GETTING STARTED WITH BDI & bdiGDB

A CD containing the firmware required to configure this product has been included with the unit. In addition, the hard copy of the User's Manual and original CD is mailed to the REGISTERED USER's address provided to USI at the time of purchase. This CD should be stored in a safe place for future installations. A fee of $20.00 will be charged to replace lost or damaged CDs.

1. INTRODUCTION (v3.9)

Congratulations on your purchase of the BDI Probe. The BDI is a powerful JTAG tool that will prove to be your biggest asset while bringing up your system and target operating system. This document will show you how to setup the configuration file of the BDI to communicate with your target hardware. It will not go into specifics on how to perform board bring up, kernel debugging or flash programming. For additional resources and Knowledgebase articles on how to perform other critical tasks using the BDI, please visit http://www.ultsol.com or contact USI by phone at 978-455-3383.

We will begin by briefly describing the host setup procedure and later, move on to editing the configuration file in order to tailor it to your target hardware.

2. HOST SETUP

To start basic communications with the BDI, you need to setup a few utilities on your host system. Each utility is common and may already be functional on your host.

2.1 TFTP Server

A Windows version of the TFTP server is provided in the zip files that are bundled with the BDI. For users running Linux, we provide an FAQ that describes how to setup the TFTP server at http://www.ultsol.com/pdfs/KB00086.pdf. Users running MAC O/S, Solaris or Unix will need to refer to documentation available from alternative resources.

2.2 Serial Port

A serial port is needed to program the initial firmware into the BDI. If you purchased the USI Cable Kit # USI07-20301K or USI07-60301K, you should have received a USB to serial (RS232) converter cable in addition to a standard RS232 to RS232 cable (Ref # 90101). Either one may be used. Keep in mind that some laptops are no longer providing RS232 connectors so in these instances; the converter cable will be required.

Note: A Windows driver for the USB to RS-232 converter cable can be downloaded at no charge from: http://www.ultsol.com/index.php/support/downloads.

If you are running a version of Linux or any other host OS that doesn't recognize the USB to RS-232 cable upon installation, please contact Ultimate Solution's to obtain assistance.

2.3 Telnet Client

The telnet client is built into Windows and Linux. From a DOS or Linux Prompt you can type in "[prompt] telnet <IP address>" and it will connect to the <IP address> of your choosing.

2.4 Installation of the BDI Firmware

Proceed to chapter 2 of the BDI User's Manual, "Installation of the BDI2000 or BDI3000". Start by unzipping the firmware into a folder on your hard drive. **It is important that all the files get unzipped and the directory structure stay preserved.** Connect the BDI to the serial port and launch the bdisetup utility to flash the BDI with the desired target firmware. Here you will specify a host IP address. This points to the TFTP server where the BDI will pull its initial configuration file. Make a note of the IP address you assign, as you will need it again later. When operating the BDI from a Windows host, you will need to specify the full path and file name for the location of the configuration file. However, when operating the BDI from a Linux host you will only need to specify the path and file name relative to the tftpboot directory that is set as an environment variable.

NOTE: The path name and file name must not contain any "white space" characters.

3. SETTING UP THE CONFIGURATION FILE

There are several sections in the configuration file. We will tackle them one at a time so that you can create a configuration file that is customized for your board. Start out by making a copy of an existing configuration file that you can use as a template. We recommend you use a configuration file that is most similar to your target processor.

3.1 [INIT]

This is one of the most important sections of the configuration file and also the most difficult. If you want to use the BDI to load programs to SDRAM or program flash, you will need to complete this section so that your board is initialized. Making a configuration file that initializes your board is discussed in a white paper entitled, "Getting Started With Configuration Files" which you can acquire from http://www.ultsol.com/pdfs/bdi_configuration_guide.pdf. For the purpose of this discussion, if you have a boot loader on your board already running, you may leave the INIT section empty. If you do not have a boot loader installed you may start out by disabling the watchdog timer on your processor if one exists. Some processors need to have the Page Tables set up before they can access memory so be familiar with the basic startup needs of your CPU. Use an existing configuration file as a template on how to initialize your processor. The rest of the [INIT] section should be left empty.

3.2 [TARGET]

In this section you will define the characteristics of your target processor. It is critical that you get this right or communication with your processor will not function correctly in most cases. There are several parameters to define. The most important are listed below.

CPUTYPE: This is the main field to define. It lets the BDI know which processor it is communicating with. Please review section 3.2.2 in the BDI User's Manual for available processor types.

STARTUP: Use STARTUP mode RUN if you have a boot loader. This bypasses the [INIT] section and starts the processor. Use STARTUP mode RESET (default mode) if you do not have a boot loader and want the BDI to perform some initialization. You may need to setup more values such as vector catching, system delays or the JTAG scan chain to make your system stable. Review section 3.2.2 in the BDI User's Manual for more details. Use an existing configuration file as a template to determine what needs to be included in the [TARGET] section of your processor.

3.3 [HOST]

The host section is used to tell the BDI where to obtain and load program files and also define what format they will be in.

NOTE: The path name and file name must not contain any “white space” characters.

IP: This is the IP Address of the TFTP server where the files to be programmed or used by the BDI are located. It can be different from your host IP address defined in the configuration utility. If your configuration host and TFTP host are the same, this field will be the same IP address you set during configuration of the BDI.

FILE: This is the file to load into SDRAM. Specify a full path for Windows and a relative path from the tftpboot directory in Linux.

FORMAT: Format of the file to be loaded into SDRAM. Can be type BIN, ELF or AOUT.

START: Enter address in SDRAM where to load the file. Format is 0x1234ABCD for hex or just type in the number for decimal. This parameter is only needed for BIN format files since they do not have offsets built into them.

PROMPT: Specify a BDI prompt that will be displayed in the telnet window, such as “bdi>”.

DUMP: Define a file here if you intend to dump memory locations into a file. The file must already exist and have the right permissions before you dump to it. Define the full path of the file for Windows or a relative path from the tftpboot directory for Linux.

NOTE: The path name and file name must not contain any “white space” characters.

3.4 [FLASH]

This section defines the flash connection on your device. It is not needed for the basic task of getting communication going with the BDI. For details on how to setup this section, see section 3.2.4 of the BDI manual. If you do not have a boot loader or a proper [INIT] section that sets up flash, you will not be able to perform flash programming anyway so it is best that you skip this section for the moment. Just keep in mind that when you want to burn flash you will need to fill out this section and specify a file to burn. Note that you will need to define the file to program into flash and the file to load into SDRAM separately.

CHIPTYPE: This parameter defines the programming algorithm to use and tells the BDI the bit-width of one flash chip. There are various algorithms available for all the different flavors of flash devices. Please see the BDI User’s Manual for available algorithms and choose one that suits your flash type.

CHIPSIZE: Define the size of the entire flash on the board.

BUSWIDTH: Size of the flash bus, normally when you have 2 flash chips on board they are connected in parallel and this parameter is used to tell the BDI how many chips in parallel are connected.

ERASE: This optional parameter is useful if you want to erase certain sectors in your flash by default. Sectors defined in the configuration file will get erased when you perform an “erase” command from the BDI.

FILE: This is the file to program into flash. Specify a full path for Windows and a relative path from the tftpboot directory in Linux.

NOTE: The path name and file name must not contain any “white space” characters. ***FORMAT:***

Format of the file to be programmed into Flash. Can be type BIN, ELF, or AOUT.

3.5 [REGS]

In this section you can define a file that will extend the list of registers the BDI knows by name. If you use a register definition file, be sure that you define the base address for the registers. See section 3.2.5 in the BDI User’s Manual for details. You can use one of the default files that come with the firmware zip file if it matches your target processor. For the moment you can leave this section blank.

4. TESTING TARGET COMMUNICATION

Once all the important fields of the configuration file are setup it is time to test communications between the BDI and the target board.

4.1 Ethernet Connection

Start out by connecting the BDI to the network through the Ethernet cable. If you purchased the cable kit # USI07-20301K or USI07-60301K, a crossover adaptor is provided that can be used to connect directly to your Windows or Linux host. Now you may telnet to the BDI from a prompt, "[prompt]telnet<IP address>". Be sure to specify the <IP address> you assigned to the BDI during setup with the configuration utility. You should now see the BDI prompt. It will be telling you that the target has no power. If you see any other messages, something has gone wrong. Go back and check the configuration file setup again. Visit: http://www.ultsol.com/pdfs/bdi_configuration_guide.pdf. to see if your setup is correct.

4.2 Target Connection

With the target powered off, connect the BDI to the Target using the supplied target cable and power up the target board. The BDI should go through its initialization sequence. If you encounter an error, check the Knowledgebase articles at http://www.ultsol.com/index.php/support/knowledge-base for possible solutions. Now you are ready to begin poking around in your target. Type in “bdi>halt” to stop the processor.

4.3 BDI Commands

The BDI understands both hexadecimal and decimal numbers. Hex numbers must be preceded with a 0x, e.g. 0x1234abcd. Decimal numbers can be typed in as 12345678. It is always best to use hex numbers because you can match the bit length of your data busses easily. Below is a list of common commands used with the BDI.

RESET: Resets the target board. This command does not re-read the configuration file and the telnet session is not lost. Type “bdi>reset run” to let the target run after reset or “bdi>reset halt” to stop the target after reset.

BOOT: Power cycles the BDI. The configuration file is re-read, the telnet session is lost and you must reconnect.

QUIT: Disconnect from the telnet session.

HALT: This command stops the processor and gives control of the processor back to the BDI. The BDI cannot perform any operations on the processor while it is running.

INFO: This command shows the current state of the processor. It also provides information about the Program Counter (PC) and a few other core registers.

GO: This command starts executing the BDI from the current PC. If you performed a "load" command then the PC gets moved to the beginning of the program execution. You can start executing from a different PC by specifying the address, "bdi>go 0x00000200". If you want to change the PC without executing, you will manually have to change the PC register thru the "RM" command.

MD: Memory Dump 32-bit. This command displays target memory from the specified address in 32-bit format. "bdi>md 0x00001000" shows the memory at 0x00001000. Subsequent md commands without an address will display the next region of memory. Use mdb for 8-bit, mdh for 16-bit and mdd for 64-bit accesses.

MM: Memory Modify 32-bit. Use this command to modify a target memory location 32-bits at a time. "bdi>mm 0x00001000 0x1234abcd", write hex 1234abcd to hex address 00001000. You can also specify a count to write the same value in multiple times in 32-bit increments, "bdi>mm 0x00001000 0x1234abcd 31" will write 1234abcd to 00001000 and repeat 31 times. Use mmb for 8-bit, mmh for 16-bit and mmd for 64-bit write operations.

MT: Use this command to perform a memory test. "bdi mt <addr> <count>[<loop>]"
The address is where you would like to start testing.
Count defines the address range: address to address+count.
Loop value is between 1 and 14 and it is optional.

Example: MT 0x40000 0xff 14

In this example your memory test starts at 0x40000 for 0xff addresses and 14 loop values. The command will report back if the test passed of failed for each loop. The BDI writes a pattern to the defined memory range and reads it back. This is done up to 14 times with a different (shifted) patter for every iteration. Note – This command is only available on certain CPUs.

RD: Register Dump. Displays general purpose registers. You can use this command to display all the general purpose registers or a specific one. "bdi>rd msr" shows the msr register. "bdi>rd" shows all the general purpose registers. Note that registers defined in the register definition file are not part of the "show-all" list, but can be called out by name.

RDSPR: Register Dump Special Purpose Registers. Displays special purpose registers. You can use this command to display all the general special registers or a specific one. "bdi>rdspr lr", shows the lr register. "bdi>rdspr" shows all the general purpose registers. Note that registers defined in the register definition file are not part of the "show-all" list, but can be called out by name.

RDUMP: Dumps all the registers defined in the register definition file into a defined file. By default the file used is the one defined by the DUMP field in the [HOST] section of the configuration file, but you can override that by specifying a file name of your own. Please make sure the file exists and that you have the correct permissions to edit it. Type "bdi>rdump" to use the default filename or type "bdi>rdump <file path>\<file name>" to define a custom location and file name.

LOAD: Loads the file defined in the [HOST] section into SDRAM. You can override the defaults by specifying your own file and target address. Type "bdi>load" to use the defaults or type "bdi>load 0x00001000 <file path>\<file name> <format>" to load <file path>\<file name> of type <format> into target SDRAM at 0x00001000. Also, you can type "bdi>load <file path>\<file name> <format>", load a <format> file into SDRAM with default loading offsets. The TFTP server used is defined by the IP field in the [HOST] section and cannot be changed.

Remember that the <format> can be type BIN, ELF or AOUT and that <file path> is the full path on a Windows host and a relative path to tftpboot on a Linux host.

NOTE: The path name and file name must not contain any "white space" characters.

PROG: Programs the file defined in the [FLASH] section into flash. You can override the defaults by specifying your own file and target address. Type "bdi>prog" to use the defaults or type "bdi>prog 0xF0001000 <file path>\<file name> <format>" to program <file path>\<file name> of type <format> into target flash at 0xF0001000. Also, you can type "bdi> prog <file path>\<file name> <format>" to program a <format> file into flash with default loading offsets. The prog command uses the flash algorithm to enable writing to flash, hence you cannot use the load command to write to flash and vise versa. The TFTP server used is defined by the IP field in the [HOST] section and cannot be changed. Remember that the <format> can be type BIN, ELF or AOUT and that <file path> is the full path on a Windows host and a relative path to tftpboot on a Linux host.

NOTE: The path name and file name must not contain any "white space" characters.

ERASE: Erase flash. By default uses the ERASE list defined in the [FLASH] section. Use "bdi>erase" to use the defaults. The erase command must be used at the beginning of a sector. In some flash devices the sector sizes can vary, hence it is best to define the sectors to erase with the ERASE list in the [FLASH] section. Type "bdi>erase" to use the ERASE list in the [FLASH] section or type "bdi>erase 0xf0000000" to erase the sector starting at 0xf0000000. Also, you can type "bdi>erase 0xf0000000 0x20000 31" to erase multiple sectors of size 0x20000 starting at 0xf0000000 and repeat for 31 more sectors.

BI: Break Instruction. This command is used to set hardware breakpoints in the BDI. Type "bdi>bi 0x00001020" to set a breakpoint at 0x00001020. You must then start executing by using the "bdi>go" command to hit the breakpoint.

CI: Clear all breakpoints. Use this command to clear hardware breakpoints after you hit them so that you don't run out of hardware breakpoints. Type "bdi> ci" to use this command.

The BDI supports many more commands. Type "bdi>help" for a complete list.

5. SUMMARY

Now that you have setup the BDI and tested basic functionality, it is time to do something useful with your setup. The BDI is a powerful and versatile programming and debugging tool. For guides on how to use the BDI for programming or debugging, please visit Ultimate Solutions white papers online at http://www.ultsol.com/index.php/support/white-paper.

Authored by: Fahd Abidi
Field Application Engineer with Ultimate Solutions, Inc.