

# Broadcom 440X Fast Ethernet Controller Engineering Diagnostics User's Guide

This chapter provides the following information:

[Introduction](#)

[Prerequisites](#)

[Diagnostic Tests](#)

[Test and Functions Description](#)

---

## Introduction

This document provides information on the b44diag.exe diagnostic program for the Broadcom 440X Fast Ethernet Controller. When the b44diag.exe program is started, a series of tests are executed on the 440X Fast Ethernet controller. If a test fails, the b44diag program displays an error and exits to DOS.

The b44diag.exe program can be run in two different modes:

- Manufacturing mode
- Engineering mode

When the b44diag program is run in engineering mode, it prompts the user for commands. In manufacturing mode, the following options are available:

-l <file>	Log data to file
-c <num>	Specify the card to be tested
-i <num>	Iteration number
-t <id>	Disable tests
-T <id>	Enable tests
-com <value>	Comm port enable (internal use only)
-r <num>	Input radix
-n	Run in manufacture loop
-e	Run program in engineering mode
-m	Program the MAC address

-f<filename>	Program eeprom content from bin before testing
-mac	Program the MAC address from the command line
-fmac <filename>	Program the MAC address from the text file through the command line
-b <num>	Enables/disables the Boot ROM 0 = Disable 1 = 16 KB 2 = 32 KB 3 = 64 KB 4 = 128 KB
-lbm	Option to specify the number of packets in the MAC Loopback test
-lbp	Option to specify the number of packets in the PHY Loopback test
-lbh	Option to specify the number of packets in the 100BT External Loopback test
-lbt	Option to specify the number of packets in the 10BT External Loopback test

**Example:**

```
>b44diag -e XXXX -c 0 -f eeprom.bin -fmac macaddr.txt -b 3 -t abc
```

-e: enter engineering mode.

-c: card select.

-f: program the SROM with eeprom.bin. Basic information is loaded to the SROM. The MAC address remains the same. Error if the file name is missing.

-fmac: program the MAC address from the MAC address file. Only the MAC address is updated. Error if the file name is missing.

-b: enable 64 KB boot ROM. 0 = disable, 1 = 16 KB, 2 = 32 KB, 3 = 64 KB, 4 = 128 KB. The Boot ROM status remains unchanged if the -b option is not entered.

-t: disable Test Group A, B, and C.

After programming the SROM with the bin file (the -f option is entered), power cycle the system to have the new SROM configuration loaded. No test is run after programming the SROM. The software disables all tests after the SROM is programmed and exits to DOS.

**MAC Address Programming Option**

Instead of entering the -fmac option, the MAC address in the SROM can also be programmed by entering either of the -m and Mac options. By entering the -m option, the user is prompted to enter the MAC address. To use the Mac option, the user needs to provide the MAC address after entering the Mac option.

**Example:** b44diag -e XXXXXX -c 0 -f eeprom.bin Mac 001018112240 -b 3 (power cycle required, and no test is run after programming the SROM).

```
> b44diag -e XXXXXX -c 0 -fmac macaddr.txt
```

```
> b44diag -e XXXXXX -m
```

XXXXX = password.

If the password is valid, the user is prompted to enter a 12-byte MAC address. The NIC card is programmed with the new MAC address before running any test.

```
> b44diag -e XXXXX -f eeprom.bin
```

The NIC card is programmed with the new content from eeprom.bin before the test.

```
> b44diag -l test.log -c 1 -l 2 -t A3
```

```
> b44udiag -l test.log -c 1 -l 2 -t A3 -lbn 3000 -lbp 3000 -lbh 3000 -lbt 3000
```

---

## Prerequisites

**OS:** DOS 6.22

**Software:** b44diag.exe

---

## Diagnostic Tests

There are three groups of tests, and each group has some sub-tests.

### Group A

A1. [Indirect Control Register Test](#)

A2. [Direct Control Register Test](#)

A3. [Interrupt Test](#)

A4. [Built-in Self Test](#)

### Group B

B1. [LEDs Test](#)

B2. [EEPROM Test](#)

B3. [MII Test](#)

B4. [Link Status Test](#)

## Group C

C1. [MAC Loopback Test](#)

C2. [PHY Loopback Test](#)

C3. [External Loopback Test 100BT](#)

C4. [External Loopback Test 10BT](#)

## Test Descriptions

### A1. Indirect Control Register Test

**Command:** regtest -i

**Function:** Each register that is specified in the configuration contents read only bit and read/write bit defines. The test writing 0 and 1 by using the indirect addressing method into the test bits ensures that the read only bits are not changed, and read/write bits are changed accordingly.

**Default:** Enabled

### A2. Direct Control Register Test

**Command:** regtest

**Function:** Each register that is specified in the configuration contents read only bit and read/write bit defines. The test writing 0 and 1 into the test bits ensures that the read only bits are not changed, and read/write bits are changed accordingly.

**Default:** Enabled

### A3. Interrupt Test

**Command:** intrtest

**Function:** Verifies the interrupt functionality by enabling interrupt, and waits for an interrupt to occur. It waits for 500 ms and reports an error if it cannot generate interrupts.

**Default:** Enabled

### A4. Built-In Self Test

**Command:** bist

**Function:** Runs the Built-in Self test.

**Default:** Enabled

## **B1. LED Test**

**Command:** ledtest

**Function:** Tests forcing of the link state for each link speed/duplex.

**Default:** Enabled

## **B2. EEPROM Test**

**Command:** setest

**Function:** Reads the Serial Prom and verifies the integrity by checking CRC.

**Default:** Enabled

## **B3. MII Test**

**Command:** miitest

**Function:** Each register that is specified in the configuration contents read only bit and read/write bit defines. The test writing 0 and 1 into the test bits ensures that the read only bits value are not changed, and read/write bits are changed accordingly.

**Default:** Enabled

## **B4. Link Status Test**

**Command:** linkstatus

**Function:** Reports the current link status.

**Default:** Enabled

## **C1. MAC Loopback Test**

**Command:** lbtest -m

**Function:** Transmits a 2000 or specified by -lbn option of 1514-byte packets with incrementing data pattern, and checks tx and rx flags and data integrity.

**Default:** Enabled

## **C2. PHY Loopback Test**

**Command:** lbtest -p

**Function:** This test is same as the [MAC Loopback Test](#), except that the data is routed back via physical layer device.

**Default:** Enabled

### C3. External Loopback Test 100BT

**Command:** lbtest -e

**Function:** This test is same as the [MAC Loopback Test](#), except that the data is routed back via a loopback device

**Default:** Disabled

### C4. External Loopback Test 10BT

**Command:** lbtest -a

**Function:** This test is same as the [MAC Loopback Test](#), except that the data is routed back via loopback device.

**Default:** Disabled

By default, all tests except C3 and C4 are covered in manufacturing mode unless disabled by the user.

The Engineering mode can be selected by option -b44eng.

#### Example:

```
> b44diag -b44eng
```

---

## Test and Functions Description

When the program is in engineering mode, it prompts the commands to be entered. The following section lists all the commands.

### lbtest

**cmd:** lbtest

**Description:** Performs various loopback tests.

**Syntax:** lbtest [ n | c | t ] [ m | p | a | e ]

'n' = iteration

'c' = maximum packet count

't' = packet type: 0 = all 0s, 1 = all 1s, 2 = 5555, 3 = AAAA, 4 = 0F0F, 5 = F0F0, 6 = FF00, 7 =

00FF, 8 = FFFF0000, 9 = 0000FFFF, 10 = Inc, 11 = Random  
 'm' = MAC Loopback  
 'p' = Phy Loopback  
 'e' = 100BT External Loopback  
 'a' = 10BT External Loopback

Default maximum packet count = 2000  
 Default iteration = 1  
 Default pattern = inc.

**Example:**

```
0:>lbtest -e -n=10 -c=2500 -t=3 (10 times external loopback test with
2500 packets, and pattern is AAAA)
```

**phyctrl**

**cmd:** phyctrl

**Description:** Configures speeds/duplex of PHY.

**Syntax:** phyctrl [ s ] [ h ] [ r ] [ f ]

's' = 0:10 Mbps, 1:100 Mbps  
 'h' = half-duplex  
 'r' = reset phy  
 'c' = force  
 'f' = write phy initialization scripts

**Example:**

```
0:> phyctrl -s=0 -h (10 Mbps half-duplex) initialization scripts
0:> phyctrl -s=1 -h -c (force 100BT half-duplex)
```

**loadd**

**cmd:** loadd

**Description:** Loads the default chip setting before the blast.

**Syntax:** loadd

**blast**

**cmd:** blast

**Description:** The Blast Packets in Poll mode.

**Syntax:** blast [ t | r | h ] [ n ] [ l ] [ i ] [ e ]

't' = TX

'r' = RX  
 'h' = host loop back (with minimum 17.6 usec ipg)  
 'n' = number of packets to transmit  
 'l' = transmit packets size (minimum = 60)  
 'i' = increment transmit packets length  
 'e' = Upper limit of TX buffer in incremental packet size  
 'p' = packet type: 0 = all 0s, 1 = all 1s, 2 = 5555, 3 = AAAA, 4 = 0F0F, 5 = F0F0, 6 = FF00, 7 = 00FF, 8 = FFFF0000, 9 = 0000FFFF, 10 = Inc, 11 = Random  
 'd' = Interpacket GAP in microseconds

**Example:**

```

0:> blast -t -r -p=11 -l=1514 (RX and TX packet with 1514 bytes of
random data)
0:> blast -t -n=10000 -l=1514 (TX 10000 packets with size of 1514 bytes
of default pattern)
0:> blast -t -n=10000 -l=1514 -p=3 (TX 10000 packets with size of 1514
bytes of AAAA pattern)
0:> blast -t -n=10000 -i -e=1514 (TX 10000 packets with inc size of
default pattern)
0:> blast -t (TX packets with size of 64 bytes of default pattern until
stop)
0:> blast -r (RX packets until stop)
0:> blast -h (with minimum 17.6 usec ipg)
  
```

\* Blast does not reset the chip anymore. The user must use the Reset command to reset the chip, and the load command to set up the chip to the default state.

**Example:**

```

0:> reset (reset chip)
0:> load (set chip to default state)
0:> do abc.do (run script or write register if needed)
0:> blast -t -r -p=11 -l=1514 (RX and TX packet with 1514 bytes of
random data)
  
```

To stop, press the Esc key.

**nicstats**

**cmd:** nicstats

**Description:** Displays the NIC statistics.

**Syntax:** nicstats [c]

c = reset counters

**Example:**

```

0:> nicstats (display NIC statistics)
0:> nicstats -c (reset counters)
  
```

**setest**



**cmd:** setest

**Description:** Serial EEPROM read/write test. The serial EEPROM tests dump the contents of the serial EEPROM to the screen, and verifies the data with a CRC check.

**Syntax:** setest [iteration]

**Example:**

1. Display Help.

```
0:> setest ?
```

```
Usage : setest [iteration]
```

```
Description:
```

```
The default iteration is 1. 0 means run forever
```

## mread

**cmd:** mread

**Description:** Read PHY registers via MII.

**Syntax:** mread <begin\_addr>[ | <len>]

Address range: 0x00 – 0x1F

**Example:**

1. Read MII register 0

```
0:> mread 0
00: 1100
```

2. Read MII registers 0 to 10

```
0:> mread 0-10

00: 1100 7949 0020 6051 01e1 0000 0004 2001

08: 0000 0300 0000 0000 0000 0000 0000 3000

10: 0002
```

3. Read 5 MII registers start from register

```
0:> mread 0 5
```

```
00: 1100 7949 0020 6051 01e1
```

## mwwrite

**cmd:** mwrite

**Description:** Write PHY registers via MII.

**Syntax:** mwrite <addr> <value>

Address range: 0x00 – 0x1F

**Example:**

1. Write 0x15 to MII register 2

```
0:> mwrite 2 15
```

## miitest

**cmd:** miitest [iteration]

**Description:** PHY registers read write test.

**Syntax:** miitest

## read

**cmd:** read

**Description:** Generic Memory read.

**Syntax:** read [!|S|X|#|m|\$||s|x]<begin\_addr> [- end\_addr | num\_bytes]

! = Configuration space (address range: 0x00 – 0xFF) (32)

S = Configuration space (address range: 0x00 – 0xFF) (16)

X = Configuration space (address range: 0x00 – 0xFF) (16)

\$ = Serial EEPROM

m = MII Registers

l = Direct access (dword)

s = Direct access (word)

x = Direct access (byte)

## write

**cmd:** write

**Description:** Generic Memory write.

**Syntax:** write [!|S|X|#|\$||s|x]<begin\_addr> [- end\_addr ] <value>

! = Configuration space (address range: 0x00 – 0xFF) (32)

S = Configuration space (address range: 0x00 – 0xFF) (16)

X = Configuration space (address range: 0x00 – 0xFF) (16)

\$ = Serial EEPROM

l = Direct access (dword)

s = Direct access (word)

x = Direct access (byte)

## intrtest

**cmd:** intrtest

**Description:** Interrupt test.

**Syntax:** intrtest

## regtest

**cmd:** regtest

**Description:** MAC registers read write test.

**Syntax:** regtest [<iteration>]

## pciscan

**cmd:** pciscan

**Description:** Scan for all PCI devices.

**Syntax:** pciscan

**Example:**

```
0:> pciscan
```

```
Scanning PCI devices ...
```

Bus	Dev	Func	Vendor ID	Device ID	Class	Base/IO Address	IRQ
0	0	0	8086	7190	06:00:00	00000000:F8000008	0
0	1	0	8086	7191	06:04:00	00000000:00000000	0
0	7	0	8086	7110	06:01:00	00000000:00000000	0
0	7	1	8086	7111	01:01:80	00000000:00000000	0
0	7	2	8086	7112	0C:03:00	00000000:00000000	9
0	7	3	8086	7113	06:80:00	00000000:00000000	0
0	14	0	12AE	0003	02:00:00	00000000:F4000004	10
1	0	0	1002	4742	03:00:00	00009001:F5000000	11

## dos

**cmd:** DOS

**Description:** Enter to DOS shell.

**Syntax:** DOS

**Example:**

```
0:> DOS
```

## pciinit

**cmd:** pciinit

**Description:** Initialize PCI configuration registers

**Syntax:** pciinit

**Example:**

```
0:misc> pciinit
Initializing PCI Configuration Space
Bus Number      : 0
Device/Function : 14/0
Base Address    : 0xf4000004
```

## q

**cmd:** q

**Description:** Exits.

**Syntax:** q

## exit

**cmd:** exit

**Description:** Exits.

**Syntax:** exit

## help

**cmd:** help

**Description:** Displays help.

**Syntax:** help

## log

**cmd:** log

**Description:** Logs data to file.

**Syntax:** log <filename>

**Example:**

```
0:> log test.log
```

```
started logfile 'test.log'
```

## nolog

**cmd:** nolog

**Description:** Closes the current log file.

**Syntax:** nolog

**Example:**

```
0:> nolog
```

```
logfile closed at Mon Mar 4 15:25:11 2002
```

## reset

**cmd:** reset

**Description:** Resets the chip.

**Syntax:** reset

**Example:**

```
0:> reset
```

## teste

**cmd:** teste

**Description:** Enables tests in the test configuration.

**Syntax:** teste <group><tests index>

**Example:**

```
0:> teste A23
```

**Enabled Tests:**

- A2 Control Register Test
- A3 Interrupt Test

## testd

**cmd:** testd

**Description:** Disables the tests in the test configuration.

**Syntax:** testd <group><tests index>

**Example:**

```
0:> testd A23
```

**Disabled Tests:**

- A2 Control Register Test

- A3 Interrupt Test

## nicetest

**cmd:** nicetest

**Description:** Runs tests in configuration.

**Syntax:** nicetest

## cls

**cmd:** cls

**Description:** Clears screen.

**Syntax:** cls

## loop

**cmd:** loop

**Description:** Runs cmd n times.

**Syntax:** loop [iteration] <cmd> [<parameter> ...]

**Example:**

```
0:> loop 3 miitest (run miitest 3 times)
```

## mrloop

**cmd:** mrloop

**Description:** A special test routine for MII read that loops on MII register read until it is aborted, or if the value is 0.

**Syntax:** mrloop <addr>

**Example:**

```
0:> mrloop 02 (Loop on MII read at reg 02)
```

## inp

**cmd:** inp

**Description:** Reads port input.

**Syntax:** inp <addr>

## outp

**cmd:** outp

**Description:** Writes to port.

**Syntax:** outp <addr> <data>

## linkstatus

**cmd:** linkstatus

**Description:** Reports link status.

**Syntax:** linkstatus

## sleep

**cmd:** sleep

**Description:** The suspense process for the Execute command from a file.

**Syntax:** sleep <ms>

## version

**cmd:** version

**Description:** Displays the current software version.

**Syntax:** version

## dev

**cmd:** dev

**Description:** Displays and selects a device.

**Syntax:** dev <device index>

## sromutil

**cmd:** sromutil



**Description:** Provides SROM access.

**Syntax:** sromutil [b<n>] [m <macaddr> <devID> <vedID> <subID>] [c|C|d] [w<location> <value>]

'b'= Enable bootrom, size encoding: 0=Disable, 1=16 KB, 2=32 KB, 3=64 KB, 4=128 KB

'm'= program addr: macaddr subvenID subdevID

'C'= check\_crc

'c'= fix crc

'd'= just dump

'w'= program word; location in hex; word in hex;

'f'= out put image to eeprom.bin and eeprom.txt or filename.bin and filename.txt.

**Example:**

```
sromutil -b 1 -m xxxxxxxxxxxx xxxx xxxx (Enable 16 KB bootrom; Program
addr)
sromutil -m xxxxxxxxxxxx xxxx xxxx (Boot rom status remain unchanged;
Program addr)
sromutil -b 0 (Disable bootrom);
sromutil -b 1 (Enable 16 KB bootrom);
sromutil -C (Check crc)
sromutil -c (Fix crc)
sromutil -d (Dumping data to screen)
sromutil -d -f<filename> (Dumping data to screen, eeprom.bin and
eeprom.txt or filename.bin and filename.txt)
sromutil -w 35 1235 (Program word)
```

## setbit

**cmd:** setbit

**Description:** Sets bit of Generic Memory.

**Syntax:** setbit [!|S|X|#|m|\$||s|x]<addr> <bit#> [<bit#>] ....

! = Configuration space (address range: 0x00 – 0xFF) (32)

S = Configuration space (address range: 0x00 – 0xFF) (16)

X = Configuration space (address range: 0x00 – 0xFF) (08)

\$ = Serial EEPROM

m = MII Registers

l = Direct access (dword)

s = Direct access (word)

x = Direct access (byte)

## clearbit

**cmd:** clearbit

**Description:** Clears bit of Generic Memory.

**Syntax:** clearbit [!|S|X|#|m|\$||s|x]<addr> <bit#> [<bit#>] ....

! = Configuration space (address range: 0x00 – 0xFF) (32)

S = Configuration space (address range: 0x00 – 0xFF) (16)

X = Configuration space (address range: 0x00 – 0xFF) (08)

\$ = Serial EEPROM  
 m = MII Registers  
 l = Direct access (dword)  
 s = Direct access (word)  
 x = Direct access (byte)

## seprg

**cmd:** seprg

**Description:** Reads data from the file and program into seeprom. The file name must be specified in the parameter. The MAC address remains unchanged.

**Syntax:** seprg <f><file name> [[o] [l]] (The file name must be specified in the parameter)

'f' = filename  
 'o' = offset of serial eeprom  
 'l' = length in bytes (Default = size of input file)

### Example:

```
seprg -f=c:\eeprom.bin
```

After programming the SROM with the bin file, power cycle the system to load the new SROM configuration.

## do

**cmd:** do

**Description:** Executes a command from a script file.

**Syntax:** do <filename.do>

### Script file example:

```
reset
linkstatus
mwrite 0 8000
sleep 1000
mread 02
```

---

[Back to Top](#)

---

This document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this document may be reproduced in any form by any means without prior written authorization of Broadcom Corporation. Documentation is provided "as is" without warranty of any kind, either express or implied, including any kind of implied or express warranty of non-infringement or the implied warranties of merchantability or fitness for a particular purpose.

Broadcom Corporation reserves the right to make changes without further notice to any products or data herein to improve reliability, function, or design. Information furnished by Broadcom Corporation is believed to be accurate and reliable. However, Broadcom Corporation does not assume any liability arising out of the application or use of this information, nor the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others.

Broadcom, the pulse logo, and QAMLink are registered trademarks of Broadcom Corporation and/or its subsidiaries in the United States and certain other countries. Microsoft Windows XP and Windows 2000 are registered trademarks of Microsoft Corporation. Intel is a registered trademark of Intel Corporation. All other trademarks are the property of their respective owners.

Broadcom Corporation disclaims any proprietary interest in trademarks and trade names other than its own.

---

## Restrictions and Disclaimers

The information contained in this document, including all instructions, cautions, and regulatory approvals and certifications, is provided by the supplier and has not been independently verified or tested by Dell. Dell cannot be responsible for damage caused as a result of either following or failing to follow these instructions. All statements or claims regarding the properties, capabilities, speeds or qualifications of the part referenced in this document are made by the supplier and not by Dell. Dell specifically disclaims knowledge of the accuracy, completeness or substantiation for any such statements. All questions or comments relating to such statements or claims should be directed to the supplier.

---

*Release: 440X-UM300-D2, October 2, 2002*