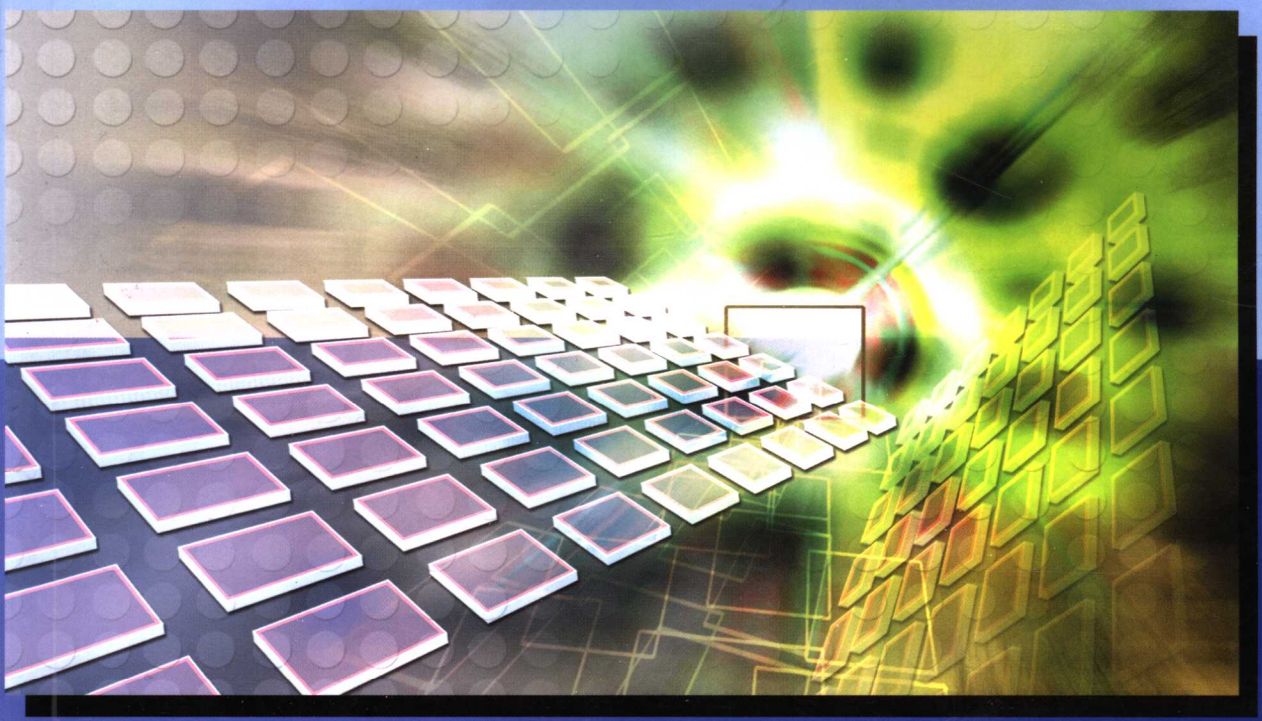
ARM开发工具ADS原理与应用



赵星寒　周春来　刘 涛　编著

北京航空航天大学出版社

ARM 开发工具 ADS 原理与应用

赵星寒　周春来　刘　涛　编著

北京航空航天大学出版社

内容简介

本书详尽地介绍了 ARM 开发工具 ARM Developer Suite(简称 ADS)的构成和应用方法。主要内容包括:ARM 调试的基本原理;ADS 中复杂的工程项目管理;ADS 中的文件和库;编译器和汇编器应用;ARM 中 C/C++语言应用;ARM 连接器应用和连接器设置;调试工具 AXD 应用;AXD 中的调试方法等。

本书可供电子信息类大学生、研究生或电子设计工程师阅读参考。

图书在版编目(CIP)数据

ARM 开发工具 ADS 原理与应用 / 赵星寒等编著. —北京:
北京航空航天大学出版社,2006.7
ISBN 7-81077-748-3

Ⅰ. A… Ⅱ. 赵… Ⅲ. 微处理器,ARM
Ⅳ. TP332

中国版本图书馆 CIP 数据核字(2006)第 070308 号

ARM 开发工具 ADS 原理与应用
赵星寒 周春来 刘 涛 编著
责任编辑 胡晓柏
*
北京航空航天大学出版社出版发行
北京市海淀区学院路 37 号(100083) 发行部电话:010-82317024 传真:010-82328026
http://www.buaapress.com.cn E-mail:bhpress@263.net
涿州市新华印刷有限公司印装 各地书店经销
*
开本:787×1 092 1/16 印张:18.75 字数:480 千字
2006 年 7 月第 1 版 2006 年 7 月第 1 次印刷 印数:5 000 册
ISBN 7-81077-748-3 定价:29.00 元

前　言

目前，32 位的 ARM 处理器正得到技术界越来越广泛的关注，其应用已经覆盖了通信、交通、军事、工业等各个领域。国内学习过 ARM 处理器的技术人员越来越多。ARM 处理器在高端产品的应用领域已经完全取代传统的 8 位单片机。随着 ARM 处理器生产成本的不断下降，在今后几年内，ARM 处理器的应用将更加普及。

ARM 公司在推出 32 位 RISC 结构的 ARM 处理器内核的同时，也推出了 ARM 处理器的开发工具 ADS(ARM Developer Suite)。其由两部分组成：一部分是由 ARM 编译器和 ARM 连接器组成的 CodeWarrior IDE，这是一个用于程序编译和程序连接的集成开发环境；另一部分是 ARM 调试器 AXD(ARM eXtended Debugger)，是一个功能强大的调试系统。

ARM 开发工具 ADS 为用户提供了两种方法来使用这些工具：图形方式和命令行方式。在图形方式中所使用的是集成的开发环境，所有的编辑、编译、连接和调试工作都在窗口中进行，开发人员和开发工具 ADS 之间的界面是一种通过鼠标操纵对话框的界面；命令行方式是一种 DOS 环境下的开发方法，开发人员通过键入命令行控制开发工具 ADS，命令行是由可选项组成的。

本书全面地阐述了 ARM 开发工具 ADS 的组成、结构和使用方法，从应用的角度出发，逐层深入地讨论了 ADS 各个层面的内容，不但详尽地说明了 ADS 的使用方法，也涉及了开发 ARM 时的一些基本原理，对于广大技术人员学习和使用 ADS 会有一定的帮助。

本书内容不包括 ARM 的基本结构、ARM 指令集、ARM 中断处理系统等，有关内容可以参阅作者编著的《从 51 到 ARM——32 位嵌入式系统入门》。

本书的基本内容简单介绍如下：

前 2 章介绍 ARM 开发的基本方法和工具，以及 ADS 系统的组成和基本使用方法。

第 3～6 章以及第 8 章介绍编辑器、浏览器、汇编器和编译器。编辑器可以用来编辑各种文本文件，使用方法和一般编辑器没有大的区别；浏览器是一个辅助的浏览工具，在使用 C/C++语言时，可以帮助用户对源程序中的函数、类等进行辅助的操作，如观察、修改、搜索、添加等，在对复杂的源程序进行编辑、修改时很有用处；汇编器可以把汇编语言源程序汇编成目标代码，与一般的汇编器的作用相同；编译器可以把 C/C++语言的源程序编译成目标代码。

在 ADS 中，工程项目是一个很重要的概念，连接和调试都是以工程项目为基本单位。源程序必须被工程项目所包含才可以应用。正确合理地设置工程项目，可以使调试工作进行得更加顺利。

ADS的所有工具都包含很多可选项，本书对这些选项都做了详尽的解释。在应用这些选项时，要注意这些选项之间的关联，本书推荐使用via格式文件描述这些选项。

第7章介绍C/C++语言的应用。在ARM的大多数实际应用中，都广泛使用C/C++语言。要注意C/C++语言的使用规则，在需要时可使用汇编语言和C/C++语言的混合编程方法，以提高效率和满足控制方面的要求。要注意编译器的可选项设置会对C/C++语言的应用有一定的影响。

第9章介绍了CodeWarrior IDE中部分选项设置和一部分操作命令的使用方法。

第10章是关于连接器的内容。ARM连接器中有很多可选项，这些可选项影响工程项目的生成，在使用连接器之前，要正确地设置这些可选项。对于比较复杂的映像文件或存储系统，使用scatter描述文件是一种比较灵活的方法。

第11章介绍ARM中的几个很有特色的功能，包括via格式文件，这是一个任何使用ADS的技术人员都应该熟悉的文件。除此之外，代码转换工具fromELF也是一个十分重要的文件，它可以帮助用户了解目标文件或映像文件的内容，如反汇编程序、符号表内容、调试信息等。

第12章和第13章介绍ARM调试工具AXD的应用。AXD是一个功能强大的调试工具，提供了多种辅助调试手段用来对用户程序进行调试，包括断点、观测点和观测项等；还可以提供一个控制台支持semihosting功能。调试器AXD还能够支持多目标系统的调试。

在本书最后，提供了按ADS中菜单顺序进行索引的附录，供使用中查找有关操作命令或设置面板以方便进行选项设置。

本书适合所有正在使用和正在学习ARM开发工具ADS的技术人员。

本书在编写过程中，周春来教授做了大量工作。本书第7章由我的同事刘涛编写，其余由作者本人编写。在编写过程中，得到张君、郑玉峰的大力帮助，同时，也得到出版社的大力支持，在此衷心的感谢。

这是第一本关于ADS方面的参考书，难免存在疏漏和误解，交流请使用zxhjeket@yahoo.com.cn。

赵星寒

2006年4月于北京

目 录

第1章

ARM 调试方法和工具

1.1 调试原理概述

1.1.1 传统调试方法

20 世纪 80 年代，单片机取代单板机所带来的变革之一就是单片机的仿真调试更加简单、灵活，所占用的目标资源更少。到目前为止，结构简单的单片机的仿真调试大致经历下列两个发展阶段：简单的单片机仿真简单的单片机和高档的单片机仿真简单的单片机。

1. 简单的单片机仿真简单的单片机

在早期，调试目标板上的 CPU 使用的是由相同 CPU（或相近 CPU）组成的仿真器，由仿真器上的 CPU 仿真目标板上的 CPU，仿真器上的 CPU 和目标板上的 CPU 引脚完全对应。仿真器通过串口或并口和主计算机相连。这种仿真方法优点在于：

- 仿真器结构简单。
- 仿真器上的 CPU 输入/输出口线的驱动能力与实际目标 CPU 输入/输出口线的驱动能力完全相同。
- 运行速度完全相同。

在单片机推广早期，这种仿真器受到了普遍欢迎。但这种仿真最大缺点是占用目标资源，仿真器上的 CPU 不但要仿真目标 CPU 运行，还要受仿真器控制，因此不能达到 100% 的仿真目的。例如，早期的 8051 仿真器是用 8051 做仿真头，这种仿真器往往占用 8051 的中断资源 INT0，因此无法调试目标板上与 INT0 有关的软件和硬件。

2. 高档的单片机仿真简单的单片机

当更高档的单片机出现以后，人们开始用高档的单片机仿真简单的单片机，这样就解决了仿真器占用目标资源的问题，因为高档单片机的功能更强大。这样所带来的问题是：仿真器输入/输出口线的驱动能力和实际目标 CPU 输入/输出口线的驱动能力可能不一致。例如，51

系列单片机引脚在低电平时的输入电流为 10 mA 以上，但很多使用高档单片机设计的仿真器都达不到这个驱动能力。这样所带来的另一个问题是由于仿真器的结构更加复杂，加大了仿真器的开发成本。

每种单片机（包括微处理器，以下同）都必须有自己的仿真调试方法和开发工具，只有这样，这种单片机才能够得到实际应用。有些单片机没有完全的仿真方法和工具，只能部分仿真。这些单片机是一些特殊应用的单片机，在开发应用中十分困难，其生命力不长久。随着科技的发展和制造工艺的进步，不断有新的单片机产生，但新的单片机必须有新的调试方法，结构复杂、功能强大、引脚数量很多的单片机使用传统的调试方法实现仿真调试是十分困难的，甚至是不可能的，因此，没有一种新的调试方法就没有功能更加强大的单片机。在这种背景下，出现了边界扫描技术，这种技术的出现解决了上述问题。

在以后的论述中，把单片机、微处理器、微控制器等不加区别，通称为单片机。

1.1.2 ARM 调试的特点

1. 使用边界扫描技术

传统的调试方法是用一个单片机仿真目标 CPU 对目标板进行系统调试，它具有下列局限性：

① 由于要使用一个比目标 CPU 更复杂的单片机做仿真器，就必须先解决这个更复杂的单片机的生产和仿真调试问题。如此类推下去，这是一个无法解决的问题。

② 必须为每一种单片机研制仿真器，这不但增加产品开发成本，在实际上也是很困难的。例如，有些在市场上已经销售十几年的单片机依然没有合适的仿真器。

③ 使用传统的调试方法必须把仿真器焊接或插接在目标板上，这对于引脚数量很大的贴片式目标 CPU，也是不可能的。

把边界扫描技术应用到单片机的调试中全面地解决了上述传统调试方法所带来的困难。ARM 处理器内核包含边界扫描结构（boundary scan architecture），是使用边界扫描技术实现调试的体系结构。近几年所生产的大多数结构复杂的单片机，都使用边界扫描技术为用户提供仿真调试方法。这种调试方法完全不同于传统的调试方法，具有下列特点：

① 需要把目标 CPU 焊接在目标板上。在调试过程中，执行用户指令和控制调试过程的是目标 CPU 本身；不需要其他单片机参与调试。

使用边界扫描技术实现系统调试的单片机（包括所有的以 ARM 为核的单片机）大都采用引脚数很多的贴片式封装形式，因此必须完全焊接才有可能调试。注意，必须保证单片机的最小系统正常工作。单片机的最小系统一般包括时钟系统、复位系统、电源系统和用于调试链接的 JTAG 接口。

② 仿真系统可以通用。在使用边界扫描技术实现仿真调试时，一个最大的优点就是仿真系统可以通用。例如，一个 ARM7 的仿真系统可以调试所有以 ARM7 为核的单片机，即使这些单片机来自不同的厂家，即使这些单片机的性能相差很大。但要注意的是，在对单片机内的程序存储器进行用户程序固化时，仿真系统不是通用的。

③ 仿真器与被调试的目标 CPU 的复杂程度无关。这是由 ARM 处理器的特点决定的。

④ 仿真系统开发成本比较低。因为系统能够通用，所以开发成本可以大幅度降低。虽然 ARM 处理器结构比较复杂，但是并没有增加仿真器的开发难度。

⑤ 结构简单。ARM 仿真器结构都比较简单，原因是调试中的大部分工作都由处理器完成。

2. 边界扫描技术

每个边界扫描单元的检测对象都是一条引线，并假定一个边界，只在这个边界位置检测和控制这条引线，对边界以外的结构不予关心。边界扫描单元如图 1.1 所示。

在图 1.1 中，在引线 A 和 B 之间插入一个边界扫描单元，当扫描单元不工作时，A 和 B 引线是“透明”的；当扫描单元工作时，A 和 B 引线也可以是“透明”的，也可以是不“透明”的。串行数据输出可以采样 A 端信号，也可以采样 B 端信号。在不“透明”的情况下，串行数据输入信号可以传送到 A 端，也可以传送到 B 端。这样就可以达到检测和控制引线 A 和 B 的目的。

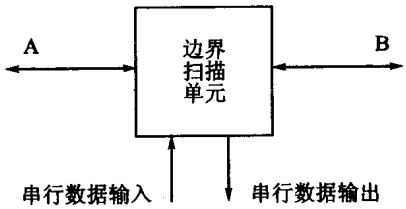


图 1.1　边界扫描单元

1.1.3　ARM 调试原理

ARM 处理器内核由 3 个部分组成，如图 1.2 所示。

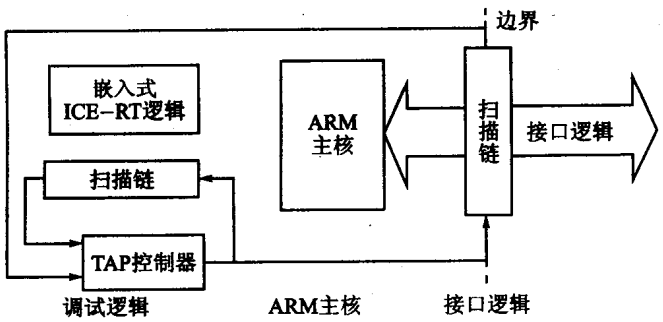


图 1.2　ARM 处理器内核的组成

除 ARM 主核外，ARM 处理器内核还包括调试逻辑和接口逻辑。对于不同版本的 ARM 处理器内核，调试逻辑和接口逻辑略有差别。ARM7TDMI 的调试逻辑包括 3 个部分：

① 嵌入式 ICE—RT 逻辑。支持实时调试的嵌入式在线仿真逻辑。实现调试系统的在线调试功能，可以通过两种方式进行访问：

- 通过 ARM 处理器主核访问。ARM 处理器主核把嵌入式 ICE—RT 作为外部协处理器 14，使用协处理器指令 MCR 和 MRC 进行访问。
- 通过扫描链进行访问。调试系统通过扫描链 2 访问嵌入式 ICE—RT。

② TAP 控制器。调试系统通过 TAP 控制器控制扫描链的工作。

③ 由多个扫描链组成的扫描逻辑。供用户使用的共有 3 个扫描链。

传统的仿真调试过程是仿真器模拟目标 CPU 的引脚发出动作，因此，主调试系统可以控制和检测目标 CPU 的引脚。与传统的调试原理不同，在 ARM 调试中，把与 ARM 主核相链接的接口逻辑作为边界，通过边界扫描单元控制和检测接口逻辑，从而达到调试的目的。从图 1.2 可以看出，这种原理有两个不可比拟的优点：

- 扫描逻辑只与 ARM 处理器内核有关，而不关注链接在接口逻辑上的外部功能模块，因此，只要 ARM 处理器内核相同，仿真系统就可以通用。
- 调试逻辑与 ARM 主核无关，因此调试不占用目标 CPU 资源。

对每一个调试中所关注的接口逻辑都有一个边界扫描单元。多个边界扫描单元串接在一起，组成扫描链，扫描链可对一组接口逻辑进行控制和检测。如图 1.2 中所示，在嵌入式 ICE—RT 逻辑结构外围也有一条扫描链。图 1.3 和图 1.4 是边界扫描单元简图，每个扫描单元都包含输入扫描单元和输出扫描单元两部分。

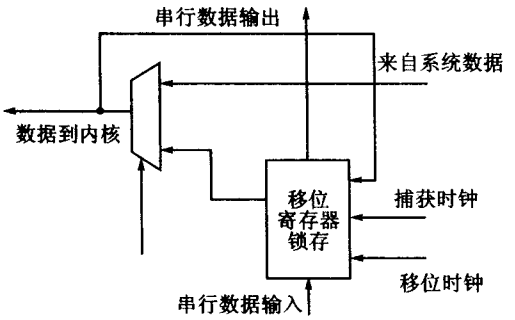


图 1.3　输入扫描单元

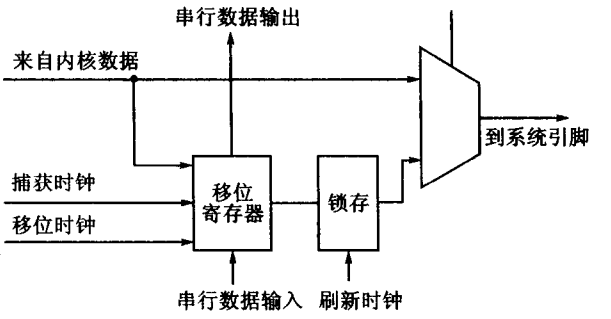


图 1.4　输出扫描单元

在图 1.3 和图 1.4 中，扫描单元的一侧是 ARM 处理器主核，另一侧是外部(片内)功能模块，这里把外部所有模块统称为系统。在没有扫描单元或扫描单元不工作的情况下，主核到系统的引线是“透明”的。

输入扫描单元在调试时会把从系统到主核的值复制到移位寄存器，并在移位时把这个值串行输出。由输入单元传送给主核的值可以来自系统，也可以来自移位寄存器，具体来自哪里由 TAP 控制器控制。

输出扫描单元在调试时会把从主核到系统的值复制到移位寄存器，并在移位时把这个值串行输出。由输出单元传送给系统的值可以来自主核，也可以来自移位寄存器，这也由 TAP 控制器控制。

在调试时，输入扫描单元和输出扫描单元不能同时工作，它们的工作过程完全受 TAP 控制器控制。可以看出，在调试时，调试系统可以隔离 ARM 主核和系统之间的关联。在输入扫描单元工作时，调试功能可以通过串行数据输入把指令或数据传送到 ARM 处理器内核，也可以通过串行数据输出把来自系统(如来自存储器)的数据读出到调试系统；在输出扫描单元工作时，调试系统可以通过串行数据输入把数据传送到系统，也可以通过串行数据输出把来自主核的数据读出到调试系统。

边界扫描单元的功能简化框图如图 1.5 所示。调试系统所关注的每条引线都包含一个边界扫描单元 BSC(Boundary Scan Ceu)，把多个这样的扫描单元串联在一起，组成一个扫描链。

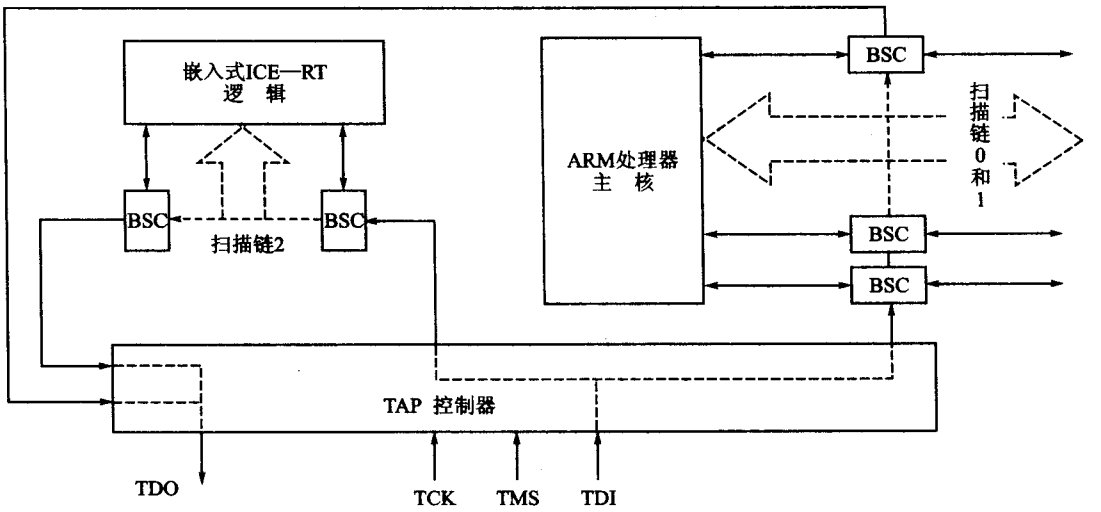


图 1.5　扫描单元和扫描链组成

在 ARM7TDMI 处理器中，主要的扫描链共有 3 条：扫描链 0、扫描链 1 和扫描链 2。

- 扫描链 0：允许调试系统对整个 ARM7TDMI 处理器主核外围进行访问，扫描链 0 长度为 105 位。包括 32 位数据总线和 32 位地址总线。
- 扫描链 1：是扫描链 0 的一部分，长度是 33 位，包括 32 位数据总线和 1 位控制位。扫描链 1 是扫描链 0 的前 33 位。
- 扫描链 2：允许调试系统对嵌入式 ICE—RT 逻辑中的寄存器进行访问。嵌入式 ICE—RT 逻辑中有多个寄存器，调试系统通过扫描链中串行数据的输入/输出访问这些寄存器。扫描链 2 长度是 38 位，包括 32 位数据、5 位地址和 1 位读/写控制位。

下面通过图 1.5 简单介绍 ARM 处理器调试逻辑的工作过程。

(1) JTAG 接口

在图 1.5 中，TAP 控制器外部有 4 条引线，分别是测试数据输入 TDI、测试数据输出 TDO、测试时钟 TCK 和测试模式选择 TMS。这 4 条引线是芯片的外部引脚，连接到外部调试系统。TDI 和 TDO 是串行数据输入/输出，TCK 用做同步时钟。由这 4 条线(还有一条可选线 TRST)组成的接口一般称为 JTAG 接口，其符合 IEEE 1149.1 标准。这个标准是一个关于测试访问接口和边界扫描结构的标准。

JTAG 接口是目标 CPU 在调试时与外部连接的最主要的接口。

(2) TAP 控制器

TAP(Test Access Port，测试访问接口)控制器，控制扫描链的动作。

(3) 嵌入式 ICE—RT 逻辑

Embedded In Circuit Emulator Real Time，嵌入式实时在线仿真逻辑。在仿真调试过程中，通过该模块可以设置断点。该模块中有多个寄存器，这些寄存器在调试时通过串行工作的扫描链访问。

(4) 仿真调试的工作过程

由外部调试器发出的调试命令经由 JTAG 接口 TDI 引线进入扫描链，由 TAP 控制器控

制扫描链的工作，对处理器进行调试；处理器的寄存器和系统内的数据经由扫描链串行到TDO引线，然后通过JTAG接口传送到调试器。

1.2 ARM调试方法

ARM调试系统应该包含调试主机、ARM仿真器(协议转换(protocol converter)等)和目标板3个部分。调试主机即一台安装有开发工具软件的通用个人计算机；目标板即被调试对象，在这块目标板上应该焊接一片所使用的目标单片机。ARM调试也有多种方法，按其原理分类叙述如下。

(1) 基于JTAG的调试方法

这是ARM系统最常用的调试方法。调试系统的结构如图1.6所示。

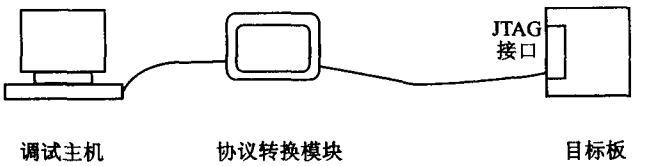


图1.6 调试系统原理图

调试主机上必须安装的工具包括程序编辑和编译系统、调试器和程序所涉及的库文件。目标板必须含有JTAG接口。在调试主机和目标板之间有一个协议转换模块，一般称为调试代理，其作用主要有两个：一个是在调试主机和目标板之间进行协议转换；另一个是进行接口转换，目标板的一端是标准的JTAG接口，而调试主机一端可能是RS-232串口，也可能是并口或是USB接口等。

一般接触的ARM仿真器大都是基于JTAG的仿真器。

(2) 基于Angel的调试方法

基于Angel调试方法的基本原理是：位于目标板上的CPU已经固化了一个完整的调试监控程序，这个监控程序可以接受来自调试主机的调试命令，并执行这些命令，如设置断点、单步运行、读/写存储器等；同时，这个监控程序也可以把数据传送到调试主机。

使用Angel调试方法的前提是：

- 目标板已经稳定工作，目标单片机的最小系统硬件没有问题；否则，无法调试。
- 被调试的目标单片机中已经固化了一个完整的调试监控程序。分为两种情况：一是监控程序由JTAG仿真器固化完成，这时必须先使用JTAG调试方法；二是监控程序由专门的程序写入设备完成，一般以ARM为核的单片机这种情况比较少。
- 调试主机和被调试的单片机之间，可以通过串口、并口或以太网口等实现通信，这个接口是目标单片机的外部输入/输出引脚(非JTAG接口)，因此占用的是用户资源。
- 有一个稳定的可以固化在目标单片机内的调试监控软件。

Angel调试方法不使用JTAG接口，但这种方法占用用户资源，主要有：

- 占用内部程序存储器以保存调试监控程序；