computer corner

Z-80 *The Z-80 instruction set and how it compares to the 8080 instruction set.*

WILLIAM BARDEN, JR.

LAST MONTH'S COLUMN DISCUSSED THE pin-out of the Z-80 IC and the Z-80's timing. Now let's take a look at the Z-80's instruction set.

One of the most confusing things to a novice microcomputer user is the use of the instruction set. A simple computer may be constructed with a set of ten instructions or so. This basic computer can perform any operation that a computer with an instruction set of 200 instructions is capable of performing. Naturally, the program, or set of instructions to perform the function, will be much longer for the simple computer than for the 200-instruction computer.

The 200-instruction computer offers a versatile set of instructions from which a programmer can choose. If the programmer is very familiar with the instruction set of a computer, he may choose instructions that optimize the program for size (number of instructions) and speed (number of instructions to be executed for a given function).

The Z-80 offers a set of 158 instructions from which the Z-80 programmer may choose. Many of the instructions are very powerful. However, many instructions simply are included to achieve compatibility with the 8080 and 8008. Since some of 8008/8080 types of instructions do not follow classical instruction lines, they prove confusing to both novice and experienced programmer alike.

Z-80 instruction set

The Z-80 instruction set may be divided into 8 major groups:

1. Load and Exchange
2. Block Transfer and Search
3. Arithmetic and Logical
4. Rotate and Shift
5. Bit Manipulation
6. Jump, Call and Return
7. Input/Output
8. CPU Control

The Load and Exchange instructions move data 8 or 16 bits at a time between CPU registers or between CPU registers and external memory including stack. Many of these types are provided in the 8080. Moving data from memory to CPU registers is necessary to load the CPU A register in preparation for arithmetic or logical operations such as adding two 8-bit operands or performing a logical AND or shift operation. When the operation has been performed, or after a series of operations, the CPU register results may be moved back to memory by the same type of instruction. Transfers between CPU registers and memory are also required in preparation for a "programmed" I/O (*I*nput/*O*utput) operation where an 8-bit operand is transferred to or from an I/O device via a CPU register. Eight- or 16-bit data may also be moved from one CPU register to another CPU register, or the contents of two CPU register pairs may be exchanged. In all load operations, the *source data* is *copied* to a *destination* register or memory location. The original data *remains unchanged*. All computers have instructions of this type.

Load and Exchange example:

LD A, 1000H Loads the A register with the contents of location 1000_{16}. Location 1000 remains unchanged. (H stands for hexadecimal, or base 16.)

The Block Transfer and Search instructions are very powerful instructions that few computers and no other current microprocessors have. With one instruction, the Z-80 can transfer a block of memory data from one set of locations in memory to another, or search a block of memory for a given character. In other microprocessors, the block transfer or search would take perhaps six to ten instructions to implement, with subsequent longer transfer and search times. The following instruction transfers 100 bytes of data from location 1000_{10} to location 2000_{10} in the Z-80. The source location (1000_{10}) and the destination location (2000_{10}) were previously loaded into the H,L and D,E register pairs by two move type instructions. The number of bytes to be moved (100_{10}) was also preloaded into the B,C register pair.

An example of Block Transfer is shown in Fig. 1.

Arithmetic and Logical instructions are found in all computers and microprocessors. In the Z-80, all 8-bit arithmetic and logical operations use the current A register (A or A'). A second 8-bit operand from another CPU register or external memory is used in an addition, subtraction, AND, OR, or similar operation with register A. The results of the operation are transferred to register A and appropriate flags are set. Certain types of 16-bit arithmetic operations are also permitted, notably additions between CPU register pairs and the H,L register pair (destination). Binary-coded-decimal (BCD) addition and subtraction can be performed by using the standard addition and subtraction instructions and executing a special DAA or *Decimal Adjust Instruction* that adjusts the results of the addition or subtraction to BCD format.

Addition example:

If register A contains $00001010_2 (10_{10})$ and register B contains $01000111_2 (71_{10})$, executing ADD B produces:
(A) = $01010001_2 = 81_{10}$ (Sum)
(B) = $01000111_2 = 71_{10}$ (Contents of B unchanged)

Rotate and Shift type instructions expand greatly upon the 8080 capabilities. The 8080 can only *rotate* the contents of the A register, or the contents of the A register and carry bit together, one bit left or right. The Z-80, however, not only can perform a rotate shift right or left, but can also perform *arithmetic* and *logical* shifts. In addition, not only can register A be shifted, but *any* CPU register, or *any* memory location. Finally, two BCD digit rotate instructions perform a rotate of an accumulator BCD digit and two BCD digits in a memory location.

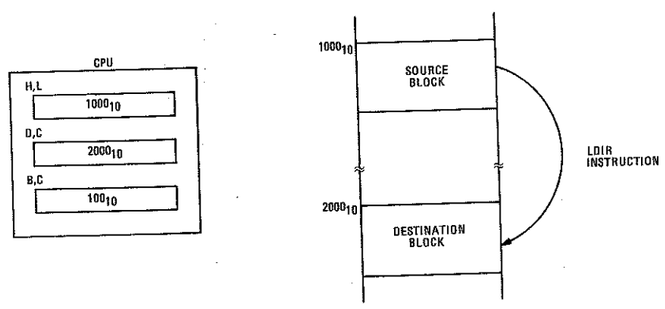

FIG. 1

The Z-80 Bit Manipulation instructions enable the programmer to test, reset, or set any bit of a CPU register or any memory location. None of this group is provided in the 8080. Testing, resetting and setting bits are common operations that would take several instructions in other microprocessors.

Bit Manipulation example:

SET 3,B Sets bit 3 of CPU register B

The Jump, Call and Return instructions include the 8080 unconditional and conditional jumps, calls and returns. A jump instruction reroutes program execution to a specified address either unconditionally or conditionally based on one of the flag bits (carry, no carry, zero, etc.). A call performs the same jump, but pushes the return address into the memory stack so that a return may be made by a return instruction, which pops the return address from the stack and causes the program to return to the next instruction following the call. Calls and returns are used for subroutine operation. A subroutine is any set of instructions that perform a certain function (square root, search a table, etc.). In addition to the 8080 instructions in this group, the Z-80 includes jump "relative" instructions, jump "indexed" and several other types.

The Input/Output group of instructions include the 8080 IN and OUT instructions. The IN and OUT instructions enable one byte of data to be transferred between the A register and an I/O device. In the Z-80, however, one byte may be transferred between any CPU register and an I/O device. Also included are block input or output instructions that effect the transfer of up to 256 bytes of data between any CPU register and the I/O device.

Input/Output example:

H,L Holds the address of the block of data to be output (1000_{10})
B Holds the number of bytes to be transferred (250_{10})
C Holds the address of the I/O device (5)
OUTD Outputs 250_{10} bytes of data to I/O device 5, starting at location 1000_{10}

The last group of instructions include miscellaneous CPU control instructions to halt the microprocessor, enable and disable interrupt, perform a "no-operation" instruction and three instructions (not in the 8080) to set several types of interrupt modes.

In next month's column, we will examine the different ways in which the Z-80 can address memory locations. **R-E**