» User Guide «

NetBootLoader

Composite User Guide for All CPU Boards Which Utilize the NetBootLoader Software

Doc. ID: 1022-8317, Rev. 2.0
November 30, 2010

Revision History

Publication Title:	NetBootLoader User Guide	
Doc. ID:	1022-8317	
Rev.	Brief Description of Changes	Date of Issue
1.0	Initial issue	19-Sep-2008
2.0	“JUMP” Command added; new functionality added for the DHCP and TFTP commands	30-Nov-2010

Imprint

Kontron Modular Computers GmbH may be contacted via the following:

MAILING ADDRESS

Kontron Modular Computers GmbH

Sudetenstraße 7

D - 87600 Kaufbeuren Germany

TELEPHONE AND E-MAIL

+49 (0) 800-SALESKONTRON

sales@kontron.com

For information about other Kontron products, please visit our Internet web site: **www.kontron.com**

Disclaimer

All data is for information purposes only and not guaranteed for legal purposes. Information has been carefully checked and is believed to be accurate; however, no responsibility is assumed for inaccuracies. Kontron and the Kontron logo and all other trademarks or registered trademarks are the property of their respective owners and are recognized. Specifications are subject to change without notice.

Table of Contents

List of Tables

This page has been intentionally left blank.

Proprietary Note

This document contains information proprietary to Kontron. It may not be copied or transmitted by any means, disclosed to others, or stored in any retrieval system or media without the prior written consent of Kontron or one of its authorized agents.

The information contained in this document is, to the best of our knowledge, entirely correct. However, Kontron cannot accept liability for any inaccuracies or the consequences thereof, or for any liability arising from the use or application of any circuit, product, or example shown in this document.

Kontron reserves the right to change, modify, or improve this document or the product described herein, as seen fit by Kontron without further notice.

Trademarks

This document may include names, company logos and trademarks, which are registered trademarks and, therefore, proprietary to their respective owners.

This page has been intentionally left blank.

1. Introduction

The NetBootLoader is a proprietary software utility for Kontron's families of PowerPC products – E²Brains, AMC CPUs, CPCI CPUs, VME CPUs – which is designed to initialize and configure these modules for operation before turning control over to either an application or to an operator. This software also provides the capability to monitor and control the operation of the NetBootLoader itself, display system status information, to program executable code and data to the NBL Flash memory, and to load and start application software.

Depending on the product type, there are either one or two NOR Flash memories (NBL-Flash0, NBL-Flash1) available which are used to store the NetBootLoader itself as well as providing memory space for bootable images or other data. Where two NOR-Flashes are available, these memories are identical and are redundant of one another. Only one of these Flash memories can be in use at any given time. The other is redundantly available, and, if required, it can be swapped with the currently used memory.

All CPU modules are delivered with the NetBootLoader (NBL) software already programmed into their NOR Flash memory chip(s) (NBL Flash).

1.1 Basic Operation

Upon power on or a system reset, the NetBootLoader is started. The CPU module is configured for operation and control is passed either to an application or an operator. In the event a valid image has been programmed into the NBL Flash memory and no operator or SCRIPT command intervention takes place, a specified image is copied into the CPU module's data buffer and control is passed to the image which can either be an operating system or a customized application. If the NetBootLoader does not find a valid image or operator intervention has occurred, control is passed to the operator. The operator now has control to determine the system status, make configuration changes, read or program the NBL Flash memory, invoke the RUN command, or to restart or power down the system.

The operator command interfacing with the NetBootLoader is accomplished either through a serial or Ethernet port. During the boot operation a command interpreter is started which allows the operator to input commands to the NetBootLoader. Prior to interfacing using an Ethernet port, the port must be configured either via the serial port or a DHCP/BOOTP server.

1.2 NetBootLoader Interfacing

There are three possibilities to interface with the NetBootLoader:

- Via the primary serial interface
- Via the secondary serial interface
- Via an Ethernet port interface

During the boot wait time, which is a time frame provided by the NetBootLoader during the boot process for operator intervention, interfacing is done directly using the primary serial interface or remotely by either performing a telnet login or using the Serial Over LAN feature via an Ethernet port. If the operator does not intervene, the boot operation is continued after the boot wait time has elapsed.

The secondary serial port is used to provide the NetBootLoader with the ability to access Motorola S-Records for programming an image to NBL Flash memory. A command interpreter is not available for this interface.

This page has been intentionally left blank.

2. NetBootLoader Operation

The basic function of the NetBootLoader is to configure the CPU module for operation and then to load and start an operating system. In addition, it provides functionality for controlling the operation of the NetBootLoader itself, for programming images or data to the NBL-Flash(es) and NAND Flash, remote operation access, accessing TFTP or FTP servers, and acquiring Motorola S-Record images.

Note ...

In the following sections the command title (CMD TITLE) is expressed in capital letters and is not the same as the syntax of the command. The command syntax is always written using small letters.

2.1 NetBootLoader Control

The NetBootLoader provides various functions for controlling the operation of the NetBootLoader itself as well as the setting of operational parameters. The following table provides an overview of available NetBootLoader control functions.

Table 1: NetBootLoader Control Commands

CMD TITLE	ALIAS	FUNCTION	REMARKS
ABORT	-	Terminate boot wait	
ASSIGN	-	Assign or display device assignments	For CPUs with multicore processors
ASSIGNMEM	-	Assign or display memory assignments	For CPUs with multicore processors
BL	Bootline	Set or display bootline	Bootline stored in RAM
BW	Boot Wait	Set or display boot wait time	
CBL	Change Bootline	Set or display a bootline	Applies to a specific image or all images in NBL Flash
DHCP	-	Dynamically set Ethernet port parameters	Requires that a DHCP or BOOTP server be available in the same network as the NetBootLoader
EXE	Execute	Execute config file	File source NAND Flash
FDT	-	Enable creation of FDT	
NET	-	Manually set Ethernet port parameters	Must be set before attempting to use an Ethernet port; see also the DHCP command
PF	Port Format	Set serial port parameters	Used for the primary and secondary ports
RS	Reset	Resets system	
RUN	-	(Loads and) starts image	If image is not in the data buffer, it is first loaded then started

Table 1: NetBootLoader Control Commands

CMD TITLE	ALIAS	FUNCTION	REMARKS
SCRIPT	-	NBL command scripting	Contents are executed only during boot up
SETCORE	-	Set or display core number	For CPUs with multicore processors
SPEED	-	Set or display fast booting	
SQ	Boot Sequence	Set or display boot sequence	Defines the image boot order
SWAP	-	Set or display NBL Flash usage	For CPUs with two NBL Flashes

2.2 Network Accessing

To support application development and operational requirements for various boot strategies, the NetBootLoader provides several functions for gaining access to network services. These functions include: access to DHCP/BOOTP servers, accessing TFTP servers, accessing FTP servers, and having access to the target CPU from remote locations.

To achieve this, certain network parameters must first be configured. This can be done manually via a terminal (primary serial port) or dynamically via the network (an Ethernet port). If the required parameters are known, they can be set with the NET command. Otherwise, the command DHCP makes it possible to download such parameters and to configure an Ethernet port for network operation. Once the Ethernet port is configured, the commands TFTP and FTP are available to download bootable images and other files as required. For remote access to the CPU module, either telnet or the Serial Over LAN feature may be used.

The following table provides an overview of available TFTP/FTP server and remote accessing functions.

Table 2: Network Accessing Commands

CMD TITLE	ALIAS	FUNCTION	REMARKS
BYE	-	Terminate session with FTP server	
CD	Change Directory	Change FTP server directory	
GET	-	Download a file from FTP server	Data buffer is target.
LOGIN	-	Login to FTP server	
LOGOUT	-	Terminate telnet session	
LS	List Directory	List FTP server directory	Lists contents of directory.
PASSWD	Password	Set telnet password	Must be set before attempting telnet login
PING	-	Verify network status	
PUT	-	Upload a file to FTP server	Data buffer is source.
PWD	Print Working Directory	Display current FTP server directory	Lists name of directory
TFTP	-	Download a file from TFTP server	Data buffer is target.

2.2.1 DHCP/BOOTP Server Access

Use of this access method requires the availability of either a DHCP or BOOTP server in the same network as the NetBootLoader. The DHCP command causes the NetBootLoader to first attempt to establish contact with a DHCP server. If contact is not achieved, it then tries to contact a BOOTP server. When contact is established, parameters required by the NetBootLoader are provided accordingly and the Ethernet port is configured and then made available for normal operation.

In the event the NetBootLoader is reset or cold started the configuration parameters set by the above method are lost. Only if the parameters have been set by the NET command are they still available.

Some products require that prior to using the DHCP command, the IP address of the CPU module be set to 255.255.255.255 using the NET command.

2.2.2 TFTP/FTP Server Access

The NetBootLoader provides various functions for interfacing with either a TFTP or FTP server.

The TFTP server access is a simple method of acquiring a userimage from a remote source. Its primary use is to download a single executable image from a given source. For example, once an image has been created it would be possible to store it at a remote location where it then would be available for remote booting of an CPU module via an Ethernet port.

The TFTP/FTP server commands provide various functions consistent with interfacing with such a server.

2.2.3 Remote Access

All products supported by the NetBootLoader may be remotely accessed via telnet. In addition to telnet, the AM4100 and AM4101 also support Serial Over LAN for remote access to the NetBootLoader. This requires appropriate software running at the remote location, for example, “ipmitool”.

In the case of Serial Over LAN, the NetBootLoader supports RMCP only. RMCP+ is not supported.

2.3 NBL Flash Operation

The NetBootLoader provides various functions for performing operations with the NBL Flash memory. The following table provides an overview of the available NBL Flash memory operation functions.

Table 3: NBL Flash Operation Commands

CMD TITLE	ALIAS	FUNCTION	REMARKS
CHECK	-	Image validation; displays information for each image	Verifies validity of user image programmed to NBL Flash memory
CLONE	-	Program NetBootLoader to NBL Flash memory	Uses data buffer or redundant NBL Flash as source
LF	Load Flash	Program image to NBL Flash memory	Uses data buffer as source

Table 3: NBL Flash Operation Commands

CMD TITLE	ALIAS	FUNCTION	REMARKS
SF	Store Flash	Reads NBL Flash memory to data buffer	Uses data buffer as target
SL	SLoad	Download Motorola S-Records	Uses data buffer as target

2.4 NAND Flash Operation

The NetBootLoader provides various functions for performing operations with the NAND Flash memory. The following table provides an overview of the available NAND Flash memory operation functions.

Table 4: NAND Flash Operation Commands

CMD TITLE	ALIAS	FUNCTION	REMARKS
CD, CDCF	Change Directory	Change NAND Flash directory	Use CDCF when logged into FTP server
ECAT	-	Create or view boot script config files on the NAND Flash	Single line files only
FDISK	-	Create or remove primary partitions on the NAND Flash	
JUMP	-	Jump to the specified script in the NAND Flash (when the current script has been executed)	Standard boot process only; if encountered during the alternate boot process (netboot) is ignored
LOAD	-	Load an operating system image and optionally a RAMdisk from NAND Flash	Uses data buffer as target
LS, LSCF	List Directory	List NAND Flash directory	Lists contents of directory.
MD5	-	Perform md5 checksum verification	Verifies data buffer contents
MKDIR, MKDIRCF	Make Directory	Create a directory on the NAND Flash	Use MKDIRCF when logged into FTP server
MKFS	Make Filesystem	Create a filesystem for the partition specified on the NAND Flash	
MV, MVCF	-	Rename a file on the NAND Flash	Use MVCF when logged into FTP server
PWD, PWDCF	-	Display the current NAND Flash directory path	Use PWDCF when logged into FTP server
RM, RMCF	-	Remove a file on the NAND Flash	Use RMCF when logged into FTP server
RMDIR, RMDIRCF	-	Remove a directory on the NAND Flash	Use RMDIRCF when logged into FTP server
SAVE	-	Store contents of the data buffer to a file in the NAND Flash	

2.5 System Status Monitoring

The NetBootLoader provides various functions for monitoring the overall status of the system during the operation of the NetBootLoader. The following table provides an overview of available system status monitoring functions.

Table 5: System Status Monitoring Commands

CMD TITLE	ALIAS	FUNCTION	REMARKS
FDT_SHOW	-	Display FDT	Flattened Device Tree
FRU	-	Display FRU info	For AMC CPUs
HELP or ?	-	Display online HELP pages	
INFO	-	Display system information	
IPMI	-	Display IPMI information	
MD	Memory Display	Display memory contents	Applies to all memory
PCI	-	Display PCI device information	

2.6 Operating the NetBootLoader

2.6.1 Initial Setup

The CPU module is delivered with the NetBootLoader already installed in the NBL Flash memory and is ready for operation. However, in order for the CPU module to be used in a system, application software must be made available for use. This may be accomplished by programming such an image either to the NBL Flash memory where the NetBootLoader is located or to the NAND Flash when available.

Upon initial power up, the NetBootLoader is started automatically. As soon as the NetBootLoader has completed initialization of the CPU module, it initiates a command interpreter which the operator can access either via the primary serial or an Ethernet interface. As long as the command interpreter is available, the operator has the opportunity to intervene for programming an image.

Prior to programming an image it may be necessary to configure the NetBootLoader or perform other functions depending on the user’s application development environment or application requirements. Once this has been accomplished and the image has been programmed, the NetBootLoader is ready for operation.

2.6.2 Accessing the NetBootLoader

Initial access to the NetBootLoader can only be achieved via the primary serial interface. Use of this interface requires either a terminal or a terminal emulator. In addition, the command interpreter is required.

Availability of the command interpreter depends on the system status. If a valid image is not available, the command interpreter is active as long as the operator requires it. If a valid image is available, the command interpreter is only active for the duration of the boot wait time. If the operator requires the command interpreter for a longer time he must terminate the boot operation before the boot wait time has elapsed.

Prior to using the network interface for remote access, Ethernet port parameters must be set and this can only be accomplished initially via the primary serial interface.

The operator must either manually set the parameters using the NET command or dynamically via the DHCP command. Some products require that prior to using the DHCP command, the IP address of the CPU module be set to 255.255.255.255 using the NET command. If the DHCP command returns a failure message, then the IP address should be set to 255.255.255.255, and the DHCP command reissued.

Once valid Ethernet port parameters have been set, the NetBootLoader may be remotely accessed either via telnet (all CPU modules) or Serial Over LAN (AM4100 and AM4101 only). When the command interpreter is available during the bootwait time, a successful telnet login or the initiation of a Serial Over LAN session will automatically terminate the boot operation, and a command prompt is sent to the remote interface.

Once the operator has control of the NetBootLoader, he may perform any required action. To continue with the operation of the CPU module, the system may either be cold started, the RS (reset) command issued, or the RUN command issued.

2.6.3 System Booting

Assuming a fully configured system, beginning with power on (to the payload), the NetBootLoader is started. It then preforms basic configuration of the system, and then waits shortly for operator intervention. After this wait period with no intervention, the boot process continues with the loading of an operating system either from NBL Flash or NAND Flash to the data buffer and then starting it. Once the operating system is started the NetBootLoader is no longer available. The OS and application now have control.

While this may seem simple, control of the boot process can be very complex, depending on the application requirements and the configuration of the system.

The NetBootLoader offers a very comprehensive set of commands not only for supporting application development but to provide a secure and highly available system.

For example, validity control of images is supported for all images, either through CRC checks or an md5 checksum verification. Selective loading and starting of images is provided by appropriate command options for the LOAD, MD5, and RUN commands.

Passing operational parameters to an operating system is also supported by various commands or configuration files, e.g. BL, CBL, FDT, nbl0..9.cfg, net.cfg, etc.

In addition, with modules that support IPMI (for example the AM4100 or AM4101), the boot process can also be controlled via IPMI and the onboard MMC (Module Management Controller).

The following sections present the two primary methods of controlling the boot process.

2.6.3.1 Standard Boot Process

The standard boot process foresees the use of predefined NetBootLoader functionality without external intervention at the time the system is to be booted. This is accomplished through the use of appropriate configuration files, pre-setting of the SCRIPT command, pre-setting of the SQ (Sequence) command, and pre-programming of bootable images to the NOR and/or NAND Flashes.

In this case, the boot process is controlled by the contents of the above mentioned entities as follows:

- With modules that provide NAND Flash capability, the NetBootLoader first searches for a FAT partition that is marked bootable (only one partition may be so marked)
- If such a partition is found, it then searches for “nbl<n>.cfg” configuration files beginning with the file “nbl0.cfg”. When a file is found, the NetBootLoader executes it.
- If the file execution does not result in an image being loaded and started or the boot process being aborted, the NetBootLoader searches for the next available configuration file, and, if one is found, it is executed. This continues until either all “nbl<n>.cfg” files have been executed, the boot process is aborted, or an image is loaded and started.
- In the event a bootable FAT partition is not found or all available “nbl<n>.cfg” files have been executed without either aborting the boot process or loading and starting an image, the NetBootLoader searches for any script previously specified by the SCRIPT command. If found, it is also executed.
- If a script is not found, or if the script execution does not result in an image being loaded and started or in the boot process being aborted, the NetBootLoader continues with the processing of image(s) stored in NOR Flash (according to the order defined by the SQ command if specified, or 0, 1, 2, 3 if not specified). The first valid (CRC check passed) image found is loaded and started. If a valid image is not found, the boot process is aborted and a command interpreter is made available.

With the introduction of NAND Flash, the boot process may indirectly be influenced via an operating system. After an initial boot up and starting of an operating system, the contents of files in the NAND Flash may be altered via OS functions, so that a very customized boot process may be specified without having to use NetBootLoader file editing functions. With the next boot up, the “new” boot process can then be executed.

2.6.3.2 Alternate Boot Process

For modules which support IPMI and NAND Flash, there is an alternate boot capability (known as netboot) which can be invoked from a remote location using IPMI commands. This can be used for situations where the standard boot process fails leaving the module payload inoperable or a non-standard application configuration is required for whatever reason. This is achieved by setting the boot order via IPMI and then having the MMC reboot the payload.

For example, using the “ipmitool” software the following command line can be invoked to set a register that is evaluated by the NetBootLoader on the next payload restart:

```
ipmitool -l 0x00 0x3E 0x20 0x9D 0x00 or 0x01
```

where 0x00 is for standard boot process and 0x01 for the alternate boot process. This value must be explicitly set (reset) as it is stored in non-volatile memory.

At reboot start, the NetBootLoader first searches for a FAT partition on the NAND Flash that is marked bootable. If such a partition is found, it then searches for a configuration file named “net.cfg”. If found, the NetBootLoader executes it.

In the event a bootable FAT partition is not found or if a “net.cfg” file is not found or its execution does not result in an image being loaded and started or the boot process being aborted, the NetBootLoader executes the following command sequence: “dhcp; tftp; run”. If this command sequence is not successfully executed, the boot process is aborted and a command interpreter is made available for operator inputs.

2.6.3.3 Dual Core Operation (SMP/ASMP)

The NetBootLoader also supports the booting of modules with dual cores (for example, the AM4100 and AM4101). Modules with dual cores can be operated using only one core, using both cores with the same operating system (SMP), or using both cores with different operating systems (ASMP).

Single core usage is principally the same as with modules with only one core. Dual core operation is a function of the application whereby two possible modes of operation are available: symmetric multiprocessing (SMP) and asymmetric multiprocessing (ASMP).

In the case of SMP operation, the NetBootLoader itself runs using core 0, loads and starts the operating system for core 0. Once the operating system is started, the OS starts the operation of the second core (core 1).

For ASMP operation, the NetBootLoader itself runs using core 0, loads and starts different operating systems for core 0 and core 1.

Essentially the standard boot process for both SMP and ASMP is the same with the exception that when using ASMP mode, the NOR Flash image sequence is not supported. In this case, the RUN command must explicitly specify the NOR Flash image to be used.

The alternate boot process is different for ASMP. In this case, only the “net.cfg” configuration file is executed. If this does not result in images being loaded and started the boot process is aborted and a command interpreter is made available.

2.7 Command Specifics

The following sections present either command specific or more detailed information on commands as well as operational aspects of the NetBootLoader with regards to particular commands.

2.7.1 ASSIGN

The use of this command requires a CPU module with two cores configured for asymmetric multiprocessing (ASMP) operation. It permits the assignment of devices to one or the other of the cores whereby a device may only be assigned to one core at a time.

A CPU module with two cores configured for symmetric multiprocessing (SMP) operation has automatically all available devices assigned to core 0.

2.7.2 BL

When specified, this command stores a bootline in RAM for use of an image in the data buffer at the time the image is started with the RUN command.

When ASMP operation is selected, this command is core sensitive, meaning that it can specify the bootline to be used with the image for either core0 or core1 depending on the core selected by the last issue of SETCORE. A bootline for each core may be specified.

Refer to the SETCORE command for further information.

When RUN is executed, BL is used in addition to the common bootline from the NBL EEPROM which was set with “cbl c <commonbootline>”.

The sum of all bootlines (bl and cbl) for one core must not exceed 256 characters. To verify the status of bootlines, use the FDT_SHOW command.

When no bootline is specified, the bootline 0 from the NBL EEPROM (cbl 0) is used as the bootline.

2.7.3 BW

This command is used to display or set the actual boot wait time used by the NetBootLoader to delay the boot operation before proceeding with the loading and starting of an application. If this time is set too short it may not be possible to gain access to the NetBootLoader.

The boot wait time value is stored in the boot section of the serial EEPROM. This section is validated with a CRC code to avoid the setting of random parameters.

Note ...

If the CRC of the boot section is not valid, changing the boot wait time will have no effect because the BW command does not validate an invalid CRC. In this case, a default timing of 5 seconds is always used.

To validate an invalid CRC, an operating system utility must be used, or, alternatively, the “-f” option of the BW command must be issued.

WARNING!

Using the “bw -f” command to validate invalid entries may adversely impact the operation of the operating system.

2.7.4 CBL

This command is used to set or display a bootline associated with a particular kernel image or which is common to all images.

When an image is programmed to NBL Flash memory, it is assigned an ID number (0, 1, 2, or 3). This number is used to identify which image is to be addressed by the command CBL.

In addition, a bootline common to all images may also be defined using the “c” parameter.

The sum of all bootlines (bl and cbl) for one image must not exceed 256 characters. To verify the status of bootlines, use the FDT_SHOW command.

2.7.5 CLONE

Products which have only one NOR Flash require access to additional Flash memory to support the CLONE command without the “-n” option. E²Brains require that their carrier boards provide this functionality if it is to be supported. Other products may or may not provide this capability with onboard Flash. Refer to the product User Guide for further information or contact Kontron for assistance.

2.7.6 DHCP

This command is used to obtain automatically networking parameters from either a DHCP or BOOTP server for an Ethernet port. Its use requires the availability of one or the other of these servers to function.

Initially the CPU module does not have a valid Ethernet interface configuration, and, therefore, this interface is inoperable. The initial configuration must be done either manually from a serial interface using the command NET, or, if a DHCP or BOOTP server is available, it can be done automatically by the DHCP command.

2.7.7 FDT

This command is used to enable the creation of a flattened device tree.

The flattened device tree provides information required by certain operating systems for operation of system devices. In the case of the AM4100 and AM4101, the flattened device tree is always generated.

2.7.8 NET

This command is used to set or display the parameters for the configuration of the specified Ethernet port of the NetBootLoader. The Ethernet interfaces are only available after these settings have been made.

2.7.9 PASSWD

This command is used to set the password used by the NetBootLoader for the operation of the telnet interface. No password is required for access from the primary serial interface or for Serial Over LAN

2.7.10 PF

This command is used to set the port parameters for the primary and secondary serial interfaces only for the current operator session. The next system restart will cause these settings to revert to the default settings of: 9600 Baud, 8 bits per character, 1 stop bit, and no parity. This is done to preclude a system lockout when restarting due to incompatible settings.

2.7.11 SCRIPT

This command permits the automatic invoking of NetBootLoader commands during boot up. The operator issues this command with appropriate options and then restarts the system. During the boot operation at boot wait time expiration, the script commands will be executed.

Use of this command permits, for example, remote booting from a TFTP server.

2.7.12 SQ

This command is used to set or display the order in which images are to be booted.

The NetBootLoader provides the capability to program up to four images into the NBL Flash memory. With this command the operator can define the order in which images may be used when the system is booted. This provides operational flexibility as well as the possibility for the system to compensate for a defective image.

For example, in the event the first image specified is defect, the NetBootLoader will attempt to load the next image specified. This is continued until either a valid image is loaded or no further image is available.

If no valid image is found, the NetBootLoader remains available for inputs.

2.7.13 telnet Login

A telnet login to the NetBootLoader is only possible during the boot wait time or when the boot operation has been terminated either by operator action or automatically by the NetBootLoader. In either case, the Ethernet port parameters and the telnet password must be set prior to attempting the telnet login.

To effect a telnet login the operator performs the standard telnet login procedure. The NetBootLoader responds by requesting a login password. The operator then enters a password. If the password is valid, the operator can now access the NetBootLoader. If the password is invalid, the telnet login procedure is terminated and the boot operation continues.

In the case of an invalid password, the login procedure may be repeated as often as required within the boot wait time or as long as the NetBootLoader telnet server is active after the boot wait time has been exceeded. Once the NetBootLoader telnet server is terminated, a telnet login is no longer possible.

2.8 NBL Flash Operations

To achieve an operable system for an application, the application software may be programmed to NBL Flash memory. As mentioned before, the NetBootLoader supports the programming of up to four images to NBL Flash memory whereby each image is assigned its own image number. In addition to this, it also supports the updating of the NetBootLoader itself as well as data transfer from the NBL Flash memory to the data buffer and from the data buffer to an FTP server. The following chapters provide information on performing the various types of NBL Flash memory operations.

2.8.1 NBL Flash Offsets

The currently used NBL Flash memory is one uniform Flash memory. All offsets are based from the beginning of this Flash memory area. The NetBootLoader itself is located at the beginning of the NBL Flash memory area and for this reason this area cannot be used for other image programming. To display an overview of the currently used NBL Flash memory organization use the INFO command.

If the image is an operating system (which is the default case), it must be programmed without an offset. When such an image is programmed to NBL Flash memory, the image length and CRC information is also programmed along with the image. This information is used by the NetBootLoader to determine the validity of the image during the boot operation. During system startup, a valid image is copied to DDR2-SDRAM address 0x0 and started at offset 0x0 (FDT enabled) or 0x100 (FDT disabled).

If an offset is specified, the image will be programmed exactly at this offset without adding length or CRC information. This option is intended for the storing of configuration information which is required to be located in NBL Flash memory.

When programming the NetBootLoader Flash memory using offsets (with the -o option), it is possible to unintentionally overwrite portions of previously programmed images. This may result in either an invalid image or erroneous data contents of the NetBootLoader Flash memory. It is recommended to first verify the contents of the NetBootLoader Flash memory using the INFO command before performing any programming with offsets. Refer to the INFO command for further information.

2.8.2 Programming an Image

The image itself must be compiled and linked to run from the DDR2-SDRAM base address 0x0 of the CPU. The entry point of the executable PPC code must be at offset 0x0 (FDT enabled) or 0x100 (FDT disabled).

Gaining access to the image for programming to NBL Flash memory depends on where it is located. The NetBootLoader can access up to five different sources for images:

- TFTP server
- FTP server
- Motorola S-Records (via the secondary serial interface)
- memory within the visible address range of the NetBootLoader
- NAND Flash

The NetBootLoader uses a single data buffer for downloading an image from a TFTP server, FTP server, or an image as Motorola S-Records. These images must first be downloaded to the data buffer prior to being programmed to NBL Flash memory. An image located within the visible address range of the NetBootLoader is directly accessible for programming.

To access an image located on a TFTP server, the TFTP command is used. To access an image located on an FTP server, the GET command is used. To perform Motorola S-Record acquisition, the SL (SLoad) command is used. Once the image is in the data buffer, the NBL Flash memory is programmed using the LF (Load Flash) command. For an image within visible memory, the LF command is used to program directly to NBL Flash memory.

2.8.3 Accessing TFTP and FTP Servers

To gain access to an image file stored on a TFTP or FTP server an Ethernet port is used. Images are downloaded to the data buffer using the FTP protocol. To use these interfaces, the Ethernet port parameters must first be set, and the operator must have control of the NetBootLoader.

To download an image from a TFTP server, the command TFTP is used. The TFTP server IP and file name of the image must be known and provided to the TFTP command or be provided by the DHCP server via the DHCP command.

To perform a download from an FTP server, the operator must first login to the FTP server. After a successful login, the operator then locates the image file required and downloads it to the data buffer. As with any type of server session, the operator should logout when the session is finished.

Note ...

The commands TFTP, GET, and LS use the same data buffer. Therefore if an LS command is issued after a TFTP or GET command the data buffer will be overwritten. If an LF command follows the LS the NetBootLoader refuses to program the overwritten data buffer to the NBL Flash memory.

2.8.4 Motorola S-Records

The NetBootLoader will also accept Motorola S-Records as an image. The SL command accepts S1, S2 and S3 records. Operation is terminated by the appropriate S9, S8 or S7 record. Other types of records are ignored.

The checksum of every record except end records is checked. Bad records are rejected by the NetBootLoader. The address range of every record is also checked. Records which fall outside of the internal buffer are rejected.

The records must be 0-based. This means that each record's address must correspond to the address where it will be loaded in the data buffer relative to its start. If necessary, the base address can be modified with the -o option of the SL command.

Note ...

If the data buffer is programmed to NBL Flash memory without the -o option (program a startable image) the downloaded image is copied to RAM during startup and is executed there. For this reason images which require to be programmed must start at the address 0x0.

The image must start at the absolute address 0x0 and the entry point of the executable PPC code must be at address 0x0 (FDT enabled) or 0x100 (FDT disabled). If S1 or S2 record input is preferred, please note that these records only include 16 and 24-bit wide addresses.

Note ...

Neither the SL nor LF command can be used to program Motorola S-Records to RAM areas.

For accessing the Motorola S-Records, both the primary and secondary serial interfaces can be used. If available, the user specific LED (ULED0) blinks green while downloading indicating that the transfer is in progress. The transfer itself may take several minutes to complete.

Ensure that the XON/XOFF protocol is used on the host side. This is a fixed setting and cannot be changed. Additionally, ensure that the host does not stop transmission after a number of lines (e.g. OS-9: use the 'nopause' attribute).

The SER0 and SER1 serial interface parameters can be modified with the PF command.

2.8.5 Updating the NetBootLoader

In addition to programming an image to NBL Flash memory, the NetBootLoader itself can be updated. The image can be downloaded in the same way as any other image to the data buffer, or, if available, a redundant NBL Flash can also be the source of a NetBootLoader image. The new version of the NetBootLoader image is then programmed to NBL Flash memory using the CLONE command.

2.8.6 Uploading an NBL Flash Memory Area

The NetBootLoader also has the possibility to upload certain areas of the NBL Flash memory to a host using an Ethernet port. To use this interface, the Ethernet port parameters must first be set and then the operator must gain control of the NetBootLoader and perform an FTP server login. After a successful login, the operator then stores the NBL Flash memory area to be uploaded to the local data buffer using the SF command. Using the PUT command transfers the contents of the data buffer to the FTP server. As with any type of server session, the operator should logout when the session is finished.

2.9 Plug and Play

The NetBootLoader includes “Plug and Play” functionality. This ensures that the CPU module is completely initialized and that all resources necessary for PCI devices (addresses, interrupts etc.) are assigned automatically. This important feature has the advantage that conflicts do not arise when PCI devices are added or removed as the operating system itself does not include the board initialization code.

2.10 Porting an Operating System to the CPU Board

The image for the absolute address 0x0 should be linked with an entry point at the absolute address 0x0 (FDT enabled) or 0x100 (FDT disabled).

One should not attempt to reassign the PCI BAR registers. The assigned values should be read back and these should always be used in the drivers.

Downloaded images are never executed from the NBL Flash memory. The programmed image is always downloaded to DDR2-SDRAM, the absolute address 0x0 being downloaded first. There is no configuration option available to amend this process. If it is necessary to relocate the image to another address after download, simply add a small assembly routine at the beginning of the code which will move the image to the correct address.

2.11 NBL Flash and NAND Flash Usage

The NetBootLoader was originally designed for CPU modules that only had a single NBL Flash memory device. The introduction of CPU modules with two NBL Flashes and NAND Flash has significantly increased the flexibility of systems not only with regards to the NetBootLoader itself, but also with regards to operating systems and ultimately applications as a whole.

The first Kontron product with these features, the AM4100 AMC CPU module, was provided with both redundant NBL Flash as well as NAND Flash. These features have contributed to increased system availability (NetBootLoader backup controlled by IPMI system management) and a very high degree of system autonomy and ruggedness (high capacity of onboard solid state memory for operating systems and applications).

To support these enhancement features, various existing commands were modified and new commands were added.

2.11.1 NBL Flash Usage

With the introduction of a redundant NBL Flash device, the capability to support different versions of the NetBootLoader and provide backup capability was added. This functionality is supported by the NetBootLoader, the onboard IPMI management system, and onboard hardware switching. The exact configuration is a function of application requirements.

2.11.2 NAND Flash Usage

For initial booting purposes, the NBL Flash has adequate storage capability for the NetBootLoader, an operating system image, and, depending on the application requirements, even space for a smaller application. However, once an operating system or application has been started, neither the NetBootLoader nor the NBL Flash is available to the OS or application. Without an onboard mass storage device, the OS and application and their related data are fully contained in RAM.

The NAND Flash, as a mass storage device, fulfills both the requirements of storing and starting an operating system and applications, as well as providing high capacity local data storage.

This page has been intentionally left blank.

3. Product Command Support

The following table provides a complete listing of all NetBootLoader commands. Each Kontron product which employs the NetBootLoader is shown along with a matrix entry indicating the applicability of the command for the product. As can readily be seen, not all commands are applicable to a given product, and, in some cases, commands available depend on the operational mode or configuration of the product indicated.

LEGEND:

AMC use this version of the command for addressing the AMC products indicated

ASMP only available when product is operated in the asymmetric multiprocessing mode

SMP only available when product is operated in the symmetric multiprocessing mode

FTP use this version of the command for addressing the FTP server

NAND use this version of the command (when logged into an FTP server) to address the NAND Flash

1/2-NBL-FL use of the command version depends on availability of NBL Flashes - one or two

NBL-FL this command only addresses the NBL Flash(es)

NAND-FL this command only addresses the NAND Flash

NON-AMC use this version of the command for addressing non-AMC products

Table 6: Command / Product Table

	PRODUCTS																
NETBOOTLOADER COMMANDS	AM4100	AM4101	CP320	CP321	CP620	CP6200	EB405	EB425	EB8245	EB8347	EB8540	EB8541	EB860	VMP1	VMP2	VMP60	DX11
ABORT	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
ASSIGN (ASMP)	X	X															
ASSIGNMEM (ASMP)	X	X															
BL	X	X															
BW	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
BYE	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
CBL	X	X					X	X	X	X	X	X					
CD (FTP)	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
CD, CDCF (NAND)	X	X															
CHECK	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
CLONE (1-NBL-FL)			X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
CLONE (2-NBL-FL)	X	X															

Table 6: Command / Product Table

	PRODUCTS																
NETBOOTLOADER COMMANDS	AM4100	AM4101	CP320	CP321	CP620	CP6200	EB405	EB425	EB8245	EB8347	EB8540	EB8541	EB860	VMP1	VMP2	VMP60	DX11
DHCP	X	X					X	X	X	X	X	X					
ECAT (NAND-FL)	X	X															
EXE (NAND-FL)	X	X															
FDISK (NAND-FL)	X	X															
FDT							X	X		X	X	X					
FDT_SHOW	X	X					X	X		X	X	X					
FRU	X	X															
GET	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
HELP or ?	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
JUMP	X	X															
INFO	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
IPMI	X	X															
LF (NBL-FL)	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
LOAD (NAND-FL)	X	X															
LOGIN	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
LOGOUT	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
LS (FTP)	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
LS, LSCF (NAND)	X	X															
MD	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
MD5 (NAND-FL)	X	X															
MKDIR, MKDIRCF (NAND)	X	X															
MKFS (NAND-FL)	X	X															
MV, MVCF (NAND)	X	X															
NET (NON-AMC)			X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
NET (AMC)	X	X															
PASSWD	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
PCI	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X
PF	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 6: Command / Product Table

	PRODUCTS																
NETBOOTLOADER COMMANDS	AM4100	AM4101	CP320	CP321	CP620	CP6200	EB405	EB425	EB8245	EB8347	EB8540	EB8541	EB860	VMP1	VMP2	VMP60	DX11
PING	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
PUT	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
PWD (FTP)	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
PWD, PWDCF (NAND)	X	X															
RM, RMCF (NAND)	X	X															
RMDIR, RMDIRCF(NAND)	X	X															
RS	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
RUN (OP / SMP)	X	X					X	X	X	X	X	X					
RUN (ASMP)	X	X															
SAVE (NAND)	X	X															
SCRIPT	X	X					X	X	X	X	X	X					
SETCORE (ASMP)	X	X															
SF	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
SL	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
SPEED							X	X		X	X	X		X	X		
SQ (NON-AMC / SMP)	X	X					X	X	X	X	X	X					
SWAP (2-NBL-FL)	X	X															
TFTP	X	X					X	X	X	X	X	X					
VER	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

This page has been intentionally left blank.

4. Command Syntax Reference

The following provides command syntax reference information, a short description, and, in some cases, usage examples. For some commands, not all products have the exact same command syntax as indicated here. Use the HELP command, to ascertain the command syntax for the product in use.

Where an ellipsis (…) appears in the command syntax it means that the command is continued from the previous line. Observe any spaces that may be between the ellipsis and the remainder of the command.

ABORT

FUNCTION:	Terminate the NetBootLoader boot operation
SYNTAX:	`abort`
DESCRIPTION:	This command is used by the operator to terminate the boot operation during the boot wait time to allow the operator to perform other NetBootLoader operations. To be asserted, either it must be issued during the boot wait time, or be issued as a script command once the bootwait time has elapsed.

ASSIGN (ASMP)

FUNCTION:

Assign devices to the core specified or display the current assignments

SYNTAX:

```
assign [default | <corenum> <devname> …
<devname> | <corenum> all]
```

where:

assign command

issuing the command without arguments will display the current assignment for both cores

default option:

set the default assignment

<corenum> parameter: string

0, 1

defines the core to which the device(s) specified is(are) to be assigned

<devname> parameter: string

<[x …]x>

defines the device(s) to be assigned

all option:

assigns all devices to core specified

DESCRIPTION:

The ASSIGN command provides for selective ordering of a given device to one or the other of the cores.

The default device assignment is as follows:

device:	core	device:	core
i2c:	0	pci:	0
eth0:	0	srio:	0
eth1:	0	ide:	0
eth2:	0	dma0:	0
eth3:	1	dma1:	0
ser0:	0	dma2:	0
ser1:	1	dma3:	0

Only one core at a time may be addressed, however, more than one device may be assigned at a time for a given core.

ASSIGNMEM (ASMP)

FUNCTION:	Assign memory to the cores or display the current assignments
SYNTAX:	`assignmem [<corenum> <size>]` where: assignmem — command issuing the command without arguments will display the current assignment for both cores <corenum> — parameter: string 0, 1 defines the core to which the memory is to be assigned <size> — parameter: string <[x …]x> defines the memory size to be assigned to the core specified entry should be hexadecimal: e.g. 0x18000000
DESCRIPTION:	Memory space allocation is done for one core whereby the remaining available space is automatically assigned to the other core. A minimum of 256 MB must be allocated to each core.

BL

FUNCTION: Set or display the parameters of the bootline stored in RAM

SYNTAX:

```
bl [-|<bootline>]
```

where:

bl	command
-	option: delete bootline
<bootline>	parameter: value: string <[x …]x> defines the bootline to be used with the next RUN command

DESCRIPTION: The commnd BL only specifies the bootline for the image in the data buffer at the time the RUN command is invoked.

If the command BL is invoked without parameters, it returns the contents of the bootline (for the currently specified core).

Invoking the command BL with the <bootline> parameter overwrites any previous invoked bootline (for the currently specified core).

USAGE: Example of bootline setting:

COMMAND / RESPONSE (none):

```
NetBtLd> bl root=nfs console=ttyS0,9600

NetBtLd>
```

BW

FUNCTION:

Set or display the parameters of the boot wait function of the NetBootLoader

SYNTAX:

```
bw [<time>|-f]
```

where:

bw	command: bootwait
<time>	parameter: numeric string 0, 1, 2, 5, 10, 20, 50 time in seconds
-f	option: force CRC update

DESCRIPTION:

The command BW displays the parameter “<time>” setting.

The parameter “<time>” specifies the waiting time in seconds that the boot operation is delayed before the image is loaded and started. No values other than these are supported.

The option “-f” is used to force updating of the CRC value of boot section of the EEPROM.

Some products do not support values of 0 and 1. Refer to the online help text for specific information.

With products that support a bootwait time of 0, care should be taken to ensure that some means are available to access the NetBootLoader (e.g. via the Abort switch; access to the abort signal; etc.) before specifying a value of 0. Otherwise it may not be possible to access the NetBootLoader again during ensuing boot operations.

USAGE:

Display setting of “<time>” parameter

COMMAND / RESPONSE:

```
bw
WaitTime: 20
```

Set boot wait time to 50 seconds

COMMAND / RESPONSE (none):

```
bw 50
```

BYE

FUNCTION:	Terminate an FTP server session
SYNTAX:	`bye`
DESCRIPTION:	An FTP server session which has been established with the command LOGIN is terminated with the command BYE.

CBL

FUNCTION: Set or display the parameters of the bootline(s) stored in NBL Flash

SYNTAX:

```
cbl [<num> (-|<bootline>)]
```

where:

cbl	command: change bootline
<num>	parameter: string 0, 1, 2, 3, c ID number of the image to be associated with the bootline or bootline which is common to all images no other values than those above are supported
-	option: delete bootline for image specified or common
<bootline>	parameter: value: string <[x …]x> defines the bootline to be used with the kernel indicated by <num> or the common bootline

DESCRIPTION: If the command CBL is invoked without parameters, it returns the contents of all bootlines.

Invoking the command CBL with the <bootline> parameter overwrites any previous bootline for the image specified.

USAGE: Display the bootline for image 2

COMMAND / RESPONSE:

```
cbl
<contents of all bootlines are displayed>
```

Set a common bootline for all images

COMMAND / RESPONSE (none):

```
NetBtLd> cbl c console=ttyS0,9600

NetBtLd>
```

CD (FTP)

FUNCTION: Change the current FTP server directory

SYNTAX:

```
cd <new-path>
```

where:

cd	command: change directory
<new-path>	parameter: string <[x …]x> new directory path

DESCRIPTION: If an FTP server session has been established with the LOGIN command, the command CD is used to change the current FTP server directory.

The argument “<new-path>” may be an absolute or relative path. The format depends on what the server accepts.

UNIX hosts require that the directory names must be entered exactly in the same case.

CD, CDCF (NAND Flash)

FUNCTION: Change the current NAND Flash directory

SYNTAX:

```
cd <new-path>
cdcf <new-path>
```

where:

cd, cdcf	command: change directory (on NAND Flash)
<new-path>	parameter: string <[x …]x> new directory path

DESCRIPTION: To access the NAND Flash when logged on to an ftp server, use the command CDCF.

CHECK

FUNCTION: Verify validity of image programmed to NBL Flash memory

SYNTAX:

```
check
```

DESCRIPTION: When an image is programmed to NBL Flash memory, a CRC is performed and the results are stored along with the image. The CHECK command provides status information for the current images in NBL Flash memory.

USAGE: Verify valid image is stored in NBL Flash memory

COMMAND / RESPONSE:

```
check
Checking Image: 0
  check image crc: ok
  length in flash: 0x0053d004
  sectors used   :    84

Checking Image: 1
  check image crc: ok
  length in flash: 0x0001029c
  sectors used   :     2

Checking Image: 2
  check image crc: fail
  length in flash: -
  sectors used   : -

Checking Image: 3
  check image crc: fail
  length in flash: -
  sectors used   : -
```

CLONE (One NBL Flash)

FUNCTION: Program the NetBootLoader to FLASH

SYNTAX:

```
clone [-n]
```

where:

clone	command
-n	option: program from data buffer

DESCRIPTION: To update the NetBootLoader itself, the command “clone” is used.

If the application image source for programming is to be the data buffer, the image must first be downloaded to the data buffer from an tftp/ftp server. Otherwise, additional Flash must be available to provide the image to be cloned (see section 2.7.5).

To program from the data buffer, the command “clone -n” is used.

The new image is checked for validity. If an image is invalid, the update is aborted. Additionally, the operation must be confirmed by typing the word “yes”. Any other or no input will cancel the operation.

USAGE: Program NetBootLoader (normal operation)

COMMAND / RESPONSE:

```
NetBtLd> clone -n
clone: Fixup FLASH info from ftp buffer
This will overwrite the current
... NetBootLoader, are you sure? [no] yes
clone: System transferred; Start again,
... assure that Bootjumper is removed.
NetBtLd>
```

Note: When responding to the overwrite query, “yes” must be spelled out. Any other response will terminate the cloning operation.

USAGE: Program NetBootLoader (image not valid)

COMMAND / RESPONSE:

```
NetBtLd> clone -n
clone: Fixup FLASH info from ftp buffer
Image length invalid, image is damaged,
abort.
NetBtLd>
```

CLONE (Two NBL Flashes)

FUNCTION:	Program the NetBootLoader to NBL Flash memory
SYNTAX:	`clone [-n] [-a] [-y]` where: clone command -n option: program from data buffer to program from either the active or inactive NBL Flash memory, do not specify this option -a option: specify this option to program the active NBL Flash memory (if -n is not specified, source is the inactive NBL Flash memory) to program the inactive NBL Flash memory, do not specify this option (if -n is not specified, source is the active NBL Flash memory) -y option: this option is used with the SCRIPT command or in “nbl<n>.cfg” or “net.cfg” files to confirm that cloning is to be performed
DESCRIPTION:	To update the NetBootLoader itself, the command CLONE is used. The image source for programming is either the data buffer (-n) (of the currently specified core) or one or the other of the two NBL Flash memories depending on the “-a” option selection. The new image is checked for validity. If an image is invalid, the update is aborted. Further, the operation must be confirmed by typing the word “yes”. Any other or no input will cancel the operation. When used with the SCRIPT command, the -y option must be specified or the clone operation will be terminated.

CLONE (Two NBL Flashes)

DESCRIPTION:

Programming the NetBootLoader:

- from the data buffer to the active NBL Flash:
  clone -n -a
- from the inactive NBL Flash to the active NBL Flash:
  clone -a
- from the data buffer to the inactive NBL Flash:
  clone -n
- from the active NBL Flash to the inactive NBL Flash:
  clone

USAGE:

Program NetBootLoader to the inactive NBL Flash

COMMAND / RESPONSE:

```
NetBtLd> clone -n
clone: Fixup FLASH info from ftp buffer
This will overwrite the current
... NetBootLoader in the fallback flash, are
... you sure? [no] yes
clone: System transferred to fallback flash
NetBtLd>
```

Note: When responding to the overwrite query, “yes” must be spelled out. Any other response will terminate the cloning operation.

Program NetBootLoader (image not valid)

COMMAND / RESPONSE:

```
NetBtLd> clone -n
clone: Fixup FLASH info from ftp buffer
Image length invalid, image is damaged,
abort.
NetBtLd>
```

DHCP

FUNCTION:

Interface to a DHCP or BOOTP server; exchange network configuration parameters

SYNTAX:

```
dhcp [-e|<timeout>]
```

where:

dhcp	command
-e	option: continue sending requests until answer is received
<timeout>	parameter: value: numeric string <[n …]n> time, in seconds; must be ≥ 5 seconds

DESCRIPTION:

This command is used to set the network parameters for operation of an Ethernet port via either a DHCP or BOOTP server.

Initially the board does not have a valid Ethernet interface configuration, and, therefore, this interface is inoperable. The initial configuration must be done either manually from a serial interface using the command NET, or, if a DHCP or BOOTP server is available, it can be done automatically by the DHCP command.

Manually configured parameters are permanently stored. Parameters configured using the DHCP command are temporary and will be lost if the system is reset or cold started.

Some products require that the IP address be set to 255.255.255.255 with the NET command prior to using the DHCP command.

USAGE:

Program NetBootLoader (normal operation)

COMMAND / RESPONSE:

```
NetBtLd> dhcp
Sending request... reply from BOOTP/DHCP
server.
Network     initializated ok.

Server address is 192.168.112.2, our IP
... address is 192.168.112.14.
Filename :

NetBtLd>
```

ECAT

FUNCTION: View or create boot script configuration files on the NAND Flash

SYNTAX:

```
ecat [-f] <filename> [<text-to-write>]
```

where:

ecat	command: edit(concatenate)
-f	option: force file overwrite if the “<filename>” file exists on NAND Flash it will be overwritten
<filename>	parameter: string name of the file to be viewed or created nbl[0 … 9].cfg or net.cfg for NetBootLoader configuration files <[x...]x[.[x..]x]> any file name that conforms to 8.3 format if a file with this name already exists on the NAND Flash and “<text-to-write>” has been specified, the error message: “File <filename> already exists, use option -f to overwrite” is returned
<text-to-write>	parameter: string <[x …]x> text to be written to file

DESCRIPTION:

This command is used to view or create configuration files (scripts) which the NetBootLoader can execute during the boot process.

Up to 11 separate single-line configuration files may be defined for automatic use during booting: nbl0.cfg to nbl9.cfg and net.cfg

In addition, any single-line file may be created with this command for whatever purpose. Such files may also be executed with the EXE command.

If the command is issued with a valid filename and no other options, the contents of the file are echoed to the console.

No editing support beyond basic support from the command interpreter (CTL-x and CTL-a) is provided. As the file can only consist of one line, entering “RETURN” will invoke the command using the previously entered text.

ECAT

USAGE:

Create config file “nbl2.cfg”; attempt to create new config file; create new config file:

COMMAND / RESPONSE:

```
NetBtLd> ecat nbl2.cfg dhcp; tftp; run
NetBtLd> ecat nbl2.cfg tftp 172.100.1.35
... root.bin; run
File nbl2.cfg already exists, use option -f
... to overwrite
NetBtLd> ecat nbl3.cfg tftp 172.100.1.35
... root.bin; run
```

EXE

FUNCTION:

Execute the contents of a configuration file on the NAND Flash

SYNTAX:

```
exe <filename>
```

where:

exe	command
<filename>	parameter: string name of the file to be executed nbl[0 … 9].cfg net.cfg <[x...]x[.[x..]x]> (must conform to 8.3 format)

DESCRIPTION:

This command is used to execute the contents of a previously created configuration file (script).

Invoking this command will terminate the current boot sequence, and give control to the contents of the configuration file to be executed.

FDISK

FUNCTION: Create or remove primary partitions on the NAND Flash

SYNTAX:

```
fdisk -p|-a <num>|-d <num>|-n <num>
… <cylStart> <cylEnd>
```

where:

fdisk	command
-p	option: display data show geometry and defined partitions
-a	option: set boot flag
<num>	parameter: numeric string 0, 1, 2, 3, 4 set the boot flag on partition <num> and remove it from the other partitions when set to 0 (zero) the boot flag is removed from all partitions
-d	option: delete partition
<num>	parameter: numeric string 1, 2, 3, 4 delete the partition specified by <num>
-n	option: create partition
<num>	parameter: numeric string 1, 2, 3, 4 specifies the primary partition to be created the partition type is set to FAT-16
<cylStart>	parameter: numeric string <[n …]n> start cylinder for the partition specified
<cylEnd>	parameter: numeric string <[n …]n> end cylinder for the partition specified

FDISK

DESCRIPTION:

This command is used to specify the partitioning of the NAND Flash and control the setting of the partition boot flag.

A maximum of four partitions are definable whereby only one partition at a time may have the boot flag set.

The cylinders specified for partition creation are inclusive. Use the “-p” option to obtain exact information concerning the status of the NAND Flash.

USAGE:

Display status of NAND Flash

COMMAND / RESPONSE:

```
NetBtLd> fdisk -p

Disk /dev/cfdisk/: 512 MB, 512483328 bytes
16 heads, 63 sectors/track, 993 cylinders
Cylinder size 1008 * 512 = 516096 bytes

Partition     Boot Start  End  Size[kiB] Ident Mounted
to
/dev/cfdisk/1  yes   200  400     101304     6  /cf1
/dev/cfdisk/2   no   522  993     237888    83

NetBtLd>
```

Create a third primary partition on the NAND Flash

COMMAND / RESPONSE:

```
NetBtLd> fdisk -n 3 401 521

Disk /dev/cfdisk/: 512 MB, 512483328 bytes
16 heads, 63 sectors/track, 993 cylinders
Cylinder size 1008 * 512 = 516096 bytes

Partition     Boot Start  End  Size[kiB] Ident Mounted
to
/dev/cfdisk/1  yes   200  400     101304     6  /cf1
/dev/cfdisk/2   no   522  993     237888    83
/dev/cfdisk/3   no   401  521      60984     6

NetBtLd>
```

FDT

FUNCTION: Enable creation of a flattened device tree (FDT)

SYNTAX:

```
fdt [<switch>]
```

where:

fdt	command
<switch>	parameter: numeric string 0,1 when set to 0 (zero) disables creation of an FDT, image is booted from 0x100 when set to 1 (one) enables creation of an FDT, image is booted from 0x0 when not specified, displays current setting

DESCRIPTION: This command is used to specify the creation of an FDT for an operating system if required.

An FDT is required for Linux kernels 2.6.17 and later.

USAGE: Create an FDT

COMMAND / RESPONSE:

```
NetBtLd> fdt 1
Enabling creation of flattened device tree.
Boot image from address 0.
NetBtLd>
```

Display the current FDT setting

COMMAND / RESPONSE:

```
NetBtLd> fdt
Creation of flattened device tree is enabled.
Image is booted from address 0.
NetBtLd>
```

FDT_SHOW

FUNCTION:

Display the flattened device tree (FDT) information, and copy FDT information to the data buffer if required

SYNTAX:

```
fdt_show [<num>] [-o]
```

where:

fdt_show	command
<num>	parameter: string
	0, 1, 2, 3
	ID number of the image for which FDT information is to be displayed
	no other values than those above are supported
	when not specified, displays FDT information for image in the data buffer
-o	option: overwrite
	copy FDT contents to the data buffer (will overwrite any image in data buffer)

DESCRIPTION:

This command is used to have the NetBootLoader display the contents of the FDT for the image specified (for the currently specified core).

Use the “-o” option to force writing of the FDT contents to the data buffer. Once in the data buffer, the FDT information can transferred to a server.

The information provided can be used for system analysis or diagnostic purposes.

Some older NetBootLoader versions do not support the “-o” option. In this case, the FDT contents are copied to the data buffer and any image in the data buffer is overwritten.

If the “-o” option is not specified with NetBootLoader versions which support it, the FDT contents are copied to the top of RAM.

FRU

FUNCTION:	Display a hexdump of the “FRU” data
SYNTAX:	`fru` where: fru command
DESCRIPTION:	This command is used to display FRU data in hexadecimal format.

GET

FUNCTION:	Download file from FTP server
SYNTAX:	`get <filename>` where: get command <filename> parameter: string <[x …]x> name of image file to be downloaded, or path and name of image file to be downloaded
DESCRIPTION:	To download a file from the FTP server to the data buffer (of the currently specified core), the command GET is used. A successful FTP server login must be carried out before a file can be downloaded. The argument “<filename>” must refer to an existing and accessible file on the server and the syntax must follow the requirements on the server, e.g. case sensitiveness. The argument may also include a path specification, if the server supports this.

HELP or ?

<table>
<tr><td>FUNCTION:</td><td>Display online help pages</td></tr>
<tr><td>SYNTAX:</td><td>help | ? [<cmdname>]
where:
help or ? command
<cmdname> parameter: string
<[x …]x>
print all commands starting with <cmdname></td></tr>
<tr><td>DESCRIPTION:</td><td>This command displays the online help pages. The syntax of every command and a brief description is shown. The display output pauses after every page. The output can be continued with any key. Entering a “.” (period) aborts the help function.</td></tr>
</table>

INFO

FUNCTION: Display system information

SYNTAX:

```
info
```

DESCRIPTION: The command INFO is used to display an information summary for the running system.

Images programmed using the “-o” option of the command LF are not shown.

USAGE: Display system information

COMMAND / RESPONSE:

```
NetBtLd> info

CPU       : PowerPC, MPC8641D
Board     : AM4101
Mode      : SMP
PCIe      : available
sRIO      : available
Ram       : 80000000
Flash     :
  Name          : AMD 29LV320
  Bank          : 0
  Bytelane      : 0
  BankPortsize  : 16
  ChipPortsize  : 16
  Offset        : 0x0
  Size          : 0x400000

NetBootLoader used FLASH: 0x0 - 0xa0000

Sector usage map is calculated, this might
take some time.

Sector usage map (Sector Size = 0x10000):
  0x0000: nnnnnnnn nn000000
  0x0010: 00000000 00000000
  0x0020: 00000111 11111111
  0x0030: ........ ........

NetBtLd>
```

Where: n = NetBootLoader; 0 = image 0; 1 = image 1; . = usage unknown

IPMI

FUNCTION:	Display IPMI information
SYNTAX:	`ipmi` where: ipmi command
DESCRIPTION:	This command is used to display the currently active IPMI firmware version and the general status of the MMC.

JUMP

FUNCTION:	Jump to script specified by <num>
SYNTAX:	`jump <num>` where: jump command <num> parameter: string 0 ... 9 ID number of the “nbl<num>.cfg” file in the NAND Flash containing the script to be executed no other values than those above are supported
DESCRIPTION:	During the standard boot process this command can be used in a script file to specify a script to be executed when the current script execution is completed. If this command is invoked during the alternate boot process it is disregarded.

LF (NBL Flash)

FUNCTION: Load Flash (active NBL Flash)

SYNTAX:

```
lf [<num>] [[-r]|[[-o[=]<offset>[ -k]]
… [-m[=]<adr> -l[=]<len>]]]
```

where:

lf	command
<num>	parameter: numeric string
	0, 1, 2, 3
	ID number assigned to this image
-r	option:
	delete image specified, or, if none specified, delete image 0
-o	option: offset
	<offset>parameter: hexadecimal
	<[x …]x>
	program to FLASH offset of ...
-k	option: keep
	retain surrounding contents
-m	option: memory (address)
<adr>	parameter: hexadecimal
	<[x …]x>
	absolute address of image to be programmed
-l	option: length
<len>	parameter: hexadecimal
	<[x …]x>
	length of image to be programmed

DESCRIPTION: If <num> is not specified, 0 is assumed.

Without options, the NBL Flash memory is programmed using the contents of the data buffer (of the currently specified core). If no image is available in the data buffer, programming is terminated.

If no offset option (“-o”) is specified the image is added along with the CRC and length information.

Normally, the local data buffer holds the image to be programmed. However, if the “-m” and “-l” options are specified, the image is programmed from the absolute address specified.

LF (NBL Flash)

DESCRIPTION:

If the “-o” option is specified, the contents are programmed exactly at this offset in NBL Flash memory. No length and no CRC information is added. In addition, no image number is assigned (even if specified), and the image cannot be loaded and started with the RUN command.

The “-k” option can be specified to prevent deletion of the surrounding NBL Flash memory contents.

Flash memory can only be erased sector-wise. If an image is programmed to a certain offset with the “-o” option, at least this sector (and maybe one or more of the following sectors depending on the size of the image) will be erased. The “-k” option can be used to retain the surrounding data, however, this slows down the operation significantly.

To achieve fast programming of parameter images without destroying other NBL Flash memory contents, the data should be placed at a sector boundary and the sector(s) must not contain any other data or executable images. If organized this way, use of the “-k” option can be avoided.

Note: The “lf” command cannot be used to program the NetBootLoader.

USAGE:

Program NBL Flash memory from data buffer and add CRC and image length (Image ID = 0 is assumed)

COMMAND / RESPONSE (none):

```
lf
```

Program NBL Flash memory from visible address at 0x87000000 for length of 0x123456

COMMAND / RESPONSE (none):

```
lf -m=87000000 -l=123456
```

Program NBL Flash memory from data buffer to offset 0xF4240 and retain adjacent NBL Flash memory contents

COMMAND / RESPONSE (none):

```
lf -o=f4240 -k
```

Delete image 1 from FLASH contents

COMMAND / RESPONSE (none):

```
lf 1 -r
```

LOAD (NAND Flash)

FUNCTION:

Load an operating system image and optionally a RAMdisk filesystem from the NAND Flash to the data buffer

SYNTAX:

```
load [-b] [-r <filenameRD>] <filenameOS>
```

where:

load	command
-b	option: backup only load file(s) if image(s) in data buffer invalid, or the md5sum check(s) failed
-r	option: ramdisk load initrd specified by <filenameRD>
<filenameRD>	parameter: string <[x …]x> filename of initrd
<filenameOS>	parameter: string <[x …]x> filename of operating system image

DESCRIPTION:

The LOAD command is basically designed to transfer an OS image from the NAND Flash to the data buffer (of the currently specified core) for system booting. For OSs requiring a RAMdisk filesystem, the option “-r <filenameRD>” is available.

The “-b” option (backup) permits selective file loading. When used in a configuration file, a series of LOAD commands may be included to ensure that if the first LOAD command fails (data buffer image invalid or md5sum check failed), the loading of a second (or third, etc.) file can be defined.

In conjunction with the MD5 and RUN commands using the “-b” and “-c” options respectively, a very high degree of system integrity can be achieved.

LOAD (NAND Flash)

USAGE:

Sample configuration file for loading files:

```
setcore 0;
load -r initrd1 img1;
md5 -r initrd1.md5 img1.md5;
load -b -r initrd2 img2;
md5 -b -r initrd2.md5 img2.md5;
load -b initrd3 img3;
setcore 1;
load -r initrd4 img4;
md5 -r initrd4.md5 img4.md5;
load -b -r initrd5 img5
run -c
```

In this example, core 0 is prepared first: If image 1 does not have a valid md5sum, image 2 is loaded. If image 2 is invalid, image 3 is loaded. On core 1 image 4 is tried. If this is invalid, image 5 is loaded. Run with the “-c” option ensures that images for cores 0 and 1 must be loaded successfully or the system will not start.

LOGIN

FUNCTION:	Initiate FTP server session
SYNTAX:	`login <ip-of-host> <username> [<password>]` where: login command <ip-of-host> parameter: string <nnn.nnn.nnn.nnn> IP address of host <username> parameter: string <[x …]x> FTP server user name <password> parameter: string <[x …]x> user's password
DESCRIPTION:	The command LOGIN is used to establish an FTP server session. The "<password>" parameter is not necessary if the server does not require one.
USAGE:	Initiate FTP server session COMMAND / RESPONSE: `login 192.168.47.12 johndoe mypassword` (Response is dependent on the server accessed)

LOGOUT

FUNCTION:	Terminate telnet session with NetBootLoader
SYNTAX:	`logout`
DESCRIPTION:	A remote telnet session will be terminated with the command LOGOUT. No image is loaded and started if the session is terminated with LOGOUT. The NetBootLoader waits for a new session to be initiated or for a command entry from the serial console.

LS (FTP)

FUNCTION:	When logged into an FTP server, this command displays a listing of the current FTP server directory
SYNTAX:	`ls`
DESCRIPTION:	To display a listing of the current FTP server directory the command LS is used. This command downloads the listing to the data buffer and then the listing is displayed. Any previously loaded image in the data buffer is overwritten. If an attempt is then made to program the NBL Flash memory after the LS command has been issued it will fail.

LS, LSCF (NAND Flash)

FUNCTION:	Display listing of the current directory on the NAND Flash

SYNTAX:	`ls [-l] <dirname>`
	`lscf [-l] <dirname>`
	where:
	ls, lscf command
	-l option: display file size
	<dirname> parameter: string
	<[x …]x>
	displays contents of the directory specified by <dirname>

DESCRIPTION:	To access the NAND Flash when logged on to an ftp server, use the command LSCF.

MD

FUNCTION:	Display visible memory
SYNTAX:	`md [<adr>]` where: md command <adr> parameter: hexadecimal <[x …]x> starting address of a visible memory area
DESCRIPTION:	To display a visible memory area the command MD is used. The first time the command MD is issued, visible memory contents starting at the address 0x0 are displayed if no “<adr>” parameter is used. If issued again without the “<adr>” parameter, the display starts with the end address of the previous display. Data is displayed as hexadecimal 32-bit words and as ASCII dump.

MD5 (NAND Flash)

FUNCTION: Performs checksum verification of the data buffer contents

SYNTAX:

```
md5 [-b|-n [-f]] [-r <filenameMd5RD>]
<filenameMd5OS>
```

where:

md5	command
-b	option: do not calculate md5sum(s) if last specified file(s) was (were) not loaded
-n	option: new create new md5sum(s) and write them to the file(s)
-f	option: force file overwrite overwrite file(s) if file(s) already exist(s)
-r	option: ramdisk check/create check or create md5sum of initrd
<filenameMd5RD>	parameter: string <[x …]x> filename of md5sum for initrd
<filenameMd5OS>	parameter: string <[x …]x> filename of md5sum for operating system image

DESCRIPTION: This command is used to verify the integrity of file(s) loaded in the data buffer (of the currently specified core). It can verify both OS images as well as the initrd filesystem.

The md5sum file must have the same 128-bit fingerprint as generated by the GNU utility “md5sum”.

The “-b” option permits selective performance of md5sum checking. In particular, in a configuration file it can be used in conjunction with the LOAD command to provide full control over the image loading and starting process. Refer to the example provided with the LOAD command.

MKDIR, MKDIRCF (NAND Flash)

FUNCTION:	Create a directory on the NAND Flash

SYNTAX:	`mkdir <dirname>`
	`mkdirCF <dirname>`
	where:
	mkdir; mkdircf — command
	<dirname> — parameter: string
	<[x …]x>
	name of the directory to be created on the NAND Flash

DESCRIPTION:	To access the NAND Flash when logged on to an ftp server, use the command MKDIRCF.

MKFS (NAND Flash)

FUNCTION: Create a filesystem for the partition specified on the NAND Flash

SYNTAX:

```
mkfs [-f] <num>
```

where:

mkfs	command
-f	option: force creation of filesystem invoking this option will overwrite any existing filesystem on the partition specified
<num>	parameter: numeric string 1, 2, 3, 4 number of the partition where filesystem is to be created

DESCRIPTION:

This command creates a FAT-16 filesystem.

If the option -f is not used and a FAT filesystem already exists on the partition specified, an error message is returned.

Invoking this command with the -f option on a partition which already has a FAT filesystem will result in the loss of any data stored on this partition.

If a partition has a filesystem other than FAT, it will not be recognized and will be overwritten.

MV, MVCF (NAND Flash)

FUNCTION:	Rename a file on the NAND Flash

SYNTAX:	`mv -f <oldfilename> <newfilename>`
	`mvcf -f <oldfilename> <newfilename>`
	where:
	mv command
	-f option: force file overwrite
	if the "<newfilename>" file exists on NAND Flash it will be overwritten
	<oldfilename> parameter: string
	<[x …]x>
	name of the file to be renamed on the NAND Flash
	<newfilename> parameter: string
	<[x …]x>
	new name for the file to be renamed
	if a file with this name already exists on the NAND Flash the error message:
	"File filename already exists, use option -f to overwrite"
	is returned

DESCRIPTION:	This command is used to rename an existing file on the NAND Flash in the current working directory. To access the NAND Flash when logged on to an ftp server, use the command MVCF.

NET (NON-AMC)

FUNCTION: Set or display the parameters for the Ethernet interface

SYNTAX:

```
net [<ip-addr>] [-netmask <netmask>]

… [-gw <gateway>] [-num <num_net>] [-f]
```

where:

net	command
<ip-addr>	parameter: numerical string IP address of CPU board: nnn.nnn.nnn.nnn
-netmask	option: netmask
<netmask>	parameter: numerical string netmask of CPU board: nnn.nnn.nnn.nnn
-gw	option: gateway
<gateway>	parameter: numerical string gateway address for network: nnn.nnn.nnn.nnn
-num	option: number
<num_net>	parameter: numerical string 0, 1 ... logical identifier of Ethernet port addressed by this command
-f	option: force CRC update

DESCRIPTION: To set or display the parameters of the Ethernet interface the command “net” is used.

Initially the CPU board does not have a valid Ethernet interface configuration, and, therefore, this interface is inoperable. The initial configuration must be done from the primary serial interface using the command “net ... -f”.

Using the “-f” option forces a CRC to be performed and stored along with the other configuration parameters in the serial EEPROM.

Once the initialization of the Ethernet interface is done, the CPU board must be restarted for the parameters to take effect. Later changes to the parameters do not require the use of the “-f” option to force a CRC. This is done automatically. Only in the event that the Ethernet interface does not properly initialize, may it be necessary to re-enter the parameters using the “-f” option.

If [-num <num_net>] is not specified, -num 0 is assumed.

NET (AMC)

FUNCTION: Set or display the parameters for an Ethernet port

SYNTAX:

```
net [<num_net>] [<ip-addr>]

… [-netmask <netmask>] [-gw <gateway>]

… [-a] [-f]
```

where:

net	command
<num_net>	parameter: numerical string 0, 1 ... n logical identifier of Ethernet port addressed by this command where: 0 = TSEC1 1 = TSEC2 2 = TSEC3 3 = TSEC4
<ip-addr>	parameter: string <nnn.nnn.nnn.nnn> IP address of NetBootLoader
-netmask	option: netmask
<netmask>	parameter: string <nnn.nnn.nnn.nnn> netmask of NetBootLoader
-gw	option: gateway
<gateway>	parameter: string <nnn.nnn.nnn.nnn> gateway address for network
-a	option: specifies that the NetBootLoader is to use this port for network operations
-f	option: force CRC update

NET (AMC)

DESCRIPTION:	To set or display the parameters of an Ethernet port the command NET is used. Initially the CPU board does not have any valid Ethernet port configurations, therefore, these interfaces are inoperable. The initial configuration of the Ethernet ports must be done from the primary serial interface using the command “net ... -f”. Using the “-f” option forces a CRC to be performed and stored along with the other configuration parameters in the serial EEPROM. Once the configuration of an Ethernet port is done, it is available for operation. Later changes to the parameters do not require the use of the “-f” option to force a CRC. This is done automatically. If the NET command is issued without any parameters, the current parameter settings of all the Ethernet ports is displayed.

PASSWD

FUNCTION:

Set the telnet password

SYNTAX:

```
passwd [-f|-d]
```

where:

passwd	command
-f	option: if password is not known
-d	option: disable disable telnet login (remote access)

DESCRIPTION:

To set the password for telnet sessions with the NetBootLoader the command PASSWD is used. This command is interactive, meaning that after it is issued, the NetBootLoader responds with an appropriate request to the operator which must be properly acknowledged or the operation fails (refer to USAGE below).

To set the password in the event it is unknown, use the option “-f”. This is can only be accomplished from the primary serial interface and not from the Ethernet interface.

With the option “-d”, the remote telnet login can be disabled by invalidating the password.

USAGE:

Set password

COMMAND / RESPONSE:

```
NetBtLd> passwd
Old Password: *****
New Password: *****
Type again   : *****
NetBtLd>
```

(The old password must be known)

Set password when the old password is not known

COMMAND / RESPONSE:

```
NetBtLd> passwd -f
New Password: *****
Type again   : *****
NetBtLd>
```

PCI

FUNCTION:	Display PCI information
SYNTAX:	`pci`
DESCRIPTION:	The command PCI is used to display detailed information on all detected PCI devices. The bus number, device number, function number, vendor, and device ID's are displayed together with the configured base addresses and the assigned IRQ number.

PF

FUNCTION:	Set or display the serial port parameters (format)

SYNTAX:	`pf [<port> [<baud>][/[<bitschar>]`
	`…[/[<parity>][/<stops>]]]]`
	where:
	pf command
	<port> parameter: string:
	"term" or "ser0"
	defines serial port to be configured (term = primary; ser0 = scondary)
	<baud> parameter: numeric string:
	"300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200, 38400, 115200"
	defines the baud rate for the port
	<bitschar> parameter: numeric string:
	"7" or "8"
	defines the number of bits per character
	<parity> parameter: string:
	"n" (none), "o" (odd), "e" (even)
	defines parity to be used
	<stops> parameter: number string:
	"1" or "2"
	defines number of stop bits

PF

DESCRIPTION:	To set or display the operational parameters for the serial interfaces the command PF is used. At startup the settings for the primary and the secondary interfaces are always set to the default values (9600/8/n/1). This is to avoid a possible system lockout. If other settings are required during operation of the NetBootLoader they may be made. If changes are made, it must be ensured that corresponding parameters are used for the operator console. Issuing this command without parameters being specified will display the current serial port settings. Syntax-wise, no spaces are permitted between the parameters and they must be separated with a slash. Not all parameters must be specified, but the “/” characters must be present to distinguish the different parameters from each other. The sequence can be aborted after every option.

USAGE:	Set “term” to 300 Baud, 7 Bits/char, odd parity, and 2 stop bits
	COMMAND / RESPONSE (none):
	`pf term 300/7/o/2`

	Set the bits per character parameter of SER0 to 7
	COMMAND / RESPONSE (none):
	`pf ser0 //7`

	Set the stop bits parameter of SER0 to 2
	COMMAND / RESPONSE (none):
	`pf ser0 ///2`

PING

FUNCTION: Verify operability of an Ethernet interface

SYNTAX:

```
ping <ip_addr> [-c <count>] [-s <size>]

… [-w <wait>]
```

where:

ping	command
<ip-addr>	parameter: string <nnn.nnn.nnn.nnn> IP address of target
-c	option: count
<count>	parameter: numeric string <[n ...]n> number of packets to send
-s	option: size
<size>	parameter: numeric string <[n ...]n> size of packet to send in bytes
-w	option: wait
<wait>	parameter: numeric string <[n ...]n> wait time between packets in seconds

DESCRIPTION: To verify the operational status of an Ethernet interface the command PING is used. This command tests the network connection and target server’s ability to respond.

If no other parameters are specified, four requests will be sent. This can be changed with the parameter “-c”. The typical size of a ping packet can be changed with the parameter “-s” and the time between requests, which is typically one second, can be changed with the parameter “-w”.

Responses to the PING command are dependent on the performance of the network.

USAGE: Send ten packets, 100 bytes long, and wait two seconds between packets

COMMAND / RESPONSE:

```
ping 192.192.158.7 -c 10 -s 100 -w 2
```

PUT

FUNCTION:	Upload contents of the data buffer to the FTP server
SYNTAX:	`put <filename>` where: put command <filename> parameter: string <x … x> file name to be used for contents of data buffer to be uploaded
DESCRIPTION:	To upload the contents of the data buffer (of the currently specified core) to a file on an FTP server, the command PUT is used. The file indicated by the parameter “<filename>” is created on the server. In the event that a file with this name already exists, its contents will be overwritten.

PWD (FTP)

FUNCTION:	Display the current FTP server directory path
SYNTAX:	`pwd`
DESCRIPTION:	If an FTP connection has been established with the LOGIN command, the command PWD is used to display the complete path of the current directory on the FTP server.

PWD, PWDCF (NAND Flash)

FUNCTION:	Display the current NAND Flash directory path
SYNTAX:	`pwd, pwdcf`
DESCRIPTION:	To access the NAND Flash when logged on to an ftp server, use the command PWDCF.

RM, RMCF (NAND Flash)

FUNCTION:	Remove a file on the NAND Flash

SYNTAX:	`rm <dirname>`
	`rmcf <dirname>`
	where:
	rm; rmcf command
	<filename> parameter: string
	<[x …]x>
	name of the file to be removed from the NAND Flash

DESCRIPTION:	To access the NAND Flash when logged on to an ftp server, use the command RMCF.

RMDIR, RMDIRCF (NAND Flash)

FUNCTION:	Remove a directory on the NAND Flash

SYNTAX:	`rmdir <dirname>`
	`rmdircf <dirname>`
	where:
	rmdir; rmdircf command
	<dirname> parameter: string
	<[x …]x>
	name of the directory to be removed from the NAND Flash

DESCRIPTION:	To access the NAND Flash when logged on to an ftp server, use the command RMDIRCF. The directory to be removed must be empty prior to removal.

RS

FUNCTION:	Reset the system
SYNTAX:	`rs`
DESCRIPTION:	To permit the operator to force a restart of the system, the command RS is used. This command terminates the NetBootLoader command interpreter and resets the entire system. If this command is issued over a remote telnet connection, the telnet session is terminated prior to the generation of the reset.

RUN (SMP)

FUNCTION:	(Loads and) starts a bootable image
SYNTAX:	`run [-c│<num>]` where: run command -c option: check do not start the image if the “md5” check failed <num> parameter: string 0, 1, 2, 3 ID number of the image to be booted no other values than those above are supported if no image is specified, the image in the data buffer is booted
DESCRIPTION:	This command is used to (load and) start a boot image. If the image to be started is located in the NBL Flash, it is loaded to the data buffer and then started. If the image to be started is already located in the data buffer it is started. When used with the “-c” option, only a valid image is started. Refer to the LOAD command for further information.

RUN (ASMP)

FUNCTION: (Loads and) starts (a) bootable image(s)

SYNTAX:

```
run [-c] [<numcore0> [<numcore1>]]
```

where:

run	command
-c	option: check do not start the image(s) if the “md5” check failed
<numcore0>	parameter: string 0, 1, 2, 3 load and start image with the number <numcore0> from NBL Flash on core 0 4 start image in the data buffer on core 0 - 1 do not start image on core 0
<numcore1>	parameter: string 0, 1, 2, 3 load and start image with the number <numcore0> from NBL Flash on core 1 4 start image in the data buffer on core 1 - 1 do not start image on core 1

DESCRIPTION: This command is used to (load and) start (a) boot images for both cores.

If the image to be started is located in the NBL Flash, it is loaded to the data buffer and then started.

If the image to be started is already located in the data buffer it is started.

When used with the “-c” option, only a valid image is started. If both cores are to be used, then both images must be in order or neither image will be started. Refer to the LOAD command for further information.

Invoking RUN without options starts images in the data buffer.

SAVE

FUNCTION:	Store contents of the data buffer to a file in the NAND Flash
SYNTAX:	**`save -f <filename>`** where: save command -f option: force file overwrite if file exists on NAND Flash it will be overwritten <filename> parameter: string <[x …]x> name of the file under which the intermediate buffer contents are to be stored on the NAND Flash if a file with this name already exists on the NAND Flash the error message: “File filename already exists, use option -f to overwrite” is returned
DESCRIPTION:	This command is used to store the contents of the data buffer (of the currently specified core).

SCRIPT

FUNCTION: Provides very basic scripting capability

SYNTAX:

```
script [<newscript>|-]
```

where:

script	command
<newscript>	parameter: string <[x …]x> “<newscript>” may only include NBL commands; flow control constructs are not permitted; commands must be separated by semi-colons
-	option: delete script contents

DESCRIPTION: With the SCRIPT command, it is possible to control the boot process. During booting, if a valid script is available, the NetBootLoader will process it once the boot wait time is expired.

If this command is issued without any parameters, the currently active script contents are displayed.

If NetBootLoader configuration files (nbl0...9.cfg or other) are executed during the boot process, it is possible that the script specified will not be executed. This applies only to modules which support NAND Flash.

USAGE: Download a boot image from a TFTP server and run the boot image.

COMMAND / RESPONSE (none):

```
script dhcp; tftp; run
```

Upon the next reset or cold start, after the boot wait time has expired the commands DHCP, TFTP, and RUN will be executed in that order.

The above command sequence configures an Ethernet port, downloads the specified bootable image from an TFTP server, and then starts this image.

SF

FUNCTION: Store NBL Flash memory contents to data buffer

SYNTAX:

```
sf (-o[=]<offset> -l[=]<length>)|

…(-m[=]<add> -l[=]<length>)|

…(<num>[ -o[=]<offset>][ -l[=]<length>])
```

where:

sf	command
-o	option: offset
<offset>	parameter: value: hexadecimal relative offset to start of FLASH contents or image to be stored to the data buffer
-l	option: length
<length>	parameter: value: hexadecimal length of FLASH contents or area of visible memory to be stored to the data buffer
-m	option: memory (address)
<adr>	parameter: hexadecimal <[x …]x> absolute address of image to be programmed
<num>	parameter: numerical string 0, 1, 2, 3 number of the image to be stored

DESCRIPTION: With the command SF a selected portion of the NBL Flash memory contents may be copied to the local data buffer (of the currently specified core), e.g. for a subsequent upload to the FTP server with the PUT command.

USAGE: Store 64 kB of NBL Flash memory contents to the data buffer beginning at an offset of 1 MB

COMMAND / RESPONSE (none):

```
sf -o=100000 -l=10000
```

Store FLASH image 2 to the data buffer

COMMAND / RESPONSE (none):

```
sf 2
```

SETCORE (ASMP)

FUNCTION:

Specify the core for which selected commands apply or display the currently selected core

SYNTAX:

```
setcore [<corenum>]
```

where:

setcore	command if command is issued without an argument, the currently set core number is displayed
<corenum>	parameter: string 0, 1 number of the core to be set (default is 0)

DESCRIPTION:

For ASMP operation each core is provided with its own data buffer and BL bootline. To specify operations for a particular core’s data buffer the SETCORE command is used.

The commands: BL, CLONE, FDT_SHOW, GET, LF, LOAD, MD5, PUT, SAVE, SL, SF, and TFTP apply only to the core data buffer specified by the last invocation of the SETCORE command.

SL

FUNCTION:	Download Motorola S-Records to data buffer
SYNTAX:	`sl [-o[=]<offset>] [-u]` where: sl command -o option: offset <offset> parameter: hexadecimal: unsigned <[x …]x> offset to be subtracted from each record's address -u option: source = secondary serial interface instead of the primary serial interface
DESCRIPTION:	With the command SL Motorola S-Records are downloaded to the data buffer (of the currently specified core) and the record addresses modified accordingly as required for DDR2-SDRAM operation (for copying to 0x0). The “<offset>” parameter may be used to change the record base to 0x0. The “-u” option selects the secondary serial interface as source for the S-Records, otherwise the primary serial interface is used.

USAGE:	Download S-Records to data buffer and reduce each record’s address by 0x10000.
	COMMAND / RESPONSE (none):
	`sl -o=10000`

SPEED

FUNCTION: Enables or disables fast booting

SYNTAX:

```
speed [<switch>]
```

where:

speed	command
<switch>	parameter: numeric string 0, 1 when set to 0 (zero) disables “fast” boot mode when set to 1 (one) enables “fast” boot mode when not specified, displays current setting

DESCRIPTION: This command is used to specify to the NetBootLoader that either the complete booting process is to be performed or that a shortened boot process is to be performed.

During the shortened boot process, network initialization is not performed, and CRC checking of the userimage is not performed.

USAGE: Enable “fast” booting

COMMAND / RESPONSE:

```
NetBtLd> speed 1
```

Display the current setting

COMMAND / RESPONSE:

```
NetBtLd> speed

Speed mode : "fast" boot up process.
```

SQ (NON-AMC / SMP)

FUNCTION: Set or display the boot sequence

SYNTAX:

```
sq [<num1><num2><num3><num4>]
```

where:

sq	command
<num1>	parameter: numeric string: “0, 1, 2, 3” ID number of image to be booted
<num2>	parameter: numeric string: “0, 1, 2, 3” ID number of image to be booted
<num3>	parameter: numeric string: “0, 1, 2, 3” ID number of image to be booted
<num4>	parameter: numeric string: “0, 1, 2, 3” ID number of image to be booted

DESCRIPTION: Up to four bootable images may be programmed into the NBL Flash memory. The boot sequence defines to the NetBootLoader the order in which images are to be accessed when booting. The NetBootLoader starts with “<num1>” and continues until a valid image is found. In the case that a valid image is not found, the NetBootLoader stops searching and waits for operator intervention.

All four number parameters must be defined even if there is not an image in the NBL Flash memory with that ID number.

Any given ID number may only be used once: e.g. a sequence of 0120 is not permitted.

The default sequence is 0123 if the boot sequence has not been programmed.

USAGE: Display the current boot sequence setting.

COMMAND / RESPONSE:

```
sq

Bootsequence: 0 - 3 - 1 - 2
```

Set the boot sequence to 3201.

COMMAND / RESPONSE(none):

```
sq 3201
```

SWAP (Two NBL Flashes)

FUNCTION: Modify or display the current NBL Flash memory device usage

SYNTAX:

```
swap [-t|-p]
```

where:

swap	command
-t	option: temporary temporary NBL Flash swap
-p	option: permanent permanent NBL Flash swap

DESCRIPTION: The configuration of the NBL Flash memory devices (active or inactive) is controlled by the IPMI controller, the payload (HOST), and the switch SW2 (DIP switch position 2). The setting combination of all three determines the current configuration.

The SWAP command is a toggle function meaning that when the configuration is to be swapped, the SWAP command is invoked with the “-t” option (temporary, at the next power on the configuration reverts to the previous setting) or the “-p” option (permanently).

The current configuration can be determined by invoking the SWAP command without either option.

If the board is not operated in an ATCA or µTCA environment (e.g. on the AMEVAL1), startup problems may arise due to the simultaneous startup of the payload (CPU) and the MMC. If the assignment is configured as 'IPMI: reverse', the MMC switches the Flashes during payload startup which normally causes a crash.

This can be circumvented by holding the payload startup with a debug probe for a few seconds or by delaying the startup of the MMC (switch 'AMC ENABLE' on the AMEVAL1) and then switching back to 'IPMI: straight' once the system is running.

SWAP (Two NBL Flashes)

USAGE:

Swap the NBL Flashes temporarily and verify

COMMAND / RESPONSE:

```
NetBtLd> swap -t

Flash config swapped till next reboot via
HOST bit

NetBtLd>
NetBtLd> swap

Active   Flash is NOR flash 1
Fallback Flash is NOR flash 0

Configuration Details:
IPMI   : straight
HOST   : reverse
DIP SW : straight == off

NetBtLd>
```

Swap the NBL flashes permanently

COMMAND / RESPONSE:

```
NetBtLd> swap -p

Flash config permanently swapped
NetBtLd>
```

TFTP

FUNCTION:

Download file from a TFTP server

SYNTAX:

```
tftp [-e] [<ip_addr>] [<filename>]
```

where:

tftp	command
-e	option: repeat tftp until download is successful
<ip_addr>	parameter: string <nnn.nnn.nnn.nnn> IP address of the TFTP server
<filename>	parameter: string <x … x> name of image file to be downloaded, or path and name of image file to be downloaded

DESCRIPTION:

The TFTP command makes it possible to download a file to the data buffer (of the currently specified core) from a TFTP server via an Ethernet port. If used with the DHCP command, it is possible to use the IP address and file information returned by the DHCP command.

If this command is issued without any parameters, it will use the previously stored information returned with the DHCP command.

USAGE:

Download a file from a TFTP server.

COMMAND / RESPONSE (none):

```
tftp 195.178.125.55 image2
```

This command downloads the file "image2" from the specified TFTP server.

VER

FUNCTION: Display version number

SYNTAX:

```
ver
```

where:

ver command

DESCRIPTION: The command VER displays the actual version number of the currently active NetBootLoader and when present the inactive NetBootLoader.

USAGE: Determine the current versions of the NetBootLoder:

```
NetBtLd> ver


Version of running NetBootLoader:
0809101408


Version in active flash:   0809101408
No valid NetBootLoader found in fallback -
… flash.
NetBtLd>
```