

ARM JTAG实时仿真器 安装使用指南

Version 1.0

2003. 2. 5

目 录

第一章 ARM JTAG介绍

1.1 ARM JTAG产品介绍	3
1.2 ARM JTAG硬件介绍	4
1.2.1 ARM JTAG接口定义	4
1.2.2 ARM JTAG接口电平	4
1.2.3 TCK信号频率设置	5
1.2.4 目标板JTAG接口设计	5
1.2.5 RTCK时钟	5
1.2.6 14脚和20脚JTAG接口转换	5

第二章 ARM开发系统安装

2.1 硬件连接	7
2.2 软件安装	7

第三章 ARM JTAG使用

3.1 Multi-ICE Server使用	9
3.1.1 Multi-ICE Server菜单选项	9
3.1.2 Multi-ICE Server状态	11
3.1.3 Multi-ICE Server中的JTAG参数设置	14
3.1.4 Multi-ICE Server配置	16
3.2 调试系统架构体系	17
3.3 ADS V1.2与ARM JTAG连接	18
3.3.1 连接AXD	18
3.3.2 连接CodeWarrior	19

第四章 最终应用程序烧写

4.1 应用程序在ARM板上运行原理	21
4.2 生成Boot.bin	21
4.2.1 系统初始化	21
4.2.2 生成BIN文件	21
4.3 烧录Boot.bin	23

附录1 TCK频率设置	24
附录2 常见问题	26

第一章 ARM JTAG介绍

1.1 ARM JTAG产品介绍

ARM JTAG是用于ARM处理器内核软件调试的专用工具符合IEEE 1149.1规范与ARM原装Multi-ICE完全兼容它可以很好的与ADS V1.2集成开发环境相结合支持全线ARM处理器内核可以满足更多用户对ARM处理器内核软件的开发调试需求



图1-1 ARM JTAG

ARM JTAG主要特征如下

- ☐ 支持所有内建embedded-ICE逻辑单元的ARM处理器
- ☐ 支持多个内核系统
- ☐ 连接简便兼容不同电压的目标系统
- ☐ 除JTAG扫描链外不占用目标板上的其它任何资源
- ☐ 支持实时硬件断点
- ☐ 用户可通过JTAG修改寄存器存储器内容
- ☐ 支持程序下载及实时调试
- ☐ 通信速度快最高可达10Mbps
- ☐ JTAG速度可配置以满足不同调试对象的需求
- ☐ 支持所有符合RDI 1.50或RDI 1.51规范的调试工具软件
- ☐ 提供丰富的例程和使用说明
- ☐ 支持多种目标平台如ATMELSamSungIntelPhilipsSharpCirrus LogicTricend等
- ☐ 体积轻巧性能稳定

ARM7TDMI ARM710T ARM7TDMI-S
ARM7DMI ARM720T ARM740T
ARM7TDI-S ARM7EJ-S ARM926EJ-S
ARM9TDMI ARM9E-S ARM940T
ARM946E-S ARM920T ARM966E-S
ARM1020E ARM922T ARM966E-S
Intel XScale PXA210 PXA250 80200

表1-1 ARM JTAG支持的ARM内核

支持的操作系统有

- ☐ Windows 95/98/NT/2000/ME/XP
- ☐ X86 RedHat Linux 6.2/7.1/7.2

1.2 ARM JTAG硬件介绍

1.2.1 ARM JTAG接口定义

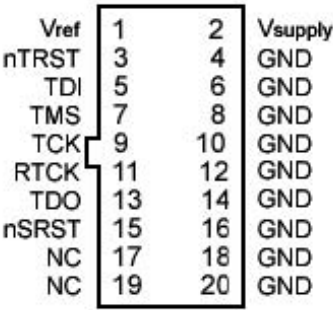


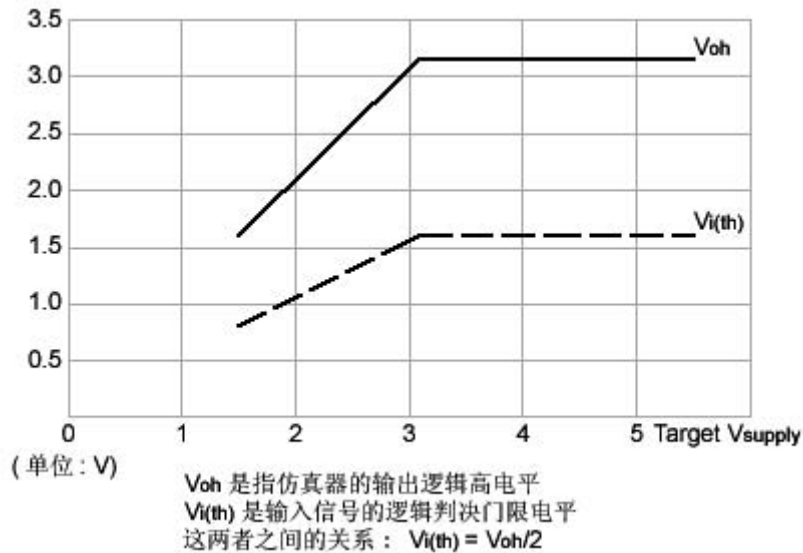
图1-2 20脚JTAG接口定义

管脚 编号	信 号	输入/输出方 向	信 号 说 明
1	V _{ref}	Input	接口信号电平参考电压一般直接连接Vsupply
2	V _{supply}	Input	电源
3	nTRST	Output	可选项JTAG复位在目标端应加适当的上拉电阻以防止误触发
4	GND		接地
5	TDI	Output	Test Data In
6	GND		接地
7	TMS	Output	Test Mode Select
8	GND		接地
9	TCK	Output	Test Clock
10	GND		接地
11	RTCK	Input	可选项Return Test Clock由目标端反馈给仿真器的时钟信号用来同步TCK信号的产生不使用时直接接地
12	GND		接地
13	TDO	Input	Test Data Out from target to仿真器
14	GND		接地
15	nSRST	Input/Output	可选项System Reset与目标板上的系统复位信号相连可以直接对目标系统复位同时可以检测目标系统的复位情况为了防止误触发应在目标端加上适当的上拉电阻
16	GND		接地
17	NC		系统保留
18	GND		接地
19	NC		系统保留
20	GND		接地

表1-2 20脚JTAG接口说明

1.2.2 ARM JTAG接口电平

仿真器推荐的正常使用电压为2.5 - 5.0V最大不超过2.0 - 5.5V随着电源电压的变化接口的逻辑高电平和判决门限电平都将随之变化这使得仿真器能够与不同电压和目标系统在逻辑上兼容



1.2.3 TCK信号频率设置

为了与不同速率的目标对象相兼容JTAG口的通信速率应调整到一个合适的水平数据传输速率是由TCK信号的频率来决定的TCK信号的真实频率跟设置值之间的对应关系参见附录1

1.2.4 目标板JTAG接口设计

目标板使用与仿真器一样的20脚针座RTCK和nTRST这两个信号根据目标RSIC是否提供对应的引脚来选用nSRST则根据目标系统的设计考虑来选择使用

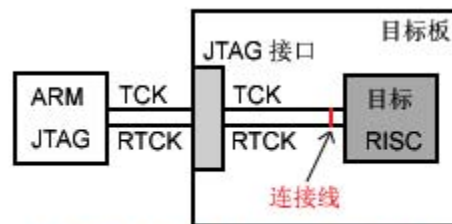
在目标系统的PCB设计中最好把JTAG接口放置的离目标RISC近一些如果这两者之间的连线过长会影响JTAG口的通信速率

另外电源的连线也需要加以额外考虑因为仿真器要从目标板上吸取超过100mA的大电流

1.2.5 RTCK时钟

RTCK信号用来同步仿真器和目标板之间的通信而不用关心TCK信号的具体频率在没有收到目标系统的反馈时钟信号之前仿真器不会触发一个新的TCK

如果不使用RTCK功能这个信号脚可以简单的接地处理如果使用了RTCK功能把这个脚直接连向ASIC中对应的管脚如果激活了RTCK功能但RISC并没有提供对应原管脚可以把RTCK和TCK这两个信号连在一起但在连线中最好遵循下面的原则这一点在边线比较长的时候尤其重要目的是确保在TCK时钟到达目标点的时候才产生反馈信号



注意: 连接线应尽可能的靠近目标 ASIC

图1-4 RTCK连接图

1.2.6 14脚和20脚JTAG接口转换

有些系统采用一种14脚的JTAG插座这两类接口的信号排列如下

Vref	1	2	Vsupply		
nTRST	3	4	GND	Vref	1
TDI	5	6	GND	nTRST	3
TMS	7	8	GND	TDI	5
TCK	9	10	GND	TMS	7
RTCK	11	12	GND	TCK	9
TDO	13	14	GND	TDO	11
nSRST	15	16	GND	VCC	13
NC	17	18	GND		14
NC	19	20	GND		

图1-5 不同的JTAG接口

这两类接口之间的信号电气特性一样因此可以把对应的信号直接相连进行转换

第二章 ARM开发系统安装

ARM JTAG物品清单

- ☐ ARM JTAG硬件实时仿真器
- ☐ 20针标准ARM JTAG接口扁平连接线缆一条
- ☐ 并口连接线缆一条
- ☐ 直流5V稳压电源一个
- ☐ ARM开发系统安装光盘及使用手册一套

2.1 硬件连接

ARM JTAG使用标准的25芯并口插座和20针的JTAG插座作为接口与PC的连接线缆使用标准的25芯并口连接线缆与目标板的连接线缆使用20芯的IDC宽带线缆有的目标板可能使用14脚的JTAG插座信号定义和JTAG接口的转换方法请参见1.2.6节内容

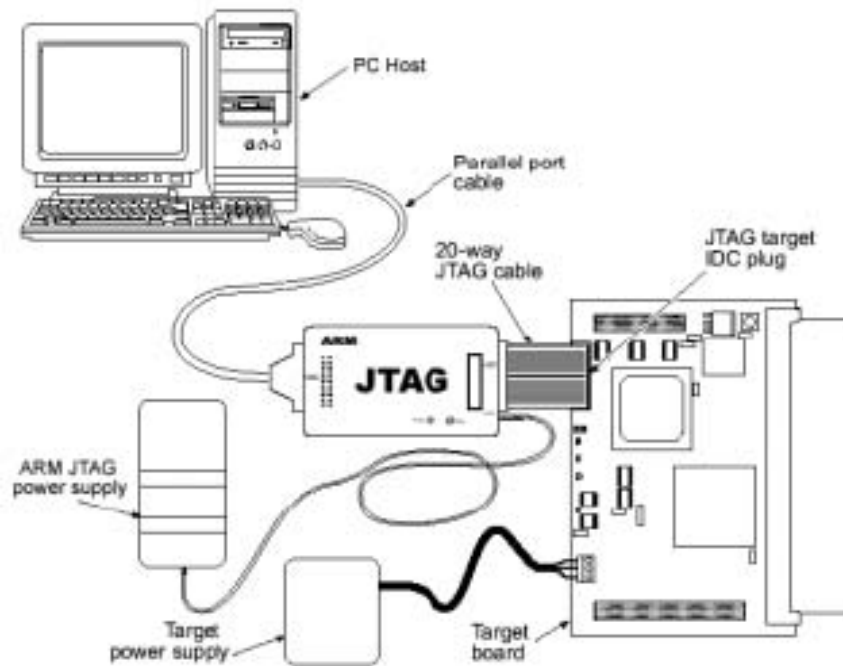


图2-1 硬件连接示意图

2.2 软件安装

开发PC主机系统需求

- ☐ Pentium IBM兼容PC带有可用的并口和串口
- ☐ 200MHz以上CPU
- ☐ 32MBFor Win98或64MBFor Win2000以上内存空间
- ☐ 300MB以上空闲硬盘空间
- ☐ CD-ROM驱动器
- ☐ 建议使用Microsoft Windows 98或2000操作系统

ARM开发系统安装步骤如下

1 把ARM开发系统软件光盘插入CD-ROM系统自动运行进入如图2-2画面如若未出现此画面请手动运行光盘根目录下autorun.exe文件



图2-2 ARM集成开发环境主界面

2 单击上述画面进入ARM开发系统程序安装选择主界面在程序安装选择主界面中按顺序依次点击需要安装的程序系统会自动进行安装

安装程序说明

- ☐ **ARM JTAG Emulator** 安装ARM JTAG驱动程序
- ☐ **ARM Developer Suite** 安装ADS V1.2集成开发环境
- ☐ **ARM Update** 运行注册程序

注意 在运行注册程序之前请确认已正确安装了**ARM JTAG驱动程序**和**ADS V1.2集成开发环境**



图2-3 ARM开发系统程序安装选择界面

3 程序全部安装完毕后点击EXIT退出

第三章 ARM JTAG使用

在使用ARM JTAG进行系统开发之前请先确定系统硬件连接正确并已连接好电源和软件安装正确

3.1 Multi-ICE Server使用

Multi-ICE是ARM JTAG的配置程序通过它可以使ARM JTAG与目标板建立通讯连接并能够反馈目标板上ARM处理器的硬件信息

在PC开发主机上选择开始>程序>ARM Multi-ICE v2.2>Multi-ICE Server进入Multi-ICE Server主界面

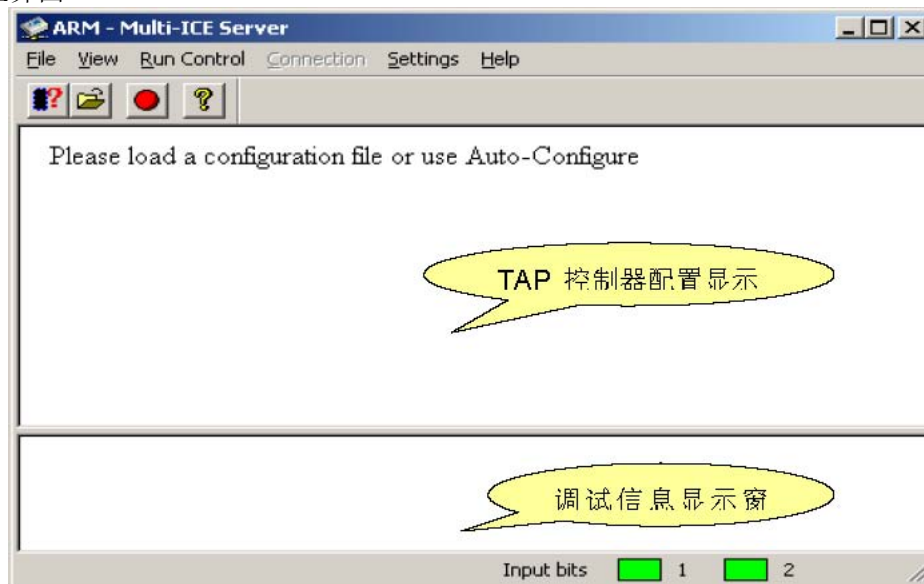


图3-1 Multi-ICE Server主界面

3.1.1 Multi-ICE Server菜单选项

这里介绍一下Multi-ICE程序中各个菜单及里面的每个选项

1 工具栏

在工具栏里总共有四个功能按钮



图3-2 工具栏按钮

2 File菜单

- ☐ Load Configuration 读取用户指定的配置文件对仿真器进行手动配置
- ☐ Auto-Configure 自动检测和配置目标系统
- ☐ Auto-Configure at 20KHz 自动检测和配置目标系统并且设置TCK信号的频率为20KHz
- ☐ Reset Target 对目标系统复位有效的复位信号在Setting/JTAG settings菜单或配置文件中设置

- ☐ **Log** 输出信息存储到指定的日志文件中
- ☐ **Set Log File** 用于指定日志文件的路径和名字
- ☐ **Recent File List** 显示最近调用过的配置文件路径
- ☐ **Exit** 退出程序

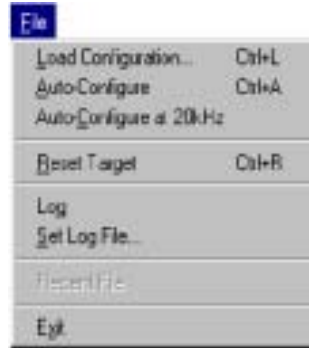


图3-3 File菜单

3 View菜单

这个菜单用于控制Multi-ICE程序界面外观和显示信息

- ☐ **Toolbar** 关闭或打开工具栏
- ☐ **Status Bar** 关闭或打开状态栏
- ☐ **RPC Calls** 允许或屏蔽RPC信息在调试信息窗口中显示
- ☐ **Clear Debug Window** 清空调试信息窗口中的显示信息



图3-4 View菜单

4 Run Control菜单



图3-5 Run Control菜单

这个菜单中的选项控制各个处理器的运行和停止

- ☐ **Independent** 使每个目标系统直接相互独立不进行交互通信缺省状态有效
- ☐ **All Run** 启动所有的目标系统
- ☐ **All Run/Stop** 当收到一条启动指令时启动所有的目标系统当任何一个目标系统停止时

其它的系统也一起停止

- ☐ Custom 执行用户自定义的设置
- ☐ Set-up Custom 打开用户自定义设置窗口设置各个目标系统之间的交互方式
- ☐ Load Settings 读取以前保存下来的设置文件
- ☐ Save Settings 把当前设置保存到文件

5 Connection菜单

这个菜单在调试程序和Server建立连接之后才有效它会给每一个TAP控制器分配一个菜单项为用户提供单独删除某个TAP控制器的选项



图3-6 Connection菜单

6 Settings菜单

这个菜单用来设置各种接口信息

- ☐ **Port Settings** 显示并口设置对话框用来选择并口的地址和是否使用4-bit通信模式同时显示当前的并口类型设置在PC机的BIOS中要把并口模式选择为EPP
- ☐ **User Output Bits** 当然版本的Server程序保留使用
- ☐ **JTAG Settings** 显示JTAG口的设置对话框用来设置JTAG口相关的时序信息和复位行为如果调用了配置文件则相应的设置会在窗口中反映出来
- ☐ **Start-up Options** 显示设置启动选项的对话框



图3-6 Settings菜单

7 Help菜单

- ☐ **Help Topics** 启动Multi-ICE的帮助系统
- ☐ **About Multi-ICE Server** 显示软件版本信息



图3-7 Help菜单

3.1.2 Multi-ICE Server状态

在Multi-ICE Server的TAP信息显示区可以直观的显示当前系统状态下面通过单内核目标系统为例进行说明

1 配置后的状态

Multi-ICE Server进行配置之后显示状态如图3-8所示

通过自动检测显示目标是一个ARM7TDMI的单内核系统在TAP配置显示区用图形的方式直观的显示出了检测结果内核的类型名称ARM7TDMI显示为绿色并且前面有一个表示内核状态的字母X表示Multi-ICE Server目前还没有连接到任何调试程序中

双击图3-8中的处理器类型名字会弹出目标处理器附加信息窗口

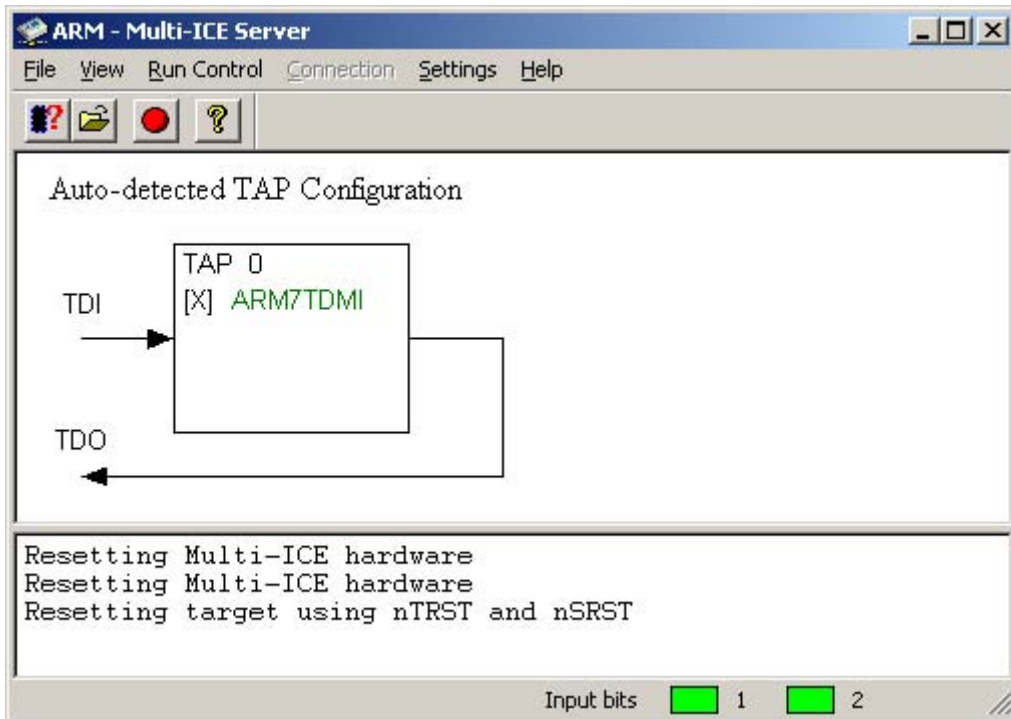


图3-8 Multi-ICE Server配置后状态



图3-9 目标处理器附加信息窗口

注意处理器类型名称前面的字母叫做状态位共有四种状态

- ☐ [S] 处理器处于暂停状态
- ☐ [R] 处理器忙运行状态
- ☐ [D] 处理器处于下载状态
- ☐ [X] 处理器类型未知或没有被调用

2 连接后的状态

如果有调试程序通过MultiICE.dll与Multi-ICE建立了连接则Multi-ICE的状态变成如图3-10所示此时处理器前面的状态字母变为S表明已建立连接但正处于空闲状态处理器名字变为红色同时信息窗口中的信息增加了相应的内容

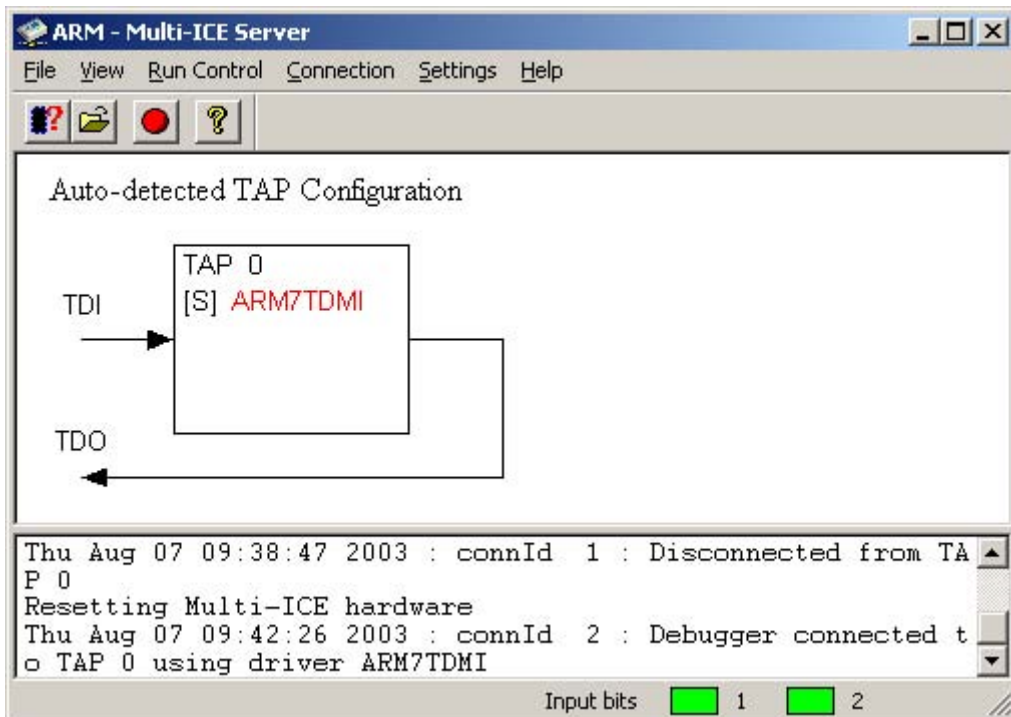


图3-10 Multi-ICE Server连接后的状态

3 激活后的状态

Multi-ICE的活动状态有下载和运行两种图3-11是代码下载时候的状态当下载操作完成后状态字母又马上变回X图3-12是代码运行中的状态当停止运行后状态字母会变回S

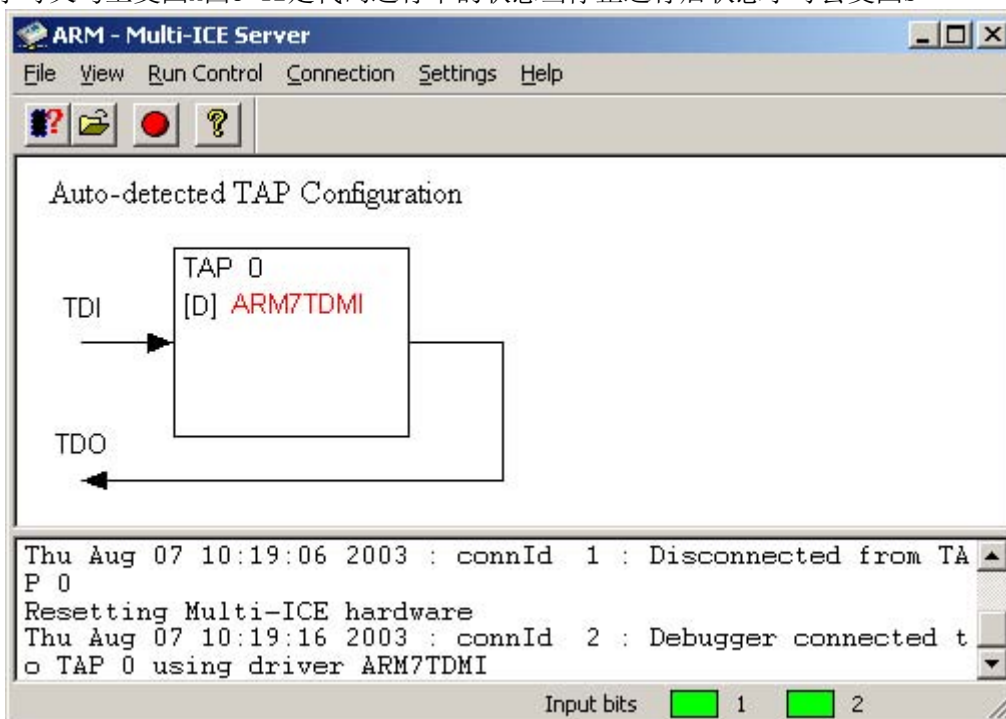


图3-11 Multi-ICE Server代码下载时的状态

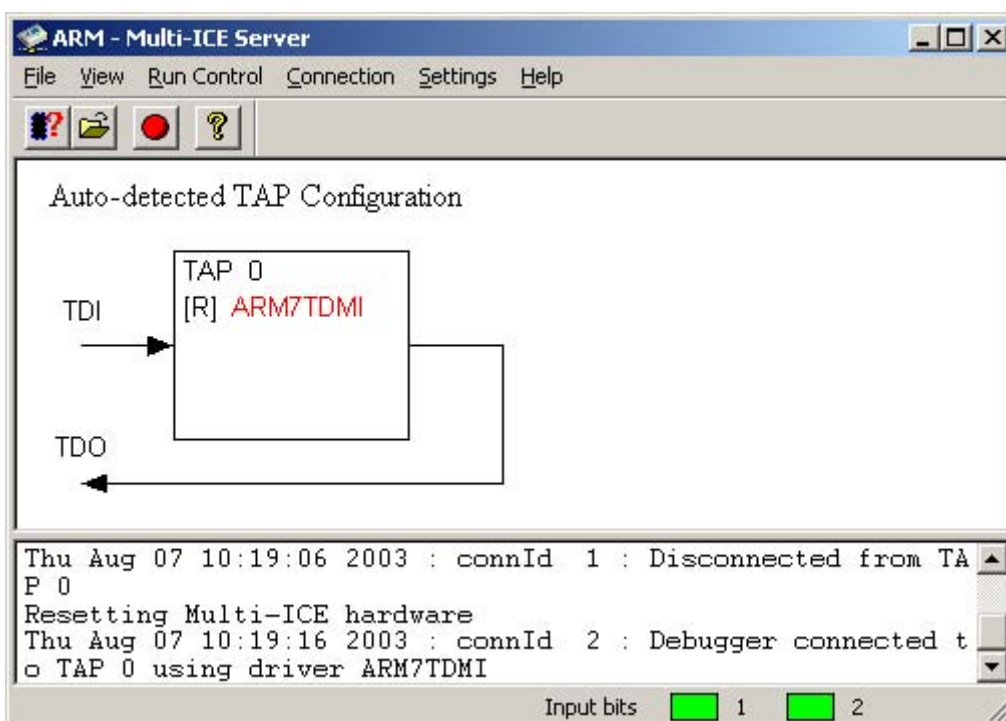


图3-12 Multi-ICE Server代码运行中的状态

3.1.3 Multi-ICE Server中的JTAG参数设置

1 并口设置

并口设置菜单位于Settings>Port Settings



图3-13 并口设置

- ☐ **Port Address** 选择使用的并口地址有三个可选项
 - AUTO 自动选择系统默认选项
 - LPT1 选择LPT1
 - LPT2 选择LPT2
- ☐ **Force 4-bit access** 使用4-bit的数据传输方式
- ☐ **Current port Mode** 显示并口类型显示BIOS中并口设置状态为只读选项

注意并口类型是在PC的BIOS中进行设置的通常提供四种类型

- ☐ Basic type 部分BIOS提供值为Default或SPP等
- ☐ EPP
- ☐ ECP
- ☐ EPP + ECP

由于ARM JTAG使用双向的并口数据总线通常ECP或EPP类型能够符合要求但在一些比较新的BIOS版本中可能要选用基本类型而不是ECP或其它增强型由于历史上的原因并口规范和IEEE1283协议的执行存在弹性所以不同计算机主板厂商在并口设计上存在一些差异当第一次使用ARM JTAG时需要对并口类型设置多作几次试验目前我们推荐用户选择EPP类型

2 时钟设置

时钟设置菜单位于Settings>JTAG settings

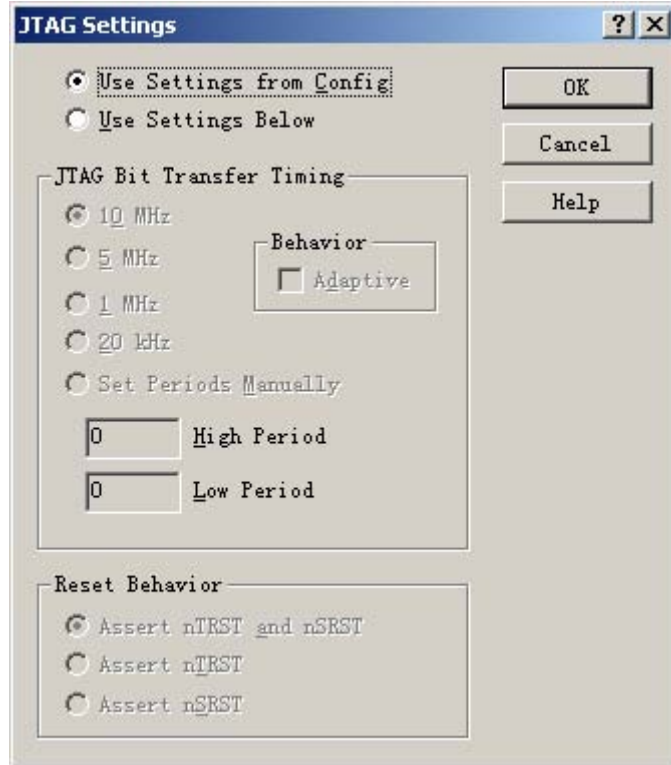


图3-14 ARM JTAG时钟设置

- ☐ **JTAG Bit Transfer Timing** 设置TCK信号频率如果需要手动设置请参见附录1
- ☐ **Behavior** 选择使用RTCK功能
- ☐ **Reset Behavior** 选择哪组信号在按下复位按钮后有效

3 启动选项设置

启动选项设置菜单位于Settings>Start-up Options这部分用来设定Multi-ICE启动时的参数和选项

- ☐ **Network Settings** 提供两个选项
 - Allow Network Connections 使用网络连接功能启动时自动检测网络设置情况
 - Start Portmap Service 仅在上一项选中时有效
- ☐ **Start-up Configuration** 设置Multi-ICE Server程序启动后自动执行的操作



图3-15 Multi-ICE启动选项设置

3.1.4 Multi-ICE Server配置

Multi-ICE Server在运行时需要目标处理器的配置信息比如内核类型和IR指令寄存器长度等等对Multi-ICE Server进行配置有自动配置和手动配置两种方式

1 自动配置

在Multi-ICE Server程序中选择File>Auto-configure菜单或者工具栏上的自动配置按钮如果目标处理器是ARM族的内核ARM JTAG能够自动检测到并进行相应配置并把结果显示在Multi-ICE Server的显示窗口中

注意有时候检测ARM710T/720T/740T/940T这一系列内核时会返回一个UNKNOWN结果这是ARM报告的一个硬件bug这时候就只能使用手动配置的方法

经过自动配置后JTAG的通信速率会自动设置为10MHz用户可以在Settings>JTAG Settings菜单中按自己的要求选择合适的速率

2 手动配置

手动配置是通过调用一个用户预先定义好的配置文件来完成对Multi-ICE Server的配置配置文件可以从File>Load configuration菜单中调入配置文件为文本格式文件文件后缀名为.cfg

配置文件主要包括以下几个部分

- ☐ Title
- ☐ TAP controller
- ☐ Devices attached to each controller
- ☐ JTAG timing information
- ☐ Other options

下面是配置文件的格式示范

[TITLE]

Double cores configuration demo 给配置方案命名

[TAP0] 目标系统中包含TAP0控制器
 ARM7TDMI TAP0控制上连着一个ARM7TDMI核
 [TAP1] 目标系统中包含TAP1控制器
 ARM7TDMI-S TAP1上连着一个ARM7TDMI-S核
 如果有更多的TAP控制器和内核依次往下增加
 [Timing] JTAG口的时序设置
 High = 9 TCK信号的高电平时间
 Low = 9 TCK信号的low电平时间
 Adaptive = ON RTCK功能开关ON或OFF
 [TAPINFO]
 YES
 [Reset]
 nTRST

通常情况下目标系统只有单内核仅使用 [TAP0] 就可以了对那些自动配置不能识别的内核需要在配置文件中指定好正确的内核类型然后在Multi-ICE Server中调入对应的配置文件就可以了

TAPINFO选项主要是为RISC开发人员测试芯片时提供的当该选项打开时ARM JTAG完成正常的配置工作后会继续从目标RISC中读取内核的其它信息供设计人员分析这些附加信息可以从双击Multi-ICE Server的TAP显示窗口中的TAP控制器图标而弹出的信息窗口中得到

使用自动配置时TAPINFO是一直打开的当调用用户自定义的配置文件时TAPINFO的默认状态时关闭

Reset选项用来定义ARM JTAG的复位动作当用户按下Multi-ICE Server工具栏上的复位按钮时Reset选项内定义的信号就有效合法的选项包括nTRST或nSRST或者两者都选中

Timing部分定义了JTAG端口的时序信息关于TCK时钟参数的设置请参见附录1

在一个配置文件中只有TAP控制器和内核类型的定义是必需的其它部分都是可选项另外文件中用户的注释语句以分号开头

3.2 调试系统架构体系

整个调试系统包括集成调试环境和硬件仿真器它们组成了一个类似于客户服务器模型的系统服务器端把所有硬件封装起来给客户端只提供了一个虚拟的处理器接口

客户端的调试工具软件通过一个动态链接库与服务器相连所以服务器一侧的硬件和客户端的调试工具软件是相互独立的在同一个动态链接库的基础上双方可以各自自动行升级或替换同样也方便了用户选择不同的仿真器和调试软件来组合自己的调试平台

一个具体的调试程序应该能够支持许多不同的链接库不同的链接库提供了不同的功能调用以ADS工具包为例ARMulate.dll提供了一个软ARM内核用作程序的逻辑验证若要把Multi-ICE Server程序连接在一起进行实时仿真则只要调入MultiICE.dll就可以了该动态库文件在Multi-ICE Server的安装目录下可以找到

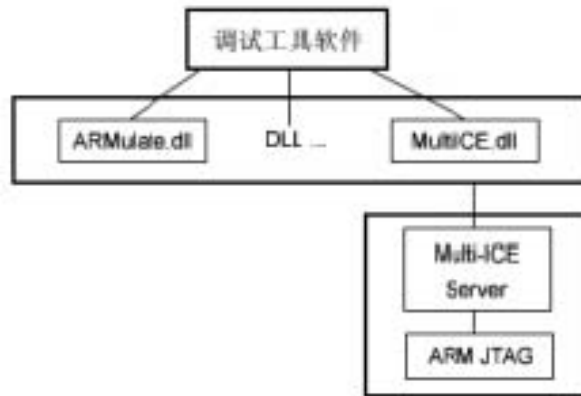


图3-16 调试系统架构

3.3 ADS V1.2与ARM JTAG连接

当Multi-ICE Server程序与目标板建立正确的连接之后下面要把调试软件连接进来组成完整的调试平台下面以ADS V1.2为例说明连接过程

3.3.1 连接AXD

在AXD的Options>Configure Target菜单中调入MultiICE.dll文件并完成目标处理器的设置只需在第一次使用之前进行设置

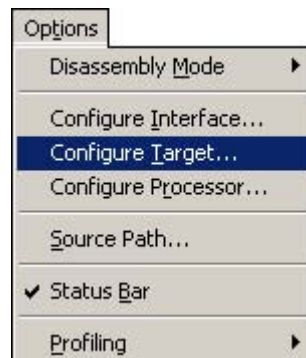


图3-17 AXD的Options菜单

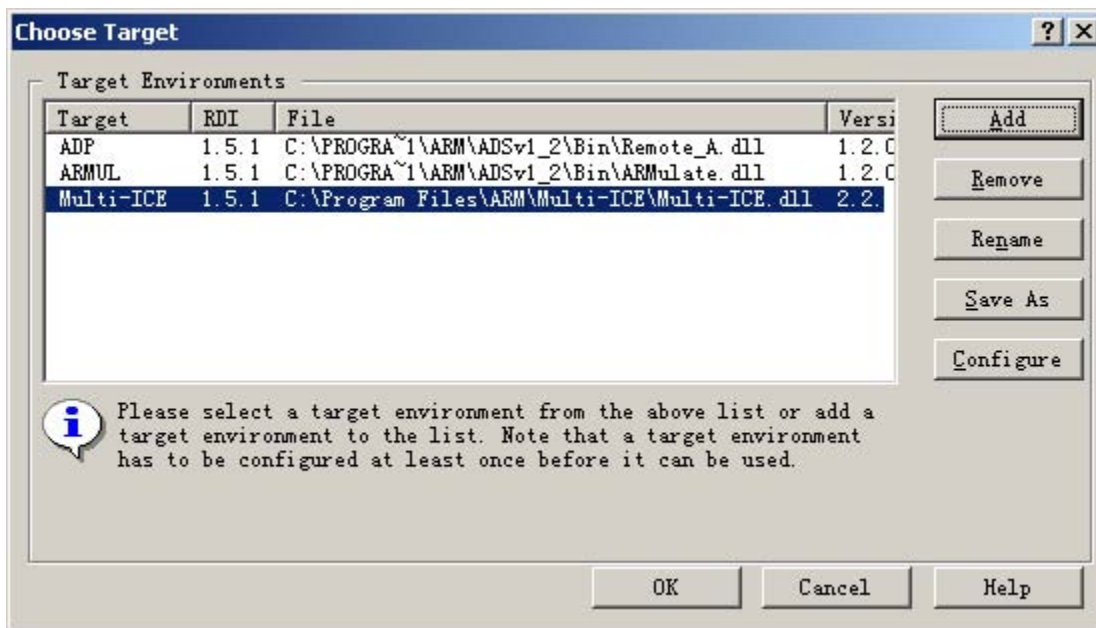


图3-18 调入MultiICE.dll

如图3-18所示如果第一次使用点击右边的Add按钮在Multi-ICE Server的安装目录下找到MultiICE.dll文件并添加到Target Environments中点击右边的Configure按钮弹出如图3-19所示的设置窗口并在Select a new processor中选择对应的处理器

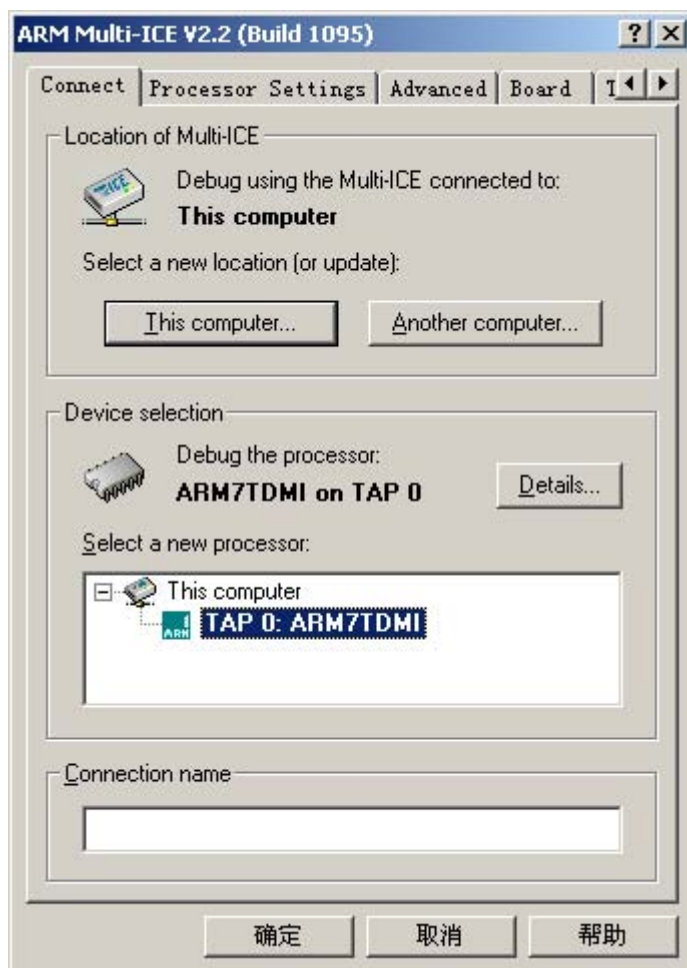


图3-19 目标处理器窗口

3.3.2 连接CodeWarrior

在CodeWarrior中打开或新建一个工程

点击Edit>DebugRel Setting菜单在CodeWarrior中进行设置

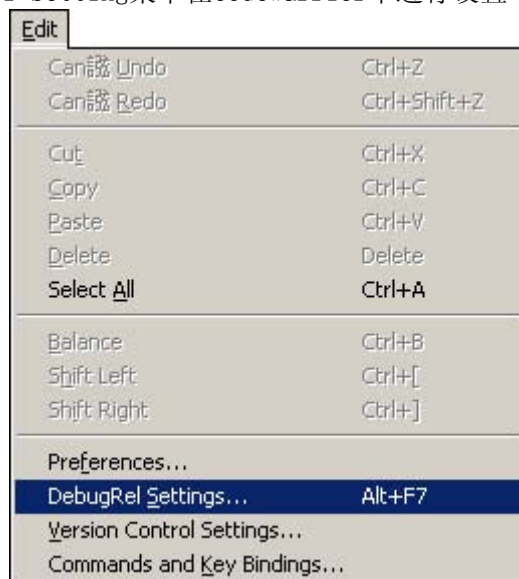


图3-20 CodeWarrior的Edit菜单

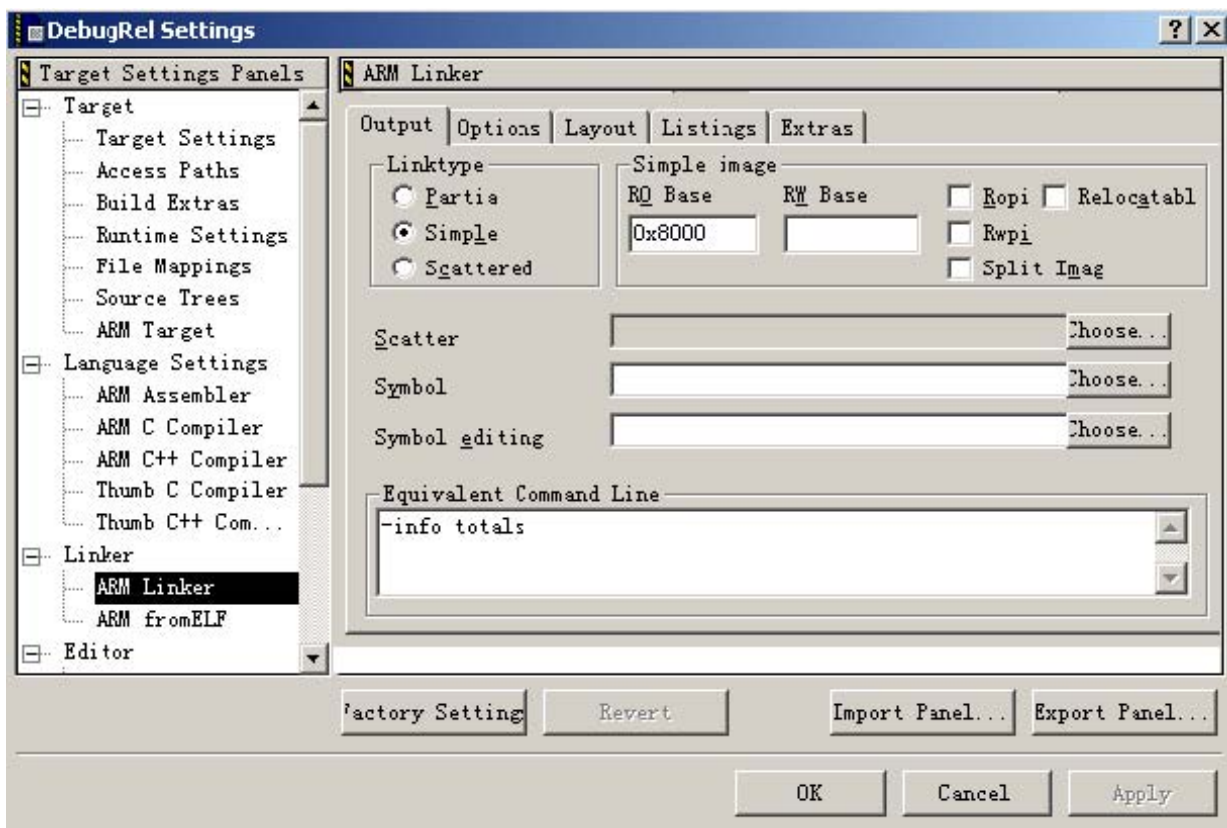


图3-21 DebugRel Settings窗口

在这里注意一下图3-21中右边的RO Base和RW Base输入框它们分别为代码区和变量区的起始地址这些设置要与具体的目标系统中使用的存储器地址相对应

第四章 最终应用程序烧写

ARM开发板文档中介绍的应用程序都是执行在板卡的SDRAM中而用户开发的最终产品是必须脱离ARM JTAG执行的本章介绍的是最终应用程序的烧写方法

本章下面按步骤讲述如何使用Flash程序向板上烧录Boot程序

4.1 应用程序在ARM板上运行原理

要想在ARM板上正确的烧录应用程序必须了解应用程序在ARM上运行的原理

首先板卡复位之后所有地址指针都被初始化为0系统指针自动指向0x0地址此时 SDRAM由于未被定义所以对于系统是可见的即用户无法对SDRAM进行访问

然后系统从0x0地址存放的用户应用程序开始执行用户程序内必须包含系统寄存器初始化代码并对外部SDRAM进行配置使系统可以进行访问此初始化代码一般用汇编语言编写

最后系统把FLASH存储器中保存的用户程序复制到SDRAM的指定地址中复制完成后程序自动跳转到SDRAM的应用程序中全速运行

4.2 生成Boot.bin

4.2.1 系统初始化

由于系统启动时从地址0x0开始执行所以程序boot必须装载在从地址0x0开始的Flash存储器中

在Boot的初始化程序Boot.s中可以看到类似下面的语句

```
ldr r0=COPY_START    pointer to ROM code
ldr r1=COPY_END       end of ROM code
ldr r2=_RAM_STARTADDRESS start of RAM
ldr PC=_RAM_STARTADDRESS jump to RAM
B .
```

这一小段程序指明了在Flash存储器中存放应用程序的起始地址结束地址以及复制到SDRAM中的起始地址当复制完成后程序转入_RAM_STARTADDRESS入口开始全速运行

在Boot程序的头文件中包含以下定义

```
_RAM_STARTADDRESS EQU 0x8000 需根据用户板SDRAM地址配置修改
BUSWIDTH      SETA 32 需根据用户实际情况修改16或32
COPY_START    EQU 0x0
COPY_END      EQU 0x5000 需根据用户程序长度修改
```

从以上定义可以看出在Flash存储器中存放的应用程序从地址0x0开始到0x5000结束复制到SDRAM的起始地址是0x8000当程序复制完成后程序转到0x8000地址开始执行

4.2.2 生成BIN文件

在生成BIN文件之前请先确认应用程序在DEBUG模式下运行正确

1 在CodeWarrior中对当前的工程Boot.mcp进行设置

进入Edit>DebugRel Settings菜单在Target Settings选项中的Post-Linker设置为ARM formELF在ARM Linker选项的Equivalent Command Line中写为-info totals -entry 0x0 -ro-base 0x0 -rw-base 0x5000

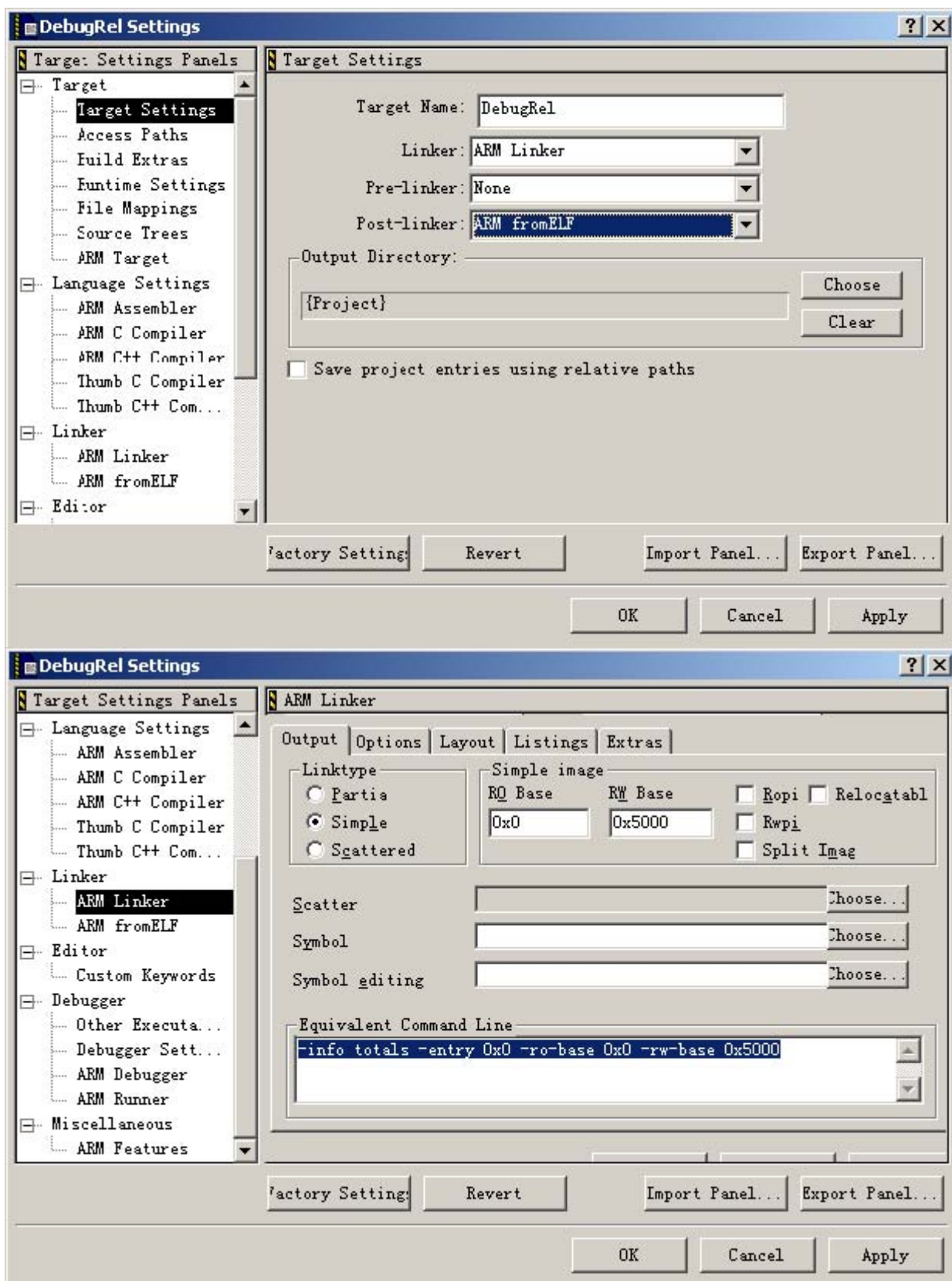


图4-1 CodeWarrior中的DebugRel Settings设置

2 对Boot.mcp进行Make生成Boot.bin文件

此时需要烧录的Boot.bin文件已经生成可以进行烧录了

4.3 烧录Boot.bin

对烧录Boot.bin文件需要用户编写一个Flash程序进行具体烧录工作在Flash程序中仍然需要对系统进行初始化的代码一般根据不同的用户硬件板不同的Flash芯片对Flash存储器的操作方法也不同因此Flash程序也有所不同相关内容请查看硬件板使用手册和Flash芯片手册

如果Flash程序调试正确就可以全速运行了如果在Flash程序内部调用了Boot.bin文件则Flash程序运行完毕后Boot.bin文件就烧录完毕了此时对硬件板进行复位Boot程序就开始在板上自动运行了

附录1 TCK频率设置

在Multi-ICE Server设置中TCK不是直接使用频率值来进行设置而是使用了一组整数它们之间的对应关系参见表附1-1用户只需要找到需要的频率把对应的设置数值填入设置窗口或是配置文件中即可

TCK时钟信号的波形不一定要方波可以高低电平设置分别不同的值但是推荐用户使用方波波形特别是不要设置占空比太大的波形

从表附1-1中可以看出频率值与对应的周期值不正好是倒数关系当高低电平的周期都是该设置值时对应的频率值才是信号频率若高低电平采用不同设置通常情况下没有必要则需要从表附1-1 中找出各自对应的周期值相加就是信号的周期再从周期算得频率

频率kHz	周期 ns	数值	频率kHz	周期 ns	数值	频率kHz	周期 ns	数值
10000	50	0	138.89	3600	81	18.38	27200	176
5000	100	1	131.58	3800	82	17.36	28800	177
3333.33	150	2	125	4000	83	16.45	30400	178
2500	200	3	119.05	4200	84	15.63	32000	179
2000	250	4	113.64	4400	85	14.88	33600	180
1666.67	300	5	108.7	4600	86	14.2	35200	181
1428.57	350	6	104.17	4800	87	13.59	36800	182
1250	400	7	100	5000	88	13.02	38400	183
1111.11	450	8	96.15	5200	89	12.5	40000	184
1000	500	9	92.59	5400	90	12.02	41600	185
909.9	550	10	89.29	5600	91	11.57	43200	186
833.33	600	11	86.21	5800	92	11.16	44800	187
769.23	650	12	83.33	6000	93	10.78	46400	188
714.29	700	13	80.65	6200	94	10.42	48000	189
666.67	750	14	78.13	6400	95	10.08	49600	190
625	800	15	73.53	6800	112	9.77	51200	191
588.24	850	16	69.44	7200	113	9.19	54400	208
555.56	900	17	65.79	7600	114	8.68	57600	209
526.32	950	18	62.5	8000	115	8.22	60800	210
500	1000	19	59.52	8400	116	7.44	67200	212
476.19	1050	20	56.82	8800	117	7.1	70400	213
454.55	1100	21	54.53	9200	118	6.79	73600	214
434.78	1150	22	52.08	9600	119	6.51	76800	215
416.67	1200	23	50	10000	120	6.25	80000	216
400	1250	24	40.08	10400	121	6.01	83200	217
384.62	1300	25	46.3	10800	122	5.79	86400	218
370.37	1350	26	44.64	11200	123	5.58	89600	219
357.14	1400	27	43.1	11600	124	5.39	92800	220
344.83	1450	28	41.67	12000	125	5.21	96000	221
333.33	1500	29	40.32	12400	126	5.04	99200	222
322.58	1550	30	39.06	12800	127	4.88	102400	223
312.5	1600	31	36.76	13600	144	4.6	108800	240

294.12	1700	48	34.72	14400	145	4.34	115200	241
277.78	1800	49	32.89	15200	146	4.11	121600	242
263.16	1900	50	31.25	16000	147	3.91	128000	243
250	2000	51	29.76	16800	148	3.72	134400	244
238.1	2100	52	28.41	17600	149	3.55	140800	245
227.27	2200	53	27.17	18400	150	3.4	147200	246
217.39	2300	54	26.04	19200	151	3.26	153600	247
208.33	2400	55	25	20000	152	3.13	160000	248
200	2500	56	24.04	20800	153	3	164400	249
192.31	2600	57	23.15	21600	154	2.89	172800	250
185.19	2700	58	22.32	22400	155	2.79	179200	251
178.57	2800	59	21.55	23200	156	2.69	185600	252
172.41	2900	60	20.83	24000	157	2.6	192000	253
166.67	3000	61	20.16	24800	158	2.52	198400	254
147.06	3400	80	19.53	25600	159	2.44	204800	255

表附1-1 TCK频率设置对应表

附录2 常见问题

随着当前系统设计复杂度的提升调试过程中碰到问题的几率大大增加在各种问题中我们需要分析问题的根源和实质以便去正确的解决ARM JTAG与目标板的连接仅限于JTAG接口两者之间的耦合度已降到最小不会对目标板的逻辑和电气性能产生影响

对于一些刚开始使用ARM JTAG的客户我们总结了一些常见问题提供给用户希望能够有所帮助

问题 **Multi-ICE Server**程序启动失败

原因 BIOS中的并口类型设置不正确请修改后重试一般推荐用户选择EPP类型

问题 **Multi-ICE Server**程序启动失败出现TCP/CP堆栈出错

原因 开发PC机没有连接网络或没有安装网络服务程序如果不需要使用网络服务功能可以在Multi-ICE Server的Settings>Start-up Options菜单中关闭该功能如果程序因为该原因自动关闭而造成无法修改选项设置可以在Multi-ICE Server的安装目录下找到一个名为的文件并双击运行则在以后的Multi-ICE Server程序运行中就会跳过网络协议堆栈的检测

问题 **Multi-ICE Server**程序自动识别目标失败

原因 1 目标内核不能识别在本手册前面章节中详细列出了目前版本的ARM JTAG所支持的内核种类如果能够检测到正确的TAP控制器并建立起通信但不能返回内核ID号将在屏幕上显示UNKNOWN这时调试程序不能正常工作用户需要从厂家得知正确的ARM芯片内核类型并新建一个配置文件来进行手动配置配置方法参见第3.1.4小节内容

2 信号问题最常遇到的信号问题包括

- ☐ Reset信号没有上拉电阻Reset包括nTRST和nSRST这两个信号应在目标板上用小于10K的电阻上拉
- ☐ TCK信号的频率太高该频率受制于TAP控制器目标板布线和连接电缆等因素有些TAP控制器只支持到1MHz速率这时候需要调整TCK频率设置

问题 同调试程序连接时提示Can't stop processor

原因 1 在当前环境下JTAG的时钟频率过高请尝试降低TCK的频率

2 如果Multi-ICE Server程序是手动配置的则需要检查

- ☐ 配置错误处理器类型连接次序等设置错误
 - ☐ JTAG问题进行手动配置时因先确保在自动时能够检测到UNKNOWN这样说明JTAG接口通信正确
- 3 ARM芯片接口的DBGEN信号被不正确的拉低如果该信号有片外接口请检查
- 4 目标处理器的时钟不正确

问题 同调试程序连接时出现Data Abort提示

原因 1 当调试程序启动时处理器将停在当前PC值指向的地方同时读取该PC值周围一部分地址的存储器值如果这些地址正好没有被使用或指向空就会出现上述提示此情况属于正常范围

2 果使用了存储器RAM或ROM出现这种问题表明在存储器访问上可能存在数据错误请检查存储器

3 当用户下载程序到目标板时下载的目标地址是在编译器中指定的如果编译器里设置的目标地址与目标板上可写存储器的物理地址不能对应也会出现上述提示因为调试程序试图往错误的物理地址写入用户程序而出错这时候需要检查并重新设置编译器

选项4JTAG的时钟速率过快

- 问题 在调试过程中经常停止有时候甚至进不了调试程序
- 原因 在排除了调试程序本身的问题后这一类带有随机性的问题可以考虑是硬件时序或逻辑问题引起的最主要的可能性是nTRST和系统Reset信号不可靠Reset信号的不可靠包含两方面正常工作时要求稳定的保持高电平复位状态时要求有足够时间的低电平Reset电路因为简单通常会被忽略但事实表明很多貌似奇怪的问题最后都归结在这里无论是从调试的角度还是最后稳定工作的要求我们都建议用户采用一个好的Reset电路或是IC来代替简单的RC电路
- 问题 调试程序出现**Hardware interface timeout**提示
- 原因
 - 1 目标板与ARM JTAG的连接断开或ARM JTAG与PC之间的连接中断
 - 2 ARM JTAG供电中断或供电不足
 - 3 ARM JTAG与目标板的通信停顿一般都是因为打开了RTCK选项但是检测不到合适的RTCK信号此时用户可以不使用RTCK功能或检查该信号的布线
- 问题 序提示**Unable to set breakpoints on exception vecors**
- 原因 射表中的起始地址对应的存储器类型是ROMARM JTAG不能在ROM上设置断点