Architectures, Processors, and Devices

Development Article

ARM®

ARM DHT 0001A

Architectures, Processors, and Devices
Development Article

Release Information

The following changes have been made to this book.

Change history

Date	Issue	Confidentiality	Change
19 May 2009	A	Non-Confidential	Issue 1

Proprietary Notice

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by ARM and the party that ARM delivered this document to.

Unrestricted Access is an ARM internal classification.

Product Status

The information in this document is final, that is for a developed product.

Web Address

`http://www.arm.com`

Contents
Architectures, Processors, and Devices Development Article

Chapter 1
Architectures, Processors, and Devices

This article introduces and explains the terms architecture, processor, and device as they apply to ARM-based systems. It contains the following sections:

1.1 The ARM processor business model

ARM does not manufacture processor hardware. Instead, ARM creates microprocessor designs that are licensed to our customers, who integrate them into *System-on-Chip* (SoC) devices.

To guarantee interoperability and to provide a common programmers model between different implementations, ARM defines architecture specifications that define how ARM products must operate. Additionally, some partners license the right to implement their own ARM processors conforming to the architecture specifications. This leads to a hierarchical split into three levels of specifications which together describe the behavior and programmers model of the entire SoC:

Architecture An architecture defines behavior that is common to many processor designs.

Processor A processor is an implementation of an architecture, and can be integrated into several different designs.

Device A device contains a processor and additional components.

Developers who are new to this model can find it difficult to find information about system-level behavior. This is because the information can be divided between several documents:

- the architecture reference manual
- the processor technical reference manual
- the device documentation.

This article identifies the borders between these levels of specification, and where to look for information.

1.2 Architecture

The ARM architecture defines how an ARM processor must operate. This includes:

- the programmers model
- the instruction set
- system configuration
- exception handling
- the memory model.

New versions of the architecture can introduce changes or additions to any of these features. These changes are normally backwards compatible, to simplify migration to new processors of more recent architectures.

Every version of the architecture specifies a number of system features and behaviors as implementation-defined. For example:

- how many levels of cache that are implemented, and their sizes
- all functionality provided by the Auxiliary Control Register
- effects of hint instructions.

An architecture can also define optional extensions, for example:

- floating point hardware support (VFP), introduced in ARMv6
- Advanced SIMD support (NEON), introduced in ARMv7.

Earlier versions of the ARM Architecture describe a common architecture, with different implementation choices. For example, a processor can be implemented with a *Virtual Memory System Architecture* (VMSA), based on a *Memory Management Unit* (MMU), or a *Protected Memory System Architecture* (PMSA), based on a *Memory Protection Unit* (MPU).

ARMv7 introduces the concept of Architecture *profiles*, defining versions of the architecture aimed at different types of processors for different market segments. The defined profiles are:

A The *Application* profile defines a VMSA based microprocessor architecture. It is targeted at high performance processors, capable of running full feature operating systems. It supports the ARM and Thumb instruction sets.

R The *Real-time* profile defines a PMSA based microprocessor architecture. It is targeted at systems that require deterministic timing and low interrupt latency. It supports the ARM and Thumb instruction sets.

M The *Microcontroller* profile provides low-latency interrupt processing accessible directly from high-level programming languages. It has a different exception handling model to the other profiles, implements a variant of the PMSA, and supports a variant of the Thumb instruction set only.

The *ARM Architecture Reference Manual ARMv7-A and ARMv7-R edition* describes the ARMv7-A and ARMv7-R profiles. It also documents the differences between ARMv7 and:

- ARMv4
- ARMv5
- ARMv6

The *ARMv7-M Architecture Reference Manual* describes the ARMv7-M profile. The *ARMv6-M Architecture Reference Manual* describes the ARMv6-M profile, which implements a subset of the ARMv7-M profile and uses the same exception model.

These documents are available on request from `http://infocenter.arm.com`.

1.3 Processor

A processor is implemented to comply with a defined version of the architecture. For example, ARM926EJ-S™ implements ARMv5 with the TEJ extensions, and Cortex™-A9 implements ARMv7-A with the multiprocessing extensions.

The *Technical Reference Manual* (TRM) of a processor describes all the features and implementation choices for the processor that are implementation-defined in its architecture specification.

Typically, there are many different implementations of the same architecture version. For example, the ARM7TDMI™ and ARM920T™ processors both implement ARMv4T, but the implementations differ in many ways.

Table 1-1 shows some recent ARM processors and the version and variant of the architecture they implement.

Table 1-1 Recent ARM processors and their architectures

Implementation	Architecture version	Architecture variant
ARM11™ MPCore™	ARMv6	ARMv6K, Improved multiprocessing support
ARM1156T2F-S™	ARMv6	ARMv6T2, Thumb-2 technology
ARM1176JZF-S™	ARMv6	ARMv6Z, ARMv6K with Security Extensions
Cortex-A9	ARMv7-A	
Cortex-R4	ARMv7-R	
Cortex-M3	ARMv7-M	

You can download ARM TRMs from `http://infocenter.arm.com`.

If ARM markets different versions of a processor, with and without functional extensions, it might provide separate TRMs for the processor and for the extensions. For example, the Cortex-A9 processor has separate TRMs for the:

- processor
- MPCore features
- Floating Point Unit
- NEON unit.

Contact the architecture implementer for documentation or support related to third-party implementations of the ARM architecture.

1.4 Device

A device is usually a SoC that incorporates an ARM processor and additional components. At the time of device implementation, options such as cache sizes, floating-point hardware support can be selected. This means that different devices based on the same ARM processor can have different cache sizes.

For example, a generic SoC might include several of the following components:

- *Level 2 Cache Controller* (L2CC)
- *Static Memory Controller* (SMC)
- *Dynamic Memory Controller* (DMC)
- bus interconnect
- interrupt controller
- timer
- external bus interfaces.

Some components central to the functionality of your device can be completely unrelated to both the ARM architecture and the processor used. This means that information about how to configure these components can be found only from the device documentation.

Documentation for ARM Fabric components such as Primecell® peripherals or bus interconnect solutions is available from `http://infocenter.arm.com`.

Note

Information about what ARM processor is used in a specific device can be found in the device documentation. There are many devices based around for example the ARM7TDMI processor, but they all have the same pipeline, instruction set and processor configuration registers.

Figure 1-1 on page 1-7 shows a block diagram of the ARM1176JZF-S development chip, produced by ARM in collaboration with a semiconductor partner. This device is a reference implementation, and is used on the ARM RealView® Platform Baseboard for ARM1176JZF-S.

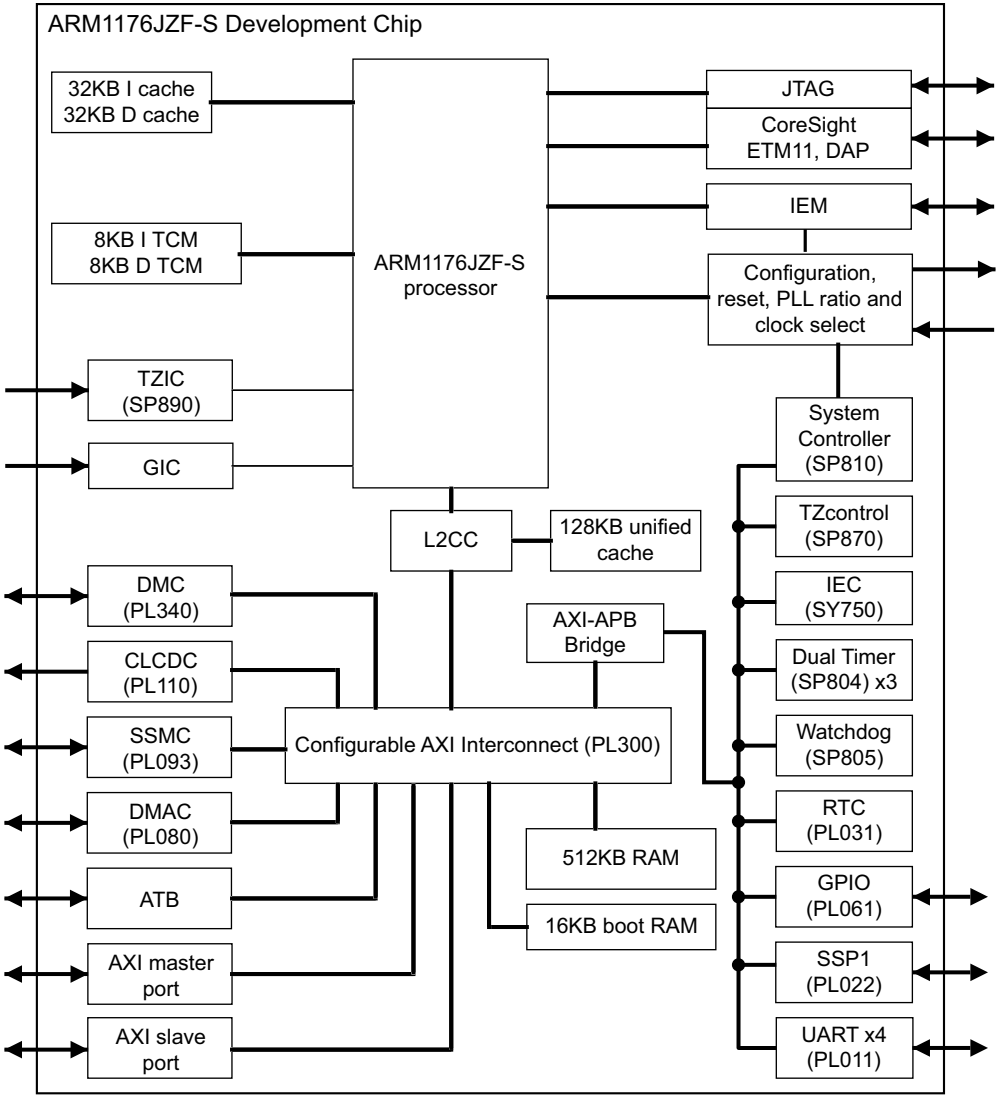

Figure 1-1 ARM1176JZF-S development chip block diagram

1.5 Putting it all together

When you design or develop your system, there might be three or more different sources of information that together describe the behavior of your ARM target. Depending on the nature of your question, you might have to combine two or more of these to cover all aspects of a specific behavior.

For example, if investigating a memory corruption problem on a target, you might have to combine information from documentation that describes:

- the architecture
- the ARM processor
- the device
- the bus interconnect used in the device
- the external cache controllers
- the memory controllers
- the memory devices.

Hopefully this article has helped in providing insight about where you are likely to find answers to questions that arise from your ARM-based design.