5 foot: Two transmission line cables, each with 22 signal lines and 44 signal return lines terminated to 26-pin connector.

Termination

6 foot: The interface assembly contains receivers for data, address, and control from the 8086/8088 Emulator Processor Module.

1.5 foot: Address and data lines terminated with Schottky diodes. Control lines driven by pod are Schottky terminated. Input lines to the emulator are received with hysteresis inputs. Clock, Ready, NMI, and INT inputs are buffered in the probe to maintain signal quality.

Real-Time Prototype Analyzer

The Real-Time Prototype Analyzer, Option 01 for the 8550 Microcomputer Development Lab, is comprised of a real-time trace module, a data acquisition interface, and an 8 channel general logic probe. This option provides a real-time trace of the user program executing on the emulator processor, with 43 channels of data acquired simultaneously. The prototype address bus, data bus, control bus, and any eight external locations on the prototype circuit may be monitored without slowing up the operational speed of the processor. The Real-Time Prototype Analyzer is indispensable when isolating critical timing errors and hardware/software sequence discrepancies during the final integration phases of prototype development.

The analyzer module is a separate plug-in circuit card that may be inserted into the 8550 system mainframe. The P6451 Probe connects to the prototype circuitry and permits data transference from the prototype to the analyzer. Data from the prototype is buffered and driven by the probe to the data acquisition interface, and then loaded into the analyzer module's realtime trace buffer.

As the user program executes on the emulator processor, 48 bit data words are sequentially acquired from the prototype and loaded into the real-time trace buffer. Each data word contains 16 bit data from the address bus; 8 bit or 16 bit data from the data bus; 8 bit data from the test probe; 3 bit data identifying cycle type (read, write, I/O, memory, or instruction fetch); and 5 bit data used internally to identify last start/stop of the emulator processor. The analyzer will continue to acquire these sequential cycles of logic input until the processor is stopped or the real-time trace buffer is frozen by a specified trigger occurrence. The real-time trace buffer can retain up to 128 data words in pre-, variable center, or post-trigger modes; thus enabling the storage of pertinent program bus transactions.

The Real-Time Prototype Analyzer offers expanded breakpoints to aid in efficient location of prototype problems. Two event comparators located within the analyzer module can be utilized to halt program execution and stop real-time trace. A trigger may be generated on any specific data occurrence in the address bus, data bus, test probe input, and instruction cycle type. Triggering may be immediate; delayed by counting the number of passes; or delayed by counting the number of clock select outputs (clock select may be by microseconds, milliseconds, emulator clocks, etc.). In addition, an output pulse may be generated, via the data acquisition interface, to trigger a logic analyzer or an oscilloscope.

The two event comparators (triggers) may be set to designate a break or halt in the program execution. These comparators may be used as independent breakpoints; or they may be used together to enable a breakpoint on a specific event combination. The program execution can be halted when two trigger events occur simultaneously; when one trigger event precedes another; or when either trigger event occurs. When a break in the program execution takes place, program transactions stored in the real-time trace buffer may be displayed or printed.

Data stored in the real-time trace buffer is displayed sequentially in the order it was acquired from the prototype. Buffer content may be displayed in whole or in part. Optional command parameters are available to limit the storing of data to any specific transaction type, such as memory reads only. If the total buffer contents are displayed, a blank line will separate the data sequence associated with each program starting point.

The Real-Time Prototype Analyzer features a convenient and easy-to-understand display format. With this format, the address location, data, probe input, and control bus data of each acquired transaction are displayed. If the transaction was an instruction fetch, the instruction is also disassembled into the appropriate mnemonic readout unique to the emulator type being used.

The Real-Time Prototype Analyzer functions in all emulation modes and operates with all commercial microprocessors supported by the 8550 MDL.

Input/Output Characteristics

Variable Threshold Range	$>+10$ V dc to <-10 V dc
Preset TTL Voltage	+1.4 V dc ±200 mV
Event Trigger Out	High level voltage out (when V_{cc} = Min, Vi ± 0.5, Ro = 50 Ω to GND) is >2 V dc.

Adjustments: Variable Threshold may be adjusted from $>+10$ V dc to <-10 V dc with a screwdriver adjustment accessible at the rear panel of the Microcomputer Lab. This voltage must be monitored with a voltmeter having an input impedance of at least 10 MΩ.

Jumpers: With the internal jumper in position '0-3' the clock threshold is designated to be the same as channels 0-3. In position '4-7' the jumper designates the clock threshold to be the same as channels 4-7.

Cable Length: 50 cm (19.5 in).

Trigger Trace Analyzer

The Tektronix Trigger Trace Analyzer is designed specifically for use with either the Tektronix 8550 Microcomputer Development Lab or the 8540 Integration Unit. As such, it is totally compatible with 8-bit and 16-bit support offered for either of these 8500-Series systems.

The primary function of the Trigger Trace Analyzer is to capture the real-time execution of code on the bus of the prototype system under design. This capture includes all address and data flow plus control information specific to the processor to be used for the prototype. In addition, an external acquisition probe allows up to eight channels of prototype hardware logic to be captured and displayed along with the bus information.

To capture specific bus transactions and hardware events, the Trigger Trace Analyzer has four trigger channels that simultaneously monitor prototype software execution and other logic transactions. These channels can be used independently or interactively to initiate a trigger upon detection of pertinent data generated from interaction of the emulator processor and prototype. For user convenience when programming the Trigger Trace Analyzer, address information can be represented by a series of symbolic labels and expressions determined by the user. Also, information acquired from the prototype address and control buses is displayed in disassembled mnemonics native to the processor in use.

62 Channels of real-time data acquisition

The Trigger Trace Analyzer (TTA) is a modular option used to monitor and store real-time execution of code on the prototype bus as generated by the emulator processor. This processor may be any one of the 16-bit or 8-bit processors supported by the Tektronix 8500-Series systems. On each cycle the TTA can acquire up to 16 bits of data bus information, up to 24 bits of address information and up to 14 control bits. The particular nature of the control bits is dependent on the specific emulator being used. Each 8500 Series-Emulator package defines and identifies the signals that are supported.

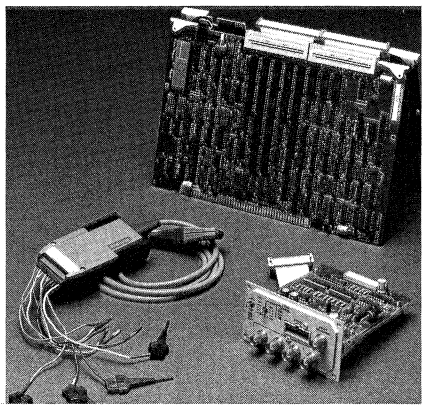
In addition to prototype bus information, the TTA can acquire up to eight channels of hardware logic through a Tektronix P6451 probe with inputs that are either TTL or plus or minus 10 volt variable threshold. This probe is part of the TTA's Acquisition Interface. The Acquisition Interface includes a BNC input for an event qualifier signal that is assignable to any of the TTA's four trigger channels. Also included are BNC outputs for the TTA's four trigger signals.

Real-time data acquired from the prototype bus and/or hardware points is captured by a high-speed buffer, the acquisition trace memory. This memory is up to 62 bits wide and 255 words deep, and is able to resolve bus cycles up to 125 nanoseconds. To optimize the capability of the acquisition trace memory, the TTA also allows data storage qualification based on the event defined by trigger channel four.

Four trigger channels

Each of the TTA's four trigger channels consists of a word recognizer and 16-bit counter that may be used together or independently to produce a trigger (see figure 1). Each channel's word recognizer simultaneously monitors all of the emulator bus and external hardware acquisition bits plus four more bits representing feedback from each channel's counter output. On each bus cycle, the word recognizer looks for a specific value that has been programmed by the user. The data and address portions of the word recognizer will accommodate a range (e.g., 01237H to 35798H) as well as an individual value, and can also exclude a range or individual value. Any of the address, data, and probe signals may also be set to a "don't care" value.

When the data present during a prototype bus cycle agrees with the preprogrammed word recognizer value, the word recognizer outputs an active EVENT signal. If the channel's counter output is also in an active state, the channel will produce a trigger signal. An active EVENT signal can also be used to increment/decrement any channel's counter.



When a given channel's preprogrammed word recognizer and counter values come true, the channel produces an active trigger output. The word recognizer and counter can both be used independently to produce a trigger by setting the value of the other to "don't care." Also the user may program the counter output to be constantly active, allowing the word recognizer to independently produce a trigger.

Each channel's counter is 16 bits (64K). The counter can be programmed to access 17 different counting sources including five clock speeds and trigger signals from other channels. It may be programmed to count up or down to a maximum of 64K, and can be reset during operation. This counting function can be enabled immediately or disabled by an active trigger from its own channel or the previous channel; or by an active counter output from the previous channel. When the counter reaches its preset value, it can be used in conjunction with an active EVENT signal to produce a trigger; or to enable the next channel's counter.

Any channel's active trigger output can cause a program execution breakpoint and halt data acquisition by the TTA's acquisition trace memory. Once the breakpoint has occurred, prototype code execution may either be stopped or allowed to continue through TTA breakpoint commands. Multiple breakpoints are possible by programming different triggers on different channels and setting each to cause a breakpoint.

Up to four prototype events occurring on consecutive bus cycles can be linked to form a single trigger. Each event is assigned to a different channel's word recognizer and then linked through a CONS command (see table 1), that also specifies the type of bus cycle. When the prototype events occur in the order specified, the last event causes a trigger.

Besides triggering capabilities, two other items extend control over data acquisition. One is data qualification, which uses the event programmed into channel four as a determinant for data storage in the acquisition trace memory (see figure 2). When the acquired prototype information agrees with event four programming, it is committed to memory. Another command allows pre-, center- or post-trigger triggering, which determines the position of the trigger event in relation to the acquired data. In this manner the user can acquire events leading up to the trigger, following the trigger, or evenly distributed on either side of the trigger.

The TTA package includes a powerful command set similar to UNIX and a display capability to enhance the user's speed and efficiency. A command summary is provided in table 1.

Table 1. TTA Command Set

Command	Description
acq	Acquire data: Controls data storage in the high-speed trace buffer; allows event four parameters to act as a storage qualifier when specified.
ad	Address comparator: Programs address portion of each channel's word recognizer.
bre	Breakpoint control: Halts TTA data acquisition when a specified channel's trigger output goes active; includes option to let prototype continue executing code
bus	Bus/control signals comparator: Programs control bus portion of each channel's word recognizer; specific control signal mnemonics supplied with each emulator package
cons	Detect consecutive cycles: Specifies sequence in which trigger channel events must come true to cause a data acquisition trigger; selects type of bus cycle to be used
cou	Counter programming: Determines counter value that will cause active counter output; includes increment/decrement select, counting source select, and reset function
ctr	Counter output feedback: Programs counter output feedback portion of each channel's word recognizer
data	Data comparator: Programs data portion of each channel's word recognizer
disp	Display acquisition memory: Displays acquired bus activity in disassembled mnemonics native to the emulator processor in use
eve	Event comparators: Single command allowing all of a channel's word recognizer values to be programmed at once
pro	External probe input comparator: Programs probe input portion of each channel's word recognizer (8 bits)
qua	External qualifier comparator: Programs external qualifier portion of each channel's word recognizer (single bit received via BNC input)
tclr	Clear an entire trigger channel (event, counter and breakpoint)
ts	Trigger status display: Shows all trigger information currently programmed on each of the four trigger channels including breakpoint status

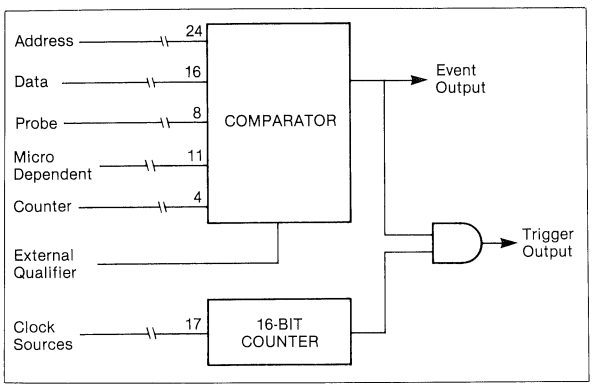


Figure 1. TTA Channel Configuration

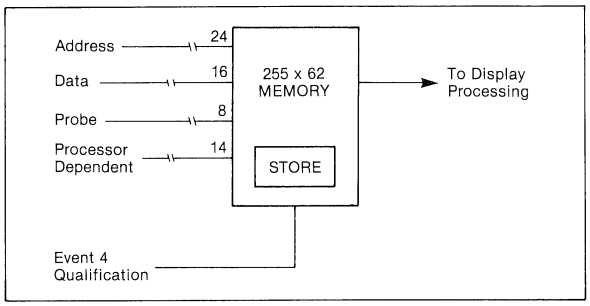
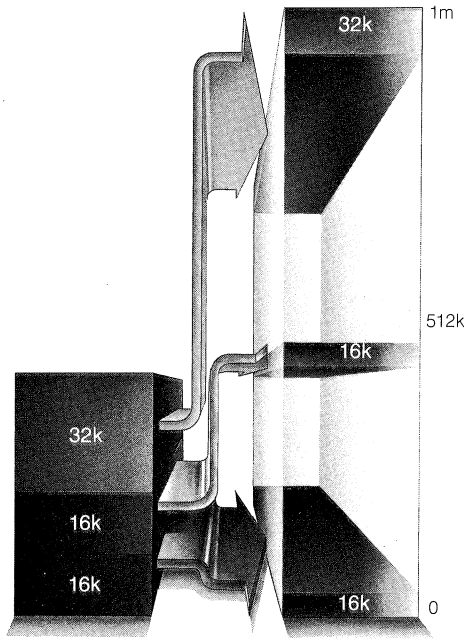


Figure 2. TTA Acquisition Memory

The Memory Allocation Controller

The Memory Allocation Controller is an option to be used with the 68000 and Z8000 support packages. It allows full support of these processors' address ranges, as well as support of the 6 address spaces of the Z8000 and the 4 address spaces of the 68000. Memory is allocated in 4K blocks over the entire address range, and can also be allocated as existent and non-existent, in which case any read or write to non-defined memory will be flagged as an error condition.



Memory Allocation allows program memory to be assigned to different logical addresses within the microprocessor address range.

Software Tools

Assemblers
Library Generator
Linker
Pascal Compiler
MDL/μ Compilers*
Tektronix Pascal Compiler Major Extensions
Advanced CRT-oriented Editor

*8550 Support only

Assemblers

The 8500 Series Microcomputer Development Lab Assembler packages offer time-saving macro, language extension, Library Generator, and Linker features. Macros enable the programmer to write a segment of source code only once, and call it up for in-line code expansion as often as required. Language extensions expand programming capabilities. The Library Generator provides facilities to save libraries of commonly used subroutines for current and future program applications. The linker collects object modules and puts them in a specified order from any number of designated object or library files.

Assembler software operations are fundamentally the same for every microprocessor supported, making it easier for the programmer to change from working with one brand of microprocessor to another. Each assembler recognizes a different instruction set, different registers, and different addressing modes. However, the programmer may use the same assembler directives, operand expressions, symbols, constants, and advanced programming features (such as macros and language extensions).

Language extension functions provide additional programming capability through logical, numeric, and string functions. These standard, pre-defined functions return a value when used in an expression.

Operand expressions may contain bit and string manipulations and special assembler functions as well as standard arithmetic operations. Data constants may be entered as numbers in binary, octal, decimal, or hexadecimal notations, or as strings of ASCII characters enclosed in quotes. Assembly-time string manipulation is allowed, including the use of variable-length strings both inside and outside of macros.

Assembler directives are available to define data items, constants, and variables, provide information to the linker, control macros and conditional assembly, and to specify options for the assembler listing. The assembler listing shows the source code, the generated object code, any error messages with descriptions, and the symbol table produced by the assembler.

All 8560-based assemblers and all 8550-based 16-bit assemblers are large address space (LAS) assemblers, which support additional features, including full 16-bit addressing capabilities, a cross-reference listing of symbols, and the ability to specify classes of logically related sections (for example, all sections to be placed in RAM).

Library Generator

A general-purpose utility program, the Library Generator is available to create and maintain object module libraries. These libraries may contain up to 100 modules from any number of object files. Object modules in library files can be individually accessed by the Linker, based on information provided in each object module. The programmer can modify library files by inserting, deleting, or replacing object modules.

Library Generator commands may be entered through the MDL system terminal or through a command file. When generated, the Library Generator listing shows the commands executed, global symbols, and a summary of Library Generator activity.

Linker

The Linker merges one or more independently-assembled object modules into a load file that is suitable to load into MDL program memory. The load file contains executable program instructions and data. Exclusive address ranges that each program section will occupy are assigned. Any conflicts are noted on the Linker memory map.

Any of the following attributes may be defined or changed at link time: relocation type of a section of object code, exact or approximate location of a section in memory, values assigned to global symbols, and the transfer address. The programmer may specify simple linking operations with a single command line. Special or complex operations can be specified with a series of linker commands entered interactivity from the terminal or from a command file.

In addition to the load file, the linker produces a listing with a detailed account of linker activity including the commands executed, global and internal symbols used, and memory and module maps. Linker statistics such as the number of errors, undefined symbols, modules, sections, and the transfer addresses are also listed.

Pascal Compilers

Pascal, a high-level programming language, is receiving much attention in the electronics industry. Features such as program structure, strong data typing, and readability greatly enhance programmer efficiency, and thereby reduce software development and maintenance costs. The Tektronix Pascal Compiler is a superset of the ISO draft standard Pascal. A true compiler rather than a P-code interpreter, the Pascal Compiler generates object code directly. Each program statement is translated to machine code only once instead of every time the statement is executed, resulting in faster and often more compact code.

Standard Pascal Features

Pascal is a block-structured language that allows the program to be divided into sub-programs called procedures and functions. This block structure encourages programmers to logically plan and construct programs, so debugging time is greatly reduced. The block structure also requires that all variable declarations occur prior to executable code.

Pascal's six control structures correspond closely with flowchart elements and make algorithm coding very natural. All control structures have a single entrance and exit unless GOTO's are used, so program modifications are unlikely to introduce errors into the program.

Pascal allows programmers to use many flexible forms of data representations and to define data types that accurately express their particular problems. Pascal also has strong data typing, which means that each variable must be defined as a single data type prior to its use and used consistently with its definitions.

Pascal programs are easy to ready, and thus to maintain. Pascal differs from most line-oriented languages by allowing extra spaces, tabs, and carriage returns almost anywhere. Variable, procedure, and function names can be meaningful and easily understood because they are not restricted in length. However, identifiers used by DOS/50 must be unique in the first eight characters, other identifiers, in the first 19.

Included with the Pascal Compiler are several predefined procedures and functions used for chip-level I/O. A procedure to send data to a specified port are included. These procedures and functions are analogous to the standard Pascal WRITE, WRITELN, READ, READLN procedures, which are available for mode 0 or 1 operations.

Tektronix Pascal Compiler Major Extensions

Separate Compilations

Separate compilations are supported by the Pascal Compiler. The main program module's first word is the keyword "PROGRAM." Submodules to be separately compiled begin with the keyword "MODULE." Global variables, procedures, and functions can be referenced between separately compiled modules and the main program via PUBLIC and EXTERN attributes. The PUBLIC and EXTERN attributes are associated with variables, procedures, and functions and cause the compiler to generate the appropriate linker text.

Linkage to Assembly Routines

Speed-critical or timing-critical applications are likely to require some program segments to be written in assembly lanaguage. Because the code generated by the Pascal Compiler is compatible with the linker, assembly code can be linked to Pascal code.

Interrupt Handling

The Pascal Compiler supports full use of interrupts. The interrupts are supported by writing the interrupt service routine as a separate procedure having the INTERRUPT attribute. Separate routines are required to connect a specific interrupt vector to the appropriate interrupt service routine. The interrupt service routines are included as convenience routine with the compiler.

Input/Output

Included with the Pascal Compiler are several predefined procedures and functions used for chip-level I/O. A procedure to send data to a specified port are included. These procedures and functions are analogous to the standard Pascal WRITE, WRITELN, READ, READLN procedures, which are available for model 0 or 1 operation when using the operating system I/O. All of the I/O capability is available to a Pascal program running in emulation mode 0 or 1, so the Pascal program can access the console terminal, discs, line printer, and auxiliary I/O ports. The Pascal Compiler also allows an ORIGIN attribute to be associated with variables. The ORIGIN attribute assigns variables to specific memory addresses and is very useful for memory-mapped I/O.

Non-Decimal Integers

In many microcomputer applications, programmers want to use non-decimal integers. The Pascal Compiler supports binary, octal, and hexadecimal integers for input and output.

ROM/RAM Applications

ROM/RAM applications are facilitated by control-section typing. Control-section typing means that the compiler gives the user the information he needs to allocate program variables into a linker section separate from literals, constants, and instructions, which are put into a second linker section.

Structured Constants

Standard Pascal allows only constants of type, integer, real, boolean, and text char. The Pascal Compiler also provides constants which are arrays, and records. The most common application of structured constants is to initialize structured variables (arrays and records) that must reside in potentially volatile RAM.

Metacommands

Metacommands are compiler directives that cause the compiler to do such thing as format the listings or generate run-time debugging code.

MDL/μ Compilers Features

An expanded form of ANSI Minimal BASIC, the MDL/μ Compiler is a high-level language designed specifically for microprocessor-based product development. A system option available with the 8500 series Microcomputer Development Labs, MDL/μ supports software development for the 8080A, the 8085A and the 8080A subset of the Z80 microprocessors. A second system option supports software development for the 6800, 6802, and 6808 microprocessors.

Like Minimal BASIC, MDL/μ is easy to learn and use. Offering additional advantages such as increased flexibility in variable name and string definition, direct access to I/O ports and absolute memory addresses, MDL/μ also provides enhanced function, statement and operator capabilities.

The MDL/μ Compiler produces executable, not interpreted assembly language code. Each program statement is compiled into machine code only once, instead of every time the statement is executed. Thus, fewer operations result in faster and often more compact code for final program execution.

MDL/μ makes possible a module-oriented approach to software development. The programmer may create user-defined libraries of assembly language code or may use routines from the MDL/μ support routine library. These callable routines are stored on flexible disc in object code and are brought into the main program at link time. All support routines except those relating to I/O activities are serially re-entrant; that is they use no temporary values other than those in the registers and on the stack. Register contents are saved when an interrupt occurs and restored when the interrupt is completed.

An additional feature of the MDL/μ Compiler is that two statements, USES and PROVIDES allow variables, functions and procedures to be shared by programmers working on different modules of the main program.

With MDL/μ, variable names and strings are expanded over those available with Minimal BASIC. Variable names can contain up to six characters; the first alphabetic and the others alphanumeric, for easy identification during program development. Strings can vary in length from 1 to 255 characters instead of the unalterable 18 used in Minimal BASIC. Substring replacement is also enhanced to assist in character manipulation.

Useful I/O features, not available with Minimal BASIC, include access to ports and absolute addressing of memory. The source program has direct control of the microprocessor's I/O structures and memory. Both I/O and memory can be accessed from the program as variables. A set of I/O statements including OPEN, CLOSE, RESTORE, READ, WRITE, PRINT, and INPUT allows both ASCII and general-purpose binary file manipulation.

Among the many MDL/μ enhancements to Minimal BASIC are logical operators such as AND, OR, XOR, and NOT. Other enhancements include shift and rotate operations for bit manipulation. DISABLE and ENABLE to turn the interrupt off and on, and built-in code optimization.

Since the MDL/μ compiler outputs assembly language source code, Compiler output may be listed and edited. Editing allows the programmer to optimize sections of code, modify routines for specific needs, or place comments in the assembly language code. The MDL/μ source code appears as comments within the assembly language compilation.

The conversion of MDL/μ source code to actual machine code is a three-step process. The first step converts MDL/μ source code into assembly language source code which is stored on a file or device. The assembly source code contains the original MDL/μ statements as comments preceding each block of assembly source code. At this stage, the assembly language can be further optimized by using the 8550's powerful editor. In the second step the assembler converts the assembly language source into object code. The third step is to link the object code with the run time support library and any other assembled object code modules.

Advanced CRT-oriented Editor

The Advanced CRT-oriented Editor, optional support software available with the 8550/8560 Microcomputer Development Labs, eases program creation and editing tasks. This state-of-the-arts text manipulation program may be used as an alternative to replace the standard 8500 series line oriented editor.

With the Advanced Editor, programmers may conveniently view and edit the program text. A window, the CRT, shows the text surrounding the Advanced Editor's pointer. The text can be edited either as a sequence of lines or as a stream of characters.

Screen Oriented Editing

The Advanced Editor divides the display screen of the CRT into two areas: a monitor area to display messages and where command parameters are input, and a window area to display text that is being edited. The window constantly shows the text surrounding the cursor, a visible pointer in the window area of the screen. The cursor is moved by the user to the specific character within the text where editing will occur. Text displayed in the window can be moved both horizontally and vertically to bring different parts of the file into view. After an edit command has been processed, the window will display the change(s) made to the text.

Character and Line Manipulation

It is often desired to edit text at one place as a sequence of lines and at another place as a stream of characters. The Advanced Editor is versatile, permitting both character and line manipulation. Character manipulation features allow the user to define "wild card" characters for pattern searching, find or find and replace multiple occurrences of a character string, advance characters to the beginning or to the end of a file or to insert or delete characters. Other character manipulation capabilities such as overtyping, copying from one position to another, and changing alphabetic character case can also be accomplished. Line manipulation features include: insertion and deletion of lines, advance lines, split one line into two and combine two lines into one. The operator can, at any time, use line-oriented commands or character-oriented commands.

During execution of the Advanced Editor, the user may define, view and invoke macros. The use of macros allows the user to define a sequence of Advanced Editor commands that may be called many times during an editing session. Additionally, the Advanced Editor can be invoked from a procedure file and process commands from a command file.

Terminal Selection

The Advanced Editor is configured to operate with the Tektronix CT8500 CRT Terminal. For optimum performance and software support under the license agreement, Tektronix recommends the CT8500. The Advanced Editor may be configured to operate with alternate CRT terminals. Depending on the terminal functions, come degradation may occur. While the Advanced Editor is designed to work with terminals of varying characteristics, Tektronix does not warrant and may not support the Advanced Editor with terminals other than the CT8500.

The Advanced Editor requires 64K bytes of program memory for correct operation. At no time does the size of the workspace affect the size of the file that can be edited.

Software Support Services

Assembler support software are in Software Category B which specifies the following support: During the 90 day period following shipment of SOFTWARE to the customer, if the customer encounters a problem with the SOFTWARE which his diagnosis indicates is caused by a SOFTWARE defect, the customer may submit a Software Performance Report (SPR) to Tektronix. Tektronix will respond to problems reported to SPRs which are caused by defects in the current unaltered release of the SOFTWARE via the Maintenance Periodical for the SOFTWARE, which reports code corrections, temporary corrections, generally useful emergency bypass and/or notice of the availability of corrected code. Software updates, if any, released by Tektronix during the 90 day period, will be provided to the customer on Tektronix' standard distribution media as specified in the applicable Data Sheet. The customer will be charged only for the media on which such updates are provided, unless otherwise stated in the applicable Data Sheet, at Tektronix' then current media prices.

Peripherals & Accessories

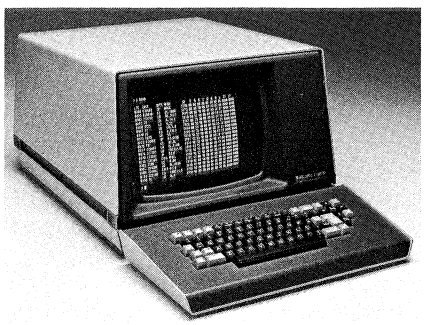
Peripherals
Accessories
Microcomputer Development
Lab Workshops

CT8500 Terminal

The CT8500 Terminal is an optional peripheral recommended for use with the Tektronix 8550/8560 Microcomputer Development Labs.

The CT8500 can display a full 25 lines at 80 characters per line on its 12-inch diagonal display screen. A complete set of upper- and lowercase ASCII characters is provided. The green-on-black display with adjustable brightness level is easy to read and can reduce eye fatigue in extended use.

The detachable keyboard for the CT8500 Terminal is arranged in an office typewriter configuration to aid new user familiarity. Eight programmable function keys can be user-defined. Up to a 64-character command or character string can be generated with a single key stroke. The two page memory of the CT8500 allows buffering and scrolling of up to 4,000 characters. Visual field attributes of blink, reduce, inverse, underline, and blank, as well as seven combinations of these attributes can be obtained. Alternative modes of operation allow the terminal to be used for local editing or to be controlled by a host computer using the many available remote commands. Other terminal operating modes are described in the following paragraphs. A list of remote commands is also included.



Terminal Modes

Local. Off line data entry with full editing capability. Block mode transmission by page or line.

Remote Power Up. Full duplex character-by-character transmission. Local functions active include: full editing capability, two page download memory with paging and scrolling.

Remote-Host Control. Full duplex character-by-character transmission. Remote Power Up local functions are disabled. Keys transmit a two byte code sequence to the host.

Monitor. Displays all transmitted characters, including control characters.

Test. Provides verification of the display unit capabilities.

Learn. Allows the terminal to learn the user definition of the eight programmable function keys.

Communications

The CT8500 Terminal can interface with the 8550/8560 Microcomputer Development Labs through an EIA standard RS-232-C interface. Selectable baud rates are: 110, 150, 300, 600, 1200, 2400, 4800, and 9600 externally switched. Communication between the terminal and associated Lab instrument is full duplex with parity enable-disable via 3 meter (approximately 10 foot) interface cable.

Display Parameters

Alphanumeric Mode Format: 25 lines at 80 characters per line

Memory: Two pages (50 lines × 80 characters per line = 4000 characters)

CRT Size: 304 mm (12 inch) diagonal

CRT Type: Non-glare (P31 phosphor)

Refresh Rate: 50 or 60 Hz, switch selectable

Visual Attributes: Blink; Reduce; Blank; Inverse; Underline; Blink and Underline; Blink and Reduce; Blink and Inverse; Reduce and Underline; Blink, Reduce and Underline; Inverse and Reduce; Blink, Inverse and Reduce

Electrical Characteristics

AC Power

Input Voltage Selection: 115 or 230 VAC switch selectable at 50 or 60 Hz.

Power Consumption: 65 Watts maximum.

Environmental Characteristics

Operating Temperature Range: 10° to 40°C (50° to 104°F)

Storage Temperature Range: −55° to 75°C (−67° to 167°F)

Operating Altitude Range: Sea level to 3 000 m (10,000 ft)

Storage Altitude Range: Sea level to 12 000 m (40,000 ft)

Operating Humidity: To 90% relative noncondensing

Physical Characteristics

Cabinet Dimensions:
Height = 318 mm (12.5 in)
Width = 456 mm (18 in)
Depth = 508 mm (20 in)

Keyboard Dimensions:
Height = 76 mm (3 in)
Width = 456 mm (18 in)
Depth = 222 mm (8.7 in)

Total Weight: 19.2 kg (42.5 lbs)

Keyboard Characteristics

Detached 79 key with auto repeat. Separate cursor positioning, programmable tab, backspace, eight programmable function keys, edit capability.

Remote Command Descriptions

1. Load or read cursor address
2. Start or stop visual attributes
3. Keyboard lock or unlock
4. Transmit page or line
5. Cursor on or off
6. Test CRT
7. Set horizontal split screen
8. Set or clear column tab
9. Insert or delete line or character
10. Clear screen or clear all pages
11. Erase to end of line or end of page
12. Home
13. Tab forward or reverse
14. Cursor up, down, right or left
15. Scroll up or down
16. Read top line address
17. Page 1 or page 2
18. Page down or up
19. Enter Local mode
20. Enter Power Up mode
21. Enter Host Control mode
22. Learn mode status
23. Enter and exit learn mode
24. Enter and exit monitor mode

4643 Printer

SPECIFICATIONS

Performance Characteristics

Printing Speed:
340 characters per second

Character Density:
Condensed: 218 characters/line
Standard: 132 characters/line
Expanded: 72 characters/line

Throughput Rate:
132 Columns Wide: 125 lines/minute
72 Columns Wide: 200 lines/minute
40 Columns Wide: 300 lines/minute

Paper Slew Rate:
254 mm/s minimum (10 in/s minimum)

Character Set:
128 (96 ASCII plus 32 commonly used international characters)

Vertical (Line) Spacing:
2.4 lines/cm

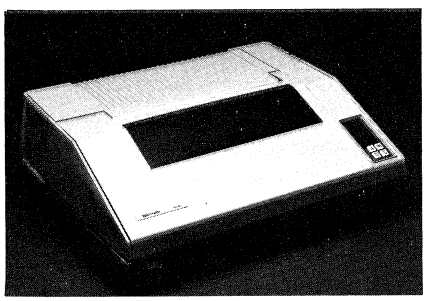
Horizontal (Standard Character) Spacing:
3.9 characters/cm
(10 or 5 characters/in)

Printing Matrix:
7-by-7 half-dot matrix

Paper:
Type: Continuous fan fold, edge perforated
Width: 76.2mm to 406.4mm (3 to 16 inches) at 0.7mm (0.028 inches) maximum thickness

Ribbon:
Type: Fabric, continuous loop, cassette
Life: 5 million characters nominal life

Power Requirements:
90 to 140 VAC or 187 to 257 VAC
50 to 60 Hertz; 250 Watts operating, 125 Watts in standby.



Physical Characteristics

Cabinet Dimensions:
Height: 203 mm (8 in)
Width: 676 mm (26.6 in)
Depth: 592 mm (23.3 in)

Cabinet Weight:
27.1 kg (60 lbs)

Ordering Information

4643 Printer
2400 Baud Standard

Standard Accessories

Ribbon Cassette	118-1314-00
Operators Manual	070-3871-00
RS232 Interface	

Optional Accessories

Service Manual	070-3870-00
Pedestal	118-1335-00
Paper Basket	118-1316-00

Options

Opt. 1 — Parallel interface
Opt. 2 — Specify Baud Rate
110, 150, 300, 600, 1200, 4800, 9600

Long, reliable service can be expected from the 14-wire matrix head component. The 4643 backs up the printing of each character with more head wires to assure an expected (head) life of more than 300 million characters with no maintenance. This figure normally means at least two full years of continuous work from a single matrix head. The fabric ribbons continuous loop cassette is usable for at least 5 million characters. Both the matrix head and ribbon cassette are quickly operator-replaceable, eliminating the need for a service call.

High quality matrix printing is assured by the unique 14-wire printing head. The 7 by 7 format print font permits easy reading and the operator can specify condensed, expanded or standard characters. In the condensed (character) face, the 4643 prints out a 132-character line format on an 8½ by 11 inch sheet.

Because the 4643 uses impact printing, six very legible copies (including five NCR or carbon copies) can be made to save time and avoid the expense of photo copies.

Make output simply with the easy-to-use 4643 Printer. The few operator controls needed are conveniently clustered at the front of the machine.

No attendance is necessary when the 4643 is running because the paper supply sensor automatically lets the machine print to the end of the last form. With either front, back or bottom loading ports, loading the machine is also an easy task. The operator may quickly position the paper to be loaded with the help of built-in horizontal and vertical alignment guides.

Compatibility. The printer of choice for high technology systems, the standard Tektronix 4643 is RS232 compatible and can be interfaced with most standard RS-232 data processing instruments and systems. Option 1 provides a parallel interface. The 4643 is compatible with the following Tektronix products: 4010 and 4110 Series Terminals, 4020 Series Terminals, and 4050 Series Desktop Graphic Computing Systems; the 8001, 8002A, 8540, 8550 and 8560 Microprocessor Labs; the S-3250, S-3270 and S-3280 Semiconductor Test Systems; the 7612D and 7912D Programmable Digitizers. grammable Digitizers.

A low cost alternative to conventional line printing, the Tektronix 4643 Printer provides fast, high-quality, impact printing that is suitable for most data processing applications. With high reliability built in, the 4643 is a convenient and economical choice requiring no preventive maintenance and infrequent servicing.

Fast but not expensive, the 4643 Printer uses bi-directional logic technology to print 340 characters per second. With a full 132 character line, speeds of 125 lines per minute are nominal. Even faster throughput rates are used for printing graphs and tables. The 4643 Printer rivals the speed of many line printers but is available for the modest price of a matrix printer.

Practical for many applications, this versatile printer can be used, for example, as a fast way to record the stages-in-process of software program development. It

can also provide the hard copy needed to analyze research and development. It can also provide the hard copy needed to analyze research and development data. Because the 4643 output is of report quality, you need only push a button to generate reports on your research, programming or statistics directly from your data base. For added convenience, you'll find the 4643 very useful in time consuming tasks such as label generation, previewing text and correcting the format of manuals or letters prior to typesetting.

Virtually no maintenance means an even greater savings, and less downtime for repairs as well. A diagnostic display and self-testing routine virtually eliminate the need for preventive maintenance calls. On the infrequent occasion that something goes wrong, you'll find out directly from the machine.

Both the matrix head and ribbon cassette are quickly operator-replaceable, eliminating the need for a service call.

PROM Programmer

An optional PROM programmer is available for use with the 8550 Microcomputer Development Lab and the 8540 Integration Unit.

The PROM programmer includes the general purpose controller board and a series of front panel plug-in modules. Each plug-in module provides support for a similar group of PROM devices, and can be changed in a few seconds. Programming features include reading, writing, and comparing (for verification) a PROM.

Chips currently supported include:

Intel	2716
	2732/32A
	2758/2758S1865
	2816
	8741A
	8748
	8749
	8755A
Motorola	MCM 68764
Texas Instruments	TMS 2508
	TMS 2516
	TMS 2532

Accessories

Flex discs (box of 10)	119-1182-01
RS-232 connecting cable (15 ft.)	012-0757-00
HSI connecting cable (8 ft./2.4m)	012-1009-00
HSI connecting cable (20 ft./6.1 m)	012-1008-00
HSI connecting cable (50 ft./15.2 m)	012-1007-00
HSI connecting cable (250 ft./76.2 m)	012-1010-00

MDL Workshops

Tektronix offers two Microcomputer Development Lab Workshops in a number of locations throughout the year. The courses are intensive, hands-on workshops designed to help the attendee meet the demanding challenges of the growing microcomputer development market.

The Development System in the Design Process Workshop

This workshop follows the microcomputer based design process from software development through hardware/software integration and PROM programming. The workshop emphasizes the overall operation of the 8550 development system, the DOS-50 operating system, and how the 8550 supports the design process.

Students will receive intensive, hands-on experience throughout the course for an in-depth understanding of all 8550 operations.

Topics covered include:

- DOS-50 in the design process
- 8550 Architecture
- DOS-50 Commands
- Volume Structures
- Program Design
- Editing
- Assembling
- Linking
- Library Generator
- Program Debugging
- I/O Simulation with Service Calls
- PROM Programmer
- Intersystem Communication

This intensive four-and-one-half day workshop will provide an educational experience for the user who has performed microprocessor-based software design at the assembly level with one or more microprocessors.

Prerequisites:

The attendee should meet one of the following general prerequisites:

- Have attended the Software Design and Debugging Workshop.
- Have used editors and assemblers or compilers on a minicomputer or mainframe computer.
- Have used a development system for microprocessors.

Software Design and Debugging Workshop

This workshop emphasizes the basics of 8-bit microprocessor software design. The workshop also covers the use of development systems for editing, assembling, debugging, hardware/software integration, and PROM programming.

The course initially provides a brief overview of the Z80 microprocessor's architecture, instruction set, and interrupt handling. The emphasis for the rest of the week will be on assembly-level software development and in learning and using the techniques for debugging code and debugging the execution of the software running on the hardware.

The course covers the following portions of the design process:

- Flowcharting
- Assembly-level Code Development
- Code Entry and Modification
- Assembling
- Software Debugging
- Real-Time Prototype Analysis
- PROM Programming

This intensive four-and-one-half day workshop will provide an educational experience for the person who has limited microprocessor-based software design experience and wants to learn about software design and the use of the development system in microcomputer designs.

For further information, contact:

U.S.A., Asia, Australia, Central & South America, Japan
Tektronix, Inc.
P.O. Box 4828
Portland, OR 97208
For additional literature, or the address and phone number of the Tektronix Sales Office nearest you, contact:
Phone: 800/547-1512
Oregon only 800/452-1877
Telex: 910-467-8708
Cable: TEKTRONIX

Europe, Africa, Middle East
Tektronix Europe B.V.
European Headquarters
Postbox 827
1180 AV Amstelveen
The Netherlands
Telex: 18312

Canada
Tektronix Canada Inc.
P.O. Box 6500
Barrie, Ontario L4M 4V3
Phone: 705/737-2700

Tektronix sales and service offices around the world:
Argentina, Australia, Austria, Belgium, Bolivia, Brazil, Canada, Chile, Colombia, Costa Rica, Denmark, East Africa, Ecuador, Egypt, El Salvador, Federal Republic of Germany, Finland, France, Greece, Hong Kong, Iceland, India, Indonesia, Ireland, Israel, Italy, Ivory Coast, Japan, Jordan, Korea, Kuwait, Lebanon, Malaysia, Mexico, Morocco, The Netherlands, New Zealand, Norway, Pakistan, Panama, Peru, Phillippines, Portugal, Republic of South Africa, Saudi Arabia, Singapore, Spain, Sri Lanka, Sudan, Surinam, Sweden, Switzerland, Syria, Taiwan, Thailand, Turkey, Tunisia, United Kingdom, Uruguay, Venezuela, Zambia, Zimbabwe.

Tektronix®
COMMITTED TO EXCELLENCE

61AX-4913 2/82