Table 4.4 Flag Names in DEBUG

	Flag Name	Reset (0)	Set (1)
Status	Auxiliary carry	NA	AC
	Carry	NC	CY
	Overflow	NV	OV
	Parity	PO	PE
	Sign	PL	NG
	Zero	NZ	ZR
Control	Interrupt	DI	EI
	Direction	UP	DN

The value of each flag bit—in mnemonic form—is also shown. For example, NV indicates no overflow (OF = 0). Table 4.4 lists the mnemonic associated with each flag value.

Finally, recognizing that the current instruction references memory location 200, DEBUG displays the data at this location (DS:0200 = 10—the number 10 we entered previously).[15]

Running the Program. Continuing with Figure 4.9, we type **G** (for Go). The program runs and displays the message *Program terminated normally*. With DEBUG back in control, we type D200 L10—display 10H bytes beginning at address 200H. Studying the last lines of Figure 4.9, notice that locations 202 and 203 now contain the result of the multiplication in *little endian* format. Reversing the order of the bytes (0320) and converting to decimal (800), we see that the program has indeed run correctly.

Single-Stepping the Program. DEBUG's GO command causes the computer to run at full speed until the end of the program is encountered. For debugging and testing purposes, however, it may be more useful to *single-step* through the program one instruction at a time. Figure 4.10 shows an example for the multiplication program we have been discussing. We begin by entering new data for N1 (4DH) and N2 (29H) in locations 200 and 201. Next, the R command is used to confirm that register IP is pointing at the start of the program. Now we are ready to begin tracing. Typing **T**, the first instruction (MOV AL,[0200]) is executed, the CPU registers and flags are updated and redisplayed, and control returns to DEBUG. Notice that IP has incremented to 0103 and register AL now stores 4D, the first number to be multiplied. Typing **T** again causes the multiply instruction to execute, IP advances to 0107, and the product (0C55) appears in register AX. One more trace and the contents of register AX is written to memory. IP advances to 010A, the site of the INT 20 instruction. It is best not to single-step this instruction, as it jumps back into the operating system and executes many instructions before returning control to DEBUG.

Saving the Program. The program we have created resides in RAM and will be lost when we quit DEBUG. To save the program, three steps are required, as shown in Figure 4.11(a). First we use the **N** (Name) command to give the program a name—

[15]Recall that when accessing general data the data segment is used. Because all of the segments in this program overlap, location 200 in the data segment (DS:200) is the same as location 200 in the code segment (CS:200).