

# ARMv7 Function Calling Conventions

In general, applications built for the ARMv7 environment are capable of running in the ARMv6 environment and vice versa. This is because the calling conventions for the ARMv7 environment are nearly identical to those found in the ARMv6 environment. Therefore, this article describes only the places where the ARMv7 environment deviates or extends the ARMv6 environment.

For detailed information about the calling conventions for the ARMv6 environment, see ARMv6 Function Calling Conventions.

## Function Calls

This section details the process of calling a subroutine and passing parameters to it, and how subroutines return values to their callers. The following sections highlight minor differences between the ARMv6 and ARMv7 environments.

### Prologs and Epilogs

The version of Thumb available in ARMv7 is capable of saving and restoring the contents of the VFP registers. In addition, ARMv7 assembly uses a unified set of mnemonics; therefore, the same assembly code can be used to generate either ARM or Thumb machine code.

Listing 1 shows an example of a prolog that saves a number of key registers, including several VFP registers. It also allocates 36 bytes for local storage.

**Listing 1** Example prolog for ARM or Thumb-2 (ARMv7)

```
push    {r4-r7, lr}           // save LR, R7, R4-R6
add     r7, sp, #12           // adjust R7 to point to saved R7
push    {r8, r10, r11}        // save remaining GPRs (R8, R10, R11)
vstmdb  sp!, {d8-d15}          // save VFP/Advanced SIMD registers D8
                                     // (aka S16-S31, Q4-Q7)
sub     sp, sp, #36           // allocate space for local storage
```

Listing 2 shows an example epilog that restores the registers saved by the preceding prolog.

**Listing 2** Example epilog for ARM or Thumb-2 (ARMv7)

```
add     sp, sp, #36           // deallocate space for local storage
vldmia  sp!, {d8-d15}          // restore VFP/Advanced SIMD registers
pop     {r8, r10, r11}         // restore R8-R11
pop     {r4-r7, pc}           // restore R4-R6, saved R7, return to saved LR
```

### Register Preservation

Register preservation on the ARMv7 architecture is the same as that for the ARMv6 architecture, except for the changes and additions noted in Table 1.

**Table 1** Additional processor registers in the ARMv7 architecture

Type	Name	Preserved	Notes
VFP register	D0–D7	No	Also known as Q0–Q3 on ARMv7. These registers are accessible from Thumb mode on ARMv7.
	D8–D15	Yes	Also known as Q4–Q7 on ARMv7. These registers are accessible from Thumb mode on ARMv7.
	D16–D31	No	Only available in ARMv7. Also known as Q8–Q15.

For information about all other registers, see Table 2.

---

Copyright © 2013 Apple Inc. All Rights Reserved. Terms of Use | Privacy Policy | Updated: 2013–09–18