DP83820 GigMAC LINUX Release Notes

Version 1.10b, Release 1.6a

9/18/2002

Read This Document Before Attempting To Install Or Use This Product!

This document contains information about factors that must be considered before, during, and after installation.

General Notice:

Table of Contents

National Semiconductor DP83820 Gigabit Ethernet Adapter

1.0 Introduction

This document presents information to users about National Semiconductor's DP83820 Gigabit Ethernet adapter and the relevant LINUX software driver version 1.10b for LINUX operating system kernel version 2.2.x and 2.4.x.

2.0 Product Overview

The DP83820 is a PCI Gigabit Ethernet adapter from National Semiconductor that supports a 33/66MHz, 32/64 bit PCI 2.1/2.2 bus interface. It complies with the emerging 1000 Mbps Ethernet 802.3z specification and supports full duplex operation.

Other features include:

• Maximized PCI efficiency and performance

• Pause Frames

• Checksum Offloading

• Jumbo Frames

• Wake-on-LAN

• VLAN support

The driver for this adapter on LINUX is designed to be a NIC driver for processing Ethernet data packets.

3.0 Installation

3.1 Release Media

The release media consists of the National Semiconductor's DP83820 GigMAC LINUX driver available on disk, on the National Semiconductor website and will soon be available on the LINUX for Redhat CD.

3.2 Installation Procedure

This section describes the installation of the adapter and the driver software for making the adapter functional on LINUX platforms.

3.2.1 Hardware Installation

This section describes the installation of the GigMAC demo card into a personal computer (PC). To install the card, you need:
• GigMAC demo card
• PC with an available PCI slot
• Screwdriver to open the PC and secure the GigMAC demo card
• Category 5 twisted-pair RJ-45 cable
• Wrist strap
• Blank, formatted floppy disk
Installing the card consists of one optional and three required stages: Copying the Driver to a Floppy Disk (optional), Inserting the Card, Connecting the Network, and Installing the Driver. If the PC into which you are installing the GigMAC demo card is equipped with a CD-ROM, you may skip the first stage and begin with stage II.

Inserting the Card
The second stage in the installation sequence is inserting the card into the PC. For this stage you will need the GigMAC demo card and the screwdriver. Unplug the PC before opening it to avoid electrocution. Be sure to ground yourself before handling the GigMAC card to avoid ElectroStatic Discharge (ESD) damage to the card.

1. Shutdown the PC, and disconnect the power cord.

2. Open the chassis of the PC.

3. Choose an unused PCI slot and remove its metal bracket by loosening the screw on the inside. You will no longer need this metal bracket as the GigMAC demo card has its own. You will need the screw, however.

4. Remove the GigMAC demo card from the ESD-safe packaging.
Warning: Static charge from your body can permanently damage the GigMAC demo card. Do not handle the card without first electrically grounding yourself via wrist strap or by touching a large piece of metal.

5. Insert the GigMAC demo card into the empty PCI slot in the PC. Orient the card so that the RJ-45 connector points out of the computer.
Warning: The card may fit backwards (with the RJ-45 pointing in) into the PCI slot. Starting the computer with the card inserted backwards may damage the card and/or the PC.

6. When you are certain the card is oriented correctly, press firmly on the card to ensure that it is completely seated in the PCI slot. Secure the GigMAC demo card using the screw from step 3.

7. Close the computer, and reconnect the power cord.

Connecting the Network

The third stage in the installation sequence is connecting the GigMAC demo card to the network. In this stage, you will need only the twisted-pair Category 5 network cable.

After you have inserted the card and closed the PC, connect one end of the twisted-pair cable to the RJ-45 connector on the GigMAC demo card, and the other end to another PC or a network hub or switch.

When connecting the card to another PC, the cable will need to be *crossover*, meaning that pins 1 and 2 must be swapped with pins 3 and 6, respectively, between the two ends of the cable. A crossover cable should not be used when connecting the GigMAC demo card to a network hub or switch.

The next time you turn on the PC, one of the two green link lights on the GigMAC demo card should become illuminated almost immediately. This signifies that the card has established link with the device at the other end of the cable and is ready to send and receive data.

3.2.2 Software Installation and Configuration

The fourth stage in the installation sequence is installing the driver for the GigMAC demo card. The following subsections describe the procedure for installing the driver for LINUX. The driver is now distributed in an RPM format and is extracted as follows: copy the .rpm file into a clean directory onto the LINUX system and type rpm -i <filename> . This command will result in an extraction of the source code to a subdirectory named /nsc.

LINUX Driver Installation

Follow this sequence for installation under LINUX.

1. To freshly make the driver

   go to the /nsc directory

   and execute "make"

2. To remove the binaries

   execute "make clean"

3. If the machine on which the testing is done is an SMP machine, please include the flag -D__SMP__ in the CFLAGS of the Makefile.

4. After the dpm.o (The driver module) is made, to load the driver

execute the command "insmod dpm". We can uload the driver by doing "rmmod dpm".

5. After loading the driver, get the interface name(eg: eth0, eth1 etc) for the device. The last line in the messages listed will indicate the name of the interface for the driver. If the message is not coming on the console, we can see the messages in the kernel buffer by executing "dmesg" on the command line. The last line in that will indicate the interface name

6. To configure IP for the interface.

ifconfig <interface-name> <ip-addr>

eg: ifconfig eth0 10.10.10.1

To shutdown the interface.

ifconfig <interface-name> down

eg: ifconfig eth0 down

Please note that it may appear to hang on doing an "ifconfig", but it is just momentary and it will soon come out of that. The link becomes active only after about 20 seconds since the "ifconfig".

7. Now we are all set to go with the driver.

8. To make the driver load when the system boots up, the following lines needs to be put in any of the startup scripts like /etc/rc.d/rc.local or /etc/rc.d/init.d/network

/sbin/insmod /working/oct1/dpm.o
/sbin/ifconfig <interface-name> <ip-addr>

9. The tunable parameters of the driver, can be given at load time as parameters to "insmod".

eg: insmod dpm.o MaxTxDesc=150 MaxRxDesc=150 RxBufSize=2048

The various tunable parameters are:

(a) MaxTxDesc

This is the maximum number of transmit descriptors that will be allocated. This value should be at least 100. The default value is 200.

(b) MaxRxDesc

This is the maximum number of receive descriptors that will be allocated. This value should be at least 100. The default value is 200.

(c) RxBufSize

This is the size of the receive descriptors that are allocated. The default value is 2048. It should be 2048 for good performances. The values of this should be 4096 when using jumbo frames to get better performances.

(d) SetAutoNeg

This is the flag to request Auto Negotiation. A value of 1 will enable auto negotiation and a value of 0 indicates disabling auto negotiation, in this scenario the MediaSpeed can be set appropriately to force the Link speed to the desired value. Default value is 1.

(e) MediaSpeed

This value indicates the MediaSpeed to which the NIC should be forced in the case of Auto Negotiation being disabled. This value assumes significance only when SetAutoNeg = 0. The default is 1000.

(f) NCBit

If IEEE Compliance then NCBit should have value 1 else 0. Default value is 1.

10.To try jumbo frames, the mtu size can be increased using the ifconfig utility, as follows

ifconfig <interface-name> mtu <mtu-size>
eg: ifconfig dpm0 mtu 3000

The "RxBufSize" tunable parameter should have a value of 4096 to get better performance when using jumbo frames.

<mtu-size> can be increased upto 4000 bytes for sane operation.

11.The failure messages, like failure in allocating resources like receive buffers etc can be printed if the driver is made with the FAILURE_MESSAGES cflag in the makefile.

12.The checksum offload feature can be turned on by compiling the driver with the CHECKSUM cflag in the makefile.

13.The driver can have both memory mapped and programmed i/o. This can be done by compiling the driver with the appropriate cflags in the makefile for memory mapped I/O the cflag used should be MEMMAPPED_IO and for programmed I/O it should be IOMAPPED_IO. In systems like the HP KAYAK, the memory mapped I/O may not work, and we may have to use programmed I/O.

3.2.3 Installation Troubleshooting

3.2.4 Disk Space Requirements

3.2.5 Errata

This section describes the features and issues to be provided upon final release of the driver.

If any messages beginning with "ASSERTION FAILED" are seen, it is an erroneous behavior.

4.0 Product Documentation

This release notes document, *National Semiconductor Corp DP83820 Gigabit Ethernet Adapter Driver Release Notes*, provides detailed information about installing the National Semiconductor Corp DP83820 Gigabit Ethernet Adapter and Driver software.

5.0 Problem Reporting

National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, CA 95051, U.S.A.
Tel: 1-800-272-9959
Fax: 1-800-737-7018
Email: support@nsc.com
WWW: www.national.com

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32

National Semiconductor Asia Pacific
Customer Response Group
Tel: 65-254-4466
Fax: 65-250-4466
Email: sea.support@nsc.com

National Semiconductor Japan Ltd.
Tel: 81-3-5620-6175
Fax: 81-3-5620-6179