KOMPUTER KORNER

How the microcomputer controls input/output devices with software

DAVID LARSEN, PETER RONY, and JOHN TITUS*

THIS MONTH WE WOULD LIKE TO EXPLAIN how computer instructions cause an I/O device to operate. The I/O device that we shall choose for our discussion is the optically isolated solid-state AC relay. These relays can control any AC power device within the output current ratings of the relay. Shown in Fig. 1 are typical solid-state relays that are available for prices ranging from $5 to $20 in quantities of one. Such relays permit a single TTL output signal (logic 0 or logic 1) to control up to 10 amperes of 220 VAC power, as is possible with the Hamlin model 7522 relay (top center of Fig. 1). Internally, each relay contains a light-emitting diode, a light-sensitive transistor, a power triac and a transparent dielectric optical path that isolates the digital and power circuitry. It can withstand a voltage difference of at least 1000 volts.

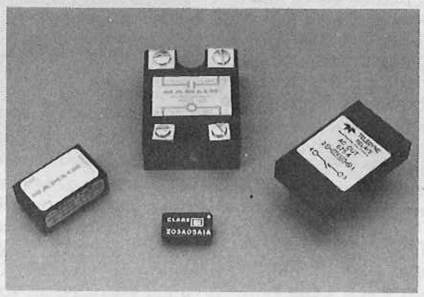

FIG. 1—TYPICAL SOLID-STATE RELAYS that are optically isolated.

A typical microcomputer I/O circuit that employs the solid-state relay is shown in Fig. 2. Recall that the microcomputer sends synchronization pulses, called *device select pulses*, to the I/O device. In Fig. 2, these are the negative-going pulses from the SN74154 decoder circuit that, for an 8080 microcomputer, have a time duration of only 500 ns. It should be clear that a single 500 ns pulse cannot sustain the continuous operation of an AC power device. What is required is a simple "interface" between the microcomputer and the solid-state relay that would permit the AC power device to operate continuously, if it is so desired. A suitable interface is a single SN7474 positive-edge triggered flip-flop and a single buffer from a SN7407 hex buffer/driver IC. The buffer is needed since it is not good engineering practice to drive a solid-state AC relay directly from the output of a flip-flop.

With suitable software, the microcomputer and SN74154 decoder can generate individual device select pulses that either clear or set the SN7474 flip-flop. To clear the flip-flop and turn on the AC power device, only a single 500 ns pulse is needed. The flip-flop output, Q, will remain at logic 0 until a single 500 ns pulse is applied to the preset input at which time the AC device will turn off.

Any simple open-collector gate or inverter can be used as the buffer between the output of the flip-flop and the input of the solid-state relay. Suitable choices would be the SN7401 or SN7403 2-input NAND gates, the SN7405 inverter, or the SN7409 2-input AND gate.

Output instructions

We will discuss microcomputer instructions in considerable detail in subsequent columns. To summarize, their exist 78 different instructions for the 8080 microprocessor IC, and a total of 256 variations of such instructions. Each instruction contains a single 8-bit *instruction code* that indicates which type of operation or group of operations the microcomputer will execute. Some instructions contain two or three 8-bit words that are present in successive memory locations. A *byte* is defined as a group of eight bits occupying a single memory location. Thus, the 8080 microprocessor instructions are either 8-, 16-, or 24-bits long, with the first eight bits always being the instruction code.

The output instruction is a 16-bit instruction that consists of two successive bytes located in successive memory locations. The first byte, in binary code, is always 11010011. The second byte can be any 8-bit binary number from 00000000 to 11111111; this is the device code of the specific output device that will receive eight bits of data from the accumulator. The contents of the accumulator remains unchanged after the instruction is executed.

A simple program

The simplest program that incorporates the output instruction is probably the one given below:

Memory address	Instruction byte	Description
0	11010011	Send device select pulse to device given by following 8-bit device code
1	00000000	Device code for clear input to SN-7474 flip-flop
2	01110110	Halt the microcomputer

An 8080 microcomputer operating at a clock rate of 2 MHz will execute the previous program in 8.5 μs. The AC-power device will remain on once the program has been executed. To turn off the device, a slightly different program is required:

(continued on page 24)

*This article is reprinted courtesy American Laboratories

KOMPUTER KORNER
(continued from page 22)

Memory address	Instruction byte	Description
0	11010011	Send device select pulse to device given by following 8-bit device code
1	00000001	Device code for preset input to SN-7474 flip-flop
2	01110110	Halt the microcomputer

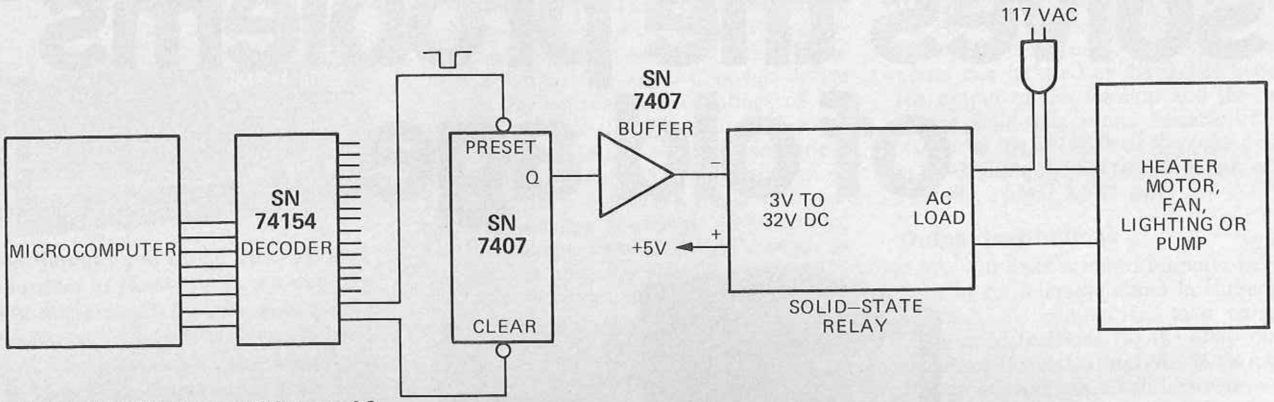

FIG. 2—I/O CIRCUIT for controlling an AC-power device such as a fan, heater or motor.

The AC-power device will turn off when the second instruction byte in the program is executed and remain off after the microcomputer halts. A more practical program requires additional instructions.

Keep in mind that a memory address contains 16 bits. When we write, "memory address 0," we really mean the memory address corresponding to the following 16-bit binary word: 00000000 00000000. Note that we have split the 16 bits into two parts, the most significant 8-bits and the least significant 8-bits. The most significant 8-bits are called the HI (or H) memory address and the least significant 8-bits are called the LO (or L) memory address. Both the LO and HI address must be present for a memory location to be accessed.

With an actual program, the solid-state relay shown in Fig. 2 will turn on and off according to various decisions made by the program. A typical microcomputer-controlled system could easily have several such relays.

In a more orderly and systematic treatment of the 8080 microprocessor, one would probably introduce the 8080 instruction set prior to the discussion of any particular instruction, such as the output instruction described this month. Since we do not believe that you are willing to wait several months until we get to the output instruction, we have decided to treat it first. In a future column, we will explain how the 8-line-to-256-line decoder circuit shown in Fig. 2 generates individual device select pulses. **R-E**