The registers are depicted in Figure 2.12. R0 to R12 are general purpose registers and contain either data or addresses. Register R13 (also called the stack pointer, SP) points to the top element of the stack. Register R14 (also called the link register, LR) is used to store the return location for functions. The LR is also used in a special way during exceptions, such as interrupts. Interrupts are covered in Chapter 12. Register R15 (also called the program counter, PC) points to the next instruction to be fetched from memory. The processor fetches an instruction using the PC and then increments the PC.
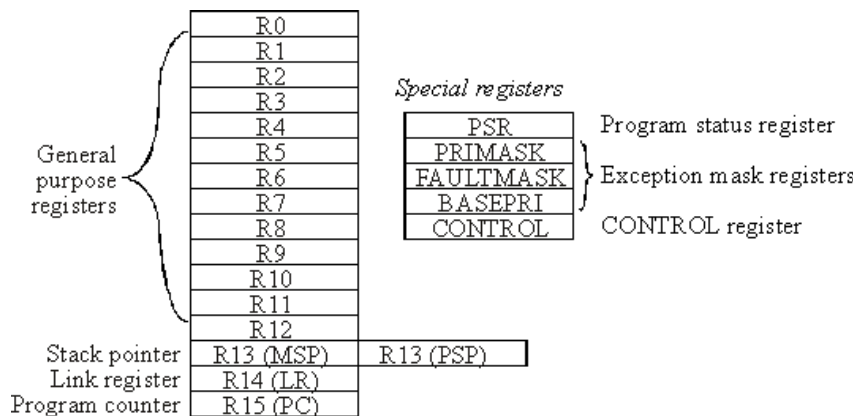

Figure 2.12. Registers on the ARM® Cortex-M processor.

The ARM Architecture **Procedure Call Standard**, AAPCS, part of the ARM **Application Binary Interface** (ABI), uses registers R0, R1, R2, and R3 to pass input parameters into a C function. Also according to AAPCS we place the return parameter in Register R0. In this class, the SP will always be the main stack pointer (MSP), not the Process Stack Pointer (PSP).

There are three status registers named Application Program Status Register (APSR), the Interrupt Program Status Register (IPSR), and the Execution Program Status Register (EPSR) as shown in Figure 2.13. These registers can be accessed individually or in combination as the **Program Status Register** (PSR). The N, Z, V, C, and Q bits give information about the result of a previous ALU operation. In general, the **N bit** is set after an arithmetical or logical operation signifying whether or not the result is negative. Similarly, the **Z bit** is set if the result is zero. The **C bit** means carry and is set on an unsigned overflow, and the **V bit** signifies signed overflow. The **Q bit** indicates that "saturation" has occurred – while you might want to look it up, saturated arithmetic is beyond the scope of this class.
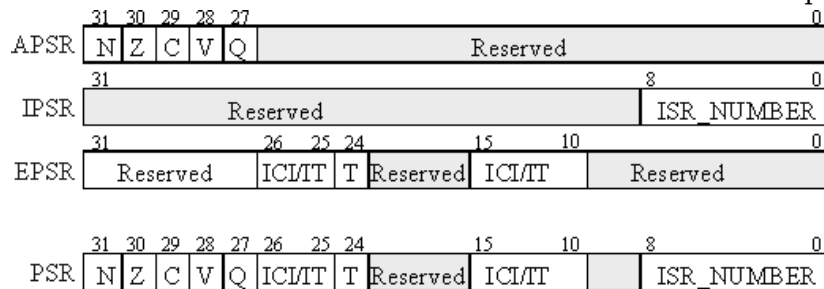
Figure 2.13. The program status register of the ARM® Cortex-M processor.

The **T bit** will always be 1, indicating the ARM®Cortex™-M processor is executing Thumb® instructions. The ISR_NUMBER indicates which interrupt if any the processor is handling. Bit 0 of the special register **PRIMASK** is the interrupt mask bit. If this bit is 1, most interrupts and exceptions are not allowed. If the bit is 0, then interrupts are allowed. Bit 0 of the special register **FAULTMASK** is the fault mask bit. If this bit is 1, all interrupts and faults are not allowed. If the bit is 0, then interrupts and faults are allowed. The nonmaskable interrupt (NMI) is not affected by these mask bits. The **BASEPRI** register defines the priority of the executing software. It prevents interrupts with lower or equal priority but allows higher priority interrupts. For example if **BASEPRI** equals 3, then requests with level 0, 1, and 2 can interrupt, while requests at levels 3 and higher will be postponed. A lower number means a higher priority interrupt. The details of interrupt processing will be presented in subsequent chapters.

Help