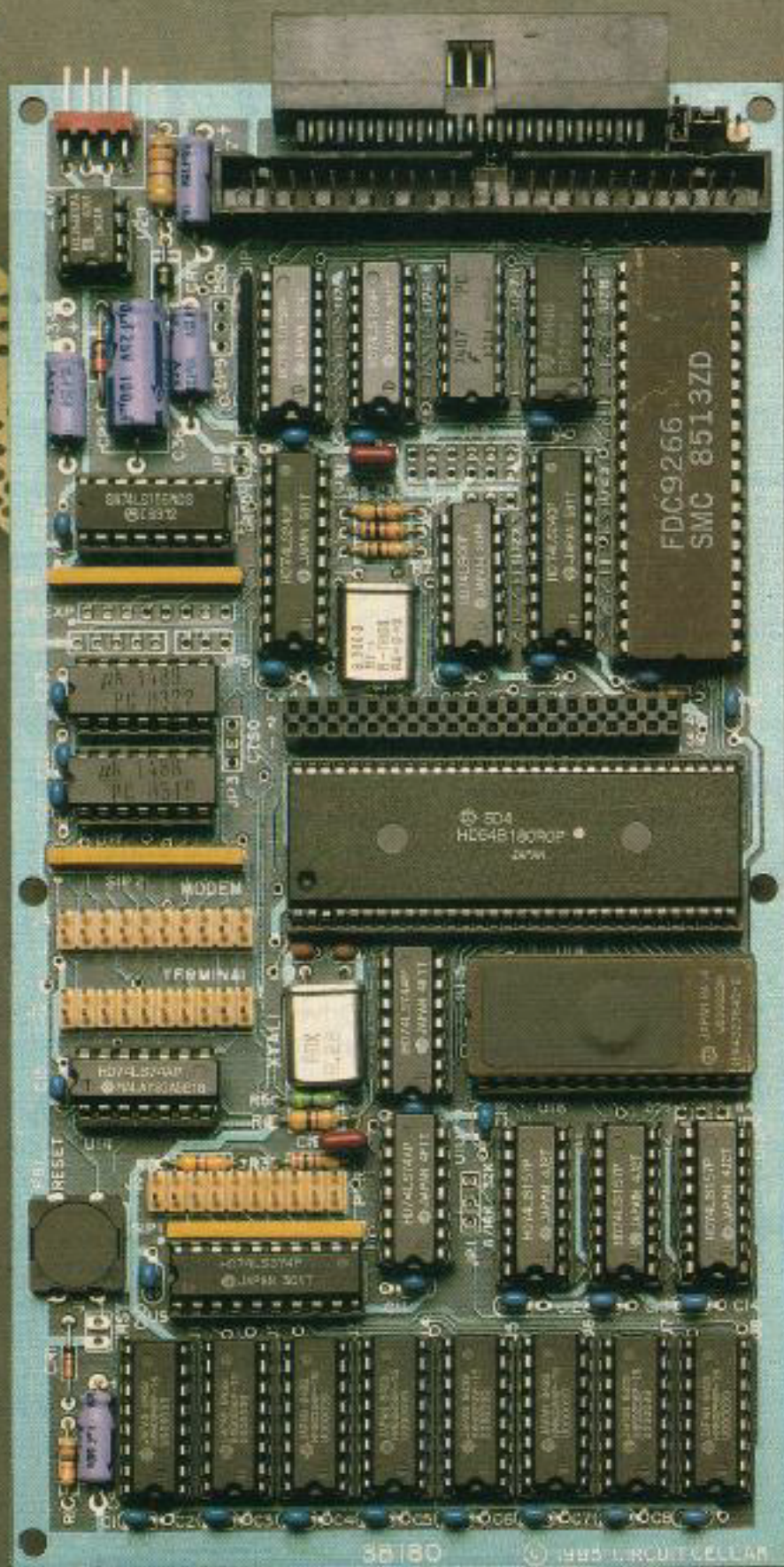


Photo 1: The SBI80 single-board computer. Installed connectors clockwise from the lower left are the parallel printer port, console serial port, auxiliary serial port, power supply, 8-inch disk-drive connector, and 3 1/2- and 5 1/4-inch disk-drive connectors. The 256K bytes of DRAM are contained in the single row of eight chips along the bottom, the HD64180 CPU is in the center, and the floppy-disk-controller chip is on the top right.



BUILD THE SB180 SINGLE-BOARD COMPUTER

PART 1: THE HARDWARE

BY STEVE CIARCIA

*This computer reasserts 8-bit computing
in a 16-bit world*



Newer, faster, better. These words are screamed at you in ads and reviews of virtually every new computer that comes to market. Unfortunately, many of the proponents of this rhetoric are going on hearsay evidence. While advertising hype has its place in our culture, a more thorough investigation may lead you to alternative conclusions.

Generally speaking, quotes of increased performance are basically comparisons of CPU (central processing unit) instruction times rarely involving the operating system. The 68000 is indeed a more capable processor than the 6502, but that doesn't necessarily mean that commercial application programs always run faster because the CPU has more capability. People owning 128K-byte Macintoshes have discovered this.

The bus size of the processor is only one factor in the performance of a computer system. Operating-system design and programming styles contribute much more to the overall throughput of a computer. It is not enough to simply compare 8 to 16 bits or 16 to 32 bits. For example, the Sieve of Eratosthenes prime-number benchmark runs faster in BASIC on the 8-bit 8052-based controller board presented in last month's

Circuit Cellar than it does on a 16-bit IBM PC.

The ultimate performance of a computer is the sum of its subsystem interaction times and not just the CPU execution speed. Using a simple database-management program involves interaction among the user input device, operating system, disk drives, firmware, system memory, and user output devices. Slow communication or a bottleneck between any two of these subsystems can degrade the performance of the entire computer.

In my opinion, the processor/operating system connection contributes most to user satisfaction. In the days before the IBM PC, the de facto computer standard was the 8080/Z80 and CP/M 2.2. Unfortunately, software developers considered it difficult to use, especially in turnkey applications. The frustration of having to warm-boot the system merely to change disks and the lack of a PATH command created many ready-and-willing PC-DOS customers. If only there were an 8-bit operating system with the capabilities and friendliness of MS-DOS.

(continued)

Steve Ciarcia (pronounced "see-ARE-see-ah") is an electronics engineer and computer consultant with experience in process control, digital design, nuclear instrumentation, and product development. He is the author of several books about electronics. You can write to him at POB 582, Glastonbury, CT 06033.

The Z80 is still considered a high-performance CPU. In reality, the 8-bit processor is a cost-effective and efficient choice for personal computers and industrial controllers. (Remember that peripheral support chips such as PIAs [peripheral interface adapters] and floppy-disk controllers are 8-bit devices, as are many memory chips.) One problem is that the Z80 has an address limitation of 64K bytes, which discourages the use of 50K-byte BASIC interpreters and 100K-byte integrated spreadsheet programs common to IBM PC users. While additional memory has been added through hardware bank switching, it has never been properly integrated into the CP/M operating system, and its function is kludgy. If only someone would design a Z80 chip that directly addresses more than 64K bytes.

THE CIRCUIT CELLAR SB180

Hitachi has recently developed a CMOS (complementary metal-oxide semiconductor) Z80-code-compatible processor, designated the HD64180, that directly addresses 512K bytes. Echelon Systems has developed an operating system for the processor that is an amalgam of CP/M, MS-DOS,

and UNIX. This operating system is called the Z-System. Using the HD64180 and the Z-System, I have produced a computer that reasserts 8-bit computing in a 16-bit world and outperforms a standard IBM PC by 20 to 100 percent.

In this two-part article, I take a look at the Hitachi HD64180 and the evolution of CP/M as embodied in the Z-System. The hardware project is a single-board computer (4 by 7½ inches) called the SB180, which is the 29-chip equivalent of a large S-100 system (see photos 1 and 2). Refer to figure 1 for a block diagram of the SB180 board.

The Circuit Cellar SB180 has the following capabilities:

- 256K-byte on-board RAM (random-access read/write memory), which can be partitioned as 64K-byte system memory and 192K-byte RAM disk or as paged system memory
- 32K-byte EPROM (erasable programmable read-only memory)
- full ROM-based monitor with system-disk boot
- two RS-232C serial ports, one with automatic data-transmission-rate detection

- a Centronics parallel printer port
- single/double-density programmable floppy-disk controller capable of handling a mix of up to four 3½-, 5¼-, or 8-inch drives; different-size drives can run concurrently
- requires only +5 volts (V) and +12 V for operation (+12 V only for RS-232C)
- I/O (input/output) expansion bus
- fits on a 4- by 7½-inch board that mounts directly to a 5¼- or 3½-inch floppy-disk drive

Disk-based software includes the Z-System DOS (disk operating system), a debugger, and HD64180 assembler.

While this is in fact a hardware project, true functionality would be an exercise left to you readers were it not for the operating system and BIOS (basic input/output system) specifically written and adapted to the SB180. The combination of the HD64180 and Z-System is what gives this tiny computer so much power. Much of the project description therefore has to be the software that uniquely sets performance levels far above traditional 8-bit computer designs. I don't want to diminish the significance of the hardware, but I am a realist.

This month, I'd like to describe the HD64180 chip and give an overview of the changing evolution of CP/M as it pertains to this project. After that, I'll describe the design of the SB180. Next month will be dedicated to the operating-system software and BIOS.

CP/M AND BEYOND

Anyone who has been involved with microcomputers for more than two years acknowledges the tremendous impact of CP/M upon the history of personal computing. While hobbyists were still debating whether the "standard" tape format should be Kansas City CUTS (cassette user tape system) or Tarbell, Gary Kildall made CP/M available at a reasonable price. It quickly became the de facto standard DOS for 8080- and Z80-based microcomputers. With a "standard" operating system and a "standard" 8-inch disk format, entrepreneurial program-

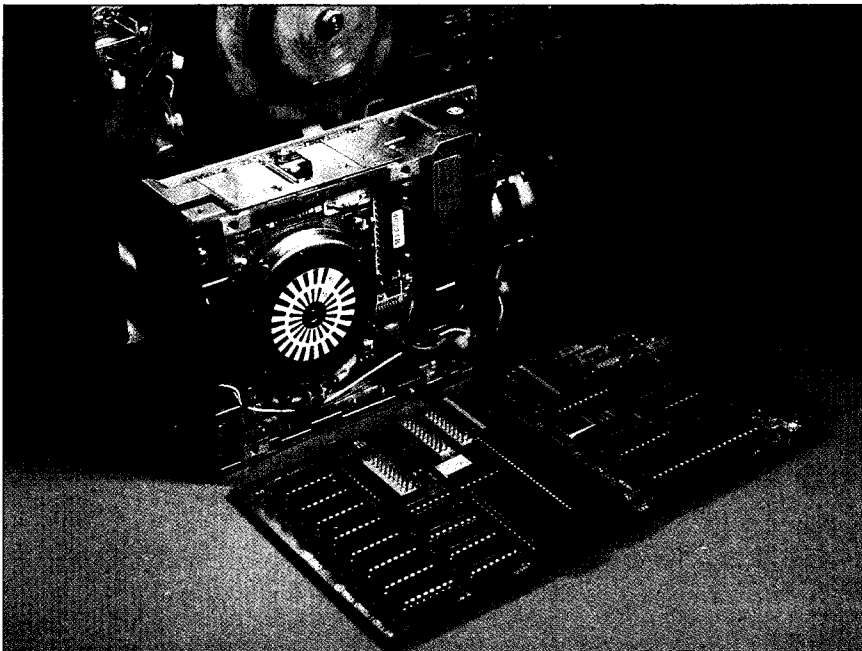


Photo 2: The SB180 shown beside a 3½-inch disk drive and an 8-inch disk drive.

mers saw the opportunity to write serious business applications that needed disk capability to be viable and that could be sold to thousands of CP/M machine owners.

The first major commercial release of CP/M was version 1.4. In an effort to fix some bugs and improve its file-handling capabilities and limits, it was upgraded first to version 2.0 and quickly to version 2.2. Version 2.2 has been a stable product that is familiar to most readers of BYTE. Version 3.0, or CP/M Plus, has been available for about two years, but it hasn't matched the popularity of version 2.2. While CP/M Plus does offer some advanced features, it is not significantly better than version 2.2.

CP/M has many quirks and shortcomings. Having to warm-boot the system when changing disks is a major inconvenience, and named directories would be more convenient than trying to remember disk designators and user-area numbers. Almost no file or system security is provided. CP/M 2.2 does not support redirection of I/O devices, as MS-DOS and UNIX do. Even the MS-DOS batch facility beats CP/M's more primitive Submit and XSUB two ways from Sunday; CP/M lacks conditional testing at the command level.

Many system integrators decry CP/M's lack of a good menu utility to integrate stand-alone, executable programs. Command-line editing is primitive, and multiple commands on one line would be appreciated; creating a single-name "alias" for multiple commands would be even better. Since CP/M has no search path (à la PC-DOS and UNIX), you must either keep multiple copies of often-used programs in many different user areas (wasting disk space) or patch CP/M with a software kludge to make user area 0 accessible from all user areas (even that doesn't help if you need to access a file from user area 3 while you are in user area 7). Many CP/M users supplement their system utilities with more useful public-domain programs like XDIR, DU, CRC, Help, Unerase, and Diff. It's too bad that CP/M wasn't upgraded more

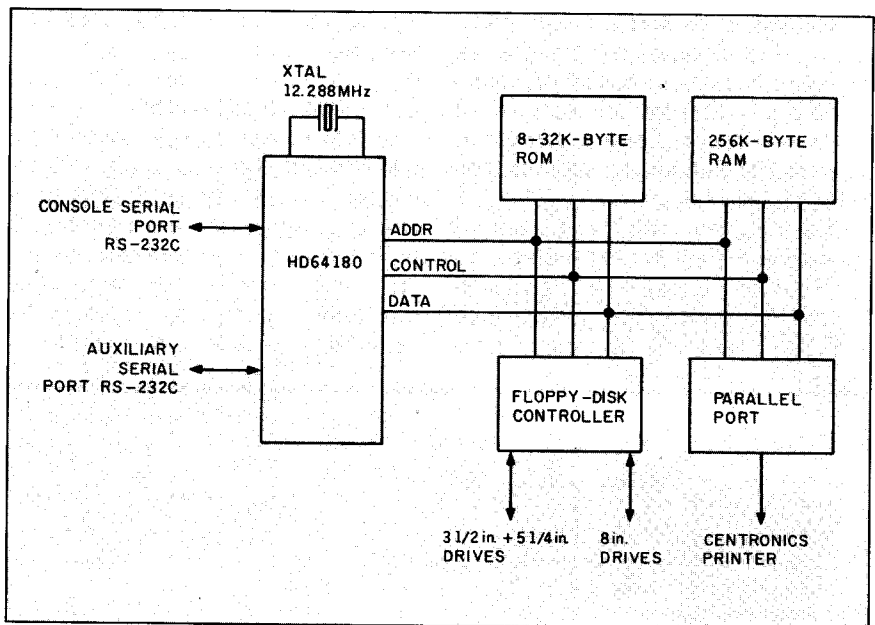


Figure 1: Block diagram of the SB180 single-board computer.

regularly so that it could have evolved into a more mature product.

These observations are certainly not unique. When millions of people use a computer and its operating system in an attempt to become more productive in some way, obstacles to efficiency and productivity are bound to show up. Computer users who have had experience with many different operating systems will miss certain features to which they had become accustomed. And creative individuals of all types always can think of a "better way" to perform some function already provided.

One of the best-known attempts to improve CP/M was ZCPR (Z80 command processor replacement), developed under the direction of Rick Conn. ZCPR was written to replace the console command processor (CCP) supplied with CP/M. This public-domain software featured scaled-down features of UNIX appropriate for a single-user CP/M system. However, since it was designed to fit into the same 2K-byte space as the CCP, it was limited in what it could offer. It did make user area 0 "public" so that executable programs there could be invoked from other user areas. It also changed the prompt so that it in-

dicated the user area as well as the drive currently logged in. As a result of user feedback, ZCPR2 evolved as an extended version of ZCPR. However, it required much more technical sophistication to install it into a CP/M system, and it started patching into and replacing parts of CP/M itself.

Within the past year, Echelon Inc. released ZCPR3, a much-improved version of ZCPR2, which provides solutions to all the problems with CP/M 2.2 and adds more features. Echelon has also developed a complete replacement for CP/M 2.2 in the form of the Z-System. It is composed of two major sections: ZCPR3 and ZRDOS. ZRDOS completely replaces CP/M's BDOS (basic disk operating system).

Written entirely in Z80 assembly language, the Z-System offers the benefits derived from the expanded Z80 instruction set (CP/M is 8080 code only) and from fixing bugs in CP/M 2.2 itself. It is downward-compatible with all CP/M software and takes advantage not only of the Z80 instruction set but of the Hitachi HD64180 CPU as well. Echelon provides both the operating system and a macro-relocating assembly lan-

(continued)

guage with linker and librarian, translators, and debuggers. The operating-system utilities, based on many good public-domain utilities, have all been rewritten to have a consistent user interface and make optimum use of ZCPR3 facilities.

The Z-System will be discussed more completely in next month's article. Suffice it to say, it has significantly greater utility and functionality than MS-DOS or plain CP/M. Table 1 gives

a comparison of the three operating systems.

THE HITACHI HD64180

The Hitachi HD64180 is based on a microcoded execution unit and advanced CMOS manufacturing technology. It provides the benefits of high performance, reduced system cost, and low-power operation while maintaining complete compatibility with the large base of standard CP/M

software. The HD64180 can be combined with CMOS VLSI (very-large-scale integration) memories and peripheral devices to form the basis for process-control applications requiring high performance, battery-power operation, and standard software compatibility.

Performance is derived from a high clock speed (6 MHz now, 9 MHz in the near future), instruction speedup, and an integrated memory-management unit (MMU) with 512K bytes of memory address space. The instruction set is a superset of the Z80 instruction set; 12 new instructions include hardware multiply, DMA (direct memory access), I/O, TEST, and a SLEEP instruction for low-power operation.

The HD64180 requires operation at specific frequencies in order to generate standard data-transmission rates. The standard operating frequency for the SB180 is 6.144 MHz (12.288-MHz crystal). Other frequencies that maintain standard data-transmission rates are 3.072 MHz, 4.608 MHz, and (later) 9.216 MHz.

System costs are reduced because many key system functions have been included on this 64-pin chip. Table 2 compares the HD64180 with other 8-bit processors.

The block diagram in figure 2 shows that the HD64180 CPU is composed of five functional blocks:

- Central processing unit: The CPU is microcoded to implement an upward-compatible superset of the Z80 instruction set. Besides the 12 new instructions, many instructions execute in fewer clock cycles than on a standard Z80.

- Clock generator: This generates the system clock from an external crystal or external clock input. The clock is programmably prescaled to generate timing for the on-chip I/O and system-support devices.

- Bus-state controller: This controller performs all status/control bus activity, including external bus-cycle wait-state timing, RESET, DRAM (dynamic RAM) refresh, and master DMA bus exchange. It generates "dual-bus" control signals for compatibility with

Table 1: A comparison of the Z-System, CP/M 2.2, and MS-DOS.

| Feature | Z-System | CP/M 2.2 | MS-DOS |
|---|----------|----------|---------|
| Software-compatible with CP/M 2.2 | yes | yes | no |
| No warm boot required when changing disks | yes | no | yes |
| Multiple commands per line | yes | no | no |
| Named directories | yes | no | yes |
| Password protection for directories | yes | no | no |
| Dynamically variable user privilege levels for commands | yes | no | no |
| Searching of alternate directories for invoked programs and files | yes | no | partial |
| Terminal-independent video capabilities | yes | no | no |
| Input/output redirection | yes | no | yes |
| Conditional testing and execution at the operating-system level (IF/ELSE/ENDIF) | yes | no | no |
| Shells and menu generators with shell variables | yes | no | no |
| Tree-structured on-line help and documentation subsystem | yes | no | no |
| 512-megabyte file sizes, 8-gigabyte disks | yes | no | no |
| Complete error trapping with recovery, customizable messages, and prompts | yes | no | no |
| Screen-oriented file manipulation and automatic archiving and backup | yes | no | no |
| Full-screen command-line editing with previous-command recall and execution | yes | no | no |

Table 2: A comparison of some 8-bit processors.

| | HD64180 | 8080/85/Z80 | NSC800 | Z800 | 80188 |
|---------------|------------|-------------|-----------|------------|------------|
| Process | CMOS | NMOS | CMOS | NMOS | NMOS |
| Power | 100 mW | 1 W | 100 mW | 2 W | 2 W |
| Maximum clock | 10 MHz | 8 MHz | 4 MHz | 10 MHz | 8 MHz |
| Address space | 512K bytes | 64K bytes | 64K bytes | 512K bytes | 1 megabyte |
| UARTs | 2-ch. | no | no | 1-ch. | no |
| DMAC | 2-ch. | no | no | 4-ch. | 2-ch. |
| Timers | 2-ch. | no | no | 4-ch. | 2-ch. |
| Clocked S/I/O | yes | no | no | no | no |
| CS/wait logic | yes | no | no | yes | yes |
| DRAM refresh | yes | yes (Z80) | no | yes | no |

Note: The availability of the Zilog Z800 is unknown, and specifications on it are subject to change.

both 68xx and 80xx family devices.

- Interrupt controller: The interrupt controller monitors and gives priorities to the four external and eight internal interrupt sources. A variety of interrupt-response modes are programmable.
- Memory-management unit: The MMU maps the CPU's 64K-byte logical-

memory address space into a 512K-byte physical-memory address space. The MMU organization preserves software object-code compatibility while providing extended memory access and uses an efficient common area/bank area scheme. I/O accesses (64K bytes of I/O address space) bypass the MMU.

The integrated I/O resources comprise the remaining four functional blocks:

- Direct-memory-access controller (DMAC): The two-channel DMAC provides high-speed memory-to-memory, memory-to-I/O, and memory-to-mem-

(continued)

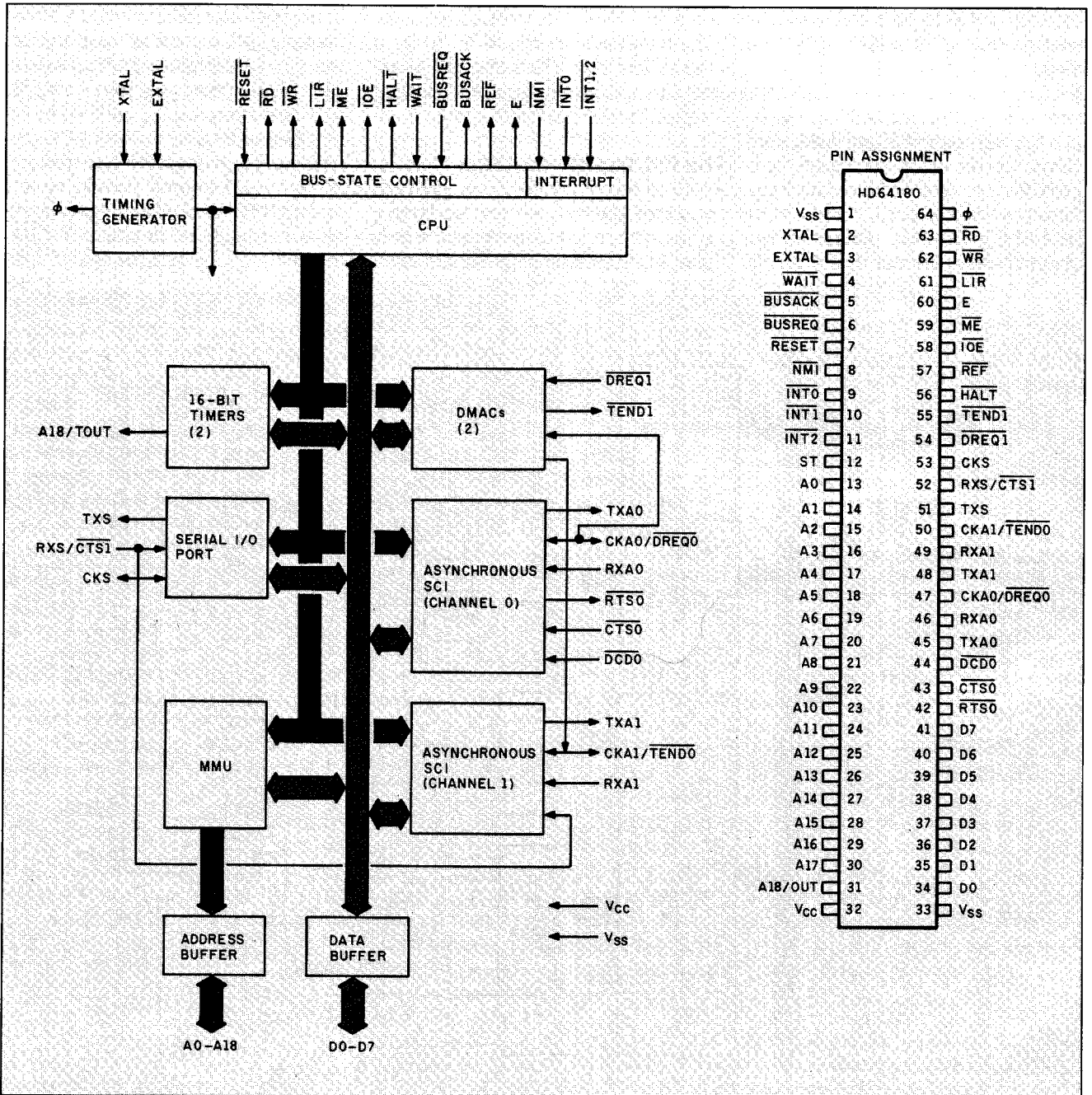


Figure 2: Block diagram and pin-out of the Hitachi HD64180.

ory-mapped-I/O transfer. The DMAC features edge or level sense-request input, address increment/decrement/no change, and (for memory-to-memory transfer) programmable-burst or cycle-steal transfer. In addition, the DMAC can directly access the full 512K bytes of physical-memory address space (the MMU is bypassed during DMA) and transfers (up to 64K bytes in length) can cross 64K-byte boundaries. At 6 MHz, DMA is 1 megabyte per second.

- Asynchronous serial communication interface (ASCI): The ASCI provides two full-duplex UARTs (universal asynchronous receiver/transmitters) and includes a programmable data-transmission-rate generator, modem-control signals, and a multiprocessor communication format. The ASCI can

use the DMAC for high-speed serial data transfer, reducing CPU overhead.

- Clocked serial I/O port (CSI/O): The CSI/O provides a half-duplex clocked serial transmitter and receiver. This can be used for high-speed connection to another microcomputer.

- Programmable reload timer (PRT): The PRT contains two separate channels, each consisting of 16-bit timer data and 16-bit timer-reload registers. The time base is divided by 20 from the system clock, and one PRT channel has an optional output allowing waveform generation.

SB180 DESIGN CRITERIA

With all this functionality on one chip, you can see why so few additional chips are needed to implement a truly sophisticated 8-bit single-board com-

puter in such a small space (less than 30 square inches). In terms of the original Altair microcomputer of 10 years ago, the functionally equivalent machine would have taken about 35 S-100 boards for a total of 1750 square inches (and that's using 8K-byte memory boards!).

In order to reduce chip count further, I decided to use an enhanced floppy-disk-controller chip (FDC) from Standard Microsystems Corporation, the 9266 (see figure 3). This 40-pin DIP (dual in-line package) chip is software-compatible with the industry-standard NEC 765A FDC and is actually an integrated combination of SMC's 9229 digital data separator and an NEC 765A FDC. It is compatible with single- and double-sided 3½-, 5¼- (40- and 80-track), and 8-inch

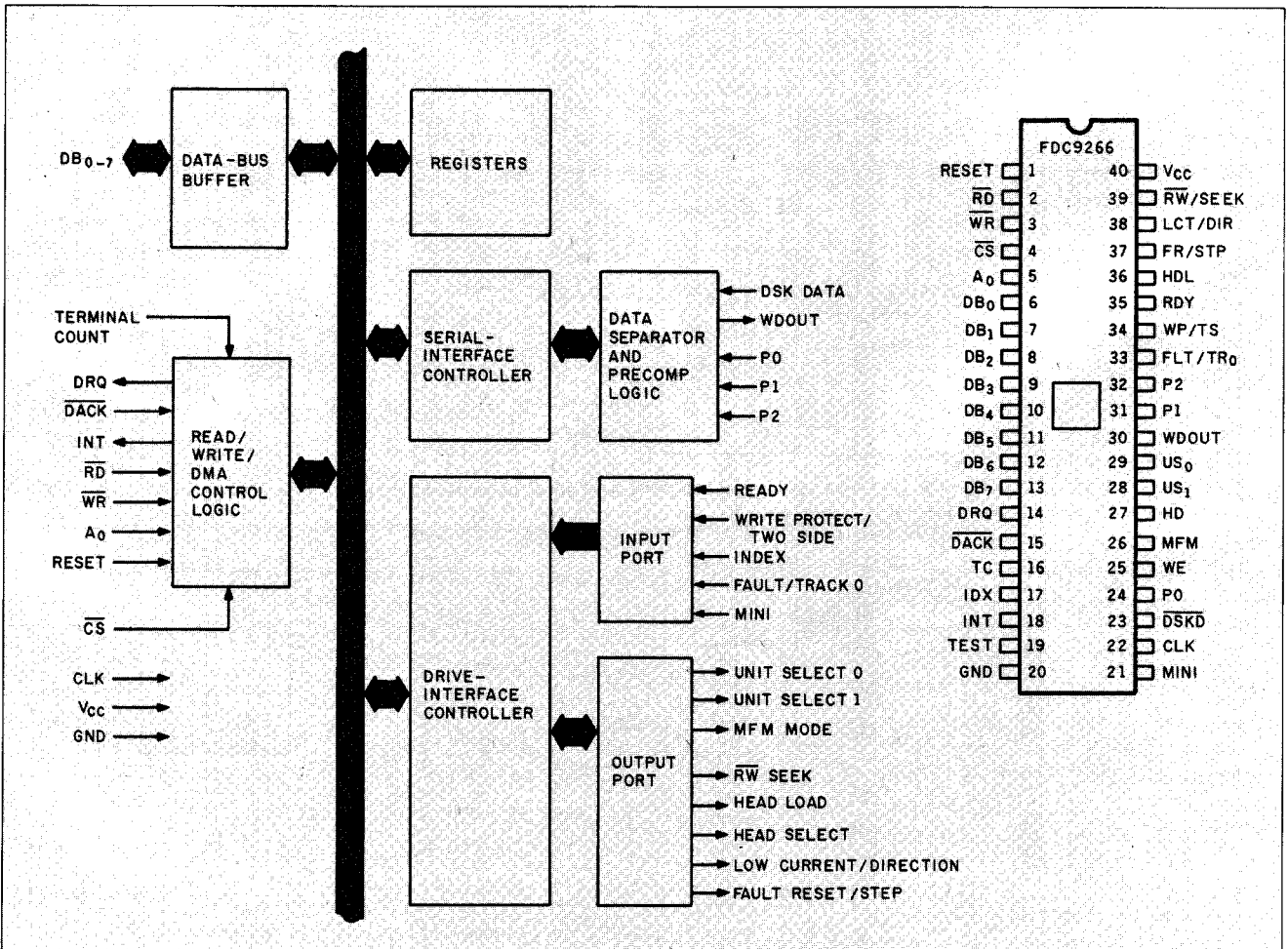


Figure 3: Block diagram and pin-out of SMC's 9266.

drives; the data separator handles both single- and double-density data. This means that it can be programmed to read and write most soft-sectored CP/M disk formats.

With the HD64180's two-channel ASCII built in, two serial ports were included in the design automatically. Provision was made for a Centronics parallel printer port as well. Since 256K-bit DRAM chips are now plentiful and inexpensive, eight were used for memory (64K-bit DRAMs can also be used). Because only 64K bytes is used for the logical-memory address space, you can optionally designate the other 192K bytes as a RAM disk in the operating system, or you may prefer to use it for other purposes (such as implementing banked memory for CP/M Plus).

THE SB180 HARDWARE

Figure 4 is the schematic diagram of the SB180 computer. Its design is primarily characterized by the high-performance, high-density MOS (metal-oxide semiconductor) devices, including 256K-bit DRAMs.

In addition to the CPU elements previously discussed, the SB180 system design implements the following functional blocks: RS-232C interface, memory interface, Centronics printer interface, floppy-disk interface, XBUS expansion bus, and power supply.

RS-232C INTERFACE

The HD64180 ASCII two-channel UART is connected to 1488/1489 RS-232C line drivers/receivers to provide two separate ports. ASCII channel 1 is used for the console; ASCII channel 0 is used for auxiliary RS-232C devices like printers, plotters, and modems. This distinction is made because modems require the extra handshake signals that are available with ASCII channel 0, while terminals do not. All primary RS-232C parameters (data-transmission rate, handshaking, data format, interrupts) are software-programmable.

MEMORY INTERFACE

The SB180 incorporates a 28-pin boot-ROM socket that can be

jumpered to hold 8K by 8-bit, 16K by 8-bit, and 32K by 8-bit memory devices. The boot ROM (contains disk boot and ROM monitor) occupies the bottom 256K bytes of the HD64180 physical-memory address space since it is selected whenever A18/TOUT is low. (Note: The TOUT timer output function is not used.) Thus, the boot-ROM contents (whatever the size) are simply repeated in the lower 256K bytes. The boot-ROM output (\overline{OE}) is enabled by the HD64180 \overline{ME} (memory enable) signal. (As configured, the maximum RAM on the SB180 is 256K bytes. To support larger memories, additional address decoding would be required to designate RAM and ROM areas in the current 256K-byte boot-ROM space.)

ROMs of 200 nanoseconds (and, marginally, 250 ns) can operate with one wait state.

At \overline{RESET} , the HD64180 begins execution at physical address 00000, the start of the boot ROM. [Editor's note: All addresses are in hexadecimal.]

256K-BIT DYNAMIC RAM

Standard 256K-bit 150-ns DRAMs, requiring 256 refresh cycles (8-bit refresh address) every 4 milliseconds (ms) are used. These DRAMs occupy the top 256K bytes of the HD64180 512K-byte physical-memory address space.

The interface is quite straightforward. Complete DRAM refresh control is provided by the HD64180 in conjunction with control logic IC14 and IC22 and address multiplexers IC11, IC12, and IC13.

The HD64180 \overline{WR} output directly generates DRAM \overline{WE} . The HD64180 \overline{ME} output directly generates RAS. During normal read/write cycles (A18 high, \overline{REF} high), \overline{CAS} goes low at the next rising edge of ϕ following the rising edge of E (enable). This provides plenty of setup time for the address multiplexers since the rising edge of E switches the address multiplexers from row to column addresses.

RAS-only refresh is used. The HD64180 generates the refresh addresses. During refresh cycles (\overline{REF} low), \overline{ME} generates RAS, while \overline{CAS} is

suppressed at IC22.

The HD64180 can be programmed to generate refresh cycles every 10, 20, 40, or 80 ϕ cycles as well as selecting a refresh every two or three clock cycles. Since the DRAM requires a refresh cycle every 15.625 microseconds (μ s) (4 ms/256), the HD64180 is programmed for 80-cycle refresh request since $80 \times (1/6.144 \text{ MHz}) = 13.02 \mu$ s. Two-cycle refresh is also programmed. Thus, refresh overhead is only 2.5 percent (2 cycles every 80 cycles).

CENTRONICS PRINTER INTERFACE

The Centronics printer interface is composed of the 8-bit latch IC9 and flip/flop IC10. The Centronics port is decoded at I/O address 0C0 by IC26. To write to the printer, the following sequence is used:

Write data to port 0C1

This sets up the data to the printer and asserts STB low. The following sequence:

Write data to port 0C0

deasserts the printer \overline{STB} signal high.

When the printer has processed the data, it will return the \overline{ACK} signal, which generates an external interrupt ($\overline{INT 1}$) to the HD64180. The interrupt handler clears the interrupt by performing a dummy output to port 0C0. This clears the $\overline{INT 1}$ interrupt request.

The printer interface is not buffered, so compatibility with all printer/cable setups cannot be guaranteed. In practice, however, problems should be rare since the software scheme provides adequate data-setup and -hold times. Also, note that this printer interface is interrupt-driven, which allows high-performance operation. In a more primitive polling design, excessive overhead limits acceptable performance in applications like background print spooling.

FLOPPY-DISK INTERFACE

SMC's 9266 FDC manages almost all details of the drive interface, including data separation and (with external logic IC22 and IC23) programmable

(continued)

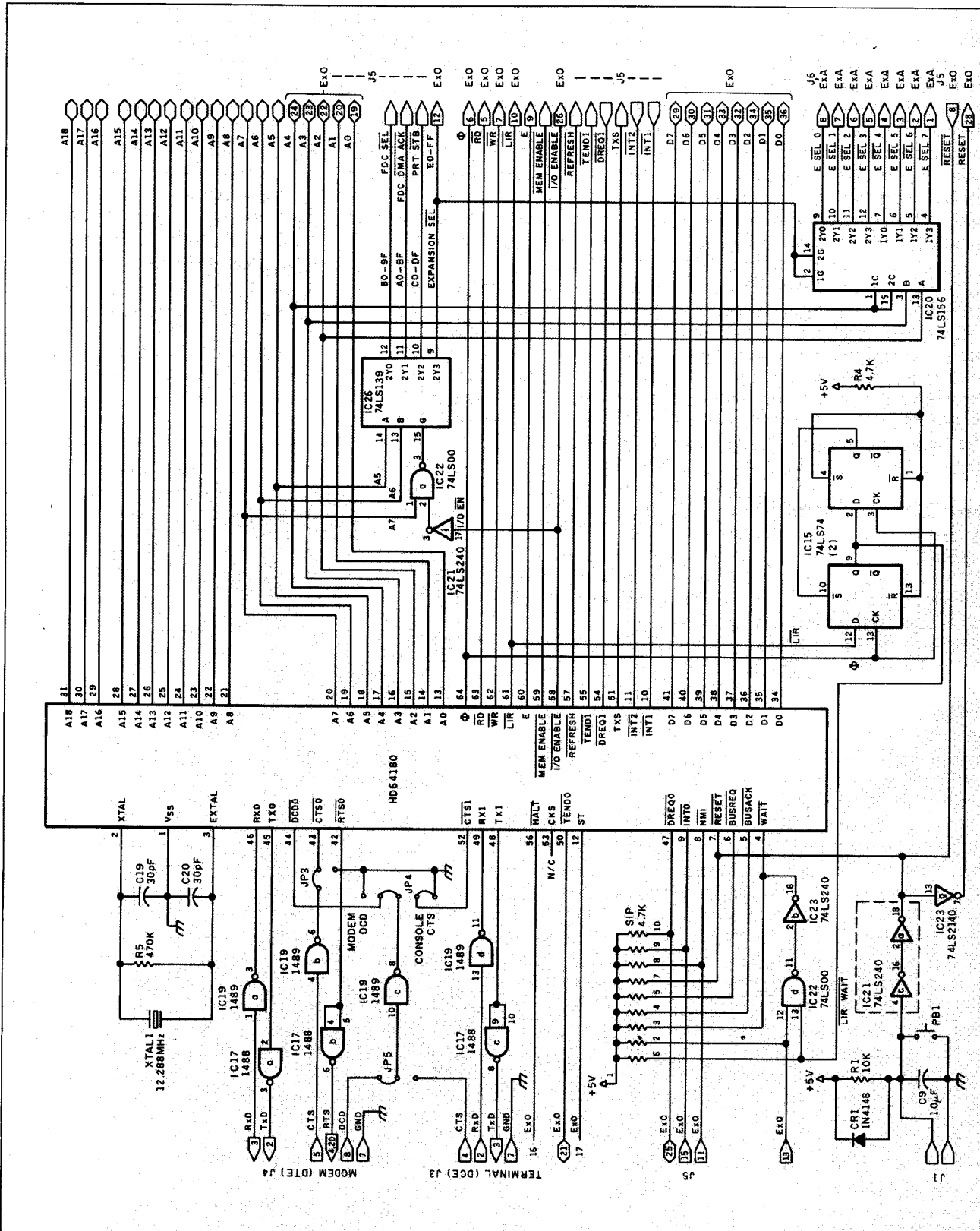


Figure 4: Schematic diagram of the SBI80 computer.

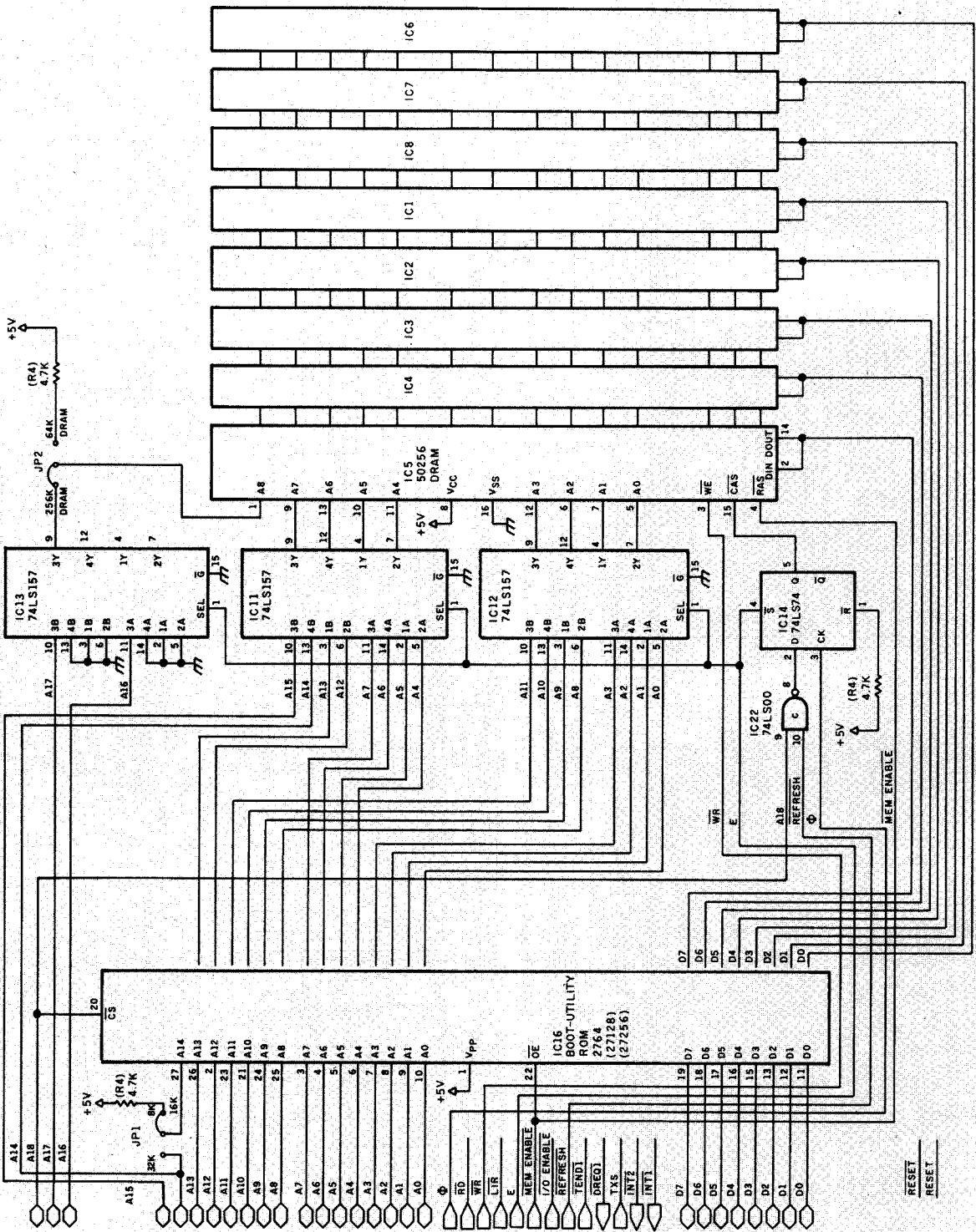
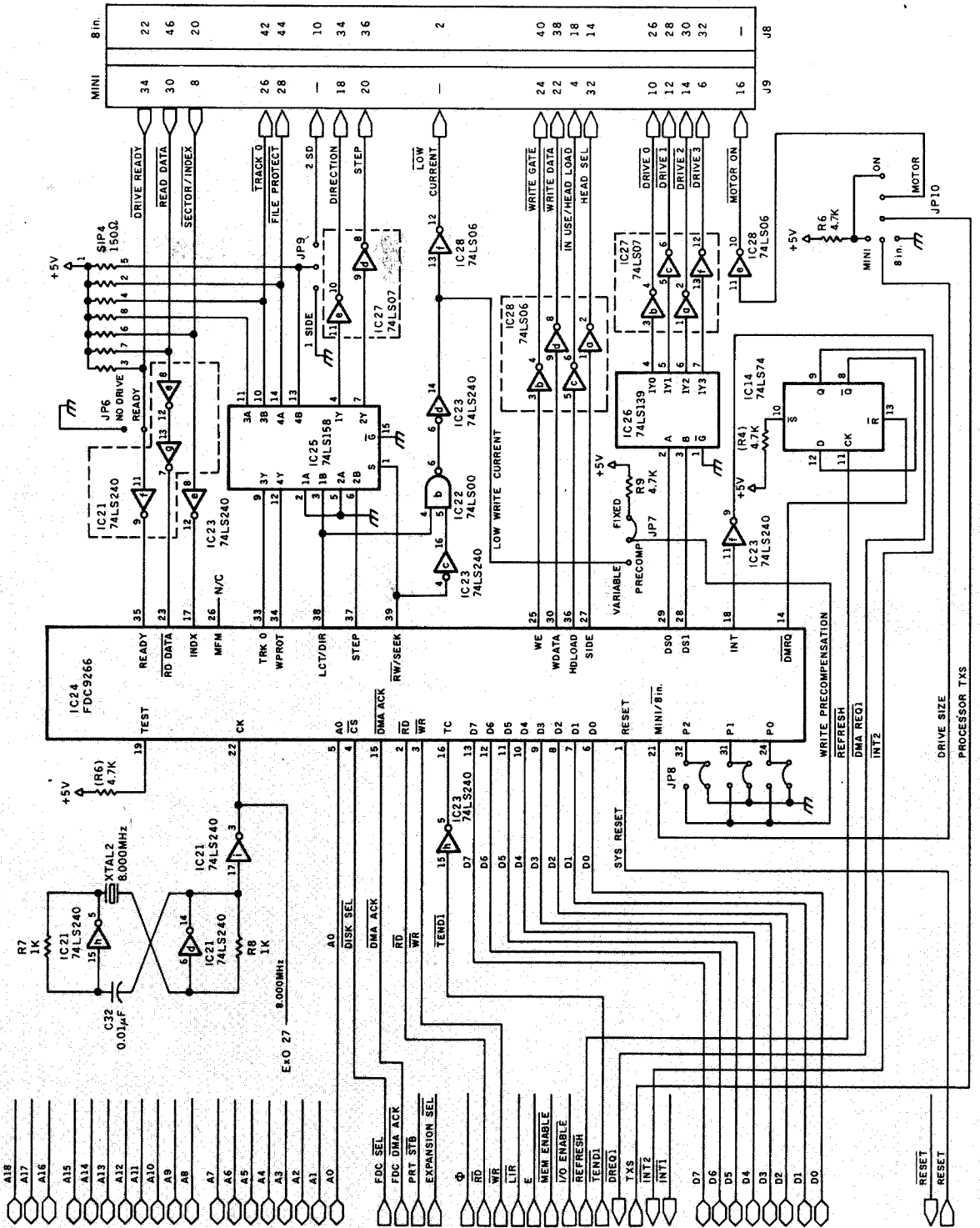
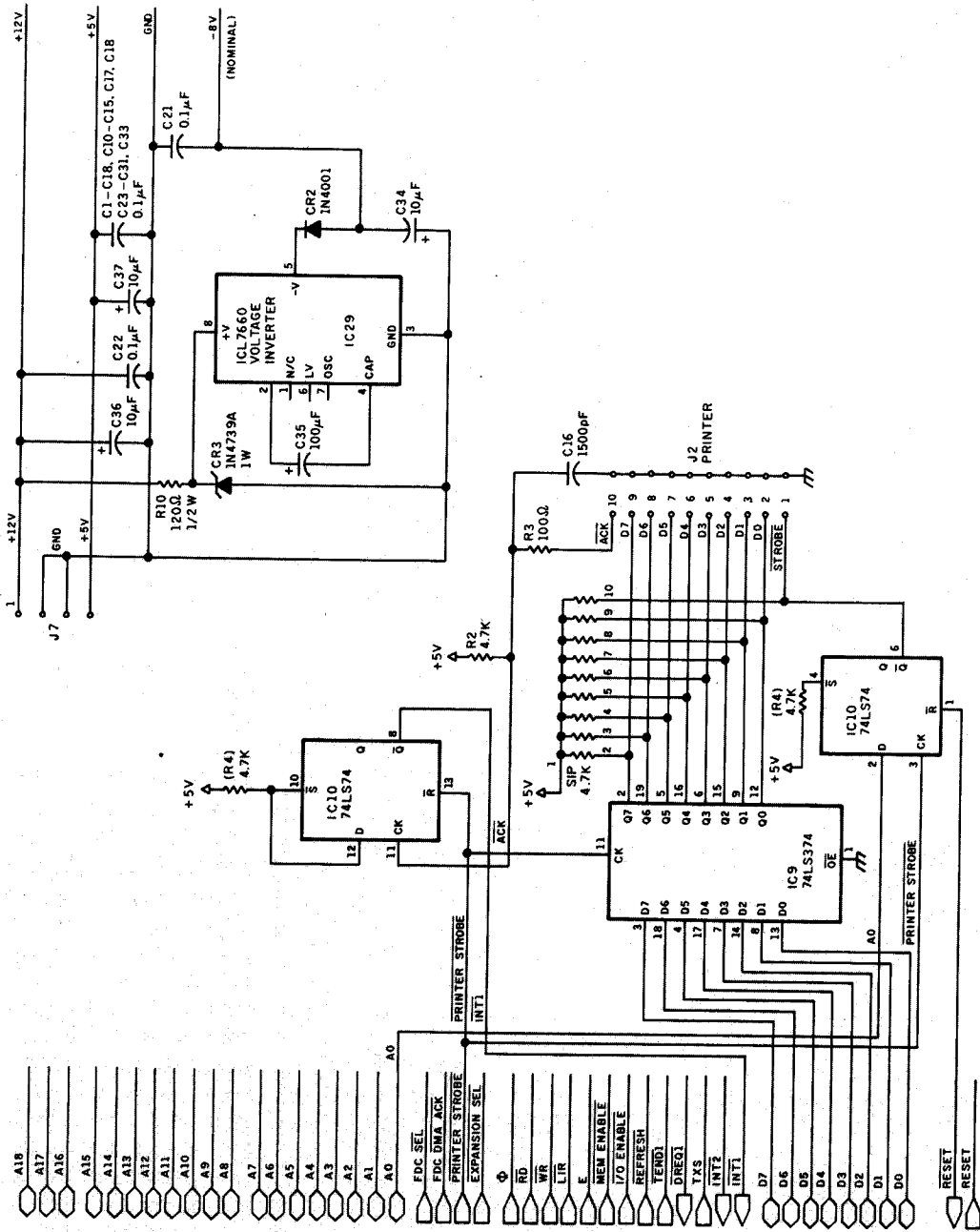


Figure 4 continued on page 96

Figure 4 continued





write precompensation. The 9266, as mentioned earlier, combines the NEC 765A/Intel 8272 FDC with SMC's 9229 digital data separator. Thus, from the host CPU side, the 9266 looks just like these devices, including hardware and software compatibility.

The 9266 clock is generated by an 8-MHz oscillator composed of a crystal and IC21. Jumpers are provided to select write precompensa-

tion and allow 8-inch floppy-disk drives to be interfaced.

On the CPU side, the key requirements are interfacing the 9266 with both programmed I/O (\overline{CS}), for things like initialization and status check, and with DMA (DRQ, DACK), for data transfer.

Programmed I/O is straightforward, with \overline{CS} generated for I/O address 80, and \overline{RD} and \overline{WR} directly generated by

the HD64180. This is the same scheme used to interface with other 80xx family peripheral devices.

DMA is a little more involved. First, DMAC channel 1 is used for the FDC since dedicated handshaking lines (\overline{DREQI} , \overline{TENDI}) are provided on the HD64180. Since DMAC channel 0 control lines are multiplexed (with ASCII clocks), DMAC channel 0 is used for memory-memory DMA. This means that the ASCII clock functions are available, although they are not currently used in this design.

For disk DMA, the 9266 asserts DRQ, which in turn causes HD64180 \overline{DREQI} assertion. The HD64180 performs DMA reads/writes to I/O address 0A0, which causes the 9266 DACK to be asserted, completing the transfer cycle. After the DMAC's programmed number of reads/writes has completed, the HD64180 \overline{TENDI} output is asserted and, after inversion, causes the 9266 TC (terminal count) input to be asserted, completing the DMA operation. This is typically followed by the 9266 generating an HD64180 $\overline{INT2}$ external interrupt. This interrupt-service routine can read the 9266's status to determine if errors occurred, etc.

However, one "gotcha," fixed by flip/flop IC14, conditions the 9266 DRQ output. It turns out that if 9266 DRQ directly generates HD64180 \overline{DREQI} , the HD64180 may respond too quickly. This is because HD64180 \overline{DREQ} input logic was designed to minimize latency, and DRQ can thus be recognized at a machine-cycle breakpoint. Unfortunately, the 9266 requires that at least 800 ns elapse from the time it asserts DRQ before the DMA transfer (\overline{DACK}) actually occurs. In other words, when the 9266 "asks" for service, it really doesn't want it . . . yet! To prevent accessing the 9266 too quickly after DRQ, DRQ from the 9266 is delayed at IC14 before issuing the \overline{DREQI} to the HD64180. DRQ is delayed by one REF cycle time.

Minifloppy double-density data transfers occur at a 250-kilohertz (kHz) data rate. Thus, each byte must be read within 32 μ s. The disk-driver software reprograms the refresh-

Table 3: Power connections for figure 4.

| | | +5 V | GND | +12 V | -8 V |
|------|-------------------------|------|------|-------|------|
| IC1 | 50256 DRAM 150 ns | 8 | 16 | — | — |
| IC2 | 50256 DRAM 150 ns | 8 | 16 | — | — |
| IC3 | 50256 DRAM 150 ns | 8 | 16 | — | — |
| IC4 | 50256 DRAM 150 ns | 8 | 16 | — | — |
| IC5 | 50256 DRAM 150 ns | 8 | 16 | — | — |
| IC6 | 50256 DRAM 150 ns | 8 | 16 | — | — |
| IC7 | 50256 DRAM 150 ns | 8 | 16 | — | — |
| IC8 | 50256 DRAM 150 ns | 8 | 16 | — | — |
| IC9 | 74LS374 | 16 | 8 | — | — |
| IC10 | 74LS74 | 14 | 7 | — | — |
| IC11 | 74LS157 | 16 | 8 | — | — |
| IC12 | 74LS157 | 16 | 8 | — | — |
| IC13 | 74LS157 | 16 | 8 | — | — |
| IC14 | 74LS74 | 14 | 7 | — | — |
| IC15 | 74LS74 | 14 | 7 | — | — |
| IC16 | 2764 monitor ROM 200 ns | 28 | 14 | — | — |
| IC17 | 1488 | — | 7 | 14 | 1 |
| IC18 | HD64180 | 32 | 1,33 | — | — |
| IC19 | 1489 | 14 | 7 | — | — |
| IC20 | (74LS156) optional | 16 | 8 | — | — |
| IC21 | 74LS240 | 20 | 10 | — | — |
| IC22 | 74LS00 | 14 | 7 | — | — |
| IC23 | 74LS240 | 20 | 10 | — | — |
| IC24 | FDC 9266 | 40 | 20 | — | — |
| IC25 | 74LS158 | 16 | 8 | — | — |
| IC26 | 74LS139 | 16 | 8 | — | — |
| IC27 | 7407 | 14 | 7 | — | — |
| IC28 | 7406 | 14 | 7 | — | — |
| IC29 | 7660 regulator | — | 3 | — | — |

Jumpers

- JP1 (1x3) ROM size
- JP2 (1x3) RAM size
- JP3 (1x3) CTS0 disable
- JP4 (1x5) RS-232C gate output
- JP5 (1x3) RS-232C gate input
- JP6 (1x2) Drive without ready
- JP7 (1x3) Write precomp mode
- JP8 (2x6) Write precomp value
- JP9 (1x3) Single-/double-sided drive
- JP10 2 - 1x3 TXS function/drive size

request rate from every 80 ϕ cycles to every 40 ϕ cycles prior to disk DMA and reassigns it back to 80 ϕ cycles after the disk DMA is completed. The 9266 DRQ is delayed from between 40 ϕ cycles to 79 ϕ cycles. This is about 6 to 14 μ s. Therefore, the 800-ns delay and 32- μ s data-transfer constraint are both met. Note that 8-inch floppy double-density data transfer is twice as fast (500 kHz) and requires a refresh-rate increase to every 20 ϕ cycles.

EXPANSION BUS

The spare \overline{CS} from address decoder IC26 (I/O addresses 0E0 to 0FF) and all major buses (address, data, control) are routed to the expansion bus. This allows an I/O expansion-board capability. The full complement of HD64180 control signals (\overline{IOE} , \overline{E} , \overline{RD} , \overline{WR} , etc.) allows easy interface to all standard peripheral LSI (large-scale integration) devices, including the 80xx, 68xx, and 65xx families. Example expansion boards could include a hard-disk controller, 1200-bits-per-second (bps) modem, or LAN (local-area network) interface.

POWER SUPPLY

The SB180 requires +5-V and +12-V power. A negative voltage is generated on board, which is used only by the RS-232C driver. The negative voltage is obtained by using a Zener diode to obtain +9 V from +12 V, which is then inverted using an Intersil 7660 converter. The +12-V power is used only for the RS-232C driver. Thus, the SB180 uses only significant power from the +5-V supply. Typically, this may be from 0.3 to 0.6 ampere (A) (depending on the proportion of the TTL [transistor-transistor logic] and memory devices that are CMOS)—about the same as a 5¼-inch floppy disk.

THE SB180 ROM MONITOR

While my initial discussion espoused the merits of the Z-System, every good single-board computer needs a ROM monitor that should be more useful than just booting the operating system from a floppy disk. The SB180

- A ASCII table: Prints an ASCII table
- B Bank select: Selects a 64K-byte memory bank
- C Copy disk: Systems with 256K bytes of RAM can perform single-drive copies
- D Display memory: Displays memory in hexadecimal and ASCII
- E Emulate terminal: Console keyboard is echoed to the auxiliary RS-232C output, and the RS-232C input is echoed on the console display
- F Fill memory: Any portion of memory is filled with a data byte
- G Go to program: Starts program execution at specified address and optionally includes a breakpoint
- H Hexmath: Prints the 20-bit sum and difference and 32-bit product of the two arguments
- I Input port: Prints the 8-bit data input from specified port
- K Klean disk: Formats a specified drive
- M Move memory: Moves a block of memory
- N New command: Enables new commands from extended ROM space
- O Output port: Byte is output to specified port address
- P Printer select: Toggles printer selection between the Centronics parallel port and the auxiliary RS-232C port
- Q Query memory: Searches memory for pattern of 1 to 4 bytes
- R Read disk: Reads specified sectors from drive into memory
- S Set memory: Displays memory contents and allows new data to be entered
- T Test system: Tests various system devices
- U Upload hexadecimal file: Uploads Intel hexadecimal file from auxiliary or console serial port
- V Verify memory: Compares two blocks of memory
- W Write disk: Writes specified sectors to disk from memory
- X eXamine CPU registers: Displays main and alternate CPU registers and prompts for modification of main registers
- Y Yank I/O registers: Displays the HD64180 on-chip I/O registers
- Z Z-System boot: Boots the Z-System from disk

Figure 5: A list of the monitor commands.

8K-byte ROM monitor includes commands for everything you need—from A to Z. It is supplied on an 8K-byte EPROM. (The SB180 also supports 16K- and 32K-byte ROMs, so additional commands or application programs can be supported.)

The SB180 ROM monitor provides commands to assist the design and debugging of SB180-related hardware and software. Also, it serves as a stand-alone training vehicle for the HD64180 CPU.

The monitor supports the following I/O devices:

- CON: Console RS-232C serial port
- AUX: Auxiliary RS-232C serial port
- CEN: Centronics parallel printer port

DSK: Floppy-disk storage devices

The monitor supports one to four 5¼-inch, 48- or 96-tpi (tracks per inch), 40- or 80-track, double-sided double-density disk drives. During initial system checkout, such a drive should be connected to verify operation of the disk interface.

At RESET, the monitor first checks for fatal RAM failure. If the RAM is bad, the monitor waits for a carriage return to be typed on the console in order to determine the console data-transmission rate and then prints 8 binary digits—each corresponding to one DRAM chip. A chip associated with a 1 is faulty. The RAM-check se-

(continued)

quence is repeated each time a carriage return is typed.

If the RAM is okay, the stack is set up, and the monitor checks to see if a disk is loaded in drive #0. If so, the DOS boot-load sequence (same as the Z command) is started.

The monitor commands are shown in figure 5.

THE LUNCHBOX COMPUTER

The only things you need to turn the SB180 into a full-fledged stand-alone microcomputer are a +5-V and +12-V

power supply (or a 12-V battery and 5-V regulator), at least one 5¼-inch floppy-disk drive (initially), and a serial terminal.

You can mount such a small computer in many ways (see photos 3 and 4). If you use a half-height drive, the SB180 will fit on top of the drive inside a single full-height disk-drive chassis including power supply! Since the console serial port can automatically detect the terminal's data-transmission rate, you can carry your SB180 with you and connect it to any terminal (or computer emulating a terminal) running at 300, 1200, 9600, or 19,200 bps (other rates are optionally selected).

It is possible to fit the SB180, power supply, and two 3½-inch floppies into a lunchbox. If you want to get fancy, you could fit the SB180, power supply, one or two half-height floppies, the Micromint Term-Mite terminal board, and a keyboard into a small attaché case. Use any handy video monitor, and . . . voilà! You have just out-*Osborned* Adam Osborne!

I can imagine many unusual ways to package the SB180. When good LCDs (liquid-crystal displays) come down in price a bit more, I think the SB180 can form the basis of a functional notebook computer. The SB180 makes minimal use of the +12-V supply (and only for RS-232C operation) and can use less than 1 A at +5 V, so battery power is a real possibility. With just one floppy disk, the SB180 can become a super data logger. While my Z8 BASIC or FORTH single-board computer might ordinarily be used for such an application, not everyone likes to program in FORTH or BASIC. With a CP/M-compatible computer, a developer can now choose Pascal, C, FORTRAN, or even PL/I for applications. Since the data is already in CP/M format, it makes data analysis convenient.

The Z-System is provided with the complete SB180 board and software package. While it comes with a utility to read several other common 5¼-inch formats (like Kaypro and Osborne), its native format is identical to the well-optimized 386K-byte

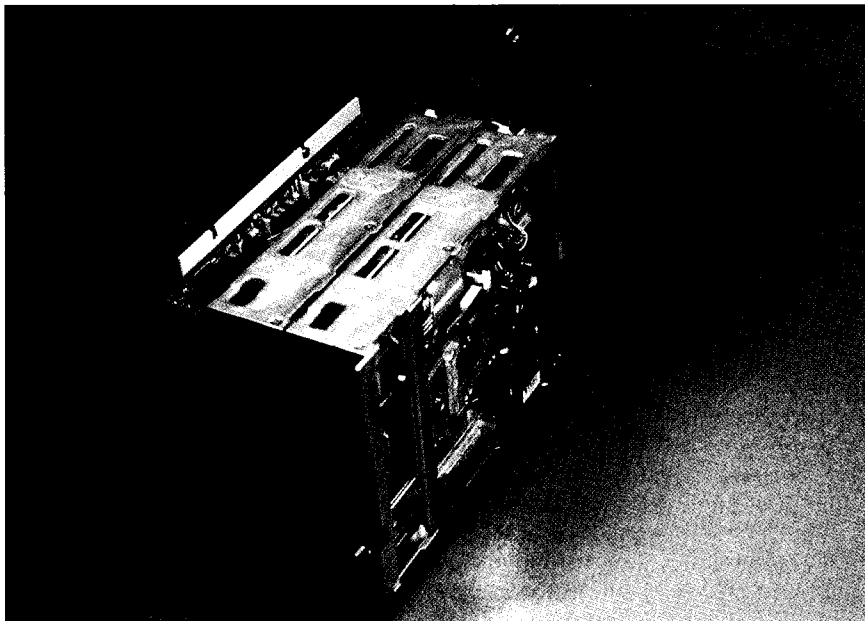


Photo 3: The SB180 can be packaged in many off-the-shelf enclosures advertised in BYTE. It fits comfortably to the left of a pair of half-height 5¼-inch disk drives in a case available from Disk Plus. This enclosure also includes the power supply.

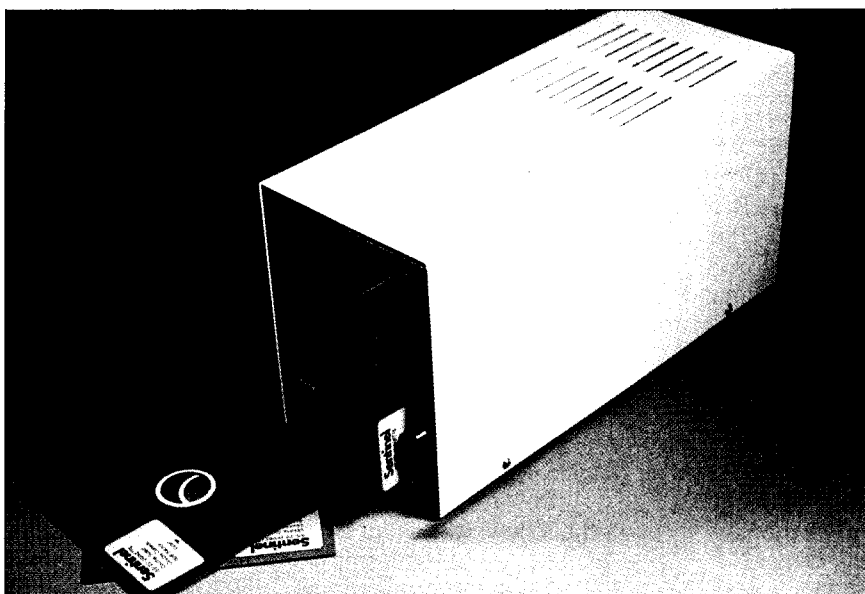


Photo 4: The enclosure from photo 3 is shown with the cover on. The only additional hardware necessary to use the system is a serial terminal.

double-sided double-density format of the Little Board from Ampro Inc. Provisions have been made to support 8-inch drives as well. Of course, it is possible to implement CP/M 2.2, CP/M Plus, MP/M II, TurboDOS, or Oasis if you prefer.

Although simple benchmarks are not to be taken as the last word in describing a computer's performance, I did run the BYTE Sieve of Eratosthenes prime-number program on the SB180 and the MPX-16 (8088-based with a 4.77-MHz clock, same as the IBM PC) using appropriate versions of Microsoft BASIC. For one iteration, the MPX using GW-BASIC took 203 seconds, while the SB180 using MS-BASIC took 147 seconds. An empty FOR . . . NEXT loop with 20,000 iterations took 27 seconds on the MPX and only 18 seconds on the SB180. Don't forget that these results are based on the 6-MHz implementation of the HD64180; with a 9-MHz clock, the results will be spectacular—not quite the speed of an IBM PC AT, but close!

EXPERIMENTERS

As always, I try to support the computer experimenter by rewarding diligence. If you build the SB180 from scratch, send me a picture, and I'll send you a copy of the BIOS and ROM monitor on disk (SB180 format, double-sided double-density) at no charge, provided it is for your personal use. A printed listing is available for a modest charge.

If you build, buy, or otherwise assemble an SB180 system, I would like to know about it. I will be designing expansion boards for the SB180 and can notify you of them in advance of publication. In addition, having your name will greatly simplify the organization of any users groups that might arise.

CIRCUIT CELLAR FEEDBACK

This month's Circuit Cellar Feedback is on page 416.

NEXT MONTH

In part 2 of this article, I'll describe the software for the SB180. ■

Special thanks to Tom Cantrell, Merrill Lathers, and Bob Stek for their contributions to this project.

The following items are available from

The Micromint Inc.
25 Terrace Dr.
Vernon, CT 06066
(800) 635-3355 for orders
(203) 871-6170 for information

1. SB180 computer board with 256K bytes of RAM. Complete with user's manual and ROM monitor.
assembled and tested . . . SB180-1, \$369
complete kit SB180-2, \$349
2. SB180 boot disk. Z-System DOS with limited utilities and BIOS. Provided on one 5¼-inch SB180 format double-sided double-density disk SB180-10, \$49
3. Z-System, including ZRDOS, ZCPR3, an editor and utilities, ZAS assembler, BIOS source, and ZDM debugger. Complete with manuals. Provided on four 5¼-inch SB180 format double-sided double-density disks SB180-20, \$190
4. HD64B180 chip (6 MHz) with data manual and 12.288-MHz XTAL \$50
5. BYTE readers' special. Complete SB180 computer board with 256K bytes of RAM, user's manual, ROM monitor, and all the software listed in item 3. (Available through December 31, 1985.)
assembled and tested. SB180-1-20, \$499
complete kit SB180-2-20, \$479

All boards are complete with the exception of the 50-pin, 8-inch drive and 44-pin expansion headers, which are not populated. They are optionally available. Printer, power, disk, and terminal cables are available separately. Call for pricing.

Please include \$10 (\$7 less on item 4) for shipping and handling in the continental United States, \$18 elsewhere. Connecticut residents please include 7.5 percent sales tax.

Editor's Note: Steve often refers to previous Circuit Cellar articles. Most of these past articles are available in book form from BYTE Books, McGraw-Hill Book Company, POB 400, Hightstown, NJ 08250.

Ciarcia's Circuit Cellar, Volume I covers articles in BYTE from September 1977 through November 1978. *Volume II* covers December 1978 through June 1980. *Volume III* covers July 1980 through December 1981. *Volume IV* covers January 1982 through June 1983.

To receive a complete list of Ciarcia's Circuit Cellar project kits, circle 100 on the reader-service inquiry card at the back of the magazine.