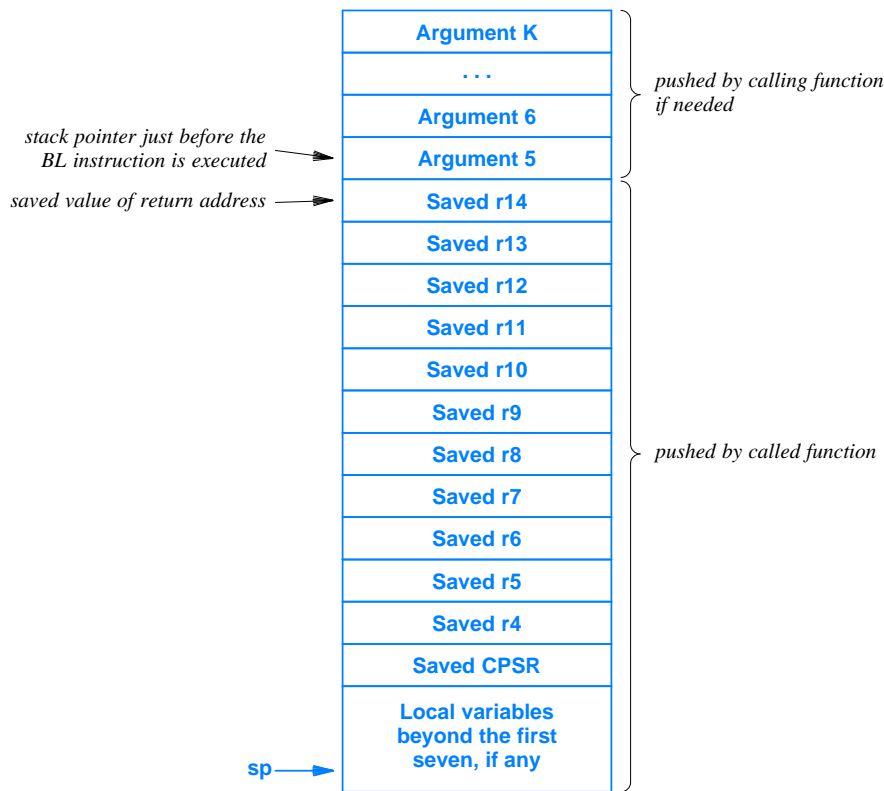


### 3.9.2 BeagleBone Black (ARM)

On an ARM processor, the first four arguments to a function are passed in registers  $a1 - a4$  (i.e., registers  $r0 - r3$ ). Arguments beyond the first four are passed on the stack. To call a function, the caller executes a *BL* (branch and link) instruction. When the *BL* executes, the hardware places the return address in register  $r14$ . The called function must save registers it will use; typically, the called function pushes copies of  $r14 - r4$  onto the stack, and then pushes the *CPSR* (status) register. Figure 3.5 illustrates values on the top of the stack immediately after a function call (remember that a stack grows downward). In the figure, the top of the stack only contains space for local variables beyond the first seven. By convention, the first seven local variables are stored in registers 4 through 8, 10, and 11.

In fact, the set of saved registers varies across versions of the ARM architecture. Furthermore, if a program uses floating point, the floating point registers must also be saved and restored.



**Figure 3.5** The stack on an ARM processor when a function is called.