



Total Performance. Total Control.™

Netronome Flow Engine

Series i8000/i4000

Reference Guide

Product code
020-00001-007

Netronome Flow Engine: Reference Guide

Copyright © 2006-2008 Netronome Systems, Inc.

COPYRIGHT

No part of this publication or documentation accompanying this Product may be reproduced in any form or by any means or used to make any derivative work by any means including but not limited to by translation, transformation or adaptation without permission from Netronome Systems, Inc., as stipulated by the United States Copyright Act of 1976. Contents are subject to change without prior notice.

WARRANTY

Netronome warrants that any media on which this documentation is provided will be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of shipment. If a defect in any such media should occur during this 90-day period, the media may be returned to Netronome for a replacement.

NETRONOME DOES NOT WARRANT THAT THE DOCUMENTATION SHALL BE ERROR-FREE. THIS LIMITED WARRANTY SHALL NOT APPLY IF THE DOCUMENTATION OR MEDIA HAS BEEN (I) ALTERED OR MODIFIED; (II) SUBJECTED TO NEGLIGENCE, COMPUTER OR ELECTRICAL MALFUNCTION; OR (III) USED, ADJUSTED, OR INSTALLED OTHER THAN IN ACCORDANCE WITH INSTRUCTIONS FURNISHED BY NETRONOME OR IN AN ENVIRONMENT OTHER THAN THAT INTENDED OR RECOMMENDED BY NETRONOME.

EXCEPT FOR WARRANTIES SPECIFICALLY STATED IN THIS SECTION, NETRONOME HEREBY DISCLAIMS ALL EXPRESS OR IMPLIED WARRANTIES OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE.

Some jurisdictions do not allow the exclusion of implied warranties, so the above exclusion may not apply to some users of this documentation. This limited warranty gives users of this documentation specific legal rights, and users of this documentation may also have other rights which vary from jurisdiction to jurisdiction.

LIABILITY

Regardless of the form of any claim or action, Netronome's total liability to any user of this documentation for all occurrences combined, for claims, costs, damages or liability based on any cause whatsoever and arising from or in connection with this documentation shall not exceed the purchase price (without interest) paid by such user.

IN NO EVENT SHALL NETRONOME OR ANYONE ELSE WHO HAS BEEN INVOLVED IN THE CREATION, PRODUCTION, OR DELIVERY OF THE DOCUMENTATION BE LIABLE FOR ANY LOSS OF DATA, LOSS OF PROFITS OR LOSS OF USE OF THE DOCUMENTATION OR FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, EXEMPLARY, PUNITIVE, MULTIPLE OR OTHER DAMAGES, ARISING FROM OR IN CONNECTION WITH THE DOCUMENTATION EVEN IF NETRONOME HAS BEEN MADE AWARE OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL NETRONOME OR ANYONE ELSE WHO HAS BEEN INVOLVED IN THE CREATION, PRODUCTION, OR DELIVERY OF THE DOCUMENTATION BE LIABLE TO ANYONE FOR ANY CLAIMS, COSTS, DAMAGES OR LIABILITIES CAUSED BY IMPROPER USE OF THE DOCUMENTATION OR USE WHERE ANY PARTY HAS SUBSTITUTED PROCEDURES NOT SPECIFIED BY NETRONOME.

Revision History

Date	Revision	Description
2 April 2008	007	Updated support for NFD version 1.3.
1 February 2008	006	Updated support for NFD version 1.2.x.
1 July 2007	005	Updated information for NFD version 1.2.
1 March 2007	004	New document layout.
15 September 2006	003	Added specification information.
1 June 2006	002	Moved certain licensed content to separate document.
1 May 2006	001	Initial release.

Table of Contents

1. Introduction	6
1.1. Scope	6
1.2. Related Documents	6
1.3. Terminology	6
2. System Overview	8
2.1. Hardware	8
2.1.1. Network Processor Unit	8
2.1.2. Network Interfaces	8
2.1.3. Field Programmable Gate Array	8
2.1.4. Cryptography	8
2.2. Software	9
2.2.1. MontaVista Linux	9
2.2.2. Ramdisk	9
2.2.3. Sample application – ns_nfe_wire	11
3. Development Environment	12
3.1. Services	12
3.1.1. Dynamic Host Configuration Protocol	12
3.1.2. Trivial File Transfer Protocol	13
3.1.3. Network File System	13
3.2. Targets	13
3.2.1. Microengines	13
3.2.2. XScale processor	13
3.2.3. Host platform	14
3.3. Resource Manager	14
3.4. Preparing IXA SDK 4.2 for use with the NFE platform	14
3.4.1. Patching the IXA SDK	14
4. Boot Monitor and Diagnostics	16
5. User Interface	17
5.1. System Details Page	17
5.2. Logs Page	18
5.3. System Management Page	19
6. Technical Support	21
Technical Specifications	22
A.1. Small Form-factor Pluggable modules	22
A.1.1. 1000Base-T Copper	22
A.1.2. 1000Base-SX	22
A.2. Management Ethernet port (i8000 only)	22
A.3. RS232 Serial port	23
A.4. PCI Express connector	23

A.5. Power consumption 23
A.6. Physical characteristics 23

Hardware Component Specifications 24
B.1. Network Processor Unit 24
B.2. Ternary content addressable memory 24
B.3. Physical random number generator 24
B.4. Modular arithmetic processor 24
B.5. Rambus Dynamic Random Access Memory 25
B.6. Quad Data Rate Static Random Access Memory 25
B.7. Flash memory 25

1. Introduction

1.1 Scope

This manual serves as a reference for the Netronome Flow Engine (NFE) series i8000/i4000. It gives an overview of the hardware- and software components, and their respective subcomponents, that constitute the NFE hardware platform, and discusses the environments in which the platform typically operates.

1.2 Related Documents

Descriptive Name	Filename
Linux Support Package User's Guide	LSP_UG.pdf
Netronome Flow Engine Getting Started Guide	NFE_GSG.pdf
Netronome Flow Engine Boot Monitor and Diagnostics Reference	NFE_BOOT.pdf
IXA SDK 4.2 documentation	-

1.3 Terminology

The following terms and abbreviations are used in this document:

Acronym	Description
API	Application Programming Interface
CPU	Central Processing Unit
DHCP	Dynamic Host Configuration Protocol
FPGA	Field Programmable Gate Array
IA	Intel Architecture
IXP	Intel IXP28xx Network Processing Unit
LSP	Linux Support Package
MSF	Media Switch Fabric
NFD	Netronome Flow Driver
NFE	Netronome Flow Engine
NFS	Network File System
NPU	Network Processing Unit
PCIe	PCI Express

Acronym	Description
QDR	Quad Data Rate
RAM	Random Access Memory
RDRAM	Rambus DRAM
SDK	Software Development Kit
SFP	Small Form-factor Pluggable
SRAM	Static RAM
TCAM	Ternary Content Addressable Memory
TFTP	Trivial File Transfer Protocol
UI	User Interface
XS	XScale processor

2. System Overview

The Netronome Flow Engine (NFE) series utilize the Intel™ IXP28x5 Network Processing Unit (NPU) for accelerated network operations. A Field Programmable Gate Array (FPGA) facilitates the PCI Express (x4) interface to the host platform while an integrated co-processor greatly accelerates asymmetric cryptographic operations. Fast memory, up to 768MB of RDRAM and 40MB of QDR SRAM, provide ample speed and storage space for large volumes of network traffic while a Ternary Content Addressable Memory (TCAM) is provided as standard. Flash memory serves as permanent storage for the NFE embedded operating system and other applications.

2.1 Hardware

2.1.1 Network Processor Unit

The NPU is a complex processor which basically consists of sixteen microengines and a XScale processor. The microengines are very fast “mini-processors” with optimized intercommunications mechanisms to distribute and share network packets between each other effectively at high speed. The microengines can be programmed to execute complex algorithms on network packets. The XScale is an ARM-based micro processor which is typically used for managerial purposes and can execute the Linux operating system.

2.1.2 Network Interfaces

The NPU hosts different communications interfaces, of which one is the Media Switch Fabric (MSF), which connects to the physical network interface controller. The NFE hosts four SFP type network connectors for 1000Base-T (copper) or SX (fiber) connectivity. Refer to ??? for software initialization of these interfaces.

2.1.3 Field Programmable Gate Array

The FPGA utilizes the NPU's QDR interfaces to connect the NPU to the Host platform via the PCI Express interface. The FPGA currently supports 4 active lanes on the PCI Express connector. The FPGA also serves as a “master” thermal fail-safe, in that it will shutdown the whole NFE platform upon detection of an over temperature condition.

2.1.4 Cryptography

The asymmetric cryptographic accelerator together with the integrated symmetric cryptographic unit on the NPU, provide a powerful cryptographic engine for security centric network applications.

2.2 Software

Refer to the *Netronome Flow Engine Boot Monitor and Diagnostics Reference* for a description of the Boot Monitor, Recovery Loader and Diagnostics applications.

2.2.1 MontaVista Linux

The MontaVista™ Professional Edition 3.1 Linux Kernel has been updated to support the integrated XScale processor of the NPU. It is therefore possible to boot the MontaVista™ Linux operating system on the NPU. Refer to the *Linux Support Package User's Guide* for a detailed explanation on how to build a custom MontaVista™ Linux Kernel for the NFE hardware platform. A pre-compiled Linux Kernel is provided by **Netronome Systems** on the distribution media.

The Linux Kernel can be loaded from two storage devices:

- Flash memory
- TFTP server via the management Ethernet interface.

Refer to the *Netronome Flow Engine Boot Monitor and Diagnostics Reference* for a more detailed explanation on how to setup the Boot Monitor boot script to select one of these two options.

2.2.1.1 Flash memory

The Linux Kernel and associated Ramdisk, for the root file system, are stored on the NFE platform in flash memory. This allows for a fast boot time and independent operation of the platform. All new NFE platforms have the Kernel and Ramdisk stored in flash memory and is by default configured to load the Linux Kernel from flash memory. The Linux Kernel initializes the RS232 serial port to provide a console, but the management Ethernet interface is disabled. The Shared Memory VNIC is enabled by default and provides for cable-less Ethernet type communications with the host platform.

2.2.1.2 TFTP server

The Linux Kernel is stored on a TFTP server and loaded into memory by the Boot Monitor via the management Ethernet interface. The Linux Kernel is configured to make use of the Network File System (NFS) service to remotely mount its root file system. The Linux image provides a console via the RS232 serial port and a Telnet service accessible via the management Ethernet interface.

2.2.2 Ramdisk

As described in the previous section, the Linux Kernel is by default loaded from flash memory on the NFE platform. A Ramdisk image, also loaded from flash memory, provides the root file system for the Linux Kernel.

The Ramdisk image provides the following features:

- Persistent storage
- NFS client
- Thermal monitor
- Remote access
- User Interface (see Chapter 5).

The Boot Monitor boot script determines if the Linux Kernel and Ramdisk will be loaded from the flash memory. Refer to the Netronome Flow Engine Boot Monitor and Diagnostics Reference for a more detailed explanation.

2.2.2.1 Persistent Flash Storage

The Ramdisk provides the facility to store data between Kernel reboots. The data is saved into a dedicated partition in the flash memory by the Kernel at shutdown and restored at boot-up. The recommended size for the dedicated flash memory partition is 512kiB. All new NFE platforms has this partition configured.

The Linux Kernel startup script `/etc/init.d/flash`, performs the save and restore operations and is configured by the file `/etc/conf.d/flash`.

Name	Description	Default Value
PARTNAME	The name of the flash memory partition in which the data is stored.	linuxdata
FILES	A space-separated list of files and/or directories to save (Note: items must not contain a leading slash).	root/.ash_history var/log etc/conf.d/hostname etc/dropbear etc/shadow
TEMPFILE	The name of the temporary archive in which the data is stored before it is written to the flash.	/mnt/shm/linuxdata.tgz

The configuration file accepts the following parameters:

2.2.2.2 Network File System Client

The setup and configuration of NFS clients and servers does not fall into the scope of this document. It is however important to mention that all NFS mount points must be mounted into the Ramdisk file system using the “nolock” option since the Ramdisk does not provide the “portmap”-service.

2.2.2.3 Thermal Monitor

A temperature monitor application which constantly monitors the temperature of the NPU, is included in the Ramdisk image and started during the boot sequence as a service.

In the event that the NPU temperature exceeds the maximum temperature threshold, the application will disable the system components in an effort the reduce the power consumption and hence reduce the NPU temperature to

prevent damage. Amber colored LEDs on the NFE platform's faceplate will continuously blink to visually indicate that the platform has been halted. A hardware reset of the NFE platform is required to resume operation.

The platform's FPGA implements a fail-safe temperature monitoring mechanism in case the Linux monitor application has been compromised due to high temperatures. The implementation stops all components on the NFE platform when a critical temperature is reached.

2.2.3 Sample application – ns_nfe_wire

The microengine sample application named `ns_nfe_wire` demonstrate the use of the MSF network interfaces (see Section 2.1.2). It forwards packets between ports 0 and 1 as well as between ports 2 and 3 of the NFE network interfaces. The source code can be found in the NFE tarball or the application can be found at the following location after NFD 1.3 has been installed on the host system:

```
/opt/netronome/nfd/ixp28x5/opt/netronome/nfd/bin
```



Note

It is assumed that NFD was installed to the default installation directory and the IXP samples have been built with the `make ixp_samples` command. Refer to the *Netronome Flow Driver User Guide* on how to install the NFD and to start components.

To execute the sample application:

1. Log into the host system and change to the following directory:

```
/opt/netronome/nfd/bin
```

2. Execute the following script to quickly load all the required Linux Kernel modules:

```
./nfeops.py -f ./nfdops-nfewire.ini -sv
```

3. Observe that all network traffic is forwarded between the respective ports.

3. Development Environment

Refer to the *Netronome Flow Engine Getting Started Guide* for detailed instructions on the installation of the NFE hardware platform into the host platform.

The following Figure 3.1 depicts a typical development environment for the NFE hardware platform. Other configurations are also possible depending on the requirements. For instance, if only the NFE embedded operating system needs to be managed, then a single platform providing an RS232 Serial console (Terminal console) will suffice. However, as for most development cases, it is required that the NFE hardware platform load and boot the Linux operating system, which requires the addition of certain services to the environment as well as the configuration of the NFE hardware platform bootstrap configuration. These extra services must be provided by a dedicated server platform on the LAN.

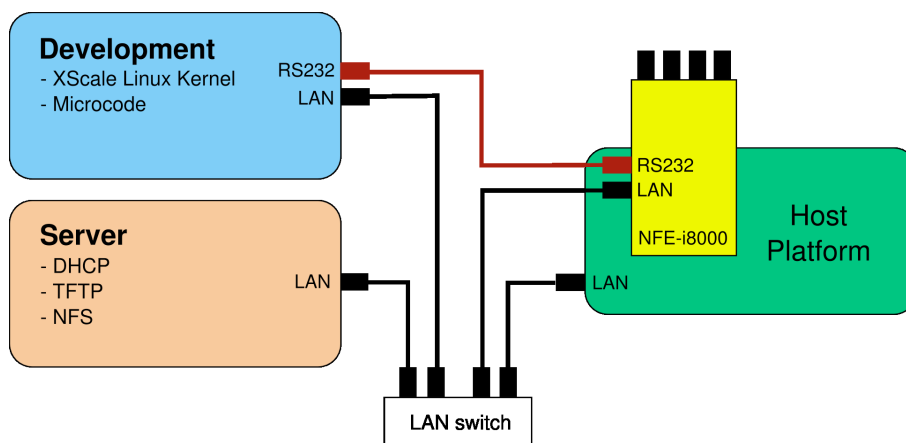


Figure 3.1. Development Environment for the NFE Hardware Platform

3.1 Services

3.1.1 Dynamic Host Configuration Protocol

The Dynamic Host Configuration Protocol (DHCP) service assigns, upon request from a client, a dynamic IP Address which is reserved and unused on the network. Additional information such as routing and other network information can optionally also be provided during the assignment process.

Upon boot-up, the NFE hardware platform Boot Monitor will generate and send a BOOTP request for an IP address, which should be serviced by this service.

It is important that the DHCP service be configured to provide additional information, such as the path of the Network File System (NFS) mount point to the NFE hardware platform when it boots the Linux operating system.

3.1.2 Trivial File Transfer Protocol

The Trivial File Transfer Protocol (TFTP) service transfers, upon request from a client, a specified file to the client. To use this service, a client must have an IP Address, statically assigned or dynamically acquired from a DHCP service, before it can send a file request to this service.

The NFE hardware platform will, after being assigned an IP address and if so configured, download the operating system image file using this service. After the file has been downloaded into memory, the platform will execute the file to boot the operating system. Refer to the *Netronome Flow Engine Boot Monitor and Diagnostics Reference* on how to change the boot configuration.

3.1.3 Network File System

The Network File System (NFS) service provides a remote file system, mounted and accessed across the network. This service makes it possible for remote systems without storage hardware, but connected to a network, to boot the Linux operating system. Files in this file system may be read, written and/or executed by all the clients platforms (permissions permitting) which mount the file system. The **Netronome Systems** provided MontaVista™ Linux Kernel image depends on this service.

3.2 Targets

The development platform (Figure 3.1) serves as an host for the development tools required to develop applications for the NFE host platform's XScale processor and microengines.

3.2.1 Microengines

To develop microengine applications, or microcode, the Intel IXA SDK 4.2 is required in addition to following the steps in Section 3.4. This SDK can be installed on the development platform from where the microcode can be uploaded to the NFE hardware platform with the use of the IXA SDK Workbench.

Refer to the IXA SDK 4.2 documentation for more details.

3.2.2 XScale processor

The NPU's XScale processor is based on the ARM architecture. If the development platform utilizes an Intel Architecture (IA) based processor, then a cross compiler will be required on the development platform.

The MontaVista™ Professional Edition 3.1 Linux installation installs these cross compilers to assist in the build process of a Linux Kernel which can execute on the XScale processor. The cross compilers are therefore available to be used to compile Linux applications for the NFE hardware platform on the IA based development platform.

3.2.3 Host platform

The development of Linux applications for the Host platform can be performed on either the IA based development platform or on the IA host platform if the same operating system is used. If not, the development must preferably occur on the IA host platform.

3.3 Resource Manager

Refer to the Intel IXA SDK 4.2 Software Framework Getting Started Guide for general instruction regarding the compilation of the Intel IXA SDK Resource Manager. Section 9.2.1 of this document indicates the environment variables that are relevant to compilation on the development platform.

The following steps are required to adapt the SDK to compile the Resource Manager to be used on the NFE platform.

1. Change into the framework directory: `cd $IXA_SDK_DEV/src/framework.`
2. Build the kernel with specified parameters:

```
make -f Makefile.linux_kernel HARDWARE_TYPE=IXP2800 APPDEFS="-DIX_PLATFORM_2801"
```

where the parameters, in the last step, are:

- `HARDWARE_TYPE`: sets the NPU architecture family.
- `IX_PLATFORM_2801`: used to disable the PCI communication functionality of the Resource Manager as it is not supported on the NFE platform.

3.4 Preparing IXA SDK 4.2 for use with the NFE platform

This section assumes that `NFE-1.2.x-bin.tar.gz` was extracted to `~/` and that IXA SDK 4.2 was installed to `/opt/ixa_sdk_4.2`.

3.4.1 Patching the IXA SDK

The `ixa_sdk_4.2.diff` patch can be obtained from the `NFE-1.2.x-bin.tar.gz` archive. Copy `ixa_sdk_4.2.diff` to `/opt/ixa_sdk_4.2` and then do:

```
cd /opt/ixa_sdk_4.2
```

```
patch -p1 < ixa_sdk_4.2.diff
```

This will update the ha1mev2 XScale driver to recognize the TCAM and SRAM configuration of the NFE platform.

The `packet_tx_16ports` microcode building block can be obtained from the `NFE-1.2.x-bin.tar.gz` archive under `me/building_blocks/perch/ixa_sdk_4.2/packet_tx_16ports` and should be copied over the original IXA SDK `packet_tx_16port` microcode building block. This can be performed by doing:

```
cp -R ~/nfe-1.2.x/me/building_blocks/perch/ixa_sdk_4.2/packet_tx_16ports  
/opt/ixa_sdk_4.2/src/building_blocks/tx/microengine/packet_tx/
```

The IXA SDK `packet_tx_16port` microcode building block will be replaced with a version that has been updated for use on the NFE platform.

4. Boot Monitor and Diagnostics

Refer to the *Netronome Flow Engine Model Boot Monitor and Diagnostics Reference* for a detailed description of the Boot monitor and Diagnostics commands.

5. User Interface

The User Interface (UI) provides easy access to system information as well as limited remote control over the NFE platform. The remote control functions include the ability to reboot the platform as well as shutdown certain processes. The system information includes temperature values, CPU and memory load as well as version information.

To access the User Interface, point an Internet Browser to the IP Address assigned to the NFE platform management Ethernet interface.

The UI consists of three separate pages shown in Figure 5.1, Figure 5.2 and Figure 5.3 and will be discussed in the following sections. These pages respectively activates via the three buttons named “System Details”, “Logs” and “System Management” present on the default UI page.

5.1 System Details Page

The default page shown in Figure 5.1 presents the current system information. The information displayed are:

- Total- and Free RAM
- Total- and Free RAM-Disk space
- XScale CPU Utilization
- Current-, Minimum- and Maximum NPU temperatures
- The Linux Kernel version
- The Ramdisk version

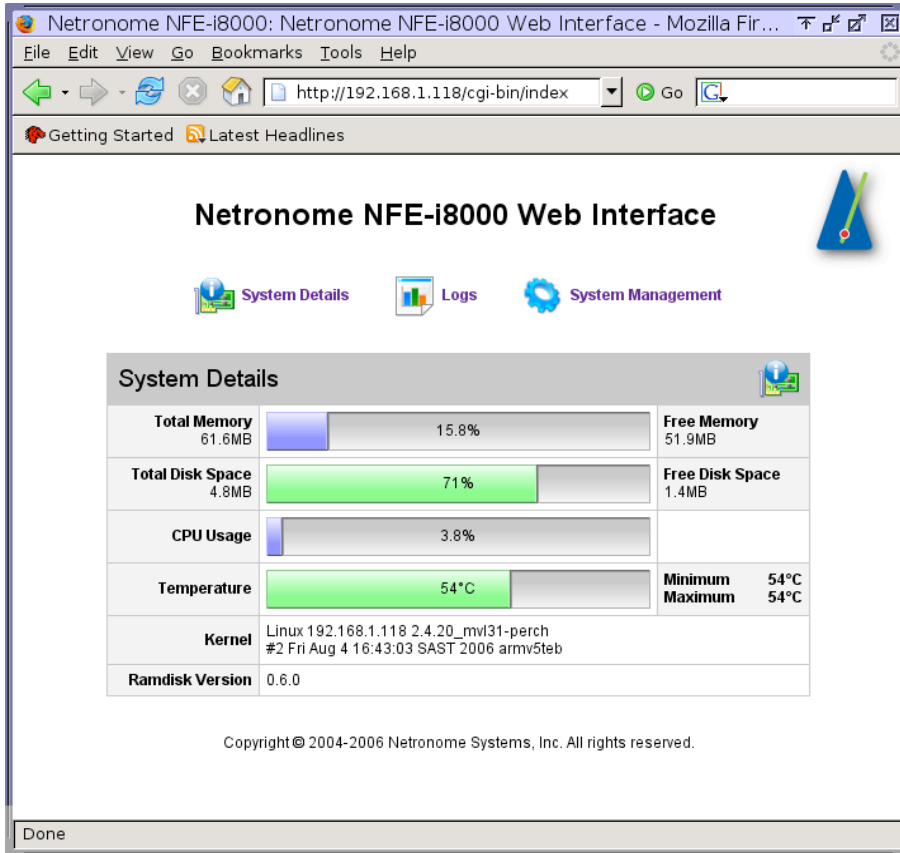


Figure 5.1. System Details page

5.2 Logs Page

The Logs page shown in Figure 5.2 presents the system log file in a date/time/description format. Important system messages as well as the minimum and maximum NPU temperature values are logged. A new log file is created once the current log file reaches a size of 128kB. The system can store up to eight log files.

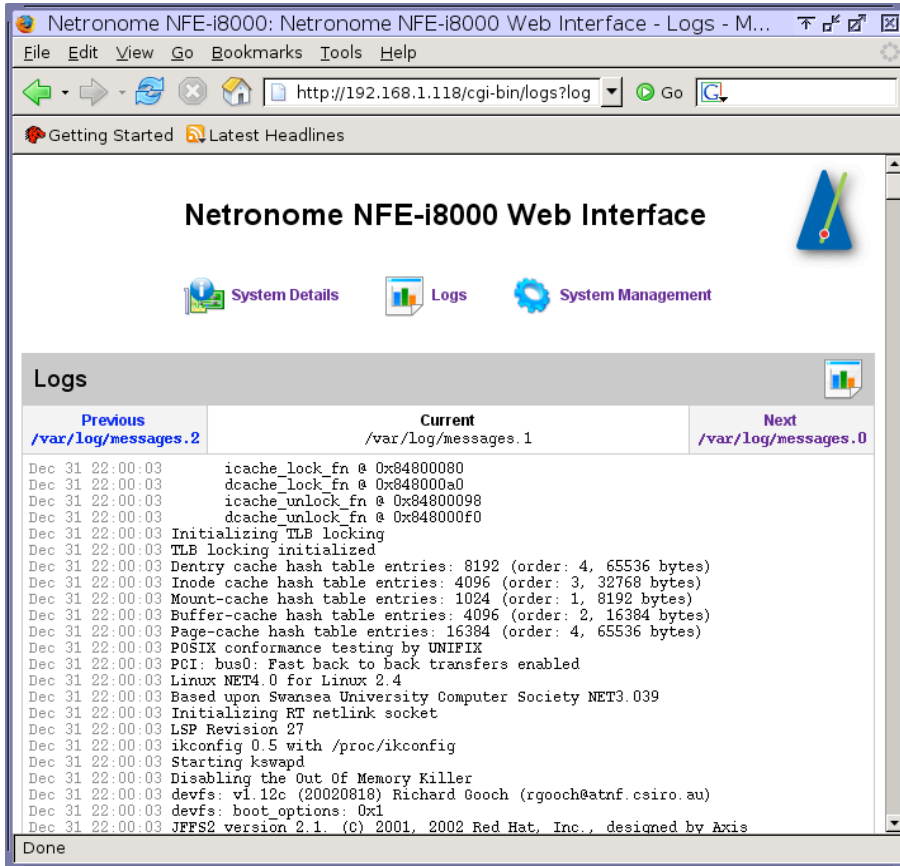


Figure 5.2. Logs page

5.3 System Management Page

The System Management page in Figure 5.3 contains two interactive components which allows the user to:

- Reboot the NFE platform and
- Stop (kill) individual processes.

The first component activates via the button named “Reboot” to **immediately** initiate the reboot of the NFE platform. The second component activates via the button name “Kill” and stops all the processes named in the adjacent input field. To stop more than one process at a time their names must be entered, space separated, into the input field.

Netronome NFE-i8000 Web Interface

System Details Logs System Management

System Management

Reboot system

Kill processes by name
(space-separated list of process names)

Process list

PID	Uid	VmSize	Stat	Command
1	root		SW	[swapper]
2	root		SW	[keventd]
3	root		SWN	[ksoftirqd_CPU0]
4	root		SW	[kswapd]
5	root		SW	[bdflush]
6	root		SW	[kupdated]
8	root		SW	[ntdblockd]
9	root	184	S	init
59	root	228	S	/usr/bin/tempd
70	root	244	S	/bin/sh /etc/init.d/nfdvnet bgwait
81	root	112	S	/usr/sbin/telnetd
86	root	176	S	/sbin/syslogd -m 0 -s 128 -b 8
91	root	172	S	/sbin/klogd
96	root	180	S	/usr/sbin/crond -c /etc/cron.d
101	root	204	S	/usr/sbin/httpd -h /var/www/htdocs
105	root	188	S	init
1176	root	96	S	sleep 1
1177	root	216	S	/usr/sbin/httpd -h /var/www/htdocs
1182	root	128	S	cat
1185	root	168	R	ps w

Copyright © 2004-2006 Netronome Systems, Inc. All rights reserved.

Done

Figure 5.3. System Management page

6. Technical Support

To obtain additional information, or to provide feedback, please email <support@netronome.com> or contact the nearest **Netronome Systems** technical support representative.

Appendix A. Technical Specifications

A.1 Small Form-factor Pluggable modules

The NFE hardware platform has four Small Form-factor Pluggable (SFP) ports which support 1000Base-T (copper) and SX (fiber) modules.

All ports are compliant with Ethernet 802.3 Frame types and the 802.3 Ethernet MAC requirements.

A.1.1 1000Base-T Copper

Port Type	1000Base-T (IEEE 802.3ab)
Connector	RJ45
Media	CAT5e UTP cable (ANSI/EIA/TIA-568-A)
Distance	100 meter (325 feet)

A.1.2 1000Base-SX

Port Type	1000BaseSX
Connector	LC
Media	50/62.5um MM
Distance	500 meter (1625 feet)

A.2 Management Ethernet port (i8000 only)

Model	Intel LU82551ER
Port Type	10/100BASE-TX (IEEE 802.3, IEEE 802.3u)
Connector	RJ45
Media	CAT5e UTP cable (ANSI/EIA/TIA-568-A)
Distance	100meter (325 feet)

A.3 RS232 Serial port

Port Type	RS232
Connector	DB9 Female
Media	Shielded RS232 Cable
Distance	1.82 meter (6 feet)

A.4 PCI Express connector

Support PCI Express v1.1 and uses four active lanes (x4).

A.5 Power consumption

Allowable Input Voltage Ranges		
	Min	Max
12 Volt supply	11.04 Volts	12.96 Volts
3.3 Volt supply	3.0 Volts	3.6 Volts

	Amps @ 12.0 Volts	Amps @ 3.3 Volts
No IXP application (typical current and power)	3.5 amps (42W)	2.5 amps (8.25W)
Typical IXP application	4.4 amps (53W)	2.5 amps (8.25W)
Maximum Netronome observed power	5.1 amps (61.2W)	2.6 amps (8.6W)
Note that NFE power consumption is application dependent and will vary.		

A.6 Physical characteristics

Dimensions	310 mm (12.223") x 102.9 mm (4.051") x 14.47 mm (0.57") With extender bracket: 336 mm (13.213") x 102.9 mm (4.051") x 14.47 mm (0.57")
Weight (Including 4 SPF modules)	With extender bracket: 453 g (16.00oz) Without extender bracket: 448 g (15.80oz)
Temperature	Operating: 0 to 50°C Storage: -10°C to 70°C
Operating Humidity	10% to 90% non-condensing

Appendix B. Hardware Component Specifications

B.1 Network Processor Unit

	NFE-i8000	NFE-i4010	NFE-i4000
Manufacturer	Intel	Intel	Intel
Model	IXP2855	IXP2855	IXP2805
Clock	1.4 GHz	1.0 GHz	1.0 GHz

B.2 Ternary content addressable memory

	NFE-i8000	NFE-i4010	NFE-i4000
Manufacturer	IDT	IDT	IDT
Model	IDT75K62134	IDT75K52134	IDT75K52134
Memory size	9 Mb	4.5 Mb	4.5 Mb

B.3 Physical random number generator

	NFE-i8000	NFE-i4010	NFE-i4000
Manufacturer	FDK	FDK	N/A
Model	RPG100	RPG100	N/A

B.4 Modular arithmetic processor

	NFE-i8000	NFE-i4010	NFE-i4000
Manufacturer	INTEL	INTEL	N/A
Model	IXCP210	IXCP210	N/A

B.5 Rambus Dynamic Random Access Memory

	NFE-i8000	NFE-i4010	NFE-i4000
Rambus Channel 0	256 MB/1016 MHz	128 MB/800 MHz	128 MB/800 MHz
Rambus Channel 1	256 MB/1016 MHz	128 MB/800 MHz	128 MB/800 MHz
Rambus Channel 2	256 MB/1016 MHz	0 MB	0 MB

B.6 Quad Data Rate Static Random Access Memory

	NFE-i8000	NFE-i4010	NFE-i4000
QDR SRAM Channel 0	8 MB/200MHz	8 MB/200MHz	8 MB/200MHz
QDR SRAM Channel 2	16 MB/200MHz	8 MB/200MHz	8 MB/200MHz
QDR SRAM Channel 3	16 MB/200MHz	8 MB/200MHz	8 MB/200MHz

B.7 Flash memory

	NFE-i8000	NFE-i4010	NFE-i4000
Flash memory	16 MB	16 MB	16 MB