

[Back to Contents page](#)

# Broadcom 440X Fast Ethernet Controller Diagnostic 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)

### Example:

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

---

## Prerequisites

The engineering diagnostic is executed under DOS protected mode, which requires dos4gw.exe to be placed in the same directory of the b44diag.exe.

**OS:** DOS 6.22

**Software:** b44diag.exe, dos4gw.exe.

---

## Diagnostic Tests

There are three groups of test, and each group has a few 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](#)

## Test Descriptions

### A1. Indirect Control Register Test

**Command:** regtest -i

**Function:** Each register specified in the configuration contents read only bit and read/write bit defines. The test writing zero and one by using indirect addressing method into the test bits to insure 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 specified in the configuration contents read only bit and read/write bit defines. The test writing zero and one into the test bits to insure 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 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 Serial Prom and verifies integrity by checking CRC.

**Default:** Enabled

### B3. MII Test

**Command:** miitest

**Function:** Each register specified in the configuration contents read only bit and read/write bit defines. The test writing zero and one into the test bits to insure 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 current link status.

**Default:** Enabled

### C1. MAC Loopback Test

**Command:** lbtest 0

**Function:** Transmits a 128-byte packet with incrementing data pattern, and checks tx and rx flags and data integrity.

**Default:** Enabled

### C2. PHY Loopback Test

**Command:** lbtest 1

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

**Default:** Enabled

By default, all tests are covered in manufacturing mode unless disabled.

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.

## **setest**

**cmd:** setest

**Description:** Serial EEPROM read write test. Serial EEPROM tests dumps 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:** mwrtie <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

**nictest**

**cmd:** nictest

**Description:** Runs tests in configuration.

**Syntax:** nictest

**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 zero.

**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>

## 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
```

---

Please read all [restrictions and disclaimers](#).

[Back to Top](#)

[Back to Contents page](#)