



Platform Notes: The eri FastEthernet Device Driver

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A. 650-960-1300

Part No. 816-2127-10
May 2002, Revision A

Send comments about this document to: docfeedback@sun.com

Copyright 2002 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd. For Netscape Communicator™, the following notice applies: (c) Copyright 1995 Netscape Communications Corporation. All rights reserved.

Sun, Sun Microsystems, the Sun logo, AnswerBook2, docs.sun.com, and Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2002 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, Californie 95054 Etats-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd. La notice suivante est applicable à Netscape Communicator™: (c) Copyright 1995 Netscape Communications Corporation. Tous droits réservés.

Sun, Sun Microsystems, le logo Sun, AnswerBook2, docs.sun.com, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REPENDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



Contents

Contents iii

Tables vii

Preface ix

1. The eri Device Driver 1

Hardware Overview 1

Operating Speeds and Modes 1

Auto-Negotiation 2

2. Configuring the Driver Software 3

Configuring the Host File 3

▼ To Configure the Host File 3

Booting From the Network 5

▼ To Boot From the Network 5

Optional Post-Installation Procedures 6

Setting Driver Parameters 6

▼ To Force Network Speed Between 10 Mbps and 100 Mbps 6

3. Parameter Definitions 9

Driver Parameter Values and Definitions 9

Defining the Current Status	11
Inter-Packet Gap Parameters	11
Defining an Additional Delay Before Transmitting a Packet Using lance_mode and ipg0	12
Operational Mode Parameters	13
Operational Mode Priorities	14
Defining the Number of Back-to-Back Packets to Transmit	15
Reporting Transceiver Capabilities	15
Reporting the Link Partner Capabilities	16
4. Setting Parameters	17
Parameter Options	17
Setting Parameters Using ndd	18
Identifying Device Instances	18
▼ To Specify the Device Instance for the ndd Utility	18
Non-Interactive and Interactive Modes	18
Using the ndd Utility in Non-Interactive Mode	19
Using the ndd Utility in Interactive Mode	19
Setting Forced Mode	20
▼ To Select a Transceiver Capability and Set Forced Mode	21
Auto-Negotiation Mode	21
▼ To Set the Mode to Auto-Negotiation	21
Setting Parameters in the /etc/system File	21
▼ To Set ipg1 to 10 and ipg2 to 5 When Rebooting	23
Setting Parameters Using the eri.conf File	24
▼ To Configure Driver Parameters Using eri.conf	24
A. Auto-Negotiation	27
The Auto-Negotiation Protocol	27

Boot Process on the Network	27
Correcting Errors in Negotiating	28
Internal (Local) Transceiver	28

Index	29
--------------	-----------

Tables

TABLE 3-1	<code>eri</code> Driver Parameter, Status, and Descriptions	9
TABLE 3-2	Read-Only Parameters for Defining the Current Status	11
TABLE 3-3	Read-Write Inter-Packet Gap Parameter Values and Descriptions	12
TABLE 3-4	Parameters Defining <code>lance_mode</code> and <code>ipg0</code>	13
TABLE 3-5	Operational Mode Parameters	13
TABLE 3-6	Operational Mode Priorities	14
TABLE 3-7	Back-to-Back Packet Transmission Capability	15
TABLE 3-8	Read-Only Transceiver Capabilities	15
TABLE 3-9	Read-Only Link Partner Capabilities	16
TABLE 4-1	Setting Variables in the <code>/etc/system</code> File	22

Preface

This book describes how to configure the `eri` driver for Sun systems using the Ethernet function of the RIO application specific integrated circuit (ASIC).

How This Book Is Organized

Chapter 1 describes the `eri` device driver and includes topics such as operating speeds and modes, and auto-negotiation.

Chapter 2 describes configuring the `eri` device driver.

Chapter 3 describes the parameter settings for the `eri` device driver.

Chapter 4 describes how to set the `eri` device driver parameter values using the `ndd` utility and also in the `/etc/system` and `/kernel/drv/eri.conf` files.

Appendix A provides information about auto-negotiation.

Using UNIX Commands

This document may not contain information on basic UNIX® commands and procedures such as shutting down the system, booting the system, and configuring devices.

See one or more of the following for this information:

- Online documentation for the Solaris™ software environment
- Other software documentation that you received with your system

Typographic Conventions

Typeface or Symbol	Meaning	Examples
AaBbCc123	The names of commands, files, and directories; on-screen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. % You have mail.
AaBbCc123	What you type, when contrasted with on-screen computer output	% su Password:
<i>AaBbCc123</i>	Book titles, new words or terms, words to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> be superuser to do this.
	Command-line variable; replace with a real name or value	To delete a file, type <code>rm filename</code> .

Shell Prompts

Shell	Prompt
C shell	<i>machine_name%</i>
C shell superuser	<i>machine_name#</i>
Bourne shell and Korn shell	\$
Bourne shell and Korn shell superuser	#

Accessing Sun Documentation Online

The `docs.sun.com`SM web site enables you to access Sun technical documentation on the Web. You can browse the `docs.sun.com` archive or search for a specific book title or subject at: <http://www.docs.sun.com>

Sun Welcomes Your Comments

We are interested in improving our documentation and welcome your comments and suggestions. You can email your comments to us at:

`docfeedback@sun.com`

Please include the part number (816-2127-10) of this document in the subject line of your email.

The `eri` Device Driver

The `eri` device driver handles the `SUNW,eri` device on Sun systems using the RIO ASIC.

This chapter includes the following sections:

- “Hardware Overview” on page 1
- “Operating Speeds and Modes” on page 1
- “Auto-Negotiation” on page 2

Hardware Overview

The `SUNW,eri` device provides a 100BASE-TX network interface using the Ethernet function of the RIO ASIC. The driver automatically sets the link speed to 10 or 100 Mbps and conforms to the *100BASE-T IEEE 802.3u Ethernet Standard*. The RIO ASIC provides the PCI interface and Media Access Control (MAC) functions. The internal transceiver, which connects to an RJ-45 connector, provides the physical layer functions.

The RIO ASIC is a chip set composed of an I/O chip and a single chip Ethernet transceiver; the `eri` device driver uses the Ethernet function of this ASIC.

Operating Speeds and Modes

You can operate the link in any of the following speeds and modes with the `SUNW,eri` device:

- 100 Mbps, full-duplex

- 100 Mbps, half-duplex
- 10 Mbps, full-duplex
- 10 Mbps, half-duplex

The *100BASE-T IEEE 802.3u Ethernet Standard* describes these speeds and modes.

Auto-Negotiation

A key feature of the Sun `eri` FastEthernet driver is auto-negotiation. The auto-negotiation protocol, as specified by the *100BASE-T IEEE 802.3u Ethernet Standard*, selects the operation mode (half-duplex or full-duplex), and the auto-sensing protocol selects the speed (10 Mbps or 100 Mbps) for the adapter. Further information can be found in Appendix A, “Auto-Negotiation” on page 27.

Configuring the Driver Software

This chapter includes information and instructions for configuring the driver software used by the Sun `eri` FastEthernet PCI adapter.

This chapter includes the following sections:

- “Configuring the Host File” on page 3
- “Booting From the Network” on page 5
- “Optional Post-Installation Procedures” on page 6

Configuring the Host File

The 64-bit driver is included with the Solaris CD.

Before using `eri` as your network interface, you will need to create and edit system host files, as described in the next section.

▼ To Configure the Host File

1. **At the command line, use the `grep` command to search the `/etc/path_to_inst` file for `eri` devices. For example:**

```
# grep eri /etc/path_to_inst
"/pci@8,700000/network@5,1" 0 "eri"
```

2. Create an `/etc/hostname.erinum` file, where *num* is the instance number of each interface you plan to use.

If you want to use the network interface from the example in Step 1, you will need to create a file:

File Name	Instance Number
<code>/etc/hostname.eri0</code>	0

- Do not create `/etc/hostname.erinum` files for Sun `eri` FastEthernet network interfaces you plan to leave unused.
- The `/etc/hostname.erinum` file must contain the host name for the appropriate network interface.
- The host name should have an IP address that will need to be entered in the `/etc/hosts` file.
- The host name should be different from any other host name of any other interface, for example: `/etc/hostname.hme0` and `/etc/hostname.eri0` cannot share the same host name.

Using the instance examples in Step 1, the following example shows the two `/etc/hostname.erinum` files required for a system called `zardoz` that has a Sun `eri` FastEthernet (`zardoz`, `zardoz-11`).

```
# cat /etc/hostname.hme0
zardoz
# cat /etc/hostname.eri0
zardoz-11
```

3. Create an appropriate entry in the `/etc/hosts` file for each active `eri` network interface.

Using the previous example, you will have:

```
# cat /etc/hosts
#
# Internet host table
#
127.0.0.1    localhost
129.144.10.57 zardoz    loghost
129.144.11.83 zardoz-11
```

Note – The Internet Protocol, version 6 (IPv6), expands the capabilities of IPv4, which is the current version and the default. The Sun `eri` FastEthernet device driver included in this release of the Solaris operating environment supports both IPv4 and IPv6. IPv4 uses the `/etc/hosts` configuration file, but IPv6 uses a different configuration file. To transition to, manage, and implement IPv6, refer to the Solaris System Administration documentation.

4. Reboot your system.

Booting From the Network

To use a Sun `eri` interface as the boot device, perform the following tasks:

▼ To Boot From the Network

1. At the `ok` prompt type:

```
ok show-nets
```

The `show-nets` command lists the system devices. You should see the full path name of the `eri` devices, similar to the following examples:

```
/pci@8,700000/network@5,1
```

Note – You need to select only one of these `eri` devices for booting.

2. At the `ok` prompt type:

```
ok boot full_path_name_of_the_eri_device
```

Optional Post-Installation Procedures

To customize the performance of the Sun `eri` FastEthernet driver, perform the tasks in the following sections.

Setting Driver Parameters

The `eri` device driver, which is loaded from the Solaris CD-ROM, controls the `SUNW,eri` Ethernet devices. The device driver selects the link speed using the auto-negotiation protocol with the link partner.

You can manually set the `eri` device driver parameters to customize each `SUNW,eri` device in your system in one of three ways.

- Set a parameter on a per-device basis by creating the `eri.conf` file in the `/kernel/drv` directory.
- Use the `ndd` utility to temporarily change a parameter. This change is lost when you reboot the system.
- Set the `eri` driver parameters generally for all `SUNW,eri` devices in the system by entering the parameter variables in the `/etc/system` file.

See Chapter 4 “Setting Parameters” for more information.

▼ To Force Network Speed Between 10 Mbps and 100 Mbps

1. At the `ok` prompt, use the `show-nets` command to list the system devices.

You should see the full path names of the `eri` devices, similar to the following example:

```
/pci@8,700000/network@5,1
```

2. Type:

```
ok nvedit
```

3. Type the following, pressing the Return key at the end of line 0:

```
0: probe-all install-console banner
1: apply transfer-speed=10 full_path_name_of_a_eri_device
```

Note – If you already have commands in NVRAM, append these lines to the end of the file.

4. Press Control-C after typing *full_path_name_of_a_eri_device*.

Perform Steps 2 to 4 to set the network speed for each `eri` network interface.

Note – In the preceding example, the speed is forced to 10 Mbps. To force the speed to 100 Mbps, replace 10 with 100.

5. At the `ok` prompt type:

```
ok nvstore
ok setenv use-nvramrc? true
```

6. Reboot your system.

See Setting Forced Mode for more information on forcing network speed.

Parameter Definitions

This chapter describes the parameters and settings for the `eri` device driver.

Driver Parameter Values and Definitions

The following sections describe the permitted values for the `eri` driver parameters, which are listed in “Driver Parameter Values and Definitions” on page 9.

- “Defining the Current Status” on page 11, Table 3-2.
- “Inter-Packet Gap Parameters” on page 11, Table 3-3.
- Define extra delay see “Defining an Additional Delay Before Transmitting a Packet Using `lance_mode` and `ipg0`” on page 12, Table 3-4.
- Operational mode parameters see “Operational Mode Parameters” on page 13, Table 3-5.
- Operational mode parameters see “Operational Mode Priorities” on page 14, Table 3-6.
- “Defining the Number of Back-to-Back Packets to Transmit” on page 15, Table 3-7.
- Reporting transceiver capabilities see “Reporting Transceiver Capabilities” on page 15, Table 3-8.
- “Reporting the Link Partner Capabilities” on page 16, Table 3-9.

TABLE 3-1 `eri` Driver Parameter, Status, and Descriptions

Parameter	Status	Description	Details
<code>transceiver_inuse</code>	Read only	Defines the current status	
<code>link_status</code>	Read only	Defines the current status	“Defining the Current Status” on page 11
<code>link_speed</code>	Read only	Defines the current status	
<code>link_mode</code>	Read only	Defines the current status	

TABLE 3-1 eri Driver Parameter, Status, and Descriptions (Continued)

Parameter	Status	Description	Details
ipg1	Read and write	Inter-packet gap parameter	“Inter-Packet Gap Parameters” on page 11
ipg2	Read and write	Inter-packet gap parameter	
pace_size	Read and write	Operational mode parameter	“Defining the Number of Back-to-Back Packets to Transmit” on page 15
adv_autoneg_cap	Read and write	Operational mode parameter	“Operational Mode Parameters” on page 13
adv_100fdx_cap	Read and write	Operational mode parameter	
adv_100hdx_cap	Read and write	Operational mode parameter	
adv_10fdx_cap	Read and write	Operational mode parameter	
adv_10hdx_cap	Read and write	Operational mode parameter	
autoneg_cap	Read only	Local transceiver auto negotiation capability	“Reporting Transceiver Capabilities” on page 15
100fdx_cap	Read only	Local transceiver capability of the hardware	
100hdx_cap	Read only	Local transceiver capability of the hardware	
10fdx_cap	Read only	Local transceiver capability of the hardware	
10hdx_cap	Read only	Local transceiver capability of the hardware	
lp_autoneg_cap	Read only	Link partner auto negotiation capability	“Reporting the Link Partner Capabilities” on page 16
lp_100fdx_cap	Read only	Link partner capability	
lp_100hdx_cap	Read only	Link partner capability	
lp_10fdx_cap	Read only	Link partner capability	
lp_10hdx_cap	Read only	Link partner capability	

TABLE 3-1 eri Driver Parameter, Status, and Descriptions (Continued)

Parameter	Status	Description	Details
instance	Read and write	Device instance	
lance_mode	Read and write	Additional delay before transmitting a packet	“Defining an Additional Delay Before Transmitting a Packet Using lance_mode and ipg0” on page 12
ipg0	Read and write	Additional delay before transmitting a packet	

Defining the Current Status

The read-only parameters described in the following table explain the operational mode of the interface. These parameters define the current status.

TABLE 3-2 Read-Only Parameters for Defining the Current Status

Parameter	Description	Values
link_status	Current link status	0 = Link down 1 = Link up
link_speed	Valid only if the link is up	0 = 10 Mbps 1 = 100 Mbps
link_mode	Valid only if the link is up	0 = Half duplex 1 = Full duplex

Inter-Packet Gap Parameters

The Ethernet function unit of RIO ASIC supports programmable Inter-Packet Gap (IPG) parameters `ipg1` and `ipg2`. The total IPG is the sum of `ipg1` and `ipg2`. The total IPG is 9.6 microseconds when the link speed set by the auto-negotiation protocol is 10 Mbps. When the link speed is 100 Mbps, the total IPG is 0.96 microseconds.

The following table lists the default values and allowable values for the IPG parameters, `ipg1` and `ipg2`.

TABLE 3-3 Read-Write Inter-Packet Gap Parameter Values and Descriptions

Parameter	Values (Byte-time)	Description
<code>ipg1</code>	0, 255	<code>ipg1 = 8</code> (default at initialization)
<code>ipg2</code>	0, 255	<code>ipg2 = 4</code> (default at initialization)

By default, the driver sets `ipg1` to 8-byte time and `ipg2` to 4-byte time, which are the standard values. (Byte time is the time it takes to transmit one byte on the link, with a link speed of either 100 Mbps or 10 Mbps.)

If your network has systems that use longer IPG (the sum of `ipg1` and `ipg2`) and if those machines seem to be slow in accessing the network, increase the values of `ipg1` and `ipg2` to match the longer IPGs of other machines.

Defining an Additional Delay Before Transmitting a Packet Using `lance_mode` and `ipg0`

The Ethernet function unit of RIO ASIC supports a programmable mode called `lance_mode`. The `ipg0` parameter is associated with `lance_mode`.

After a packet is received with `lance_mode` enabled (default), an additional delay is added by setting the `ipg0` parameter before transmitting the packet. This delay, set by the `ipg0` parameter, is in addition to the delay set by the `ipg1` and `ipg2` parameters. The additional delay set by `ipg0` helps to reduce collisions. Systems that have `lance_mode` enabled might not have enough time on the network.

If `lance_mode` is disabled, the value of `ipg0` is ignored and no additional delay is set. Only the delays set by `ipg1` and `ipg2` are used. Disable `lance_mode` if other systems usually send a large number of back-to-back packets.

You can enable the additional delay by setting the `ipg0` parameter from 0 to 31, which is the nibble time delay. Nibble time is the time it takes to transfer four bits on the link. If the link speed is 10 Mbps, nibble time is equal to 400 ns. If the link speed is 100 Mbps, nibble time is equal to 40 ns.

For example, if the link speed is 10 Mbps, and you set `ipg0` to 20 nibble times, multiply 20 by 400 ns to get 8000 ns. If the link speed is 100 Mbps, and you set `ipg0` to 30 nibble times, multiply 30 by 40 ns to get 1200 ns.

The following table defines the `lance_mode` and `ipg0` parameters.

TABLE 3-4 Parameters Defining `lance_mode` and `ipg0`

Parameter	Values	Description
<code>lance_mode</code>	0	<code>lance_mode</code> disabled
	1	<code>lance_mode</code> enabled (default)
<code>ipg0</code>	0-31 — See the following Note.	Additional IPG before transmitting a packet (after receiving a packet)

Note – The default value of `ipg0` is 16 nibble-times, which is 6.4 microseconds for 10 Mbps and 0.64 microseconds for 100 Mbps.

Operational Mode Parameters

The following table describes the operational mode parameters and their default values.

TABLE 3-5 Operational Mode Parameters

Parameter	Description	Values
<code>adv_autoneg_cap</code>	Local transceiver capability advertised by the hardware	0 = Forced mode 1 = Auto-negotiation (default)
<code>adv_100fdx_cap</code>	Local transceiver capability advertised by the hardware; read/write parameter	0 = Not 100 Mbit/sec full-duplex capable 1 = 100 Mbit/sec full-duplex capable (default)

TABLE 3-5 Operational Mode Parameters *(Continued)*

Parameter	Description	Values
adv_100hdx_cap	Local transceiver capability advertised by the hardware; read/write parameter	0 = Not 100 Mbit/sec half-duplex capable 1 = 100 Mbit/sec half-duplex capable (default)
adv_10fdx_cap	Local transceiver capability advertised by the hardware; read/write parameter	0 = Not 10 Mbit/sec full-duplex capable 1 = 10 Mbit/sec full-duplex capable (default)
adv_10hdx_cap	Local transceiver capability advertised by the hardware; read/write parameter	0 = Not 10 Mbit/sec half-duplex capable 1 = 10 Mbit/sec half-duplex capable (default)

Operational Mode Priorities

Operational Mode Priorities lists the priority of the operational mode parameters. When the eri driver negotiates with a partner on the network, it implements the operational mode in the order shown in Operational Mode Priorities.

TABLE 3-6 Operational Mode Priorities

Priority	Parameter
First	adv_100fdx_cap
Second	adv_100hdx_cap
Third	adv_10fdx_cap
Fourth	adv_10hdx_cap

Defining the Number of Back-to-Back Packets to Transmit

The `pace_size` parameter (see the following table) defines the maximum number of back-to-back packets you can transmit at one time. If the value is zero, there is no limit to the number of back-to-back packets that can be transmitted..

TABLE 3-7 Back-to-Back Packet Transmission Capability

Parameter	Values	Description
<code>pace_size</code>	1-255	Number of back-to-back packets transmitted at one time
	0	No limit to the number of back-to-back packets that can be transmitted (default)

Reporting Transceiver Capabilities

The following table describes the read-only transceiver capabilities. These parameters define the capabilities of the hardware. The local transceiver can support all of these capabilities.

TABLE 3-8 Read-Only Transceiver Capabilities

Parameter	Description	Values
<code>autoneg_cap</code>	Local transceiver capability of the hardware	0 = Not capable of auto-negotiation 1 = Auto negotiation capable
<code>100fdx_cap</code>	Local transceiver capability of the hardware; initialized at startup	0 = Not 100 Mbit/sec full-duplex capable 1 = 100 Mbit/sec full-duplex capable
<code>100hdx_cap</code>	Local transceiver capability of the hardware; initialized at startup	0 = Not 100 Mbit/sec half-duplex capable 1 = 100 Mbit/sec half-duplex capable
<code>10fdx_cap</code>	Local transceiver capability of the hardware; initialized at startup	0 = Not 10 Mbit/sec full-duplex capable 1 = 10 Mbit/sec full-duplex capable
<code>10hdx_cap</code>	Local transceiver capability of the hardware; initialized at startup	0 = Not 10 Mbit/sec half-duplex capable 1 = 10 Mbit/sec half-duplex capable

Reporting the Link Partner Capabilities

The following table describes the read-only link partner capabilities.

TABLE 3-9 Read-Only Link Partner Capabilities

Parameter	Values	Description
lp_autoneg_cap	0=	No auto-negotiation
	1=	Auto-negotiation
lp_100fdx_cap	0=	No 100Mbit/sec full-duplex transmission
	1=	100Mbit/sec full-duplex
lp_100hdx_cap	0=	No 100Mbit/sec half-duplex transmission
	1=	100Mbit/sec half-duplex
lp_10fdx_cap	0=	No 10Mbit/sec full-duplex transmission
	1=	10Mbit/sec full-duplex
lp_10hdx_cap	0=	No 10Mbit/sec half-duplex transmission
	1=	10Mbit/sec half-duplex

If the link partner is not capable of auto-negotiation (when `lp_autoneg_cap` is 0), the information described in the previous table is not relevant and the parameter value = 0.

If the link partner is capable of auto-negotiation (when `lp_autoneg_cap` is 1) then the speed and mode information is displayed when you use auto-negotiation and get the link partner capabilities.

Setting Parameters

This chapter describes how to configure the `eri` driver parameters. Use the `ndd` utility to configure parameters that are valid until you reboot the system.

To configure the `eri` driver parameters for all devices in the system so that the parameter values are always in effect (even after rebooting the system), enter the parameter values in the `/etc/system` file. When the system is rebooted, it reads the `/etc/system` file and sets the parameter values in that file.

To set the parameters for a particular device in the system, set the parameters in the `eri.conf` file in the `/kernel/drv` directory. The parameters set in the `eri.conf` file have precedence over the parameters set in the `/etc/system` file and override the parameters set in the `/etc/system` file. The parameters values set in `eri.conf` are always in effect (even after rebooting the system).

Parameter Options

You can set the `eri` device driver parameters in three ways (`ndd`, `/etc/system`, and `eri.conf`), depending on your needs. To set parameters that are valid until you reboot the system, use the `ndd` utility. Using `ndd` is a good way to test parameter settings.

To set parameters so they remain in effect after you reboot the system:

- Add the parameter values to `/etc/system` when you want to configure parameters for all devices in the system.
- Create the `eri.conf` file and add parameter values to `eri.conf` when you need to set a particular parameter for a device in the system.

If you want to test parameter settings, use the `ndd` utility described in “Setting Parameters Using `ndd`” on page 18. With `ndd`, the parameters are effective until you reboot the system. To make the parameter settings permanent, enter the values in `/etc/system` or `eri.conf` as described in this chapter.

Setting Parameters Using `ndd`

Use the `ndd` utility to configure parameters that are valid until you reboot the system. The `ndd` utility supports any networking driver, which implements the Data Link Provider Interface (DLPI).

The following sections describe how you can use the `eri` driver and the `ndd` utility to modify (with the `-set` option) or display (without the `-set` option) the parameters for each `SUNW,eri` device.

Identifying Device Instances

Before you use the `ndd` utility to get or set a parameter for the `eri` device, you must specify the device instance for the utility if there is more than one `SUNW,eri` device.

Note – If there is only one `SUNW,eri` device, the device is automatically chosen by the `ndd` utility.

▼ To Specify the Device Instance for the `ndd` Utility

1. Check the `/etc/path_to_inst` file to identify the instance associated with a particular device.
2. Use that instance number to select the device as follows:

```
% ndd -set /dev/eri instance instance number
```

The device remains selected until you change the selection.

Non-Interactive and Interactive Modes

You can use the `ndd` utility in two modes:

- Non-interactive
- Interactive

In non-interactive mode, you invoke the utility to execute a specific command. Once the command is executed, you exit the utility. In interactive mode, you can use the utility to get or set more than one parameter value. (Refer to the `ndd (1M)` man page for more information.)

Using the `ndd` Utility in Non-Interactive Mode

1. **To modify a parameter value, use the `-set` option.**

If you invoke the `ndd` utility with the `-set` option, the utility passes *value*, which must be specified down to the named `/dev/eri` driver instance, and assigns it to the parameter:

```
% ndd -set /dev/eri parameter value
```

2. **To display the value of a parameter, specify the parameter name (and omit the value).**

When you omit the `-set` option, a query operation is assumed and the utility queries the named driver instance, retrieves the value associated with the specified parameter, and prints it:

```
% ndd /dev/eri parameter
```

Using the `ndd` Utility in Interactive Mode

1. **To modify a parameter value in interactive mode, specify `ndd eri`, as shown below.**

The `ndd` utility then prompts you for the name of the parameter:

```
% ndd /dev/eri  
name to get/set? (Enter the parameter name or ? to view all parameters)
```

After you enter the parameter name, the `ndd` utility prompts you for the parameter value (see “Driver Parameter Values and Definitions” on page 9 for parameter descriptions).

2. To list all the parameters supported by the `eri` driver, type:

```
% ndd /dev/eri \?
```

(See “Driver Parameter Values and Definitions” on page 9 for parameter descriptions.)

CODE EXAMPLE 4-1 Example of Listing All Parameters Supported by the `eri` Driver

```
example # ndd /dev/eri \?  
?  
transceiver_inuse      (read only)  
link_status           (read only)  
link_speed            (read only)  
link_mode             (read only)  
ipg1                  (read and write)  
ipg2                  (read and write)  
use_int_xcvr          (read and write)  
pace_size             (read and write)  
adv_autoneg_cap       (read and write)  
adv_100fdx_cap        (read and write)  
adv_100hdx_cap        (read and write)  
adv_10fdx_cap         (read and write)  
adv_10hdx_cap         (read and write)  
autoneg_cap           (read only)  
100T4_cap             (read only)  
100fdx_cap            (read only)  
100hdx_cap            (read only)  
10fdx_cap             (read only)  
10hdx_cap             (read only)  
lp_autoneg_cap        (read only)  
lp_100fdx_cap         (read only)  
lp_100hdx_cap         (read only)  
lp_10fdx_cap          (read only)  
lp_10hdx_cap          (read only)  
instance              (read and write)  
lance_mode            (read and write)  
ipg0                  (read and write)  
example #
```

Setting Forced Mode

The procedure that follows describes how to set forced mode (not capable of auto-negotiation).

▼ To Select a Transceiver Capability and Set Forced Mode

1. **Select one of the following capabilities:** `adv_100fdx_cap`, `adv_100hdx_cap`, `adv_10fdx_cap`, **or** `adv_10hdx_cap`, **and set its value to 1.**

If you select more than one of the local transceiver capabilities, the driver selects the one that is highest in the priority order.

2. **Set the local transceiver capabilities advertised by the hardware to forced mode = 0, which is not capable of auto-negotiation:** `adv_autoneg_cap 0`

Use the `ndd` utility as described in “Using the `ndd` Utility in Interactive Mode” on page 19 in this chapter.

Auto-Negotiation Mode

▼ To Set the Mode to Auto-Negotiation

1. **Select at least one of the four capabilities** (`adv_100fdx_cap`, `adv_100hdx_cap`, `adv_10fdx_cap`, `adv_10hdx_cap`) **that you want to advertise to the remote system, and set its value to 1.**

2. **Set the local transceiver capabilities advertised by the hardware to 1, the auto-negotiation setting:** `adv_autoneg_cap 1`

Use the `ndd` utility as described in “Using the `ndd` Utility in Interactive Mode” on page 19 in this chapter.

Setting Parameters in the `/etc/system` File

To configure the `eri` driver parameters for all `SUNW,eri` devices in the system so that the parameter variables are always effective (even after rebooting the system), enter the parameter variables in the `/etc/system` file. When you reboot the system, the system reads the `/etc/system` file and sets these parameter variables in the `eri` module in the operating system kernel.

The following table lists the variables you need to set in the `/etc/system` file.

TABLE 4-1 Setting Variables in the `/etc/system` File

Parameter	Values	Description
<code>ipg1</code>	0, 255	<code>ipg1</code> = 8 (default at initialization)
<code>ipg2</code>	0, 255	<code>ipg2</code> = 4 (default at initialization)
<code>pace_size</code>	1-255	Number of back-to-back packets transmitted at one time
	0	No limit to the number of back-to-back packets that can be transmitted (default)
<code>adv_autoneg_cap</code>	Local transceiver capability advertised by the hardware	0 = Forced mode 1 = Auto-negotiation (default)
<code>adv_100fdx_cap</code>	Local transceiver capability advertised by the hardware; read/write parameter	0 = Not 100 Mbit/sec full-duplex capable 1 = 100 Mbit/sec full-duplex capable (default)
<code>adv_100hdx_cap</code>	Local transceiver capability advertised by the hardware; read/write parameter	0 = Not 100 Mbit/sec half-duplex capable 1 = 100 Mbit/sec half-duplex capable (default)
<code>adv_10fdx_cap</code>	Local transceiver capability advertised by the hardware; read/write parameter	0 = Not 10 Mbit/sec full-duplex capable 1 = 10 Mbit/sec full-duplex capable (default)

TABLE 4-1 Setting Variables in the `/etc/system` File (Continued)

Parameter	Values	Description
<code>adv_10hdx_cap</code>	Local transceiver capability advertised by the hardware; read/write parameter	0 = Not 10 Mbit/sec half-duplex capable 1 = 10 Mbit/sec half-duplex capable (default)
<code>lance_mode</code>	0	<code>lance_mode</code> disabled
	1	<code>lance_mode</code> enabled (default)
<code>ipg0</code>	0-31 — Note: The default value is 16 nibble-times, which is 6.4 microseconds for 10 Mbps and 0.64 microseconds for 100 Mbps	Additional IPG before transmitting a packet (after receiving a packet)

These parameter values, described in Chapter 3, are applicable to all SUNW,eri devices on the system. See TABLE 3-2 through TABLE 3-9 for parameter descriptions. An example follows.

▼ To Set `ipg1` to 10 and `ipg2` to 5 When Rebooting

1. Become superuser.
2. Add the following lines to the `/etc/system` file:

```
set eri:ipg1 = 10
set eri:ipg2 = 5
```

3. Save the `/etc/system` file.
4. Save all files and exit all programs. Exit the windowing system.
5. Reboot the system by typing `init 6` at the superuser prompt.
The system is halted and then rebooted.

Setting Parameters Using the `eri.conf` File

You can also specify the properties described in the section, “Setting Parameters in the `/etc/system` File” on page 21,” in this chapter on a per-device basis by creating the `eri.conf` file in the `/kernel/drv` directory. The properties set in the `eri.conf` file will override the parameters set in the `/etc/system` file. Use `eri.conf` when you need to set a particular parameter for a device in the system. The parameters you set are read and write parameters that are listed in Chapter 3.

The man pages for `prtconf (1M)`, `system (4)` and `driver.conf (4)` include additional details. An example follows:

▼ To Configure Driver Parameters Using `eri.conf`

1. Obtain the hardware path name for the device in the device tree.

Typically this path name and the associated instance number will be present in the `/etc/path_to_inst` file. For example, on a Sun Blade 1000 PCI system, the `/etc/path_to_inst` file will have the following entry:

```
"/pci@8,700000/network@5,1" 0 "eri"
```

- In the entry:
 - The first part within the double quotes specifies the hardware node name in the device tree.
 - The second number is the instance number.
 - The last part in double quotes is the driver name.
- In the device path name, the last component after the last `/` character and before the `@` character is the device name.
- The path name before the last component is the parent name.

- The comma-separated numbers after the @ character at the end represent the device and function numbers, which are together referred to as unit-address.

To identify a PCI device unambiguously in the `eri.conf` file, use the name, parent name, and the unit-address for the device. Refer to the `pci(4)` man page for more information about PCI device specification.

In the first line of the previous example:

- Name = `eri`
- Parent = `/pci@8,700000`
- Unit-address = `5,1`

2. Set the `ipg1` and `ipg2` parameters for the above device in the `/kernel/drv/eri.conf` file:

```
name = "eri" parent = "/pci@8,700000" unit-address = "5,1" ipg1=10 ipg2=5;
```


Auto-Negotiation

An understanding of auto-negotiation will help you manage your Sun *eri* FastEthernet interface. Auto-negotiation automates the speed and mode of the interface. Auto-negotiation detects the link partner capabilities and accommodates to them.

The Auto-Negotiation Protocol

Autonegotiation is a key feature of the Sun *eri* FastEthernet driver. The auto-negotiation protocol, as specified by the *100BASE-T IEEE 802.3u Ethernet Standard*, selects the operation mode (half-duplex or full-duplex), and the auto-sensing protocol selects the speed (10 Mbps or 100 Mbps) for the adapter.

Boot Process on the Network

The auto-negotiation protocol does the following when the system is booted:

- Identifies all link partner-supported modes of operation
- Advertises its capabilities to the link partner
- Selects the highest common denominator mode of operation based on the following priorities (in decreasing order):

Priority	Line Speed and Mode
First	100 Mbps, full-duplex
Second	100 Mbps, half-duplex
Third	10 Mbps, full-duplex
Fourth	10 Mbps, half-duplex

The link partner is the networking device (system, Ethernet hub, or Ethernet switch) at the other end of the link or cable.

Correcting Errors in Negotiating

If the `SUNW,eri` device is connected to a remote system or interface that is not capable of auto-negotiation, your system automatically selects the correct speed and half-duplex mode.

If the Sun `eri` FastEthernet is connected to a link partner with which the auto-negotiation protocol fails to operate successfully, you can configure the device so it does not use this protocol. This forces the driver to set up the link in the mode and speed of your choice.

Internal (Local) Transceiver

The internal transceiver also supports a forced mode of operation. This is where the user selects the speed and mode using the `ndd` utility, by editing the `/etc/system` file, or by creating an `eri.conf` file in the `kernal/drv/` directory. The `ndd` utility makes calls to the `eri` driver to choose the speed and mode.

Index

A

Auto-Negotiation

hardware characteristics, 2

`eri.conf`, 17, 24
`path_to_inst`, 3

B

Booting From the Network, 5

H

Hardware Overview, 1
host file, configuring, 3

C

configuring the driver software, 3
configuring the host file, 3

I

instance, 18
instance number, 4

D

device, 18, 24
devices, 3
DLPI
Data Link Provider Interface, 18
Driver Parameter Values and Definitions, 9

N

`ndd` Utility in Interactive Mode, 19
`ndd` Utility in Non-Interactive Mode, 19
`nvedit` command, 6
`nvstore` command, 7

F

Files

`/etc/hostname.erinum`, 4
`/etc/hosts`, 4
`/etc/path_to_inst`, 18, 24
`/etc/system`, 17, 21, 23
`/kernel/drv/eri.conf`, 25

O

Operating Speeds and Modes, 1
Operational Modes
Auto-Negotiation Mode, 21
Setting Forced Mode, 20
Optional Post-Installation Procedures, 6

P

Parameter Definitions, 9

Parameter Options, 17

Parameters

100fdx_cap, 10, 15

100hdx_cap, 10, 15

10fdx_cap, 10, 15

10hdx_cap, 10, 15

adv_100fdx_cap, 10, 13

adv_100hdx_cap, 10, 14

adv_10fdx_cap, 10, 14

adv_10hdx_cap, 10, 14

adv_autoneg_cap, 10, 13

autoneg_cap, 10, 15

Defining an Additional Delay, 12

Defining the Current Status, 11

instance, 11

Inter-Packet Gap Parameters, 11

ipg0, 11, 13

ipg1, 10, 12

ipg2, 10, 12

lance_mode, 11, 13

link_mode, 9, 11

link_speed, 9, 11

link_status, 9, 11

lp_100fdx_cap, 10, 16

lp_100hdx_cap, 10, 16

lp_10fdx_cap, 10, 16

lp_10hdx_cap, 10, 16

lp_autoneg_cap, 10, 16

Number of Back-to-Back Packets to Transmit, 15

Operational Mode, 13

pace_size, 10, 15

Reporting the Link Partner Capabilities, 16

Reporting Transceiver Capabilities, 15

Setting in `/etc/system` file

adv_100fdx_cap, 22

adv_100hdx_cap, 22

adv_10fdx_cap, 22

adv_10hdx_cap, 23

adv_autoneg_cap, 22

ipg0, 23

ipg1, 22

ipg2, 22

lance_mode, 23

pace_size, 22

transceiver_inuse, 9

parent name, 24

Priorities

Operational Mode Priorities, 14

S

Setting Driver Parameters, 6

Setting Parameters, 17

Setting Parameters in the `/etc/system` File, 21

Setting Parameters Using `ndd`, 18

Setting Parameters Using the `eri.conf` File, 24

`show-devs` command, 6

`show-nets` command, 5

T

The `eri` Device Driver, 1

U

unit-address, 25