MACROTECH'S ADIT AN INTELLIGENT I/O BOARD

Connie Kelly
Macrotech International

Macrotech International's ADIT is an intelligent I/O board that will support up to 16 terminals, printers, modems, or other serial devices. A ***non***-intelligent I/O board requires the host CPU to spend an inordinate amount of time processing I/O on a character-by-character basis. In small systems with a minimal number of users, this is not a problem. However, as the number of users and peripherals increases, especially in an I/O-intensive situation, the host CPU spends so much time on I/O that response time becomes unacceptably slow. When using nonintelligent I/O, every character to be transferred interrupts the host, requiring the CPU's valuable time to save its current state and then process the character. The energies of the CPU are constantly diverted from application processing to handling this steady stream I/O.

The ADIT solves this I/O bottleneck with a 6-MHz Z80 CPU, full DMA capability, the on-board ADIT Operating System (AOS), and static memory. These features make it possible for the host CPU to offload a substantial amount of I/O processing onto the ADIT. The host CPU is not interrupted until the ADIT is finished with any assigned I/O tasks. This allows time-consuming I/O processing to take place in the background, without bogging down the host CPU.

ABOUT THE TECH FILE

The articles appearing on each issue in Tech File *are produced by companies about one of their own boards. They are not reviews.*

Every S-100 company is invited to submit articles. Please request our 'Tech File Guidelines' before writing an article. Tech File *articles are published one per issue on a first-come-first-served basis.*

BENCHMARKS ILLUSTRATE ADIT SPEED

Before covering the finer points of the ADIT, let's look at some benchmarks that demonstrate the ADIT's talent for increasing system throughput.

Interfacer 3 vs. ADIT. The first test was done in a CompuPro environment running MP/M-86. The system included a Macrotech 80286/Z80H CPU, 1 Megabyte of memory, System Support 1, Disk 1 controller, and either the Interfacer 3 or the ADIT-8. Response time was clocked by doing a barber pole (this is an I/O-intensive test that continually outputs characters to a terminal) to all eight ports. The ADIT outperformed the Interfacer 3 by a margin greater than 160 percent.

DR-200 vs. ADIT. The second test was performed by Barry Hamilton of the TWB Group, an Alpha Micro systems house in Illinois. The test was run under the AMOS operating system on an Alpha Micro computer, pitting the Macrotech ADIT-16 against the Dravac DR-200. Both are intelligent I/O boards. The DR-200 has an on-board 68000 CPU. Barry set up 16 ports all running barber poles at 19.2 Kbaud. Against this I/O-intensive backdrop, a simple-benchmark BASIC program was run. The BASIC program used a FOR-NEXT loop with 25000 iterations:

```
100 FOR I=1 to 25000
110 NEXT
```

The results were decisive. The total execution time for the DR-200 was ►

Connie Kelly is Director of Marketing at Macrotech International. She is also experienced in management and, of course, S-100s. Connie is a flying enthusiast and will soon be getting her pilot's licence.

INTERFACE REGISTERS	
AIC	Adit interface command register.
AIR1,2,3	Adit interface parameter registers.
ASEMA4	Adit semaphore flag register.
AIACK	Adit interrupt acknowledge register.

Table 1. *Interface registers used to pass commands to the ADIT I/O board.*

4 minutes and 42 seconds. The total time for the ADIT was only 34 seconds. As Barry Hamilton stated in his report, 'This is an apparent reduction in system overhead [...] of almost 9/1 in this configuration.

DMA AND ON-BOARD OS OFFLOAD HOST PROCESSING

The ADIT board implements DMA in accordance with IEEE-696 standards, using 24-bit addressing. The I/O channels can transfer data directly to and from memory without tying up the host CPU. When the ADIT board uses DMA, it controls the transaction by issuing strobes, control signals, and data to read or write in system memory. In a DMA transfer, the ADIT first requests the host CPU to relinquish control of the bus. When the host acknowledges the request, the I/O board places the desired address on the bus. When writing into system memory, it also places the data byte on the bus. The board returns control of the bus to the host after the transfer is completed.

Consider an output operation in which 32K bytes of data are sent to a printer. The host CPU would normally be involved in the output as each character was processed, pointing to each character in memory, reading it, and sending it to the printer. However, the ADIT makes this unnecessary. It accepts the output command from the host, and the address of the data. It then directly copies the information from system memory into its own memory while the host continues with other tasks, uninterrupted. After copying the information, the ADIT performs the full output to the printer. The entire output process is transparent to the host, with all interrupts generated by the ADIT's on-board USARTs and associated overhead being handled internally by the ADIT. There are no interrupts on the bus during the transfer. The ADIT interrupts the host only after the full output command has been completed.

ISSUING COMMANDS TO THE ADIT

The ADIT supports two types of commands. The lowest level of data transfer is performed by *immediate* commands. These commands manipulate I/O one character at a time. The majority, however, are *extended* commands. Extended commands use DMA to transfer character strings.

The host initiates I/O processing by issuing an immediate command through the port interface. These commands are defined in the I/O driver. In addition, the extended set of commands recognized by the ADIT's AOS gives the programmer great flexibility in configuring the I/O process for a given system.

Four 8-bit interface registers (Table 1) on the ADIT are used to pass immediate commands and responses between the host and AOS. A 'mailbox' scheme is used: the host deposits a command and parameters in these four registers, the mailbox. The AOS performs the command and outputs information and status back into the registers. The host need not

EXTENDED COMMAND CONTROL BLOCK	
ECR0	Byte count register (low byte).
ECR1	Byte count register (high byte).
ECR2	Source address (low byte).
ECR3	Source address (middle byte).
ECR4	Source address (high byte).
ECR5	Destination address (low byte).
ECR6	Destination address (middle byte).
ECR7	Destination address (high byte).
ECMD	Extended command.
ECRCL, ECRCM	Byte count registers, Count low, Count middle. 0 to 64K. DMA maximum is 64K.
ECRA0,1,2	LMH of data buffer; Address low-middle-high.
ECEOM	Must contain the EOM byte value for those specific commands that require an end-of-message condition test.
ECPARM	Parameter byte-command specific.
ECMASK1	Not yet used. Filled with OFFH.
ECMASK2	Not yet used. Filled with OFFH.
ECW0 - ECW15	Bit definitions of the sixteen 8530 write registers in the Serial Communications Controller Technical Reference Manual.

Table 2. *The Command Block resident in system memory and used to process extended commands. Registers ECR0 through ECR7 are for M-channel (memory-to-memory) data transfers.*

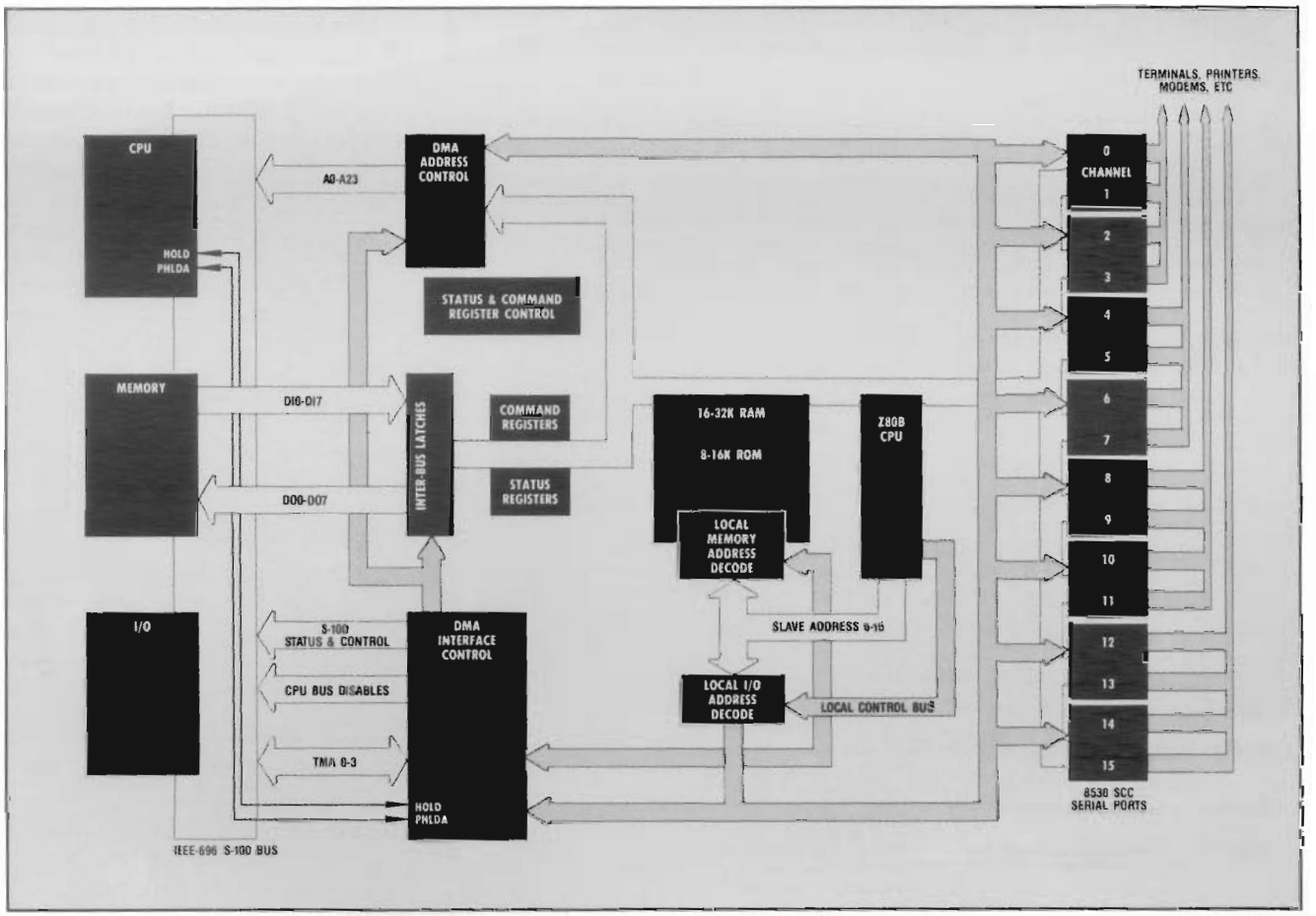


Figure 1. *Block diagram of the ADIT on-board components and their interface to the S-100 bus and peripherals.*

waste time waiting for the AOS's response. Instead, when the host later wants the status, it returns and checks the 'mailbox' registers.

Extended commands must pass more information than can be contained in the four interface registers. Therefore, they use an *Extended Command Control Block* (Table 2) set up in system memory to pass additional parameters and data. To initiate an extended command, the host issues an immediate command (AEXT) that includes a 24-bit address of the extended command block. For efficiency, the AOS interleaves commands (directed at any particular I/O channel) that might not be processed immediately with commands directed at other channels.

A special subset of the extended commands, M-channel commands, are used for memory-to-memory transfer of data and allow access to memory that is outside the host system address space. There is also a special subset of immediate commands, the V-channel commands, that are used to define and access a memory area as a virtual disk.

The ADIT board contains eight 8530 serial communications controllers with two channels on each controller. The extended command set includes the ability to issue primitive commands directly to the 8530. This gives the programmer complete freedom to use all capabilities of the 8530, such as establishing synchronous communication with mainframes, setting the number of stop bits per character, even or odd parity, number of bits per character transferred, sending a break, disabling of lower chain bit to prevent lower-priority devices on daisy chain from requesting interrupts, and the ability to write or read directly to any 8530 register.

A number of options, such as parity, data bits, baud rate, XON/XOFF support, are jumper selectable. Any of these jumpered defaults can be dynamically reconfigured by the host at any time. For example, the ADIT could be programmed to transmit at one baud rate and receive at another baud rate on any given channel.

Sample I/O drivers for MP/M-86, CP/M-2.2, and AMOS operating systems are included in the ADIT documentation. Additional support programs are provided for the AMOS operating system with each board purchased.

DYNAMIC ALLOCATION OF ON-BOARD MEMORY

The AOS, a multitasking, real-time operating system, resides in 8K of ROM. All 16 channels have access to 16K or 32K of RAM -- divided into discrete 32-byte packets. Each packet contains 4 bytes of chain-pointer information and 28 bytes of data storage space. If a message is longer than 28 characters, the AOS links additional packets.

ADIT IMMEDIATE COMMANDS

ASTAT	Request channel status	*Gives current channel status, including 8530 status bits.*
ARSET	I/O channel reset	*Recovers host control of I/O channel when interrupt service has been requested but has not occurred. Can also be used to abort current I/O or clear all conditions to prepare for reconfiguring channel.*
AERSET	I/O channel error reset	*If host drivers implement parity error detection, lost data recovery, or break monitoring, this command can be used to clear these latched status flags.*
ARECN	I/O channel reconfiguration	*Used to modify a channel's parameters*
ASDI	Single data character input	*Obtains a single input data character.*
AISDA	Interrupt, data available	*Notifies host that single character input data is available.*
ASDO	Single data output	*Transfers a single output data byte.*
AISDOB	Interrupt, output space	*Notifies the host when buffer space is available for single character output.*
ASPCL	Special conditions	*This command is a collection of six subcommands that allow special control and cancellation processes.*
AMEMTST	Nondestruct memory test	*Reports if any errors are found after one nondestructive pass of ADIT RAM.*
AMULTI	Multichannel input status	*Sets flag bits corresponding to each channel to indicate if input data was received and is in ADIT input buffer.*
AMULTO	Multichannel output status	*Flag bits set to indicate ADIT output buffer has space to accept additional output character(s).*
AFRSET	Full board reset	*Full reset of ADIT board without requiring full host reset.*
AEXT	Set up extended command	*Sets up registers with address of the Extended Command Block (ECB). The extended command to be performed is located in the ECB.*
AEXTI	Set up ext com w/ interrupt	*Same as AEXT, with interrupt on completion.*

ADIT M-CHANNEL AND V-CHANNEL COMMANDS

AMSTAT	M-channel request status	*Returns current status of the M-channel (used for memory-to-memory transfers).*
AMMVE	M-channel extended move	*For utility use by host operating system. Gives software access to memory that may not normally be within the host operating system address space.*
AMMVEI	M-chan ext mv w/ interrupt	*Same as AMMVE, except interrupt is generated when move is completed.*
AMSETA	M-channel address set	*Stores the M-channel (memory channel) control block address permanently, so that it is used by future M-channel commands.*
AVSTAT	V-channel request status	*Returns status for the virtual disk (vdisk) channel.*
AVSETM	V-channel address set	*Issues 24-bit address of the start of memory to be used for virtual disk operations.*
AVSETP	V-channel set parameters	*Sets parameters for vdisk operations, including size of vdisk area, sector size, and number of sectors per track.*
AVSETC	V-channel current sector	*Issued before a fill command to give desired track and sector number.*
AVSETD	V-channel set DMA addr	*Sets address of buffer area used to send or receive data from the vdisk memory area.*
AVFILL	V-channel fill vdisk	*Aids in formatting vdisk by filling the vdisk memory locations with the specified fill byte value.*
AVRD	V-channel read	*Transfers data from the vdisk to the host DMA buffer area.*
AVRDI	V-channel read w/ interrupt	*Same as AVRD, but with interrupt generated when read is complete.*
AVWR	V-channel write	*Transfers data from host DMA buffer area to vdisk.*
AVWRI	V-channel write w/ interrupt	*Same as AVWR, but interrupt is generated when write is complete.*

ADIT EXTENDED COMMANDS

SYNRCON	Channel reconfiguration	*Reinitializes channel, outputting values to 8530 registers.*
SETECBA	Set ECB address	*Permanently stores the Extended Command Block (ECB) address for future ECB commands.*
OUTWBC	Output with byte count	*Moves a message of specified length from output buffer to channel via DMA.*
OUTWEOM	Output with EOM	*Moves a message terminated by a specified EOM (end of message) from output buffer to channel via DMA.*
INWBC	Input with byte count	*Stores input characters in the host input buffer, via DMA, until the total number stored equals a specified byte ct.*
INWEOM	Input with EOM	*Same as INWBC, except input is terminated by EOM of specified value, rather than byte count.*

Table 3. *ADIT commands. Execution of immediate commands requires a command name and parameters to be placed in the ADIT registers. Extended commands require the transfer of more information than can be held in the ADIT registers, so the registers will contain only the location of the Extended Command Block (ECB) resident in system memory. The ECB then contains the command to be executed and information for its execution. The M-channel subset of commands allows memory-to-memory tranfer of data, and V-channel commands are used to define and access a virtual disk set in memory.*

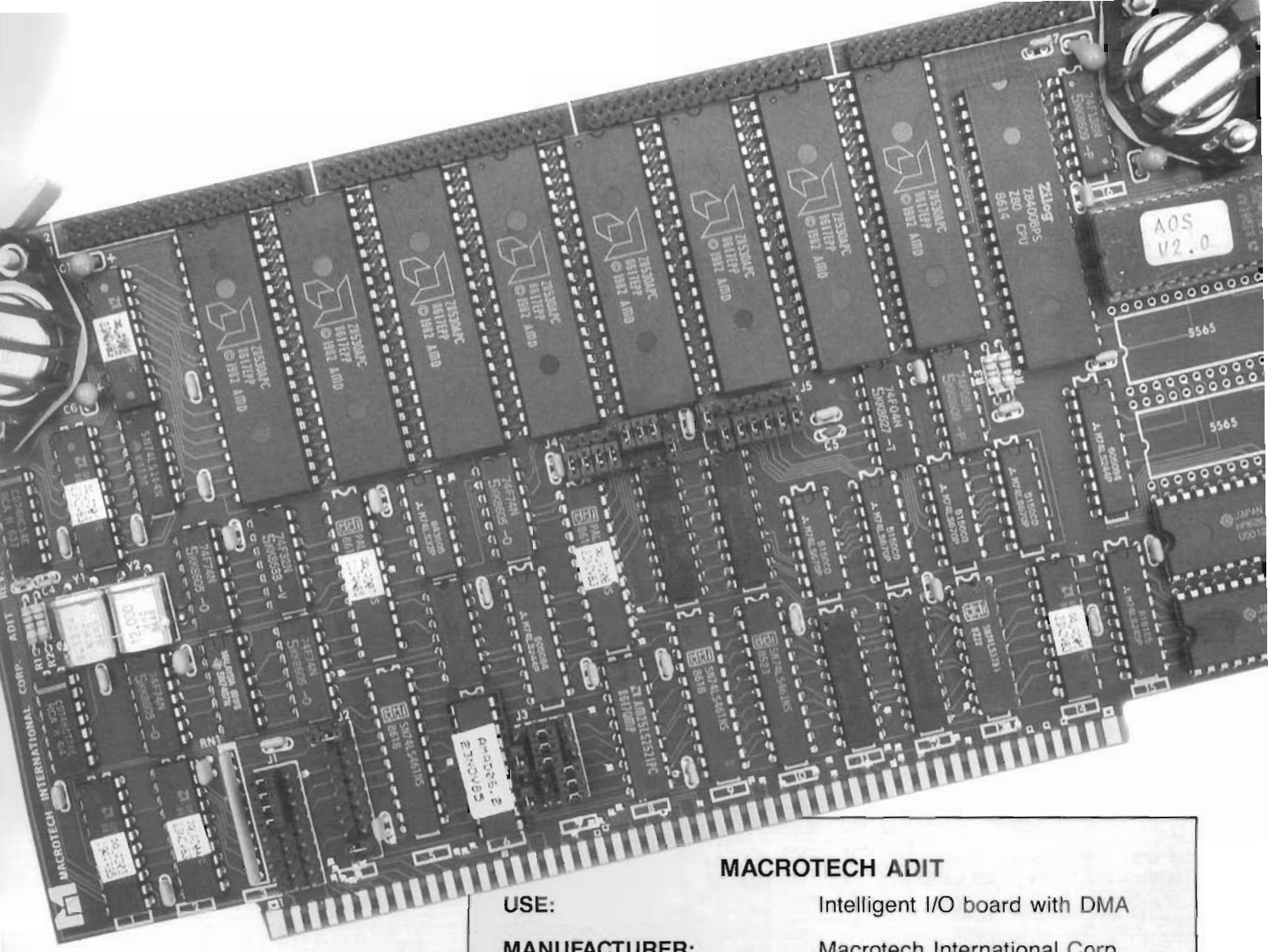
After the board transfers a group of 28 characters over a channel, it returns the empty packet to a pool of available packets. Each channel needing memory space obtains these packets from a pool. At any given instant, one of the 16 channels might not need memory or might need thousands of bytes of memory. The AOS dynamically allocates memory to the channels. This process is transparent to the host and to the user.

The ADIT board uses closed-loop feedback techniques and dynamic allocation of on-board memory to maintain control during heavy usage. When the board is handling communications on multiple channels, some channels will be doing output from the host while others will be active on input. If the capacity is nearing saturation, the AOS slows down output-character handling by allocating less of the available memory to the output channels while preventing data loss. However, the system must process input data as it occurs to prevent data loss or overrun conditions. The AOS maintains sufficient memory packets for input use so that input is not slowed and data is not lost. The AOS dynamically adjusts memory allocation as needed in the real-time environment to prevent data loss.

The AOS command set includes a nondestructive memory test for the ADIT RAM. This command will return a zero in two of the interface registers if no compare errors in its random memory test are found. The memory test may be run while all channels and all other commands are active.

PALS AND SIX-LAYER ARCHITECTURE CONCENTRATE LOGIC

The ADIT is a six-layer printed-circuit board. Programmed-array-logic (PAL) integrated circuits contribute to logic concentration. All decoding and creation of output terms and subterms can be implemented on PALs, eliminating the need for separate chips. The use of static RAM further reduces the



chip-count of the board by eliminating the need for additional refresh circuitry.

The Z80B-based ADIT contains eight 8530 controllers with two channels per controller. Each pair of 8530s connects to a single 34-pin ribbon connector that has four paddle cards, each providing the cable connection for a single channel. All channels support asynchronous I/O. In addition, four channels support synchronous communications, with the software for synchronous I/O to be supplied by the user. In a fully implemented ADIT board, there are four ribbons with four channels each, providing a total of 16 channels.

MACROTECH ADIT

USE:	Intelligent I/O board with DMA
MANUFACTURER:	Macrotech International Corp. 21018 Osborne Street, Bldng 5 Canoga Park, CA 91304 (818)-700-1501
COMMAND PORTS:	Five ports Address width: 8 bits Address select: 8-byte boundaries Data width: 8 bits
DIRECT MEMORY ACCESS:	Address width: 24 bits Data width: 8 bits
ON-BOARD PROCESSOR:	6-MHz Z80B, no wait states
ON-BOARD MEMORY:	ROM: 150ns, 8K or 16K RAM: 120ns, 16K or 32K, static
SERIAL CONTROLLERS:	Eight 6-MHz 8530 chips
SERIAL PORTS:	4, 8, 12, or 16. Two per controller. Synch or Async Transmission Flexible baud-rate generation On-board handshaking control
MANUAL:	104-page technical manual with schematics
RETAIL PRICE:	$985 to $1,795 (4 to 16 ports)