

bdI GDB

JTAG interface for GNU Debugger

中文手册



北京麦克泰软件技术有限公司

北京麦克泰软件技术有限公司

目录

1 BDI 简介	3
1.1 bdiGDB 调试环境.....	3
1.2 BDI 支持的调试器与 CPU.....	3
1.3 bdiGDB 具体支持的 CPU	5
2 Windows 环境下使用 BDI	6
2.1 BDI 与主机的连接.....	6
2.1.1 连接串口.....	6
2.1.2 更新固件.....	6
2.1.3 TFTP Server 与 Telnet	7
2.2 BDI 与目标板的连接.....	8
3 Linux/UNIX 环境下使用 BDI.....	9
4 配置文件.....	10
4.1 [INIT].....	10
4.2 [TARGET].....	10
4.3 [HOST]	10
4.4 [FLASH].....	10
4.5 [REGS]	11
4.6 一个完整的例子	11
5 BDI2000 命令	14
6 使用开源的 GUN 调试器 Insight	16
6.1 下载安装 Insight	16
6.2 启动 Cygwin.....	16
6.3 编译程序	16
6.4 连接 BDI.....	17
6.5 启动 Insight	17

北京麦克泰软件技术有限公司

7 调试 U-Boot	20
7.1 U-Boot 移植方法	20
7.2 使用 BDI2000 调试 U-boot 注意事项	20
7.3 BDI 调试 U-BOOT 步骤	21
8 调试 Linux 内核	22
8.1 选择断点	22
8.2 调试内核步骤	22
9 连接 BDI 常出现的几个错误	23
9.1 JTAG 错误	23
9.1.1 waiting for VCC	23
9.1.2 JTAG Exits Check Failed, JTAG Bypass Check Failed	23
9.1.3 Target reset detected, restarting target	23
9.2 FLASH	23
9.2.1 Flash Programming Failed	23
9.2.2 程序不能烧入正确的地址	24
9.2.3 Can the BDI program NAND flash?	24
9.3 其他	24
9.3.1 cannot connect to BDI loader	24
9.3.2 cannot open file on host	24
9.3.4 The BDI mode LED keeps blinking	25
9.3.5 cannot load file to the memory	25
10 附录	26
10.1 附录一：BDI2000 支持的 FLASH	26

北京麦克泰软件技术有限公司

1 BDI简介

BDI2000 是一款功能强大的JTAG/BDM 仿真器，可以在Windows/Linux平台下使用，它能准确的反映用户板子的硬件信息、烧写FLASH等。BDI支持很多调试器，其中bdiGDB采用标准的GDB远程协议，支持GNU debugger (GDB)。

1.1 bdiGDB 调试环境

使用bdiGDB进行调试，BDI通过网线与主机相连，通过BDM/JTAG接口与目标板相连，如图 1.1-1所示，BDI2000的BDM/JTAG电缆电压范围是1.8V-5.0V。

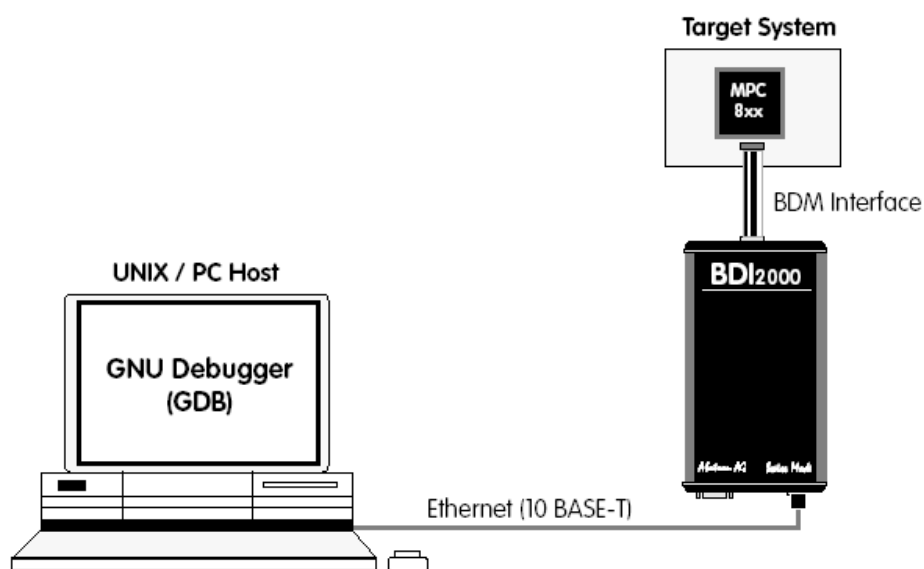


图 1.1-1 bdiGDB 与主机/目标板的连接

1.2 BDI 支持的调试器与 CPU

BDI 仿真器目前有 BDI1000 与 BDI2000 两种，其中后者支持的 CPU 更多，下载/烧写代码的速度更快。目前 BDI 支持的调试器与 CPU 的情况见下图 1.2-1。

北京麦克泰软件技术有限公司

BDI Family Overview

Last update: May 20, 2005

Debugger & Application Support		BDI1000										BDI2000													
Application	Description	CPU12	CPU16	CPU32/32+	MPC5xx8xx	PPC4xx	ARM7/9/9E	ARM11	M-CORE	ColdFire	XScale	CPU16	CPU32/32+	MPC5xx8xx	PPC4xx	PPC6xx7xx82xx83xx 1)	MPC744x, MPC745x	MPC85xx	ARM7/9/9E	ARM11	M-CORE	ColdFire	MIPS32	MIPS64	vc-1
bdiAda	Debugger support for Ada from Aonix																								
bdiCW	Debugger support for CodeWarrior from Metrowerks																								
bdiGDB	Debugger support for GNU																								
bdiHCL	Debugger support for HI-CROSS/Hi-WAVE from Metrowerks (ex. Hiware)																								
bdiRDI	Debugger support for ADW/AXD from ARM, IAR EWB, Metrowerks																								
bdiSCI	Debugger support for SeeCode from ARC International (MetaWare)																								
bdiWind	Debugger support for Tornado from Wind River Systems																								
bdiXRI	Debugger support for XRAY from Mentor Graphics																								
Stand-alone Tools:																									
bdiPro	Flash memory on-board programming software for Windows hosts																								
bdiAccess	Library to access BDM/JTAG functions from within own applications																								

1) PPC6xx/7xx/82xx/83xx, MPC7400/7410, MPC5200

Legend:

- Available
- Planned / under development
- Not available

Please contact Abatron for availability or for specific CPU support.



© 1992-2005 by Abatron AG
CH-6343 Rotkreuz / Switzerland
E-Mail: info@abatron.ch
http://www.abatron.ch

Specification subject to change without notice.
All trademarks are property of their respective holders.

图 1.2-1 BDI 支持的调试器与 CPU

1.3 bdiGDB 具体支持的 CPU

» CPU32/CPU32+

» ColdFire

» MPC5xx/8xx

» PPC6xx/7xx, MPC82xx, MPC83xx, MPC7400/7410, MGT5100, MPC5200

» MPC744x, MPC745x

» MPC85xx

» PPC4xx, Xilinx Virtex-II Pro

» ARM7/9/9E:

ARM7TDMI, ARM710T, ARM720T, ARM740T

ARM9TDMI, ARM920T, ARM922T, ARM940T, TI925T

ARM9E, ARM926E, ARM946E, ARM966E

» ARM11:

ARM1136

» XScale:

IOP310, IOP321, IOP331, IOP332

PXA210, PXA250, PXA255, PXA270

IXP420, IXP421, IXP422, IXP425, IXP460, IXP465

IXP2325, IXP2350, IXP2400, IXP2800, IXP2850

» MIPS32:

4Kx, IDT RC323xx, IDT RC324xx, AU1xxx

» MIPS64:

5KC, Wintergra

» M-CORE

北京麦克泰软件技术有限公司

2 Windows环境下使用BDI

在使用 BDI 之前，最好先检查一下板子电源正常与否，JTAG 接口是否与 BDI 手册上面的定义一样。

2.1 BDI 与主机的连接

bdi 通过串口（RS232）来更新固件（firmware）,通过网络方式进行调试。

2.1.1 连接串口

串口仅仅是在初始化（来更新固件）BDI 才会用到，BDI 支持 4 个串口：COM1—COM4。如果没有串口或者被占用（开着超级终端等）可以用 USB 转串口。连接电路以及信号定义如下图所示：

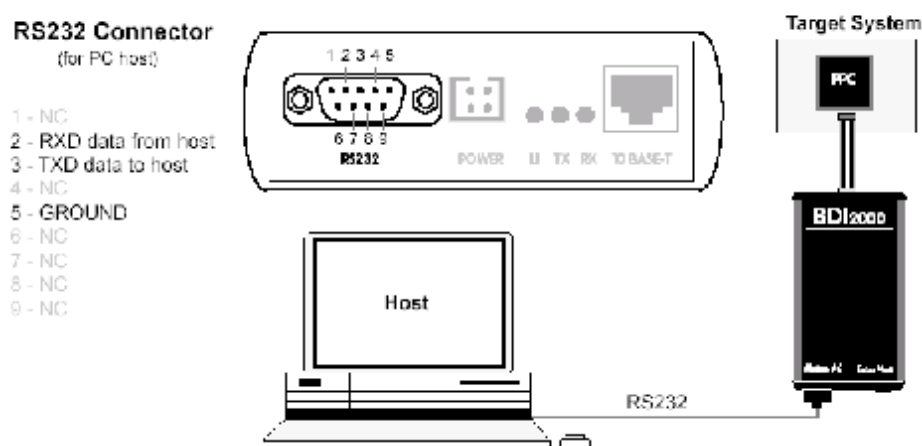


图 2.1-1

2.1.2 更新固件

并非每次使用都要更新固件（firmware），只有在初次使用或者更换平台时才需要更新。打开 B20XXXGD.EXE，选择 setup—BDI2000，出现如下画面：

选好串口与波特率后点 Connect，而后点 Update，等一会完成更新，接着设置 BDI 的 IP 以及子网掩码，以及主机 IP，最后一栏添写配置文件的名字（关于配置文件后面有详细介绍）。

注：B20XXXGD.EXE，其中 B20 指 BDI2000；XXX 是 CPU 类型，如 ARM,xscale (XSC)；GD

北京麦克泰软件技术有限公司

指 GDB 固件。

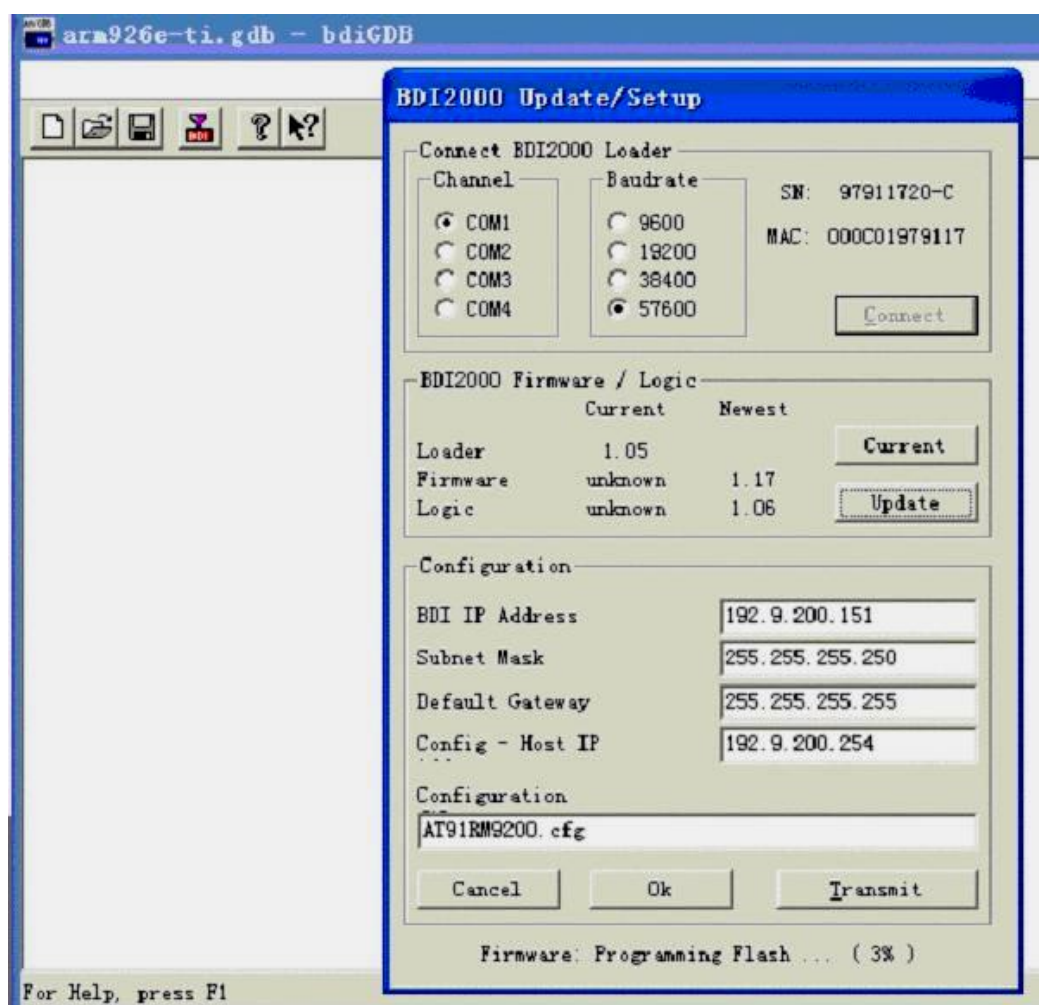


图 2.1-2 固件更新与设置

注意：在更新固件时先断开 BDI 与目标板的连接。

2.1.3 TFTP Server 与 Telnet

BDI 通过以太网与主机进行通讯，主机需要先运行 TFTP Server，而后再 Telnet 到 BDI。BDI 压缩包里面有一个 Windows 版本的 TFTP Server: tftpsrv.exe，双击运行即可。更新完固件后，接好目标板，加电，然后从主机 PC 的开始一运行里面输入 telnet: <BDI IP>。

注：必须先开 TFTP Server 再 Telnet 到 BDI。

北京麦克泰软件技术有限公司

2.2 BDI 与目标板的连接

BDI 与目标板之间通过 BDM/JTAG 电缆进行通讯，不同的平台 JTAG 不同，常见的定义如下表所示：

表 2.2-1 常用的 BDM/JTA 定义

pin 内核	ARM/XSCALE	MIPS		PPC5xx/8xx	PPC6xx/7xx/82xx/83xx/74xx/85xx
	20	14	24	10	16
1	Vcc Target	/TRST	/TRST	VFLS0	TDO
2	NC	NC	GROUND	/SRESET	/QACK
3	/TRST	TDI	TDI	GROUND	TDI
4	NC	NC	GROUND	DSCK	/TRST
5	TDI	TDO	TDO	GROUND	NC
6	NC	GROUND	GROUND	VFLS1	Vcc Target
7	TMS	TMS	TMS	/HRESET	TCK
8	GROUND	GROUND	GROUND	DSDI	RXD
9	TCK	TCK	TCK	Vcc Target	TMS
10	GROUND	NC	GROUND	DSDO	TXD
11	RTCK	/RESET	/RESET		/SRESET
12	NC	NC	/RESET		GROUND
13	TDO	DINT	NC		/HRESET
14	NC	Vcc Target	GROUND		NC
15	/RESET		NC		NC
16	NC		GROUND		GROUND
17	NC		NC		
18	NC		GROUND		
19	NC		NC		
20	NC		GROUND		
21			DBGBOOT		
22			GROUND		
23			Vcc Target		
24			GROUND		

注：Vcc Target (1.8v—5.0v)

北京麦克泰软件技术有限公司

3 Linux/UNIX环境下使用BDI

与Windows平台一样，Linux环境下使用BDI之前，先检查一下板子电源正常与否，JTAG接口是否与BDI手册上面的定义一样，在更新固件（firmware）之时断开JTAG连接。另外，在Linux下需要预先安装TFTP Server压缩包，然后开启TFTP Server。

在RedHat 8/9 下到etc/xinetd.d/tftp，设置disable = no

```
service tftp
{
    disable = no
    socket_type = dgram
    protocol = udp
    wait = yes
    user = root
    server = /usr/sbin/in.tftpd
    server_args = -s /tftpboot
    per_source = 11
    cps = 1002
    flags = IPv4
}
```

1) BDI 安装包里面有个压缩文件 bdisetup.zip，解压之。

2) 编译：

```
[local@linux bdisetup]make
cc -O2 -c -o bdisetup.o bdisetup.c
cc -O2 -c -o bdicnf.o bdicnf.c
cc -O2 -c -o bdidll.o bdidll.c
cc -s bdisetup.o bdicnf.o bdidll.o -o bdisetup
```

3) 更新固件：

```
#!/bdisetup -u -b57 -p/dev/ttyS0 -aGDB tARM
```

(-u: 更新固件；-b57: 波特率 57600；-p/dev/ttyS0: 串口 1（缺省）；tARM: ARM 平台)

4) 设置 IP 及其配置文件：

```
#!/bdisetup -c -i192.9.200.151（此处是 bdi ip） -h192.9.200.150（主机 ip） -fxxx.cfg
```

5) 完成退出：

```
#!/bdisetup -v -s
```

6) 拷贝配置文件到 tftp 根目录下：

```
#cp xxx.cfg /tftpboot
```

7) Telnet:

```
#telnet <bdi ip>
```

北京麦克泰软件技术有限公司

4 配置文件

BDI2000使用最难的一部分就是创建正确的配置文件(*.cfg)，通常cfg文件要初始化板子的RAM、FLASH、串口、网口等。

4.1 [INIT]

init顾名思义就是初始化，它是由各种命令组成，每次板子重启都要执行这些命令来初始化板子。这部分是cfg文件中最难的一部分，通常需要用户自己改动，但要正确的填写就需要对硬件非常了解。除非板子上面有正确的BOOT程序，那可以把[INIT]空着不写。不同的内核命令会有差异，一般来说它的命令格式是：命令 <地址> <数据>，例如
WM32 address value 就是把32位数据写入指定地址；

4.2 [TARGET]

这部分主要定义板子内核信息，主要的几项如下：

CPUTYPE:告诉BDI要通讯的内核。

STARTUP: 如果板子有boot loader，使用STARTUP STOP xxxx，就是让BOOT程序运行xxxx毫秒再接管CPU；如果没有boot loader，使用默认的设置STARTUP RESET。这里面还有其他选项，例如vector catch, system delays, JTAGCLOCK等，具体看BDI手册。

4.3 [HOST]

这里要告诉BDI2000从哪里load 程序及其文件格式。

注意：文件名和路径里面不要带有空格。

IP: TFTP Server所在计算机的IP。

FILE: load到SDRAM的文件，注意完整的路径。

FORMAT: 可以Load到SDRAM的文件格式：BIN, ELF ,AOUT,SREC等。

START: 16进制地址，只有BIN格式文件（不带调试信息的文件格式）需要指定LOAD的SDRAM地址。

PROMPT: BDI2000提示符，在telnet窗口下显示，例如："bdi>"。

4.4 [FLASH]

本节是设置板子上的Flash，它并非是与BDI2000通讯所必须的部分，但是如果要进行Flash

北京麦克泰软件技术有限公司

操作，烧写Flash，那么必须正确设置才能够进行。

注意：没有[INIT]里面没有正确配置，或者板子上面没有正确的boot loader，那么FLASH烧写仍是无法进行。

CHIPTYPE: Flash类型，请参考BDI相关手册。.

CHIPSIZE:板子上面一片FLASH的大小。.

BUSWIDTH: Flash bus。

ERASE:擦除，要烧写Flash需要先擦除Flash。.

FILE:想要烧写到Flash的文件。

FORMAT: 可以烧入Flash的文件格式：SREC, BIN, ELF, AOUT。

4.5 [REGS]

这里主要填写寄存器定义文件*.def,def文件需要在放在TFTP Server目录下。例如：

[REGS]

FILE reg926e.def

4.6 一个完整的例子

```
; bdiGDB configuration for Samsung SMDK2400 evaluation board
; -----
;
[INIT]
;
WM32    0x15300000  0x00008000  ; WTCN: disable watch dog
;
; Setup memory controller based on 24XMON
; WM32    0x14000000  0x22111122 ; BWSCON: AMD x16 boot flash
WM32    0x14000000  0x22111114 ; BWSCON: Intel x32 boot flash
WM32    0x14000004  0x00000700 ; BANKCON0
WM32    0x14000008  0x00000700 ; BANKCON1
WM32    0x1400000C  0x00000700 ; BANKCON2
WM32    0x14000010  0x00000700 ; BANKCON3
WM32    0x14000014  0x00000700 ; BANKCON4
WM32    0x14000018  0x00000700 ; BANKCON5
WM32    0x1400001C  0x00018005 ; BANKCON6
WM32    0x14000020  0x00018005 ; BANKCON7
WM32    0x14000024  0x008e0459 ; REFRESH
WM32    0x14000028  0x00000010 ; BANKSIZE
WM32    0x1400002C  0x00000030 ; MRSRB6
WM32    0x14000030  0x00000030 ; MRSRB7
;
```

北京麦克泰软件技术有限公司

```
[TARGET]
CPUTYPE      ARM920T
CLOCK        1                ; JTAG clock (0=Adaptive, 1=16MHz, 2=8MHz, 3=4MHz)
; STARTUP    STOP 6000        ; let the monitor setup the memory controller.
ENDIAN        LITTLE          ; memory model (LITTLE | BIG)
; WORKSPACE   0x0c000020      ; workspace in target RAM for fast download
VECTOR        CATCH 0x1f      ; catch D_Abort, P_Abort, SWI, Undef and Reset
BDIMODE        AGENT          ; the BDI working mode (LOADONLY | AGENT)
BREAKMODE     SOFT 0xDFFFDFDF ; SOFT or HARD, ARM / Thumb break code
; SIO         7 9600          ; TCP port for serial IO
; DCC         7              ; TCP port for DCC I/O

[HOST]
IP            151.120.25.119
FILE          E:\cygwin\home\bdidemo\arm\vxworks
FORMAT        COFF 0x0c000000
LOAD          MANUAL          ;load code MANUAL or AUTO after reset

[FLASH]
; ===== AMD x 16 Boot Flash =====
WORKSPACE     0x0c000000      ; workspace in target RAM for fast programming algorithm
CHIPTYPE       AM29BX16       ; Flash type (AM29F | AM29BX8 | AM29BX16 | I28BX8 | I28BX16)
CHIPSIZ        0x100000       ; The size of one flash chip in bytes
BUSWID        16              ; The width of the flash memory bus in bits (8 | 16 | 32)
FILE           E:\cygwin\home\bdidemo\arm\s2400mon.bin ;The file to program
FILE           E:\cygwin\home\bdidemo\arm\vxWorks   ;The file to program
FORMAT         BIN 0x00010000
; ERASE        0x00000000      ; erase sector 0 of flash in U11 (AM29LV800BB)
; ERASE        0x00004000      ; erase sector 1
; ERASE        0x00006000      ; erase sector 2
; ERASE        0x00008000      ; erase sector 3
ERASE          0x00010000      ; erase sector 4
ERASE          0x00020000      ; erase sector 5
ERASE          0x00030000      ; erase sector 6
ERASE          0x00040000      ; erase sector 7
ERASE          0x00050000      ; erase sector 8
ERASE          0x00060000      ; erase sector 9
ERASE          0x00070000      ; erase sector 10
ERASE          0x00080000      ; erase sector 11
ERASE          0x00090000      ; erase sector 12
ERASE          0x000A0000      ; erase sector 13
ERASE          0x000B0000      ; erase sector 14
ERASE          0x000C0000      ; erase sector 15
ERASE          0x000D0000      ; erase sector 16
ERASE          0x000E0000      ; erase sector 17
ERASE          0x000F0000      ; erase sector 18

; ===== Intel x 32 Strata Flash =====
```

北京麦克泰软件技术有限公司

```
; WORKSPACE    0x0c000000    ; workspace in target RAM for fast programming algorithm
; CHIPTYPE      STRATAX16     ; Flash type (AM29F | AM29BX8 | AM29BX16 | I28BX8 |
I28BX16)
; CHIPSIZE      0x800000      ; The size of one flash chip in bytes
; BUSWIDTH      32            ; The width of the flash memory bus in bits (8 | 16 | 32)
; FILE          E:\cygwin\home\bdidemo\arm\s24wince.bin    ; The file to program
; FILE          E:\cygwin\home\bdidemo\arm\vxWorks          ; The file to program
; FORMAT        BIN 0x00040000
; ERASE         0x00000000    ; erase block 0 of flash in U10/U12 (i28F640J3A)
; ERASE         0x00040000    ; erase block 1
; ERASE         0x00080000    ; erase block 2
; ERASE         0x000C0000    ; erase block 3
; ERASE         0x00100000    ; erase block 4
; ERASE         0x00140000    ; erase block 5
; ERASE         0x00180000    ; erase block 6
; ERASE         0x001C0000    ; erase block 7
```

[REGS]

FILE regS2400.def

北京麦克泰软件技术有限公司

5 BDI2000命令

使用BDI提供的命令可以检查memory状态、显示/修改寄存器(通用、特殊)数值,烧些FLASH等等。下面列出常用的命令及其用法,如果要查看全部的命令在telnet下输入help即可。

MD [<address>] [<count>] 显示存储器数据(32bit), 常用命令, 检查存储器。

例如显示0x10000000地址数据: md 0x10000000

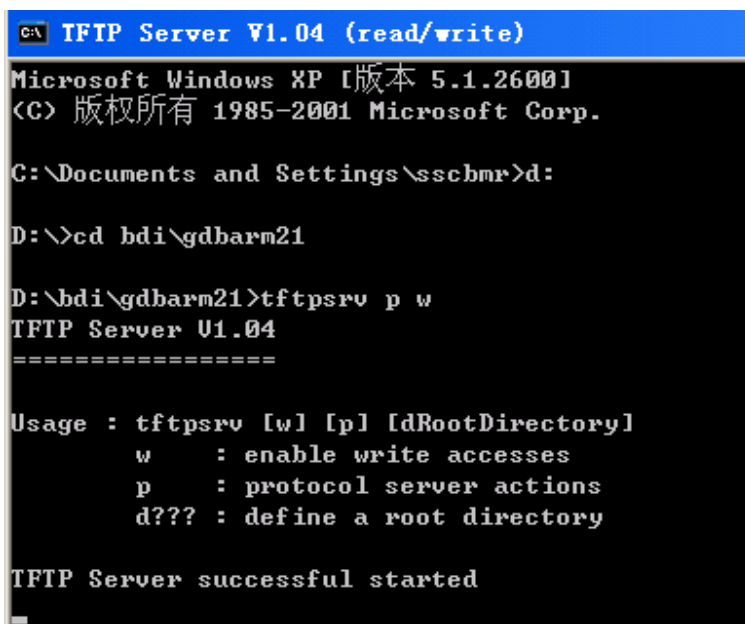
MDH [<address>] [<count>] 显示存储器数据(16bit)。

MDB [<address>] [<count>] 显示存储器数据(8bit)。

DUMP <addr> <size> [<file>] 把某段存储器数据保存到某个文件, 例如在FLASH操作中想保留原来的数据, 就可以利用此命令先把Flash里面的数据存保存起来。

例如把FLASH里面0x1000000开始的1M代码保存到boot.bin 文件: DUMP 0x1000000 0x100000 boot.bin

注: 使用DUMP命令需要TFTP Server有读写权限。在Dos下输入 tftpsrv p w



```
C:\> TFTP Server V1.04 (read/write)
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Documents and Settings\sschmr>d:

D:\>cd bdi\gdbarm21

D:\bdi\gdbarm21>tftpsrv p w
TFTP Server V1.04
=====

Usage : tftpsrv [w] [p] [dRootDirectory]
        w      : enable write accesses
        p      : protocol server actions
        d???   : define a root directory

TFTP Server successful started
```

MM <addr> <value> [<cnt>] 修改内存数据(32bit), 一般用来检查RAM运行与否。

例如 检查0x100000处RAM是否正常工作或者是不是FLASH:

mm 0x100000 0xaa

md 0x100000

如果数据修改成功, RAM正常工作; 否则RAM初始化有问题或者是FLASH地址(如果对memory地址不是很清楚的话)。

RD [<name>] 显示通用寄存器或者用户定义的寄存器数值

RM {<nbr>#<name>} <value> 修改通用寄存器或者用户定义的寄存器数值

注: 特殊功能寄存器的显示以及修改命令不同的内核会不一样, 请在telnet下输入help查看。

北京麦克泰软件技术有限公司

BOOT 修改cfg文件后用boot命令重新引导。

RESET [HALT | RUN [time]] 重启

GO [<pc>] 指定PC地址，然后从那里运行程序。

TI [<pc>] 单步运行程序

HALT [<n>[<n>[<n>[<n>]]]] 进入调试模式。

BI <addr> [<mask>] 设置硬件断点。

CI [<id>] 清除硬件断点。

BD [R|W] <addr> [<data>] 设置数据断点(32bit access)",

CD [<id>] 清除数据断点。

INFO 显示当前CPU状态以及当前PC和几个寄存器。

RAM操作命令：

LOAD [<offset>] [<file> [<format>]] load代码到RAM。

例，把load test (ELF格式)：load D:\bdi\gdbarm21\test ELF

FLASH操作命令：

ERASE [<address> [<mode>]] FLASH擦除，要烧些FLASH需要先用此命令来擦除。

<mode> : CHIP, BLOCK or SECTOR (default is sector)。

ERASE <addr> <step> <count> 擦除多块。

例如，擦除0x100000开始的8块1M大小的数据：erase 0x100000 0x100000 8

PROG [<offset>] [<file> [<format>]] 烧写FLASH。格式：SREC, BIN, AOUT, ELF or COFF",

例如，把test (bin格式) 文件烧到0x100000处：prog 0x100000 test BIN

注：程序烧写完毕后可以用md来检查：md 0x100000

HOST <ip> 主机IP操作命令。

显示主机IP：host

制定主机IP为192.168.0.6：host 192.168.0.6

CONFIG 显示/更新BDI配置

CONFIG <file> [<hostIP> [<bdiIP> [<gateway> [<mask>]]]] 修改配置文件

HELP 帮助,

QUIT 退出

北京麦克泰软件技术有限公司

6 使用开源的GUN调试器Insight

BDI 自己不带调试器，虽然在 Telnet 下通过命令方式可以完成相关的调试工作，但那样略显麻烦一点。不过 BDI 可以与许多种调试器连接，其中 Insight 是一种开源的 GNU 调试软件，用户可以自己来编译以使其更加适合自己的开发平台。

必须声明的是，BDI 自己不带调试器，所以我们可以协助用户拿 BDI 连接 Insight 或其他调试软件，但是如何使用调试软件并不在我们支持之列。

6.1 下载安装 Insight

Insight 可以从 http://www.macroigor.com/full_gnu.htm 下载。其中 Windows 系统要先下载安装 CYGWIN™ (2.0.0) (Windows 下模拟 linux 环境) 与 gnutools-xxx-elf-x.xx (先装 Cygwin)。

Linux 环境下载相应的 rpm 安装包安装即可。

6.2 启动 Cygwin



图 6.2-1 启动 Cygwin

6.3 编译程序

在 Cygwin 提示符下用 `cd` 命令转到要编译的程序目录下（图 2.2，此处以 MPC8XX 为例，注意区分大小写）然后输入命令：`make`（图 2.3）。

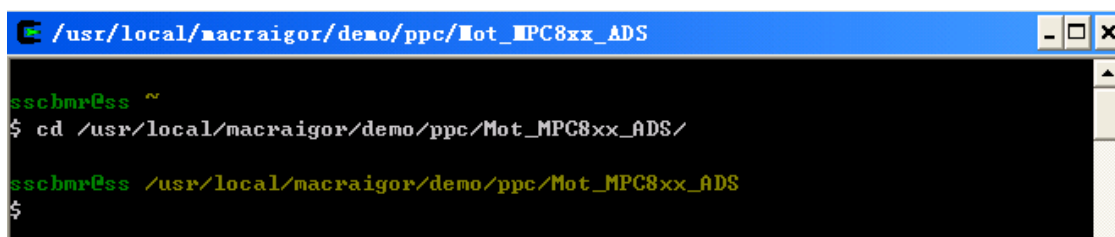


图 6.3-1 在 Cygwin 提示符下用 `cd` 命令

北京麦克泰软件技术有限公司

```
$ make
powerpc-elf-gcc -mbig-endian -g -c -gstabs+ -Wa,-alh=crt0.lst,-L -o crt0.o crt0.
$
powerpc-elf-gcc -mbig-endian -g -c -gstabs+ test.c
test.c:46: warning: conflicting types for built-in function 'exit'
powerpc-elf-ld -EB -N -g -v -Tldscript -o test crt0.o test.o
GNU ld version 2.15
cp gdbinit .gdbinit
powerpc-elf-nm test
ff002004 t .Laddr
ff002000 t .Lptr
ff00202c t .Lstk
0000842c G __SDA2_BASE_
ff00a1bc G __SDA_BASE_
ff0021e8 A __bss_end
ff0021d8 A __bss_start
ff00216c T __gccmain
ff0021f8 A __sbss_end
ff0021e8 A __sbss_start
ff002208 A __stack
ff002208 A __end
ff002000 T __start
ff00219c T atexit
ff00218c T exit
ff0020f8 T inc_var
ff002030 T main
ff0021c4 G svar
ff0021c6 G svar1
ff0021bc G varc
ff0021c0 G vard

sschmr@ess /usr/local/macraigor/demo/ppc/Mot_MPC8xx_ADS
$
```

图

6.3-2 在 Cygwin 下用 make 命令编译程序

6.4 连接 BDI

连接好 BDI 与目标板, 把上面编译的 test(elf 格式) 文件拷贝到 tftp Server 目录下, 然后在 Telnet 下 laod 到 RAM 中。

6.5 启动 Insight

编译通过后, 输入 powerpc-elf-insight, 弹出一对话框, 选择 Y。



北京麦克泰软件技术有限公司

图 2.4 启动 insight

从 File->Open, 选择 test 打开文件后, 设置一个断点 (如在第 23 行), 调试器显示如下:

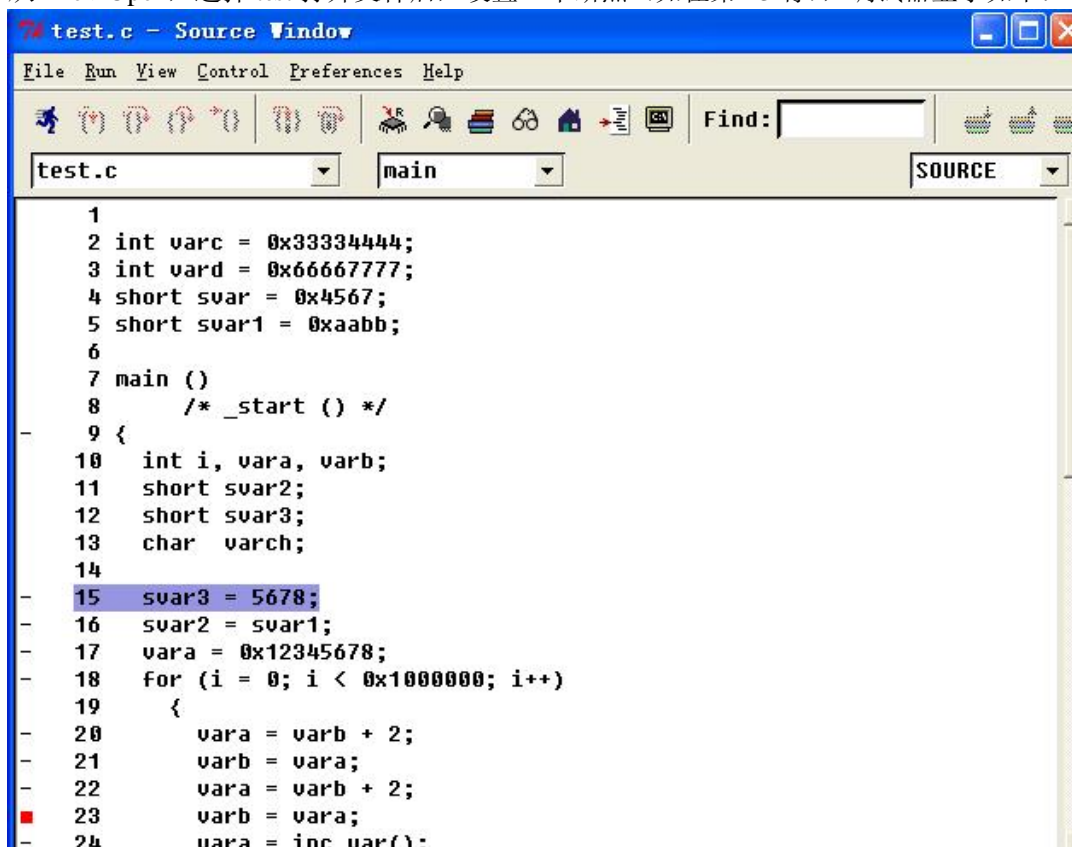


图 2.5 insight 环境下调试程序

按小人, 将弹出 Target selection 窗口。Target: 选择 Remote/TCP, Hostname: 192.9.200.145, Port: 2001, 按 OK, 即可。

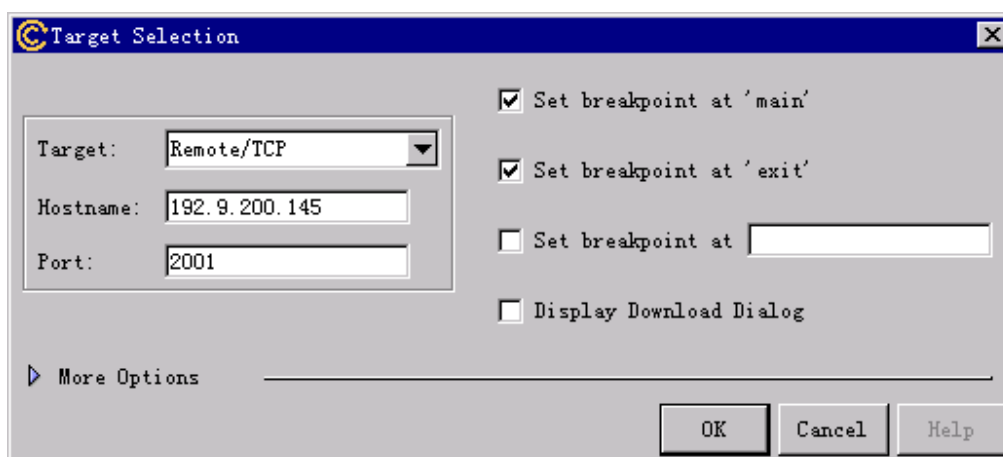


图 2.5 Target selection

在调试器窗口按 Continue, 系统就停在 23 行上了, 这时我们可以继续单步调试, 看变量, 看寄存器, 看内存等。

北京麦克泰软件技术有限公司

按 View->Memory, 显示内存的内容。

按 View->Registers, 显示寄存器的内容。

按 View->Local Variables, 显示局部变量的内容。

按 View->Stack, 显示堆栈的内容。

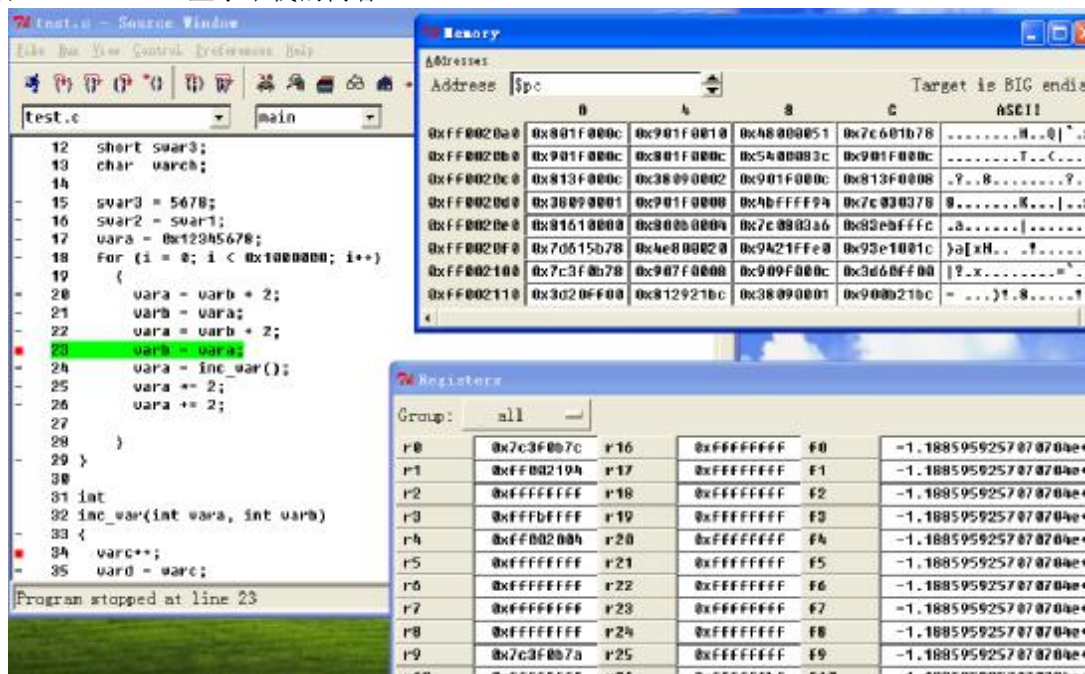


图 2.6 程序执行及观看相关变量

按 View->Watch Expressions, 显示表达式的内容。

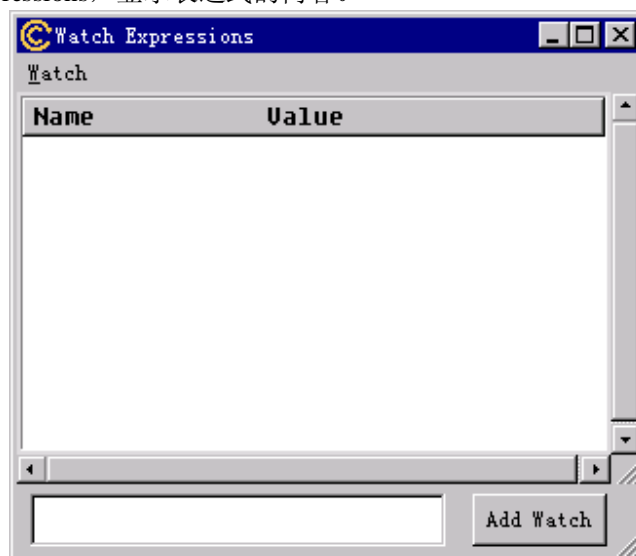


图 2.7 显示表达式的内容

北京麦克泰软件技术有限公司

7 调试U-Boot

U-Boot, 全称 Universal Boot Loader, 是在 PPC-Boot 的基础上进化而来的一个开放源码的嵌入式 BootROM 程序, 是用于初始化目标板硬件, 给嵌入式操作系统提供板上硬件资源信息, 并进一步装载、引导嵌入式操作系统运行。

7.1 U-Boot 移植方法

当前, 对于 U-Boot 的移植方法, 大致分为两种。一种是用 BDI2000 创建目标板初始运行环境, 将 U-Boot 镜像文件 u-boot.bin 下载到目标板 RAM 中的指定位置, 然后, 用 BDI2000 进行跟踪调试。其好处是不用将 U-Boot 镜像文件烧写到 FLASH 中去。但弊端在于对移植开发人员的移植调试技能要求较高, BDI2000 的配置文件较为复杂。另外一种方法是用 BDI2000 先将 U-Boot 镜像文件烧写到 FLASH 中去, 然后利用 GDB 和 BDI2000 进行调试。这种方法所用 BDI2000 的配置文件较为简单, 调试过程与 U-Boot 移植后运行过程相吻合, 即 U-Boot 先从 FLASH 中运行, 再重载至 RAM 中相应位置, 并从那里正式投入运行。唯一感到有些麻烦的就是需要不断烧写 FLASH。但考虑到 FLASH 常规擦写次数基本为 10 万次左右, 作为移植 U-Boot, 不会占用太多的次数, 建议采用第二种移植方法。

7.2 使用 BDI2000 调试 U-boot 注意事项

在建立基于 BDI2000 的源码级 BootROM 调试环境之前, 先要明确主机-BDI-目标板之间的逻辑关系, 主机端是一个 GDB 的前端环境, BDI2000 则是主机和目标板之间连接的中介, 即所谓的 GDBSERVER, 它负责把主机端发出的 gdb 命令传送到目标板上运行的程序。可以用 Insight 或者 DDD 来调试 U-BOOT, 对于 BDI 来说, 调试 U-BOOT 与调试其他程序没有多大的区别。

但是需要注意的是:

- 1) U-BOOT 比较复杂, 需要调试人员对其比较清楚, 包括对硬件有较深的掌握。
- 2) BDI 要调试的代码需要有调试信息, 例如 ELF 文件格式这样的, 因为 BDI 需要有符号表, 所以在编译的时候注意文件格式。当然最后调试通过, 再编译 BIN 格式的烧入 FLASH 即可。
- 3) 如果采用先把 U-BOOT 烧到 FLASH 的方式, BDI 配置文件[TARGET]一项里面需要把调试模式设为 hard,

BREAKMODE HARD ; SOFT or HARD, HARD uses PPC hardware breakpoint.

另外, hard 断点有限, 不同的 CPU 硬件断点有差异, 请察看 BDI 手册。

北京麦克泰软件技术有限公司

7.3 BDI 调试 U-BOOT 步骤

- 1) 连接 BDI 与板子, 先给 BDI 上电后给板子上电
- 2) 启动 TFTP Server
- 3) Telnet <BDI IP>
- 4) 查看 BDI 当前状态, 输入命令: INFO
- 5) Load U-BOOT 到 RAM(已经烧入 FLASH 则省略此步): <Core#0> load u-boot ELF
- 6) 可以用 BI 设置断点, CI 清除断点
- 7) 启动 Insight
- 8) 在 Insight 中打开 u-boot
- 9) 连接 BD
- 10) 设置断点, 调试

在实际的 U-Boot 移植中无法避免地会遇到一些难以预料的问题, 甚至出现倒退, 尤其 U-Boot 移植新手, 更会平添诸多难度。在逐步熟悉 U-Boot 移植方法、过程中, 随着自身经验的不断积累, 坚冰终会消融。

北京麦克泰软件技术有限公司

8 调试Linux内核

要调试内核需要在配置文件[TARGET]里面设置：

MMU XLAT ; translate effective to physical address;

STARTUP RUN

第一条命令是MMU内存管理，后面一条命令是让U-BOOT初始化板子（所以在配置文件里面无需初始化存储器或者寄存器—让U-BOOT做就行了）。

准备两个vmlinux：一个编译时不带“-g”选项的vmlinux；另外一个编译时带“-g”选项的vmlinux。

8.1 选择断点

Linux内核启动后MMU就会工作，这样要设置的断点就要设在MMU的范围之内，例如可以选择start_kernel作为硬件断点，在内核停在start_kernel后再连接GDB。可以通过查看Linux目录里面的系统符号表得到start_kerne的地址（假设在0xC0061550）。

```
BDI>bi 0xc0061550 /* set breakpoint at vmlinux */
BDI>go
..... /* target stops at start_kernel */
BDI>ci /*clear breakpoint*/
```

8.2 调试内核步骤

- 1) 连接 BDI 与板子，先给 BDI 上电后给板子上电
- 2) 启动 TFTP Server
- 3) BDI>Telnet <BDI IP>
- 4) BDI> INFO ; 查看BDI当前状态
- 5) start_kernel; 运行，可以通过超级终端查看串口输出信息
- 6) 如果通过U-BOOT装载内核，则选择编译时不带“-g”选项的vmlinux
- 7) 输入命令：halt停止程序运行，使用bi命令在start_kernel处设置断点
- 8) 如果不是通过U-BOOT装载内核，那么现在可以使用load命令装载；否则跳过此步
- 9) BDI>go ; 运行程序
- 10) BDI>CI; 清除断点
- 11) BDI>TI; 单步执行，查看是否能执行
- 12) 启动GDB（例如Insight），打开编译时带“-g”选项的vmlinux，连接BDI（target remote）
- 13) 设置断点、调试

北京麦克泰软件技术有限公司

9 连接BDI常出现的几个错误

9.1 JTAG 错误

并非所有开发板上面的JTAG 都是标准的，所以在使用BDI连接前最好先看一下JTAG 的定义是否一样。

9.1.1 waiting for VCC

- 1) 板子没有上电。
- 2) JTAG 数据线的JTAG VCC引脚断路或者电压过低（BDI要求1.8v-5.0v）。

上面两种情况BDI中间的JTAG电源灯都不会亮。

9.1.2 JTAG Exits Check Failed, JTAG Bypass Check Failed

```
- CONFIG: loading configuration file passed
> - TARGET: processing reset request
> - TARGET: BDI asserts TRST and RESET
> - TARGET: BDI removes TRST
> - TARGET: Bypass check 0x000000001 => 0x00FFFFFF
> - TARGET: JTAG exits check failed
```

可能的原因有：

- 1) cfg文件里面正确的CPUTYPE 不正确，不同的CPU 旁路寄存器（bypass register）不一样，处理器类型不对BDI可能找错地方。
- 2) 多核，板子上面不止一个JTAG 接口，设置SCANPRED 和SCANSUCC。
- 3) 外接电源不稳定，或者超出BDI要求的范围。

9.1.3 Target reset detected, restarting target

可能的原因：

- 1) JTAG 的TRST引脚电压过低或者接地。.

9.2 FLASH

9.2.1 Flash Programming Failed

FLASH 不能烧写最可能的原因是flash 相关设置不正确。因为BDI 利用板上处理器来进行flash

北京麦克泰软件技术有限公司

操作, 因此需要memory controller 和flash能够正常工作, 也就是说需要正确地初始化板子。另外, 除非手册特别注明, 否则去掉cfg 文件里面[FLASH] 一节的WORKSPACE一行。

注意: 使用"load"命令来写memory , "prog"命令烧写flash.

- 1) 查看配置文件里面[FLASH] 选项里面CHIPTYPE, CHIPSIZE 和BUSWIDTH. 设置是否正确。
- 2) 查看配置文件里面[INIT]memory controller 初始化正确否, flash 定义是否和板子情况一致。
- 3) 大多数flash需要在烧写之前先要擦除flash。
- 4) 编写一个测试小程序TEST.bin, 指定FLASH 地址烧入, 如果通过而用户自己要烧写的程序却出现错误, 那么需要检查要烧入的程序。Srec和ELF文件格式带调试信息, 内部有偏移量, 他们在程序编译的时候就需要指定到正确的FLASH 地址。
- 5) 如果对flash操作时好时坏, 可能的原因是需要在cfg 文件[INIT]一项里面增加等待时间。

9.2.2 程序不能烧入正确的地址

首先检查flash/memory是否工作正常, 例如用MD 命令查看能否正确显示, 用MM 命令查看能否修改指定地址数据。

```
MM 0xff000000 0x12345678
```

```
MD 0xff000000
```

如果flash能够被烧写, RAM 也可以被写入, 那么就是烧写文件的问题。

BDI总是按照用户指定的地址去烧写FLASH, 因此如果BDI不能写入, 它将返回错误信息。.

9.2.3 Can the BDI program NAND flash?

BDI只支持NOR flash., 具体情况请查看BDI支持FLASH 类型的最新手册。

9.3 其他

9.3.1 cannot connect to BDI loader

- 1) 查看串口电缆是否松动, COM端口是否正确。
- 2) BDI所用的串口没有被其他工具占用, 例如开了超级终端等等。
- 3) 有时候JTAG电缆或者网线接入BDI也会干扰串口工作。

9.3.2 cannot open file on host

当BDI读取TFTP server 路径下的文件而TFTP server 通讯有问题或者文件不存在, 那么就会出现这个错误。

北京麦克泰软件技术有限公司

检查host IP是否正确或者TFTP server 路径下的文件是否存在。

9.3.3 cannot ping the BDI on our network

- 1) 断开BDI后再ping 那个IP地址，查看有没有其他设备使用。
- 2) 断开一台能够正常使用的计算机网线，把BDI 的IP 改成那台电脑的，看看能否正常。注意IP地址的子网掩码要正确配置。
- 3) 网线是否有问题？

9.3.4 The BDI mode LED keeps blinking

当BDI 的状态灯(红色) 一秒钟闪一次，说明你的BDI2000 里面没有固件. 如果它一秒钟闪三次，说明初始化的跳线(手册第16 页) 在初始化位置。一直发亮表明有故障。另外在更新固件的时候它也会不停闪烁直至结束。

9.3.5 cannot load file to the memory

当在配置文件的[TARGET]里面没有设置WORKSPACE 一项，当load 程序到memory 时出现这个错误，表明DRAM controller 没有正确的初始化。

北京麦克泰软件技术有限公司

10 附录

10.1 附录一：BDI2000 支持的 FLASH

详细列表可以去 <http://www.abatron.ch> 下载或者与我公司联系。