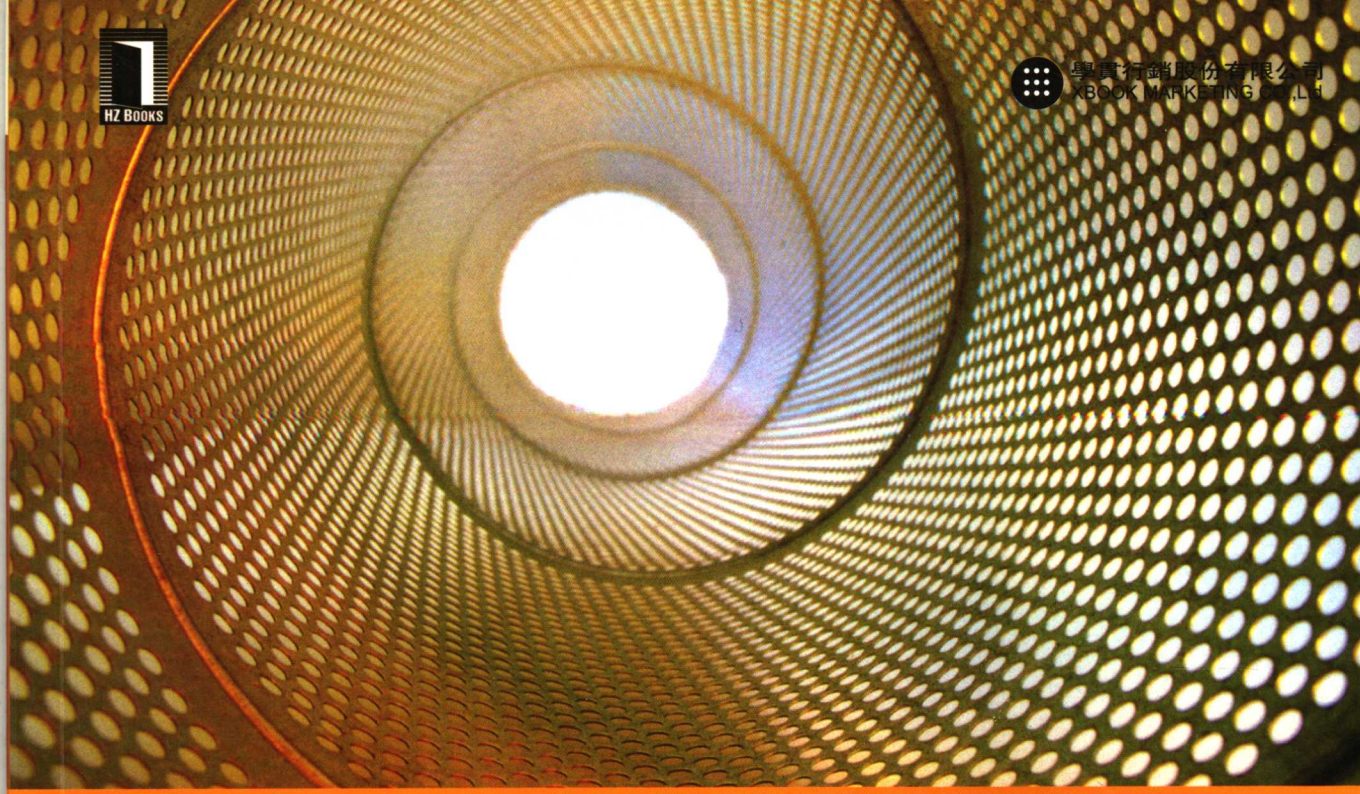

ARM原理与嵌入式系统实战

范圣一 ◎ 著

机械工业出版社
China Machine Press

ARM原理与嵌入式系统实战

范圣一 ◎ 著

机械工业出版社
China Machine Press

本书讲解嵌入式系统基本原理与ARM SoC的实际应用，主要内容包括与ARM相关的知识，以ARM CPU为基础的基本概念，用实例介绍ARM设计方法。本书涵盖ARM SoC架构与开发工具的介绍，并说明了相关的接口电路与内存等，使读者能快速掌握ARM的设计技术。

本书适合从事嵌入式系统开发的技术人员参考。

本书版权登记号：图字：01-2006-6888

图书在版编目（CIP）数据

ARM原理与嵌入式系统实战/范圣一著. －北京：机械工业出版社，2007.7

ISBN 978-7-111-21501-1

Ⅰ.A… Ⅱ.范… Ⅲ.①微处理器，ARM ②微型计算机－系统设计 Ⅳ.TP332 TP360.21

中国版本图书馆CIP数据核字（2007）第069381号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：李南丰

北京牛山世兴印刷厂印刷 · 新华书店北京发行所发行

2007年7月第1版第1次印刷

186mm×240mm · 22.25印张

定价：42.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书电话（010）68326294

推 荐 序

随着技术的进步，32位的ARM处理器已经取代了大多数8051的市场。在许多嵌入式系统和消费性电子产品中都可以看到ARM的踪迹。但ARM有许多先进的架构，因此在软件、固件设计乃至于硬件设计上，都不同于 8051。加上SoC的设计隐藏其中，更加提高了技术层次。这些优点都可以从过去工作中所接触到的众多研发工程师身上得到验证。就大多数工程师的需求而言，一本涵盖ARM软件开发与硬件架构，甚至外围接口的整合设计的参考书籍，是完成产品设计不可缺少的工具。但是市面上可获得的信息相当有限。

与作者认识多年，得知他深厚的微处理器实战经验，丰富的产品设计经历。从硬件实验板的设计到软件链接库的开发，他均能独立完成。再加上他在授课当中，跟实际从事研发的工程师有许多的互动，使得在规划本书章节与内容时，多了一份符合研发人员需要的贴心考虑，让本书不同于其他介绍ARM或微处理器的相关书籍。本书由基础到进阶实例的章节安排，涵盖架构与开发工具的介绍，并说明了相关的外围电路与内存等，让读者很容易循序进入ARM的设计领域，并且可以充分掌握设计者的系统观。这是一本适合研发人员的工具书，更是一本适合微处理器应用的教学参考书。

传识信息股份有限公司
课程开发经理
林昭宏
2006年2月

序　言

笔者自1995年开始接触ARM SoC，在使用过程中深感学习ARM不仅是学习一个16/32位的CPU。要了解ARM必须从整体的嵌入式系统开始，有非常多的相关知识是在学习ARM之前必须具备的，我们不仅要具备微处理器的概念，还要了解Flash、SDRAM的原理和使用方式，又要学习嵌入式编译器。如果只是写一写ARM的应用程序实在无法描述ARM SoC的深奥，但是市面上确实很难找到一本从基本嵌入式系统原理到结合ARM SoC的书，这也是笔者在使用ARM十年后想要写这本书的原因。

本书的结构分为3大部分，分别是基础篇、原理篇及实例篇。基础篇讲述在学习ARM前必须具备的相关知识，原理篇讲述以ARM CPU为基础的基本概念，实例篇是将前两篇所讲述的概念以S3C44B0X（韩国三星公司所出的以ARM7TDMI为内核）为例设计出一套系统。

笔者学习的过程中深感我们是使用ARM SoC而不是设计ARM SoC，所以有关ARM内核概念的阐述，并没有非常多的篇幅，而多着墨于探讨如何应用ARM SoC，这不只是一本探讨ARM CPU内核的书，更是一本探讨ARM的原理和应用的书。

要了解ARM SoC，势必要了解微处理器控制，微处理器控制是嵌入式系统的基础，为了让读者更容易了解ARM SoC，笔者在各个不同的章节穿插了微处理器控制的概念，这不只是一本探讨16/32位微控制器的书，更是一本探讨嵌入式系统的书。

本书着重于如何以ARM为基础来设计一套嵌入式系统，既然是设计嵌入式系统，当然就不只是软／固件设计，还包含了硬件的设计，这不只是一本探讨硬件或软件设计的书，更是一本探讨如何设计一个嵌入式系统的书。

本书许多地方的程序均用汇编语言编写，但限于篇幅的关系并没有讲述ARM的汇编语言，请读者自行参考其他书。

本书在硬件上会从电源、Reset开始到设计一个以ARM为主的最基本电路，软件上会从Reset exception开始，到建立一个以ARM为主的引导程序（Boot Loader），为使读者容易学习，所使用的硬件电路都会先以方块图表示，最后以完整电路呈现，软件亦用最简单的程序解释。本书的主要内容包括：

- ARM的内核及SoC的结构。
- 如何编写ARM的开机程序。
- 嵌入式系统的编译器及汇编语言。
- 各种内存的原理。
- 如何设计一套嵌入式系统（软件及硬件）。
- 以S3C44B0X为例设计一套嵌入式系统。

目　录

实 例 篇

第1章 ARM处理器概述

1.1 ARM公司简介

ARM（Advanced RISC Machine）是全球微处理器行业中一家知名的公司，该公司于1990年11月在英国剑桥大学成立。ARM公司是由苹果电脑、Acorn Computer Group和VLSI Technology联合建立。苹果电脑和VLSI Technology出资，Acorn提供技术和12名工程师。1985年，负责技术的Acorn小组开发了世界上第一个商用单片RISC微处理器。由于苹果电脑当时希望将RISC技术应用于苹果计算机系统。由此，ARM希望能够实现精简型RISC，随后成功地研制了首个低成本RISC架构。

20世纪80年代末90年代初，半导体产业出现分工，积体电路、晶园等半导体代工厂逐渐崛起，IC设计公司（自己设计芯片，但是生产过程则包给代工厂生产）也纷纷涌现。

微处理器内核是ARM技术的核心，目前市场上能见到的有ARM7、ARM9、ARM9E、ARM10E、SecuRCore、ARM11，还有Intel的 Xscale微体系架构及StrongARM等系列。ARM专利技术收入主要来自两个方面：一个是专利授权费用，客户如果采用ARM专利时，需一次性付给ARM费用；另一部分是按照一定比例收取客户产品的专利使用费，即客户每卖出一个芯片，就收取同等比例的费用。

ARM是一家设计32位嵌入式RISC芯片内核的公司。该公司的产品“ARM嵌入式内核”已被全球各大芯片厂商采用，基于ARM的开发技术也席卷了全球嵌入式市场，已成为嵌入式系统的主流技术之一。

全球厂商都可以通过获得授权进行ARM 32位CPU核芯片的生产。其中低端产品可以采用8051等8位的单片机，而中高端产品则采用ARM的核心技术。

1.2 ARM微处理器的种类

ARM处理器采用RISC的架构技术，它具备小体积、低功耗、低成本、高性能等特色，支持Thumb（16位）和ARM（32位）双指令集，能很好地兼容8位/16位器件。

ARM按照内核可以分为以下几种：

1. ARM7微处理器系列

ARM7为低功率的32位RISC处理器，包括32位地址线和数据线，最适合用于对价位和功率要求较高的应用，其特点如下：

- 具有嵌入式ICE逻辑，调试开发方便。
- 低功耗，适合便携式产品。

• 具16位Thumb 指令集。

• 主频可达130MIPS。

常用的ARM7的内核有：ARM7TDMI、ARM7TDMI-S、ARM720T、ARM7EJ-s等，其中各个英文字母所代表的意义如下:

T：支持16位Thumb指令集。

D：支持在线Debug。

M：内嵌乘法器Multiplier。

I：内嵌入式ICE，支持在线断点和调试。

E：DSP指令，表示支持DSP的特定指令，主要是16位的Thumb指令。

S：可以综合，提供VHDL或者Verilog语言设计，可以实现自己特定的硬件。

J：支持新的Java功能。

本书后半部分介绍的Samsung S3C44B0X，就是采用ARM7TDMI的内核。

2. ARM9微处理机系列

除了ARM7的功能外，ARM9还具有以下特点：

• 支持32位的AMBA总线接口。

• 支持MMU（Memory Management Unit），可提供Window's CE及Linux Kernel 2.6以上的操作系统。

• 支持数据缓存（Data Cache）及指令缓存（Instruction Cache），提高了CPU的处理效率。

常用的ARM9的内核有ARM920T、ARM922T和ARM940T。

3. ARM9E微处理器系列

它是一个综合性的处理器，除了具有ARM9的功能外，还增加了DSP和Java应用系统的处理。其主要特色如下：

• 支持DSP指令集，适合高速运算。

• 支持VFP9浮点运算。

• CPU主频高达300MIPS。

常用的ARM9E的内核有：ARM926EJ-S、ARM946E-S和ARM966E-S。

4. ARM10E微处理器系列

它采用了新的架构，使其效率比ARM9提高将近50%，并且提升了能源管理的功能，从而降低了功耗。除了ARM9的功能外，ARM10E还具有以下功能：

• 支持VFP10浮点运算。

• 主频高达400MIPS。

• 内部并行读／写。

常用的ARM10E的内核有：ARM1020E、ARM1022E及ARM1026EJ-S。

ARM公司各类产品的功能见表1-1，表1-2，表1-3，表1-4，表1-5，表1-6，表1-7。

表1-1　ARM7微处理器的功能

微处理器	Cache（指令/数据）	TCM	存储管理	AHB	Thumb	DSP	Jazelle
ARM7TDMI	×	×	×	○	○	×	×
ARM7TDMI-S	×	×	×	○	○	×	×
ARM7EJ-S	×	×	×	○	○	○	○
ARM720T	×	×	MMU	○	○	×	×

表1-2　ARM9微处理器的功能

微处理器	Cache（指令/数据）	TCM	存储管理	AHB	Thumb	DSP	Jazelle
ARM920T	16KB/16KB	×	MMU	○	○	×	×
ARM922T	8KB/8KB	×	MMU	○	○	×	×
ARM940T	4KB/4KB	×	MMU	○	○	×	×

表1-3　ARM9E微处理器的功能

微处理器	Cache（指令/数据）	TCM	存储管理	AHB	Thumb	DSP	Jazelle
ARM926EJ-S	可变	○	MMU	双AHB	○	○	○
ARM946E-S	可变	○	MMU	AHB	○	○	×
ARM966E-S	×	○	×	AHB	○	○	×
ARM968E-S	×	○	DMA	AHB-Lite	○	○	×

表1-4　ARM10E微处理器的功能

微处理器	Cache（指令/数据）	TCM	存储管理	AHB	Thumb	DSP	Jazelle
ARM1020E	32KB/32KB	×	MMU	双AHB	○	○	×
ARM1022E	16KB/16KB	×	MMU	双AHB	○	○	×
ARM1026EJ-S	可变	○	MMU或MPU	双AHB	○	○	×

表1-5　ARM11S微处理器的功能

微处理器	Cache（指令/数据）	TCM	存储管理	AHB	Thumb	DSP	Jazelle
ARM1136J(F)-S	可变	○	MMU	5×AMB	○	○	○
ARM1156T(F)-S	可变	○	MPU	4×AXI	○	○	×
ARM1176JZ(F)-S	可变	○	MMU	4×AXI	○	○	○

表1-6　ARM SecurCore的功能

微处理器	Cache（指令/数据）	TCM	存储管理	AHB	Thumb	DSP	Jazelle
SC100	×	×	MPU	×	○	×	×
SC110	×	×	MPU	×	○	×	×
SC200	可选	×	MPU	×	○	○	○
SC210	可选	×	MPU	×	○	○	○

表1-7　INTEL with ARM Base的功能

微处理器	Cache（指令/数据）	TCM	存储管理	AHB	Thumb	DSP	Jazelle
Strong ARM	16KB/8KB	×	MMU	×	×	×	×
XScale	32KB/32KB	×	MMU	×	○	○	×

1.3　ARM的广泛应用

ARM近10年来以高性能、低价位、低功耗、小体积等特色广泛地进入商业领域。

（1）无线通信领域。目前已有超过85%的一线通信设备采用ARM技术，ARM以其高性能和低成本，使其在该领域的地位日益巩固。

（2）通信产品。一般的ARM SoC都支持嵌入式操作系统。ARM在通信产品上已经有了完整的解决方案，如网络电话（VoIP）、ADSL Router和IP Router等。

（3）消费类电子产品。ARM技术在目前流行的MP3播放器、电视机顶盒（Set-up-Box）和游戏机中均得到广泛应用。

（4）数码相机系列。现在流行的数码相机和打印机，绝大部分采用了ARM技术。

（5）手机。ARM技术已经应用到手机行业的各个领域，可以说它结合了上述所有特色于一身。

（6）工业控制领域。由于网络通信功能的增加及嵌入式系统的支持，使得许多控制系统的控制由X86系列向ARM系列发展。

（7）安全控制系统。网络摄影机、4/16压缩处理机、数码录放机等安全自动化，也已逐渐采用ARM的核心技术。

以上对ARM所涉及的行业作了初步的介绍，下面提供几点作为读者选用ARM微处理器进行开发的参考。

（1）从速度上考虑。如果所做的产品人机接口比较多，不需要CPU工作太多时间，通常会选用ARM7的CPU，例如PDA、MP3播放器、录音笔等产品。但是如果需要CPU的工作量很大，就会考虑ARM9以上的CPU，如矩阵式硬盘。

（2）从计算上考虑。如视频压缩和解压缩的产品PMP（Portable Media Player），还有一些高端的手机也都有视频压缩和解压缩功能。此类产品会选用有浮点运算及DSP功能的ARM9E以上的CPU。

（3）从操作系统上考虑。一类是需要CPU提供MMU的操作系统，如WinCE或Linux等，就需要ARM9以上的CPU；另一类是不需要MMU的操作系统，如μClinux。

（4）从控制器接口考虑。除ARM微处理器核以外，几乎所有的ARM芯片都根据各自不同的应用领域，扩展了相关功能控制器，并整合在SoC之中，即控制接口。如USB控制器、IIS控制器、LCD控制器、键盘控制器、RTC、ADC和DAC等。设计者应根据产品的需求，尽量采用SoC内的控制器来完成所需的功能。这样既可简化系统的设计，同时又提高了系统的可靠性。

1.4 如何学习ARM

笔者一直认为想要进入ARM领域需要很高的门坎。它并不像8051这一类单片机。下面大致归纳了一些学习的顺序供读者参考。

（1）了解各种不同存储器的工作原理。因为ARM的标准接口是Memory Interface，特别是有关ROM/SDRAM的Memory Interface。

（2）学习ARM Core的汇编语言。如果对启动载入程序（Boot Loader）有兴趣，或者想要深入了解ARM，汇编语言是很好的工具。否则读者不一定要学习汇编语言，因为ARM主要希望使用者从C语言的眼光去看ARM SoC，这样可以加快开发的进度。

（3）C语言。如果不会C语言，那么学习ARM会十分困难。ARM常用的C语言，根据不同的环境，可分为ANSI C、GNUC或C++。

（4）嵌入式编译器。不管以后是否要在目标板上运行操作系统，笔者强烈建议，如果要了解ARM，要先使用ADS（ARM Develop Suit）编译器，因为此编译器无需任何操作系统，是最接近ARM的编译器，其他如GNUC或C++等编译器，多少都跟硬件或操作系统有关。

（5）嵌入式调试器。当然ADS 有一个搭配的调试器AXD，也是ARM建议使用的调试器，它的接口就是用标准的JTAG。

（6）硬件电路结构。它包含了电源电路、Reset电路、Clock电路及存储器电路，只有读者理解了这些硬件，才能说对ARM的硬件架构有了初步的认识。

（7）异常事件及中断。这是撰写Boot Loader程序的重点，撰写Boot Loader的读者可以多读这些方面的资料，如果对于低端的控制不想了解太多，可以跳过这部分，原厂一定会提供BSP（Board Support Package，目标板数据包）设备及数据。

（8）相关控制接口。控制接口是ARM SoC除了Memory Interface以外的另一个重要特色，不同的SoC会有不一样的控制接口。分别有UART、PWM Timer、RTC Timer、GPIO、WTD、USB、SPI等。

1.5 如何使用这本书

建议读者在阅读本书前最好有C语言设计的经验。C语言是学习ARM 16/32位处理器的基础，虽然ARM处理器有ARM mode（32位）及Thumb mode（16位）两种模式，限于篇幅的关系，本书对于比较少用的Thumb mode 不会介绍太多，如果读者对此有兴趣，可以登录ARM的官方网站http://www.arm.com/，下载相关资料。

本书分为3大部分：基础篇、原理篇和实例篇，所有篇幅均围绕如何编写引导程序及硬件设计，如果读者对于微处理器的相关知识有一定的经验，可直接阅读原理篇和实例篇，如果对原理没有太大的兴趣，也可直接从实例篇开始阅读。

本书所附的所有程序及其工程都经过笔者测试，保证能正常运行，如果读者需要有目标板实验，可直接联系嵌匠工作室的网站http://www.esdesigner.com.tw/洽询ESD_44B0开发板。因为本书的篇幅限制，所以大部分实例的工程无法在书中呈现，此目标板可作为有兴趣者的辅助教材。

1.6 参考资料

- ARM Software Development Toolkit Version 2.50 Reference Guide
- ARM Developer Suite Version 1.2 AXD and armsd Debuggers Guide
- S3C44B0X DataSheet

基 础 篇

第2章 ARM的常用存储器

在以后的章节中会有很多的篇幅介绍存储器的存取，ARM之所以得到广泛的应用，其中一部分原因就是因为它有一个功能非常强大的存储器控制器（Memory Controller），而且不同的ARM SoC会提供不同功能的存储器控制器。所以本章也无法把所有存储器逐一详细的介绍。

本章暂且介绍ARM SoC最常用的存储器：ROM/FLASH/SRAM、DRAM、SDRAM 3大类，其他的存储器不在本章探讨范围内。

2.1 存储器的分类与存取原理

存储器有几种不同的分类方式，以存取的方式可以分为对称式（Symmetrical）存储器和非对称式（Asymmetrical）存储器两种。

2.1.1 对称式

存储器主要的作用，就是供CPU在需要时存取储存代码和数据。我们可以将这种存储器比喻成图书馆中存放书籍的书架。不但要把书放到书架上，还要能够将书在需要的时候准确地找到并取出来。虽然都是书，但是每本书都是不同的。

用书和书架的原理来解释，如果有一个书架上有10行和10列格子（每行和每列分别都有0～9的编号），有100本书要存放在里面，那么我们使用一个行的编号加一个列的编号就能确定每一本书的位置。例如已知一本书的编号为87，那么我们首先选择第8行，然后找第7列就能准确地找到这本书了。每一本书都有一个不同的编号，我们就可以根据编号迅速、准确地将书取出和存放。对称式（Symmetrical）存储器也是同样原理。存储器中，数据总线是用来传入数据和传出数据的。对CPU来说，存储器就像图书馆中的书架，是一条长长且有很多空格的细线。每个空格都有一个惟一的地址与它相对应。如果CPU想要从存储器中存取数据，首先地址总线发送地址数据，确定要存取的数据，然后在等待若干个时钟周期之后，数据总线就会把数据传输给CPU。

图2-1可以帮助你理解这个过程。

图2-1中存储器的每一个储存空间都有一个惟一的地址线与它相连。当地址译码器接收到地址总线送来的地址数据后，会根据这个地址数据，确定CPU想要存取的数据所在的存储空间，然后数据总线就会把这个存储空间中的数据传送到CPU。因为每一组地址线代表了惟一的一组

数据，所以叫作对称式存储器。如果存储器的大小增加，那么相对应的地址线也会增加。如果存储器的大小减少，地址线也会相应的减少。

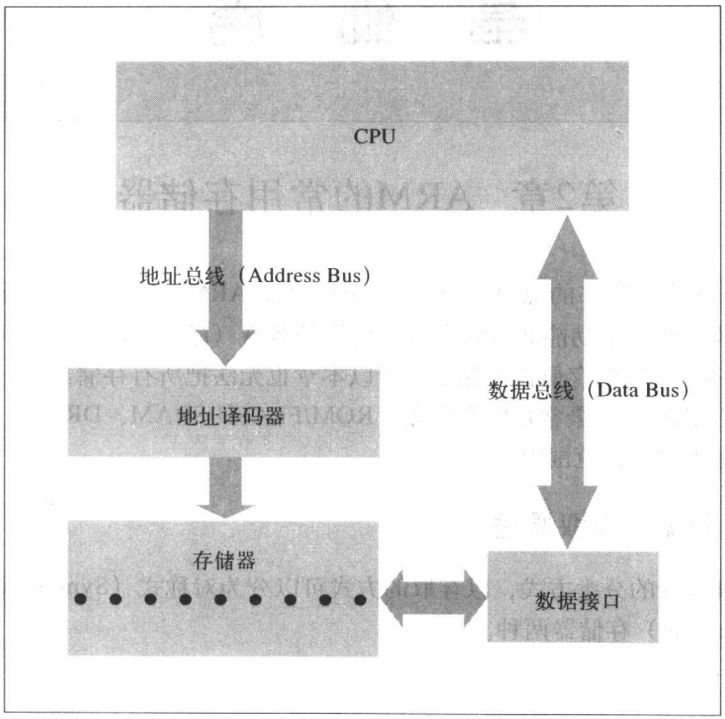

图2-1 对称式（Symmetrical）存储器架构图

上面所举的例子中，CPU在一行数据中每次存取一个单位的数据。而一个单位是由CPU的存取单位的大小来决定，有可能为8/16/32 bits（这是由不同系统的数据总线的位宽度所决定的）。

对称式存储器的特点是存取动作简单，缺点是它受地址线的限制。所以通常对称式存储器容量比较小。假设地址线是10条的话，我们能存取存储器的大小就是1024即（2^{10}）个单位。

2.1.2 非对称式

在设计大容量存储器时，为了降低成本，就要将储存信息的“空格”排列成很多行，即从线性到矩阵的处理方法。这样，如果要储存1024单位的数据，只要使用32×32的矩阵就能够达到这个目的了，那么原先10条地址线就变成了6条。原理如下：

（1）行及列都共享5条地址线，再选一条地址线作为行或列的选择。

（2）存取步骤为：先选择行，送出行地址；再选择列，送出列地址，此时可做读/写。

很明显，一个32×32的矩阵比一个1024行线性设备更紧凑，使用起来也更容易，这种矩阵结构在存储器越大时越能显示出其优势。非对称式存储器的存取原理如图2-2所示。