Attache 8:16 Technical Manual Supplement

for Attache 8:16 and Attache 8:16S

Portable Computer Systems

This supplement describes the Attache 8:16 8086 16-bit processor board and is to be used in conjunction with the Attache Technical Manual (92-051202).

Published by Otrona Advanced Systems Corp.

Otrona Pub. No. 92051242 Version 1.0 December, 1983

Contents

3. Software (Continued)

4. Appendixes

Introduction

Overview

The Attache 8:16 consists of the Attache modules (the main processor board, the display screen, the keyboard, the diskette drives, and the power supply) plus one additional module, the 8086 16-bit Processor Board.

The overall features of the 8086 16-bit board are described in this chapter. Attache modules that are enhanced in Attache 8:16 are also described in this chapter. These modules include the keyboard, the display, the Z-80A main processor board, and the diskette drives.

The 8086 16-bit Board

The Attache 8:16 contains three separate memories: a 256K byte RAM for the 8086 (16-bit) processor's programs and data, a 64K byte RAM for the Z-80A (8-bit) processor's programs, and a 28K byte RAM for display. The 64K under Z80 control provides I/O buffering for the 8086.

ROM (Read-Only Memory) on the Z-80A processor board includes 4K bytes that are mapped in and out of the lower section of processor memory space via software, providing automatic startup diagnostics and machine initialization, subassembly tests for servicing and performance verification, CRT emulation (VT-52 and ADM-3A), and general purpose subroutines for reading from and writing to disks, displaying characters, and accessing the printer and communications ports.

The contents of two 4K byte ROM chips reside at the highest memory addresses of the 8086 processor board. The ROM contains device initialization and diagnostic routines.

Processors

The dual Central Processing Units (CPU's) are an Intel 8086 processor operating at 8MHz (interfacing with the 256K bytes RAM memory) and a Zilog Z-80A main processor operating at 4MHz (interfacing with the 64K bytes RAM memory).

The 8086 is a true 16-bit processor in that both its logic and data paths are a full 16 bits.

A separate 9517A direct memory access controller controls data transfers between Z80 memory and the Input/Output devices (the disk system, communications and printer ports, etc).

When the system is started by the user, the Z80 tests to see which operating system has been selected. If MS/DOS is loaded, the keyboard, graphics, disk format, and I/O are set to the appropriate configuration and control is passed to the 8086. If CP/M is loaded, that configuration is selected and the Z80 retains control.

Input/Output

The Attache 8:16 comes standard with two asynchronous communications ports and provision for an optional synchronous communication port and an optional GPIB (IEE-488) port. One asynchronous port is configured as an RS-232 printer interface and the other port is configured as an RS-232 data communications interface. Either of these ports may be easily reconfigured to support RS-422 and/or RS-423 line and protocol standards.

Asynchronous transmission rates of 19200, 9600, 4800, 2400, 1200, 600, 300, 150, 134.5, 110 or 75 baud may be selected independently for each port via keyboard or software control.

Attache interface cables (available from your dealer) allow direct connection to standard serial devices, local peripheral devices, and communications devices such as modems. In all local cable uses, Attache 8:16 appears as a Data Communications Equipment (DCE) device and the peripheral appears as a Data Terminal Equipment (DTE) device. For communications, Attache 8:16 appears as the DTE and the peripheral appears as the DCE.

Optional I/O available with Attache 8:16 includes a GPIB (General Purpose Interface Bus) controller and a serial synchronous port. The GPIB controller allows you to interface with the DA-10 Hard Disk Subsystem, a 10 Megabyte hard disk.

Display Screen

Display typeface is a 6 x 7 character cell in an 8 x 10 block (with descenders). Software-selectable formats include 24 lines of 80 characters and 24 lines of 40 (double size) characters for CP/M and 25 lines of 80 characters or 40 double-size characters for MS-DOS.

The standard character set contains 512 character types, including the full ASCII character set (upper and lower case), complete IBM-PC character set and symbols, special word processing and journalism symbols, Greek alphabet characters (upper and lower case), mathematical symbols, business form and graph drawing symbols, and symbols to complete German, Spanish, French, Italian, Swedish, and other alphabets.

Graphics Mode

The Attache 8:16 provides both medium- and high-resolution bit-mapped graphics capabilities. When operated as an 8-bit computer under CP/M, the system defaults to the Attache standard of 320 X 240 pixels, organized as 24 lines of 80 characters.

When operated as a 16-bit computer under MS/DOS, the system selects either high-resolution mode of 640 X 200 pixels or medium-resolution of 320 X 200 pixels. Both display modes are completely compatible with the IBM-PC display.

A character ROM contains both the complete Attache character set and the complete IBM character set. The appropriate set is automatically selected when the operating system is loaded.

Keyboard

The keyboard contains a standard IBM Selectric™ style arrangement augmented with cursor direction, delete, and other keys required to implement the full set of ASCII characters.

Several keyboard modes are supported. With MS-DOS, the Attache 8:16 keyboard can be used to emulate IBM-PC keys and functions. With either CP/M or MS-DOS, Valet Set-Up Mode can be used for accessing Valet programs (screen dump, alarms, and calculator) and controlling the operating environment (screen brightness, keyboard sound and volume, printer and communications port baud rates, etc.). Other keyboard modes include 10-Key Mode for simulated 10-Key keyboard numerics and WordStar Mode for dedicated word processing functions.

Disk System

When operated as an 8-bit computer, diskettes are written and read in the Attache standard format, which is 360K bytes per diskette (after CP/M) on diskettes that are soft sectored, (2D) double density and double sided, written 512 bytes per sector, 10 sectors per track, 38 tracks per side, and 48 tracks per inch (TPI). Disk transfers are handled by the direct memory access controller, freeing the main processor for other tasks.

When operated as a 16-bit computer, diskettes may be read or written in all IBM PC-DOS formats, including 8-sector DOS 1.x and 9-sector DOS 2.0 formats with both single/double-sided and single/dual-density media. Maximum capacity with DOS 2.0 9-sector dual-sided, dual-density format is 360K bytes per diskette.

Theory of Operations

Overview

This chapter describes the 8086 16-bit processor board. The 8086 16-bit board fits into the expansion board connector on Attache's main processor board, providing access to the system bus.

The following section describes the overall Attache 8:16 system flow. Subsequent sections describe in detail the individual logic sections of the 8086 16-bit board, which include: the 8086 CPU (Central Processing Unit) and associated control logic, 256K bytes of RAM (Random Access Memory), interface circuitry to the Z-80A main processor board, and optional circuitry. Optional circuitry includes the GPIB controller (General Purpose Interface Bus), the serial synchronous port, and the 8087 numeric coprocessor.

Hardware modifications upon Attache to construct Attache 8:16 include a secondary board connected to the 8086 16-bit board, a high-resolution graphics board fit onto the Z-80A main processor board, and alterations to the display circuitry on the main processor board. These modifications are described in the final sections of this chapter.

Schematics of the 8086 16-bit board, secondary board, graphics modification, and display modification are located in Appendix A. Specific chips are identified in this chapter by their schematic location number in parenthesis, as well as by their manufacturer's part number. Note that the manufacturers identified in this manual are for the primary chip source.

Logic Overview

The logic sections on the 8086 16-bit board are the processor, the RAM (Random Access Memory), the Z-80A interface, and circuitry for optional features.

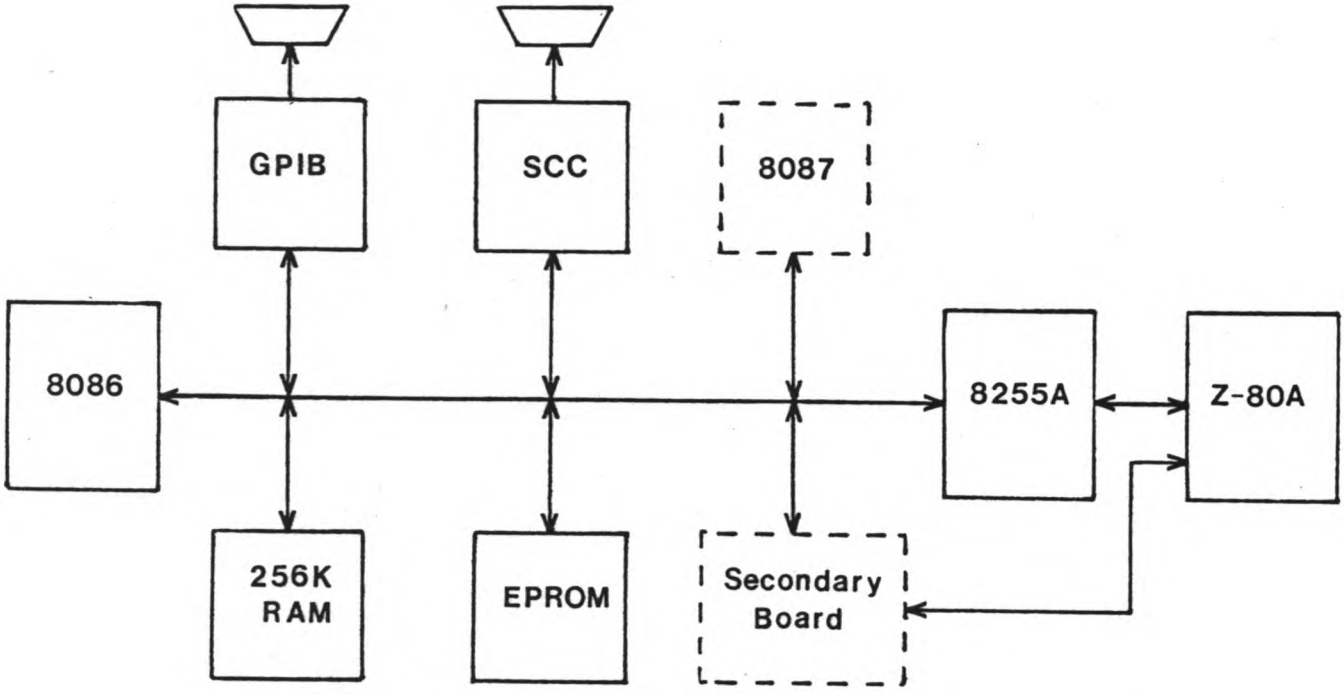
The CPU circuitry consists of an 8086, a clock generator, a wait state generator, and a processor command decoder. The 8086 processor is a true 16-bit processor with 20 address lines, capable of addressing up to 1 Megabyte of RAM.

The 8086 16-bit board contains 256K bytes of dynamic RAM, a capacitive method of storage which requires periodic refreshes.

A socket is located in parallel with the 8086 processor. This socket can contain either an 8087 numeric coprocessor or a secondary board. The secondary board issues hardware interrupts that allow Attache 8:16 to emulate IBM-PC hardware.

All interface between the Z-80A and the 8086 is handled through an 8255A interface chip. The Z-80A operates basically as an Input/Output handler, with the capability to interrupt the 8086. The 8086 cannot interrupt the Z-80A. The Z-80A/8086 interface is described in detail in the following section.

In addition to the 8087, optional features of the 8086 16-bit board include a synchronous serial port, and a GPIB (General Purpose Interface Bus) controller. These options are described in following sections.



System Block Diagram

Z-80A/8086 Interface

The Z-80A handles all low-level I/O in the system: character input and output buffering, disk reads and writes, and console (display and keyboard) activity. The 8086 BIOS (Basic Input Output System) converts DOS requests into requests to the Z-80A. The 8086 does very little I/O processing itself.

All data transfers are either data reads from the I/O devices on the Main Processor Board (via the Z-80A) or data writes from the 8086 to the Z-80A I/O devices.

Where possible, write commands (from 8086 to Z-80A) take advantage of the dual processor system. The 8086 does not wait for completion status of a given command, allowing simultaneous I/O processing. Read commands (from Z-80A to 8086) require the 8086 to wait for completion.

The two processors communicate through a single 8-bit parallel port. Since all transfers (commands, status, and data) flow through this port, a master/slave communication protocol is required. The 8086 acts as the master; all actions are the result of 8086 commands. The Z-80A receives commands, parameters and data, and returns data and status.

Interface with the Attache Z-80A bus is handled by an Intel 8255A Programmable Peripheral Interface chip. The 8255A has 24 I/O pins, which are programmed as two groups of 12 pins. One group operates in MODE 2 (a bidirectional mode using eight lines for a bus and five lines, borrowing one from the other group, for handshaking), and the other group operates in MODE 0 (which is programmed for output).

The Z-80A can interrupt the 8086, but the 8086 cannot interrupt the Z-80A. (The secondary board can interrupt the Z-80A, which places the 8086 into a wait state.) When the 8086 requires data from the Z-80A interface, it pulls signal INTREQ true. The Z-80A detects this flag, retrieves the requested information, and passes it to the 8086 via the 8255A.

The Z-80A is constantly operating as an I/O handler, handling all of the Z-80A board devices and awaiting INTREQ from the 8086.

Port assignments are as follows:

100	Port A - Read/Write
102	Port B - Read/Write
104	Port C - Read/Write
106	Control Port - Write only

Port A is used for bidirectional interface with the Z-80A bus.

Port B Bits 0 - 3 are used by the interrupt controller logic to mask out interrupts. Bit 4 enables WAIT logic, and Bit 5 is connected to J4 to enable or disable high-resolution graphics (1 = medium resolution, 0 = high resolution).

Port C Bits 3-7 contain PORT A status information as follows:

Bit 3	Interrupt request to 8086 if ON
Bit 4	Interrupt enable associated with IBF
Bit 5	IBF (Input Buffer Full) if ON
Bit 6	Interrupt enable associated with OBF/
Bit 7	OBF/ (Output Buffer Full) if low
Bit 0	DSR/ (Data Set Ready) from serial interface

Port C Bits 4 and 6 can be set or reset by writing to the control port as follows:

109H	Set IBF interrupt enable
108H	Clear IBF interrupt enable
10DH	Set OBF interrupt enable
10CH	Clear OBF interrupt enable

0C1H is written to the control port to initialize the 8255A with Port A in MODE 2 and Port B in MODE 0 (output).

The 8255A can interrupt the 8086 at interrupt level 3.

Z-80A Ports

The Z-80A communicates with Port A of the 8255A as follows:

Z-80A Port	Read	Write
B8	Data	Data
B9	Status	Control

Status Bits:

Bit 0	1 = no data ready 0 = data ready for read
Bit 1	1 = ready to accept data 0 = not ready to accept data

Control Bits:

Bit 0	1 = Reset 8086 0 = Set 8086 to run state

The Z-80A/8086 interface which occurs during the system boot and during specific commands is described in Chapter 3.

8086 CPU

The CPU circuitry consists of an 8086 processor (U5), a clock generator (U11), a wait state generator (U27), and a processor command decoder (U28). Additionally, chip location U10's signal lines are in parallel with the 8086.

The secondary board connects to the 8086 16-bit board at socket U10. U10 may optionally be used to attach an 8087 numeric coprocessor instead of the secondary board. This option is called an Attache 8:16A, and features an 8087 coprocessor, but does not feature IBM compatibility. The 8087 is described in a following section in this chapter.

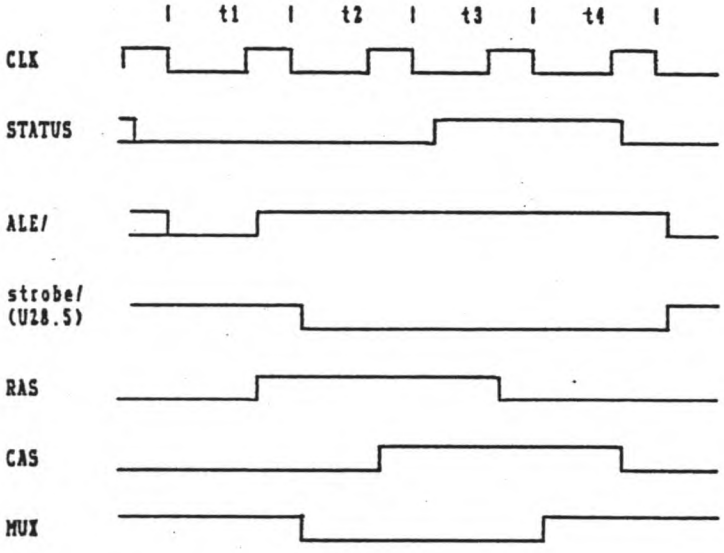
The 8086 (U5) is operated in maximum mode. The clock generator is an 8284A (U11) which derives its signal from a 24MHz crystal (X1) to provide an 8MHz, 1/3 duty cycle clock. Signal line /CLK is an inverted output from the 8284A, and DCLK is a twice-inverted delayed clock.

The 8284A (U11) also provides READY line synchronization. RAM cycles do not use wait states, except as provided by RAS/CAS circuitry during refreshes. For non-RAM cycles, wait states are provided according to the setting of jumper L on the output of flip-flop U27. In Attache 8:16, this jumper is connected to J to provide two wait states. When this jumper is connected from L to K, U27 generates a single wait state, L to J generates two wait states, and L to I generates three wait states.

Note that no bus controller chip is used. Instead, Address Latch Enable (ALE) is simulated by the flip-flop circuitry U62, U29, and U27. When ALE is high, addresses and status lines are allowed to settle. At the middle of T1, address and status lines have settled and are valid. The rising edge of the clock then sets ALE low, and address lines are latched by U16, U17 and U18, and the status lines are latched by an LS137 decoder (U28).

The LS137 decoder (U28) also decodes the status lines and provides strobed control lines which resemble 8288 bus controller lines. Control lines are enabled by U28 Pin 5, valid at the start of T2. For non-RAM cycles, U27 provides two wait states, as determined by the setting of jumper L.

Note that U28 is a socket which contains a header connecting to the secondary board. The LS137 decoder is actually U9 on the secondary board.



RAM Cycle

Dynamic RAM

The 64K x 1 dynamic RAM (Random Access Memory) chips comprise a 256K byte memory array on the 8086 16-bit board. Thirty-two chips (U34-41, U44-51, U54-61, U64-71) are organized into two 128K byte banks, addressed from 0 to 128K bytes and 128 to 256K bytes.

RAM address multiplexers (U25 and U26) multiplex 16 latched address lines (A1 - A16) into a row address and a column address. Gates U53 and U30 ensure that a RAM cycle is initiated on the rising edge of the clock during T1 when A18 and A19 are low (address 0 to 256K), and when a memory cycle has been indicated on 8086 status lines S0, S1, and S2.

Memory cycle timing is controlled by flip-flops U43 and U62. RAS (Row Address Strobe) is active for two clock cycles. CAS (Column Address Strobe) is also active for two cycles, but follows RAS by one cycle via the feed-back delay caused by U53 and U43.

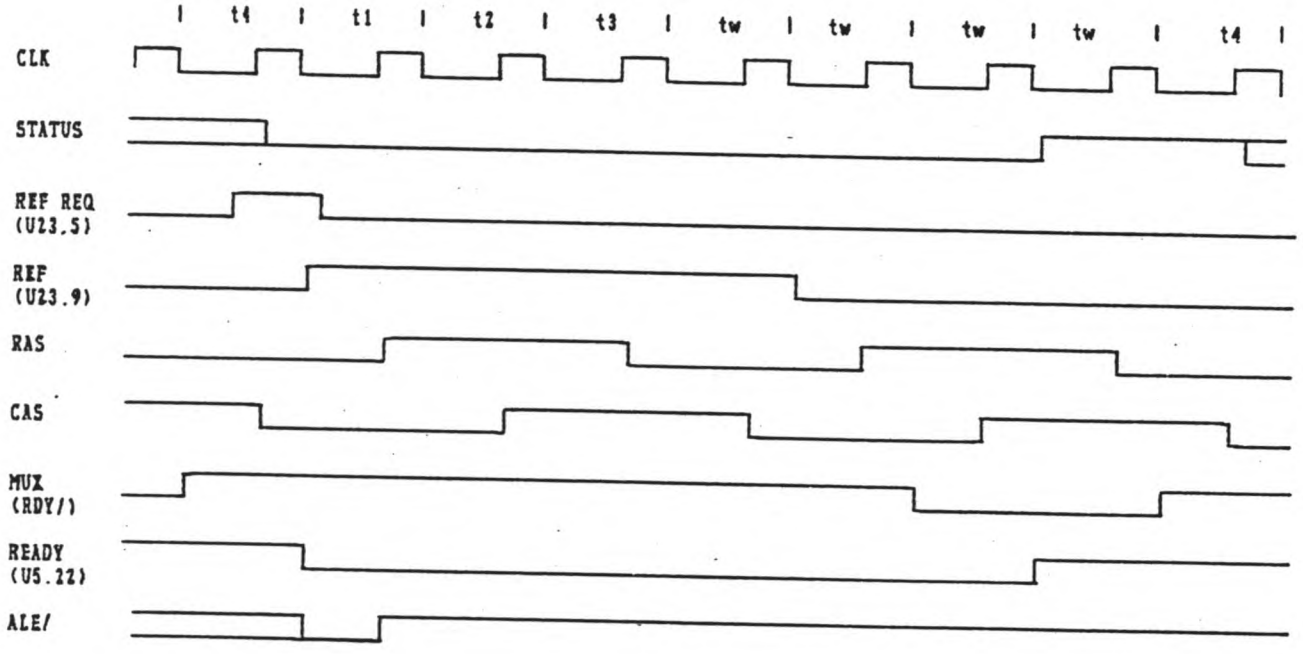
Signal line MUX toggles on the falling edge of RAS and CAS becoming active, which selects column address rather than row address from the U25 and U26 outputs.

Depending on the state of A17 (input to decoder U42), either signal RAS0 (driving the first 128K) or signal RAS1 (driving the second 128K) is generated by decoder U42 and gate U52. The 8086 can perform operations on a word (A0 and /BHE both low), on the high-byte only (A0 high /BHE low) or on the low-byte only (A0 low, /BHE high). OR gates in chip U29 control CAS for high-byte, low-byte or word operations, as required.

Counter (U14) divides the 8MHz system clock by 128, to provide a refresh request every 16 microseconds. A refresh is granted via flip-flop U23 Pin 9 when RAS and CAS are idle, which must occur within four clock cycles of a refresh request.

During refreshes, U25 and U26, the RAM address multiplexers, are disabled, and buffer (U24) is enabled. This gates the current refresh address from U15, an 8-bit counter which advances with every refresh request, onto the RAM address lines. RAS and CAS then perform a normal memory cycle sequence, except that signal line MUX remains high and CAS is blocked by NAND gate U72 so that CAS cannot reach the RAM.

During refresh cycles, memory cycles are held, which places the 8086 into a wait state. At the conclusion of the refresh cycle (when RAS and CAS are again inactive), flip-flop U23's output line REF is cleared. This allows any pending memory cycle to proceed, and consequently releases the 8086 wait state.



Refresh Cycle
(with delayed memory cycle)

Interrupt Structure

The Z-80A can interrupt the 8086, but the 8086 cannot interrupt the Z-80A. The secondary board interrupts the Z-80A and places the 8086 into a wait state during read/write operations to unique IBM-PC addresses. SINTR allows the Z-80A to interrupt the 8086 during timer tick interrupts (INT 1CH).

There are four signal lines capable of initiating on-board interrupts, depending on optional features which your 8086 16-bit board may contain. These interrupt lines are: the 8255A (PIO), the 8087 numeric processor extension (NPX), the Z8530 (SCC), and the TMS-9914A (GPIB). **Note:** When the secondary board is installed, PIO is driven by line SINTR (the interrupt signal from the Z-80A), and NPX is not applicable, as there is no numeric processor.

These interrupts can be masked by Port B Bits 0 - 3 of the 8255 as follows:

Bit 0	PIO
Bit 1	GPIB
Bit 2	SCC
Bit 3	NPX

An interrupt is allowed when the corresponding bit is set to one, and masked out if the bit is set to zero. Priority among any non-masked interrupt is determined by software. Fifteen interrupt vector locations (F0 - FE) are used. Each combination of interrupting device vectors form a unique location as follows:

With Secondary Board	
F1	SCC GPIB SINTR
F3	GPIB SINTR
F5	SCC SINTR
F7	SINTR
F9	SCC GPIB
FB	GPIB
FD	SCC

With 8087	
F0	NPX SCC GPIB PIO
F1	SCC GPIB PIO
F2	NPX GPIB PIO
F3	GPIB PIO
F4	NPX SCC PIO
F5	SCC PIO
F6	NPX PIO
F7	PIO
F8	NPX SCC GPIB
F9	SCC GPIB
FA	NPX GPIB
FB	GPIB
FC	NPX SCC
FD	SCC
FE	NPX

Interrupt priority is then determined by which of the four routines each of these fifteen vectors indicate.

Interrupts are discussed in Chapter 3 of this manual.

TEST Circuit

The 8086 /TEST input, together with the WAIT instruction, can be used for synchronizing I/O to the NPX, SCC, GPIB, or PIO. If Bit 4 of the 8255 PORT B is set to one, the /TEST input is connected to signal line NPX BUSY, (Which NPX software expects). To use /TEST with other devices, bit 4 is set to zero, which essentially connects test to an interrupt request. By disabling 8086 interrupts and masking off all but the desired device, very fast transfers between the CPU and the SCC, GPIB, or PIO, can occur.

Optional Feature - Serial Port

The 8086 16-bit board has provisions for an optional serial port, which uses a Z8530 SCC (Synchronous Communications Controller) chip. The following ports are used:

144H	A	Control and Status
146H	A	Data
140H	B	Control and Status
142H	B	Data

RTxCA is connected to Pin 15 (transmit clock).

TRxCA is connected to Pin 17 (receive clock).

Only the A channel of the SCC chip is connected.

PCLK is driven at 4HMz. The SCC's baud rate generator divides the 4MHz clock to provide a 9600 baud rate.

Optional Feature - GPIB

A TMS9914A provides an optional GPIB (General Purpose Interface Bus) port. This GPIB port can be used to interface Attache 8:16 with the DA-10 Hard Disk Subsystem, a 10 Megabyte hard disk.

The following addresses are used as the TMS9914A registers:

Address (hex)	Read	Write
180	Int Status 0	Int mask 0
182	Int Status 1	Int mask 1
184	Address Status	
186	Bus Status	Auxiliary Command
188	Address Switch 1	Address Register
18A		Serial Poll
18C	Cmnd Pass Thru	Parallel Poll
18E	Data In	Data Out

Optional Feature - 8087

The 8087 Numeric Data Processor is a coprocessor that adds extensive high-speed numeric processing capabilities to the 8086. The 8087 performs arithmetic and logical operations on a variety of numeric data types.

The 8087 cannot coexist with the secondary board. (the board which fits onto the 8086 16-bit board and which makes the Attache 8:16 IBM compatible). The 8087 is an option available on the 8086 16-bit Processor Board instead of the IBM compatibility provided by the secondary board. A computer which contains this option is called an Attache 8:16A, and must be specially ordered from Otrona.

The Secondary Board - Overview

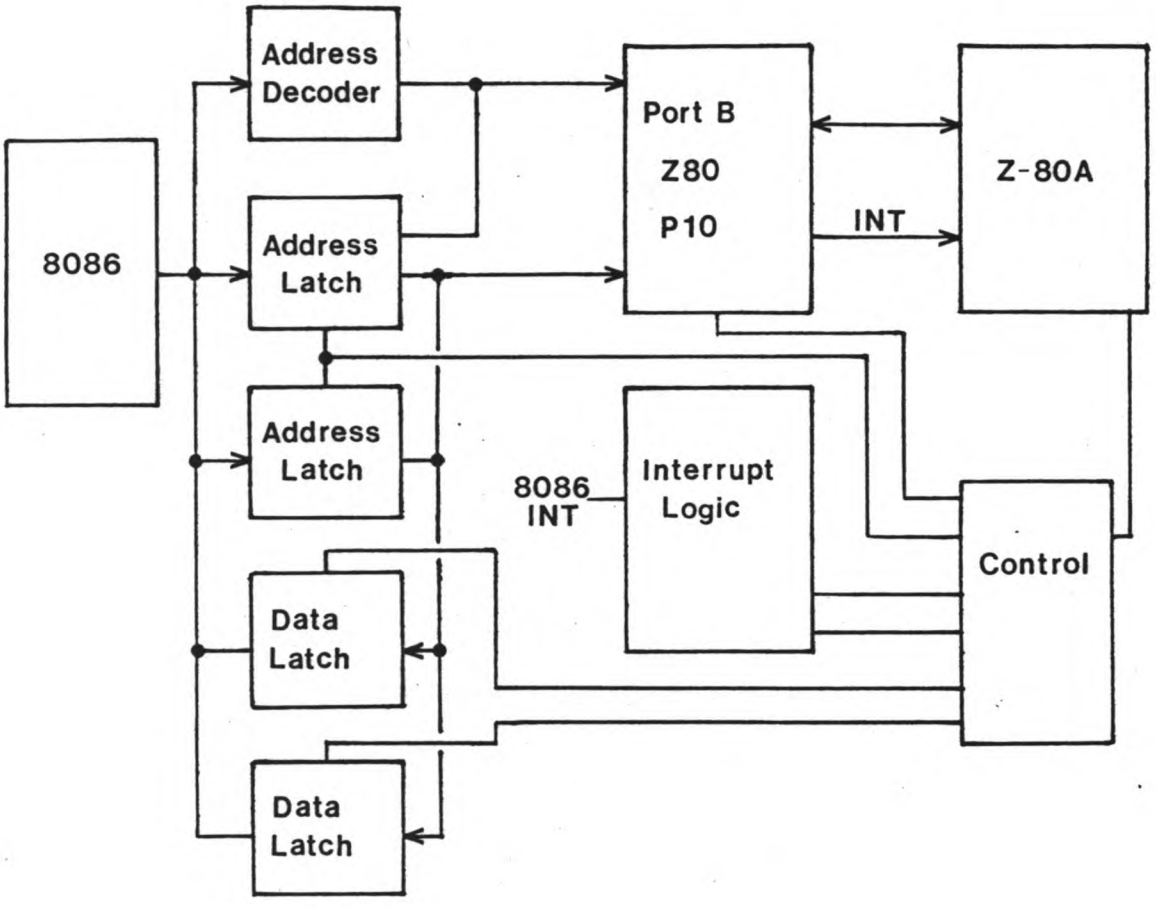
The secondary board on the 8086 16-bit board is designed to allow hardware which is not IBM compatible to be IBM compatible. The secondary board intercepts any 8086 requests that can be unique to IBM-PC hardware, and simulates this IBM-PC hardware.

To accomplish this PC simulation, the Z-80A is interrupted during 8086 I/O reads and writes **outside** the address range 0100-01FFH, and during 8086 memory reads and writes **inside** the range B0000-BFFFFH. When the Z-80A is interrupted, the 8086 is set into a wait state (via signal SRDY). The Z-80A interrupt service routine processes the respective read and write, simulating the IBM hardware, and then releases the 8086 from its wait state.

The process of interrupting the Z-80A during IBM-PC unique address calls is referred to as "trapping". The IBM-PC uses memory addresses B0000-BFFFFH for display memory and the 8086 16-bit board does not, so memory reads/writes in that range are trapped. The 8086 16-bit board uses I/O addresses in the range 0100-01FFH, which the IBM-PC does not, so all I/O read/writes except those in the identified range are trapped.

Initialization

Both ports of the Z-80A PIO (U8) operate in MODE 3, i.e., bidirectional bit-mode, and interrupts are enabled. (Port A is actually used as a 1-byte, 8-bit, bidirectional internal bus, but the port is initialized in bit mode to simplify the interrupt schemes.) Port B has two outputs, Bits 3 and 4, which are used to enable latch outputs onto the Port A bus. Bits 5 through 7 are inputs, used as the sources of interrupts. Bits 0 through 2 of Port B are inputs which indicate to the Z-80A how to process the interrupt.



Secondary Board Block Diagram

Address Decoding

As the 8086 executes code, it places addresses on lines AD0 through AD19 and strobes these addresses with ALE (Address Latch Enable, U9 Pin 4). The high- and low-order addresses are latched by U16 and U13 when ALE goes low.

The 8086 then puts data onto the AD0-AD15 lines, and continues operating unless the address is within the ranges that must be trapped. If the address must be trapped, the 8086 is placed into a wait state. The 8086 data is held on the bus for a write operation, or the 8086 waits to read data for a read operation. Note that regardless of whether or not the operation is trapped, U13 and U16 will have latched the address before any wait state occurs.

Two sets of address decoding logic are provided on the secondary board, each to detect the occurrence of one of two trapping conditions (either an 8086 memory read/write memory operation within addresses B0000H-BFFFFH, or an 8086 I/O read/write outside addresses 0100-01FFH).

The first set of address decoding logic detects memory addresses B0000-BFFFF {hex}. This logic consists of an LS20 (U6), two inverters (in U10 and U5), and an LS374 (U12). If the address output by the 8086 is in the range B0000H-BFFFFH, the output of the first LS20 gate is driven low.

Signal /S2 at a high state indicates that the address is a memory address, as opposed to an I/O address. /S2 ANDed with the inverted output of the first LS20 drives the output of the second LS20 low if this address is a memory address in the range B0000-BFFFFH. This low signal is then latched by U12 when ALE goes low. The output of this LS374 is the Bit 7 input of PIO Port B.

The second set of address decoding logic detects any I/O addresses which are not in the range 0100 through 01FF {hex}. This circuitry consists of two LS00 NAND gates (U10) and latch U12. The first NAND gate inverts AD9, thereby obtaining the proper level for the second NAND of U10.

If the I/O address is outside the range 0100-01FFH, the output of the second NAND goes high. This output is latched by the LS374 (U12) when ALE goes low. If the Pin 5 output of the latch is low, then the address is on the 8:16 board (i.e., within addresses 0100-01FFH).

The output of latch U12 is fed back (via signal line /8:16I/O) to the 8086 16-bit board to enable its address decoder when the address is on the 8086 16-bit board. If the output of the latch is high, then the address is not on the 8086 16-bit board and the 8086 address decoder is not enabled.

Latch U12's output is then NANDed with lines IOWR and IORD by the LS00 (U4). IOWR and IORD are the decoded and inverted outputs from the LS137 (U9). U4 consequently generates a signal indicating an I/O read or write outside 0100-01FFH to Bits 5 (I/O write) and 6 (I/O read) of PIO Port B.

Generating an Interrupt

The decoded signals /MEMACC (U12 output Pin 2, the decoded memory access signal within address range B0000 - BFFFF), /IORD, and /IOWR are the three inputs to the Z-80A PIO which cause it to interrupt the Z-80A (Port B, Bits 7, 6, and 5 respectively). The PIO issues an interrupt if any of the three inputs go low. This interrupt to the Z-80A causes it to jump to an interrupt service routine.

/MRD, /IORD, and /IOWR are decoded by U9 (the LS137) from 8086 inputs /S0 (U9 Pin 3), /S1 (U9 Pin 2), and /S2 (U9 Pin 1). /MRD is output on U9 Pin 10, /IORD on Pin 11, and /IOWR on Pin 13.

PIO inputs /IOWR, /IORD, and the /MEMACC signal are ANDed together by two LS08s (U3), so that any one of them going low causes the D input of U1 (half of an LS74 flip-flop) to go low. This input is then latched by the clock input (the output of the 8086 16-bit board wait state generator U27 Pin 6 inverted by an LS00 {U4}). This starts a synchronizing chain of LS74 flip-flops, which finally generates SRDY low, setting the 8086 into a wait state.

Decoding the Interrupt

With the 8086 in a wait state, the Z-80A interrupt service routine may now read Port B of the PIO to determine which Bit (5-7) caused the interrupt. It also reads Bits 0-2 of PIO Port B, which indicates how the interrupt should be handled. Bit 0 (/MWR) indicates whether the interrupt caused by a memory address in the range B0000-BFFFFH was a read or write operation, while Bits 1 and 2 (AD0 and BHE, latched by U12) indicate whether the data transfer was one or two bytes, according to the table below:

AD0	BHE	Bytes Transferred
0	0	Upper-byte on AD8-AD15 Lower-byte on AD0-AD7
0	1	Upper-byte only on AD8-AD15
1	0	Lower-byte only on AD0-AD7
1	1	No data is transferred

An LS138 (U7) appears as an I/O port to the Z-80A. It decodes the addresses of the PIO, and provides to the Z-80A a means to control the secondary board. The I/O map is as follows:

00-03H	CLEAR INTERRUPT TO 8086
04-07H	CLK Z-80A DATA INTO LOW BYTE LATCH
08-0BH	CLK 8086 DATA INTO U13 and U16
0CH	PIO PORT A DATA
0DH	PIO PORT A COMMAND
0EH	PIO PORT B DATA
0FH	PIO PORT B COMMAND/
10-13H	CLK Z-80A DATA INTO HIGH BYTE LATCH
14-17H	INTERRUPT THE 8086
18-1BH	UNUSED
1C-1FH	SET 8086 TO RUN STATE

The Z-80A reads the address that caused the interrupt (latched in U13 and U16) by using Bits 3 and 4 of the PIO to enable the output of these latches, and then reading Port A of the PIO for the data, one byte at a time. It then causes the 8086 data to be latched into these same latches by doing a dummy I/O write to address 08 - 0BH. The Z-80A can now use Bits 3 and 4 of the PIO Port B to enable the outputs of the latches, and then read the data into Port A of the PIO, one byte at a time.

If the cause of the interrupt was an I/O or memory read, the Z-80A must write data into U14 and U15 for the 8086. This is a one- or two-byte transfer, determined by the state of AD0 and BHE as read from the PIO Port B (listed in the preceding table.) This is accomplished by putting the data out on Port A of the PIO and latching it with a dummy write operation to I/O address 04-07H (for low-order bytes) or 10-13H (for high-order bytes). This data latching also sets U1 (the lower half of an LS74), which enables the output of these data latches onto the 8086 data bus.

Releasing the Interrupt

After the appropriate service has been performed, the Z-80A releases the 8086 from its wait state by doing a dummy write to an I/O address in the range 1C-1FH. This dummy write clears U1 (upper-half of an LS74) which clears SRDY.

The Z80 then executes a return from interrupt instruction. The 8086 now finishes the cycle it had started by reading the data that is on its bus from U14 and U15, if the operation was an 8086 read, or simply moving to the next instruction if it was a write.

The output of the LS08 that clocks the lower half of the LS74 (U1) goes low on either an /IORD or a /MRD. This readies the clock input for the positive-going transition that clocks the flip-flop. When the 8086 continues with its read operation, it brings /IORD or /MRD high, (the signal that caused the clock to go low). This low-to-high transition clocks the grounded D input to the output, causing /Q to go high. This, in turn, disables the output of U14 and U15, which completes the cycle and frees the bus. The next 8086 instruction is handled in the same manner.

To allow the Z80 to interrupt the 8086, the Y5 and Y0 outputs of the LS138 (U7) are tied to the Set and Reset inputs of half of an LS74 (U11) (and the 8086 must set 8255 Bit PB0 high). This flip-flop allows the Z80 to set the interrupt request to the 8086 by writing to addresses 14 - 17H and to clear the request by writing to 00 - 03H.

Additional logic on the secondary board is the LS08 (U3) on the M1 input to the PIO. Since the PIO interprets /M1 without an /IORQ or /RD as a reset, the LS08 combines /RES with /M1 to reset the PIO when necessary.

25-Line Circuitry

The 25-line modification is a small circuit board with headers that fit into sockets U422 and U423. U422 is a 74LS157 which multiplexes the display's line number. The board contains this relocated multiplexer (U422) and a 6349 PROM which replaces an S283 counter (U423).

The S283 adder is used to remap the line and column address bits and output the unique character addresses to the character and attribute RAM chips.

The 25-line modification replaces adder U423 with a 6349 PROM, encoded to output to the character RAM the proper address for the line and column position of each character to be displayed.

A 25-line, 80 column display contains 2000 unique locations for characters (25 x 80 = 2000). The binary representation of 2000 is 2^{11}, which means that 11 address bits are required to address all 2000 locations. However, representing the column positions (80) in binary requires 2^7 (seven address lines) and the line number (25) requires 2^5 (five address lines). This results in 12 address lines, even though the upper address combinations of the columns and lines are not used ($2^7 = 120$; $2^5 = 32$)). The PROM is used to remap the 12 lines into 11 for optimized usage of RAM.

The PROM (U423) remaps the address bits by combining the upper three column address bits with all five line number address bits and reassigning the otherwise unused address combinations. Therefore, the lower four column address bits are allowed to input directly to the attribute and character RAM chips, while the PROM provides unique 7-bit outputs to any required address combination on its eight input lines.

High-Resolution Graphics Board

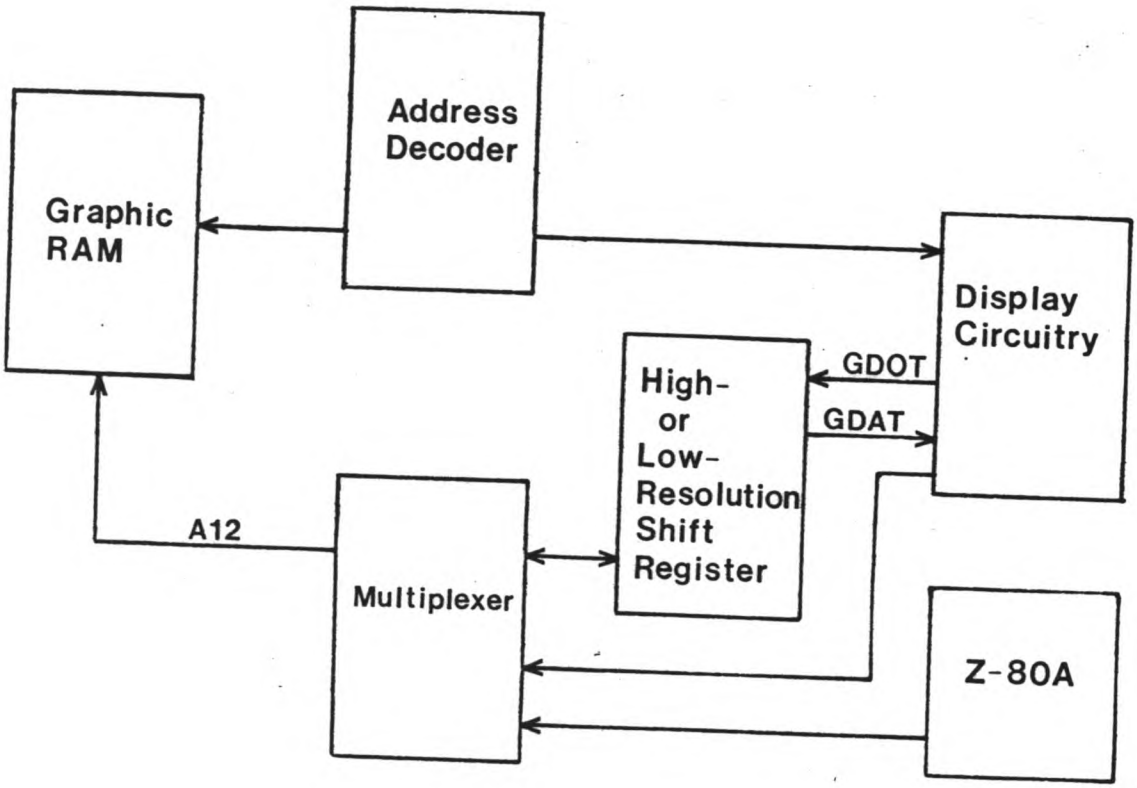
The high-resolution graphics board doubles the memory of the graphic RAM and provides the circuitry necessary to select between low-resolution and high-resolution graphics.

Hardware Modification

Several chips are removed from the Z-80A Main Processor Board, and are replaced with a graphics high-resolution board which plugs directly into the emptied chip sockets.

The modification removes the following chips: U703, an LS166 eight-bit shift register which serially outputs data to the display; U426, the LS138 address decoder which selects one of the graphic RAM chips (or the CRT controller, Attribute RAM or Character RAM); the five graphic RAM chips (MK4802's U704 - U708); and U416, a 2732 EPROM character generator.

The graphics high-resolution board contains three graphic RAM chips (HM6264's U6, U7, and U8), two 8-bit LS166 shift registers (U3 and U4), an LS138 decoder (U2), AND gates (LS08 U5), an LS153 multiplexer (U1), and a 2764 EPROM (U416). The board connects to the Z-80A Main Processor Board via connectors J1 through J7. Connector J1 fits into the U426 socket, J2 into the U703 socket, and J3 into the U708 socket, and J4 through J7 are hard-wired to the appropriate signal lines on the Z-80A processor board.



High-Resolution Graphics Block Diagram

Graphics Board Theory of Operations

The high-resolution graphics board replaces five 2K byte x 8-bit graphic RAM chips with three 8K byte x 8-bit chips. Since the graphics memory is doubled, the graphics board requires an additional address line, A12.

Address line A12 is input to graphic memory through multiplexer U1. This allows graphic memory to be selected by either the processor or the CRT (Cathode Ray Tube) controller. The CRT accesses graphics memory to display the current data via U1 input line R0, and the processor accesses graphics to update data via line A15.

Decoder U2 provides the chip selects to address RAM chip U6, U7, or U8. Additionally, U2 can select the attribute RAM (U432) or the character RAM (U433).

Multiplexer U1's enable line, Pin 14, selects between two 8-bit shift registers, U3 and U4. U3's input lines are coupled such that dots are addressed in pairs, as opposed to individually. U4's input lines select individual dots. When U1's Pin 14 enable is high, U3 is selected for medium-resolution. When Pin 14 is low, U4 is selected and high-resolution graphics is selected.

The shift registers (U3 and U4) are clocked by GDOT, which is input to the board through J2 Pin 7. Each GDOT clock pulse is the time that it takes to write one dot (pixel) to the display. The Pin 15 input to U3 and U4 select parallel load or serial output. When Pin 15 is low, the selected register loads data in parallel.

When Pin 15 is high, the selected register shifts serial graphic data out of Pin 13 to J2 Pin 13, which forms the display signal line GDAT, a direct output to the display interface circuitry.

Additional Circuitry

In addition to the circuitry provided by the high-resolution graphics board, EPROM U416 on the main processor board is replaced.

The new character generator EPROM (U416) contains the IBM character set, in addition to the complete ASCII character set and alternate character sets (Greek, mathematical symbols, business form and graph drawing symbols, German, Spanish, French, Italian, Swedish, and other alphabet symbols.)

Software

Software Overview

The software chapter of this manual is intended for readers with programming knowledge who wish to learn about Attache 8:16's software or to make adaptations for specific applications.

Attache 8:16 is actually two computers in one. As a 16-bit computer, the 8:16 uses an 8086 processor and 256K bytes of RAM with the MS-DOS 2.x operating system (where x is the version number). As a 8-bit computer, the 8:16 uses a Z-80A processor and 64K bytes of RAM with the CP/M 2.2.5 operating system.

CP/M and MS-DOS cannot be run simultaneously. Either system is selected automatically by simply loading the appropriate software. If a disk containing MS-DOS is loaded, the computer is in 16-bit mode running MS-DOS. If a disk containing CP/M is loaded, the computer is in 8-bit mode running CP/M.

Attache 8:16 automatically configures the keyboard, disk format, graphics format, and input/output structure for the selected mode of operation each time a system is loaded.

Applications software written for 8-bit operation under CP/M cannot be run under MS-DOS. Applications software written for 16-bit operation under MS-DOS (or PC-DOS) cannot be run under CP/M. For example, a version of the Multiplan spreadsheet program that is written for MS-DOS cannot be run under CP/M, nor can a spreadsheet program written for CP/M be run under MS-DOS.

You may, however, convert data files that were created under a CP/M version of the application program to an MS-DOS format and vice versa. This is accomplished with the MS-DOS CONVERT utility, which is described in the MS-DOS Guide.

This chapter describes the MS-DOS operating system as implemented on Attache 8:16. For information about the CP/M operating system, refer to the previous section of the Attache Technical Manual.

MS-DOS Introduction

Attache uses Microsoft Corporation's MS-DOS 2.x. MS-DOS is an operating system for the 8086 16-bit processor which allows software to interface with the hardware.

The operating system consists of three parts: BIOS (Basic Input/Output System), MS-DOS (Microsoft's Disk Operating System), and a command processor.

BIOS is the hardware dependent portion of the operating system. The BIOS is specifically tailored for Attache and contained in the file IO.SYS. It defines the hardware environment in which the operating system executes and interfaces with MS-DOS to execute program instructions at the physical level.

MS-DOS performs the logical system input and output. BIOS then translates these logical operations into physical operations. MS-DOS is contained in the file MSDOS.SYS. A complete discussion of MS-DOS is contained in the MS-DOS Guide.

The command processor provides the initial interface between the user and the system. The command processor supplied with MS-DOS (file COMMAND.COM) consists of three parts:

1. **Resident part.** This immediately follows MSDOS.SYS and its data area. This part contains routines to process Interrupts 23H (CTRL C Exit Address) and 24H (Fatal Error Abort Address), as well as a routine which reloads the transient part as required. All standard MS-DOS error handling is performed within this part of COMMAND.COM. This includes displaying error messages and processing the Abort, Retry, or Ignore messages.

2. **Initialization part.** This follows the resident part. During startup, the initialization is given control. Initialization contains the AUTOEXEC file processor setup routine and determines the segment address at which programs can be loaded. It is overlaid by the first program COMMAND.COM loads, since initialization is no longer needed.

3. **Transient part.** This is loaded at the high end of memory and contains all of the internal command processors and the batch file processor. The transient part of the command processor produces the system prompt (such as A>), reads the command from the keyboard or batch file, and causes it to be executed. For external commands, this part builds a command line and issues the EXEC system call (Function Request 4BH) to load and transfer control to the program.

Compatibility With PC-DOS

Attache 8:16's enhanced version of Microsoft Corporation's MS-DOS 2.x is fully compatible with IBM PC-DOS 2.0.

The differences between IBM's PC-DOS and Microsoft's "generic" MS-DOS result from the IBM-PC's physical and logical architecture. Many operations that would normally be handled by the operating system (such as logical input/output instructions) are imbedded in the PC's ROM BIOS.

Since IBM's ROM is proprietary, Attache's MS-DOS handles these functions in the portion of the operating system known as the BIOS (Basic Input/Output System). This section of MS-DOS (IO.SYS) has been customized for Attache 8:16 to emulate the PC's ROM BIOS.

Attache 8:16's BIOS for MS-DOS is completely PC-compatible. Any program written for the PC that performs all input/output operations through IO.SYS will run without modification.

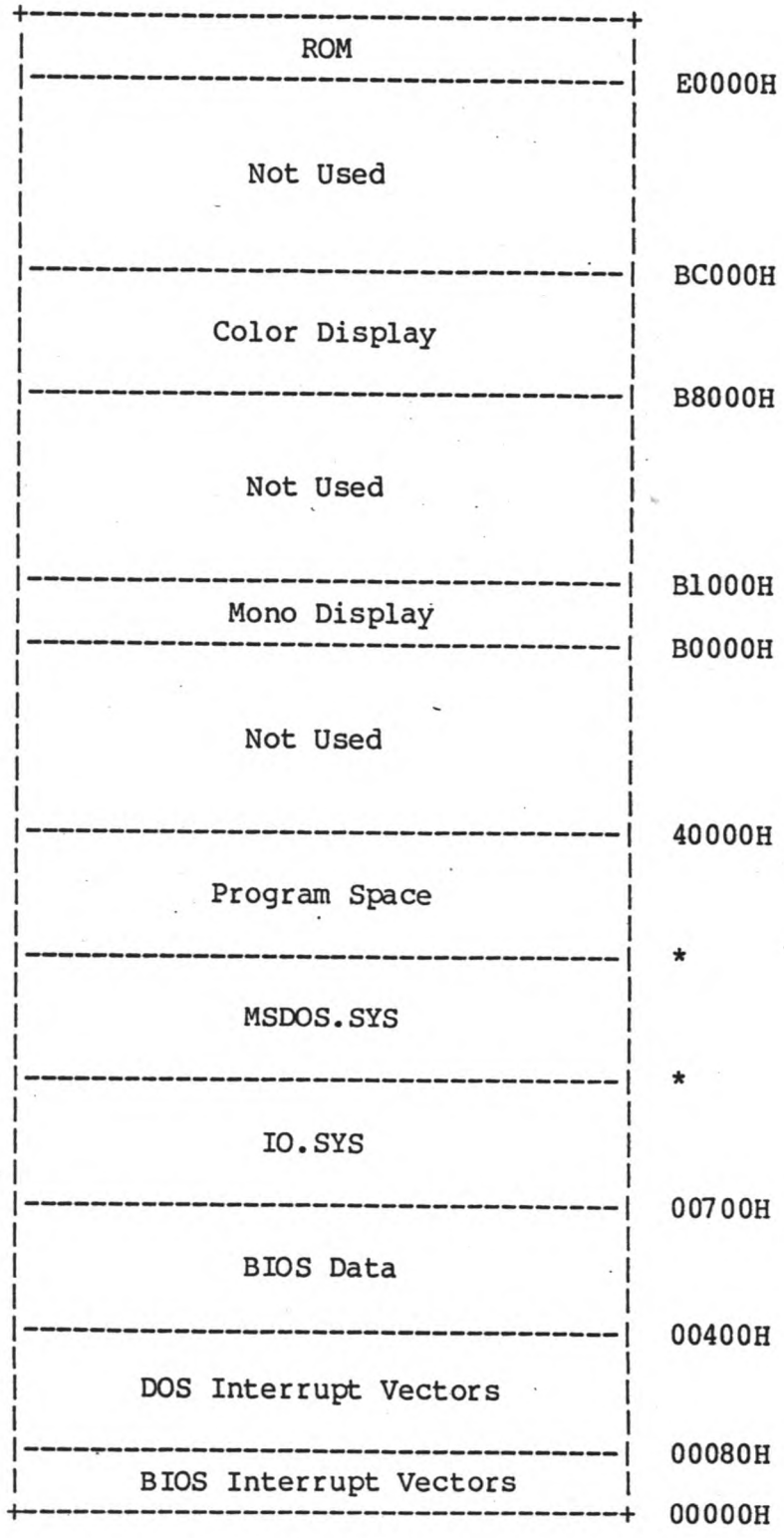
For programs that are written to the IBM ROM rather than the BIOS, Attache's BIOS emulates PC I/O port routines and screen memory. This allows Attache 8:16 to run most applications software that is written for the IBM-PC.

Read-Only Memory Software

Since BIOS is contained in the file IO.SYS, the remaining functions of the 8086 processor's local read-only memory (ROM) are limited to internal diagnostics as part of the boot process.

When the system is booted, the 8086 processor performs the simple internal diagnostics contained in its local ROM. Once the 8086 processor has passed its internal diagnostics, it sends a boot command to the Z-80A. The Z-80A then sends local offset, load segment, byte count, MSDOS.SYS, and the 8086 BIOS to the 8086. If the 8086 internal diagnostics fail, an error command is issued, and the boot process is aborted.

DOS Memory Map



* Determined at run time.

BIOS (Basic Input/Output System) Introduction

BIOS is the hardware dependent portion of the operating system. The BIOS is specifically tailored for Attache 8:16 and contained in the file IO.SYS. It defines the hardware environment in which the operating system executes and interfaces with MS-DOS to execute program instructions at the physical level.

IO.SYS contains code for both the Z-80A 8-bit processor and the 8086 16-bit processor. The Z-80A handles all low level I/O in the system: character input and output buffering, disk reads and writes, and console activity. The 8086 BIOS converts DOS requests into requests to the Z-80A, and does very little I/O processing itself.

The two processors communicate through a single 8255A 8-bit parallel port in bidirectional mode. Since all commands, status information, and data flow through this port, a master/slave communication protocol is used. Most actions are the result of 8086 commands. The Z-80A receives commands, parameters and data from the 8086, and returns status information, error information, and data.

All data transfers are either read operations (from the peripherals through the Z-80A processor to the 8086 processor) or write operations (from the 8086 processor through the Z-80A processor to the peripherals).

Whenever possible, write operations take advantage of the dual processor system. The 8086 processor does not wait for the Z-80A processor to complete a write operation before issuing another write command unless two write operations are already processing. This feature permits simultaneous processing of write operations.

Read operations, however, cannot process simultaneously. The 8086 must wait for the Z-80A to complete the previous read operation before issuing another read command.

Communication between the 8086 processor and the Z-80A processor consists of commands, parameters, status messages, error messages, and data. These take the following form:

Command — An unsigned byte value from the 8086 to the Z-80A.

Parameter — A byte from the 8086. Its meaning is context dependent.

Status Message — A byte from the Z-80A, consisting of eight 1-bit fields.

Error Message — An unsigned byte value. Zero indicates no error condition.

Data Bytes from either the 8086 or the Z-80A.

The following examples of typical command sequences illustrate the logical operation of the 8086/Z-80A interface:

Read Status The 8086 requests the status of some resource (serial channel, diskette drive, printer, etc.), and the Z-80A replies with the status. This command sequence generally processes very rapidly. A single status byte is always sent to the 8086, and the 8086 does not issue another command until the status is received.

8086: Command
Z-80A: Status

Read Character The 8086 requests a byte to be read from a specified device, and the Z-80A replies with a data byte. A Read Status command precedes the Read Character command to check the status of the device.

8086: Command
Z-80A: Status Data

Read Block The 8086 requests a block of data to be read from a specified block device such as a diskette drive. Parameters specify how the block should be read. The Z-80A replies with the status of the block device followed by the data. If the Z-80A replies with an error message, no data will be sent.

8086: Command Parameters
Z-80A: Status Data

or

8086: Command Parameters
Z-80A: Error

Write Character The 8086 requests that a character be written on a specified device. The status of the device is checked by a preceding status request. The Z-80A simply sends the character to the device.

8086: Command Data
Z-80A:

Write Block

The 8086 requests that a block of data be written to the specified block device. Parameters specify the unit, block number, and block count. The Z-80A replies that no error conditions exist. The 8086 sends the data, and the Z-80A sends a completion status byte when the operation has been completed.

8086:	**Command**	**Parameters**		**Data**
Z-80A:			**Error**	**Error**

Additional Interface Operations

In addition to command processing, other operations are also monitored at the interface between the 8086 and the Z-80A processors.

The Z-80A constantly monitors the screen memory and I/O trap hardware to emulate the IBM-PC's hardware environment.

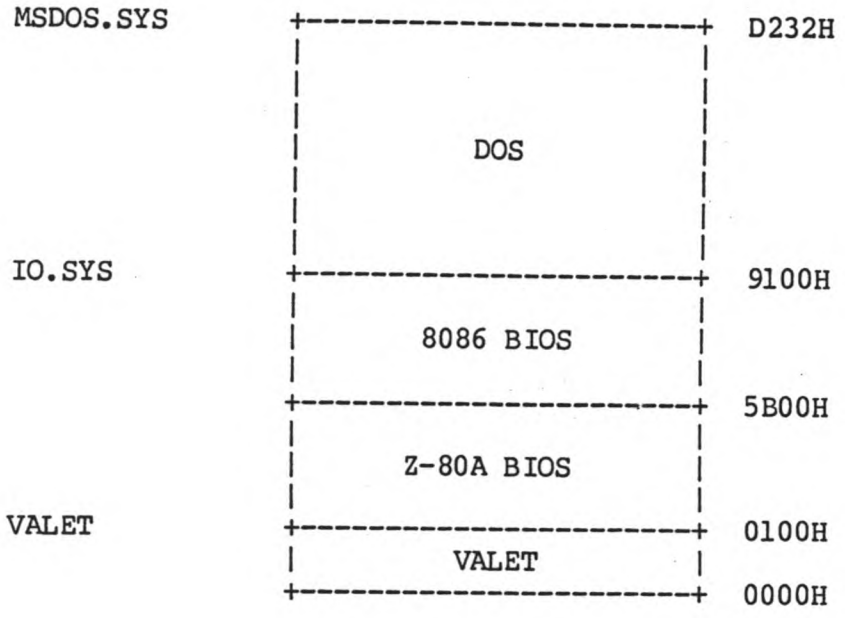
Approximately every 55 milliseconds, the Z-80A sends a timer tick interrupt to the 8086 to emulate the 72 Hz timer of the IBM-PC. The 8086 then issues a command to turn off the interrupt.

Whenever **CTRL ESC** is pressed, the setup mode flag is set. The Z-80A runs the setup program while the rest of the system comes to a halt.

Whenever **CTRL**, **SHIFT**, and **CAPS LOCK** are pressed simultaneously, the pause flag is set. The pause flag causes the Z-80A to ignore any commands and effectively halts the system until another key is pressed.

Booting MS-DOS

BIOS and MS-DOS reside in the files IO.SYS and MSDOS.SYS, respectively. IO.SYS contains both Z-80A and 8086 code. The IO.SYS file has been allocated enough space to allow additional 8086 or Z-80A code to be added without affecting the disk layout or IO.SYS size.



When the system is booted, IO.SYS, MSDOS.SYS, and the Valet data area are loaded into Z-80A memory from the system diskette. The system immediately begins to execute the Z-80A code contained in IO.SYS to initialize the Z-80A processor. At the same time, the 8086 processor performs the simple diagnostics contained in its local ROM.

Once the Z-80A has initialized itself, it waits for the 8086 to send either a boot or error command. This is an exception to the 8086 master/Z-80A slave protocol of the interface. Since the 8086 has no direct contact with the system resources, the Z-80A must perform the required diagnostics on the system resources.

When the 8086 successfully completes its internal diagnostics, it sends a boot command to the Z-80A. The Z-80A then sends load offset, load segment, byte count, MSDOS.SYS, and the 8086 BIOS to the 8086. If the 8086 internal diagnostics fail, an error command is issued, and the boot process is aborted.

Once it receives the code from the Z-80A, the 8086 executes the 8086 BIOS. This step initializes the 8086 processor. When the 8086 has been successfully initialized, it sends a boot complete message to the Z-80A. If the 8086 does not respond, the Z-80A resets the 8086, reports an error, and aborts the boot process.

After the boot process is complete, the Z-80A is in its normal operating mode, waiting for commands from the 8086. The 8086 BIOS and MSDOS.SYS, loaded into Z-80A memory at boot time, can now be overwritten.

BIOS Interrupt Structure

Routines contained in the BIOS can be accessed through software interrupts. Each interrupt accesses a different BIOS routine. For example, interrupt 12H (INT 12H) invokes the BIOS routine that determines the memory size.

Note that if you wish to use BIOS interrupts, do not "hard code" BIOS addresses into your applications. BIOS addresses should never be "hard coded". The internal workings and absolute addresses within BIOS can change without notice.

The 8086 processor provides an interrupt scheme that allows you to define up to 256 subroutines which are invoked through interrupt calls. The addresses of these interrupts are stored in the first 1024 bytes of memory (four bytes per interrupt). You may define the starting address of every interrupt service routine.

To call an interrupt, use the instruction **INT xx** where xx identifies the desired interrupt in hexadecimal notation (00H through FFH). This interrupt number is multiplied by four to yield the absolute address of the interrupt vector's first byte in the entry table. Control is then passed to the interrupt service routine.

There are three types of interrupts: predefined interrupts that are requested by specific functions within the 8086, user defined hardware interrupts, and software interrupts.

The predefined interrupts suggested by Intel are:

Interrupt	0	divide by zero
	1	single step
	2	non-masking interrupt (NMI)
	3	one byte breakpoint interrupt
	4	interrupt on overflow

These interrupts must still be defined by the programmer, and their addresses must be loaded to the proper locations in the vector table.

Interrupt Parameters and Registers

Parameters are passed to and from the BIOS routines through the 8086 registers. When an interrupt can perform several different functions, the value of the AH register is used to indicate the desired operation. For example, to set the time of day, the following code is required:

```
MOV  AH,1           ;load AH to set the time of day
MOV  CX,HIGH_DAY    ;establish the current time
MOV  DX,LOW_DAY
INT  1AH            ;set the time
```

To read the time of day:

```
MOV  AH,0           ;load AH to read the time of day
INT  1AH            ;read the time
```

Generally, the interrupt routines save all registers except for AX and the flags. Other registers are modified only if they are used to return values to the calling routine.

IBM-PC Interrupt Compatibility

To be compatible with software written specifically for the IBM-PC, Attache 8:16's BIOS has implemented the entire set of software interrupts provided by the PC's ROM BIOS in interrupts 00-1FH. MS-DOS itself uses interrupts 20-3FH. Interrupts 90-9FH are used by the 8086 local BIOS, and interrupts F0-FF are hardware vectors.

Any program written for the IBM-PC that uses only the software interrupts IBM has defined should run without modification on the Attache 8:16. Each Attache interrupt routine inputs and outputs its parameters and results through the same registers as the IBM routines (the 8088 processor in the IBM-PC and the 8086 in the Attache 8:16 have identical registers and instruction sets). Programs using these routines will detect no difference between the behavior of corresponding interrupts in the two operating environments.

The IBM interrupts are provided for emulation purposes only. The Attache 8:16 BIOS itself does not use these interrupts. These interrupts are summarized in the following table:

BIOS INTERRUPTS (Numeric Order)

Interrupt Hex	Interrupt Dec	Description
0H	0	Divide by Zero
1H	1	Single Step
2H	2	Non-Maskable Interrupt
3H	3	Breakpoint
4H	4	Overflow
5H	5	Print Screen
6H,7H	6,7	RESERVED
8H	8	Timer
9H	9	Keyboard Hardware (N/A)
AH	10	RESERVED
BH	11	Communications (N/A)
CH	12	Communications (N/A)
DH	13	Disk (N/A)
EH	14	Diskette (N/A)
FH	15	RESERVED
10H	16	Video I/O
11H	17	Equipment Check
12H	18	Memory
13H	19	Diskette I/O
14H	20	Communications Port I/O
15H	21	Cassette (RESERVED) (15H not implemented)
16H	22	Keyboard I/O
17H	23	Printer I/O
18H	24	ROM BASIC (RESERVED) (18H not implemented)
19H	25	Bootstrap (RESERVED) (19H not implemented)
1AH	26	Time of Day
1BH	27	Keyboard Break
1CH	28	Timer Tick
1DH	29	Video Initialize (RESERVED) (1DH not implemented)
1EH	30	Diskette Parameters
1FH	31	Video Graphics Characters (1FH not implemented)

BIOS Interrupt Descriptions

The software interrupts defined by IBM as implemented by the Attache 8:16 are described below. Some interrupts have not been implemented due to hardware restrictions (using a light pen, for example) or extremely low frequency of use (cassette I/O). In these cases, a description of possible error returns is given.

Interrupt 0H - Divide by Zero

This hardware interrupt is requested when a quotient exceeds the maximum value that the division instruction allows.

Interrupt 1H - Single Step

Interrupt 1H occurs one instruction after the trap flag (TF) is set in the program status word. This interrupt is used to execute program instructions one step at a time.

Interrupt 2H - Non-Maskable Interrupt (NMI)

Interrupt 2H is the highest priority (non-maskable) hardware interrupt.

Interrupt 3H - Breakpoint

Interrupt 3H is a software interrupt that occupies one byte of object code. It is used to set breakpoints in software debugging programs and is non-maskable.

Interrupt 4H - Overflow

Interrupt 4H occurs if the overflow flag (OF) is set in the program status word. The INTO instruction is then executed, which allows the 8086 to trap an error service routine. It is non-maskable.

Interrupt 5H - Print Screen

Interrupt 5H prints the contents of the screen.

Interrupt 6H - Reserved

Interrupt 7H - Reserved

Interrupt 8H - Timer

Interrupt 8H handles the timer interrupt from the 8235 timer on the IBM-PC. It maintains the count of interrupts that have been sent from the 8235 since power on, which may be used to calculate the time of day. It calls INT 1CH (timer tick) to execute the code in that interrupt.

Interrupt 9H - Keyboard Hardware (N/A)

Interrupt AH - Reserved

Interrupt BH - Communications (N/A)

Interrupt CH - Communications (N/A)

Interrupt DH - Disk (N/A)

Interrupt EH - Diskette (N/A)

Interrupt FH - Reserved

Interrupt 10H - Video I/O

Interrupt 10H handles the CRT interface. The routine contains 16 function choices. Load the AH register with the appropriate byte identified in the following list, then call INT 10H.

AH = 0 **Set Mode.** Sets the display's mode of operation. Load the AL register with the value that specifies one of the following options. Choosing any of the color options selects the black and white counterpart:

AL = 0 40x25 (double-size character) BW display
AL = 1 40x25 color (automatically selects 40x25 BW)
AL = 2 80x25 (alphanumeric character) BW display
AL = 3 80x25 color (automatically selects 80x25 BW)
AL = 4 320x200 pixel color display (selects 320x200 BW)
AL = 5 320x200 pixel (medium resolution) BW display
AL = 6 640x200 pixel (high resolution) BW display

AH = 1 **Set cursor type.** Not supported.

AH = 2 **Set cursor position.**
DH = the row 0 - 24
DL = the column 0 - 39 (double-size) or 0 - 79
BH = the page number 0 - 3 (must be zero for graphics)

AH = 3 **Return cursor position.** Returns the cursor position in the following registers, with 0,0 the upper left corner:
BH = page number (you supply). Must be 0 for Graphics mode. The system returns:
DH = the row
DL = the column
CH = cursor start line (always 0)
CL = cursor end line (always 0)

AH = 4 **Read light pen.** Not supported. Returns zeros in registers DH, DL, CH, and BX.

AH = 5 **Set page number.** Selects the page number placed in AL. Valid entries are 0 - 3, alpha modes only.

AH = 6 **Scroll up.** Scrolls an area of the screen up.
AL = number of lines to scroll. Input lines blanked at bottom of window (0 means blank entire window)
CH,CL = row,column of upper left corner of scroll
DH,DL = row,column of lower right corner of scroll
BH = attribute to be used on blank line

AH = 7 **Scroll down.** Scrolls an area of the screen down.
AL = number of lines to scroll. Input lines blanked at top of window (0 means blank entire window)
CH,CL = row,column of upper left corner of scroll
DH,DL = row,column of lower right corner of scroll
BH = attribute to be used on blank line

AH = 8 **Read character at current cursor position.**
BH = the page number (you supply). Valid for alpha modes only. The system returns:
AL = the character read upon return
AH = the attribute of the character read upon return (alpha modes only)

AH = 9 **Write character at current cursor position.**
BH = the display page (valid for alpha modes only)
CX = the count of characters to write
AL = the character to write
BL = the attribute of the character

AH = 10 **Write character only at current cursor position.**
BH = the display page (valid for alpha modes only)
CX = the count of characters to write
AL = the character to write

AH = 11 **Set color palette.** Not supported.

AH = 12 **Write dot.** Writes a dot to the graphics display.
DX = the row number
CX = the column number
AL = the color value (If bit 7 of AL is set, then the dot is exclusive ORed.)

AH = 13 **Read dot.** Reads a dot from the graphic display.
DX = the row number
CX = the column number
AL = contains the dot upon return

AH = 14 **Write teletype to active page.** Writes an ASCII character to the active display page.
AL = the character to write
BL = the color value (Note: The screen width is controlled by the previous mode set.)

AH = 15 **Current video state.** Returns the current video state.
AL = the current mode (as described in AH = 0)
AH = the number of character columns on the screen
BH = the current active display page

Alphanumeric modes support four pages numbered from zero to three. This allows you to paint a display page in memory.

Graphic modes support only one page (page zero). If you attempt to select a display or reference page that is too large for the current video mode, the request will be ignored.

"Write character" in graphics mode only works for characters contained on the same row. Continuation to succeeding lines does not work. For "read/write" character, 256 characters may be output in any graphics mode.

When dots are read or written, the 8086 calculates the line and scan and then sends the dot address to the Z-80A. The graphics resolution on the IBM-PC is set at 640 x 200. Attache 8:16's screen has 25 lines and 80 columns, and character cells are each 10 scans by 8 pixels. The Z-80A converts the address from the 8086 to a value for the Attache before attempting to read or write a dot.

The bit values of the character attribute stored in the BL register are:

BL bit 7 = blink
BL bit 6 (BR) = background
BL bit 5 (BG) = background
BL bit 4 (BB) = background
BL bit 3 (I) = intensity
BL bit 2 (FR) = foreground
BL bit 1 (FG) = foreground
BL bit 0 (FB) = foreground

Interrupt 11H - Equipment Check

This routine is used to check which optional devices are attached to the system. Results are output to the AX register. The AX register has the following bit values:

Bit	Value
15, 14	Number of printers attached
13	Not used
12	Game I/O used 1 = yes 0 = no
11, 10, 9	Number of RS-232 cards attached
8	Not used
7, 6	Number of diskette drives 00 = 1 01 = 1 10 = 2 11 = 3
5, 4	Initial video mode 00 = not used 01 = 40X25 BW using color card 10 = 80X25 BW using color card 11 = 80X25 BW using BW card
3, 2	Planar RAM size 00 = 16K 01 = 32K 10 = 48K 11 = 64K
1	Not used

0 Diskette drive(s) attached
1 = yes
0 = no

The hardware configuration of the Attache 8:16 is actually fixed:

1 printer
no game paddles
1 RS-232 card
2 drives
80 x 25 BW
64K planar RAM
drives exist

Interrupt 12H - Memory

Interrupt 12H is used on the IBM-PC to determine the amount of memory from the switch settings on its planar board. The result is stored in the AX register as the number of contiguous 1K blocks in memory. The Attache 8:16 always returns a value of 256 in the AX register.

Interrupt 13H - Diskette I/O

Interrupt 13H provides access to the diskette drive(s). The following options can be selected by loading the corresponding value in the AH register:

AH = 0 **Reset diskette system.** This marks all drives as unknown tracks.

AH = 1 **Read status of diskette system into AL.** This is generated from the last known error from the Z-80A.

AH = 2 **Read.** See Format below.

AH = 3 **Write.** See Format below.

AH = 4 **Verify.** See Format below.

AH = 5 **Format.** Read, write, verify, and format operations use the settings of the following registers:
DL = drive number
DH = head
CH = track
CL = first sector
AL = sector count
ES:BX = DMA address

The number of sectors actually read is returned in the AL register. If an error occurs, returns carry set and code in the AH register. The BX, CX, DX, and DS registers are preserved. The bit values of the error byte returned to the AH register are:

AH bit 0 = write protect
AH bit 1 = unknown unit
AH bit 2 = not ready
AH bit 3 = bad command
AH bit 4 = data error
AH bit 5 = bad structure length
AH bit 6 = seek error
AH bit 7 = unknown media
AH bit 8 = sector not found
AH bit 9 = out of paper
AH bit 10 = write fault
AH bit 11 = read fault
AH bit 12 = other failure

The drive number is used to access the physical parameters necessary for the read and write operations. These parameters are:

BPB pointer
fixed/removable disk
read/write driver address
physical drive number
density flag
sector size
current track
sectors per track
density check track
NEC765 parameters

When a floppy (removable) diskette drive is accessed, the system cannot determine whether the diskette has been changed. The disk change check routine always returns a 0 (do not know) in this case.

Interrupt 14H - Communications Port I/O

Communication through the serial port is handled through the Interrupt 14H routine. Serial device protocols are described at the end of this section.

Interrupt 14H (INT 14H) accesses the RS-232 communications port. The 8086 registers identify the following options:

DX = 0 **Parameter indicating the RS-232 port.** DX = 0 (one Comm port) and DX = 1 (printer port serving as a second Comm port) are supported; other DX codes are ignored.

AH = 0 **Initialize the communications port.** AL contains the following initialization parameters:

Bit:

7 6 5	4 3	2	1 0
--Baud Rate--	--Parity--	--Stopbit--	--Word Length--
0 0 0 = 110	x 0 = none	0 = 1	1 0 = 7 bits
0 0 1 = 150	0 1 = odd	1 = 2	1 1 = 8 bits
0 1 0 = 300	1 1 = even		
0 1 1 = 600			
1 0 0 = 1200			
1 0 1 = 2400			
1 1 0 = 4800			
1 1 1 = 9600			

AL returns status in AX (see AH = 3 below).

Note: Use of Baud rate 75, 134.5, or 19200 is not IBM-compatible and must be handled through Set-up Mode.

AH = 1 **Send character in AL over the communications line.** This preserves the AL register.

On exit, bit 7 of AH is set if the routine was unable to transmit the byte of data over the line. If bit 7 of AH is not set, the rest of AH is set as a status request reflecting the current status of the line.

AH = 2 **Receive a character in AL from the communications line before returning to caller.**

On exit, AH has the current line status as set by the status routine. The only bits left on are error bits 7, 4, 3, 2, and 1. If AH has bit 7 on, the remaining bits are unpredictable. AH is non-zero only if an error occurred.

AH = 3 **Return communications port status in AX.** AH contains line status in bits and AL contains modem status in bits.

AH bits are as follows:
AH bit 7 = time out
AH bit 6 = transmit shift register empty (always = 0)
AH bit 5 = transmit buffer empty (TBE)
AH bit 4 = break detect
AH bit 3 = framing error
AH bit 2 = parity error
AH bit 1 = overrun error
AH bit 0 = data ready

AL bits are as follows:
AL bit 7 = received line signal detect (RLSD) (always 0)
AL bit 6 = ring indicate (RI) (always 0)
AL bit 5 = data set ready (DSR) (always 1)
AL bit 4 = clear to send (CTS)
AL bit 3 = delta receive line signal detect (DRLSD) (always 0)
AL bit 2 = trailing edge ring detector (DRD) (always 0)
AL bit 1 = delta data set ready (DDSR) (always 0)
AL bit 0 = delta clear to send (DCTS) (always 0)

The three protocols for the two serial devices PRN: and AUX: are CTS, NONE, and XON/XOFF. The defaults are CTS for PRN: and NONE for AUX:. To change these, use the following ANSI ESC sequence:

ESC[<number><port>

where <number> is 0 for NONE, 1 for CTS, and 2 for XON/XOFF. The <port> may be a greater-than sign (>) or a right arrow to indicate the printer port, or a less-than sign (<) or a left arrow to indicate the communications port.

For example, to enable XON/XOFF protocol for the printer port, specify the following ESC sequence:

ESC[2>

XON/XOFF causes data output to stop if a CTRL S (XOFF) is sent to the input side of PRN:. Any other character causes output to resume. CTS causes data output to stop whenever the CTS line is false. NONE outputs data blindly.

If XON/XOFF is enabled, it affects the Transmit Buffer Empty (TBE) bit. The TBE bit is shown as AH bit 5 under the Interrupt 14H AH=3 option above. You do **not** want XON/XOFF enabled when transferring binary data (XON/XOFF is disabled by default). If XON/XOFF is enabled and XOFF is received, the transmit buffer fills up and returns TBE false. When an XON character is received, character transmission is resumed and TBE is reset to true.

AH bit 7 under the Interrupt 17H AH=2 option operates the same as TBE, but is opposite in polarity. For example, when an XON character is received, character transmission is resumed and BUSY is set to false.

Interrupt 15H - Cassette

This interrupt has not been implemented. If INT 15H is called, a read error will be returned.

Interrupt 16H - Keyboard I/O

The Interrupt 16H routines provide a keyboard interface with the following options contained in the AH register:

AH = 0 **Read next ASCII character from the keyboard.** Results are returned in AL, and the scan code is returned in AH.

AH = 1 **Console input status.** Indicates if an ASCII character is available to be read.
ZF = 0 if no code is available in buffer.
ZF = 1 if a code is available in buffer. The next character in the buffer is in AX and the entry remains in the buffer (non-destructive read).

AH = 2 **Return the current shift status in AL.** AL bits are defined as follows:
AL bit 7 = insert is active
AL bit 6 = CAPS LOCK toggled
AL bit 5 = NUM LOCK toggled
AL bit 4 = SCROLL LOCK toggled
AL bit 3 = ALT (CTRL + SHIFT) SHIFT pressed
AL bit 2 = CTRL SHIFT pressed
AL bit 1 = left SHIFT pressed
AL bit 0 = right SHIFT pressed

Interrupt 17H - Printer I/O

The routine to access the printer port is Interrupt 17 (INT 17H). The 8086 registers identify the following options:

AH = 0 **Print the character in AL.** On return, AH = 1 if a time out occurred after 10 seconds and the character did not print. Other bits are set as described in AH = 2.

AH = 1 **Initialize the printer port (i.e., flush print buffer).** On return, AH is set with the printer status.

AH = 2 **Read the printer status into AH.** AH bits are as follows:

AH bit 7 = busy: equals 0 (busy) if output buffer full (either full, or XON/XOFF enabled and XOFF received)
AH bit 6 = acknowledge (always 0)
AH bit 5 = out of paper (always 0)
AH bit 4 = selected (always 0)
AH bit 3 = I/O error (always 0)
AH bit 2 = not used
AH bit 1 = not used
AH bit 0 = time out (usually equal to zero, unless a time-out occurs after 10 seconds and the character has not printed.

DX = 0 **Printer to be used.** Only DX = 0 (one printer) is supported.

Interrupt 18H - ROM BASIC

This interrupt has not been implemented on Attache 8:16.

Interrupt 19H - Bootstrap

Interrupt 19H has not been implemented on Attache 8:16.

Interrupt 1AH - Time of Day

Interrupt 1A reads or sets the clock according to the options contained in the AH register. The options are as follows:

AH = 0 **Read current clock setting.** This option returns the following register values:
CX = high portion of count
DX = low portion of count
AL = 0 if 24 hours has not passed since the last read

AH = 1 **Set the current clock,** where:
CX = high portion of count
DX = low portion of count

Counts occur at the rate of 18.2 per second (1193180 counts/65536 seconds).

Interrupt 1BH - Keyboard Break

This vector points to the code that is to be executed when **CTRL C** is pressed. Pressing **CTRL C** is equivalent to pressing the **CTRL** and **BREAK** keys on the IBM-PC. During power on, it is initialized to point to a dummy return instruction (IRET), which the application program may change. The vector is invoked by pressing **CTRL C**, and control should be returned through an IRET instruction.

Interrupt 1CH - Timer Tick

This vector points to the code that is to be executed on every system clock tick. During power on, it is initialized to point to a dummy return instruction (IRET). This vector is invoked while responding to the timer interrupt (INT 8H), and control should be returned through the IRET instruction. It is the responsibility of the application to save and restore all registers that will be modified if this vector points to another routine.

Interrupt 1DH - Video Initialization

Interrupt 1DH has not been implemented on Attache 8:16.

Interrupt 1EH - Diskette Parameters

This vector points to a data region that contains the parameters required for the diskette drive. To change the parameters, a new block must be written, and this vector must be modified to point to the new block.

Interrupt 1FH - Video Graphics Characters

Interrupt 1FH has not been implemented on Attache 8:16.

Compatibility With Color Monitors

Attache 8:16 does not support color modes. The following information is provided to assist in writing IBM-compatible programs.

The IBM-PC has eight video modes, as follows:

Mode:	Description:
0 =>	Text, 25 rows by 40 columns, monochrome
1 =>	Text, 25 rows by 40 columns, color
2 =>	Text, 25 rows by 80 columns, monochrome
3 =>	Text, 25 rows by 80 columns, color
4 =>	Graphics, 200 dot rows by 320 dot columns, color
5 =>	Graphics, 200 dot rows by 320 dot columns, monochrome
6 =>	Graphics, 200 dot rows by 640 dot columns, color
7 =>	Wrap-around mode enabled

Attache 8:16 video modes 0 through 3 support four pages numbered from 0 to 3. The remaining modes support only one page (page 0). If you attempt to select a display or reference page that is too large for the current video mode, an error message is sent to the selected screen.

Eight-Bit Code for Video Attributes

An eight-bit code defines video attributes as follows:

bit 0-2 — Foreground color code
bit 3 — Intensity
bit 4-6 — Background color code
bit 7 — Blink (not supported)

IBM Color Codes

The IBM color adaptor recognizes a 5-bit code for the background color, with the bits defined as follows:

bit 0 => Blue
bit 1 => Green
bit 2 => Red
bit 3 => Intensity
bit 4 => Blink (not supported)

LOW INTENSITY COLORS:		HIGH INTENSITY COLORS:	
00H	00000 -- 0 -- BLACK	08H	01000 -- 8 -- BLACK
01H	00001 -- 1 -- BLUE	09H	01001 -- 9 -- BLUE
02H	00010 -- 2 -- GREEN	0AH	01010 -- 10-- GREEN
03H	00011 -- 3 -- YELLOW	0BH	01011 -- 11-- YELLOW
04H	00100 -- 4 -- RED	0CH	01100 -- 12-- RED
05H	00101 -- 5 -- MAGENTA	0DH	01101 -- 13-- MAGENTA
06H	00110 -- 6 -- CYAN	0EH	01110 -- 14-- CYAN
07H	00111 -- 7 -- WHITE	0FH	01111 -- 15-- WHITE

Bits 3 and 4 of the 5-bit code are not recognized by all color monitors. Bit 3 is used to specify intensity. Bit 4 specifies blink: if Bit 4 is 1, the character blinks; if Bit 4 is 0, the character is steady. Monochrome display screens produce the following results:

Colornum	Color	Result
0	0	BLACK
0	1	UNDERLINE
0	7	WHITE
0	15	HIGH INTENSITY WHITE

Bit 4 can also be used with the color codes listed above to specify the blinking option. For example, 31 (1F hex) produces high intensity white, blinking characters.

IBM Color Palettes

The IBM color adaptor provides two palettes, each of which contains four colors. The palettes are defined as follows:

Color	Palette 0	Palette 1
0	Background	Background
1	Green	Cyan
2	Red	Magenta
3	Yellow	White

Accessing the Display With Escape Sequences

An escape sequence consists of an **ESC** key followed by a series of characters and numbers. Escape sequences perform a variety of functions. For example, an escape sequence can allow you to reassign keys, change graphics functions and modes, erase lines or screens, or affect cursor movement as shown below.

Notes:

1. Spaces appear between the characters of the escape sequence for purposes of readability only. Do not include spaces when you type these escape sequences.

2. The default value is used when no explicit value or when a value of zero is specified.

3. <n> represents a "numeric parameter." This is a decimal number specified with ASCII digits (0 - 9). The maximum value that can be specified is 255.

4. <s> represents a "selective parameter," which is a decimal number that can be used to select a subfunction. Multiple subfunctions may be selected by separating the parameters with semicolons. Up to 16 semicolons may be used at one time. A semicolon not preceded by a number is equivalent to the default value.

Cursor Functions

The following escape sequences affect the cursor position on the screen.

Scrolling Region - Set Margins

ESC [<t> ; <b> ; r

Sets the scrolling region (margins) to the lines specified for the top of the region <t> and the bottom <b>. The region must be two or more lines. Once set, the cursor cannot be moved beyond the margins except with the absolute cursor position command (see below) unless origin mode has been selected (see "Modes of Operation" below). If <t> and <b> are not specified or are 0 and 1, they default to the physical top and bottom of the screen. The sequence **ESC[r** effectively clears the margins.

CUP - Absolute Cursor Position

ESC [<l> ; <c> H

HVP - Horizontal and Vertical Position

ESC [<l> ; <c> f

CUP and HVP move the cursor to the position specified by the parameters, where the first parameter <l> specifies the line number and the second parameter <c> specifies the column number. The default value is 1. If no parameters are specified, the cursor is moved to the home position.

If a scrolling region has been specified and origin mode is in effect (see "Scrolling Region" above), the cursor position is relative to the set margins (where the top margin is line 1). If the line <l> is beyond the margin, the cursor does not move. If the column <c> is beyond the margin, it is placed at the end of the (new) line.

CUU - Cursor Up

ESC [<n> A

CUD - Cursor Down

ESC [<n> B

These sequences move the cursor up or down one line without changing columns. The value of <n> determines the number of lines moved. The default value for <n> is 1. If wrap has not been specified, the sequence is ignored when the cursor is already on the top line (for CUU) or bottom line (for CUD). If wrap is in effect, the cursor will wraparound as necessary.

CUF - Cursor Forward

ESC [<n> C

CUB - Cursor Backward

ESC [<n> D

These sequences move the cursor forward or back one column without changing lines. The value of <n> determines the number of columns moved. The default value for <n> is 1. The sequence is ignored when the cursor is already in the far right column (for CUF) or far left column (for CUB) if wrap is not specified. If wrap is in effect, the cursor will wraparound as necessary.

LF - Line Feed

ESC D

This escape sequence moves the cursor down one line. It has the same effect as pressing **LINE FEED.**

RLF - Reverse Line Feed

ESC M

This sequence moves the cursor up one line. Column position is not affected. If the cursor is at the top of the screen, the screen scrolls down one line. The top line is cleared if wrap is in effect.

CLF - Control Line Feed

ESC E

The cursor moves to the beginning of the next line. If the cursor is at the bottom of the screen, the screen scrolls up one line.

DSR - Device Status Report

ESC [6 n

The console driver outputs a CPR sequence (see below) on receipt of the DSR escape sequence.

CPR - Cursor Position Report (from console driver to system)

ESC [<l> ; <c> R

The CPR sequence reports current cursor position via standard input. The first parameter <l> specifies the current line and the second parameter <c> specifies the current column.

SCP - Save Cursor Position

ESC [s

The current cursor position is saved. This cursor position can be restored with the RCP sequence (see below).

RCP - Restore Cursor Position

ESC [u

This sequence restores the cursor position to the value it had when the console driver received the SCP sequence (see above). If no SCP was specified, the cursor moves to the home position.

Erase Functions

The following escape sequences affect erase functions for the screen display or various lines.

ED - Erase Display

ESC [<n> J

where <n> may be:

0 - erase to end of screen (default)
1 - erase to beginning of screen
2 - erase entire screen and place the cursor in the home position

Origin or margins do not affect this sequence. 0 and 1 options do not change the cursor position.

EL - Erase Line

ESC [<n> K

where <n> may be:

0 - erase to end of line (default)
1 - erase to beginning of line
2 - erase entire line

The 0 option erases from the cursor to the end of the line (including the cursor position). None of the options affect the cursor position.

Adding or Deleting Lines

IBL - Insert Blank Lines

ESC [<n> L

This sequence inserts <n> blank lines in front of the cursor. It does not affect cursor position. <n> defaults to one. The current line and all following lines are scrolled down to make room for the new line(s). Scrolling is affected by set margins.

DL - Delete Lines

ESC [<n> M

This sequence deletes <n> lines starting at the cursor. It does not affect the cursor position. Subsequent lines are scrolled up, overwriting the current line. Blank lines are inserted at the bottom of the screen. <n> defaults to one.

DC - Delete Characters

ESC [<n> P

This sequence deletes <n> characters from the cursor position.

Setting or Clearing Tabs

ST - Set Tabs

ESC H

This sequence sets a tab at the current cursor column position.

CT - Clear Tabs

ESC [<n> g

This sequence clears a tab at the current cursor column position (<n>=0) or clears all tabs (<n>=3).

Modes of Operation

The following escape sequences affect screen graphics.

SGR - Set Graphics Rendition

ESC [<s> ; ... ; <s> m

The SGR escape sequence invokes the graphic functions specified in the following table, where <s> is the number of the desired subfunction. The graphic functions remain in effect until the next occurrence of an SGR escape sequence. If no arguments are given, all attributes are cleared.

Parameter	Parameter Subfunction	
0	All Attributes off	
1	Bold on	
4	Underscore on	(monochrome displays only)
5	Blink on	(not available)
7	Reverse Video on	
8	Concealed on	(ISO 6429 standard)*
30	Black foreground	(ISO 6429 standard)*
31	Red foreground	(ISO 6429 standard)*
32	Green foreground	(ISO 6429 standard)*
33	Yellow foreground	(ISO 6429 standard)*
34	Blue foreground	(ISO 6429 standard)*
35	Magenta foreground	(ISO 6429 standard)*
36	Cyan foreground	(ISO 6429 standard)*
37	White foreground	(ISO 6429 standard)*
38	Superscript on	
39	Subscript on	
40	Black background	(ISO 6429 standard)*
41	Red background	(ISO 6429 standard)*
42	Green background	(ISO 6429 standard)*
43	Yellow background	(ISO 6429 standard)*
44	Blue background	(ISO 6429 standard)*
45	Magenta background	(ISO 6429 standard)*
46	Cyan background	(ISO 6429 standard)*
47	White background	(ISO 6429 standard)*
100	Highlight on	
101	Strikethrough on	

* Ignored

SM - Set Mode

ESC [<s> h

RM - Reset Mode

ESC [<s> l

The SM and RM escape sequences change the screen width or type (see notes below). "l" is a lowercase "L." <s> is one of the following parameters:

Parameter	Parameter Subfunction
0	40 x 25 black and white
1	40 x 25 color
2	80 x 25 black and white
3	80 x 25 color
4	320 x 200 color
5	320 x 200 black and white
6	640 x 200 black and white
7	Wrap at end of line
10	Insert mode
11	Origin mode

Notes:

1. 0 or 1 -- Selects 40-column mode (default). The screen is cleared and all subsequent characters are displayed in double-width.

2. 2 or 3 -- Selects 80-column mode. The screen is cleared and an 80-character screen is used.

3. 7 -- Autowrap is set with **ESC[7h** to allow the cursor to wrap around the screen to the following line. To cancel the wraparound, use the sequence **ESC[7l.**

4. 10 -- The sequence **ESC[10h** enables insert mode, which allows characters to be inserted at the cursor position. To cancel insert mode, use the sequence **ESC[10l.**

5. 11 -- The sequence **ESC[11h** enables origin mode, which causes absolute cursor positions to become relative to the set margins (if any). To disable origin mode, specify **ESC[11l.**

Monitor Mode - Diagnostics

Diagnostic programs for troubleshooting and performance verification on the Attache 8:16 are available in monitor mode. To enter monitor mode:

1. Turn on power.
2. Open Drive A door.
3. Press **RESET** and the right **SHIFT** key.
4. The message **Now in Terminal Mode** will be displayed.
5. Press **CTRL** and **LINE FEED** at the same time.
6. The @ prompt indicates that you are in monitor mode.

The Z-80A main processor retains control of monitor functions. If the 8086 16-bit processor board is not installed, all tests except H will return a lowercase "x".

Commands for running diagnostics on the 8:16 are:

[-- 8086 RAM Test	(-- GPIB Listener/Talker Test*
] -- SCC Test*	) -- GPIB Controller Test*
H -- Display RAM Test**	

* Optional features

** The H (Display RAM) Test does not return an "x" error if 8086 16-bit processor board is not present

Notes on the 8:16 Diagnostic Tests

1. You must be in monitor mode to run the diagnostic tests.

2. The Serial Communications Controller (SCC) Test and the two GPIB tests are tests of optional features. These tests return an "x" if the feature is not available on the 8:16 being tested.

3. You may run a test in a continuous loop by activating the L (Loop) test before typing the command for the program you wish to run. For example, to run the 8086 RAM Test in a continuous loop, type L after the @ prompt, then type [on the following line. The test will run continuously until you press any key. See the description of the L Test in Section 2 of the Service Guide.

[— 8086 RAM Test

Format: [(no parameters)

Function: Tests the 8086 16-bit processor board system RAM of 256K bytes.

Screen Shows: [M

Run Time: Less than 3 seconds.

Exit: Automatic at end of test.

Reports: Returns to the next line and displays an @ prompt if the test is successful.

Errors are reported in the format Unn, where nn is the U code of the bad RAM chip as follows:

Row 1: U34 - U41

Row 2: U44 - U51

Row 3: U54 - U61

Row 4: U64 - U71

Failures: Suspect (1) loose cable connections, (2) 8086 16-bit processor board, (3) power supply module.

] — Serial Communications Controller (SCC) Test

Format:] (no parameters)

Function: Performs a test of the Serial Communications Controller (SCC). A loopback connector is required for this test.

To create a loopback connector, attach a wire from pin 2 to pin 3 of the 8086 16-bit processor board's communications port (the transmit/receive lines). The communications port, if installed, is at the upper left of the option board plate.

Screen Shows: [S

Run Time: Less than 2 seconds.

Exit: Automatic at end of test.

Reports: Returns and displays @ prompt if test is successful. An "x" indicates that the SCC option is not installed. A "?" indicates SCC failure.

Failures: Suspect (1) loose cable connections, (2) SCC option failure.

(-- GPIB Listener/Talker Test

Format: ((no parameters)

Function: Exercises all General Purpose Interface Bus (GPIB) lines. This test only works when the unit is connected to another Attache 8:16 running the GPIB Controller Test. This test must be started before the GPIB Controller Test.

Screen Shows: [L

Run Time: Less than 2 seconds if successful, or 45 seconds if GPIB option failure occurs.

Exit: Automatic at end of test.

Reports: Returns and displays @ prompt if test is successful. An "x" indicates that GPIB option is not installed. A "?" indicates GPIB test failure.

Failures: Suspect (1) loose cable connections, (2) GPIB option is not installed on this machine, (3) Listener/Talker Test was not started before GPIB Controller Test, (4) GPIB option failure.

) — GPIB Controller Test

Format:) (no parameters)

Function: Exercises all General Purpose Interface Bus (GPIB) lines. Must be connected to another Attache 8:16 which is running the GPIB Listener/Talker Test. The GPIB Controller Test must be started after the GPIB Listener/Talker Test.

Screen Shows: [T

Run Time: Less than 2 seconds if successful, or 45 seconds if GPIB option failure occurs.

Exit: Automatic at end of test.

Reports: Returns and displays the @ prompt if test is successful. An "x" indicates that no GPIB option is installed. A "?" indicates GPIB test failure.

Failures: Suspect (1) loose cable connections, (2) GPIB option is not installed on this machine, (3) Controller Test was not started after Listener/Talker Test, (4) GPIB option failure.

H -- Display RAM Test

Format: H (no parameters)

Function: This test is an extension of the H Display RAM test described in Section 2 of the Service Guide.

Tests all bits and locations of the alphanumeric, graphic, and attribute display memories. Automatically tests for 25 lines and increased graphics on the Z-80A main processor board if the 8086 board is installed and functional, and tests if the extra memory is present.

Screen Shows: H

Run Time: Less than 10 seconds.

Exit: Automatic at end of test.

Reports: Returns and displays @ prompt if test is successful.

An error is reported in the following format:

llccddff

where llccdd is the line-character-bit position of the error, and ff is the frame code as described for the Display RAM Test in Section 2 of the Service Guide.

This error code is not reliable under certain conditions. If ff of the error code is C0 or E0, the error message is correct. If ff is 00, 20, 40, 80, or A0, the error is in the graphics RAM section of the Z-80A main processor board, but you do not know which of the three RAMs has the error.

Because this test is an extension of the Z-80A main processor board Display RAM Test, an error will not show if the 8086 board is present and the graphics option is not installed. Instead, the regular Z-80A Display RAM Test is run.

Failures: Suspect (1) loose cable connections, (2) Z-80A main processor board.

IBM Hardware Emulation

Attache 8:16 software emulates a variety of IBM-PC hardware devices. Emulated hardware is described below.

Alphanumeric Memory

The Z-80A processor constantly checks to see if the 8086 is attempting to access memory. When the 8086 requests a memory read or write cycle, the address requested by the 8086 is latched, a hardware interrupt is issued to the Z-80A, and the 8086 is placed in a wait state. The Z-80A reads the requested address from the latch and up to two data bytes. The Z-80A then processes the 8086 request as follows:

Memory Write If a word access is requested, the low 8 bits of data are written to a character cell (all even addresses are characters, all odd addresses are attributes), then the attribute bits are written to the high 8 bits.

If a byte access is requested, /BHE (Bus High Enable) is checked to determine whether the request is for a character or an attribute byte. The 8086 always puts even addresses (characters) on the lower-half of the bus and odd addresses (attributes) on the upper-half.

Memory Read Since the 8086 ignores the data that it does not want (either the upper- or lower-half of the bus), the Z-80A just reads both the character and the attribute and places both on the 16-bit bus. The 8086 address is stored in the HL register.

DIP Switches

Attache 8:16 software emulates IBM-PC 8255 ports. The emulation of the IBM-PC's 8255 Port A and Port C is coded to assume that DIP switch data has been stored in CMOS RAM. For example, bit 1 can equal switch position 1. In this case, a 1 in a bit position means a switch is off, a 0 means it is on. Bits PB7 and PB2 of Port B control which DIP switches are read.

Register E contains data to simulate data on the bus of the 8255. Register L contains a simulated hardware port address on the IBM 8255 Port B.

The 8255 Port A emulation routine reads DIP switch 1 data if PB7 is high. The 8255 Port C emulation routine reads DIP switch 2, bits 1 through 4 data into PC0 through PC3 if PB2 is high. Otherwise, it reads DIP switch 2, bit 5 data into PC0 and writes ones into PC1 through PC3.

Graphics Memory

The Z-80A constantly checks to see if the 8086 is attempting to access graphics memory. If the 8086 is waiting to access graphics memory, the Z-80A reads the 8086 address stored in register HL and up to two data bytes. The Z-80A then processes the 8086 request as follows:

Graphic Write The Z-80A converts the data from PC format to Attache format. It then writes the data to the next location.

Graphic Read The Z-80A converts the PC screen location to an Attache screen location then reads two bytes.

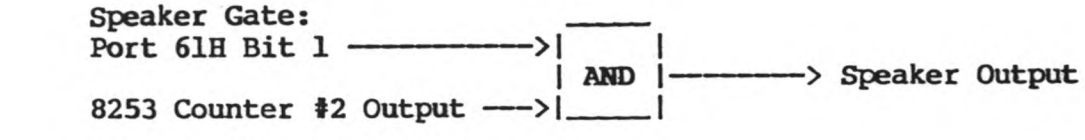
A table is used to convert the 8086 16-bit address to line and column numbers in the 24 x 80 character format used by the Attache's Standard Microsystems 5027 CRT Controller.

Sound Generator

The sound generator on Attache 8:16 is set up to emulate the IBM-PC hardware. It is designed to allow IBM-PC programs which use sound to function on Attache 8:16.

IBM-PC sound hardware consists of an Intel 8253 counter/timer chip. The 8253 includes a timer clock channel which is used for square wave generation (Counter #2 output); this produces tone for the speaker. The IBM-PC's Port B 61 hex (Port 61H) bits 0 and 1 also control speaker output by gating the signal from Counter #2.

The following diagram illustrates sound generation using an 8253. As shown below, both Port 61H bit 1 and Counter #2 output must produce a high signal to activate the speaker. Port 61H bit 1 or Counter #2 output may go low to deactivate the speaker.



The frequency of Counter #2's square wave may be set and changed by programming the 8253 mode and count registers (refer to Intel documentation for details). Bit 0 of Port 61H causes Counter #2 output to be high (with a 0) or square wave (with a 1). Bit 1 of Port 61H turns the speaker gate on (with a 1) and off (with a 0).

The speaker gate must be set high before output from Counter #2 on the 8253 can be passed to the speaker. Similarly, Counter #2 output may be set high (bit 0=0), allowing the speaker to be toggled by turning the speaker gate bit (bit 1) on and off. This second method is commonly used in IBM-PC programs.

Attache 8:16 software emulates the IBM hardware using Attache's sound generator chip. The following port addresses simulate the IBM ports and can be used to affect speaker output:

PORT:	FUNCTION:
61 (HEX)	BIT 0=COUNTER #2 GATE, BIT 1=SPEAKER GATE
42 (HEX)	COUNTER REGISTER #2 IN THE 8253 EMULATOR
43 (HEX)	EMULATED 8253 MODE REGISTER

Only MODE 3 (square wave output) of the 8253 is emulated. The count register may be loaded with a low, high, or 2 byte load using the proper mode register commands. Only Counter #2 output is available. The counter register cannot be read back at any time.

Note: The constant high output on the 8253 Counter #2 output is simulated by outputting a 125KHz square wave to the speaker. When using the speaker gate as a toggle, higher frequencies may beat with the 125KHz signal and cause a warbling sound.

Timer Tick

Approximately every 55 milliseconds, the Z-80A processor sends a timer tick interrupt to the 8086 processor to emulate the 72 Hz timer used in the IBM-PC. The 8086 processor then commands the Z-80A to turn off the timer tick interrupt. All other Z-80A command sequences cannot be interrupted.

6845 CRT Controller

Attache 8:16 software emulates the IBM 6845 CRT Controller. Commands to either the color or the monochrome display are accepted. The cursor location is contained in registers R14 (high-byte) and R15 (low-byte). Line and column are returned as an address in the HL register. A conversion table is then used to convert the address to a 24 x 80 character format for the Attache's Standard Microsystems 5027 CRT Controller. Horizontal and vertical retrace are simulated by generating fast horizontal pulses and a vertical pulse after every 64 horizontal pulses.

Keyboard Functions

Keyboard functions include keyboard modes, special key sequences, and keyboard reassignment as described below. Detailed charts of the Attache 8:16 keyboard appear at the end of the keyboard section. Access to the keyboard through Interrupt 16H is discussed in the BIOS Interrupt Descriptions section of this chapter.

Keyboard Modes

The Attache 8:16 has three different keyboard modes: ASCII, WordStar, and IBM-PC. To choose a mode, send the proper escape code as shown below:

ESC[<n>.

where <n> may be:
0 = ASCII mode
1 = WordStar mode
2 = IBM-PC mode

When MS-DOS is loaded, the default mode is IBM-PC. Each mode is briefly described below.

ASCII Mode -- Shows the character codes generated by keystrokes in base state (a key pressed without any modifying keys). It also shows characters produced in the SHIFT, CTRL, 10-Key Mode, and CAPS LOCK shift states.

IBM-PC Mode -- Shows the keystrokes on the IBM-PC and the keystrokes on the Attache 8:16 which produce the same code(s). IBM-PC mode is the default mode when MS-DOS is loaded.

WordStar Mode -- Shows the mapping of the Attache keyboard to WordStar command character sequences. Single keystrokes may send more than one character sequence depending on the mapping.

Special Key Sequences

The following key sequences perform the described function at all times, regardless of the keyboard mode. A + indicates that these keys must be pressed simultaneously.

RESET + RIGHT SHIFT	This always performs a cold boot of the machine.
CTRL + ESC	This activates VALET and places you in Set-up Mode.
CTRL + CAPS LOCK	This activates or deactivates 10-Key Mode.

Keyboard Reassignment

Although not part of the ANSI 3.64-1979 or ISO 6429 standard, the following keyboard reassignments are compatible with these standards:

```
     ESC [ <n> ; <n> ; ... <n> p
or   ESC [ "string" ; p
or   ESC [ <n> ; "string" ; <n> ; <n> ; "string" ; <n> p
or   any other combination of strings and decimal numbers
```

The final code in the control sequence (p) is one reserved for private use by the ANSI 3.64-1979 standard.

The first ASCII code in the control sequence defines which code is being mapped. The remaining numbers define the sequence of ASCII codes generated when this key is intercepted. There is one exception: if the first code in the sequence is zero (NUL), the first and second code make up an extended ASCII redefinition.

Examples:

1. Reassign the "Q" and "q" key to the "A" and "a" key (and vice versa). All numbers in this example are decimal:

ESC[65;81p	A becomes Q
ESC[97;113p	a becomes q
ESC[81;65p	Q becomes A
ESC[113;97p	q becomes a

2. Reassign the **CTRL 0** key to a DIR command followed by a return:

```
ESC[0;68;"dir";13p
```

The 0;68 is the extended ASCII code for the **CTRL 0** key; 13 decimal is a carriage return.

The Attache 8:16 Keyboard

In normal operating modes, the Attache keyboard returns ASCII to the processor. During IBM-PC emulation mode, Attache 8:16 emulates the IBM keyboard and returns make and break scan codes.

The IBM-PC keyboard typewriter keys return make scan codes of 1 through 58. Code 1 is the upper left key (ESC) and 58 is the lower right key (CAPS LOCK). Function keys F1 through F10 comprise scan codes 59 through 68, and keypad keys (beginning with NUM LOCK and SCROLL LOCK) produce codes 69 through 83. Break codes are determined by adding hex 80 to make codes.

The following charts describe keys and codes used by the Attache 8:16. ASCII mode, ASCII character codes, IBM-PC mode, and an IBM-PC keyboard conversion table are described in turn.

ASCII Mode

ASCII mode returns character codes as designated by ASCII. All entries in the table are actual characters except for the two columns containing the decimal and hexadecimal codes. A blank spot means no code is generated.

ASCII Mode Key Codes

ATTACHE KEY	Base State ASCII CODE dec	hex	BASE CHAR	Shift States (ASCII Characters) SHIFT	CTRL	CTRL SHIFT	10-KEY MODE	CAPS LOCK
0	48	30H	0	^			0	0
1	49	31H	1	!			1	1
2	50	32H	2	@			2	2
3	51	33H	3	#			3	3
4	52	34H	4	$			4	4
5	53	35H	5	%			5	5
6	54	36H	6	&			6	6
7	55	37H	7	*			7	7
8	56	38H	8	(			8	8
9	57	39H	9	)			9	9
A	97	61H	a	A	^A		A	A
B	98	62H	b	B	^B		B	B
C	99	63H	c	C	^C		C	C
D	100	64H	d	D	^D		D	D
E	101	65H	e	E	^E		E	E
F	102	66H	f	F	^F		F	F
G	103	67H	g	G	^G		G	G
H	104	68H	h	H	^H		H	H
I	105	69H	i	I	^I		5	I
J	106	6AH	j	J	^J		1	J
K	107	6BH	k	K	^K		2	K
L	108	6CH	l	L	^L		3	L
M	109	6DH	m	M	^M		0	M
N	110	6EH	n	N	^N		N	N
O	111	6FH	o	O	^O		6	O
P	112	70H	p	P	^P		P	P
Q	113	71H	q	Q	^Q		Q	Q
R	114	72H	r	R	^R		R	R
S	115	73H	s	S	^S		S	S
T	116	74H	t	T	^T		T	T
U	117	75H	u	U	^U		4	U
V	118	76H	v	V	^V		V	V
W	119	77H	w	W	^W		W	W
X	120	78H	x	X	^X		X	X
Y	121	79H	y	Y	^Y		Y	Y
Z	122	7AH	z	Z	^Z		Z	Z

ASCII Mode (continued)

ATTACHE KEY	Base State ASCII dec	CODE hex	BASE CHAR	Shift States (ASCII Characters) SHIFT	CTRL	CTRL SHIFT	10-KEY MODE	CAPS LOCK
-	45	2DH	-	_			-	-
=	61	3DH	=	+	=		=	=
[	91	5BH	[	{	GS	GS	[	[
]	93	5DH	]	}	ESC	ESC	]	]
;	59	3BH	;	:	;	:	*	;
'	39	27H	'	"	'	"	'	'
,	44	2CH	,	<	,	<	,	,
.	46	2EH	.	>				
/	47	2FH	/	?	/	?	/	/
`	96	60H	`	~			`	`
\	92	5CH	\	|	FS	FS	\	\
ESC	27	1BH	ESC	ESC	VALET	ESC	ESC	ESC
BS	08	08H	BS	BS	DEL	DEL	BS	BS
TAB	09	09H	HT	HT	HT	HT	HT	HT
LF	10	0AH	LF	LF	LF	LF	LF	LF
CR	13	0DH	CR	CR	CR	CR	CR	CR
DEL	127	7FH	DEL	DEL	DEL	DEL	DEL	DEL
LEFT	08	08H	BS	BS	BS	BS	BS	BS
DOWN	10	0AH	LF	LF	LF	LF	LF	LF
UP	11	0BH	VT	VT	VT	VT	VT	VT
RIGHT	12	0CH	FF	FF	FF	FF	FF	FF
SPACE	32	20H	SP	SP	SP	SP	SP	SP
CAPS RESET SHIFT								

ASCII Character Codes

ASCII dec	CODE hex	ASCII CHAR.	ASCII dec	CODE hex	ASCII CHAR.	ASCII dec	CODE hex	ASCII CHAR.
000	00	NUL	043	2B	+	086	56	V
001	01	SOH	044	2C	'	087	57	W
002	02	STX	045	2D	-	088	58	X
003	03	ETX	046	2E	.	089	59	Y
004	04	EOT	047	2F	/	090	5A	Z
005	05	ENQ	048	30	0	091	5B	[
006	06	ACK	049	31	1	092	5C	\
007	07	BEL	050	32	2	093	5D	]
008	08	BS	051	33	3	094	5E	^
009	09	HT	052	34	4	095	5F	_
010	0A	LF	053	35	5	096	60	`
011	0B	VT	054	36	6	097	61	a
012	0C	FF	055	37	7	098	62	b
013	0D	CR	056	38	8	099	63	c
014	0E	SO	057	39	9	100	64	d
015	0F	SI	058	3A	:	101	65	e
016	10	DLE	059	3B	;	102	66	f
017	11	DC1	060	3C	<	103	67	g
018	12	DC2	061	3D	=	104	68	h
019	13	DC3	062	3E	>	105	69	i
020	14	DC4	063	3F	?	106	6A	j
021	15	NAK	064	40	@	107	6B	k
022	16	SYN	065	41	A	108	6C	l
023	17	ETB	066	42	B	109	6D	m
024	18	CAN	067	43	C	110	6E	n
025	19	EM	068	44	D	111	6F	o
026	1A	SUB	069	45	E	112	70	p
027	1B	ESCAPE	070	46	F	113	71	q
028	1C	FS	071	47	G	114	72	r
029	1D	GS	072	48	H	115	73	s
030	1E	RS	073	49	I	116	74	t
031	1F	US	074	4A	J	117	75	u
032	20	SPACE	075	4B	K	118	76	v
033	21	!	076	4C	L	119	77	w
034	22	"	077	4D	M	120	78	x
035	23	#	078	4E	N	121	79	y
036	24	$	079	4F	O	122	7A	z
037	25	%	080	50	P	123	7B	{
038	26	&	081	51	Q	124	7C	|
039	27	'	082	52	R	125	7D	}
040	28	(	083	53	S	126	7E	~
041	29	)	084	54	T	127	7F	DEL
042	2A	*	085	55	U			

IBM-PC Mode

The Attache 8:16 keyboard is capable of producing the same codes as an IBM-PC keyboard. The tables below show which keys on an Attache 8:16 produce codes that correspond to IBM-PC keyboard codes.

When struck, a key returns either an ASCII character code or a two-byte sequence in registers AL, AH. The two-byte sequence returns NUL (00) in AL and a code indicating the function in AH. The following tables show the character that corresponds to the ASCII code generated or, if a keystroke generates a two-byte sequence, the actual code generated.

To produce the codes in the last four columns of the tables, two "keystrokes" are required. The **CTRL** or **SHIFT** combination must be struck first and then **released,** followed by the second key (or key combination) shown in the column. The code is **not** generated by striking the key in the column while simultaneously holding down the other keys.

Special Key Combinations

Special key combinations are required on an Attache 8:16 to emulate the functions available on an IBM-PC keyboard. These special key combinations are summarized in the list below. In this list, + indicates a simultaneous strike and , indicates a pause.

CTRL + <number> **<1 - 0>**	This sends the same code as the IBM function keys F1 - F10 (Attache number keys 1-0).
CTRL + SHIFT + <number>	This sends the same code as IBM-PC keys SHIFT + F1 - F10.
CTRL + CAPS LOCK , **CTRL + SHIFT +** <number>	This sends the same code as IBM-PC keys ALT 1-0, -, and =. Release **CTRL + CAPS LOCK** before pressing **CTRL + SHIFT + <number>.**
CTRL + TAB , <number>	This sends the same code as IBM-PC keys CTRL + F1 - F10. Release **CTRL + TAB** before pressing <number>.
SHIFT + ESC , <number>	This sends the same code as IBM-PC keys ALT + F1 - F10. Release **SHIFT + ESC** before pressing <number>.

IBM-PC Mode Key Codes

ATTACHE KEY	BASE	SHIFT	CTRL	CTRL SHIFT (ALT)	(CTRL + CAPSLOCK) 10-KEY MODE	+ CTRL SHIFT	2 keystrokes CTRL TAB	SHIFT ESC
1	1	!	0,59	0,84			0,94	0,104
2	2	@	0,60	0,85			0,95	0,105
3	3	#	0,61	0,86			0,96	0,106
4	4	$	0,62	0,87			0,97	0,107
5	5	%	0,63	0,88			0,98	0,108
6	6	&	0,64	0,89			0,99	0,109
7	7	*	0,65	0,90	7	0,126	0,100	0,110
8	8	(	0,66	0,91	8	0,127	0,101	0,111
9	9	)	0,67	0,92	9	0,128	0,102	0,112
0	0	^	0,68	0,93			0,103	0,113
A	a	A	SOH	0,30				
B	b	B	STX	0,48				
C	c	C	ETX	0,46				
D	d	D	EOT	0,32				
E	e	E	ENQ	0,18				
F	f	F	ACK	0,33				
G	g	G	BEL	0,34				
H	h	H	BS	0,35				
I	i	I	HT	0,23	5	0,124		
J	j	J	LF	0,36	1	0,120		
K	k	K	VT	0,37	2	0,121		
L	l	L	FF	0,38	3	0,122		
M	m	M	CR	0,50	0	0,129		
N	n	N	SO	0,49				
O	o	O	SI	0,24	6	0,125		
P	p	P	DLE	0,25	+			
Q	q	Q	DC1	0,16				
R	r	R	DC2	0,19				
S	s	S	DC3	0,31				
T	t	T	DC4	0,20				
U	u	U	NAK	0,22	4	0,123		
V	v	V	SYN	0,47				
W	w	W	ETB	0,17				
X	x	X	CAN	0,45				
Y	y	Y	EM	0,21				
Z	z	Z	SUB	0,44				

IBM-PC Mode (continued)

ATTACHE KEY	BASE	SHIFT	CTRL	CTRL SHIFT (ALT)	(CTRL + CAPSLOCK) 10-KEY MODE	+ CTRL SHIFT	2 keystrokes CTRL TAB	SHIFT ESC
-	-	_	US	US	-	0,130		
=	=	+	0,71	0,119	=	0,131		
[	[	{	GS	GS				
]	]	}	ESC	ESC				
;	;	:	;	:	*			
'	'	"	0,79	0,117				
,	,	<	,	<				
.	.	>			.			
/	/	?	/	?	/			
`	`	~		~				
\	\	|	FS	FS				
ESC	ESC		VALET	ESC				
BS	BS	BS	DEL	DEL				
TAB	HT	0,15		HT				
LF	LF	LF	BREAK	SCROLL				
CAPS								
CR	CR	CR	CR	CR				
DEL	0,83	0,82	0,82	DEL				
LEFT	0,75	0,75	0,115	0,75				
DOWN	0,80	0,81	0,81	0,118				
UP	0,72	0,72	0,73	0,132				
RIGHT	0,77	0,77	0,116	0,77				
SPACE	SP	SP	SP	SP				

IBM-PC Keyboard Conversion Table

Special Editing Keys

Function	IBM Key	8:16 Key	Description
Copy one character	F1	**CTRL 1**	Copies one character from the template to the new line.
Copy up to a character	F2	**CTRL 2**	Copies all characters from the template to the new line up to the specified character.
Copy template	F3	**CTRL 3**	Copies all remaining characters in the template to the screen.
Skip one character	DEL	**DEL**	Does not copy (skips over) a character.
Skip up to a character	F4	**CTRL 4**	Does not copy (skips over) the characters in the template, up to the specified character.
Quit Input	ESC	**CTRL x**	Voids current input. Template is unchanged.
Insert	INS	**CTRL DEL**	Turns insert on or off.
Replace	INS	**CTRL DEL**	Turns insert mode off.
New template	F5	**CTRL 5**	Makes the new line the new template.

IBM-PC Keyboard Conversion Table: Common Functions (continued)

Function	IBM Key	8:16 Key	Description
Cold boot	CTRL ALT DEL	**right SHIFT RESET**	Reloads the operating system.
Input	return arrow	**RETURN**	Inputs commands.
Stop command	CTRL SCROLL LOCK	**CTRL C**	Halts the execution of a command.
Stop	CTRL NUM LOCK	**CTRL S**	Freezes the display. **CTRL S** again restarts.
Print screen	SHIFT PrtSc	**CTRL SHIFT right arrow**	Prints the entire screen as currently displayed.
Print as displayed	CTRL PrtSc	**CTRL P** or **CTRL N**	Information prints as it displays. **CTRL N** stops print action.
Cancel command	ESC	**CTRL X**	Removes the current input line for corrections.
Insert mode	INS	**CTRL DEL**	Enters and exits insert mode.
Tab	arrows	**TAB**	Moves the cursor to the next tab stop.

IBM-PC Keyboard Conversion Table: Common Function Keys

Function	IBM Key	8:16 Key	Description
Shift	shift arrow	**SHIFT**	Shifts keys to upper case.
Backspace	backspace arrow	**BACK SPACE**	Removes the character left of the cursor.

Programming Keys

Function	IBM Key	8:16 Key	Description
KEY	F9	**CTRL 9**	Changes the function of other function keys.
SCREEN	F10	**CTRL 0**	Returns to character mode from graphics mode.
Numeric keypad	NUM LOCK	**CTRL CAPS LOCK**	Turns the numeric keypad on or off.
Home cursor	HOME	**CTRL =**	Moves the cursor to the first character on the top line.
Cursor to line end	END	**CTRL '**	Moves the cursor to the end of current line.

IBM-PC Keyboard Conversion Table: Programming Keys (continued)

Function	IBM Key	8:16 Key	Description
Clear screen	CTRL HOME	**CTRL SHIFT =**	Clears the screen.
Alternate Functions	ALT A-Z	**CTRL SHIFT A to Z**	Can assign keys a function.
Alternate Functions	ALT 1-0	**CTRL CAPS LOCK; then CTRL SHIFT** decimal value on keypad	User-assigned alternate key function.
Suspend system	CTRL NUM LOCK	**CTRL SHIFT CAPS LOCK**	Stops all execution. Press any key to restart.
Page up	PG UP	**CTRL up arrow**	Scrolls back 25 lines and homes the cursor.
Top of file	CTRL PG UP	**CTRL SHIFT up arrow**	Moves cursor to the top of the current file.
Page down	PG DN	**CTRL down arrow**	Scrolls forward 25 lines and homes cursor.
Delete to page end	CTRL PG DN	**CTRL SHIFT down arrow**	Delete from cursor to the end of the screen.
Delete to line end	CTRL END	**CTRL SHIFT '**	Deletes from the cursor to end of current line.

IBM-PC Keyboard Conversion Table: Programming Keys (continued)

Function	IBM Key	8:16 Key	Description
Advance word	CTRL right arrow	**CTRL right arrow**	Moves cursor forward one word.
Return word	CTRL left arrow	**CTRL left arrow**	Moves cursor back one one word.
Delete	DEL	**DEL**	Backspaces and removes the character.
Secondary Function	SHIFT F1 to F10	**CTRL SHIFT 1 to 0**	Additional function keys.
Secondary Function	CTRL F1 to F10	**CTRL TAB, 1 to 0**	Additional function keys.
Secondary Function	ALT F1 to F10	**SHIFT ESC, 1 to 0**	Additional function keys.
Bell	CTRL G	**CTRL ESC 0**	Turns bell on.
Line feed	CTRL enter	**LINE FEED**	Feeds one line to the printer.

Interrupts and Function Requests

The following charts provide a quick reference for locating interrupts and function requests. XENIX-compatible calls, interrupts, and MS-DOS function requests appear below.

XENIX-Compatible Calls

Function Code	Description
Function 39H	Create Subdirectory
Function 3AH	Remove a Directory Entry
Function 3BH	Change the Current Directory
Function 3CH	Create a File
Function 3DH	Open a File
Function 3FH	Read From File/Device
Function 40H	Write to a File or Device
Function 41H	Delete a Directory Entry
Function 42H	Move a File Pointer
Function 43H	Change Attributes
Function 44H	I/O Control for Devices
Function 45H	Duplicate a File Handle
Function 46H	Force a Duplicate of a Handle
Function 4BH	Load and Execute a Program
Function 4CH	Terminate a Process
Function 4DH	Retrieve Return Code of "Child"

INTERRUPTS (Numeric Order)

Interrupt Hex	Interrupt Dec	Description
0H	0	Divide by Zero
1H	1	Single Step
2H	2	Non-Maskable Interrupt
3H	3	Breakpoint
4H	4	Overflow
5H	5	Print Screen
6H,7H	6,7	RESERVED
8H	8	Time of Day
9H	9	Keyboard (RESERVED) (9H not implemented)
AH	10	RESERVED
BH	11	Communications
CH	12	Communications
DH	13	Disk
10H	16	Video I/O
11H	17	Equipment Check
12H	18	Memory
13H	19	Diskette I/O
14H	20	Communications Port I/O
15H	21	Cassette (RESERVED) (15H not implemented)
16H	22	Keyboard I/O
17H	23	Printer I/O
18H	24	ROM BASIC (RESERVED) (18H not implemented)
19H	25	Bootstrap (RESERVED) (19H not implemented)
1AH	26	Time of Day
1BH	27	Keyboard Break
1CH	28	Timer Tick
1DH	29	Video Initialize (RESERVED) (1DH not implemented)
1EH	30	Diskette Parameters
1FH	31	Pointer to Video Characters (1FH not implemented)
20H	32	Program Terminate
21H	33	Function Request
22H	34	Terminate Address
23H	35	CTRL C Exit Address
24H	36	Fatal Error Abort Address
25H	37	Absolute Disk Read
26H	38	Absolute Disk Write
27H	39	Terminate But Stay Resident
28-3FH	40-63	RESERVED -- DO NOT USE

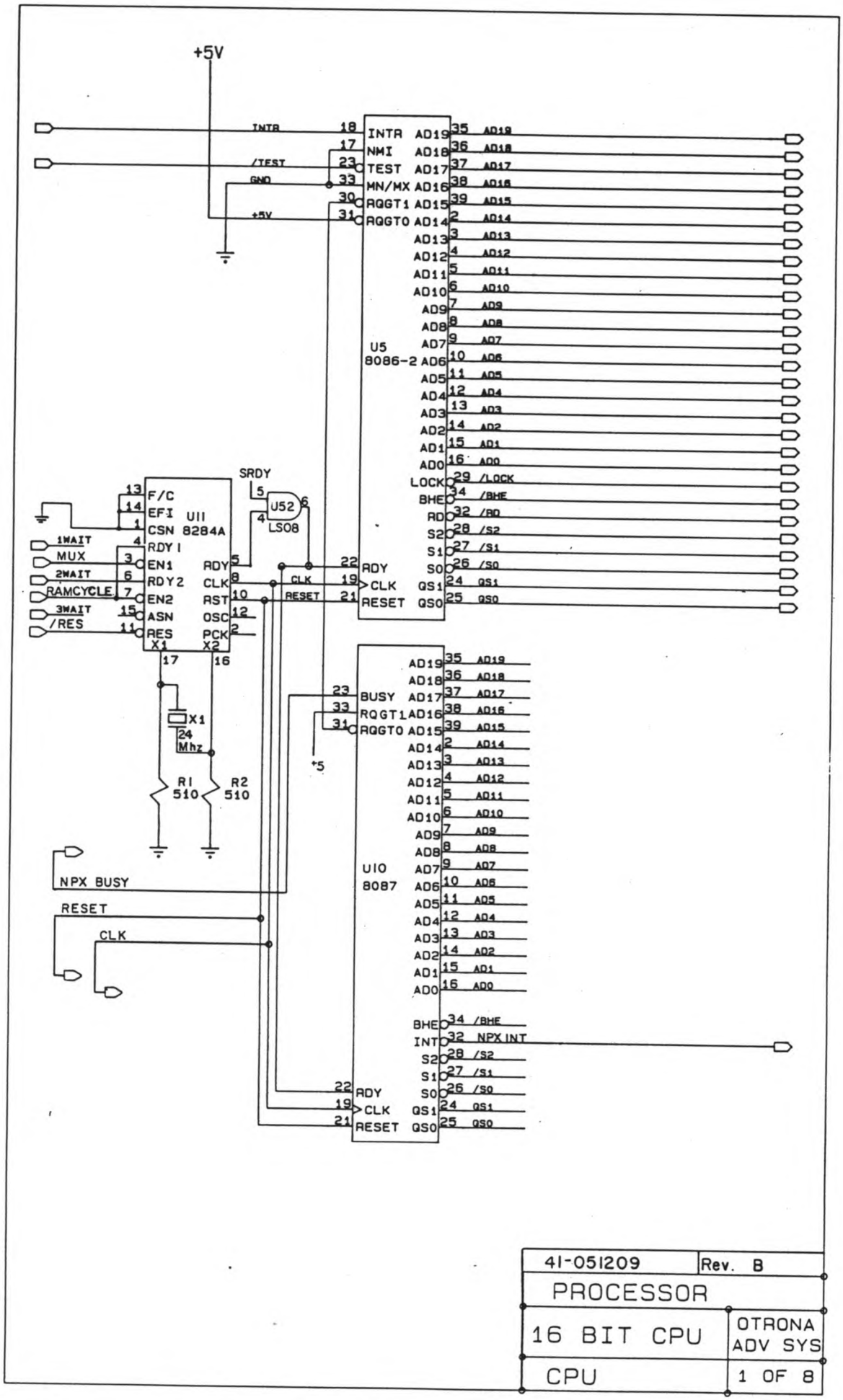
MS-DOS FUNCTION REQUESTS (Numeric Order)

Function Code	Function Name
00H	Terminate Program
01H	Read Keyboard and Echo
02H	Display Character
03H	Auxiliary Input
04H	Auxiliary Output
05H	Print Character
06H	Direct Console I/O
07H	Direct Console Input
08H	Read Keyboard
09H	Display String
0AH	Buffered Keyboard Input
0BH	Check Keyboard Status
0CH	Flush Buffer, Read Keyboard
0DH	Disk Reset
0EH	Select Disk
0FH	Open File
10H	Close File
11H	Search for First Entry
12H	Search for Next Entry
13H	Delete File
14H	Sequential Read
15H	Sequential Write
16H	Create File
17H	Rename File
18H	RESERVED
19H	Current Disk
1AH	Set Disk Transfer Address
1BH-20H	RESERVED
21H	Random Read
22H	Random Write
23H	File Size
24H	Set Relative Record
25H	Set Vector
26H	RESERVED
27H	Random Block Read
28H	Random Block Write
29H	Parse File Name
2AH	Get Date
2BH	Set Date
2CH	Get Time
2DH	Set Time
2EH	Set/Reset Verify Flag
2FH	Get Disk Transfer Address

MS-DOS FUNCTION REQUESTS (Continued)

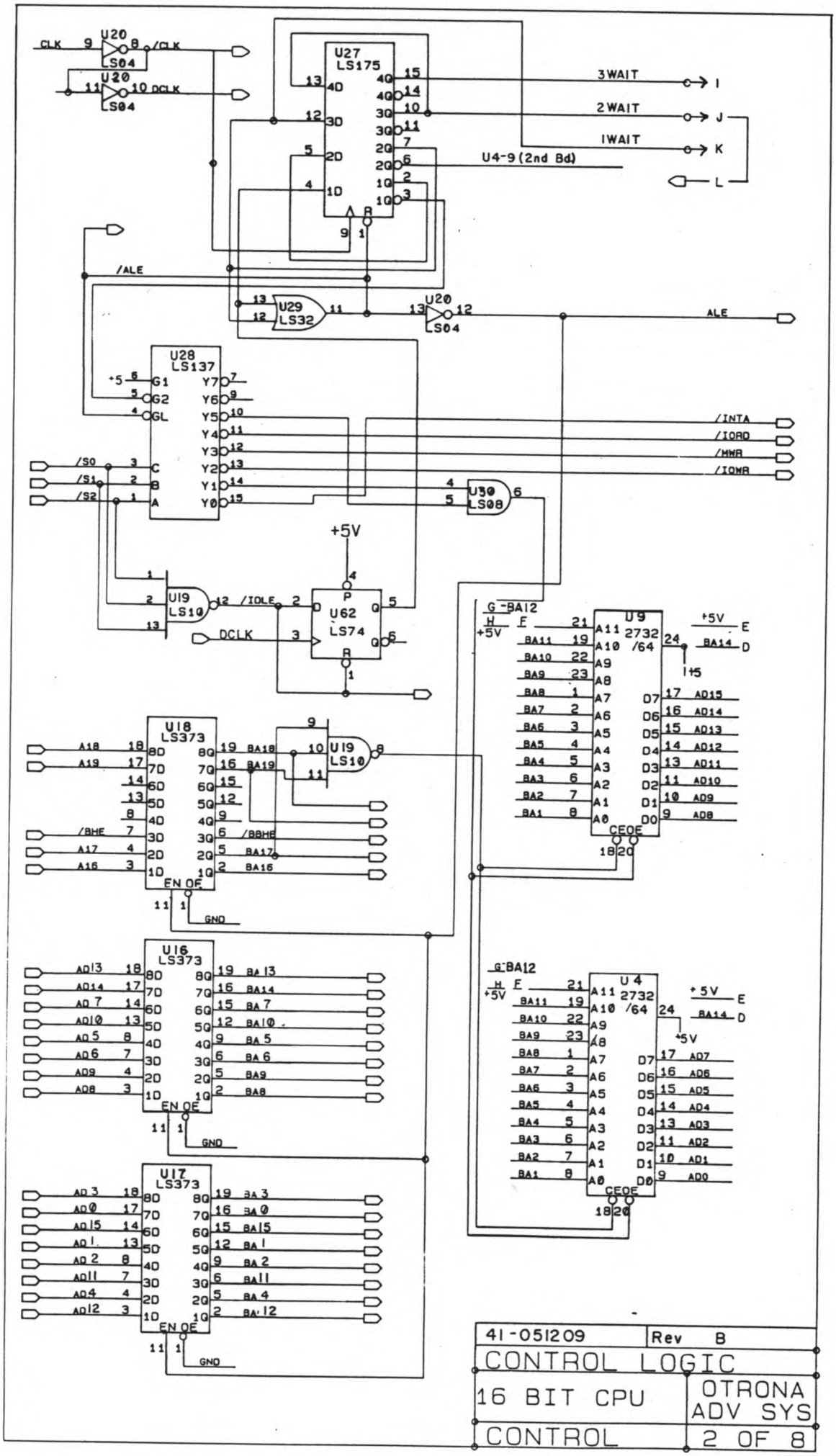
Function Code	Function Name
30H	Get DOS Version Number
31H	Keep Process
32H	RESERVED
33H	CTRL C Check
34H	RESERVED
35H	Get Interrupt Vector
36H	Get Disk Free Space
37H	RESERVED
38H	Return Country-Dependent Information
39H	Create Subdirectory
3AH	Remove a Directory Entry
3BH	Change Current Directory
3CH	Create a File
3DH	Open a File
3EH	Close a File Handle
3FH	Read From File/Device
40H	Write to a File/Device
41H	Delete a Directory Entry
42H	Move a File Pointer
43H	Change Attributes
44H	I/O Control for Devices
45H	Duplicate a File Handle
46H	Force a Duplicate of a Handle
47H	Return Text of Current Directory
48H	Allocate Memory
49H	Free Allocated Memory
4AH	Modify Allocated Memory Blocks
4BH	Load and Execute a Program
4CH	Terminate a Process
4DH	Retrieve the Return Code of a Child
4EH	Find Match File
4FH	Step Through a Directory Matching Files
50H-53H	RESERVED
54H	Return Current Setting of Verify
55H	RESERVED
56H	Move a Directory Entry
57H	Get/Set Date/Time of File

Appendixes

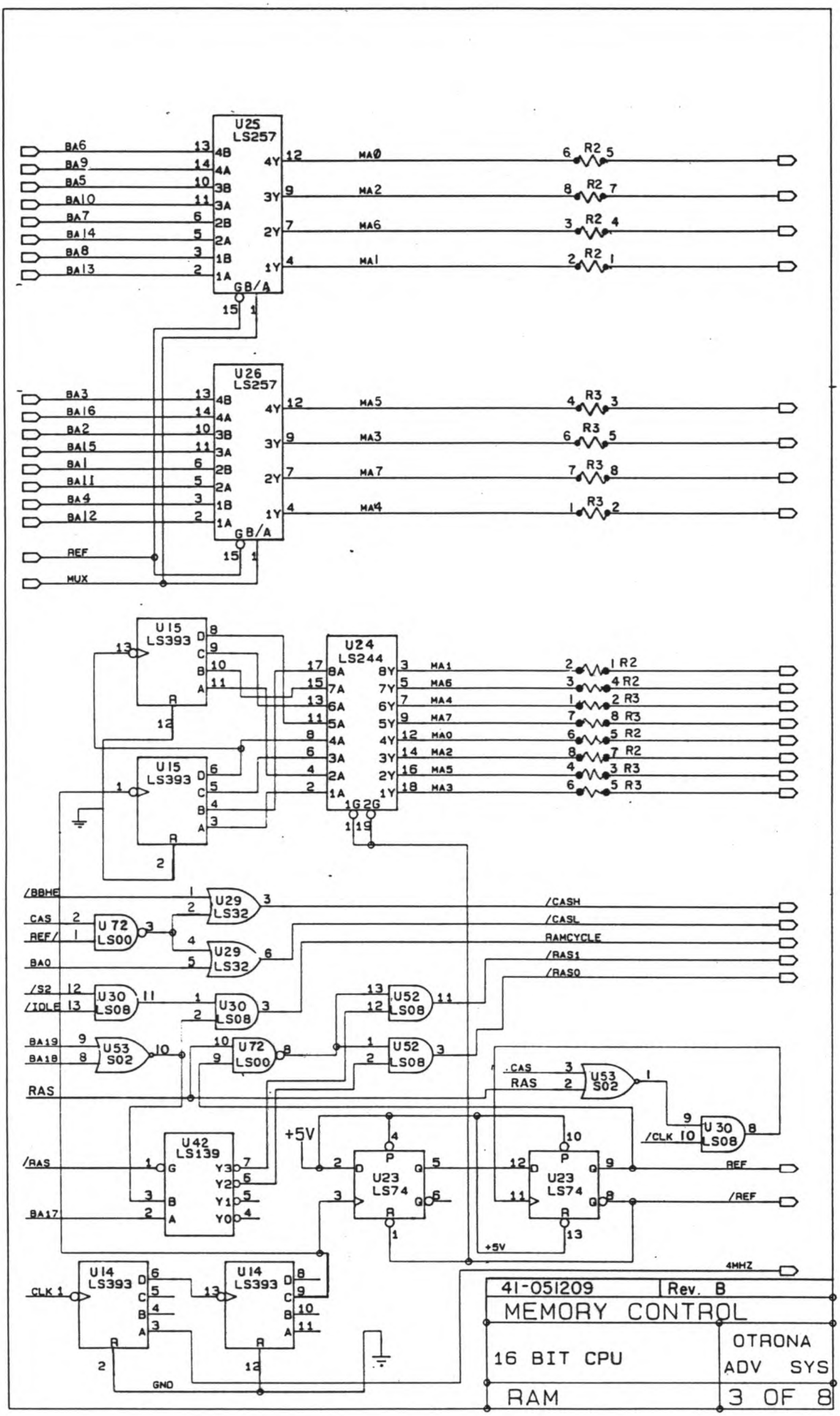


41-051209	Rev. B
PROCESSOR	
16 BIT CPU	OTRONA ADV SYS
CPU	1 OF 8

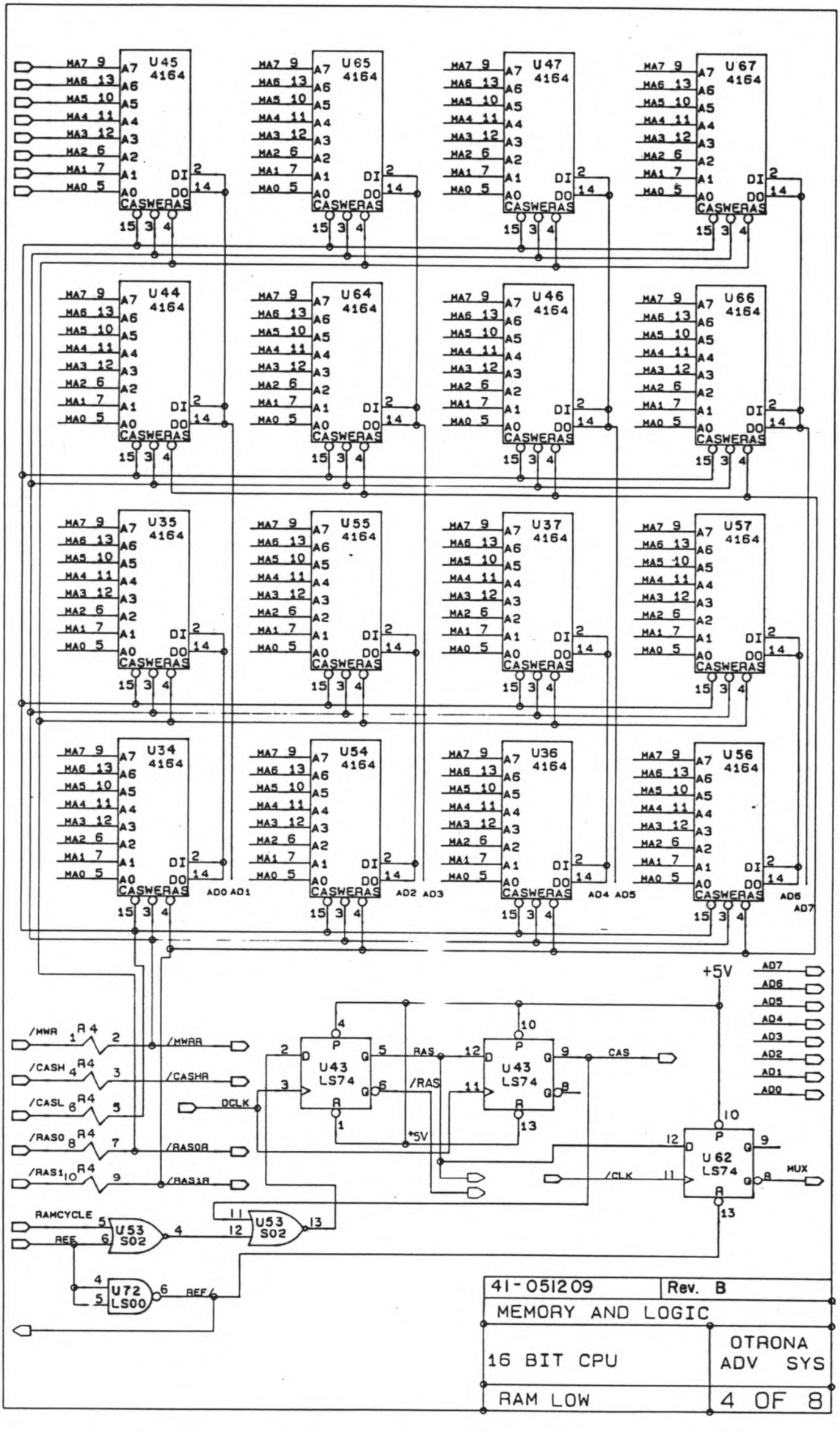
A1


U20 LS04
CLK
/CLK
DCLK
U27 LS175
3WAIT
2WAIT
1WAIT
U4-9 (2nd Bd)
/ALE
U29 LS32
ALE
U28 LS137
/INTA
/IORD
/MWR
/IOWR
/S0
/S1
/S2
U30 LS08
+5V
U19 LS10
/IDLE
U62 LS74
DCLK
U18 LS373
U16 LS373
U17 LS373
U9 2732 /64
U4 2732 /64
BA12
BA14
AD15
AD14
AD13
AD12
AD11
AD10
AD9
AD8
AD7
AD6
AD5
AD4
AD3
AD2
AD1
AD0
/BHE
GND
41-051209
Rev B
CONTROL LOGIC
16 BIT CPU
OTRONA ADV SYS
CONTROL
2 OF 8

A2

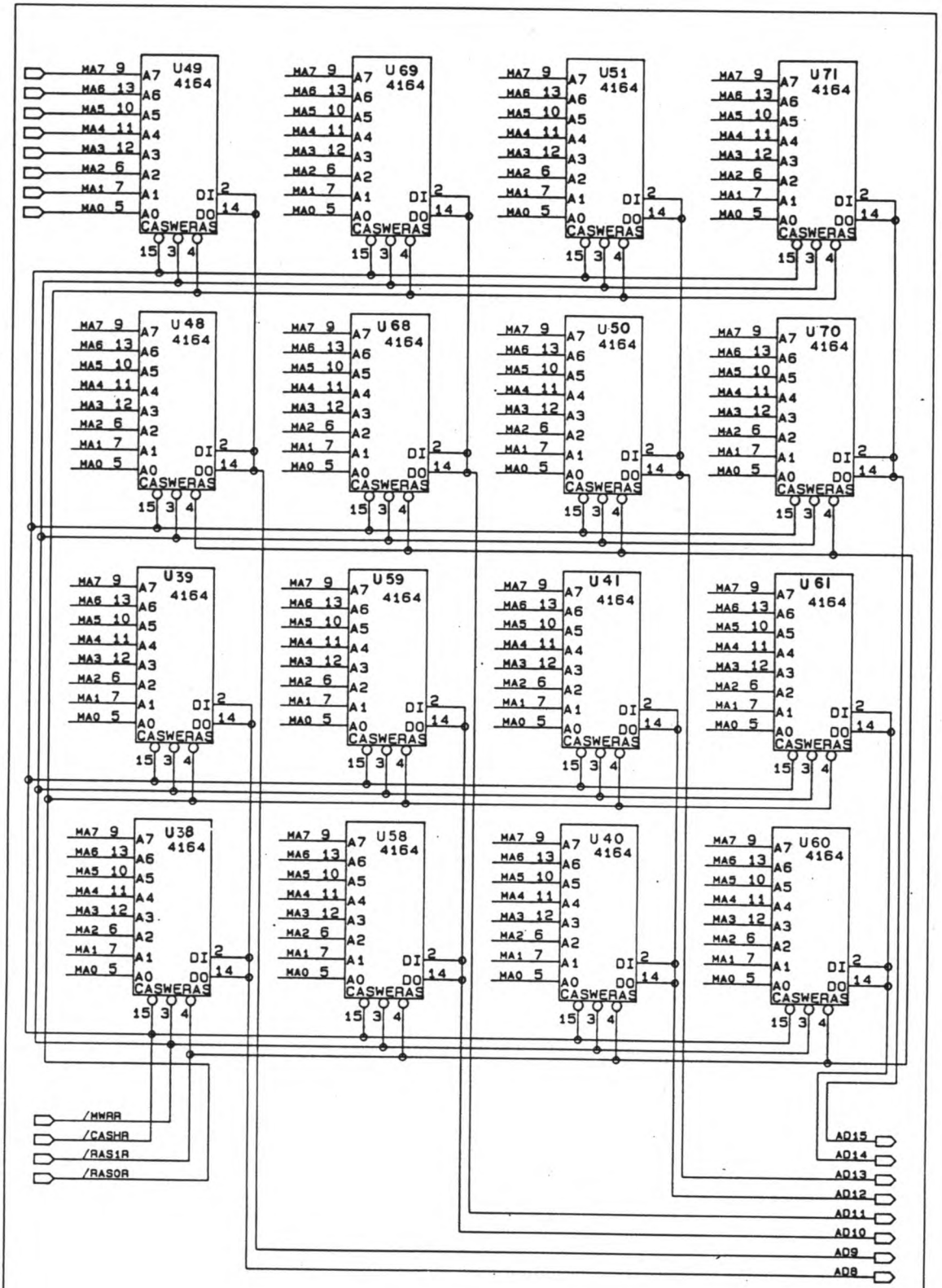

U25 LS257
U26 LS257
U15 LS393
U24 LS244
U29 LS32
U72 LS00
U30 LS08
U52 LS08
U53 S02
U42 LS139
U23 LS74
U14 LS393
BA6
BA9
BA5
BA10
BA7
BA14
BA8
BA13
BA3
BA16
BA2
BA15
BA1
BA11
BA4
BA12
REF
MUX
MA0
MA2
MA6
MA1
MA5
MA3
MA7
MA4
/BBHE
CAS
REF/
BA0
/S2
/IDLE
BA19
BA18
RAS
/CASH
/CASL
RAMCYCLE
/RAS1
/RAS0
.CAS
/CLK 10
/RAS
BA17
+5V
REF
/REF
4MHZ
CLK 1
GND
41-051209
Rev. B
MEMORY CONTROL
16 BIT CPU
OTRONA
ADV SYS
RAM
3 OF 8

A3

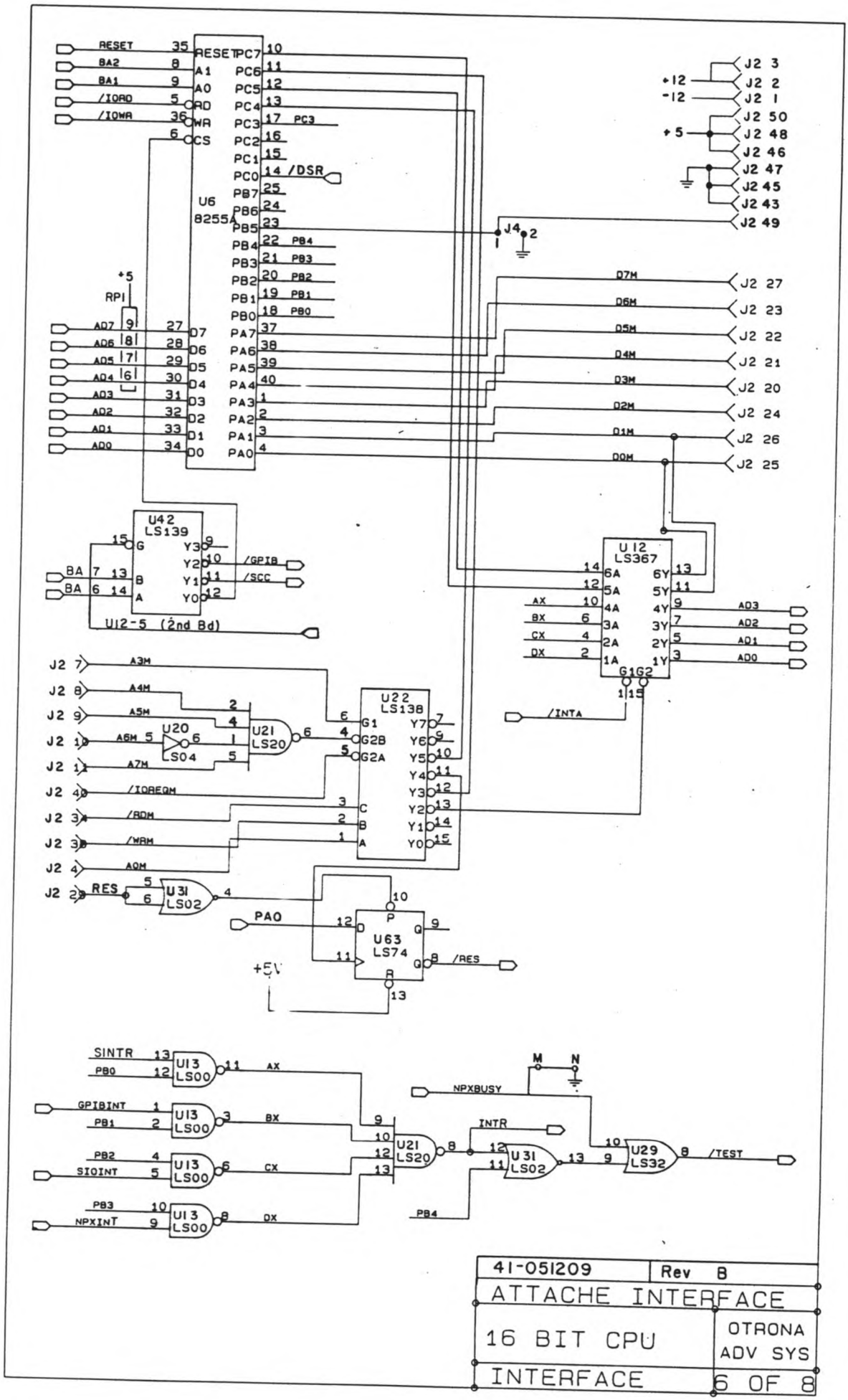


41-051209	Rev. B
MEMORY AND LOGIC	
16 BIT CPU	OTRONA ADV SYS
RAM LOW	4 OF 8

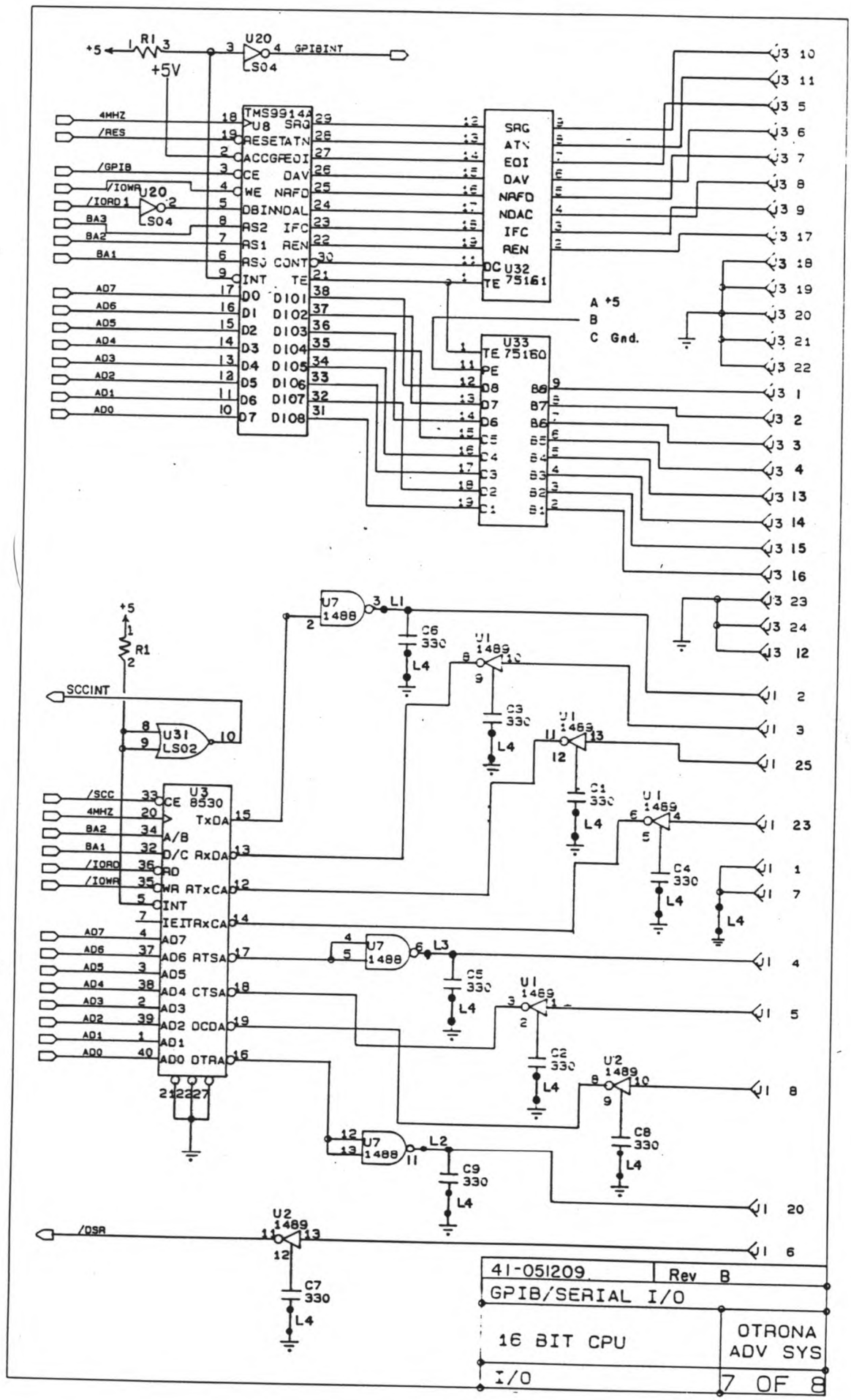
A4

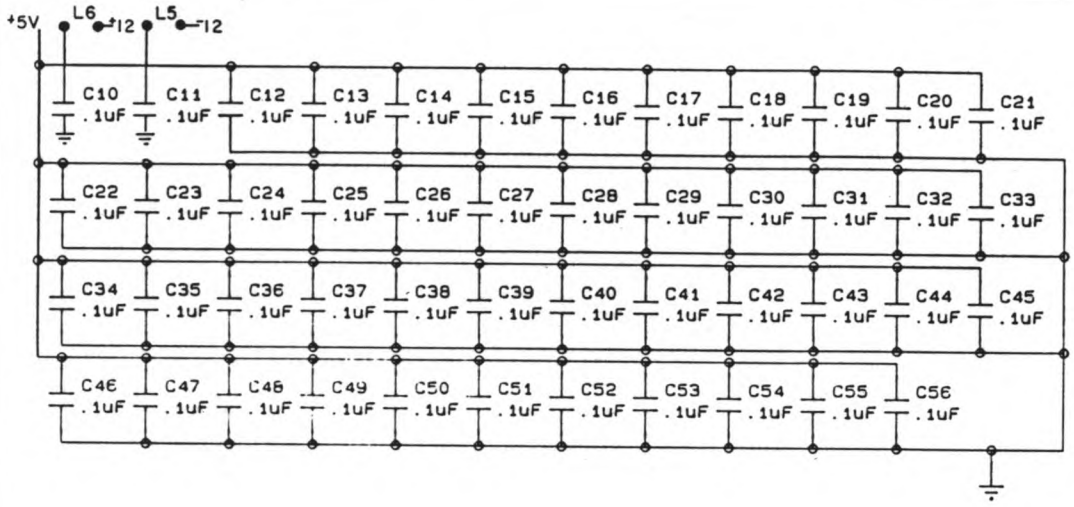


41-051209	Rev. B
RAM HIGH DATA	
16 BIT CPU	OTRONA ADV SYS
RAM HIGH	5 OF 8

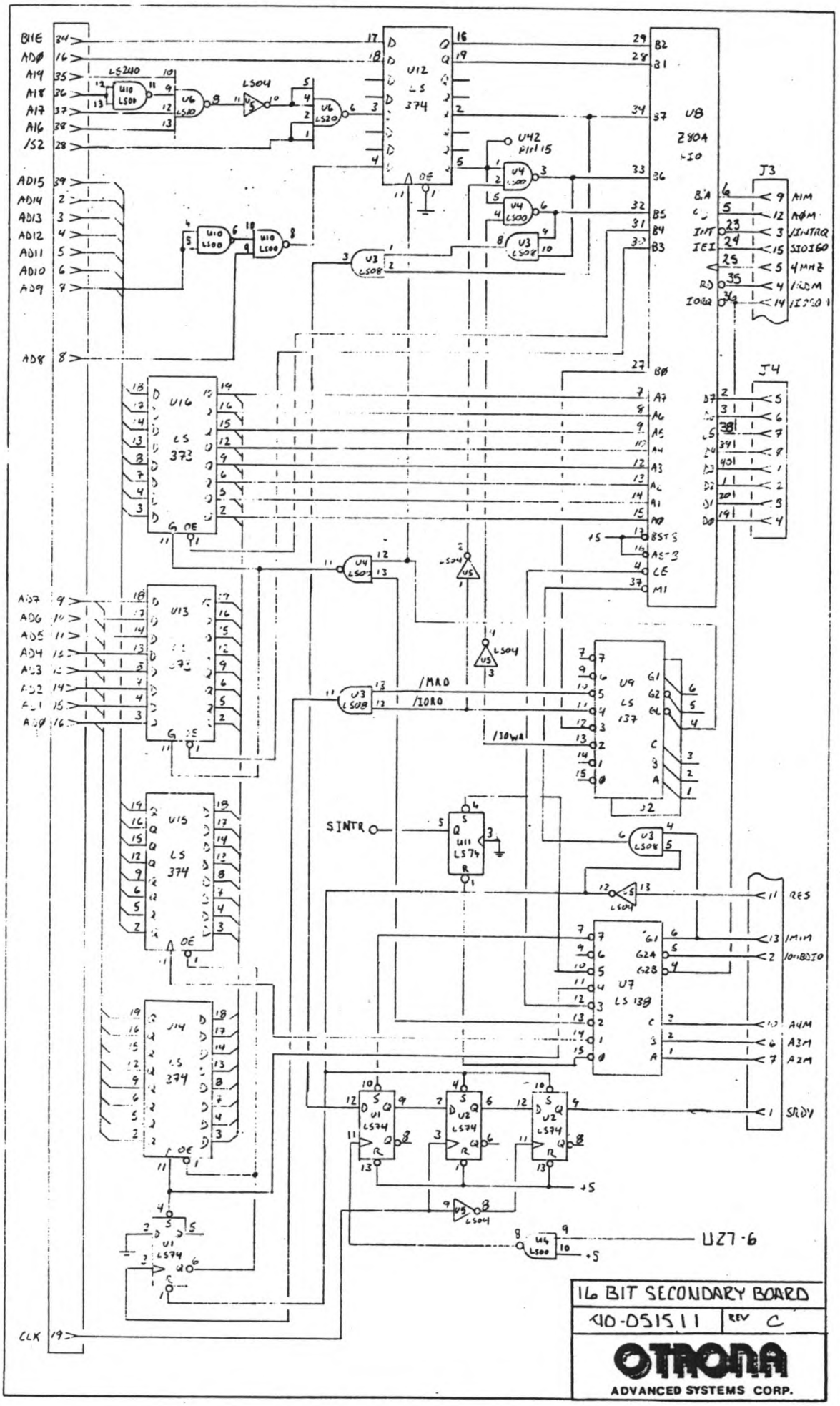


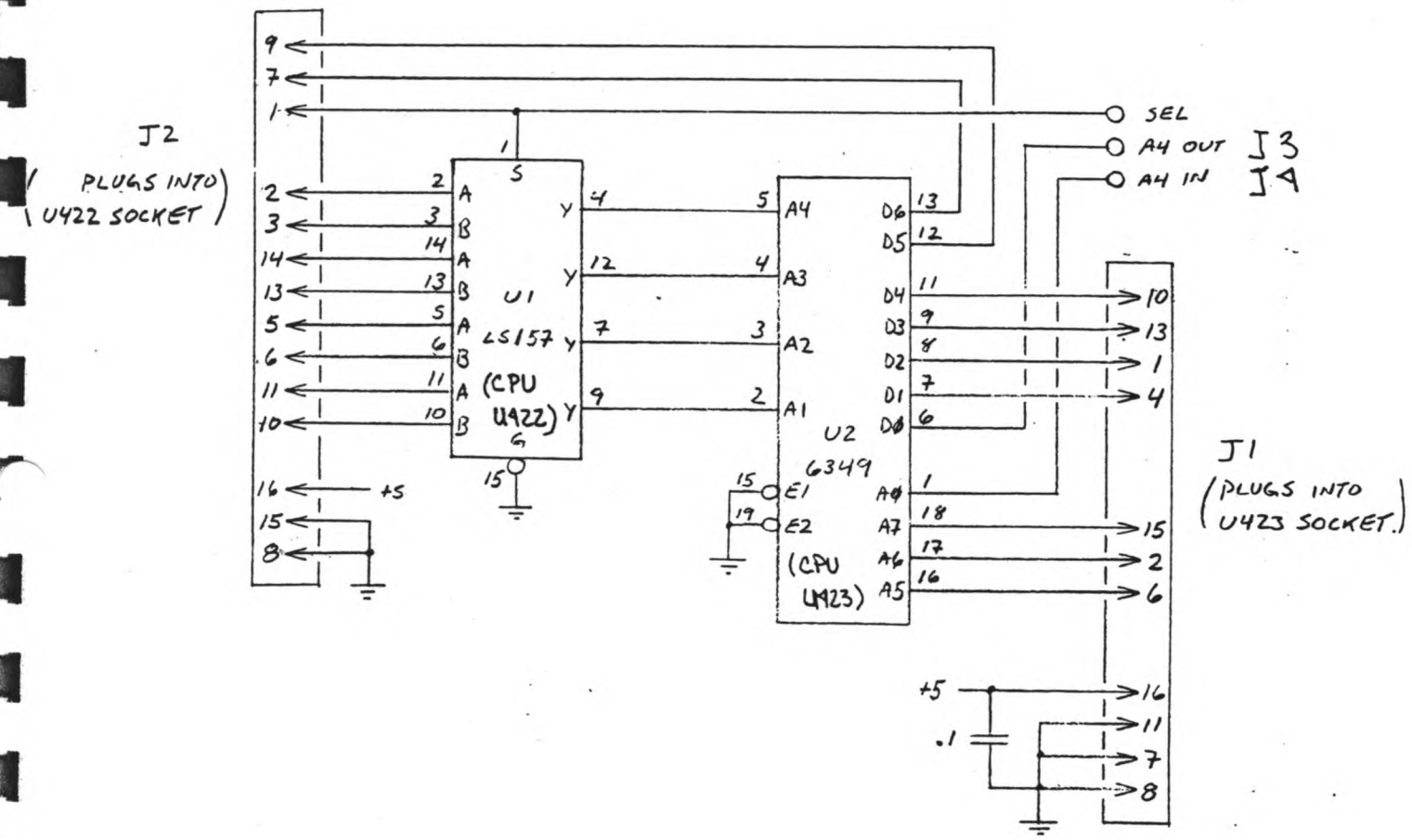
41-051209	Rev B
ATTACHE INTERFACE	
16 BIT CPU	OTRONA ADV SYS
INTERFACE	6 OF 8


+5
R1
+5V
U20
LS04
GPIBINT
TMS9914A
U8
4MHZ
/RES
/GPIB
/IOWR
/IORD
BA3
BA2
BA1
AD7
AD6
AD5
AD4
AD3
AD2
AD1
AD0
SRQ
RESETATN
ACCGREOI
CE DAV
WE NRFD
DBINNDAL
AS2 IFC
AS1 REN
AS0 CONT
INT TE
D0 DIO1
D1 DIO2
D2 DIO3
D3 DIO4
D4 DIO5
D5 DIO6
D6 DIO7
D7 DIO8
U32
75161
SRQ
ATN
EOI
DAV
NRFD
NDAC
IFC
REN
DC
TE
U33
75160
PE
D8
D7
D6
D5
D4
D3
D2
D1
B8
B7
B6
B5
B4
B3
B2
B1
A +5
B
C Gnd.
U3 10
U3 11
U3 5
U3 6
U3 7
U3 8
U3 9
U3 17
U3 18
U3 19
U3 20
U3 21
U3 22
U3 1
U3 2
U3 3
U3 4
U3 13
U3 14
U3 15
U3 16
U3 23
U3 24
U3 12
U7
1488
L1
L2
L3
L4
C6
330
C3
C1
C4
C5
C2
C8
C9
C7
U1
1489
U2
U31
LS02
SCCINT
U3
8530
/SCC
CE
TxDA
A/B
D/C RxDA
RD
WR RTxCA
INT
IEIRxCA
AD6 RTSA
AD4 CTSA
AD2 DCDA
AD0 DTRA
/DSR
U1 2
U1 3
U1 25
U1 23
U1 1
U1 7
U1 4
U1 5
U1 8
U1 20
U1 6
41-051209
Rev B
GPIB/SERIAL I/O
16 BIT CPU
OTRONA
ADV SYS
I/O
7 OF 8

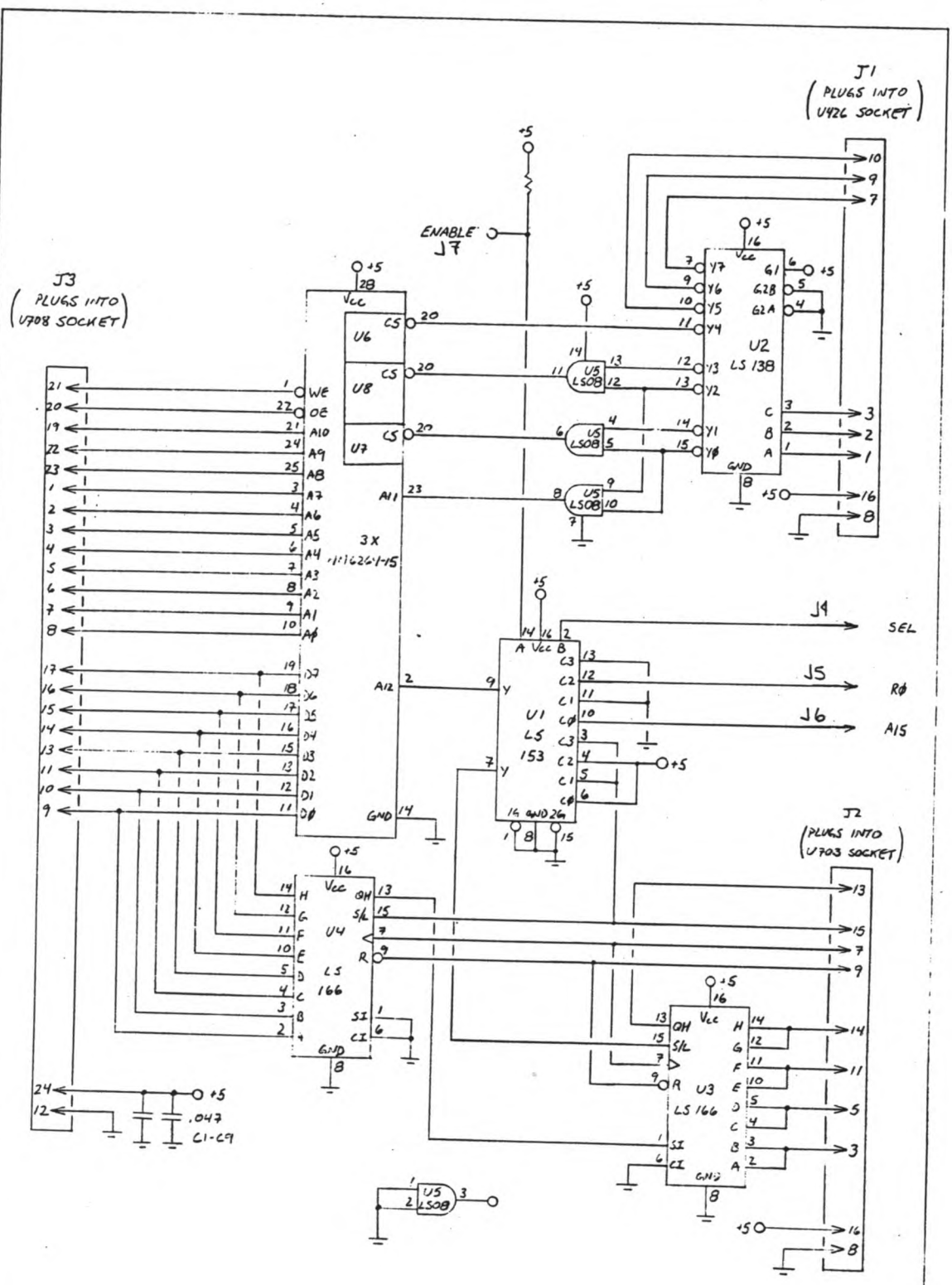


41-051209	Rev B
BY-PASS CAPS	
16 BIT CPU	OTRONA ADV SYS
BY-PASS	8 OF 8


BHE
AD0
A19
A18
A17
A16
/S2
AD15
AD14
AD13
AD12
AD11
AD10
AD9
AD8
AD7
AD6
AD5
AD4
AD3
AD2
AD1
AD0
CLK
LS240
LS04
U12 LS 374
U42 Pin 15
U8 Z80A PIO
J3
J4
U16 LS 373
U13 LS 373
U15 LS 374
U14 LS 374
U9 LS 137
U7 LS 138
U11 LS74
U1 LS74
U2 LS74
/MRD
/IORD
/IOWR
SINTR
RES
/MIM
/IOBDIO
A4M
A3M
A2M
SRDY
+5
UZ7-6
BSTS
MI
CE
16 BIT SECONDARY BOARD
40-051511
REV C
OTRONA
ADVANCED SYSTEMS CORP.



25 - LINE MODIFICATION



HIGH - RESOLUTION GRAPHICS BOARD

Glossary

ALE — Address Latch Enable: The signal which, when true, indicates a valid address is on the bus.

Attache 8:16 — An Attache which contains an 8086 16-bit board and secondary board, in addition to the Z-80A main processor board.

Attache 8:16A — An Attache which contains an 8087 16-bit board and an 8087 coprocessor, in addition to the Z-80A main processor board.

AUTOEXEC — An MS-DOS command which allows user-created batch files to execute automatically during cold boots.

BHE — Bus High Enable: A signal used with A0 to indicate whether data on the bus is a high-byte, low-byte, or word.

CP/M — Control Program for Microcomputers: An 8-bit control program written by Digital Research which Attache 8:16 uses when the Z80 is the active processor.

CTS — Clear To Send: A communications protocol which allows I/O to transpire whenever the CTS signal is true.

GPIB — General Purpose Interface Bus: A communications protocol established by the IEEE-488 standard for interface with electronic instruments.

IBM-PC — International Business Machines Personal Computer.

IO.SYS — A hidden file on the MS-DOS System Diskette which contains the 8086 and Z80 BIOS code.

Master/Slave — An interface protocol in which one processor has complete control over tasks for the second processor to perform.

MS-DOS — Microsoft Disk Operating System: A 16-bit control program written by Microsoft which Attache 8:16 uses when the 8086 is the active processor.

SCC — Synchronous Communications Controller: An option on the Attache 8:16 that controls the synchronous communications on a serial port.

Synchronous	Having the same phase and frequency. Synchronous communications is data transfer with characters at the beginning of each message synchronized between the devices.
Trapping	The process of intercepting a processor function.
XON/XOFF	A communications protocol which uses start and stop bytes sent by the peripheral to indicate when it is able to accept data.
8-bit	An operating mode which transfers data eight bits at a time.
16-bit	An operating mode which transfers data 16 bits at a time.
25-Line	A modification to Attache's display circuitry that changes the display from 24 lines to 25 lines, compatible with the IBM-PC.
8086	The main processor located on the 16-bit board; An Intel processor which transfers data 16 bits at a time.
8087	An Intel numeric coprocessor.
8255A	A programmable peripheral interface port with 24 bidirectional lines.

Index
