Chapter 7 showed how floating-point values can be placed onto the FPU stack using the `FLD` instruction, integers using the `FILD` instruction, and BCD data using the `FBLD` instruction. Various floating-point constant values are also available to load constant values into the stack. There are also commands for storing the values in the FPU register into memory locations in each of the different data types.

The FPU status, control, and tag registers

Because the FPU is independent of the main processor, it does not normally use the `EFLAGS` register to indicate results and determine behavior. The FPU contains its own set of registers to perform these functions. The status, control, and tag registers are used to access features and determine the status of the FPU.

This section describes these three FPU registers and shows how to access them in your programs.

The status register

The status register indicates the operating condition of the FPU. It is contained in a 16-bit register, with different bits assigned as different flags. The following table describes the status register bits.

Status Bit	Description
0	Invalid operation exception flag
1	Denormalized operand exception flag
2	Zero divide exception flag
3	Overflow exception flag
4	Underflow exception flag
5	Precision exception flag
6	Stack fault
7	Error summary status
8	Condition code bit 0 (C0)
9	Condition code bit 1 (C1)
10	Condition code bit 2 (C2)
11-13	Top of stack pointer
14	Condition code bit 3 (C3)
15	FPU busy flag

The four condition code bits (8, 9, 10, and 14) are used together to indicate specific error codes from the result of floating-point operations. They are often used with the exception flags to indicate a specific exception condition. You will see more of these bits in action later in this chapter.

The first six bits are the FPU exception flags. They are set by the FPU when a floating-point exception has occurred during processing. The flags remain set until a program manually clears them. The stack