Upgrading the Heath H8 With a Z-80 Microprocessor

Now that Heath has boarded the Z-80 bandwagon, keep in step with the HZ8 adapter.

Patrick Swayne
290 Springdale
Sebastopol, CA 95472

When the Heath Company decided to get into the hobby computer business, the 8080 was the most popular micro, and thus the most logical choice for their CPU. But the Z-80 has since eclipsed it in popularity, and now Heath has joined the parade by introducing their own Z-80-based machine, the H89. This computer uses the same disk and cassette operating systems as the H8. Currently, the only software available for it is 8080 software, but sooner or later software will appear that takes advantage of the Z-80's expanded instruction set. Over 15,000 H8 owners may be left behind as a result.

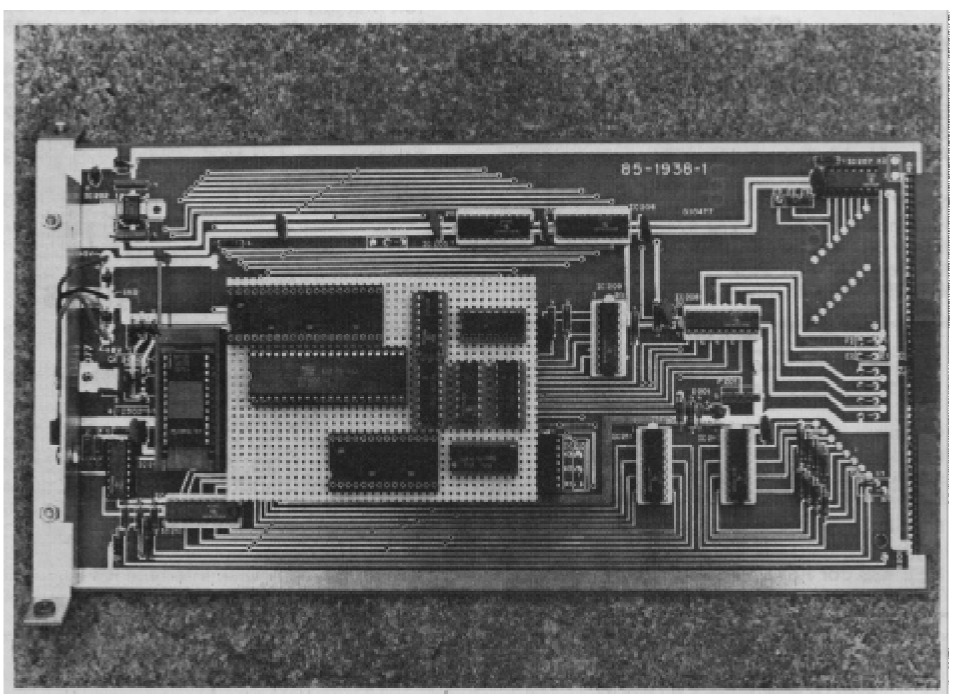

Photo 1. H8 CPU board with the adapter plugged in. The Z-80 (D780C) and some TTL ICs replace the original 8080 and 8238. I placed two extra ICs for the optional section, which I designed after I took this photo, in the lower-left corner of the perfboard. The Textool socket to the left of the adapter is for testing homemade ROM monitors.

Photo 1 shows my solution to this problem. It is a Z-80 adapter that mounts piggyback onto the H8 CPU board. It plugs into the sockets normally occupied by the 8080 CPU, the 8238 (or 74S438) system controller and the 8224 clock driver. No modifications are required on the CPU board itself, and at any time you can remove the adapter and replace the original ICs. It can be built for about $30 to $40, including the Z-80 (which has dropped to less than $15 at some mail-order houses).

How the Adapter Works

The schematic of the adapter is shown in Fig. 1. In this circuit, the original CPU and system controller are replaced by a Z-80, six commonly available TTL chips, a capacitor and some resistors. (Two of the TTLs and the capacitor may be eliminated if the optional section is not built.) The clock driver is also shown because some connections were made to it. (A socket for the clock driver was included in the actual assembly to make these connections possible, even though it is still electrically on the CPU board.)

This circuit effectively emulates all of the signals normally produced by the 8080 and 8238 that are used by the H8. This is easy to accomplish because Heath chose to use the fully decoded 8238 system controller signals, rather than the undecoded status or control signals (except M1) that are multiplexed onto the data bus of the 8080.

You can produce those signals by simply ANDing together the appropriate Z-80 outputs. For example, the $\overline{\text{MEMR}}$ (memory read) signal is produced by ANDing the $\overline{MREQ}$ (memory request) and $\overline{RD}$ (read) outputs. The NOR gates on the schematic are used as negative AND gates in this application (except those used as inverters).

Noninverting OR gates (such as the 74LS32) could have been used, eliminating the need for inverters at the outputs of the NORs, but the chip count would not have been reduced, since the circuit requires at least five ORs and seven inverters. I chose to use 74LS02s because they are more readily available. Pull-up resistors are required at each of the four memory and I/O control outputs of the Z-80, because they are Tri-state and might, at some time, be undefined.

Deriving the 8238-type memory and I/O control signals was easy, but some of the others are not so obvious. The $\overline{INTA}$ (interrupt acknowledge) signal is derived by ANDing the $\overline{IORQ}$ (I/O request) and $\overline{M1}$ (first machine cycle) outputs. The H8 uses an M1 signal and decodes it by ANDing the 8080's PD_5 and SYNC outputs. Since the Z-80 already has an $\overline{M1}$ output, I simply ran it (through an inverter) to the SYNC input of the 8224 and tied PD_5 (actually, D_5 on the 8238) high through a resistor.

The $\overline{WAIT}$ input on the Z-80 is the same as the READY input on the 8080, except that it does not have to be timed, so I connected the raw RDYIN input at the 8224 to the Z-80 $\overline{WAIT}$ input. The Z-80 $\overline{INT}$, $\overline{BUSRQ}$ and $\overline{BUSAK}$ signals are just inverted counterparts of the 8080 INT, HOLD and HLDA pins, so I connected them together through inverters. The $\overline{NMI}$ (non-maskable interrupt) on the Z-80 is not used, so I tied it high. (Later on, I may write a monitor that uses that interrupt to return to monitor control from a user program, which would make possible debugging a program that had interrupts disabled.)

The $\overline{HALT}$ and $\overline{RFSH}$ outputs of the Z-80 are not needed, and were left unconnected. The address pins on the Z-80 were connected to the corresponding pins on the 8080 socket, and the data pins go to the DB pins on the 8238 socket.

The Z-80 does not produce a counterpart to the 8080 INTE (interrupt enable flip-flop) output. The H8 uses that signal to light an LED on the front panel and to operate the single instruction (SI) button. INTE is produced by the optional section of the circuit, and non-machine-language hackers who never use the SI button may leave that section out and tie pin 16 on the 8080 socket high instead. If you do that, the SI button will do nothing, and the ION light will always be on.

The INTE output of the 8080 is set or reset according to whether interrupts are enabled or disabled. Interrupts may be enabled and disabled by software, using the EI and DI instructions, and are always disabled when the processor receives an interrupt. The HZ8 adapter circuitry responds only to EI and DI, but that is sufficient to achieve normal operation of the H8. It does this by examining the data bus at M1 time, when op codes are fetched.

The EI instruction in binary is 11111011, and DI is 11110011. Bit 3, the only one that changes, goes to the data input of a D-type flip-flop, while the others, along with M1, are used to clock the data through when they are all present. Capacitor C_1 slows the clocking down just enough to ensure that D_3 has settled down before it is sampled.

Single Stepping

Before I designed the INTE circuitry on the adapter, I looked into various software single-step schemes, which I considered using, and found that they are all deficient in some way. Most cannot handle certain instructions, and all require considerably more code than the Heath method. The H8 can single-step through all of the 8080's instructions, and, with my adapter, through all of the Z-80's codes in only a few bytes of

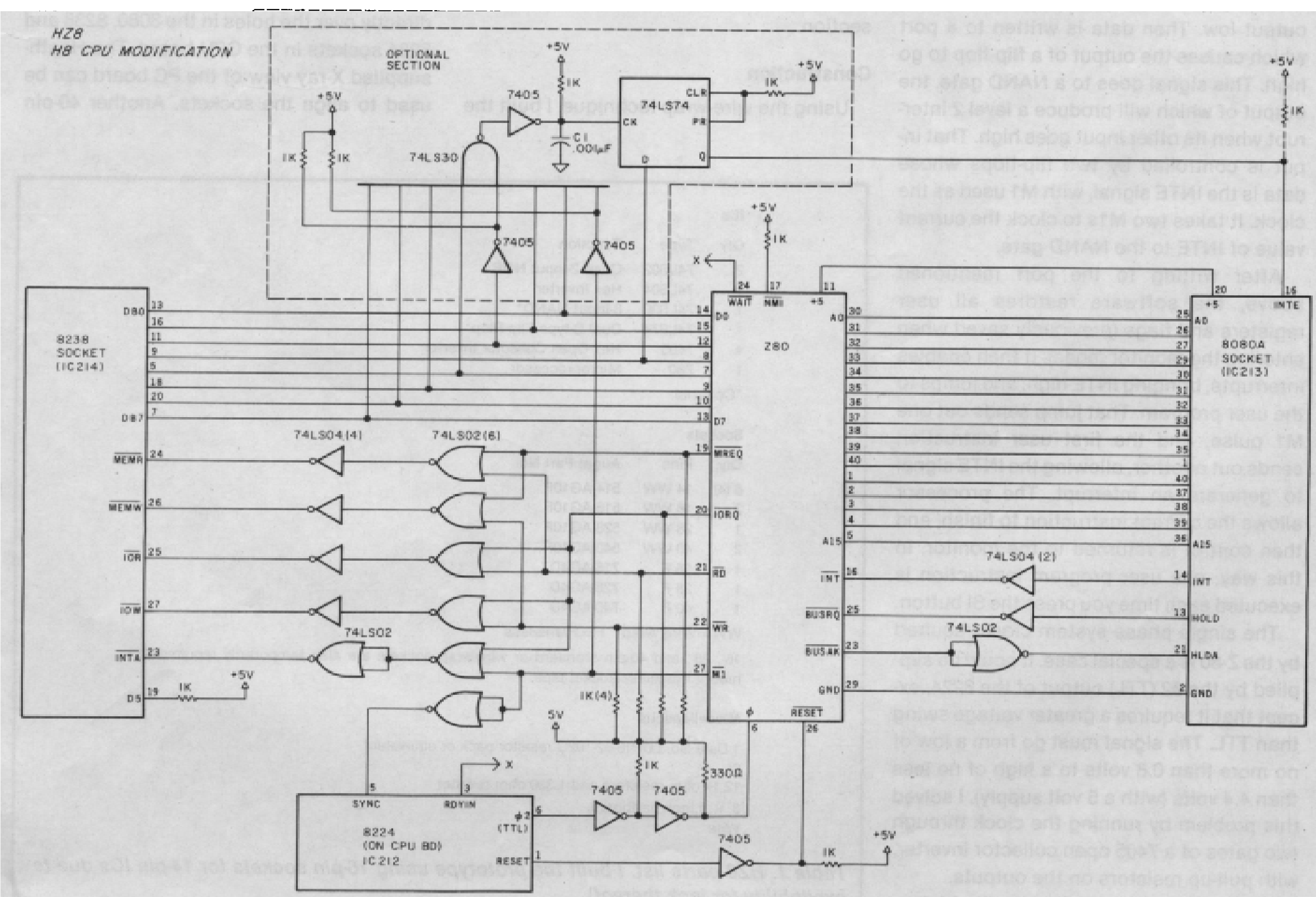

Fig. 1. Z-80 adapter circuit. Connect the points marked "X"; don't tie INTE (pin 16) on the 8080 socket high if optional section isn't built.

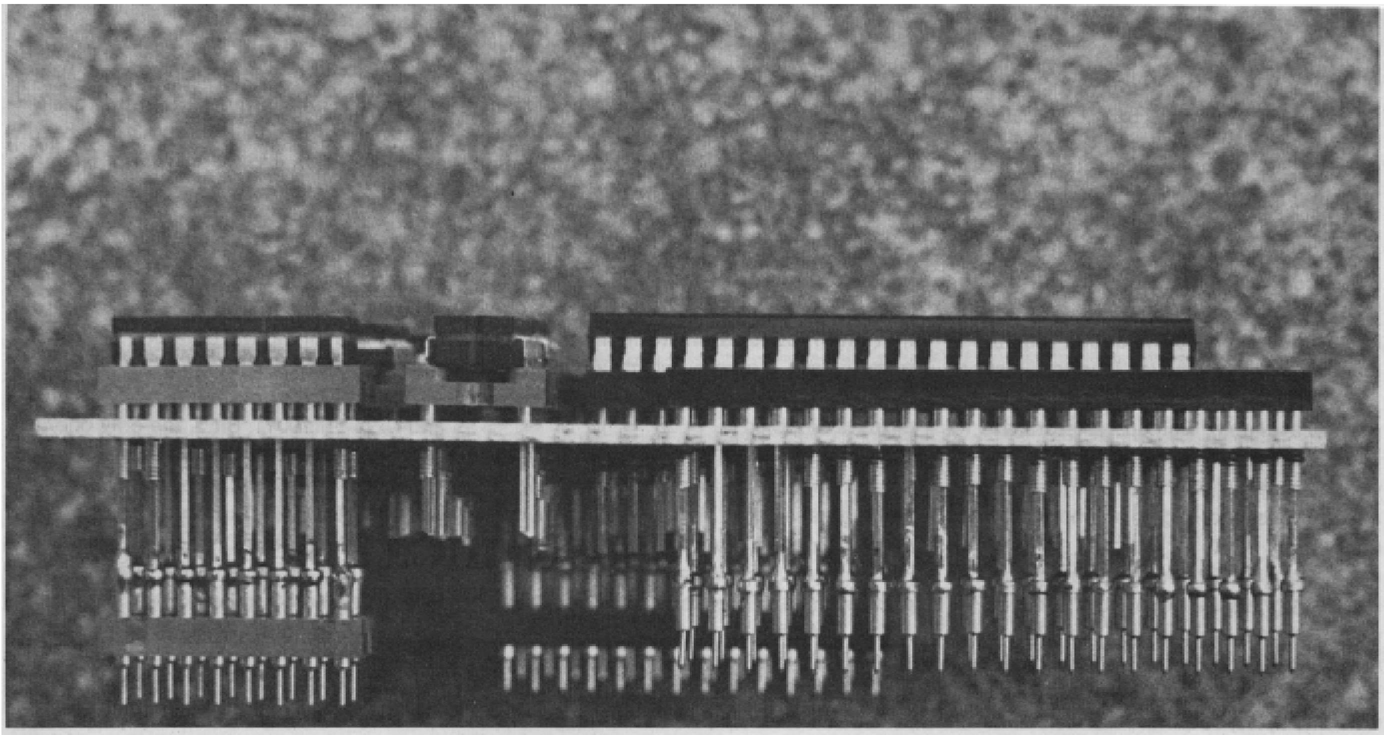

Photo 2. Side view of adapter board shows the frameless socket pins attached to the ends of the wire-wrap pins, using an ordinary framed socket (left) for alignment. If pins remain at full length, the CPU card won't fit in the first motherboard position.

code.

When you press the SI on the H8 front panel, the monitor jumps to a routine that first disables interrupts, bringing the INTE output low. Then data is written to a port which causes the output of a flip-flop to go high. This signal goes to a NAND gate, the output of which will produce a level 2 interrupt when its other input goes high. That input is controlled by two flip-flops whose data is the INTE signal, with M1 used as the clock. It takes two M1s to clock the current value of INTE to the NAND gate.

After writing to the port mentioned above, the software restores all user registers and flags (previously saved when entering the monitor mode). It then enables interrupts, bringing INTE high, and jumps to the user program. That jump sends out one M1 pulse, and the first user instruction sends out another, allowing the INTE signal to generate an interrupt. The processor allows the current instruction to finish, and then control is returned to the monitor. In this way, one user program instruction is executed each time you press the SI button.

The single phase system clock required by the Z-80 is a special case. It could be supplied by the Ø2 (TTL) output of the 8224, except that it requires a greater voltage swing than TTL. The signal must go from a low of no more than 0.8 volts to a high of no less than 4.4 volts (with a 5 volt supply). I solved this problem by running the clock through two gates of a 7405 open collector inverter, with pull-up resistors on the outputs.

To ensure a fast rise-time to the higher-than-normal voltage, a 330-ohm resistor pulls up the input to the Z-80. Another gate of the 7405 is used to invert the 8224 RESET to supply the Z-80's inverted version of the signal, and the rest are used in the optional section.

Construction

Using the wire-wrap technique, I built the adapter on a 3 × 4 inch piece of standard perfboard with 0.1 inch spaced holes. I placed 40-pin, 28-pin and 16-pin wire-wrap sockets on the boards so that their pins are directly over the holes in the 8080, 8238 and 8224 sockets in the CPU board. The Heath-supplied X-ray view of the PC board can be used to align the sockets. Another 40-pin

ICs

Qty.	Type	Function
2	74LS02	Quad 2-Input NOR
1	74LS04	Hex Inverter
1	74LS30	8-Input NAND*
1	74LS74	Dual D-type Flip-Flop*
1	7405	Hex Open Collector Inverter
1	Z80	Microprocessor

*Optional

Sockets

Qty.	Pins	Augat Part No.
6 (4)	14 WW	514-AG10F
2	16 WW	516-AG10F
1	28 WW	528-AG10F
2	40 WW	540-AG10F
1	16 F	716-AG4D
1	28 F	728-AG4D
1	40 F	740-AG4D

WW = Wire Wrap F = Frameless

16-, 28-, and 40-pin standard or wire-wrap sockets are also temporarily required to mount frameless socket pins.

Miscellaneous

1 Dale No. LDP16-02-102G resistor pack or equivalent
or
12 1k ohm resistors and 1 330 ohm resistor
3 × 4 inch perfboard
Wire

Table 1. HZ8 parts list. I built the prototype using 16-pin sockets for 14-pin ICs due to availability (or lack thereof).

wire-wrap socket, which holds the Z-80, was installed between the other 40-pin and 28-pin sockets. Placement of the other sockets is not important, but wire lengths should be kept short.

Power and ground for the TTL were derived from the VCC and ground pins of the 8238 socket, except for the 7405, which gets its ground and power from the 8224 socket. The two extra chips for the optional section were added later, and get their power from the Z-80 socket. For the pull-up resistors, I used a resistor package containing 15 resistors, all 1k ohm, connected internally to a common pin. To make the 330 ohm resistor, three resistors in the package were paralleled. Discrete resistors may also be used.

After being wired, the circuit should be checked with an ohmmeter or continuity tester for correctness, because when it is finished it will be difficult to make changes. To make it possible to plug the adapter into the CPU board, frameless socket pins were soldered to the ends of the wire-wrap pins on the 8080, 8238 and 8224 sockets. Frameless sockets are socket pins that are installed on an aluminum frame that is removed after the sockets are soldered into a PC board.

To connect these pins to a wire-wrap socket, they should first be removed from the frame and plugged into an ordinary framed socket. That will hold them in place while they are soldered to the wire-wrap pins.

Photo 2 shows how the pins are connected. If the wire-wrap pins are left their full length, the CPU card with the adapter installed will not fit in the first motherboard position. If the wire-wrapping is kept close to the board, and all pins are cut so that they protrude no more than 1/4 inch from the perfboard (before the frameless pins are attached), then the board will just fit in the first slot.

Those H8ers capable of making their own PC boards and using that technique would not have a thickness problem. The frameless socket pins could be soldered directly to the back of the PC board over the protruding pins of the sockets.

Installation and Checkout

The adapter cannot be plugged into the CPU board unless the 8080, 8238 and 8224 sockets on the board are all of the flat, low-profile type. If any are not, they will have to be changed. The 8080 and 8238 ICs should be stored in anti-static material (foil will do) while they are not in use.

After the TTL ICs, resistor pack and 8224 are installed, the adapter can be plugged into the CPU board. If the CPU is to be used in the second motherboard slot, framed sockets can be plugged onto the socket pins to protect them before the adapter board is plugged in. Then the Z-80 can be installed, and the CPU board replaced in the computer.

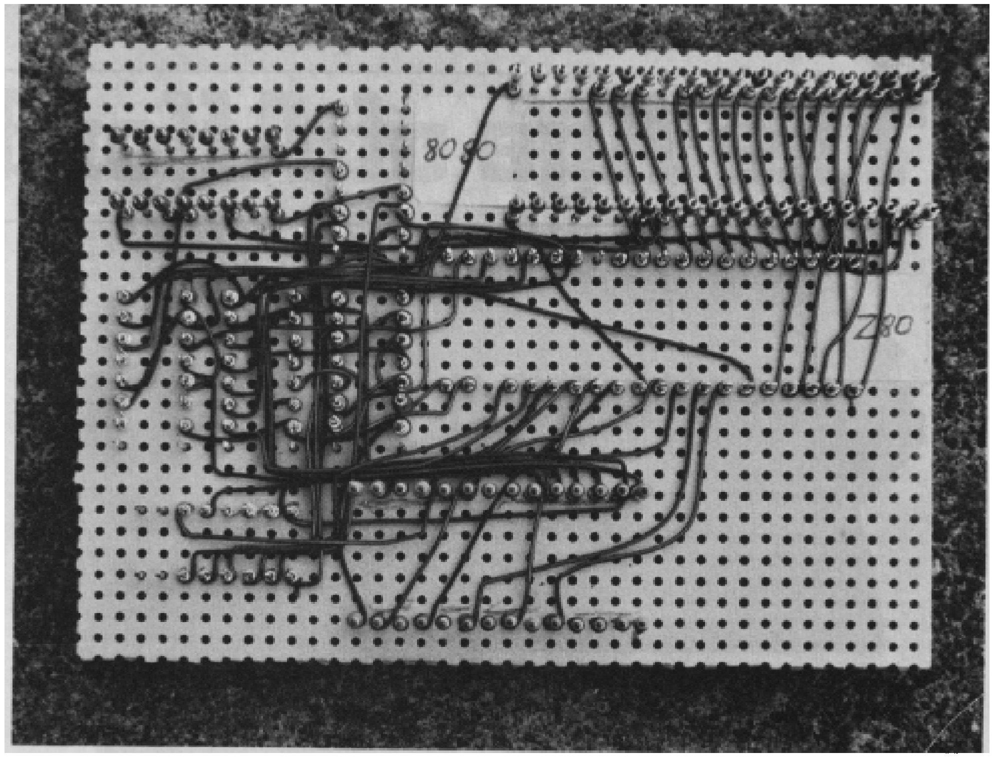

Photo 3. Bottom view of the adapter showing wire-wrap technique.

Checkout is simply a matter of turning the computer on. If PAM-8 (the front-panel monitor) signs on normally by lighting the displays and beeping the horn, everything else should also run normally. If there is trouble, the first area to check is the wiring and then the chips.

The Z-80

The 8080 instruction set consists of 244 individual op codes. Like most 8-bit processors, each op code consists of one byte, making 256 the total number of codes possible. The Z-80, however, uses four of the 12 op codes not used by the 8080 as the first byte of several two-byte op codes. You can think of these as 16-bit op codes that are fetched one byte at a time. In that sense, the Z-80 is a predecessor of the Intel 8088, a 16-bit micro designed to use an 8-bit data bus.

The two-byte op codes are used for a variety of purposes, including 16-bit arithmetic, bit manipulation, working with two index registers and some versatile block move and search instructions. These last types are an elementary form of microcode, another way in which the Z-80 looks forward to the big machines.

Two of the Z-80's 16-bit instructions adversely affect the operation of the optional section of the adapter circuit during normal running (but not during single stepping). The second bytes of these instructions, SET 6,E and SET 7,E, have the same binary code as DI and EI. Since the Z-80 issues an M1 pulse for each byte of two-byte op codes, the adapter circuitry sees the second bytes of those SET instructions as DI and EI. The result is that the front panel ION light may be lying if those instructions are in the code. Upon return to monitor, however, an EI instruction is encountered, and proper indication is restored.

The Z-80 uses five of the other eight unused 8080 op codes for an unconditional and four conditional relative jumps. In the 8080, all jumps require three bytes—one for the op code and two for the address. With the Z-80 relative jumps, a single byte following the op code specifies the jump destination as a signed 8-bit offset added to the program counter.

Another of the unused 8080 codes is a special relative jump instruction that decrements the B register each time it is executed and jumps if B is greater than zero. The remaining two extra op codes are used to switch between alternate sets of the six general-purpose registers and alternate flag registers and accumulators. In all, the Z-80 has about 700 op codes in its instruction set.

A Final Note

Those who use Heath's cassette assembler, HASL-8 version 4.01.01 or 4.02.00, will have to make a patch before it will work on a Z-80. The program contains one of the 8080's unused op codes, 40 (octal), which is one of the Z-80's relative jumps. To correct this bug in version 4.01.01, change the contents of address 055.265 (split octal) to 000. To correct version 4.02.00, change 055.365 to 000. ■