《ARM 嵌入式体系结构与接口技术》

作者：华清远见

第 2 章 ARM 技术概述

本章目标

ARM 体系结构的处理器在嵌入式中的应用是非常广泛的，本章将向读者介绍 ARM 处理器的基本知识。通过阅读本章，读者将了解以下主要内容：

- ARM 体系结构的技术特征及发展
- ARM 微处理器简介
- ARM 微处理器结构
- ARM 微处理器的应用选型
- ARM920T 内部功能及特点
- 数据类型
- ARM920T 存储系统
- 流水线
- 寄存器组织
- 程序状态寄存器
- 三星 S3C2410X 处理器介绍

2.1 ARM 体系结构的技术特征及发展

ARM（Advanced RISC Machines）有 3 种含义，它是一个公司的名称，是一类微处理器的通称，还是一种技术的名称。

2.1.1 ARM 公司简介

1991 年 ARM 公司（Advanced RISC Machine Limited）成立于英国剑桥，最早由 Arcon、Apple 和 VLSI 合资成立，主要出售芯片设计技术的授权，在 1985 年 4 月 26 日，第一个 ARM 原型在英国剑桥的 Acorn 计算机有限公司诞生（在美国 VLSI 公司制造）。目前，ARM 架构处理器已在高性能、低功耗、低成本的嵌入式应用领域中占据了领先地位。

ARM 公司最初只有 12 人，经过十多年的发展，ARM 公司已拥有近千名员工，在许多国家都设立了分公司，包括 ARM 公司在中国上海的分公司。目前，采用 ARM 技术知识产权（IP）核的微处理器，即我们通常所说的 ARM 微处理器，已遍及工业控制、消费类电子产品、通信系统、网络系统、无线系统等各类产品市场，基于 ARM 技术的微处理器应用约占据了 32 位 RISC 微处理器 80%以上的市场份额，其中，在手机市场，ARM 占有绝对的垄断地位。可以说，ARM 技术正在逐步渗入到人们生活中的各个方面，而且随着 32 位 CPU 价格的不断下降和开发环境的不断成熟，ARM 技术会应用得越来越广泛。

ARM 公司是专门从事基于 RISC 技术芯片设计开发的公司，作为嵌入式 RISC 处理器的知识产权 IP 供应商，公司本身并不直接从事芯片生产，而是靠转让设计许可由合作公司生产各具特色的芯片，世界各大半导体生产商从 ARM 公司购买其设计的 ARM 微处理器核，根据各自不同的应用领域，加入适当的外围电路，从而形成自己的 ARM 微处理器芯片进入市场，利用这种合伙关系，ARM 很快成为许多全球性 RISC 标准的缔造者。目前，全世界有几十家大的半导体公司都使用 ARM 公司的授权，其中包括 Intel、IBM、Samsung、LG 半导体、NEC、SONY、PHILIP 等公司，这也使得 ARM 技术获得更多的第三方工具、制造、软件的支持，又使整个系统成本降低，使产品更容易进入市场并被消费者所接受，更具有竞争力。

2.1.2 ARM 技术特征

ARM 的成功，一方面得益于它独特的公司运作模式，另一方面，当然来自于 ARM 处理器自身的优良性能。作为一种先进的 RISC 处理器，ARM 处理器有如下特点。

- 体积小、低功耗、低成本、高性能。
- 支持 Thumb（16 位）/ARM（32 位）双指令集，能很好的兼容 8 位/16 位器件。
- 大量使用寄存器，指令执行速度更快。
- 大多数数据操作都在寄存器中完成。
- 寻址方式灵活简单，执行效率高。
- 指令长度固定。

此处有必要解释下 RISC 处理器的概念及与 CISC 微处理器的区别

（1）嵌入式 RISC 微处理器

RISC（reduced instruction set computer）是精简指令集计算机，RISC 把着眼点放在如何使计算机的结构更加简单和如何使计算机的处理速度更加快速上。RISC 选取了使用频率最高的简单指令，抛弃复杂指令，固定指令长度，减少指令格式和寻址方式，不用或少用微码控制。这些特点使得 RISC 非常适合嵌入式处理器。

（2）嵌入式 CISC 微处理器

传统的复杂指令级计算机（CISC）则更侧重于硬件执行指令的功能性，使 CISC 指令、以及处理器的

硬件结构变得更复杂。这些会导致成本、芯片体积的增加，影响其在嵌入式产品的应用。在嵌入表 2-1 所示描述了 RISC 和 CISC 之间的主要区别。

表 2-1　　RISC 和 CISC 之间主要的区别

指　　标	RISC	CISC
指令集	一个周期执行一条指令，通过简单指令的组合实现复杂操作；指令长度固定	指令长度不固定，执行需要多个周期
流水线	流水线每周期前进一步	指令的执行需要调用微代码的一个微程序
寄存器	更多通用寄存器	用于特定目的的专用寄存器
Load/Store 结构	独立的 Load 和 Store 指令完成数据在寄存器和外部存储器之间的传输	处理器能够直接处理存储器中的数据

2.1.3　ARM 体系结构的发展

在讨论 ARM 体系结构前，先解释下体系结构的定义。

体系结构，定义了指令集（ISA）和基于这一体系结构下处理器的编程模型。基于同种体系结构可以有多种处理器，每个处理器性能不同，所面向的应用不同，每个处理器的实现都要遵循这一体系结构。ARM 体系结构为嵌入系统发展商提供很高的系统性能，同时保持优异的功耗和面积效率。

ARM 体系结构为满足 ARM 合作者以及设计领域的一般需求正稳步发展。目前，ARM 体系结构共定义了 7 个版本，从版本 1 到版本 7，ARM 体系的指令集功能不断扩大，不同系列的 ARM 处理器，性能差别很大，应用范围和对象也不尽相同，但是，如果是相同的 ARM 体系结构，那么基于它们的应用软件是兼容的。

1．V1 结构

V1 版本的 ARM 处理器并没有实现商品化，采用的地址空间是 26 位，寻址空间是 64MB，在目前的版本中已不再使用这种结构。

2．V2 结构

与 V1 结构的 ARM 处理器相比，V2 结构的 ARM 处理器的指令结构要有所完善，比如增加了乘法指令并且支持协处理器指令，在该版本的处理器仍然是 26 位的地址空间。

3．V3 结构

从 V3 结构开始，ARM 处理器的体系结构有了很大的改变，实现了 32 位的地址空间，指令结构相对前面的两种结构也所完善。

4．V4 结构

V4 结构的 ARM 处理器增加了半字指令的读取和写入操作，增加了处理器系统模式，并且有了 T 变种-V4T，在 Thumb 状态下所支持的是 16 位的 Thumb 指令集。属于 V4T（支持 Thumb 指令）体系结构的处理器（核）有 ARM7TDMI，ARM7TDMI-S（ARM7TDMI 可综合版本），ARM710T（ARM7TDMI 核的处理器），ARM720T（ARM7TDMI 核的处理器），ARM740T（ARM7TDMI 核的处理器），ARM9TDMI，ARM910T（ARM9TDMI 核的处理器），ARM920T（ARM9TDMI 核的处理器），ARM940T（ARM9TDMI 核的处理器），StrongARM（Intel 公司的产品）。

5．V5 结构

V5 结构的 ARM 处理器提升了 ARM 和 Thumb 两种指令的交互工作能力，同时有了 DSP 指令-V5E 结构、Java 指令-V5J 结构的支持。

属于 V5T（支持 Thumb 指令）体系结构的处理器（核）有 ARM10TDMI，ARM1020T（ARM10TDMI 核处理器）。

属于 V5TE（支持 Thumb，DSP 指令）体系结构的处理器（核）有 ARM9E，ARM9E-S（ARM9E 可综合版本），ARM946（ARM9E 核的处理器），ARM966（ARM9E 核的处理器），ARM10E，ARM1020E（ARM10E 核处理器），ARM1022E（ARM10E 核的处理器），Xscale（Intel 公司产品）。

属于 V5TEJ（支持 Thumb，DSP 指令，Java 指令）体系结构的处理器（核）有 ARM9EJ，ARM9EJ-S（ARM9EJ 可综合版本），ARM926EJ（ARM9EJ 核的处理器），ARM10EJ。

6．V6 结构

V6 结构是在 2001 年发布的，在该版本中增加了媒体指令，属于 V6 体系结构的处理器核有 ARM11（2002 年发布）。V6 体系结构包含 ARM 体系结构中所有的 4 种特殊指令集：Thumb 指令（T）、DSP 指令（E）、Java 指令（J）和 Media 指令。

7．V7 结构

ARMv7 架构是在 ARMv6 架构的基础上诞生的。该架构采用了 Thumb-2 技术，它是在 ARM 的 Thumb 代码压缩技术的基础上发展起来的，并且保持了对现存 ARM 解决方案的完整的代码兼容性。Thumb-2 技术比纯 32 位代码少使用 31%的内存，减小了系统开销，同时能够提供比已有的基于 Thumb 技术的解决方案高出 38%的性能。ARMv7 架构还采用了 NEON 技术，将 DSP 和媒体处理能力提高了近 4 倍。并支持改良的浮点运算，满足下一代 3D 图形、游戏物理应用以及传统嵌入式控制应用的需求。

Cortex 系列处理器是基于 ARMv7 架构的，分为 Cortcx-M3、Cortex-R 和 Cortex-A 三类。本章的 2.28 节将会例举一些 Cortex 的特性。

2.2 ARM 微处理器简介

ARM 处理器的产品系列非常广，包括 ARM7、ARM9、ARM9E、ARM10E、ARM11 和 SecurCore、Cortex 等。每个系列提供一套特定的性能来满足设计者对功耗、性能、体积的需求。SecurCore 是单独一个产品系列，是专门为安全设备而设计的。

表 2-2 总结了 ARM 各系列处理器所包含的不同类型。

表 2-2 ARM 各系列处理器所包含的不同类型

ARM 系列	包 含 类 型
ARM7 系列	ARM7EJ-S ARM7TDMI ARM7TDMI-S ARM720T

ARM9/9E 系列	ARM920T ARM922T ARM926EJ-S ARM940T ARM946E-S ARM966E-S ARM968E-S
向量浮点运算（Vector Floating Point）系列	VFP9-S VFP10
ARM10E 系列	ARM1020E ARM1022E ARM1026EJ-S
ARM11 系列	ARM1136J-S ARM1136JF-S ARM1156T2(F)-S ARM1176JZ(F)-S ARM11 MPCore
Cortex 系列	Cortex-A Cortex-R Cortex-M
SecurCore 系列	SC100 SC110 SC200 SC210
其他合作伙伴产品	StrongARM XScale MBX

本节简要介绍 ARM 各个系列处理器的特点。

2.2.1 ARM7 处理器系列

ARM7 内核采用冯·诺伊曼体系结构，数据和指令使用同一条总线。内核有一条 3 级流水线，执行 ARMv4 指令集。

ARM7 系列处理器主要用于对功耗和成本要求比较苛刻的消费类产品。其最高主频可以到达 130MIPS。ARM7 系列包括 ARM7TDMI、ARM7TDMI-S、ARM7EJ-S 和 ARM720T 四种类型，主要用于适应不同的市场需求。

ARM7 系列处理器主要具有以下特点：

（1）成熟的大批量的 32 位 RICS 芯片；

（2）最高主频达到 130MIPS；

（3）功耗低；

（4）代码密度高，兼容 16 位微处理器；

（5）开发工具多，EDA 仿真模型多；

（6）调试机制完善；

（7）提供 0.25μm、0.18μm 及 0.13μm 的生产工艺；

（8）代码与 ARM9 系列、ARM9E 系列及 ARM10E 系列兼容。

ARM7 系列处理器主要应用于下面一些场合：

（1）个人音频设备（MP3 播放器、WMA 播放器、AAC 播放器）；

（2）接入级的无线设备；

（3）喷墨打印机；

（4）数码照相机；

（5）PDA。

2.2.2 ARM9 处理器系列

ARM9 系列于 1997 年问世。由于采用了 5 级指令流水线，ARM9 处理器能够运行在比 ARM7 更高的时钟频率上，改善了处理器的整体性能；存储器系统根据哈佛体系结构（程序和数据空间独立的体系结构）重新设计，区分了数据总线和指令总线。

ARM9 系列的第一个处理器是 ARM920T，它包含独立的数据指令 Cache 和 MMU（Memory Management Unit，存储器管理单元）。此处理器能够被用在要求有虚拟存储器支持的操作系统上。该系列中的 ARM922T 是 ARM920T 的变种，只有一半大小的数据指令 Cache。

ARM940T 包含一个更小的数据指令 Cache 和一个 MPU（Micro Processor Unit，微处理器）。它是针对不要求运行操作系统的应用而设计的。ARM920T、ARM940T 都执行 v4T 架构指令。

ARM9 系列处理器主要应用于下面一些场合：

（1）下一代无线设备，包括视频电话和 PDA 等；

（2）数字消费品，包括机顶盒、家庭网关、MP3 播放器和 MPEG-4 播放器；

（3）成像设备，包括打印机、数码照相机和数码摄像机；

（4）汽车、通信和信息系统。

2.2.3 ARM9E 处理器系列

ARM9 系列的下一代处理器基于 ARM9E-S 内核，这个内核是 ARM9 内核带有 E 扩展的一个可综合版本，包括 ARM946E-S 和 ARM966E-S 两个变种。两者都执行 v5TE 架构指令。它们也支持可选的嵌入式跟踪宏单元，支持开发者实时跟踪处理器上指令和数据的执行。当调试对时间敏感的程序段时，这种方法非常重要。

ARM946E-S 包括 TCM（Tightly Coupled Memory，紧耦合存储器）、Cache 和一个 MPU。TCM 和 Cache 的大小可配置。该处理器是针对要求有确定的实时响应的嵌入式控制而设计的。ARM966E-S 有可配置的 TCM，但没有 MPU 和 Cache 扩展。

ARM9 系列的 ARM926EJ-S 内核为可综合的处理器内核，发布于 2000 年。它是针对小型便携式 Java 设备，如 3G 手机和 PDA 应用而设计的。ARM926EJ-S 是第一个包含 Jazelle 技术，可加速 Java 字节码执行的 ARM 处理器内核。它还有一个 MMU、可配置的 TCM 及具有零或非零等待存储器的数据/指令 Cache。

ARM9E 系列处理器主要应用于下面一些场合：

（1）下一代无线设备，包括视频电话和 PDA 等；

（2）数字消费品，包括机顶盒、家庭网关、MP3 播放器和 MPEG-4 播放器；

（3）成像设备，包括打印机、数码照相机和数码摄像机；

（4）存储设备，包括 DVD 或 HDD 等；

（5）工业控制，包括电机控制等；

（6）汽车、通信和信息系统的 ABS 和车体控制；

（7）网络设备，包括 VoIP、WirelessLAN 等。

2.2.4 ARM11 处理器系列

ARM1136J-S 发布于 2003 年，是针对高性能和高能效应而设计的。ARM1136J-S 是第一个执行 ARMv6 架构指令的处理器。它集成了一条具有独立的 Load/Stroe 和算术流水线的 8 级流水线。ARMv6 指令包含了针对媒体处理的单指令流多数据流扩展，采用特殊的设计改善视频处理能力。

2.2.5 SecureCore 处理器系列

SecurCore 系列处理器提供了基于高性能的 32 位 RISC 技术的安全解决方案。SecurCore 系列处理器除了具有体积小、功耗低、代码密度高等特点外，还具有它自己特别优势，即提供了安全解决方案支持。下面总结了 SecurCore 系列的主要特点：

（1）支持 ARM 指令集和 Thumb 指令集，以提高代码密度和系统性能；

（2）采用软内核技术以提供最大限度的灵活性，可以防止外部对其进行扫描探测；

（3）提供了安全特性，可以抵制攻击；

（4）提供面向智能卡和低成本的存储保护单元 MPU；

（5）可以集成用户自己的安全特性和其他的协处理器。

SecurCore 系列包含 SC100、SC110、SC200 和 SC210 四种类型。

SecureCore 系列处理器主要应用于一些安全产品及应用系统，包括电子商务、电子银行业务、网络、移动媒体和认证系统等。

2.2.6 StrongARM 和 Xscale 处理器系列

StrongARM 处理器最初是 ARM 公司与 Digital Semiconductor 公司合作开发的，现在由 Intel 公司单独许可，在低功耗、高性能的产品中应用很广泛。它采用哈佛架构，具有独立的数据和指令 Cache，有 MMU。StrongARM 是第一个包含 5 级流水线的高性能 ARM 处理器，但它不支持 Thumb 指令集。

Intel 公司的 Xscale 是 StrongARM 的后续产品，在性能上有显著改善。它执行 v5TE 架构指令，也采用哈佛结构，类似于 StrongARM 也包含一个 MMU。前面说过，Xscale 已经被 Intel 卖给了 Marvell 公司。

2.2.7 MPCore 处理器系列

MPCore 在 ARM11 核心的基础上构建，架构上仍属于 V6 指令体系。根据不同的需要，MPCore 可以被配置为 1 到 4 个处理器的组合方式，最高性能达到 2600 Dhrystone MIPS，运算能力几乎与 Pentium III 1GHz 处于同一水准（Pentium III 1GHz 的指令执行性能约为 2700 Dhrystone MIPS）。多核心设计的优点是在频率不变的情况下让处理器的性能获得明显提升，在多任务应用中表现尤其出色，这一点很适合未来家庭消费电子的需要。例如，机顶盒在录制多个频道电视节目的同时，还可通过互联网收看数字视频点播节目；车内导航系统在提供导航功能的同时，可以向后座乘客提供各类视频娱乐信息等。在这类应用环境下，多核心结构的嵌入式处理器将表现出极强的性能优势。

2.2.8 Cortex 处理器系列

1、 ARM Correx 处理器技术特点

ARMv7 架构是在 ARMv6 架构的基础上诞生的。该架构采用了 Thumb-2 技术，它是在 ARM 的 Thumb 代码压缩技术的基础上发展起来的，并且保持了对现存 ARM 解决方案的完整的代码兼容性。Thumb-2 技术比纯 32 位代码少使用 31%的内存，减小了系统开销，同时能够提供比已有的基于 Thumb 技术的解决方案高出 38%的性能。ARMv7 架构还采用了 NEON 技术，将 DSP 和媒体处理能力提高了近 4 倍。并支持改良的浮点运算，满足下一代 3D 图形、游戏物理应用以及传统嵌入式控制应用的需求。此外，ARMv7 还支持改良的运行环境，以迎合不断增加的 JIT(Just In Time)和 DAC(DynamicAdaptlve Compilation)技术的使用。

在与早期的 ARM 处理器软件兼容性方面，ARMv7 架构在设计时充分考虑到了。ARM Cortex-M 系列支持 Thumb-2 指令集(Thumb 指令集的扩展集)，可以执行所有已存的为早期处理器编写的代码。通过一个前向的转换方式，为 ARM Cortex-M 系列处理器所写的用户代码可以与 ARM Cortex-R 系列微处理器完全兼容。ARMCortex-M 系列系统代码(如实时操作系统)可以很容易地移植到基于 ARMCortex-R 系列的系统上。ARMCortex-A 和 Cortex-R 系列处理器还支持 ARM 32 位指令集，向后完全兼容早期的 ARM 处理器，

包括从 1995 年发布的 ARM7TDMI 处理器到 2002 年发布的 ARMll 处理器系列。由于应用领域的不同，基于 v7 架构的 Cortex 处理器系列所采用的技术也不相同。在命名方式上，基于 ARMv7 架构的 ARM 处理器已经不再延用过去的数字命名方式，而是冠以 Cortex 的代呼。基于 v7A 的称为“Cortex-A 系州”，基于 v7R 的称为“Cortex-R 系列”，基于 v7M 的称为“Cortex-M3”。

2、ARM CorteX-M3 处理器技术特点

ARM Cortex-M3 处理器是为存储器和处理器的尺寸对产品成本影响极大的各种应用专门开发没计的。它整合了多种技术，减少使用内存，并在极小的 RISC 内核上提供低功耗和高性能，可实现由以往的代码向 32 位微控制器的快速移植。ARM Cortex-M3 处理器是使用最少门数的 ARM CPU，相对于过去的设计大大减小了芯片面积，可减小装置的体积或采用更低成本的工艺进行生产，仅 33000 门的内核性能可达 1，2DMIPS/MHz。此外，基本系统外设还具备高度集成化特点，集成了许多紧耦合系统外设，合理利用了芯片空间，使系统满足下一代产品的控制需求。

ARM Cortex-M3 处理器结合了执行 Thumb-2 指令的 32 位哈佛微体系结构和系统外设，包括 Nested Vectored Interrupt Controller 和 Arbiter 总线。该技术方案在测试和实例应用中表现出较高的性能：在台机电 180 nm 工艺下，芯片性能达 1．2 DMIPS／MHz，时钟频率高达 l00 MHz。Cortex-M3 处理器还实现了 Tail-Chaining 中断技术。该技术是一项完全基于硬件的中断处理技术，最多可减少 12 个时钟周期数，在实际应用中可减少 70％中断；推出了新的单线调试技术，避免使用多引脚进行 JTAG 调试，并全面支持 RealVicw 编译器和 RealView 调试产品。Realview 工具向设计者提供模拟、创建虚拟模型、编译软件、调试、验证和测试基于 ARMv7 架构的系统等功能。

为微控制器应用而开发的 Cortex-M3 拥有以下性能：

- 实现单周期 Flash 应用最优化；
- 准确快速地中断处理。永不超过 12 周期，仅 6 周期 tail-chaining(末尾连锁)；
- 有低功耗时钟门控(Clock Gating)的 3 种睡眠模式；
- 单周期乘法和乘法累加指令；
- ARM Thumb-2 混合的 16／32 位固有指令集，无模式转换；
- 包括数据观察点和 Flash 补丁在内的高级调试功能；
- 原子位操作，在一个单一指令中读取/修改／编写；
- 1．25DMIPS／MHz(与 0．9DMIPS／MHz 的 ARM7 和 1．1DMIPS／MHz 的 ARM9 相比)。

3、ARM Cortex-R4 处理器技术特点

Cortex-R4 处理器支持手机、硬盘、打印机及汽车电子设计，能协助新一代嵌入式产品快速执行各种复杂的控制算法与实时工作的运算；可通过内存保护单元(MPU，Memory Protection Unit)、高速缓存以及紧密耦合内存(TCM，Tightly Coupled Mennory)让处理器针对各种不同的嵌入式应用进行最佳化调整，且不影响基本的 ARM 指令集兼容性。这种设计能够在延用原有程序代码的情况下，降低系统的成本与复杂度，同时其紧密耦合内存功能也能提供更小的规格及更高效率的整合，并带来快速的响应时闻。

Cortex-R4 处理器采用 ARMv7 体系结构，让它能与现有的程序维持完全的回溯兼容性，能支持现今建立在全球各地数十亿的系统；并已针对 Thumb-2 指令进行最佳化设计。此项特性带来很多的利益，其中包括：更低的时钟速度所带来的省电效益；更高的性能将各种多功能特色带入移动电话与汽车产品的设计；更复杂的算法支持更高性能的数码影像与内建硬盘的系统。运用 Thumb-2 指令集，加上 RealView 开发套件，使芯片内部存储器的容量最多得以降低 30％，大幅降低系统成本，其速度比在 ARM9tt6E-S 处理器所使用的 Thumb 指令集高出 40％。由于存储器在芯片中的占用空间愈来愈多，因此这项设计将大幅节省芯片容量，让芯片制造商运用这款处理器开发各种 SoC(System on a Chip)器件。

相比于前几代的处理器，Cortex-R4 处理器高效率的设计方案，使其能以更低的时钟达到更高的性能；经过最佳化设计的 Artisan Mctro 内存，则进一步降低嵌入式系统的体积与成本。处理器搭载一个先进的微架构，具备双指令发送功能，采用 90nm 工艺并搭配 Artisan Advantage 程序库的组件，底面积不到 lmm2，耗电最低于 0．27mW／MHz，并能提供超过 600 DMIPS 的性能。

Cortex-R4 处理器在各种安全应用上加入容错功能和内存保护机制，支持最新版 OSEK 实时操作系统；支持 RealView Develop 系列软件开发工具、RealView Create 系列 ESL 工具与模块，以及

Core Sight 除错与追踪技术，协助设计者迅速开发各种嵌入式系统。

4、 ARM Cortex-A8 处理器技术特点

ARM Cortex-A8 处理器是一款适用于复杂操作系统及用户应用的应用处理器，支持智能能源管理(IEM，Inte11igent Energy Manger)技术的 ARM Artisan 库以及先进的泄漏控制技术，使得 Cortex-A8 处理器实现了非凡的速度和功耗效率。在 65nm 工艺下，ARM Conex-A8 处理器的功耗不到 300mw，能够提供高性能和低功耗。它第一次为低费用、高容量的产品带来了台式机级别的性能。

Conex-A8 处理器是第一款基于下一代 ARMv7 架构的应用处理器，使用了能够带来更高性能、更低功耗和更高代码密度的 Thumb-2 技术。它首次采用了强大的 NEON 信号处理扩展集，为 H．264 和 MP3 等媒体编解码提供加速。Cortex-A8 的解决方案还包括 Jazelle-RCTJava 加速技术，对实时(JTT)和动态调整编译(DAC)提供最优化，同时减少内存占用空间高达 3 倍。该处理器配置了先进的超标量体系结构流水线，能够同时执行多条指令。处理器集成了一个可调尺寸的二级高速缓冲存储器，能够同高速的 16KB 或者 32KB 一级高速缓冲存储器一起工作，从而达到最快的读取速度和最大的吞吐量。新处理器还配置了用于安全交易和数字版权管理的 Trust Zone 技术，以及实现低功耗管理的 IEM 功能。

Cortex-A8 处理器使用了先进的分支预测技术，并且具有专用的 NEON 整型和浮点型流水线进行媒体和信号处理。在使用小于 4mm2 的硅片及低功耗的 65 nm 工艺的情况下，Cortex-A8 处理器的运行频率将高于 600MHz(不包括 NEON 追踪技术和二级高速缓冲存储器)。在高性能的 90nm 和 65nm 工艺下，Cortex-A8 处理器运行频率最高可达 1GHz，能够满足高性能消费产品设计的需要。

2.3 ARM 微处理器结构

ARM 内核采用 RISC 体系结构。ARM 体系结构的主要特征如下（在本书的后续章节中将对这些特征做详细讲解）：

（1）大量的寄存器，它们都可以用于多种用途；

（2）Load/Store 体系结构；

（3）每条指令都条件执行；

（4）多寄存器的 Load/Store 指令；

（5）能够在单时钟周期执行的单条指令内完成一项普通的移位操作和一项普通的 ALU 操作；

（6）通过协处理器指令集来扩展 ARM 指令集，包括在编程模式中增加了新的寄存器和数据类型。

（7）如果把 Thumb 指令集也当作 ARM 体系结构的一部分，那么还可以加上：在 Thumb 体系结构中以高密度 16 位压缩形式表示指令集。

2.4 ARM 微处理器的应用选型

随着国内嵌入式应用领域的发展，ARM 芯片必然会获得广泛的重视和应用。但是由于 ARM 芯片有多达十几种的芯核结构、70 多芯片生产厂家以及千变万化的内部功能配置组合，开发人员在选择方案时会有一定的困难。所以对 ARM 芯片做对比研究是十分必要的。

2.4.1 ARM 芯片选择的一般原则

- 功能
  考虑处理器本身能够支持的功能，如：usb、网络、串口、液晶显示功能等。
- 性能
  从处理器的功耗、速度、稳定可靠性等方面考虑。
- 价格
  通常产品总是希望在完成功能要求的基础上，成本越低越好。在选择处理器时需要考虑处理的价

格，及由处理器衍生出的开发价格。如：开发板价格、处理器自身价格、外围芯片、开发工具、制版价格等。

- 熟悉程度及开发资源

通常公司对产品的开发周期都有严格的要求，选择一款自己熟悉的处理器可以大大降低开发风险。在自己熟悉的处理器都无法满足功能的情况下，可以尽量选择开发资源丰富的处理器。

- 操作系统支持

在选择嵌入式处理器时，如果最终的程序需要运行在操作系统上，那么还应该考虑处理器对操作系统的支持。

- 升级

很多产品在开发完成后都会面临升级的问题，正所谓人无远虑必有近优。所以在选择处理式必须要考虑升级的问题。如：尽量选择具有相同封装的不同性能等级的处理器；考虑产品未来可能增加的功能。

- 供货稳定

供货稳定也是选择处理器时的一个重要参考因素。尽量选择大厂家，比较通用的芯片。

2.4.2 选择一款适合高职、高专教学的 ARM 芯片

高职、高专教学中，在选择芯片作为学习目标时，主要从芯片功能、开发平台价格、开发资源等方面考虑。

- 芯片功能

（1）ARM 芯核

如果希望学习使用 Windows CE 或 Linux 等操作系统，就需要选择 ARM720T 以上带有 MMU（Memory Management Unit）功能的 ARM 芯片，ARM720T、StrongARM、ARM920T、ARM922T、ARM946T 都带有 MMU 功能。而 ARM7TDMI 没有 MMU，不支持 Windows CE 和大部分的 Linux；目前有 uCLinux 及 linux2.6 内核等 Linux 系统不需要 MMU 的支持。

（2）系统时钟控制器

系统时钟决定了 ARM 芯片的处理速度。ARM7 的处理速度为 0.97MIPS/MHz，常见的 ARM7 芯片系统主时钟为 20～133MHz，ARM9 的处理速度为 1.1MIPS/MHz，常见的 ARM9 的系统主时钟为 100～233MHz。ARM10、ARM11、CORTEX-A、CORTEX-R 系列的系统主时钟也越来越高。如果希望学习可以支持较为复杂的操作系统的芯片时，可以选择 ARM9 及 ARM9 以上的芯片

（3）内部存储器容量

在不需要大容量存储器时，可以考虑选用有内置存储器的 ARM 芯片。表 2-3 列出了内置存储器的 ARM 芯片。如果需要学习 Linux 或 wince 这样的操作系统，就需要外部存储器了。

表 2-3　　内置存储器的 ARM 芯片

芯 片 型 号	供　应　商	Flash 容量	ROM 容量	SDAM 容量
AT91F40162	ATMEL	2MB	256KB	4KB
AT91FR4081	ATMEL	1MB		128KB
SAA7750	Philips	384KB		64KB
PUC3030A	Micornas	256KB		56KB
LC67F500	Snayo	640KB		32KB

（4）USB 接口

USB 接口在产品的使用越来越广泛，许多 ARM 芯片内置有 USB 控制器，有些芯片甚至同时有 USB Host 和 USB Slave 控制器。表 2-4 显示了内置 USB 控制器的 ARM 芯片。

表 2-4　　内置 USB 控制器的 ARM 芯片

芯 片 型 号	ARM 内核	供　应　商	USB Slave	USB Host	IIS 接口

S3C2410	ARM920T	Samsung	1	2	1
S3C2400	ARM920T	Samsung	1	2	1
S5N8946	ARM7TDMI	Samsung	1	0	0
L7205	ARM720T	Linkup	1	1	0
L7210	ARM720T	Linkup	1	1	0
EP9312	ARM920T	Cirrus logic	0	3	1
Dragonball MX1	ARM920T	Motorola	1	0	1
SAA7750	ARM720T	Plilips	1	0	1
TMS320DSC2x	ARM7TDMI	TI	1	0	0
PUC3030A	ARM7TDMI	Micronas	1	0	5
ML67100	ARM7TDMI	OKI	1	0	0
ML7051LA	ARM7TDMI	OKI	1	0	0
SA-1100	StrongARM	Intel	1	0	0

（5）GPIO 数量

在某些芯片供应商提供的说明书中，往往申明的是最大可能的 GPIO 数量，但是有许多引脚是和地址线、数据线、串口线等引脚复用的。这样在系统设计时需要计算实际可以使用的 GPIO 数量。

（6）中断控制器

ARM 内核只提供快速中断（FIQ）和标准中断（IRQ）两个中断向量。但各个半导体厂家在设计芯片时加入了自己定义的中断控制器，以便支持诸如串行口、外部中断、时钟中断等硬件中断。外部中断控制是选择芯片必须考虑的重要因素，合理的外部中断设计可以很大程度地减少任务调度工作量。例如 PHILIPS 公司的 SAA7750，所有 GPIO 都可以设置成 FIQ 或 IRQ，并且可以选择上升沿、下降沿、高电平和低电平 4 种中断方式。这使得红外线遥控接收、指轮盘和键盘等任务都可以作为背景程序运行。而 Cirrus Logic 公司的 EP7312 芯片只有 4 个外部中断源，并且每个中断源都只能是低电平或高电平中断，这样在接收红外线信号的场合必须用查询方式，浪费大量 CPU 时间。

（7）IIS（Integrate Interface of Sound）接口

即集成音频接口。如果设计音频应用产品，IIS 总线接口是必需的。

（8）nWAIT 信号

这是一个外部总线速度控制信号。不是每个 ARM 芯片都提供这个信号引脚，利用这个信号与廉价的 GAL 芯片就可以实现与符合 PCMCIA 标准的 WLAN 卡和 Bluetooth 卡的接口，而不需要外加高成本的 PCMCIA 专用控制芯片。另外，当需要扩展外部 DSP 协处理器时，此信号也是必需的。

（9）RTC（Real Time Clock）

很多 ARM 芯片都提供实时时钟功能，但方式不同。如 Cirrus Logic 公司的 EP7312 的 RTC 只是一个 32 位计数器，需要通过软件计算出年月日时分秒；而 SAA7750 和 S3C2410 等芯片的 RTC 直接提供年月日时分秒格式。

（10）LCD 控制器

有些 ARM 芯片内置 LCD 控制器，有的甚至内置 64KB 彩色 TFT LCD 控制器。在设计 PDA 和手持式显示记录设备时，选用内置 LCD 控制器的 ARM 芯片（如 S3C2410）较为适宜。

（11）PWM 输出

有些 ARM 芯片有 2～8 路 PWM 输出，可以用于电机控制或语音输出等场合。

（12）ADC 和 DAC

有些 ARM 芯片内置 2～8 通道 8～12 位通用 ADC，可以用于电池检测、触摸屏和温度监测等。PHILIPS 的 SAA7750 更是内置了一个 16 位立体声音频 ADC 和 DAC，并且带耳机驱动。

（13）扩展总线

大部分 ARM 芯片具有外部 SDRAM 和 SRAM 扩展接口，不同的 ARM 芯片可以扩展的芯片数量即片选线数量不同，外部数据总线有 8 位、16 位或 32 位。为某些特殊应用设计的 ARM 芯片（如德国 Micronas 的 PUC3030A）没有外部扩展功能。

（14）UART 和 IrDA

几乎所有的 ARM 芯片都具有 1～2 个 UART 接口，可以用于和 PC 机通信或用 Angel 进行调试。一般的 ARM 芯片通信波特率为 115200bit/s，少数专为蓝牙技术应用设计的 ARM 芯片的 UART 通信波特率可以达到 920kbit/s，如 Linkup 公司 L7205。

（15）时钟计数器和看门狗

一般 ARM 芯片都具有 2～4 个 16 位或 32 位时钟计数器和一个看门狗计数器。

（16）电源管理功能

ARM 芯片的耗电量与工作频率成正比，一般 ARM 芯片都有低功耗模式、睡眠模式和关闭模式。

（17）DMA 控制器

有些 ARM 芯片内部集成有 DMA（Direct Memory Access）接口，可以和硬盘等外部设备高速交换数据，同时减少数据交换时对 CPU 资源的占用。

另外，还可以选择的内部功能部件有：HDLC、SDLC、CD-ROM Decoder、Ethernet MAC、VGA controller、DC-DC。可以选择的内置接口有：IIC、SPDIF、CAN、SPI、PCI、PCMCIA。

(18) 封装类型

最后需说明的是封装问题。ARM 芯片现在主要的封装有 QFP、TQFP、PQFP、LQFP、BGA、LBGA 等形式，BGA 封装具有芯片面积小的特点，可以减少 PCB 板的面积，但是需要专用的焊接设备，无法手工焊接。另外一般 BGA 封装的 ARM 芯片无法用双面板完成 PCB 布线，需要多层 PCB 板布线。

最后，根据大专、高职院校的实际情况结合当前及未来一段时间的市场人才需求，经过综合考虑，本书教学选取的是三星公司的 S3C2410x 芯片。S3C2410x 是一款基于 ARM920T 核心的微处理器芯片。本章的后面部分章节将 ARM920T 的一些特性及 S3C2410x 的详细介绍。

2.5 ARM920T 内部功能及特点

ARM920T是一种高性能的32位单片系统处理器。它提供完善的高性能CPU 子系统：

- ARM9TDMI RISC 整数 CPU
- 16-K字节指令与16-K 字节数据缓存
- 指令与数据存储器管理单元(MMUs)
- 写缓冲器
- 高级微处理器总线架构(AMBA™)总线接口
- ETM(内置追踪宏单元) 接口

ARM920T中的ARM9TDMI内核可执行32位ARM及16位Thumb指令集。ARM9TDMI处理器是哈佛结构的，有包括取指、译码、执行、存储及写入的五级流水线。

ARM920T 处理器包括两个协处理器：

- CP14 -控制软件对调试信道的访问。
- CP15 - 系统控制处理器，提供16 个额外寄存器用来配置与控制缓存、MMU、系统保护、时钟模式及其他系统选项。

ARM920T 处理器主要特性如下：

- ARM9TDMI®- 内核， ARM® v4T 架构
- 两套指令集
  - ARM 高性能32位指令集
  - Thumb高代码密度16位指令集
- 5级流水线结构
  - 取指(F)
  - 指令译码(D)
  - 执行(E)
  - 数据存储访问(M)
  - 写寄存器(W)

- 16-K字节数据缓存， 16-K 字节指令缓存
  - 虚拟地址64路相关缓存
  - 每线8 字
  - 正向及反向写操作
  - 伪随机或循环置换
  - 低功耗CAM RAM设备
- 写缓冲器
  - 16字的数据缓冲器
  - 4地址的地址缓冲器
  - 软件控制消耗
- 标准的ARMv4 存储器管理单元(MMU)
  - 区域访问许可
  - 允许以1/4页面大小对页面进行访问
  - 16个嵌入域
  - 64个输入指令TLB及64个输入数据TLB
- 8位、16位、32位的指令总线与数据总线

2.6 数据类型

2.5.1 ARM 的基本数据类型

ARM 采用的是 32 位架构，ARM 的基本数据类型有以下 3 种。

- Byte：字节，8bit。
- Halfword：半字，16bit（半字必须于 2 字节边界对齐）。
- Word：字，32bit（字必须于 4 字节边界对齐）。

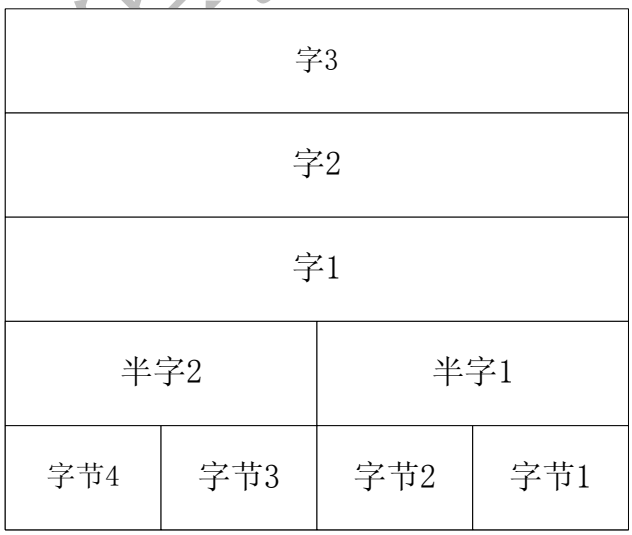

图 2-1 ARM 存储器组织结构

存储器可以看作是序号为 0～2^{32}−1 的线性字节阵列。图 2-1 所示为 ARM 存储器的组织结构。其中每一个字节都有惟一的地址。字节可以占用任一位置，图中给出了几个例子。长度为 1 个字的数据项占用一组 4 字节的位置，该位置开始于 4 的倍数的字节地址（地址最末两位为 00）。半字占有两个字节的位置，该位置开始于偶数字节地址（地址最末一位为 0）。

注意

① ARM 系统结构 v4 以上版本支持以上 3 种数据类型，v4 以前版本仅支持字节和字。
② 当将这些数据类型中的任意一种声明成 unsigned 类型时，*N* 位数据值表示范围为 $0 \sim 2^n-1$ 的非负数，通常使用二进制格式。
③ 当将这些数据类型的任意一种声明成 signed 类型时，*N* 位数据值表示范围为 $-2^{n-1} \sim 2^{n-1}-1$ 的整数，使用二进制的补码格式。
④ 所有数据类型指令的操作数都是字类型的，如“ADD r1，r0，＃0x1”中的操作数“0x1”就是以字类型数据处理的。
⑤ Load/Store 数据传输指令可以从存储器存取传输数据，这些数据可以是字节、半字、字。加载时自动进行字节或半字的零扩展或符号扩展。对应的指令分别为 LDR/BSTRB（字节操作）、LDRH/STRH（半字操作）、LDR/STR（字操作）。详见后面的指令参考。
⑥ ARM 指令编译后是 4 个字节（与字边界对齐）。Thumb 指令编译后是 2 个字节（与半字边界对齐）。

2.5.2 浮点数据类型

浮点运算使用在 ARM 硬件指令集中未定义的数据类型。

尽管如此，但 ARM 公司在协处理器指令空间定义了一系列浮点指令。通常这些指令全部可以通过未定义指令异常（此异常收集所有硬件协处理器不接受的协处理器指令）在软件中实现，但是其中的一小部分也可以由浮点运算协处理器 FPA10 以硬件方式实现。

另外，ARM 公司还提供了用 C 语言编写的浮点库作为 ARM 浮点指令集的替代方法（Thumb 代码只能使用浮点指令集）。该库支持 IEEE 标准的单精度和双精度格式。C 编译器有一个关键字标志来选择这个历程。它产生的代码与软件仿真（通过避免中断、译码和浮点指令仿真）相比既快又紧凑。

2.5.3 存储器大/小端

从软件角度看，内存相对于一个大的字节数组，其中每个数组元素（字节）都是可寻址的。

ARM 支持大端模式（big-endian）和小端模式（little-endian）两种内存模式。

图 2-2 显示了 V4/5 版本体系结构中的内存的大端模式和小端模式。

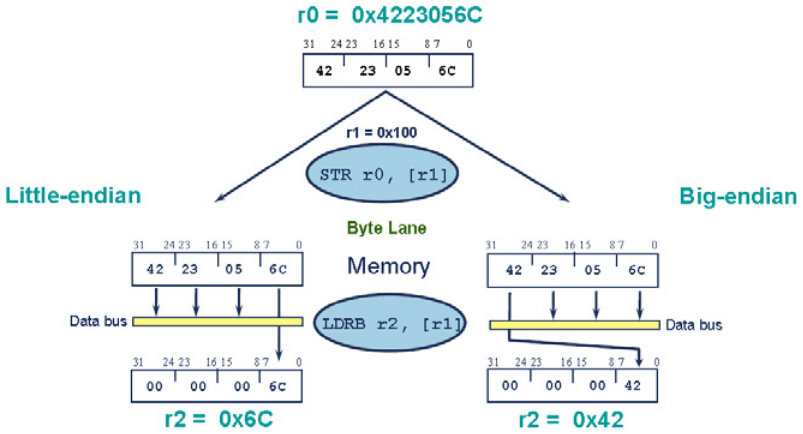

图 2-2

下面的例子显示了使用内存大/小端（big/little endian）的存取格式。

【例 3.1】

程序执行前：

```
r0=0x11223344
```

执行指令：

```
r1=0x100
STR r0，[r1]
LDRB r2，[r1]
```

执行后：

```
小端模式下：r2=0x44
大端模式下：r2=0x11
```

上面的例子向我们提示了一个潜在的编程隐患。在大端模式下，一个字的高地址放的是数据的低位，而在小端模式下，数据的低位放在内存中的低地址。要小心对待存储器中一个字内字节的顺序。

图2-2中描述的大端模式名称是BE-32，这种模式将在V7中抛弃，从V6开始引入了BE-8模式，特点如图2-3。

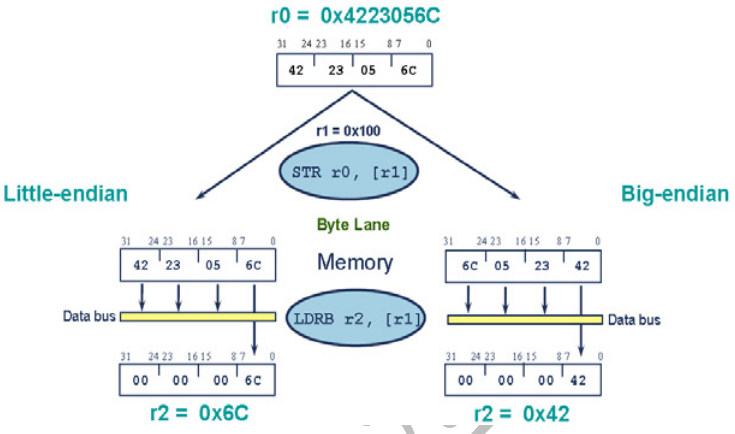

图2-3

2.7 ARM920T 内核工作模式

ARM920T是基于ARM V4T 架构，共有7种工作模式，如表2-5所示。

表2-5 ARM920T处理器的工作模式

处理器工作模式	简 写	描 述
用户模式（User）	usr	正常程序执行模式，大部分任务执行在这种模式下
快速中断模式（FIQ）	fiq	当一个高优先级（fast）中断产生时将会进入这种模式，一般用于高速数据传输和通道处理
外部中断模式（IRQ）	irq	当一个低优先级（normal）中断产生时将会进入这种模式，一般用于通常的中断处理
特权模式（Supervisor）	svc	当复位或软中断指令执行时进入这种模式，是一种供操作系统使用的保护模式
数据访问中止模式（Abort）	abt	当存取异常时将会进入这种模式，用于虚拟存储或存储保护
未定义指令中止模式（Undef）	und	当执行未定义指令时进入这种模式，有时用于通过软件仿真协处理器硬件的工作方式
系统模式（System）	sys	使用和 User 模式相同寄存器集的模式，用于运行特权级操作系统任务

除用户模式外的其他 6 种处理器模式称为特权模式（Privileged Modes）。在特权模式下，程序可以访问所有的系统资源，也可以任意地进行处理器模式切换。其中以下 5 种又称为异常模式：

（1）快速中断模式（FIQ）；
（2）外部中断模式（IRQ）；
（3）特权模式（Supervior）；
（4）数据访问中止模式（Abort）；
（5）未定义指令中止模式（Undef）。

处理器模式可以通过软件控制进行切换，也可以通过外部中断或异常处理过程进行切换。

大多数的用户程序运行在用户模式下。当处理器工作在用户模式时，应用程序不能够访问受操作系统保护的一些系统资源，应用程序也不能直接进行处理器模式切换。当需要进行处理器模式切换时，应用程序可以产生异常处理，在异常处理过程中进行处理器模式切换。这种体系结构可以使操作系统控制整个系统资源的使用。

当应用程序发生异常中断时，处理器进入相应的异常模式。在每一种异常模式中都有一组专用寄存器以供相应的异常处理程序使用，这样就可以保证在进入异常模式时用户模式下的寄存器（保存程序运行状态）不被破坏。

2.8 ARM920T 存储系统

ARM 存储系统有非常灵活的体系结构，可以适应不同的嵌入式应用系统的需要。ARM 存储器系统可以使用简单的平板式地址映射机制（就像一些简单的单片机一样，地址空间的分配方式是固定的，系统中各部分都使用物理地址），也可以使用其他技术提供功能更为强大的存储系统。例如：

（1）系统可能提供多种类型的存储器件，如 Flash、ROM、SRAM 等；
（2）Cache 技术；
（3）写缓存技术（write buffers）；
（4）虚拟内存和 I/O 地址映射技术。

大多数的系统通过下面的方法之一可实现对复杂存储系统的管理。

（1）使用 Cache，缩小处理器和存储系统速度差别，从而提高系统的整体性能。

（2）使用内存映射技术实现虚拟空间到物理空间的映射。这种映射机制对嵌入式系统非常重要。通常嵌入式系统程序存放在 ROM/Flash 中，这样系统断电后程序能够得到保存。但是，通常 ROM/Flash 与 SDRAM 相比，速度慢很多，而且基于 ARM 的嵌入式系统中通常把异常中断向量表放在 RAM 中。利用内存映射机制可以满足这种需要。在系统加电时，将 ROM/Flash 映射为地址 0，这样可以进行一些初始化处理；当这些初始化处理完成后将 SDRAM 映射为地址 0，并把系统程序加载到 SDRAM 中运行，这样很好地满足嵌入式系统的需要。

（3）引入存储保护机制，增强系统的安全性。

（4）引入一些机制保证将 I/O 操作映射成内存操作后，各种 I/O 操作能够得到正确的结果。在简单存储系统中，不存在这样问题。而当系统引入了 Cache 和 write buffer 后，就需要一些特别的措施。

在 ARM 系统中，要实现对存储系统的管理通常是使用协处理器 CP15，它通常也被称为系统控制协处理器（System Control Coprocessor）。

ARM 的存储器系统是由多级构成的，可以分为：内核级、芯片级、板卡级、外设级。图 2-4 所示为存储器的层次结构。

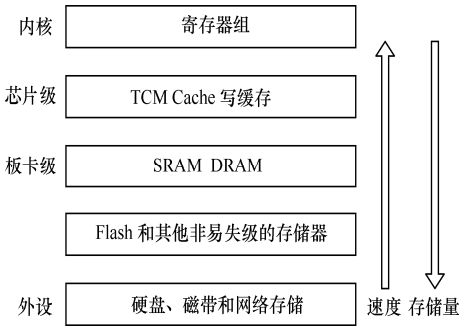

图 2-4　存储器的层次结构

每级都有特定的存储介质，下面对比各级系统中特定存储介质的存储性能。

（1）内核级的寄存器。处理器寄存器组可看作是存储器层次的顶层。这些寄存器被集成在处理器内核中，在系统中提供最快的存储器访问。典型的 ARM 处理器有多个 32 位寄存器，其访问时间为 ns 量级。

（2）芯片级的紧耦合存储器 TCM。为弥补 Cache 访问的不确定性增加的存储器。TCM 是一种快速 SDRAM，它紧挨内核，并且保证取指和数据操作的时钟周期数，这一点对一些要求确定行为的实时算法是很重要的。TCM 位于存储器地址映射中，可作为快速存储器来访问。

（3）芯片级的片上 Cache 存储器的容量在 8KB～32KB 之间，访问时间大约为 10ns。高性能的 ARM 结构中，可能存在第二级片外 Cache，容量为几百 KB，访问时间为几十 ns。

（4）板卡级的 DRAM。主存储器可能是几 MB 到几十 MB 的动态存储器，访问时间大约为 100ns。

（5）外设级的后援存储器，通常是硬盘，可能从几百 MB 到几个 GB，访问时间为几十 ms。

2.8.1 协处理器（CP15）

ARM 处理器支持 16 个协处理器。在程序执行过程中，每个协处理器忽略属于 ARM 处理器和其他协处理器的指令。当一个协处理器硬件不能执行属于它的协处理器指令时，将产生一个未定义指令异常中断，在该异常中断处理程序中，可以通过软件模拟该硬件操作。例如，如果系统不包含向量浮点运算器，则可以选择浮点运算软件模拟包来支持向量浮点运算。CP15，即通常所说的系统控制协处理器（System Control Coprocesssor），它负责完成大部分的存储系统管理。除了 CP15 外，在具体的各种存储管理机制中可能还会用到其他的一些技术，如在 MMU 中除了 CP15 外，还使用了页表技术等。

在一些没有标准存储管理的系统中，CP15 是不存在的。在这种情况下，针对 CP15 的操作指令将被视为未定义指令，指令的执行结果不可预知。

CP15 包含 16 个 32 位寄存器，其编号为 0～15。实际上对于某些编号的寄存器可能对应多个物理寄存器，在指令中指定特定的标志位来区分这些物理寄存器。这种机制有些类似于 ARM 中的寄存器，当处于不同的处理器模式时，某些相同编号的寄存器对应于不同的物理寄存器。

CP15 中的寄存器可能是只读的，也可能是只写的，还有一些是可读可写的。在对协处理器寄存器进行操作时，需要注意以下几个问题：

（1）寄存器的访问类型（只读/只写/可读可写）；

（2）不同的访问引发的不同功能；

（3）相同编号的寄存器是否对应不同的物理寄存器；

（4）寄存器的具体作用。

2.8.2 存储管理单元（MMU）

在创建多任务嵌入式系统时，最好有一个简单的方式来编写、装载及运行各自独立的任务。目前大多数的嵌入式系统不再使用自己定制的控制系统，而使用操作系统来简化这个过程。较高级的操作系统采用基于硬件的存储管理单元（MMU）来实现上述操作。

MMU 提供的一个关键服务是使各个任务作为各自独立的程序在其自己的私有存储空间中运行。在带 MMU 的操作系统控制下，运行的任务无须知道其他与之无关的任务的存储需求情况，这就简化了各个任务的设计。

MMU 提供了一些资源以允许使用虚拟存储器（将系统物理存储器重新编址，可将其看成一个独立于系统物理存储器的存储空间）。MMU 作为转换器，将程序和数据的虚拟地址（编译时的连接地址）转换成实际的物理地址，即在物理主存中的地址。这个转换过程允许运行的多个程序使用相同的虚拟地址，而各自存储在物理存储器的不同位置。

这样存储器就有两种类型的地址：虚拟地址和物理地址。虚拟地址由编译器和连接器在定位程序时分配；物理地址用来访问实际的主存硬件模块（物理上程序存在的区域）。

2.8.3 高速缓冲存储器（Cache）

Cache 是一个容量小但存取速度非常快的存储器，它保存最近用到的存储器数据副本。对于程序员来说，Cache 是透明的。它自动决定保存哪些数据、覆盖哪些数据。现在 Cache 通常与处理器在同一芯片上实现。Cache 能够发挥作用是因为程序具有局部性特性。所谓局部性就是指在任何特定的时间，处理器趋于对相同区域的数据（如堆栈）多次执行相同的指令（如循环）。

Cache 经常与写缓存器（write buffer）一起使用。写缓存器是一个非常小的先进先出（FIFO）存储器，位于处理器核与主存之间。使用写缓存的目的是，将处理器核和 Cache 从较慢的主存写操作中解脱出来。当 CPU 向主存储器做写入操作时，它先将数据写入到写缓存区中，由于写缓存器的速度很高，这种写入操作的速度也将很高。写缓存区在 CPU 空闲时，以较低的速度将数据写入到主存储器中相应的位置。

通过引入 Cache 和写缓存区，存储系统的性能得到了很大的提高，但同时也带来了一些问题。例如，由于数据将存在于系统中的不同的物理位置，可能造成数据的不一致性；由于写缓存区的优化作用，可能有些写操作的执行顺序不是用户期望的顺序，从而造成操作错误。

2.9 流 水 线

2.9.1 流水线的概念与原理

处理器按照一系列步骤来执行每一条指令，典型的步骤如下：

（1）从存储器读取指令（fetch）；

（2）译码以鉴别它是属于哪一条指令（decode）；

（3）从指令中提取指令的操作数（这些操作数往往存在于寄存器中）（reg）；

（4）将操作数进行组合以得到结果或存储器地址（ALU）；

（5）如果需要，则访问存储器以存储数据（mem）；

（6）将结果写回到寄存器堆（res）。

并不是所有的指令都需要上述每一个步骤，但是，多数指令需要其中的多个步骤。这些步骤往往使用不同的硬件功能，如 ALU 可能只在第 4 步中用到。因此，如果一条指令不是在前一条指令结束之前就开始，那么在每一步骤内处理器只有少部分的硬件在使用。

有一种方法可以明显改善硬件资源的使用率和处理器的吞吐量，这就是当前一条指令结束之前就开始执行下一条指令，即通常所说的流水线（Pipeline）技术。流水线是 RISC 处理器执行指令时采用的机制。使用流水线，可在取下一条指令的同时译码和执行其他指令，从而加快执行的速度。可以把流水线看作是汽车生产线，每个阶段只完成专门的处理器任务。

采用上述操作顺序，处理器可以这样来组织：当一条指令刚刚执行完步骤（1）并转向步骤（2）时，下一条指令就开始执行步骤（1）。从原理上说，这样的流水线应该比没有重叠的指令执行快 6 倍，但由于硬件结构本身的一些限制，实际情况会比理想状态差一些。

2.9.2 流水线的分类

1．3 级流水线 ARM 组织

到 ARM7 为止的 ARM 处理器使用简单的 3 级流水线，它包括下列流水线级。

（1）取指令（fetch）：从寄存器装载一条指令。

（2）译码（decode）：识别被执行的指令，并为下一个周期准备数据通路的控制信号。在这一级，指令占有译码逻辑，不占用数据通路。

（3）执行（excute）：处理指令并将结果写回寄存器。

图 2-5 所示为 3 级流水线指令的执行过程。

图 2-5　3 级流水线

当处理器执行简单的数据处理指令时，流水线使得平均每个时钟周期能完成 1 条指令。但 1 条指令需要 3 个时钟周期来完成，因此，有 3 个时钟周期的延时（latency），但吞吐率（throughput）是每个周期 1 条指令。

2．5 级流水线 ARM 组织

所有的处理器都要满足对高性能的要求，直到 ARM7 为止，在 ARM 核中使用的 3 级流水线的性价比是很高的。但是，为了得到更高的性能，需要重新考虑处理器的组织结构。有两种方法来提高性能。

第一，提高时钟频率。时钟频率的提高，必然引起指令执行周期的缩短，所以要求简化流水线每一级的逻辑，流水线的级数就要增加。

第二，减少每条指令的平均指令周期数 CPI。这就要求重新考虑 3 级流水线 ARM 中多于 1 个流水线周期的实现方法，以便使其占有较少的周期，或者减少因指令相关造成的流水线停顿，也可以将两者结合起来。

3 级流水线 ARM 核在每一个时钟周期都访问存储器，或者取指令，或者传输数据。只是抓紧存储器不用的几个周期来改善系统系统性能，效果并不明显。为了改善 CPI，存储器系统必须在每个时钟周期中给出多于一个的数据。方法是在每个时钟周期从单个存储器中给出多于 32 位数据，或者为指令或数据分别设置存储器。

基于以上原因，较高性能的 ARM 核使用了 5 级流水线，而且具有分开的指令和数据存储器。把指令的执行分割为 5 部分而不是 3 部分，进而可以使用更高的时钟频率，分开的指令和数据存储器使核的 CPI 明显减少。

在 ARM9TDMI 中使用了典型的 5 级流水线，5 级流水线包括下面的流水线级。

（1）取指令（fetch）：从存储器中取出指令，并将其放入指令流水线。

（2）译码（decode）：指令被译码，从寄存器堆中读取寄存器操作数。在寄存器堆中有 3 个操作数读端口，因此，大多数 ARM 指令能在 1 个周期内读取其操作数。

（3）执行（execute）：将其中 1 个操作数移位，并在 ALU 中产生结果。如果指令是 Load 或 Store 指令，则在 ALU 中计算存储器的地址。

（4）缓冲/数据（buffer/data）：如果需要则访问数据存储器，否则 ALU 只是简单地缓冲 1 个时钟周期。

（5）回写（write-back）：将指令的结果回写到寄存器堆，包括任何从寄存器读出的数据。

图 2-6 列出了 5 级流水线指令的执行过程。

图 2-6　5 级流水线

在程序执行过程中，PC 值是基于 3 级流水线操作特性的。5 级流水线中提前 1 级来读取指令操作数，得到的值是不同的（PC＋4 而不是 PC＋8）。这产生的代码不兼容是不容许的。但 5 级流水线 ARM 完全仿真 3 级流水线的行为。在取指级增加的 PC 值被直接送到译码级的寄存器，穿过两级之间的流水线寄存器。

下一条指令的 PC + 4 等于当前指令的 PC + 8，因此，未使用额外的硬件便得到了正确的 R15。

3．6 级流水线 ARM 组织

在 ARM10 中，将流水线的级数增加到 6 级，使系统的平均处理能力达到了 1.3DMIPS/MHz。CPU 性能评估采用合成测试程序，较流行的有 Whetstone 和 Dhrystone 两种。Dhrystone 主要用于测整数计算能力，计算单位就是 DMIPS。DMIPS 可以理解为处理器单位时间内执行处理整数的指令的百万次数。而因为处理器的性能与工作频率密切相关，在不同工作频率下测算出的 DMIPS 是不同的，所以通常使用 DMIPS/MHz 作为标准，评估各个处理器的结构优劣和性能高低。图 2-7 所示为 6 级流水线上指令的执行过程。

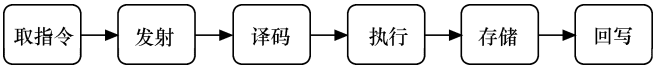

图 2-7　6 级流水线

2.9.3　影响流水线性能的因素

1．互锁

在典型的程序处理过程中，经常会遇到这样的情形，即一条指令的结果被用作下一条指令的操作数，如：

有如下指令序列：

```
LDR R0,[R0,#0]
ADD R0,R0,R1     ;在 5 级流水线上产生互锁
```

从例子中可以看出，流水线的操作产生中断，因为第 1 条指令的结果在第 2 条指令取数时还没有产生。第 2 条指令必须停止，直到结果产生为止。

2．跳转指令

跳转指令也会破坏流水线的行为，因为后续指令的取指步骤受到跳转目标计算的影响，因而必须推迟。但是，当跳转指令被译码时，在它被确认是跳转指令之前，后续的取指操作已经发生。这样一来，已经被预取进入流水线的指令不得不被丢弃。如果跳转目标的计算是在 ALU 阶段完成的，那么在得到跳转目标之前已经有两条指令按原有指令流读取。

显然，只有当所有指令都依照相似的步骤执行时，流水线的效率达到最高。如果处理器的指令非常复杂，每一条指令的行为都与下一条指令不同，那么就很难用流水线实现。

2.10　寄存器组织

ARM 处理器有如下 37 个 32 位长的寄存器：

（1）30 个通用寄存器；

（2）6 个状态寄存器：1 个 CPSR（Current Program Status Register，当前程序状态寄存器），5 个 SPSR（Saved Program Status Register，备份程序状态寄存器）；

（3）1 个 PC（Program Counter，程序计数器）。

ARM 处理器共有 7 种不同的处理器模式，在每一种处理器模式中有一组相应的寄存器组。表 2-6 列出了 ARM 处理器的寄存器组织概要。

表 2-6　　寄存器组织概要

User	FIQ	IRQ	SVC	Undef	Abort
R0	User mode R0～R7,R15 和 CPSR	User mode R0～R12,R15 和 CPSR	User mode R0～R12,R15 和 CPSR	User mode R0～R12,R15 和 CPSR	User mode R0～R12,R15 和 CPSR
R1					
R2					
R3					
R4					
R5					
R6					
R7					
R8	R8				
R9	R9				
R10	R10				
R11	R11				
R12	R12				
R13(SP)	R13(SP)	R13	R13	R13	R13
R14(LR)	R14(LR)	R14	R14	R14	R14
R15(PC)					
CPSR					
	SPSR	SPSR	SPSR	SPSR	SPSR

当前处理器的模式决定着哪组寄存器可操作，任何模式都可以存取下列寄存器：

（1）相应的 R0～R12；

（2）相应的 R13（the Stack Pointer，SP，栈指向）和 R14（the Link Register，LR，链路寄存器）；

（3）相应的 R15（PC）；

（4）相应的 CPSR。

特权模式（除 System 模式）还可以存取相应的 SPSR。

2.10.1　通用寄存器

通用寄存器根据其分组与否可分为以下 2 类。

（1）未分组寄存器（the Unbanked Register），包括 R0～R7。

（2）分组寄存器（the Banked Register），包括 R8～R14。

1．未分组寄存器

未分组寄存器包括 R0～R7。顾名思义，在所有处理器模式下对于每一个未分组寄存器来说，指的都是同一个物理寄存器。未分组寄存器没有被系统用于特殊的用途，任何可采用通用寄存器的应用场合都可以使用未分组寄存器。但由于其通用性，在异常中断所引起的处理器模式切换时，其使用的是相同的物理寄存器，所以也就很容易使寄存器中的数据被破坏。

2．分组寄存器

R8～R14 是分组寄存器，它们每一个访问的物理寄存器取决于当前的处理器模式。

对于分组寄存器 R8～R12 来说，每个寄存器对应两个不同的物理寄存器。一组用于除 FIQ 模式外的所有处理器模式，而另一组则专门用于 FIQ 模式。这样的结构设计有利于加快 FIQ 的处理速度。不同模式下寄存器的使用，要使用寄存器名后缀加以区分。例如，当使用 FIQ 模式下的寄存器时，寄存器 R8 和寄存器 R9 分别记为 R8_fiq、R9_fiq；当使用用户模式下的寄存器时，寄存器 R8 和 R9 分别记为 R8_usr、R9_usr 等。在 ARM 体系结构中，R8～R12 没有任何指定的其他的用途，所以当 FIQ 中断到达时，不用保存这些通用寄存器，也就是说，FIQ 处理程序可以不必执行保存和恢复中断现场的指令，从而可以使中断处理过程非常迅速。所以 FIQ 模式常被用来处理一些时间紧急的任务，如 DMA 处理。

对于分组寄存器 R13 和 R14 来说，每个寄存器对应 6 个不同的物理寄存器。其中的一个是用户模式和系统模式公用的，而另外 5 个分别用于 5 种异常模式。访问时需要指定它们的模式。名字形式如下：

（1）R13_<mode>

（2）R14_<mode>

其中，<mode>可以是以下几种模式之一：usr、svc、abt、und、irp 及 fiq。

R13 寄存器在 ARM 处理器中常用作堆栈指针，称为 SP。当然，这只是一种习惯用法，并没有任何指令强制性的使用 R13 作为堆栈指针，用户完全可以使用其他寄存器作为堆栈指针。而在 Thumb 指令集中，有一些指令强制性的将 R13 作为堆栈指针，如堆栈操作指令。

每一种异常模式拥有自己的 R13。异常处理程序负责初始化自己的 R13，使其指向该异常模式专用的栈地址。在异常处理程序入口处，将用到的其他寄存器的值保存在堆栈中，返回时，重新将这些值加载到寄存器。通过这种保护程序现场的方法，异常不会破坏被其中断的程序现场。

寄存器 R14 又被称为连接寄存器（Link Register，LR），在 ARM 体系结构中具有下面两种特殊的作用。

（1）每一种处理器模式用自己的 R14 存放当前子程序的返回地址。当通过 BL 或 BLX 指令调用子程序时，R14 被设置成该子程序的返回地址。在子程序返回时，把 R14 的值复制到程序计数器（PC）。典型的做法是使用下列两种方法之一。

① 执行下面任何一条指令。

```
MOV  PC, LR
BX  LR
```

② 在子程序入口处使用下面的指令将 PC 保存到堆栈中。

```
STMFD  SP!, {<register>,LR}
```

在子程序返回时，使用如下相应的配套指令返回。

```
LDMFD  SP!, {<register>,PC}
```

（2）当异常中断发生时，该异常模式特定的物理寄存器 R14 被设置成该异常模式的返回地址，对于有些模式 R14 的值可能与返回地址有一个常数的偏移量（如数据异常使用 SUB PC，LR，#8 返回）。具体的返回方式与上面的子程序返回方式基本相同，但使用的指令稍微有些不同，以保证当异常出现时正在执行的程序的状态被完整保存。

R14 也可以被用作通用寄存器使用。

2.11 程序状态寄存器

当前程序状态寄存器（Current Program Status Register，CPSR）可以在任何处理器模式下被访问，它包含下列内容：

（1）ALU（Arithmetic Logic Unit，算术逻辑单元）状态标志的备份；

（2）当前的处理器模式；

（3）中断使能标志；

（4）设置处理器的状态（只在 4T 架构）。

每一种处理器模式下都有一个专用的物理寄存器作备份程序状态寄存器（Saved Program Status Register，SPSR）。当特定的异常中断发生时，这个物理寄存器负责存放当前程序状态寄存器的内容。当异常处理程

序返回时，再将其内容恢复到当前程序状态寄存器。

CPSR 寄存器（和保存它的 SPSR 寄存器）中的位分配如图 2-8 所示。

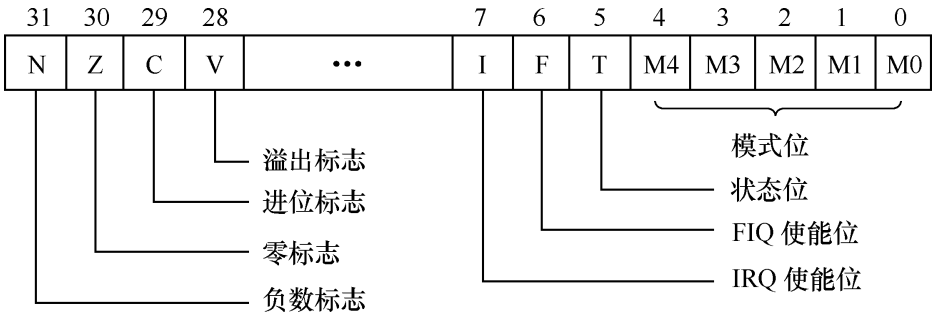

图 2-8　程序状态寄存器格式

1．标志位

N（Negative）、Z（Zero）、C（Carry）和 V（oVerflow）通称为条件标志位。这些条件标志位会根据程序中的算术指令或逻辑指令的执行结果进行修改，而且这些条件标志位可由大多数指令检测以决定指令是否执行。

在 ARM 4T 架构中，所有的 ARM 指令都可以条件执行，而 Thumb 指令却不能。

各条件标志位的具体含义如下。

（1）N

本位设置成当前指令运行结果的 bit[31]的值。当两个由补码表示的有符号整数运算时，N = 1 表示运算的结果为负数；N = 0 表示结果为正数或零。

（2）Z

Z = 1 表示运算的结果为零，Z = 0 表示运算的结果不为零。

（3）C

下面分 4 种情况讨论 C 的设置方法。

① 在加法指令中（包括比较指令 CMN），当结果产生了进位，则 C = 1，表示无符号数运算发生上溢出；其他情况下 C = 0。

② 在减法指令中（包括比较指令 CMP），当运算中发生错位（即无符号数运算发生下溢出），则 C = 0；其他情况下 C = 1。

③ 对于在操作数中包含移位操作的运算指令（非加/减法指令），C 被设置成被移位寄存器最后移出去的位。

④ 对于其他非加/减法运算指令，C 的值通常不受影响。

（4）V

下面分两种情况讨论 V 的设置方法。

① 对于加/减运算指令，当操作数和运算结果都是以二进制的补码表示的带符号的数时，且运算结果超出了有符号运算的范围是溢出。V = 1 表示符号位溢出。

② 对于非加/减法指令，通常不改变标志位 V 的值（具体可参照 ARM 指令手册）。

尽管以上 C 和 V 的定义看起来颇为复杂，但使用时在大多数情况下用一个简单的条件测试指令即可，不需要程序员计算出条件码的精确值即可得到需要的结果。

2．Q 标志位

在带 DSP 指令扩展的 ARM v5 及更高版本中，bit[27]被指定用于指示增强的 DAP 指令是否发生了溢

出，因此也就被称为 Q 标志位。同样，在 SPSR 中 bit[27]也被称为 Q 标志位，用于在异常中断发生时保存和恢复 CPSR 中的 Q 标志位。

在 ARM v5 以前的版本及 ARM v5 的非 E 系列处理器中，Q 标志位没有被定义，属于待扩展的位。

3．控制位

CPSR 的低 8 位（I、F、T 及 M[4：0]）统称为控制位。当异常发生时，这些位的值将发生相应的变化。另外，如果在特权模式下，也可以通过软件编程来修改这些位的值。

（1）中断禁止位

I = 1，IRQ 被禁止。

F = 1，FIQ 被禁止。

（2）状态控制位

T 位是处理器的状态控制位。

T = 0，处理器处于 ARM 状态（即正在执行 32 位的 ARM 指令）。

T = 1，处理器处于 Thumb 状态（即正在执行 16 位的 Thumb 指令）。

当然，T 位只有在 T 系列的 ARM 处理器上才有效，在非 T 系列的 ARM 版本中，T 位将始终为 0。

（3）模式控制位

M[4：0]作为位模式控制位，这些位的组合确定了处理器处于哪种状态。表 2-6 列出了其具体含义。只有表中列出的组合是有效的，其他组合无效。

表 2-6　　状态控制位 M[4：0]

M[4：0]	处理器模式	可以访问的寄存器
0b10000	User	PC，R14～R0，CPSR
0b10001	FIQ	PC，R14_fiq～R8_fiq，R7～R0，CPSR，SPSR_fiq
0b10010	IRQ	PC，R14_irq～R13_irq，R12～R0，CPSR，SPSR_irq
0b10011	Supervisor	PC，R14_svc～R13_svc，R12～R0，CPSR，SPSR_svc
0b10111	Abort	PC，R14_abt～R13_abt，R12～R0，CPSR，SPSR_abt
0b11011	Undefined	PC，R14_und～R13_und，R12～R0，CPSR，SPSR_und
0b11111	System	PC，R14～R0，CPSR（ARM v4 及更高版本）

2.12 三星 S3C2410X 处理器介绍

S3C2410 是著名的半导体公司 SAMSUNG 推出的一款 32 位 RISC 处理器，它为手持设备和一般类型的应用提供了低价格、低功耗、高性能微控制器的解决方案。S3C2410 的内核基于 ARM920T，带有 MMU（Memory Management Unit）功能，采用 0.18μm 工艺，其主频可达 203MHz，适合于对成本和功耗敏感的需求。同时它还采用了 AMBA（Advanced Microcontr-oller Bus Architecture）的新型总线结构，实现了 MMU、AMBA BUS、Harvard 的高速缓冲体系结构，同时支持 Thumb16 位压缩指令集，从而能以较小的存储空间需求，获得 32 位的系统性能。

其片上功能如下：

（1）内核工作电压为 1.8/2.0V，存储器供电电压为 3.3V，外部 I/O 设备的供电电压为 3.3V；

（2）16KB 的指令 Cache 和 16KB 的数据 Cache；

（3）LCD 控制器，最大可支持 4K 色 STN 和 256 色 TFT；

（4）4 通道的 DMA 请求；

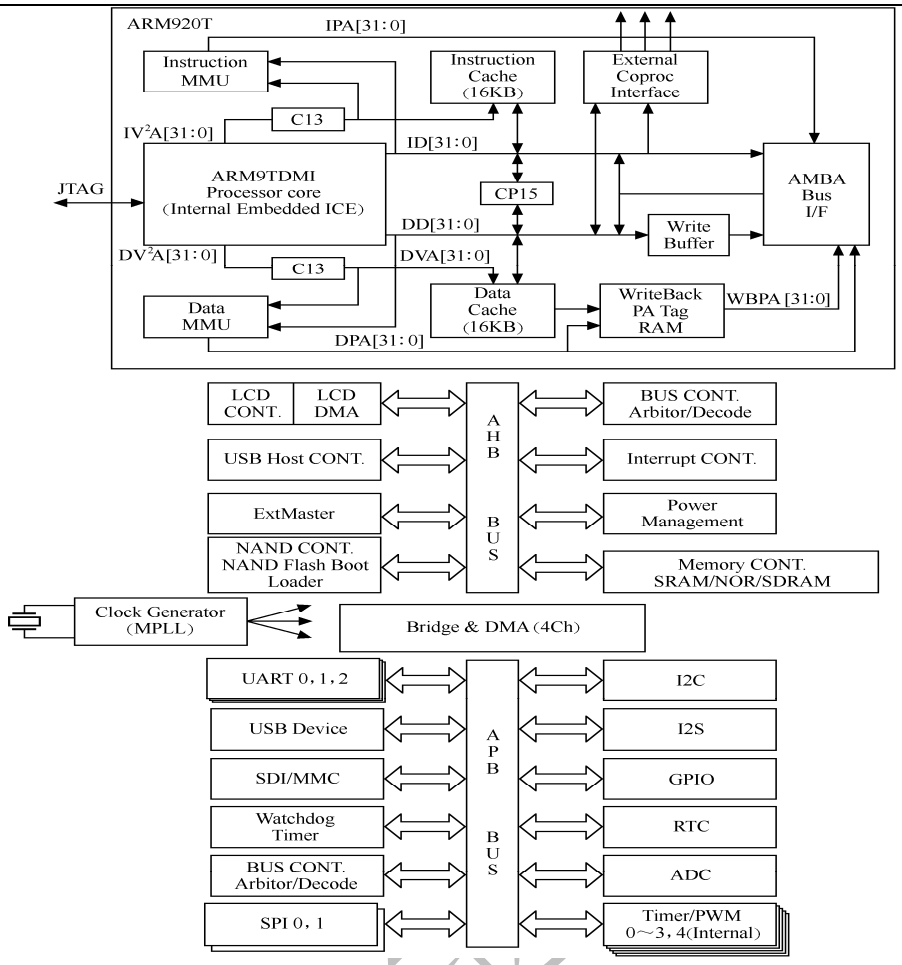

图 2-9　S3C2410 结构框图

（5）3 通道的 UART（IrDA1.0、16 字节 TxFIFO、16 字节 RxFIFO），2 通道的 SPI 接口；
（6）2 通道的 USB（Host/Slave）；
（7）4 路 PWM 和 1 个内部时钟控制器；
（8）117 个通用 I/O，24 路外部中断；
（9）272Pin FBGA 封装；
（10）16 位的看门狗定时器；
（11）1 通道的 IIC/IIS 控制器；
（12）带有 PLL 片上时钟发生器。

S3C2410 处理器支持大/小端模式存储字数据，其寻址空间可达 1GB，对于外部 I/O 设备的数据宽度，可以是 8/16/32 位，所有的存储器 Bank（共有 8 个）都具有可编程的操作周期，而且支持各种 ROM 引导方式（NOR/Nand Flash、EEPROM 等），其结构框图如图 2-8 所示。

2.13 小结

本章介绍了 ARM 处理器的一些关键技术，如：ARM 核的工作模式、存储系统、流水线、寄存器组织等。并且列举了一款基于 ARM920T 核的处理器芯片 S3C2410。通过本章的学习，学员可以对 ARM 核的一些关键技术有所认识。

2.14 练习题

1、 简述 ARM 可以工作的几种模式

2、 ARM 核有多少个寄存器？
3、 什么寄存器用于存储 PC 和 LR 寄存器？
4、 R13 通常用来存储什么？
5、 哪种模式使用的寄存器最少？
CPSR 的哪一位反映了处理器的状态？

联系方式

集团官网：www.hqyj.com 嵌入式学院：www.embedu.org 移动互联网学院：www.3g-edu.org

企业学院：www.farsight.com.cn 物联网学院：www.topsight.cn 研发中心：dev.hqyj.com

集团总部地址：北京市海淀区西三旗悦秀路北京明园大学校内 华清远见教育集团

北京地址：北京市海淀区西三旗悦秀路北京明园大学校区，电话：010-82600386/5

上海地址：上海市徐汇区漕溪路 250 号银海大厦 11 层 B 区，电话：021-54485127

深圳地址：深圳市龙华新区人民北路美丽 AAA 大厦 15 层，电话：0755-25590506

成都地址：成都市武侯区科华北路 99 号科华大厦 6 层，电话：028-85405115

南京地址：南京市白下区汉中路 185 号鸿运大厦 10 层，电话：025-86551900

武汉地址：武汉市工程大学卓刀泉校区科技孵化器大楼 8 层，电话：027-87804688

西安地址：西安市高新区高新一路 12 号创业大厦 D3 楼 5 层，电话：029-68785218

广州地址：广州市天河区中山大道 268 号天河广场 3 层，电话：020-28916067