

Ingenic[®] JDI用户手册

User Manual

Revision: 1.3

Date: Jan. 2007



北京君正集成电路有限公司
Ingenic Semiconductor Co. Ltd

JDI 用户手册

Copyright © Ingenic Semiconductor Co. Ltd 2006. All rights reserved.

Release history

Date	Revision	Change
Jan. 2007	1.3	- Added new commands: nquery, nreadraw, nreadoob, memtest, gpios, gpioc. - Modified some commands usage: nread, nprog. - Added comments to scale PLL etc.
Aug. 2006	1.2	- Added new commands (showcfg/reset). - Removed command (config). - Modified command (debug).
July 2006	1.1	- Added description for new commands.
Apr. 2006	1.0	- First release.

Disclaimer

This documentation is provided for use with Ingenic products. No license to Ingenic property rights is granted. Ingenic assumes no liability, provides no warranty either expressed or implied relating to the usage, or intellectual property right infringement except as provided for by Ingenic Terms and Conditions of Sale.

Ingenic products are not designed for and should not be used in any medical or life sustaining or supporting equipment.

All information in this document should be treated as preliminary. Ingenic may make changes to this document without notice. Anyone relying on this documentation should contact Ingenic for the current documentation and errata.

北京君正集成电路有限公司

北京市海淀区上地东路 1 号

盈创动力 E 座 801C

Tel: 86-10-58851003

Fax: 86-10-58851005

Http: //www.ingenic.cn

目录

1	概述	1
2	开始使用	3
2.1	使用前的准备	3
2.1.1	使用前的准备	3
2.1.2	启动TFTP服务	3
2.1.3	准备配置文件	4
2.1.4	配置文件的[INIT]段	5
2.1.5	配置JDI的操作环境	6
2.1.6	进入JDI的操作界面	6
2.1.7	退出JDI	7
2.2	EEPROM的烧录	7
2.2.1	修改配置文件	7
2.2.2	烧录EEPROM	7
2.3	NOR FLASH的烧录	8
2.3.1	修改配置文件	8
2.3.2	烧录NOR FLASH	9
2.3.2.1	读NOR FLASH中的数据	9
2.3.2.2	擦除NOR FLASH	9
2.3.2.3	烧录NOR FLASH	10
2.4	NAND FLASH的烧录	10
2.4.1	修改配置文件	10
2.4.2	烧录NAND FLASH	11
2.4.2.1	读NAND FLASH中的数据	11
2.4.2.2	擦除NAND FLASH	11
2.4.2.3	烧录NAND FLASH	12
2.5	用GDB调试程序	12
3	使用JDI的命令界面	15
3.1	进入及退出JDI的命令界面	15
3.1.1	通过串口来进入及退出JDI的命令界面	15
3.1.2	通过远程登录来进入及退出JDI的命令界面	15
3.2	配置及查询命令	16
3.2.1	帮助命令help	16
3.2.2	配置命令ipconfig	16
3.2.3	配置命令fconfig	17
3.2.4	配置命令netmask	17
3.2.5	配置命令gateway	17
3.2.6	配置命令bcast	17
3.2.7	显示配置命令showcfg	18

3.3	调试命令debug	18
3.4	FLASH及EEPROM操作的命令	18
3.4.1	EEPROM读取操作命令eread	19
3.4.2	将字符串写入EEPROM的命令eprogv	19
3.4.3	将文件写入EEPROM的命令eprog	20
3.4.4	NOR FLASH查询命令query	20
3.4.5	NOR FLASH及其sector查询命令querya	21
3.4.6	NOR FLASH的读取命令readf	21
3.4.7	NOR FLASH的擦除命令erase	22
3.4.8	NOR FLASH的烧录命令prog	22
3.4.9	NAND FLASH的查询命令nquery	23
3.4.10	NAND FLASH的读取命令nread	23
3.4.11	NAND FLASH的读取命令nreadraw	24
3.4.12	NAND FLASH的读取命令nreadoob	24
3.4.13	NAND FLASH的擦除命令nerase	25
3.4.14	NAND FLASH的烧录命令nprog	25
3.5	测试、升级及退出命令	26
3.5.1	测试命令readids	26
3.5.2	测试命令cputest	26
3.5.3	测试命令memtest	27
3.5.4	测试命令gpios和gpioc	27
3.5.5	测试命令hwtest1 和hwtest2	27
3.5.6	复位命令reset	27
3.5.7	查看Firmware版本命令version	27
3.5.8	FirmWare升级命令upgrade	28
3.5.9	批命令运行命令run	28
3.5.10	退出telnet界面的命令exit	29
4	使用及修改配置文件	31
4.1	[INIT]段的定义	32
4.1.1	写CP0 寄存器的命令WCPO	32
4.1.2	对某一地址进行写操作的命令	32
4.2	[INIT]段的定义	33
4.3	[FLASH]段的定义	33
4.3.1	AUTO算法类型的[FLASH]段定义	33
4.3.2	I28F算法类型的[FLASH]段定义	34
4.3.3	AM29 算法类型的[FLASH]段定义	35
4.4	[NAND]段的定义	36
4.5	[EEPROM]段的定义	37
4.6	[SDRAM]段的定义	37
4.7	[GDB]段的定义	37
5	JDI的硬件说明	39

5.1	JDI的端口说明	39
5.2	JDI上的指示灯	39
5.3	JDI的跳线保护	39
5.4	JDI主要端口的信号定义	40

1 概述

JDI 是君正集成电路有限公司专门为各种基于君正 CPU 的开发板和产品板设计的一个开发和调试工具。它本着使用简单，操作方便，配置灵活，使用范围广的设计原则，为用户的开发和调试提供更好的支持。JDI 主要用来提供目标板上 Flash 的烧录，EEPROM 的烧录，目标板硬件和设备驱动程序调试，及操作系统级的最低层调试。

JDI 具有以下的特点：

- 基于 Ethernet 来传输数据，提高了数据的吞吐量
- 使用主机的 USB 来提供电源，增加了使用的方便性
- 使用 RS-232C 接口来提供配置界面
- JDI 内部嵌入 Linux™操作系统，提高了稳定性
- 通过 Ethernet 提供 Telnet 的用户界面，方便用户的操作
- 通过 TFTP 来访问开发板和产品板的配置文件，增加了方便性和灵活性
- 通过 EJTAG 和目标板连接，可以提供最低层的调试支持

JDI 的调试功能主要用于操作系统级的低层调试，它并不适合调试操作系统以上的应用程序。对于应用程序的调试，可以通过操作系统提供的功能来进行（如使用 gdbserver 等）。

通过更换 JDI 中的 FirmWare，也可以将 JDI 用于其它体系的目标板。

2 开始使用

JDI 的主要功能有 FLASH 的烧录, EEPROM 的烧录及程序的调试, 本章将对它的这些功能做个简单的展示, 其详细的操作需要参阅后继的章节。

使用 JDI 时, 需要有一台可以提供 TFTP 服务的主机 Host, 在本章中我们使用一台 Linux 服务器作为 Host, 其 IP 地址为 192.168.1.20。

当 JDI 启动时, 会通过 TFTP 协议到提供 TFTP 服务的主机上去访问关于目标板的配置文件。

本章中所使用的开发板上的 NOR FLASH 为 Intel 公司的 E28F128(J3A150), NAND FLASH 是 SAMSUNG 公司的 K9F280。

2.1 使用前的准备

第一次使用 JDI 时, 仅需要将 2 根电缆和它相连 (USB, RS232 线)。以后再次使用时仅有三根电缆是必要的 (USB, Ethernet, JTAG 线), RS232 可以连接也可以不连接, 不会影响系统的使用。

2.1.1 使用前的准备

使用 JDI 之前, 应按下面的方法将 JDI 和目标板连接起来:

- 用 USB 电缆将 JDI 和计算机或目标板相连, 以提供电源。
- 用 RS232 电缆将 JDI 和计算机相连, 以便在串口上显示配置界面。
- 通过 JDI 上的 Ethernet 接口将 JDI 连接到 Ethernet 上。
- 用 JTAG 线将 JDI 和目标板相连。

若以上连接正确, 将会在串口上看到有信息输出 (串口的配置为: 115200bps, 8N1)。

2.1.2 启动 TFTP 服务

在 Host 上, 进行适当的配置并启动 TFTP 服务。

在 Linux 操作系统下, 启动 TFTP 服务的过程为:

首先, 编辑/etc/xinetd.d/tftp 为如下的内容(假定使用/tftpboot 作为提供 TFTP 服务的目录):

```

service tftp
{
    disable            = no
    socket_type        = dgram
    protocol           = udp
    wait               = yes
    user               = root
    server             = /usr/sbin/in.tftp.d
    server_args        = -s /tftpboot
    ... ..
}

```

然后，运行如下命令：

```
/etc/init.d/xinetd restart
```

在 Windows 操作系统下，需要安装第三方提供的 TFTP 服务器程序方可使用 TFTP 服务。

2.1.3 准备配置文件

配置文件用于描述目标板的硬件特征，并可以定义一系列的预定义的操作，这些预定义的操作可以对目标板进行一些必要的配置，以方便后继的操作。

下面是一个文件名为 `jz4730-pmp.cfg` 的配置文件的內容：

```

;-----
; JDI configuration file for Jz4730 PMP board
;-----
;
[INIT]
; Init INTC
WM32    0xB0001010  0xFFFFFFFF  ;clear all intrs
WM32    0xB0001008  0xFFFFFFFF  ;mask all intrs
; Init PLL
WM32    0xB0000000  0x10422220  ;PLL frequency division register (CFRCR)
WM32    0xB0000010  0x50800520  ;PLL control register (PLCR1)
; Init memory controller
WM32    0xB0010070  0x40000000  ;GPAUR2, GPIO as emc
WM32    0xB0010074  0x00005555  ;GPAUR2, GPIO as emc

```

```
;  
[HOST]  
IP      192.168.1.20      ;The Host IP  
;  
[FLASH]  
CHIPTYPE    AM29          ;Flash type (AUTO|AM29|I28F)  
CHIPSIZE    0x00800000    ;The size of the flash in bytes  
;; Parameters for chiptype I28F and AM29  
CHIPWIDTH    16          ;The width of the flash chip in bits (8|16|32)  
BUSWIDTH     16          ;The width of the flash memory bus in bits (8|16|32)  
;; Parameters for chiptype AM29  
SETUPADDR1   0x555        ;The first setup address of chiptype AM29  
SETUPADDR2   0x2AA        ;The second setup address of chiptype AM29  
;  
[NAND]  
BUSWIDTH     8           ;The width of the NAND flash chip in bits (8|16|32)  
ROWCYCLES    3           ;The row address cycles (2|3)  
PAGESIZE     2048        ;The page size of the NAND chip in bytes(512|2048)  
FORCEERASE   0           ;The force to erase flag (0|1). When set, all blocks including that were  
marked as bad blocks will be erased. Don't set it during normal operation.  
;  
[EEPROM]  
ADDRESS      7           ;The device address of EEPROM  
;  
[SDRAM]  
BUSWIDTH     32          ;The bus width of the SDRAM in bits (16|32)  
BANKS        4           ;The bank number (2|4)  
ROWADDR      13          ;Row address width in bits (11-13)  
COLADDR      9           ;Column address width in bits (8-12)  
CASLATENCY   2           ;CAS latency (2|3)  
;  
[GDB]  
INITFILE     gdbinit.bin  ;Platform-dependent init code, required for gdb debugging  
;  
;; -- END --
```

关于配置文件的内容，请参考<使用和修改配置文件>一章的介绍。

将该配置文件作为目标板的配置文件，并将它拷贝到主机 Host 提供 TFTP 服务的目录中，这样就完成了配置文件的准备工作。

2.1.4 配置文件中的[INIT]段

配置文件中的[INIT]段用来在运行 JDI 命令时对目标板进行初始化，如上面配置文件的例子：关闭中断，

初始化 PLL，初始化 GPIO 管脚等。其中初始化 PLL 用来产生 CPU 和各种系统时钟，设置值跟使用的外部晶振的频率和 CPU 频率有关。下面表格列出几个具体的例子，具体设置方法请参考 JZ4730 的技术规范文档。

表格 JZ4730 PLL 设置值

PLL 输出时钟 (MHz)	PLCR1 (0xB000010)		PLL 时钟分频 (I:S:M:P)	CFCR (0xB000000)
	外部晶振频率 =3.6864MHz	外部晶振频率 =12MHz		
50	0x0d000120	0x03800120	1:2:2:2	0x10411110
100	0x1a800120	0x07800120	1:3:3:3	0x10422220
200	0x35800120	0x10000120	1:4:4:4	0x10433330
300	0x50800120	0x18800120	1:6:6:6	0x10444440
400	0x6c000120	0x20800120	1:8:8:8	0x10455550
关闭	0	0	1:12:12:12	0x10466660

2.1.5 配置 JDI 的操作环境

将 JDI 接入 Ethernet 时，需要给定一个 IP 地址，我们使用 192.168.1.201 作为其 IP 地址。第一次使用 JDI 时，需要配置 JDI 的 IP 地址，提供 TFTP 服务的主机 IP 地址以及配置文件的文件名和路径。当需要改变这些信息时，也需要重新配置它。

先将 JDI 的 Ethernet 连接线拔掉，用（115200bps, 8N1）的配置，在与 JDI 串口相连的计算机上配置串口的输出界面，然后重新启动 JDI（通过插拔 USB 电缆），我们将在串口的输出界面上看到如下的字符输出：

```
JDI>
```

接下来，按照如下的步骤来配置 JDI 的 IP 地址，提供 TFTP 服务的主机 IP 地址，配置文件的文件名和路径信息，以及网络掩码，网关 IP 地址和广播 IP 地址。

```
JDI> ipconfig 192.168.1.201
JDI> fconfig 192.168.1.20 jz4730-pmp.cfg
JDI> netmask 255.255.255.0
JDI> gateway 255.255.255.255
JDI> bcast 192.168.1.255
```

以上操作完成后，这些配置信息将存放到 JDI 的 FLASH 中，以后就可以直接使用它了。

2.1.6 进入 JDI 的操作界面

将 JDI 的 Ethernet 连接线重新插好，然后重新启动 JDI(重新插拔 USB 电缆)，接着我们将在串口的输出上看到如下的操作界面：

```
JDI >
```

同时 JDI 向用户提供了 Telnet 的操作界面，用户也可以通过 telnet 的方式进入 JDI 的操作界面来进行相关的操作。如下所示：

```
$ telnet 192.168.1.201  
JDI>
```

接着，我们就可以在该操作界面上键入 help 命令来看一看帮助信息了。

2.1.7 退出 JDI

直接拔掉 USB 电缆，然后将目标板断电，并将连接在 JDI 上的其它电缆拔掉即可。

2.2 EEPROM 的烧录

目标板上如果有 EEPROM，我们也可以用 JDI 对其进行烧录操作。在进行烧录之前，我们需要在配置文件中加入相应的配置信息，以使 JDI 能正确地访问 EEPROM。

2.2.1 修改配置文件

在配置文件中加入如下的信息：

```
[EEPROM]  
ADDRESS          7
```

[EEPROM]表示 EEPROM 段的开始，该段用来描述关于 EEPROM 操作所需的参数。数值‘7’表示 EEPROM 的地址，这是一个和目标板相关的参数，不同的目标板其数值可能会不同，请参阅相关的硬件文档。

配置文件修改之后不需要重新启动 JDI，因为 JDI 在执行命令之前会重新读取它的配置文件。

2.2.2 烧录 EEPROM

在本节，我们假定将 EEPROM 中的第 8 – 15 个字节的数据更新为 (0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88)。

首先，我们可以将这 8 个字节数据读出来看一下其数值，其操作如下：

```
JDI> eread 8 8  
0x008: 00 00 00 00 00 00 00 00  
JDI>
```

然后，我们可以将这 8 个数据更新为 (0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88)，其操作如下所示：

```
JDI> eprogv 8 1122334455667788  
JDI>
```

最后，我们再将这 8 个数据读出来查看一下，其操作如下：

```
JDI> eread 8 8  
0x008: 11 22 33 44 55 66 77 88  
JDI>
```

按照上述方法，用户可以更新 EEPROM 中任意位置处的任意长度的数据。若烧录到 EEPROM 中的数据量较大时，用户可以将要写入的数据放到文件中，并将该文件拷贝到提供 TFTP 的目录中，然后用文件的方式将数据写入到 EEPROM 中，其详细的操作过程可参阅后继的章节。

2.3 NOR FLASH 的烧录

烧录 NOR FLASH 之前，需要参阅一下有关目标板的硬件文档和所使用的 NOR FLASH 的规格文档。然后根据实际的情况来修改 JDI 的配置文件。在本章节中，我们选择一款较为常用的 jz4730 开发板为例来陈述有关 NOR FLASH 的烧录过程。

2.3.1 修改配置文件

在本例中，我们在配置文件 jz4730.cfg 中加入如下的信息：

```
[FLASH]  
CHIPTYPE          I28F  
CHIPSIZE          0x01000000  
CHIPWIDTH         16  
BUSWIDTH          16
```

‘[FLASH]’ 关键字表示 [FLASH] 段的开始，该段用于描述 NOR FLASH 的相关参数。

在 NOR FLASH 的烧录过程中，同种类型的 FLASH 可以使用相同的算法，所以我们在 CHIPTYPE 描述中指定其类型为 I28F。

在本例中，我们使用的 FLASH 大小为 128M bits，所以定义 CHIPSIZE 为 0x01000000 (16M bytes)。

‘CHIPWIDTH 16’ 表示我们在目标板上将该 FLASH 按 16 位的数据宽度接入。

‘BUSWIDTH 16’ 表示我们在访问该 FLASH 时，在数据总线上看到的数据为 16 位宽度。(有些目标板可以用 32 位的数据总线宽度来同时访问两个 16 位宽的 FLASH，这样，就可以用两个 16 位

宽的 FLASH 来形成一个 32 位宽的数据。而对于每个 FLASH 来说，其宽度还是 16 位的宽度)。

配置文件修改之后不需要重新启动 JDI，因为 JDI 在执行命令之前会重新读取它的配置文件。

2.3.2 烧录 NOR FLASH

将所要烧录的 BIN 文件拷贝到提供 tftpboot 服务的目录或其子目录中，在这里我们使用一个大小为 256K 的 BIN 文件 test256k.bin，并将其拷贝到子目录 jdi 中。在这里，假定提供 TFTP 服务的主机的 IP 地址为 192.168.1.20。

2.3.2.1 读 NOR FLASH 中的数据

在烧录 NOR FLASH 之前，我们可以先看一下 NOR FLASH 中的内容。所使用的命令如下所示：

```
JDI> readf 0xbfc00000 128

0xbfc00000: 00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff
0xbfc00010: 00 10 20 30 40 50 60 70 80 90 a0 b0 c0 d0 e0 f0
0xbfc00020: ... ..
... ..
0xbfc00070: ... ..

JDI>
```

在这里，参数 0xbfc00000 为 NOR FLASH 的起始地址，参数 128 为所要读出的字节数。

2.3.2.2 擦除 NOR FLASH

用 JDI 烧录 NOR FLASH 之前，需要先将该 NOR FLASH 需要烧录的块擦除掉，然后才可以烧录。这里，我们要将 256K 的 Binary 文件 test256k.bin 烧录到 NOR FLASH 开头的 256K 的空间中。我们这里所使用的 NOR FLASH 的块的大小为 128K，所以要先将 NOR FLASH 的头两个块擦除掉。其执行的命令如下所示：

```
JDI> erase 0xbfc00000 0x20000 2
JDI>
```

命令中的第一个参数 0xbfc00000 为 NOR FLASH 的起始地址，第二个参数 0x20000 为 NOR FLASH 一个数据块的大小（128K），第三个参数 2 为所要擦除的数据块个数。该操作执行完后，我们用 readf 命令再读一下 NOR FLASH 的内容，可以看到其数据均为 0xFF，即这两个数据块已经被正确地擦除了。如下所示：

```
JDI> readf 0xbfc00000 128

0xbfc00000: ff ff
0xbfc00010: ff ff
```

```

... ..
0xbfc00070:  ff ff

JDI>

```

2.3.2.3 烧录 NOR FLASH

在 NOR FLASH 中，若将要被烧录的数据块已经被正确的擦除，则可以进行下面的烧录操作。将一个 Binary 文件烧录到 NOR FLASH 中的过程如下所示：

```

JDI> prog 0xbfc00000 192.168.1.20 jdi/test256k.bin
JDI>

```

命令中的第一个参数 0xbfc00000 为 NOR FLASH 的起始地址，第二个参数 192.168.1.20 为提供 TFTP 服务的主机 IP 地址，第三个参数 jdi/test256k.bin 为要烧录的二进制文件。若要烧录的数据块没有被正确的擦除，将不能将该二进制文件烧录到 NOR FLASH 中。我们下面可以执行一下读操作，以检察一下数据是否被正确的写入。如下所示：

```

JDI> readf 0xbfc00000 128
0xbfc00000:  12 34 56 78 23 45 67 89 ab cd ef 34 56 78 9a bc
0xbfc00010:  aa bb cc dd ee ff 24 68 ac e0 13 57 9b df 11 22
... ..
0xbfc00070:  11 33 55 77 99 00 22 44 66 88 ab cd ef 10 22 30

JDI>

```

查看 jdi/test256k.bin，可以发现数据已经被正确地写入到了 NOR FLASH 中。

2.4 NAND FLASH 的烧录

在该章节中，我们使用的 NAND FLASH 为 8 位，共有 1024 个数据块，每个数据块共有 32 个页，每个页的大小为 512 字节，行地址的写入需要 2 个周期。在烧录 NAND FLASH 之前，需要参阅有关目标板的硬件文档和所使用 FLASH 的规格文档。然后根据实际的情况来修改 JDI 的配置文件。

2.4.1 修改配置文件

根据所使用的 NAND FLASH 的参数及实际的情况，我们在配置文件 jz4730.cfg 中加入如下的信息：

```

[NAND]
BUSWIDTH          8
ROWCYCLES         2
PAGESIZE          512
FORCEERASE        0

```

在该配置文件中：

‘[NAND]’ 表示 NAND FLASH 段的开始，该段用来描述与 NAND FLASH 烧录相关的参数。
‘BUSWIDTH 8’ 表示所要烧录的 NAND FLASH 的宽度为 8 位。
‘ROWCYCLES 2’ 表示所要烧录的 NAND FLASH 的行地址需要用两个时钟周期来写入。
‘PAGESIZE 512’ 表示所要烧录的 NAND FLASH 的每个页的大小为 512 字节。
‘FORCEERASE 0’ 表示在擦除 NAND FLASH 时对坏块不进行擦除操作。

配置文件修改之后不需要重新启动 JDI，因为 JDI 在执行命令之前会重新读取它的配置文件。

2.4.2 烧录 NAND FLASH

将所要烧录的 BIN 文件拷贝到提供 tftpboot 服务的目录或其子目录中，在这里我们使用一个大小为 256K 的 BIN 文件 test256k.bin，并将其拷贝到子目录 jdi 中。在这里，假定提供 TFTP 服务的主机的 IP 地址为 192.168.1.20。

2.4.2.1 读 NAND FLASH 中的数据

在烧录 NAND FLASH 之前，我们可以先看一下 NAND FLASH 中的内容，以便于比较。所使用的命令如下所示：

```
JDI> nread 0 512

0x00000000: 00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff
0x00000010: 00 10 20 30 40 50 60 70 80 90 a0 b0 c0 d0 e0 f0
0x00000020: ... ..
... ..
0x000001f0: ... ..

JDI>
```

在这里，nread 命令的第一个参数为 0，表示从 NAND FLASH 的第 0 个数据页开始读，nread 命令的第二个参数为 512，表示共读出 512 个字节的数据。

2.4.2.2 擦除 NAND FLASH

用 JDI 烧录 NAND FLASH 之前，需要先将该 NAND FLASH 需要烧录的块擦除掉，然后才可以烧录。这里，我们要将 256K 的 Binary 文件 test256k.bin 烧录到 NAND FLASH 开头的 256K 的空间中。我们这里使用的 NAND FLASH 每个块的大小为 16K (32 x 512bytes)，所以要先将 NAND FLASH 的头 16 个块擦除掉。其执行的命令如下所示：

```
JDI> nerase 0 16
JDI>
```

命令 `nerase` 中的第一个参数为 0，表示从 NAND FLASH 的第 0 个数据块开始擦除，第二个参数为 16，表示共擦除 16 个数据块。当“FORCEERASE”设置为 0 时，该命令会跳过 NAND 中标记为坏块的块。该操作执行完后，我们用 `nread` 命令再读一下 NAND FLASH 的内容，可以看到其数据均为 0xFF，即这两个数据块已经被正确地擦除了。如下所示：

```
JDI> nread 0 512

0x00000000: ff ff
0x00000010: ff ff
0x00000020: ... ..
... ..
0x000001f0: ... ..

JDI>
```

2.4.2.3 烧录 NAND FLASH

在 NAND FLASH 中，若将要被烧录的数据块已经被正确的擦除，则可以进行下面的烧录操作。将一个 Binary 文件烧录到 NAND FLASH 中的过程如下所示：

```
JDI> nprog 0 192.168.1.20 jdi/test256k.bin
JDI>
```

`nprog` 命令中的第一个参数为 0，表示从第 0 个数据页开始烧录文件，第二个参数 192.168.1.20 为提供 TFTP 服务的主机 IP 地址，第三个参数 `jdi/test256k.bin` 为要烧录的文件。若要烧录的数据页没有被正确的擦除，将不能将该文件烧录到 NAND FLASH 中。我们下面可以执行一下读操作，以检查一下数据是否被正确的写入。如下所示：

```
JDI> nread 0 512
0x00000000: 12 34 56 78 23 45 67 89 ab cd ef 34 56 78 9a bc
0x00000010: aa bb cc dd ee ff 24 68 ac e0 13 57 9b df 11 22
... ..
0x000001f0: 11 33 55 77 99 00 22 44 66 88 ab cd ef 10 22 30

JDI>
```

查看 `jdi/test256k.bin`，可以发现数据已经被正确地写入到了 NAND FLASH 中。

2.5 用 GDB 调试程序

除了上述介绍的内容之外，JDI 还向 GDB 调试器提供了 Ethernet 层的调试接口。用户可以用 GDB 通过 Ethernet 连接到 JDI 来调试底层的程序。在这里，我们通过一个简单的例子来示例一下一个小程序

的具体的调试过程。

在调试程序之前，我们首先需要将程序下载到目标板上，这就需要在目标板上有相应的内存，并且需要作相应的初始化操作。在不同的目标板上，其内存的配置不尽相同，其初始化操作更是千差万别。因此 JDI 需要用户在配置文件中指定目标板的初始化文件（INITFILE）。INITFILE 所指的 BINARY 文件完成目标板的初始化，如初始化 CPU 时钟模块和 SDRAM 存储。要求该 BINARY 的入口地址为 0xff2f,0000，并且必须链接到 0xff2f,0000-0xff2f,1000 地址范围内。GDB 在下载被调试程序到目标板之前，会先执行该 BINARY 完成初始化工作。

在开始启动 GDB 调试之前，需要在 JDI 命令行运行“debug”命令，如下：

```
JDI> debug [program_entry]
***** pid = 0x3e8 *****
Listening on port 6666
```

[program_entry]表示被调试程序的入口地址，为可选参数。这时，在主机端运行 mipsel-linux-gdb 就可以调试代码了。这里，我们使用一个 ELF 格式的文件 t.elf，该文件由一个简单的汇编文件产生，并且已被正确的连接到了从 0x80001000 开始的地址处。其调试过程如下所示：

```
$. /mipsel-linux-gdb .t.elf
(gdb) target remote 192.168.1.201:6666
Remote debugging using 192.168.1.201:6666
0x00000000 in ?? ()
(gdb)
```

在这里，命令‘target remote 192.168.1.201:6666’中，‘192.168.1.201’表示 JDI 的 IP 地址，‘:6666’表示 JDI 向 GDB 提供网络调试接口所使用的 TCP 端口号。该命令正确执行后，就已经与 JDI 建立起了连接。下面就可以将程序下载到目标板上了，如下所示：

```
(gdb) load
```

该命令正确执行后，就已经将可执行程序 t.elf 下载到了目标板上了。下面就可以调试程序了，如下所示：

```
(gdb) disp/i $pc
1: x/i $pc 0x800010b0 <_ftext>: lui a3, 0xb600
(gdb) si
1: x/i $pc 0x800010b0 <_ftext>: lui a3, 0xb600
(gdb) si
0xffffffff800010b4 in _ftext ()
1: x/i $pc 0x800010b4 <_ftext + 4>: ori a3, a3, 0x58
(gdb) p/x $a3
$1 = 0xb6000000
(gdb) si
```

```
0xffffffff800010b8 in _ftext ()
1: x/i $pc 0x800010b8 <_ftext + 8>: lui t0, 0x30
(gdb) si
0xffffffff800010bc in _ftext ()
1: x/i $pc 0x800010bc <_ftext + 12>: sb t0, 0(a3)
(gdb) p/x $t0
$2 = 0x30
(gdb) q
```

以上的调试过程所使用的命令均为 `gdb` 的标准命令，关于 `gdb` 详细的使用方法可参阅相关的文档。需要说明的是，具体的程序调试过程是在 `gdb` 中来使用 `gdb` 命令进行的，`JDI` 和 `gdb` 之间的操作对用户来说是透明的，用户也不需要关心具体的细节。

3 使用 JDI 的命令界面

JDI 向用户提供了命令界面，用户可以在该界面上执行所需要的操作。当前，JDI 向用户提供的操作命令主要有三类，那就是配置及查询命令，FLASH 及 EEPROM 操作的命令，和测试升级及退出命令。

当在命令界面上执行所需要的操作时，请确认没有 gdb 在使用该 JDI 进行调试操作，否则操作将不能正确完成。

3.1 进入及退出 JDI 的命令界面

进入及退出 JDI 命令界面的方法有两种，一种是通过串口的方式，另外一种是通过 Telnet 来远程登录。

3.1.1 通过串口来进入及退出 JDI 的命令界面

首先，将 JDI 正确地连接到目标板上，并将一台计算机的串口和 JDI 的串口相连，在该计算机上用下列参数来配置其串口：

波特率为：115200
数据位为：8
奇偶校验：无
停止位： 1
数据流控制：无

然后，重新启动 JDI（插拔 USB 电缆），我们将在计算机的串口界面上看到 JDI 的输出，最后我们可以看到 JDI 的命令提示符“JDI>”，这就说明我们已经通过串口进入到了 JDI 的命令界面。在该串口界面上键入 help，我们可以看到如下的输出：

```
JDI> help
JDI Commands:

help          print this help
ipconfig      set local ip
... ..       ... ..

JDI>
```

这就说明我们已经可以在该操作界面来执行相应的操作了。

要退出该操作界面，可以直接在计算机上退出相应的操作窗口就可以了。

3.1.2 通过远程登陆来进入及退出 JDI 的命令界面

JDI 向用户提供了标准的 Telnet 远程登陆窗口，所以用户可以在任何计算机上用标准的 telnet 命令通

过远程登陆来进入 JDI 的操作界面。如下所示:

```
$ telnet 192.168.1.201
JDI>
```

当我们看到 JDI 的命令提示符“JDI>”时,说明我们已经进入了 JDI 的命令界面。`telnet` 命令中的参数‘192.168.1.201’为该 JDI 的 IP 地址。此时,就可以在该命令提示符下执行相应的操作了,如下所示:

```
JDI> help
```

执行该命令后,就可以在屏幕上看到相应的输出了,这就说明现在可以执行所需要的操作了。要退出 JDI 的命令界面,仅需要在其操作符下执行 `exit` 命令,如下所示:

```
JDI> exit
$
```

3.2 配置及查询命令

配置及查询命令主要用来完成下述的功能:

- 修改 JDI 的配置信息
- 查询目标板上的 NOR FLASH 的情况
- 查看 JDI 在命令界面中支持的所有命令

3.2.1 帮助命令 help

`help` 命令用来查询 JDI 在其命令界面中支持的所有命令。其使用方法如下所示:

```
JDI> help
JDI commands:

help          print this help
ipconfig      set local ip
... ..       ... ..

JDI>
```

该命令在屏幕上列出了 JDI 命令界面中所有向用户提供的命令,同时也给出了命令的解释。该命令的使用会帮助用户快速找到其操作所使用的命令。

3.2.2 配置命令 ipconfig

`ipconfig` 命令用来查询和修改 JDI 的 IP 地址,该命令的格式为:

`ipconfig` <该 JDI 所要使用的 IP 地址>

举例如下:

```
JDI> ipconfig 192.168.1.101
JDI>
```

3.2.3 配置命令 `fconfig`

`fconfig` 命令用来查询和修改 JDI 的配置文件信息, 该命令的格式为:

`fconfig` <配置文件所在的主机 IP 地址> <配置文件名及其存放位置>

举例如下:

```
JDI> fconfig 192.168.1.20 jz4730/pmp.cfg
JDI>
```

3.2.4 配置命令 `netmask`

`netmask` 命令用来查询和修改 JDI 的网络掩码, 该命令的格式为:

`netmask` <掩码地址>

举例如下:

```
JDI> netmask 255.255.255.0
JDI>
```

3.2.5 配置命令 `gateway`

`gateway` 命令用来查询和修改 JDI 的网关 IP 地址, 该命令的格式为:

`gateway` <网关 IP 地址>

举例如下:

```
JDI> gateway 255.255.255.255
JDI>
```

3.2.6 配置命令 `bcast`

`bcast` 命令用来查询和修改 JDI 的广播 IP 地址, 该命令的格式为:

```
bcast <广播 IP 地址>
```

举例如下:

```
JDI> bcast 192.168.1.255  
JDI>
```

3.2.7 显示配置命令 **showcfg**

showcfg 命令用来显示 JDI 的所有配置信息, 该命令的格式为:

```
showcfg
```

举例如下:

```
JDI> showcfg  
  
LOCAL IP: 192.168.1.201  
HOST IP : 192.168.1.20  
CFGFILE: jdi.cfg  
...
```

3.3 调试命令 **debug**

debug 命令用来启动 GDB 的远程调试功能, 该命令的格式为:

```
debug [被调试程序入口地址]
```

参数[被调试程序入口地址]用来通知 GDB 被调试程序的入口地址。此参数为可选项。

举例如下:

```
JDI> debug  
***** pid=0x3e8 *****  
Listening on port 6666
```

3.4 FLASH 及 EEPROM 操作的命令

FLASH 及 EEPROM 操作的命令主要用来完成下述的功能:

- EEPROM 的读写操作
- NOR FLASH 的查询操作, 读取操作, 擦除操作以及烧录操作
- NAND FLASH 的读取操作, 擦除操作以及烧录操作

在执行这些命令之前，需要根据目标板的实际配置情况，在 JDI 的配置文件中来增加或修改相应的配置信息。

3.4.1 EEPROM 读取操作命令 `eread`

`eread` 命令用读取 EEPROM 中的数据，其命令行格式有两类，分别为：

‘`eread start count`’

该命令表示从 EEPROM 上偏移为 **start** 的位置开始，连续读出 **count** 个字节，然后在屏幕上显示。

‘`eread start count file-name`’

该命令表示从 EEPROM 上偏移为 **start** 的位置开始，连续读出 **count** 个字节，然后将这些数据写入到文件名为 ‘**file-name**’ 的文件中。该命令要求该文件为提供 TFTP 服务的目录或其子目录中已经存在的文件，也就是说该命令将不会在提供 TFTP 服务的主机上新建文件，但是它可以覆盖原有的文件。文件名 ‘**file-name**’ 中也可以包含相对的路径，表示从提供 TFTP 服务的目录开始的相对路径。该命令中，用于提供 TFTP 服务的主机的 IP 地址在 JDI 配置文件中的 [HOST] 段中给出。

‘`eread start count host file-name`’

该命令的说明及使用方法与上面的命令类似。所不同的是，该命令中的参数 **host** 用于指定提供 TFTP 服务的主机的 IP 地址，而不是用 JDI 配置文件中的 [HOST] 段来指定。

例如，读取 EEPROM 中的第 12 到第 19 个字节的命令如下所示：

```
JDI> eread 12 8
0x00c: 11 22 33 44 55 66 77 88
JDI>
```

将 EEPROM 中开头的 256 个字节写入到文件 `jdi/eeprom.bin` 的示例如下：

```
JDI>eread 0 256 192.168.1.20 jdi/eeprom.bin
```

3.4.2 将字符串写入 EEPROM 的命令 `eprogv`

在对 EEPROM 中个别数据进行改写时，我们可以使用命令 `eprogv`。其命令行格式如下所示：

‘`eprogv start hex`’ 该命令中，‘**hex**’ 为用 16 进制表示的字符串。该命令表示将 EEPROM 中，从偏移为 **start** 的位置处开始的若干个字节更新为 ‘**hex**’ 表示的 16 进制字符串。其示例如下：

```
JDI> eprogv 12 1020304050607080
```

该命令表示将 EEPROM 中的第 12 到第 19 个字节更新为：‘0x10, 0x20, 0x30, 0x40, 0x50, 0x60, 0x70, 0x80’。为便于验证，我们可以用命令将其读出，如下所示：

```
JDI> erread 12 8
0x00c: 10 20 30 40 50 60 70 80
JDI>
```

从上述结果可以看出，命令 `eprogv` 已经被正确执行完成了。

3.4.3 将文件写入 EEPROM 的命令 `eprog`

该命令将用于提供 TFTP 服务的目录或其子目录中的一个文件写入到 EEPROM 中。其命令有两种格式，分别如下所示：

‘`eprog start file-name`’

该命令中，‘`file-name`’ 表示用来更新 EEPROM 的文件名。该文件名中没有路径信息时，表示该文件在用于提供 TFTP 服务的目录下。该文件名中有相对路径时，表示该文件在从用于提供 TFTP 服务的目录开始的相对路径下。该命令中，用于提供 TFTP 服务的主机的 IP 地址在 JDI 配置文件中的 [HOST] 段中给出。

该命令的执行结果是将 EEPROM 中，从偏移为 ‘`start`’ 开始的若干个字节，更新为 ‘`file-name`’ 所指定的文件内容。

‘`eprog start host file-name`’

该命令中，‘`host`’ 为另外一台可以提供 TFTP 服务的计算机的 IP 地址。该命令的执行过程同前面提到的命令类似，所不同的是该命令存取的文件 ‘`file-name`’，是在 ‘`host`’ 所指定的提供 TFTP 服务的主机上。

例如，可以将 EEPROM 中以偏移 0 开始的若干个字节，更新为文件 `jdi/eprom.bin`（该文件位于提供 TFTP 服务的目录中）的内容。该命令中，用于提供 TFTP 服务的主机的 IP 地址在 JDI 配置文件中的 [HOST] 段中给出。其命令如下所示：

```
JDI> eprog 0 jdi/eprom.bin
```

若该文件位于另一台提供 TFTP 服务的计算机上（假定该计算机的 IP 地址为 192.168.1.30），则其命令如下所示：

```
JDI> eprog 0 192.168.1.30 jdi/eprom.bin
```

3.4.4 NOR FLASH 查询命令 `query`

该命令可以用来查询目标板上 NOR FLASH 的有关信息。若该 JDI 的配置文件关于 NOR FLASH 的类型的定义不是 AUTO 时，使用该命令将不能查询到有关 NOR FLASH 的信息。该命令使用方法如下所示：

```
JDI> query
... ..
```

JDI>

3.4.5 NOR FLASH 及其 sector 查询命令 querya

该命令用来查询目标板上 NOR FLASH 的相关信息及其每个 sector 的地址。若该 JDI 的配置文件中关于 NOR FLASH 的类型的定义不是 AUTO 时，使用该命令将不能查询到有关 NOR FLASH 的信息及其 sector 的地址。该命令使用方法如下所示：

```
JDI> querya
... ..
JDI>
```

3.4.6 NOR FLASH 的读取命令 readf

‘readf’ 命令用于将从某一地址开始 NOR FLASH 中的若干个字符读出，该命令要求该地址必须在 NOR FLASH 上，并且保证该读取操作不会超出 NOR FLASH 的边界。该命令的格式有两种，分别如下所示：

‘readf address count’

该命令表示从地址 ‘address’（该地址必须位于 NOR FLASH 上）开始，连续读出 ‘count’ 个字节，然后在屏幕上显示。

‘readf address count file-name’

该命令表示从地址 ‘address’（该地址必须位于 NOR FLASH 上）开始，连续读出 ‘count’ 个字节，然后将这些数据写入到文件名为 ‘file-name’ 文件中。该命令要求该文件为提供 TFTP 服务的目录或其子目录中已经存在的文件，也就是说该命令将不会在提供 TFTP 服务的主机上新建文件，但是它可以覆盖原有的文件。文件名 ‘file-name’ 中也可以包含相对的路径，表示从提供 TFTP 服务的目录开始的相对路径。该命令中，用于提供 TFTP 服务的主机的 IP 地址在 JDI 配置文件中的[HOST]段中给出。

‘readf address count host file-name’

该命令的说明及使用方法与上面的命令类似。所不同的是，该命令中的参数 **host** 用于指定提供 TFTP 服务的主机的 IP 地址，而不是用 JDI 配置文件中的[HOST]段来指定。

例如，将 NOR FLASH 中从 0xbfc00000 开始的 256 个字节读出，并在屏幕上显示的命令如下所示：

```
JDI> readf    0xbfc00000    256

0xbfc00000: 00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff
0xbfc00010: 00 10 20 30 40 50 60 70 80 90 a0 b0 c0 d0 e0 f0
0xbfc00020: ... ..
... ..
0xbfc00100: ... ..
```

```
JDI>
```

这里，‘0xbfc00000’为 NOR FLASH 中的一个地址。

将 NOR FLASH 中从 0xbfc00000 开始的 256K 字节读出，并将其写入到文件 `jdi/norflash.bin` 中的示例如下：

```
JDI> readf    0xbfc00000    256    jdi/norflash.bin
JDI>
```

3.4.7 NOR FLASH 的擦除命令 `erase`

‘`erase`’命令用来将 NOR FLASH 中的数据擦除。当我们将一个文件烧录到 NOR FLASH 时，必须先用该命令将 NOR FLASH 擦除，然后才可以烧录，否则不能将文件正确的烧录。其命令行格式有两种，分别为：

‘`erase address chip`’

该命令用于将整个芯片进行擦除操作，对于支持该命令的 NOR FLASH 方可执行该操作。该命令中，‘`address`’为 NOR FLASH 上的任意地址，‘`chip`’的取值为字符串“`chip`”。

‘`erase address step count`’

该命令用于将 NOR FLASH 上的一个或多个连续的数据块进行擦除。命令中‘`address`’为第一个数据块的起始地址，‘`step`’为一个数据块的大小（以字节为单位），‘`count`’为连续擦除的数据块的个数。该命令要求用户必须知道所使用的 NOR FLASH 数据块的大小和其起始地址。

例如，将目标板上的 NOR FLASH 中的数据全部擦除的命令如下所示（本例中假定目标板上只有一块 NOR FLASH，并且其起始地址为 0xbfc00000）：

```
JDI> erase    0xbfc00000    chip
JDI>
```

将目标上 NOR FLASH 前 256K 中的内容全部擦除的命令如下所示（本例中假定目标板上的 NOR FLASH 的起始地址为 0xbfc00000，其每个数据块的大小为 0x20000）：

```
JDI> erase 0xbfc00000    0x20000    2
JDI>
```

3.4.8 NOR FLASH 的烧录命令 `prog`

该命令将用于提供 TFTP 服务的目录或其子目录中的一个文件烧录到 NOR FLASH 中，该命令使用之前必须先对该 NOR FLASH 执行擦除操作。其命令有两种格式，分别如下所示：

‘prog address file-name’

该命令中，‘file-name’表示用来烧录 NOR FLASH 的文件名。该文件名中没有路径信息时，表示该文件在用于提供 TFTP 服务的目录下。该文件名中有相对路径时，表示该文件在从用于提供 TFTP 服务的目录开始的相对路径下。该命令中，用于提供 TFTP 服务的主机的 IP 地址在 JDI 配置文件中的 [HOST]段中给出。

该命令的执行结果是将 NOR FLASH 中，从地址 ‘address’ 开始的若干个字节，烧录成 ‘file-name’ 所指定文件的内容。

‘prog address host file-name’

该命令中，‘host’为另外一台可以提供 TFTP 服务的计算机的 IP 地址。该命令的执行过程同前面提到的命令类似，所不同的是该命令存取的文件 ‘file-name’，是在 ‘host’ 所指定的提供 TFTP 服务的主机上。

例如，将一个大小为 256K 的文件 `jdi/test256k.bin`（该文件位于提供 TFTP 服务的目录中），烧录到 NOR FLASH 上开头 256K 字节中的命令如下所示（假定目标板上的 NOR FLASH 的起始地址为 `0xbfc00000`）：

```
JDI> prog 0xbfc00000 jdi/test256k.bin
JDI>
```

若该文件位于另一台提供 TFTP 服务的计算机上（假定该计算机的 IP 地址为 `192.168.1.30`），则其命令如下所示：

```
JDI> prog 0xbfc00000 192.168.1.30 jdi/test256k.bin
```

3.4.9 NAND FLASH 的查询命令 `nquery`

‘nquery’命令用于查询 NAND FLASH 信息。通过该命令可以知道 NAND FLASH 的生产产家，型号和容量大小等信息：

```
JDI> nquery
NAND querying ... completed.
NAND device: Vendor ID 0xec, Chip ID 0xf1 (Samsung NAND 128MiB 3,3V 8-bit)
```

3.4.10 NAND FLASH 的读取命令 `nread`

‘nread’命令用于将 NAND FLASH 中的数据读出，并在屏幕上显示或将其存入到文件中。在读取数据时，该命令会跳过标记为坏块的数据块。在读取某页的数据时，该命令还会做硬件 ECC 的检测和校验。该命令的格式有三种，分别如下所示：

‘nread start count’

该命令表示在 NAND FLASH 中，从地址为 ‘start’ 的数据页中，连续读出 ‘count’ 个字节的数，然后在屏幕上显示。

‘nread start count file-name’

该命令表示在 NAND FLASH 中，从地址为 ‘start’ 的数据页中，连续读出 ‘count’ 个字节的数据，然后将这些数据写入到文件名为 ‘file-name’ 文件中。该命令要求该文件为提供 TFTP 服务的目录或其子目录中已经存在的文件，也就是说该命令将不会在提供 TFTP 服务的主机上新建文件，但是它可以覆盖原有的文件。文件名 ‘file-name’ 中也可以包含相对的路径，表示从提供 TFTP 服务的目录开始的相对路径。该命令中，用于提供 TFTP 服务的主机的 IP 地址在 JDI 配置文件中的[HOST]段中给出。

‘nread start count host file-name’

该命令的说明及使用方法与上面的命令类似。所不同的是，该命令中的参数 **host** 用于指定提供 TFTP 服务的主机的 IP 地址，而不是用 JDI 配置文件中的[HOST]段来指定。

例如，将 NAND FLASH 中，第 0 个数据页中的 512 字节的数据读出，并将其显示在屏幕上的操作为：

```
JDI> nread 0 512

0x00000000: 00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff
0x00000010: 00 10 20 30 40 50 60 70 80 90 a0 b0 c0 d0 e0 f0
0x00000020: ... ..
... ..
0x000001f0: ... ..

JDI>
```

将 NAND FLASH 中，第 0 个数据页中 512 字节的数据读出，并将其写入到文件 `jdi/nandflash.bin` 中的示例如下：

```
JDI> nread 0 512 jdi/nandflash.bin
JDI>
```

3.4.11 NAND FLASH 的读取命令 **nreadraw**

‘nreadraw’ 命令用于将 NAND FLASH 中的数据读出，并在屏幕上显示或将其存入到文件中。该命令与 ‘nread’ 命令的区别在于：在读取数据时，该命令不会跳过标记为坏块的数据块。在读取某页的数据时，该命令也不做硬件 ECC 的检测和校验。该命令的格式与 ‘nread’ 相同。

3.4.12 NAND FLASH 的读取命令 **nreadoob**

‘nreadoob’ 命令用于将 NAND FLASH 中的 oob 读出，并在屏幕上显示或将其存入到文件中。在读取 oob 时，该命令不会跳过标记为坏块的数据块。在读取某页的数据时，该命令也不做硬件 ECC 的检测和校验。该命令的格式与 ‘nread’ 相同。

所指定文件的内容。

‘nprog start host file-name’

该命令中，‘host’ 为另外一台可以提供 TFTP 服务的计算机的 IP 地址。该命令的执行过程同前面提到的命令类似，所不同的是该命令存取的文件 ‘file-name’，是在 ‘host’ 所指定的提供 TFTP 服务的主机上。

例如，将一个大小为 256K 的文件 **jdi/test256k.bin**（该文件位于提供 TFTP 服务的目录中），烧录到 NAND FLASH 上开头 256K 字节中的命令如下所示：

```
JDI> nprog 0 jdi/test256k.bin
JDI>
```

若该文件位于另一台提供 TFTP 服务的计算机上（假定该计算机的 IP 地址为 192.168.1.30），则其命令如下所示：

```
JDI> nprog 0 192.168.1.30 jdi/test256k.bin
```

3.5 测试、升级及退出命令

测试升级及退出命令主要用来完成下述的功能：

- 测试 JDI 与目标板的连接是否正常
- 升级 JDI 的 FirmWare 版本
- 退出 JDI 的 telnet 登陆界面

3.5.1 测试命令 readids

该命令用来读出目标板上的 EJTAG TAP 中的 IDCODE 及 IMPCODE 寄存器的值，以测试 JDI 与目标板的连接是否正常，或者用于检查目标板 CPU 的类型。其命令格式及使用示例如下所示：

```
JDI> readids
target idcode is 0x0000024f, imPCODE is 0x20404000
JDI>
```

3.5.2 测试命令 cputest

该命令用来测试 CPU 对 JTAG 端口的具体访问，其命令格式及使用示例如下所示：

```
JDI> cputest
CPU test ... completed.
CPU runs normally.
```

上述结果表示 CPU 运行正常。

3.5.3 测试命令 memtest

该命令用来测试目标板 SDRAM 是否工作正常，其命令格式及使用示例如下所示：

‘memtest address size’

该命令中，‘address’表示测试的起始地址，‘size’表示测试范围的字节大小。

```
JDI> memtest 0x80000000 0x1000000  
Checking memory from 0x80000000 to 0x80ffffff.  
Checking memory passed.
```

运行该命令前，必须根据目标板 SDRAM 来修改 JDI 配置文件中[SDRAM]段的参数并与之相符。以上提示信息说明目标板的 SDRAM 工作正常。如果不正常，则会显示出错的地址及对应的值。

3.5.4 测试命令 gpios 和 gpioc

‘gpios’和‘gpioc’命令分别用来设置和清除目标板的 GPIO 管脚，其命令格式如下所示：

‘gpios pin’

‘gpioc pin’

其中‘pin’代表管脚号，如 0, 1, ..., 127。

3.5.5 测试命令 hwtest1 和 hwtest2

该命令用来使 CPU 在 memory 总线上输出方波，通过该命令可以检查目标板的电路连接。其命令格式及使用示例如下所示：

```
JDI> hwtest1  
JDI> hwtest2
```

3.5.6 复位命令 reset

该命令用来产生硬件复位信号以复位目标板。其命令格式及使用示例如下所示：

```
JDI> reset  
Resetting target ... done.  
JDI>
```

3.5.7 查看 Firmware 版本命令 version

该命令用来查看 JDI 的 firmware 版本信息。其命令格式及使用示例如下所示：

```
JDI> version
```

```
The current jdi version is 01.01.06.
```

3.5.8 FirmWare 升级命令 upgrade

该命令用来查询和升级 JDI 中的 FirmWare 的版本。在升级 FirmWare 的版本之前，用户需要将 JDI 的盒子打开，将主板上的跳线 JP1 连上，升级完成以后，记着还要将该跳线断开。关于该跳线的说明，请参考 JDI 的硬件说明一章中的介绍。

其命令的格式如下所示：

‘upgrade’

该命令后没有参数时表示用来查询 JDI 中的 FirmWare 的版本。

‘upgrade <hostip> <binary file>’

该命令中，参数 **‘<binary file>’** 用来指定该命令所使用的用来升级的文件名，该文件名中没有路径信息时，表示该文件在用于提供 TFTP 服务的目录下。该文件名中有相对路径时，表示该文件在从用于提供 TFTP 服务的目录开始的相对路径下。该命令中，用于提供 TFTP 服务的主机的 IP 地址由参数 **‘<hostip>’** 来指定。

举例来说，查询 JDI 中的 FirmWare 版本信息的示例如下：

```
JDI> upgrade
```

```
The current firmware version is 01.01.06
```

```
JDI>
```

升级 JDI 中的 FirmWare 版本的示例如下：

```
JDI> upgrade 192.168.1.20 JDI-FW-01.01.08
```

```
JDI>
```

3.5.9 批命令运行命令 run

该命令用来运行位于主机上的批命令文件，用来一次性运行多个 JDI 命令。其命令格式及使用示例如下所示：

```
run <HOST IP> <FILE>
```

```
JDI> run 192.168.1.20 jdi-batch
```

```
JDI>
```

‘jdi-batch’ 文件位于主机 192.168.1.20 的 TFTP 目录下，其文件内容示例如下：

```
nquery  
nerase 0 32  
nprog 0 t.bin  
...
```

3.5.10 退出 telnet 界面的命令 **exit**

该命令用于退出 telnet 的登陆界面。其命令格式及使用示例如下所示：

```
JDI> exit  
Connection closed by foreign host.  
$ >
```


4 使用及修改配置文件

JDI 的配置文件主要用来对所使用的目标板的配置情况进行描述，同时也可以定义一些有关目标板的初始化操作，以便让 JDI 在其上电之后就可以首先执行这些初始化的操作，为后面的操作做好准备。在配置文件中，使用不同的段来定义不同类型的配置或相应的操作。完整的 JDI 配置文件共包含 5 种类型的段定义，如下所示：

- [INIT]段：该段用来定义一些有关目标板的初始化操作。
- [HOST]段：该段用来定义用于提供 TFTP 服务的缺省主机的 IP 地址
- [FLASH]段：该段用来定义目标板上 NOR FLASH 的配置情况
- [NAND]段：该段用来定义目标板上 NAND FLASH 的配置情况
- [EEPROM]段：该段用来定义目标板上 EEPROM 的配置情况
- [SDRAM]段：该段用来定义目标板上 SDRAM 的配置情况
- [GDB]段：该段用来定义 GDB 调试时的配置情况

根据目标板的实际的情况和所执行的操作，有些定义段可以没有，甚至可以不定义任何段，即允许 JDI 的配置文件为一个空文件。

同样，也可以在执行某一特定的操作时，临时地在配置文件中增加或修改配置信息。配置文件修改之后，不需要重新启动 JDI，因为 JDI 在执行每一条命令之前，会自动通过 TFTP 服务来访问该配置文件。

当配置文件不存在或有错误时，在 JDI 启动时，将会在串口上打印出错误的信息。

配置文件文件的书写格式如下所示：

```
[段名]
操作或参数定义一
操作或参数定义二
```

.....

```
[段名]
操作或参数定义一
操作或参数定义二
```

.....

.....

一个示例的文件如下所示：

```
[INIT]
WM32      0xb3010010    0x0fff7700
```

```
[HOST]
IP          192.168.1.20
```

```
[FLASH]
.....
.....
```

4.1 [INIT]段的定义

在该段中，用来定义其预处理操作的方式主要是通过写命令行来实现的。其格式如下所示：

```
<命令>    <参数 1>    <参数 2>
```

若在 JDI 的配置文件中加入这样的命令行，那么在 JDI 执行命令之前，便会根据该命令行的定义来执行相应的操作。

目前，在该段中用来定义预处理操作的命令，仅提供了两类写操作命令，那就是 CP0 寄存器的写操作命令和对某一个地址进行写操作的命令。

4.1.1 写 CP0 寄存器的命令 WCP0

该命令用来将一个值写入到某一个 CP0 寄存器中，其命令行格式如下所示：

```
WCP0 <Reg.select> <Value>
```

该命令行表示将值<Value>，写入到编号为<Reg>，Select 为<Select>的 CP0 寄存器中。例如，将值 0x1 写入到 CP0 Config7（编号为 16，Select 为 7）寄存器的命令为：

```
WCP0 16.7 0x1
```

该命令表示用来关闭 JZ4730 CPU 核的 BTB 优化。

4.1.2 对某一地址进行写操作的命令

在配置文件中，可以将一特定值写入到一个指定的地址中。通过这种方法，用户可以对目标板的 CPU 进行合理的配置。根据写入数据宽度的不同，该命令共有 3 种类型，分别如下所示：

```
'WM32 <ADDRESS> <VALUE>'
```

该命令表示将值<Value>按 32 位的数据宽度写入到<Address>指定的地址中。

```
'WM16 <ADDRESS> <VALUE>'
```

该命令表示将值<Value>按 16 位的数据宽度写入到<Address>指定的地址中。

```
'WM8 <ADDRESS> <VALUE>'
```

该命令表示将值<Value>按 8 位的数据宽度写入到<Address>指定的地址中。

例如，对 JZ4730 CPU 的 EMC 进行配置时，可使用如下的命令：

```
WM32    0xb3010010    0x0fff7700
```

在实际的目标板上，根据其硬件的特性，可能要执行多条类似这样的配置语句，才能完成 EMC 的配置。

4.2 [INIT]段的定义

该段用于提供一台用于提供 TFTP 服务的计算机的 IP 地址。在使用 JDI 的过程中，有时需要根据所执行的命令，使用用户给定的 IP 地址，通过 TFTP 服务到相应的计算机上来存取相应的文件。在这些操作中，若用户没有指定 IP 地址，则使用[HOST]段定义中给定的 IP 地址。其格式为：

```
IP      <计算的 IP 地址>
```

在这里，若访问的文件为配置文件时，将使用用户在 JDI 的配置中所指定的 IP 地址。也就是说，该段的定义将不会影响 JDI 的配置文件的访问。

该段定义的一个示例如下：

```
[HOST]  
IP      192.168.1.20
```

4.3 [FLASH]段的定义

该段用来定义目标板上有关 NOR FLASH 的一些配置参数值。用 JDI 对目标板上的 NOR FLASH 进行烧录或擦除操作时，首先要根据配置文件中[FLASH]段中的定义选择一种算法，然后再根据其它的配置值来进行相应的操作。用户需要根据目标板上 NOR FLASH 的配置情况和该 NOR FLASH 规格书的说明，来选择不同的算法，并根据算法的不同来定义不同的配置项。

目前，JDI 支持的烧录或擦除的算法有 3 种类型，分别为 AUTO，I28F 和 AM29。下面，我们将针对这 3 种算法类型的 NOR FLASH，给出关于[FLASH]段的定义要求和方法。这里要说明的是，这 3 种算法类型的 NOR FLASH 可能会有交叉，这种情况下，用户只需选择其中的一种即可。

4.3.1 AUTO 算法类型的[FLASH]段定义

属于该算法类型的 NOR FLASH 有以下的类型：

```
SST28SF020, SST28SF040  
SST39VF010, SST39VF020, SST39VF040  
SST39SF010, SST39SF020, SST39SF040  
SST39VF1601, SST39VF1602
```

SST39VF3201, SST39VF3202
SST39VF6401, SST39VF6402
 以及所有支持 CFI 算法 NOR FLASH

该种类型的 NOR FLASH 需要在[FLASH]段中给出以下两个参数的定义:

‘CHIPTYPE type’

该参数用来选择 ‘CHIPTYPE’ 的类型, 其类型值为 ‘type’。即用来定义该 NOR FLASH 应该属于何种算法类型, 所以在这里, ‘type’ 的值应该为 ‘AUTO’。

‘CHIPSIZE Value’

该参数用来定义目标板上 NOR FLASH 的大小, 其取值为 ‘Value’。‘Value’ 是用字节数来表示的大小值, 例如可以将其取为 0x00400000, 即为 4M 字节大小。

下面为一个使用该种类型 NOR FLASH 的目标板上, 其配置文件中关于[FLASH]段定义的示例:

```
[FLASH]
CHIPTYPE            AUTO
CHIPSIZE            0X00400000
```

4.3.2 I28F 算法类型的[FLASH]段定义

属于该算法类型的 NOR FLASH, 其擦除和烧录算法与 Intel 公司 28F 系列的 NOR FLASH 相同, 用户需要根据目标板上 NOR FLASH 的配置情况和该 FLASH 的规格书的说明, 来正确地判断所使用的 FLASH 是否属于该类型。

该种类型的 NOR FLASH 需要在[FLASH]段中给出以下 4 个参数的定义:

‘CHIPTYPE type’

该参数用来选择 ‘CHIPTYPE’ 的类型, 其类型值为 ‘type’。即用来定义该 NOR FLASH 应该属于何种算法类型, 所以在这里, ‘type’ 的值应该为 ‘I28F’。

‘CHIPSIZE Value’

该参数用来定义目标板上 NOR FLASH 的大小, 其取值为 ‘Value’。‘Value’ 是用字节数来表示的大小值, 例如可以将其取为 0x00400000, 即为 4M 字节大小。

‘CHIPWIDTH Value’

该参数用来定义目标板上 NOR FLASH 的数据宽度, 其取值为 ‘Value’。‘Value’ 是用位数来表示的数据的宽度, 其取值为 8, 16 或 32。用户需要根据 NOR FLASH 的型号和目标板上的连接情况来正确地定义该参数。

‘BUSWIDTH Value’

该参数用来定义目标板上访问 NOR FLASH 时, 所使用的数据总线宽度, 其取值为 ‘Value’。有些目

标板上，有可能将两个 16 位宽的 NOR FLASH 并行地连接到一条 32 位宽的数据总线上，这样就可以一次存取一个 32 位宽度的数据。而对于每一个 NOR FLASH 来讲，其访问的数据宽度仍旧为 16 位。所以，对于配置文件中的[FLASH]段来讲，参数 ‘CHIPWIDTH’ 和参数 ‘BUSWIDTH’ 可能一致，也可能不一致，需要分别定义。

‘Value’ 是用位数来表示的数据总线的宽度，其取值为 8，16 或 32。用户需要根据目标板上的实际连接情况来正确地定义该参数。

下面为一个使用该种类型 NOR FLASH 的目标板上，其配置文件中关于[FLASH]段定义的示例：

```
[FLASH]
CHIPTYPE          AUTO
CHIPSIZE          0X00400000
CHIPWIDTH         16
BUSWIDTH          16
```

4.3.3 AM29 算法类型的[FLASH]段定义

属于该算法类型的 NOR FLASH，其擦除和烧录算法与 AMD 公司的 AM29 系列的 NOR FLASH 基本相同，所不同的是在对该 FLASH 发送操作命令时，所使用的命令地址有可能不同，其具体的地址值需要在[FLASH]段中给出。用户需要根据目标板上 NOR FLASH 的配置情况和该 FLASH 的规格书的说明，来正确地判断所使用的 FLASH 是否属于该类型。

该种类型的 NOR FLASH 需要在[FLASH]段中给出以下 6 个参数的定义：

‘CHIPTYPE type’

该参数用来选择 ‘CHIPTYPE’ 的类型，其类型值为 ‘type’。即用来定义该 NOR FLASH 应该属于何种算法类型，所以在这里，‘type’ 的值应该为 ‘AM29’。

‘CHIPSIZE Value’

该参数的使用及其说明同 I28F 算法类型的 FLASH。

‘CHIPWIDTH Value’

该参数的使用及其说明同 I28F 算法类型的 FLASH。

‘BUSWIDTH Value’

该参数的使用及其说明同 I28F 算法类型的 FLASH。

‘SETUPADDR1 Value’

该参数用来定义当向 NOR FLASH 发送操作命令时，所使用的第一个地址值，该参数说明中的 ‘Value’ 给出了具体的值。不同型号的 NOR FLASH，该值是不同的，用户需要根据 NOR FLASH 规格书的说明来给出该值。以 AMD 公司的 AM29F040B 型号的 NOR FLASH 为例，则 ‘Value’ 应取值为 0x555。

‘SETUPADDR2 Value’

该参数用来定义当向 NOR FLASH 发送操作命令时,所使用的第二个地址值,该参数说明中的‘Value’给出了具体的值。不同型号的 NOR FLASH,该值是不同的,用户需要根据 NOR FLASH 规格书的说明来给出该值。以 AMD 公司的 AM29F040B 型号的 NOR FLASH 为例,则‘Value’应取值为 0x2AA。

举例来说,一个使用 AMD 公司 AM29F040B 型号的 NOR FLASH 的目标板,其[FLASH]段的一种可能配置如下所示:

```
[FLASH]
CHIPTYPE          AM29
CHIPSIZE          0X00400000
CHIPWIDTH         8
BUSWIDTH          8
SETUPADDR1       0x555
SETUPADDR2       0x2AA
```

4.4 [NAND]段的定义

该段用来定义目标板上有关 NAND FLASH 的一些配置参数值。当 JDI 对 NAND FLASH 进行操作时,将会使用这些参数,对这些参数值不定义或定义有误,都会导致对 NAND FLASH 的操作错误,所以用户要根据目标板的实际情况来正确地定义这些参数的值。其主要的配置参数及其定义格式有以下几项:

‘BUSWIDTH Value’

该参数用来定义目标板上 NAND FLASH 的数据宽度为<Value>。例如,可以将其定义为 8 或 16,分别表示使用 8 位或 16 位的 NAND FLASH。

‘ROWCYCLES Value’

在对 NAND FLASH 进行访问时,根据 NAND FLASH 本身的要求,需要将其行地址分 2 次或多次写入,该参数即用来定义其需要写入的次数。该表达式中,‘Value’表示需要写入的次数。就目前情况来讲,‘Value’的取值一般为 2 或 3。

‘PAGESIZE Value’

该参数用来定义目标板上 NAND FLASH 每个页的大小。‘Value’为用字节表示的页的大小,例如可以为 512 或 1024 字节等。

‘FORCEERASE Value’

该参数表示在对 NAND FLASH 进行擦除操作时,对其上面的坏块是否执行该操作。‘Value’的取值为 1 或 0,为 1 时表示对坏块执行擦除操作,为 0 表示不执行。

举例来说,本手册所使用的一个实际目标板的配置情况如下所示:

```
[NADN]
BUSWIDTH          8
```

ROWCYCLES	2
PAGESIZE	512
FORCEERASE	0

4.5 [EEPROM]段的定义

在实际的目标板上，对 EEPROM 的访问一般是通过 I2C 总线进行的，这样就需要给出一个设备的地址值。该段中仅有一个用来定义该值的参数，其格式如下所示：

‘ADDRESS Value’

该格式的定义中，‘Value’ 为 EEPROM 设备的地址值。

该段定义的一个示例如下：

```
[EEPROM]
ADDRESS          7
```

4.6 [SDRAM]段的定义

当使用“memtest”命令测试目标板的 SDRAM 时，通过指定[SDRAM]段的配值来完成目标板 SDRAM 的初始化和测试工作。其主要的配置参数及其定义格式有以下几项：

‘BUSWIDTH Value’

该参数用来定义目标板上 SDRAM 的数据宽度为<Value>。例如，可以将其定义为 16 或 32，分别表示使用 16 位或 32 位的 SDRAM。

‘BANKS Value’

该参数用来定义目标板上 SDRAM 的 bank 为<Value>。例如，可以将其定义为 2 或 4，分别表示使用的 SDRAM 为 2 或 4 个 bank。

‘ROWADDR Value’

该参数用来定义目标板上 SDRAM 行地址的宽度为<Value>。例如，可以将其定义为 11-13。

‘COLADDR Value’

该参数用来定义目标板上 SDRAM 列地址的宽度为<Value>。例如，可以将其定义为 8-12。

‘CASLATENCY Value’

该参数用来定义目标板上 SDRAM 的 CAS latency 为<Value>。例如，可以将其定义为 2 或 3。

4.7 [GDB]段的定义

在使用 GDB 进行远程调试时，需要指定目标板的初始化 BINARY 文件名。GDB 在下载被调试代码到目标板前会先执行该文件，以完成必要的初始化工作，如初始化 SDRAM 以及 CPU 时钟等。该段中

仅有一个用来定义该初始化文件的参数，其格式如下所示：

'INITFILE FILENAME'

该格式的定义中，'**FILENAME**' 为 **BINARY** 的文件名。该文件必须位于主机的 **TFTP** 服务目录下。

5 JDI 的硬件说明

本章将对 JDI 的硬件特征作一个简单的说明，以使用户对 JDI 硬件方面的特征有一个了解。其内容主要包含以下几个方面：

- 端口的说明：将对 JDI 上端口的用途及使用方法进行相应的说明
- 指示灯：将对 JDI 上指示灯所指示的信息进行描述
- 跳线保护：将对 JDI 上用于升级的跳线进行说明
- 端口定义：将对 JDI 上端口的信号分布进行说明

5.1 JDI 的端口说明

目前，JDI 上共引出了 5 个接口，分别如下：

USB 接口

该接口用来向 JDI 提供电源，除此之外不作其它的用途。

以太网接口

该接口用来向 JDI 提供网络服务，用于在 JDI 和提供 TFTP 服务的服务器之间提供数据的传输。

RS-232 接口(PS2)

JDI 用该端口来提供串口的输入与输出，以便在串口上对 JDI 进行相应的操作。

JTAG 接口

JDI 用该端口与目标板的 JTAG 口连接，以便对目标板进行相应的操作。

5V 电源接口

该接口用来向 JDI 提供电源，当不用 USB 接口提供电源时，可以用该接口向 JDI 提供 5V 的电源输入。

5.2 JDI 上的指示灯

红色的电源灯亮，表示 JDI 已有电源输入。

中间的绿色灯闪烁，表示 JDI 已进入了工作状态，可以对其进行相应的操作了。

边上的绿色灯闪烁，表示 JDI 正通过 JTAG 口对目标板进行操作。

5.3 JDI 的跳线保护

在 JDI 的主板上，有一根跳线用来保护 JDI 板上的 FirmWare，以防止其受到破坏。在主板上，该跳线的编号 JP1。正常的情况下，该跳线是断开的，这样就不可以更新 JDI 上的 FirmWare。

当需要更新 FirmWare 时，需要将 JDI 盒子打开，将主板上的 JP1 跳线连上，这样就可以用命令来升级 JDI 中 FirmWare 的固件版本了。在升级完成以后，记着将该跳线断开，以保护 JDI 上的 FirmWare。

5.4 JDI 主要端口的信号定义

JDI 的 JTAG 端口信号定义与 MIPS™ 的 EJTAG 标准完全兼容。请参见下图：

