



# **Broadcom NetXtreme Ethernet Adapter Diagnostic User's Guide**

**B57diag DOS Version 12.xx**

**B57diag UEFI Version 12.xx**

Copyright © 2000-2009 Broadcom Corporation  
All Rights Reserved

No part of this document may be reproduced, in any form or by any means, without permission in writing from Broadcom Corporation.

Broadcom Corporation reserves the right to make changes to the products or information contained in this document without notice. No liability is assumed as a result of their use or application. No rights under any patent accompany the sale of any such products or information.

Epigram, InsideLine, and iLine10 are trademarks of Broadcom Corporation.

Broadcom Corporation  
5300 California Ave  
Irvine, CA 92617

[www.broadcom.com](http://www.broadcom.com)

## TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION</b> .....	<b>11</b>
<b>2</b>	<b>PREREQUISITES</b> .....	<b>12</b>
<b>3</b>	<b>DIAGNOSTIC TESTS</b> .....	<b>13</b>
3.1	TEST NAMES .....	13
3.2	ERROR CODES .....	14
3.3	TEST DESCRIPTIONS .....	14
3.3.1	A1. Indirect Register Test.....	14
3.3.2	A2. Control Register Test.....	14
3.3.3	A3. Interrupt Test.....	15
3.3.4	A4. BIST.....	15
3.3.5	A5. PCI Cfg Register Test.....	15
3.3.6	A6. Serial Number Reg Test.....	15
3.3.7	A7. Power Register Test.....	15
3.3.8	B1. Scratch Pad Test.....	16
3.3.9	B2. BD SRAM Test.....	16
3.3.10	B3. DMA SRAM Test.....	16
3.3.11	B4. MBUF SRAM Test.....	17
3.3.12	B5. MBUF SRAM via DMA Test.....	17
3.3.13	B6. External SRAM Test.....	17
3.3.14	B7. CPU GPR Test.....	17
3.3.15	C1. EEPROM Test.....	18
3.3.16	C2. CPU Test.....	18
3.3.17	C3. DMA Test.....	18
3.3.18	C4. MII Test.....	18
3.3.19	C5. VPD Test.....	19
3.3.20	C6. ASF Test.....	19
3.3.21	C7. ROM Expansion Test.....	20
3.3.22	C8. CPU Fetch Test.....	20
3.3.23	D1. Mac Loopback Test.....	20
3.3.24	D2. Phy Loopback Test.....	20
3.3.25	D3. RJ45 Loopback Test.....	21
3.3.26	D4. 1G False Carrier Test.....	21
3.3.27	D5. MII Miscellaneous Test.....	21
3.3.28	D6. MSI Test.....	21
3.3.29	D7. E-Switch Test.....	22
3.3.31	D8. SADB Test.....	22
3.3.32	D9. IPsec Loopback Test.....	22
3.3.33	E1. 1G Wire Open/Short (** ZERO LEN LB RJ45 **) Test.....	22
3.3.34	F1. Indirect APE Ctrl Register Test.....	23
3.3.35	F2. APE Control Register Test.....	23
3.3.32	F3. Indirect APE Peripheral Register Test.....	23
3.3.33	F4. APE Peripheral Register Test.....	23
3.3.34	F5. APE Scratch Pad Test.....	23
3.3.35	F6. APE Shared Memory Test.....	24
3.3.36	F7. APE Shared Memory Indirect Access Test.....	24
3.3.37	F8. APE Mutex Register Test.....	24
3.3.38	F9. APE Timers Test.....	25
3.3.39	G1. APE CPU Memory Test.....	25
3.3.40	G2. APE CPU Packet Test.....	25

3.3.41	G3. APE CPU SMBus Loopback Test .....	26
3.3.42	G4. APE CPU GPIO Register Test.....	26
3.3.43	G5. APE CPU Event Register Test.....	26
3.3.44	G6. APE CPU Mutex Register Test.....	26
3.3.45	G7. APE CPU Timers Test .....	26
3.3.46	G8. APE CPU GRC Reset Test.....	27
3.3.47	G9. APE USB Test (UEFI only).....	27
<b>4</b>	<b>COMMAND LINE OPTION PARAMETERS .....</b>	<b>28</b>
<b>5</b>	<b>RECOVER DEVICE FROM CORRUPT NVAM.....</b>	<b>43</b>
<b>6</b>	<b>EEPROM.TXT FORMAT .....</b>	<b>44</b>
<b>7</b>	<b>USER INTERFACE COMMANDS .....</b>	<b>49</b>
<b>8</b>	<b>SPECIAL INSTRUCTION .....</b>	<b>55</b>
<b>9</b>	<b>SAME SYSTEM SEND/RESPOND TEST.....</b>	<b>56</b>
9.1	IDENTIFYING A REFERENCE DEVICE .....	56
9.2	TESTING THE DUT AGAINST THE REFERENCE DEVICE .....	56
9.3	OTHER OPTIONS FOR THE SEND/RESPOND TEST .....	56
<b>10</b>	<b>TEST AND FUNCTIONS DESCRIPTION .....</b>	<b>57</b>
10.1	VPDWRITE.....	57
10.2	VPDREAD .....	57
10.3	VPDINFO .....	57
10.4	SEMODE.....	57
10.5	SERead.....	58
10.6	SEWRITE .....	58
10.7	SECFG .....	59
10.8	SEPRG .....	77
10.9	UPGFRM .....	79
10.10	SEVER.....	79
10.11	SECHKSUM .....	80
10.12	SEDUMP.....	80
10.13	ASF CFG.....	80
10.14	MANCFG .....	82
10.15	ASFMBOX .....	83
10.16	FLSHMODE .....	83
10.17	FLSHREAD .....	83
10.18	CHKPXE.....	83
10.19	DIR .....	84
10.20	ERASE.....	84
10.21	PXEPRG .....	84
10.22	FLSHDEV .....	84
10.23	BITBANG .....	84
10.24	SELOCK .....	85
10.25	SETORTURE .....	85
10.26	SEINIT.....	85
10.27	SEARB .....	85
10.28	SEPROTECT.....	86
10.29	SELCLOCK .....	86
10.30	SEMAP .....	86
10.31	SETWOL.....	86

10.32	SETPXE.....	86
10.33	SETASF.....	87
10.34	SECOMP.....	87
10.35	DRESET.....	87
10.36	USERBLOCK.....	88
10.37	SETMBA.....	88
10.38	SEGENCRC.....	88
10.39	SECFG1.....	88
10.40	SECFG2.....	89
10.41	SECFG3.....	89
10.42	SECFG4.....	90
10.43	SECFG5.....	91
10.44	SETIPMI.....	91
10.45	LOADFW.....	92
10.46	CPUDTT.....	92
10.47	CPUDRT.....	93
10.48	CPUTRACE.....	93
10.49	HALTCPU.....	93
10.50	LOADBOOTCODE.....	94
10.51	DISASM.....	94
10.52	STEP.....	94
10.53	GO.....	94
10.54	SHOWGPR.....	95
10.55	PC.....	95
10.56	BREAKPOINT.....	95
10.57	SELECT.....	95
10.58	U.....	95
10.59	TRAP.....	95
10.60	CPUINFO.....	96
10.61	CPUGPRSTATUS.....	96
10.62	CPUGPRRESTART.....	96
10.63	CPUFETCH.....	96
10.64	ROMCODETEST.....	96
10.65	APEINFO.....	97
10.66	APECTL.....	97
10.67	APEPING.....	98
10.68	APERREAD.....	98
10.69	APEOTPKEY.....	98
10.70	APELOG.....	98
10.71	DMAW.....	99
10.72	DMAR.....	100
10.73	DMA_H.....	102
10.74	DMA_D.....	102
10.75	DMA_ALLOC.....	102
10.76	MACLPK.....	102
10.77	BLAST.....	103
10.78	NICSTATS.....	105
10.79	RINGINDEX.....	107
10.80	PHYCTRL.....	108
10.81	TXPKT.....	108
10.82	STATUSBLK.....	110
10.83	STSBLK.....	110
10.84	TXCFG.....	111
10.85	RXCFG.....	112
10.86	TPROT.....	112

10.87	QSTAT .....	112
10.88	DRVRCFG.....	113
10.89	IRT .....	113
10.90	MACMRD .....	113
10.91	MIIMRD .....	114
10.92	PCIMRD .....	114
10.93	INBAND .....	114
10.94	MWRITE.....	115
10.95	MREAD .....	115
10.96	MDEV .....	116
10.97	M2DEV .....	116
10.98	MIIMODE .....	116
10.99	LBERTRAM .....	116
10.100	DBERTRAM.....	117
10.101	BERTSTATS.....	117
10.102	RM.....	117
10.103	MRLOOP .....	117
10.104	PHYMSE.....	118
10.105	INITPHY .....	118
10.106	TAPDUMP .....	118
10.107	MEMSEARCH .....	118
10.108	READ .....	118
10.109	WRITE.....	119
10.110	POLL.....	120
10.111	SETBIT .....	121
10.112	CLEARBIT .....	122
10.113	READBIT .....	122
10.114	CREAD .....	123
10.115	CWRITE .....	123
10.116	PCIRD .....	123
10.117	PCIWR.....	124
10.118	VPDTEST.....	124
10.119	REGTEST.....	124
10.120	MITEST .....	125
10.121	MSI.....	125
10.122	MEMTEST .....	125
10.123	SETEST .....	126
10.124	BIST.....	126
10.125	NICTEST.....	126
10.126	INTRTEST.....	127
10.127	PKTTEST .....	127
10.128	CPUTEST.....	128
10.129	DMATEST.....	128
10.130	TESTE .....	128
10.131	TESTD.....	129
10.132	ASFTEST .....	129
10.133	BUSTEST.....	129
10.134	SRAMTEST .....	131
10.135	MSITEST .....	131
10.136	ROMTEST.....	131
10.137	GPIOTEST.....	132
10.138	CPUDIAG .....	132
10.139	PCICFGTEST .....	132
10.140	PETEST .....	133
10.141	ERRCTRL .....	133

10.142	SEDVT .....	133
10.143	MIIMISCTEST .....	134
10.144	CPUGPRTEST.....	134
10.145	DMASHASTA.....	134
10.146	BINCHKSUM.....	135
10.147	PMDCFG.....	135
10.148	PMPD .....	135
10.149	INTR .....	136
10.150	INTRCTRL .....	136
10.151	INTT .....	136
10.152	MBUF.....	136
10.153	LOADDRV .....	137
10.154	UNLOADDRV .....	138
10.155	MACHALT .....	138
10.156	FTQ.....	138
10.157	ADDMC.....	139
10.158	DELMC .....	139
10.159	TXMACDES .....	139
10.160	TXMACSRC .....	139
10.161	CHKLDDRV .....	140
10.162	VLANTAG .....	140
10.163	REG .....	140
10.164	EXIT .....	140
10.165	DEBUG.....	140
10.166	GPIOWRITE .....	141
10.167	GPIOREAD.....	141
10.168	PXECPY .....	142
10.169	DEVICE.....	142
10.170	VERSION.....	142
10.171	HELP.....	142
10.172	?.....	143
10.173	RADIX.....	143
10.174	NOLOG .....	143
10.175	LOG .....	143
10.176	PCIINIT .....	143
10.177	PCISCAN .....	144
10.178	DOS .....	144
10.179	SHELL .....	144
10.180	DIAGCFG .....	144
10.181	RESET.....	147
10.182	QUIT.....	147
10.183	SMBUS.....	147
10.184	CLS.....	147
10.185	LOOP .....	147
10.186	DBMODE.....	148
10.187	ASF.....	148
10.188	NEW .....	148
10.189	ASFPRG.....	148
10.190	SLEEP .....	149
10.191	FILLPATTERN.....	149
10.192	INP .....	149
10.193	OUTP .....	149
10.194	SWITCH .....	150
10.195	DO.....	150
10.196	TXFILL.....	150

10.197	WBUF .....	151
10.198	RBUF .....	151
10.199	CPBUF .....	151
10.200	ECHO .....	152
10.201	PAUSE.....	152
10.202	Q .....	152
10.203	VERBOSE .....	152
10.204	BEEP.....	153
10.205	VAR.....	153
10.206	MEMINFO.....	153
10.207	DELVAR.....	153
10.208	REGDUMP .....	153
10.209	REGCOMP .....	154
10.210	REGRESTORE .....	154
10.211	NVSIZE .....	154
10.212	ASPM.....	155
10.213	DIDS .....	155
10.214	SERIAL .....	156
10.215	POWER .....	156
10.216	READBR.....	156
10.217	WRITEBR .....	156
10.218	FINDBRIDGE .....	156
10.219	BRIDGE.....	156
10.220	PERE.....	157
10.221	PERD .....	157
10.222	PECLR.....	157
10.223	PECHK .....	157
10.224	ISCSIPRG.....	157
10.225	UMPECHO .....	158
10.226	UMPCFG.....	158
10.227	SETIPMI .....	159
10.228	SETUMP .....	159
10.229	SEOTP .....	159
10.230	OTPCBK.....	160
10.231	DEFRAGMENT .....	160
10.232	SECFGSB1 .....	160
10.233	SECFGSB2.....	161
10.234	SECFGSB3.....	162
10.235	SECFGHWSB1.....	162
10.236	SECFGHWSB2.....	163
10.237	SECFGHWSB3.....	163
10.238	ESWITCH .....	164
10.239	SBFCFG.....	164
10.240	LINKINTRTEST .....	164
10.241	IPV6EXTLD .....	164
10.242	IPV6EXTEN .....	165
10.243	CFGPAGE SIZE .....	165
10.244	CPMU.....	165
10.245	MDIO .....	165
10.246	WAIT .....	166
10.247	IPSECSUPPORT .....	166
10.248	SADBTEST.....	166
10.249	SADBUTIL .....	166
10.250	APETEST .....	167



11 ERROR MESSAGES ..... 168



## 1 Introduction

This program runs in two modes: Manufacturing mode and Engineering mode. The mode is determined with the command line option or the configuration file. When the program is running in manufacturing mode, it starts to run all tests in the configuration. If it detects an error, it displays an error and exits the program. When the program is in engineering mode, it prompts user to enter commands. The commands are explained in the later chapters. This document provides the information on configuration file specification, command line options and engineering diagnostic commands on Broadcom NetXtreme Ethernet adapter, in particular to check out the functionality of the BCM5700 Family of Ethernet controllers and its related components. In general, this program has a set of default configuration. It is overwritten by configuration file. The command line option overwrites both default and the configuration files.

## 2 Prerequisites

The engineering diagnostic is executed under DOS protected mode or under Unified Extensible Firmware Interface (UEFI).

**OS:** Dos 6.22 or UEFI v2.0x or later

**Software:** b57diag.exe for DOS and b57diag64.efi, b57diag32.efi for UEFI (version 11.05 or later).

**Input File List:** The following files should be found in the same location of the b57diag.

ee57xxyy.yy (TX & RX CPUs Firmware file, xx chip type, yy version number)

eeprom.bin (Serial EEPROM/FLASH config input file)

cpu.bin, cpu05.bin, cpu14a.bin, cpu14b.bin or cpusj.bin (CPU Instruction test)

cpudiag.bin or cpudg05.bin (CPU Accessing test)

flshdiag.bin, flashdg05.bin flashdg 14a.bin, flashdg 14b.bin, flashdgsj.bin, flahdg5x.bin

ump14a.bin or ump14b.bin

config.sys (not required for UEFI)

himem.sys (not required for UEFI)

### Output File List:

The following file may be generated in run time depending execution option(s).

diagcfg.bin

### 3 Diagnostic Tests

The tests are divided into four groups: Register Tests, Memory Tests, Miscellaneous Tests, and Data Tests. They numbered as group 'A', 'B', 'C', 'D', 'E', 'F', and 'G'.

Note: For UEFI version, not all tests are supported. Refer to online help for more information.

#### 3.1 Test Names

##### Group A.

- A1. Indirect Register Test
- A2. Control Register Test
- A3. Interrupt Test
- A4. BIST
- A5. PCI Cfg Register Test
- A6. Serial Number Reg Test
- A5. Power Register Test

##### Group B.

- B1. Scratch Pad Test
- B2. BD SRAM Test
- B3. DMA SRAM Test
- B4. MBUF SRAM Test
- B5. MBUF SRAM via DMA Test
- B6. External SRAM Test
- B7. CPU GPR SRAM Test

##### Group C.

- C1. EEPROM Test
- C2. CPU Test
- C3. DMA Test
- C4. MII Test
- C5. VPD Test
- C6. ASF Hardware Test
- C7. ROM Expansion Test
- C8. CPU Fetch Test

##### Group D.

- D1. Mac Loopback Test
- D2. Phy Loopback Test
- D3. RJ45 Loopback Test
- D4. 1G False Carrier Test
- D5. MII Miscellaneous Test
- D6. MSI Test
- D7. E-Switch Test (5756 and 5761E devices only)
- D8. SADB Test (available on NetXtreme Controllers with hardware IPsec support)
- D9. IPsec Loopback Test (available on NetXtreme Controllers with hardware IPsec support)

### Group E.

E1. 1G Wire Open/Short (\*\* ZERO LEN LB RJ45 \*\*) Test

### Group F. (valid on APE enabled NetXtreme Controllers only)

- F1. Indirect APE Ctrl Register Test
- F2. APE Control Register Test
- F3. Indirect APE Peripheral Register Test
- F4. APE Peripheral Register Test
- F5. APE Scratch Pad Test
- F6. APE Shared Memory Test
- F7. APE Shared Memory Indirect Access Test
- F8. APE Mutex Register Test
- F9. APE Timers Test

### Group G. (valid on APE enabled NetXtreme Controllers only)

- G1. APE CPU Memory Test
- G2. APE CPU Packet Test
- G3. APE CPU SMBus Loopback Test
- G4. APE CPU GPIO Register Test
- G5. APE CPU Event Register Test
- G6. APE CPU Mutex Register Test
- G7. APE CPU Timers Test
- G8. APE CPU GRC Reset Test
- G9. APE USB Test (UEFI only)

## 3.2 Error Codes

The Error Codes are presented in Section 9 - ERROR MESSAGES.

## 3.3 Test Descriptions

### 3.3.1 A1. Indirect Register Test

**Command:** regtest -i

**Function:** Using indirect addressing method, writing increment data into MAC hash Register table and read back for verification. The memory read/write is done 100 times while increment test data.

**Default:** Enabled

### 3.3.2 A2. Control Register Test

**Command:** regtest

**Function:** Each Register specified in the configuration contents is tested for read only bit and read/write bit defines. The test writes zeroes and ones into the test bits to insure the read only bits are not changed, and read/write bits are changed accordingly.

**Default:** Enabled.

### 3.3.3 A3. Interrupt Test

**Command:** intrtest

**Function:** This test verifies the interrupt functionality. It enables interrupt and waits for interrupt to occur. It waits for 500ms and reports error if could not generate interrupts.

**Default:** Enabled

### 3.3.4 A4. BIST

**Command:** bist

**Function:** Hardware Built-In-Self-Test (BIST). This test initiates BIST, and wait for the test result returned by hardware.

**Default:** Enabled

### 3.3.5 A5. PCI Cfg Register Test

**Command:** pcicfg

**Function:** This test verifies the access integrity of the PCI config registers.

**Default:** Enabled

### 3.3.6 A6. Serial Number Reg Test

**Command:** serial

**Function:** Tests the PCI-E Serial Number capabilities registers on chips that support the Serial Number capability. This command is supported on 5751Cx, 5755, 5787.

**Default:** Enabled.

### 3.3.7 A7. Power Register Test

**Command:** power

**Function:** Tests the PCI-E Power capabilities registers on chips that support the Power capability. This command is supported on 5751Cx, 5755, 5787.

**Default:** Enabled.

### 3.3.8 B1. Scratch Pad Test

**Command:** memtest -s

**Function:** This test tests the scratch pad SRAM on board. The following tests are performed:

**Data Pattern Test:** Write test data into SRAM, read back to ensure data is correct. The test data used is 0x00000000, 0xffffffff, 0xaa55aa55, and 0x55aa55aa.

**Alternate Data Pattern Test:** Write test data into SRAM. Write complement test data into next address. Read back both data to insure the data is correct. After the test, the program reads back data one more time to insure the data stays correct. The test data used is 0x00000000, 0xffffffff, 0xaa55aa55, and 0x55aa55aa.

**Address Test:** Write each address with unique increment data. Read back data to insure data is correct. After fill the entire data with the unique data, the program reads back data again to insure data stays the same.

**Walking Bit Test:** For each address location, starting at bit 0, each bit is set, tested and then shifted left by one. This process is repeated for each of the 32 bits in each address location in the entire memory test range.

**Pseudo Random Data Test:** A pre-calculated pseudo random data is used to write a unique data into each test RAM. After the first pass the test, the program reads back one more time to insure data stays correct.

**Default:** Enabled

### 3.3.9 B2. BD SRAM Test

**Command:** memtest -b

**Function:** This test tests the BD SRAM. This performs exact the same way of testing as described in B1. Scratch Pad Test.

**Default:** Enabled

### 3.3.10 B3. DMA SRAM Test

**Command:** memtest -d

**Function:** It tests DMA SRAM by performing the tests described in test B1. The Scratch Pad Test.

**Default:** Enabled



### 3.3.11 B4. MBUF SRAM Test

**Command:** memtest -m

**Function:** It tests DMA SRAM by performing the tests described in test B1. The Scratch Pad Test.

**Default:** Enabled

### 3.3.12 B5. MBUF SRAM via DMA Test

**Command:** memtest -x

**Function:** Eight test pattern data are used in the test. They are described below. A 0x1000 sized data buffer is used for this test. Before each pattern test, the buffer is initialized and filled with the test pattern. It then, performs size 0x1000 transmit DMA from host buffer to NIC MBUF memory. Verify the data integrity in MBUF against host memory and repeat the DMA for the entire MBUF buffer. Then it performs receive DMA from NIC to host. The 0x1000-byte test buffer is cleared to zero before each receive-DMA. Verify the data integrity and test is repeated for the entire MBUF SRAM range.

Test Pattern	Description
“16 00's 16 FF's”	Full the entire host DMA buffer with 16 bytes of 00's and then 16 bytes of FF's.
“16 FF's 16 0's”	Full the entire host DMA buffer with 16 bytes of 00's and then 16 bytes of FF's.
“32 00's 32 FF's”	Full the entire host DMA buffer with 32 bytes of 00's and then 32 bytes of FF's.
“32 FF's 32 00's”	Full the entire host DMA buffer with 32 bytes of FF's and then 32 bytes of 00's.
“00000000's”	Full the entire host DMA buffer with all zeros.
“FFFFFFFF's”	Full the entire host DMA buffer with all FF's.
“AA55AA55's”	Full the entire host DMA buffer with data 0xAA55AA55.
“55AA55AA's”	Full the entire host DMA buffer with data 0xAA55AA55.

**Default:** Enabled

### 3.3.13 B6. External SRAM Test

**Command:** memtest -e

**Function:** It tests DMA SRAM by performing the tests described in test B1. The Scratch Pad Test.

**Default:** Disabled

### 3.3.14 B7. CPU GPR Test

**Command:** memtest -b

**Function:** This test tests the CPU General Purpose Registers. This performs exact the same way of testing as described in B1 over 3 voltages (1.1V, 1.2V, 1.3V).

**Default:** Enable

### 3.3.15 C1. EEPROM Test

**Command:** setest

**Function:** An increment test data is used in EEPROM test. It fills the test data into the test range and read back to verify the content. After the test, it fills data with zero to clear the memory.

**Default:** Enabled

### 3.3.16 C2. CPU Test

**Command:** cputest

**Function:** This test opens the file cpu.bin. If file exists and content is good, it loads code to rx and tx CPU and verifies CPU execution.

**Default:** Enabled

### 3.3.17 C3. DMA Test

**Command:** dmatest

**Function:** Both high and low priorities DMA are tested. It moves data from host memory to NIC SRAM, verifies data, and then moves data back to host memory again to verify data.

**Default:** Enabled

### 3.3.18 C4. MII Test

**Command:** miitest

**Function:** The function is identical to A2. Control Register Test. 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.

#### **Default Register table**

The test will try to read the register configuration file 'miireg.txt' for the register defines. If the file does not exists, the following table is used:

Offset	R/O Mask	R/W Mask
0x00	0x0000	0x7180
0x02	0xffff	0x0000
0x03	0xffff	0x0000
0x04	0x0000	0xffff
0x05	0xffff	0x0000
0x06	0x0001	0x0000
0x07	0x0800	0xb7ff
0x08	0xffff	0x0000
0x09	0x0000	0xff00
0x0a	0x7c00	0x0000
0x10	0x0000	0xffbf
0x11	0x3300	0x0000
0x19	0x001f	0x0000
0x1e	0x0000	0xffff
0x1f	0x0000	0xffff

### 3.3.19 C5. VPD Test

**Command:** vpdtest

**Function:** It saves the content of VPD first before perform the test. Once it is done, it writes one of the five pattern test data, 0xff, 0xaa, 0x55, increment data, or decrement data, into VPD memory. By default, increment data pattern is used. It writes and reads back the data for the entire test range, and then restores the original content.

**Default:** Enabled

### 3.3.20 C6. ASF Test

**Command:** asftest

**Function:**m

**1. Reset test.**

Setting reset bit, poll for self-clearing. Verify reset value of registers.

**2. Event Mapping Test**

Setting SMB\_ATTEN bit. By changing ASF\_ATTEN LOC bits, verify the mapping bits in TX\_CPU or RX\_CPU event bits.

**3. Counter Test**

Clear WG\_TO, HB\_TO, PA\_TO, PL\_TO, RT\_TO bits by setting those bits. Make sure the bits clear.

Clear Timestamp Counter. Writing a value 1 into each PL, PA, HB, WG, RT counters. Set TSC\_EN bit.

Poll each PA\_TO bit and count up to 50 times. Check if PL\_TO gets set at the end of 50 times. Continue to count up to 200 times. Check if all other TO bits are set and verify Timestamp Counter is incremented.

**Default:** Enabled

### 3.3.21 C7. ROM Expansion Test

**Command:** romtest

**Function:** This function tests the ability to enable/disable/access the expansion rom on the device.

**Default:** Enabled

### 3.3.22 C8. CPU Fetch Test

**Command:** cpufetch

**Function:** Test the CPU instruction-fetch logic a 100 times on 5705 and later devices. The voltage is also varied to 1.1V and 1.3V on devices that support voltage variation.

**Default:** Enabled

### 3.3.23 D1. Mac Loopback Test

**Command:** pkttest -m

**Function:** This is internal loopback data transmit/receive test. It initializes MAC into internal loopback mode, and transmits 200 packets. The data should be routed back to receive channel and receive by the receive routine, which verifies the integrity of data. One Giga bit rate is used for this test.

**Default:** Enabled

### 3.3.24 D2. Phy Loopback Test

**Command:** pkttest -p

**Function:** This test is same as D1. Mac Loopback Test except, the data is routed back via physical layer device. One Giga bit rate is used for this test.

Please note if the device supports E-Switch, such as 5756, the test is performed on both laptop mode and docking mode. To perform the test only on the default port, two method can be used: 1. command line option '-disableeswitch'; 2. 'eswitch -f' in engineering mode before the test.

**Default:** Enabled

### 3.3.25 D3. RJ45 Loopback Test

**Command:** pkttest -e

**Function:** This is external loopback test. From the UUT point of view, no loopback mode is configured. The data expected to be routed back by RJ45 loopback connector. 10M/s, 100M/s, and 1000M/s are used for this test.

Please note if the device supports E-Switch, such as 5756, the test is performed on both laptop mode and docking mode. To perform the test only on the default port, two method can be used: 1. command line option '-disableeswitch'; 2. 'eswitch -f' in engineering mode before the test.

**Default:** Disabled

### 3.3.26 D4. 1G False Carrier Test

**Command:** None

**Function:** This test executes the same procedure as test D3 and tests to see if a *false carrier* was detected at the end of the test. The -l200, -dispgbpkt, -disppkt and -errlimit command line options were added to configure a number of operational parameters. The details of the command line options are provided in the "COMMAND LINE OPTION PARAMETERS" section.

Please note if the device supports E-Switch, such as 5756, the test is performed on both laptop mode and docking mode. To perform the test only on the default port, two method can be used: 1. command line option '-disableeswitch'; 2. 'eswitch -f' in engineering mode before the test.

**Default:** Disabled

### 3.3.27 D5. MII Miscellaneous Test

**Command:** None

**Function:** This function tests the auto-polling and phy-interrupt capabilities. These are the functionalities of the phy.

**Default:** Enabled

### 3.3.28 D6. MSI Test

**Command:** msitest

**Function:** Testing Message Signaled Interrupt Function to see if it handles this interrupt correctly.

**Default:** Enabled

### 3.3.29 D7. E-Switch Test

**Command:** nictest d7

**Function:** This function tests the E-Switch Ethernet porting switching hardware.

**Default:** Enabled (available only for NetXtreme controllers with E-Switch support)

### 3.3.31 D8. SADB Test

**Command:** nictest d8

**Function:** This function tests the Security Association Data Base hardware in IPsec supported NetXtreme Ethernet controllers.

**Default:** Enabled (available only on NetXtreme controllers with IPsec support)

### 3.3.32 D9. IPsec Loopback Test

**Command:** nictest e1

**Function:** An external loopback test is performed at 10/100/1000 Mb/s using IPsec encrypted packets. Various hardware supported encryption ciphers are used to encrypt and decrypt packets and verify the proper operation of the IPsec hardware.

A zero length loopback connector is placed at the RJ45 connector of the LOM/NIC. After running the loopback test at 1000 Mb/s the line signal quality is tested order to catch line faults (shorts or opens). This test is used during manufacturing in order to detect board/system build issues on the line/analog side of the Ethernet controller.

**Default:** Disabled (available only on NetXtreme controllers with IPsec support)

### 3.3.33 E1. 1G Wire Open/Short (\*\* ZERO LEN LB RJ45 \*\*) Test

**Command:** nictest e1

**Function:** An external loopback test is performed at 10/100/1000 Mb/s. A zero length loopback connector is placed at the RJ45 connector of the LOM/NIC. After running the loopback test at 1000 Mb/s the line signal quality is tested order to catch line faults (shorts or opens). This test is used during manufacturing in order to detect board/system build issues on the line/analog side of the Ethernet controller.

**Default:** Disabled

**NOTE:** The following tests in test groups F and G are only enabled for NetXtreme Ethernet Controllers which support the Application Processing Engine (APE).

### 3.3.34 F1. Indirect APE Ctrl Register Test

**Command:** nictest f1

**Function:** Using indirect addressing method, incremented data is written into APE control registers and read back for verification. The memory read/write is done 100 times while incrementing test data.

**Default:** Disabled

### 3.3.35 F2. APE Control Register Test

**Command:** nictest f2

**Function:** Each Register specified in the APE control register block is tested for read only bits and read/write bits. The test writes zeroes and ones into the test bits to insure the read only bits are not changed, and read/write bits are changed accordingly.

**Default:** Disabled

### 3.3.32 F3. Indirect APE Peripheral Register Test

**Command:** nictest f3

**Function:** Using indirect addressing method, incremented data is written into APE Peripheral registers and read back for verification. The memory read/write is done 100 times while incrementing test data.

**Default:** Enabled

### 3.3.33 F4. APE Peripheral Register Test

**Command:** nictest f4

**Function:** Each Register specified in the APE Peripheral register block is tested for read only bits and read/write bits. The test writes zeroes and ones into the test bits to insure the read only bits are not changed, and read/write bits are changed accordingly.

**Default:** Enabled

### 3.3.34 F5. APE Scratch Pad Test

**Command:** nictest f5

**Function:** This test tests the APE scratch pad SRAM on the APE enabled NetXtreme controllers. The following tests are performed:

**Data Pattern Test:** Write test data into SRAM, read back to ensure data is correct. The test data used is 0x00000000, 0xffffffff, 0xaa55aa55, and 0x55aa55aa.

**Alternate Data Pattern Test:** Write test data into SRAM. Write complement test data into next address. Read back both data to insure the data is correct. After the test, the program reads back data one more time to insure the data stays correct. The test data used is 0x00000000, 0xffffffff, 0xaa55aa55, and 0x55aa55aa.

**Address Test:** Write each address with unique increment data. Read back data to insure data is correct. After filling the entire memory length with the unique data, the program reads back the data again to insure data stays the same.

**Walking Bit Test:** For each address location, starting at bit 0, each bit is set, tested and then shifted left by one. This process is repeated for each of the 32 bits in each address location in the entire memory test range.

**Pseudo Random Data Test:** A pre-calculated pseudo random data is used to write a unique data into each test RAM. After the first pass the test, the program reads back one more time to insure data stays correct.

**Default:** Enabled

### 3.3.35 F6. APE Shared Memory Test

**Command:** nictest f6

**Function:** Performs memory data verification tests on the APE Shared Memory. This test uses the same the method of testing as described in F5. APE Scratch Pad Test.

**Default:** Enabled

### 3.3.36 F7. APE Shared Memory Indirect Access Test

**Command:** nictest f7

**Function:** Performs memory data verification tests using Indirect Addressing Method, on the APE Shared Memory. This test uses the same the method of testing as described in F5. APE Scratch Pad Test.

**Default:** Enabled

### 3.3.37 F8. APE Mutex Register Test

**Command:** nictest f8



**Function:** Each of the registers in the APE Mutex Register block is tested to verify that each bit set in the Mutex Request register can acquire a corresponding grant bit in its paired Mutex Grant register. After each grant bit is set it is written with back to verify the grant bit can be cleared.

**Default:** Disabled

### 3.3.38 F9. APE Timers Test

**Command:** nictest f9

**Function:** This test each of the APE internal timer registers from the host interface. The test verifies that the APE timers are functioning and within acceptable tolerances.

**Default:** Disabled

### 3.3.39 G1. APE CPU Memory Test

**Command:** nictest g1

**Function:** This test executes a series of memory tests using the APE CPU.

**Data Pattern Test:** Write test data into SRAM, read back to ensure data is correct. The test data used is 0x00000000, 0xffffffff, 0xaa55aa55, and 0x55aa55aa.

**Alternate Data Pattern Test:** Write test data into SRAM. Write complement test data into next address. Read back both data to insure the data is correct. After the test, the program reads back data one more time to insure the data stays correct. The test data used is 0x00000000, 0xffffffff, 0xaa55aa55, and 0x55aa55aa.

**Address Test:** Write each address with unique increment data. Read back data to insure data is correct. After fill the entire data with the unique data, the program reads back data again to insure data stays the same.

**WalkingOne bit Test:** For each address. Data one is written and read back for testing. Then shift the data left one bit, so the data becomes two and do the same test again. It repeats for 32 times until the test bit is shifted out of test data. The same is test is repeated for entire test range.

**Pseudo Random Data Test:** A pre-calculated pseudo random data is used to write a unique data into each test RAM. After the first pass the test, the program reads back one more time to insure data stays correct.

**Default:** Enabled

### 3.3.40 G2. APE CPU Packet Test

**Command:** nictest g2

**Function:** An Ethernet packet is transmitted via internal loopback from the APE CPU and verified for data integrity and that packet interrupts are generated.

**Default:** Enabled

#### 3.3.41 G3. APE CPU SMBus Loopback Test

**Command:** nicetest g3

**Function:** This test sends and receives SMBus messages from the APE CPU via loopback on the SMBus controllers and verifies that correct data was received. (Requires loopback jumpers to be installed).

**Default:** Disabled

#### 3.3.42 G4. APE CPU GPIO Register Test

**Command:** nicetest g4

**Function:** This test writes zeroes and ones to the GPIO output pins via the APE CPU to insure the bits are changed accordingly and that GPIO state change interrupts are generated.

**Default:** Disabled

#### 3.3.43 G5. APE CPU Event Register Test

**Command:** nicetest g5

**Function:** In this test the APE CPU writes to the APE event registers to verify that each event state can generate a corresponding interrupt.

**Default:** Enabled

#### 3.3.44 G6. APE CPU Mutex Register Test

**Command:** nicetest g6

**Function:** Each of the registers in the APE Mutex Register block is tested from the APE CPU. This test verifies that each bit set in the Mutex Request register can acquire a corresponding grant bit in its paired Mutex Grant register. After each grant bit is set it is written with back to verify the grant bit can be cleared.

**Default:** Enabled

#### 3.3.45 G7. APE CPU Timers Test

**Command:** nicetest g7

**Function:** This test each of the APE internal timer registers using the APE CPU. The test verifies that the timers are functioning and within acceptable tolerances.

**Default:** Enabled

### 3.3.46 G8. APE CPU GRC Reset Test

**Command:** nictest g8

**Function:** This test verifies that the GRC port interrupt function can be detected via the APE CPU.

**Default:** Enabled

### 3.3.47 G9. APE USB Test (UEFI only)

**Command:** nictest g9

**Function:** This test verifies that the APE USB port can be detected from the host controller by executing a series USB descriptor query commands. This test is only available on the UEFI version of B57diag.

**Default:** Disabled

## 4 Command line option parameters

When users invoke this program, a set of option parameter can be used to overwrite the configuration file or the default configuration. This section summarizes the options. The options are case sensitive.

**-putil            Call “b57putil.exe” utility to unload PXE driver.**

When "-putil" was entered, diag will call another application "b57putil.exe". "b57putil.exe" will unload PXE and such that diag can have full control over the device. This option switch should be used when updating firmware with "-firmall" option switch via PXE connection.

**-seldev <VID\_DID>    option to select devices that match VID and DID**

Use this option to select target devices. Other devices, which have different VIDs and DIDs, will not be selected and listed. This option switch can be used along with “-firm” and “-firmall” option switches.

Example: a:\b57diag -b57eng -seldev 14e411677

**-uumpp <filename>    used for field program of UMP firmware**

The feature is used to execute a field upgrade of UMP firmware. The firmware is programmed into a/the device/s specified by “-c” option switch if UMP firmware is originally loaded in NVRAM

**-dir    used for displaying file directory in NVRAM**

The feature is used to display file directory in NVRAM. The file directory in the NVRAM of device/s specified by "-c" option switch will be display. If no "-c" option switch has been entered, the file directory of all detected devices will be displayed. In order to log the file directory to a log file, "-l" option switch option must be entered BEFORE "-dir" option switch. Since this is a single function command, the will be no "nictest" or other functions will be preformed.

**-pump <file>        Program UMP firmware**

This option needs to follow the -e and -c options. If this option is entered, the program will retrieve the UMP firmware filename from the command line then it will start programming.

Example:

b57diag -e <code> -c 0 -pump ee5714c1.00

**-pump1 <file>        Program UMP firmware only**

The feature is used to execute a field upgrade of NVRAM to add UMP firmware. The firmware is programmed into a/the device/s specified by “-c” option switch.

**-u <value>        : Enable/Disable (value = 1/0) UMP in manufacture mode**

**-piscsi <file>      Program ISCSI firmware**

This option needs to follow the `-e` and `-c` options. If this option is entered, the program will retrieve the ISCSI firmware filename from the command line then it will start programming.

Example:

```
b57diag -e <code> -c 0 -piscsi iscsi.bin
```

**-piscsicfg      Force program ISCSI CFG firmware**

If this option is entered, it will force program the ISCSI CFG firmware to NVRAM. This option must be used along with `-piscsi` option.

Example:

```
b57diag -e <code> -c 0 -piscsi iscsi.bin -piscsicfg
```

**-piscsiprg      Force program ISCSI PRG firmware**

If this option is entered, it will force program the ISCSI PRG firmware to NVRAM. This option must be used along with `-piscsi` option.

Example:

```
b57diag -e <code> -c 0 -piscsi iscsi.bin -piscsiprg
```

**-piscsi1 <filename>      used for field program of ISCSI firmware**

The feature is used to execute a field upgrade of NVRAM to add ISCSI firmware. The firmware is programmed into a/the device/s specified by “-c” option switch.

**-smbaddr <hex1> | <hex2> | ..... | <hexn> used for programming SMBus Address for ASF/IPMI firmware.**

The feature will allow user to program SMBus Address for ASF/IPMI firmware. This switch can take multiple parameters in HEX.

Example

1. `-smbaddr A4` (SMB Address = 0xA4)
2. `-smbaddr A4 A6` (SMB Address for 1st device=0xA4 and 2nd device = 0xA6)

This switch must be used along with `-c` option to indicate the target device.

Example:

```
b57diag -c 0 3 6 -smbaddr A4 A6 A8 -t abcd.  
b57diag -c 0 -e b57kia -pasf asf.bin -smbaddr A4 -t abcd
```

**-chksecfg <file\_p> | <file\_s> will enable the NVRAM SecfgTest**

Boot code configuration will be checked against input files. The input files are in the same format as EEPROM.TXT. `<file_p>` is for the primary port and `<file_s>` is for the

secondary port. Software will read command from input files and compare with the boot code configuration and return “Passed” or “Failed” accordingly.

Example:

b57diag -chksecfg file\_p.txt (for single port devices)

b57diag -chksecfg file\_p.txt file\_s.txt (for dual port devices)

**-nosz      disable storing NVRAM and TPM size in NVRAM**

Disable the storing of the TPM and NVRAM size in NVRAM when diagnostics is run from the command prompt. Use the "-nosz" option when repeatedly power cycling a system in order not to exhaust the total NVRAM write cycles.

**-smbaddr <hex1> | <hex2>...|<hex3> Configure ASF SMBus Addresses.**

The option will configure the ASF SMBus Address field in ASF Configuration block.

The option can take multiple parameters in HEX.

Example:

a. -smbaddr 45 (SMB Address = 0x45)

b. -smbaddr 45 6c (SMB Address for 1st device=0x45 and 2nd device = 0x6c)

This option must be used along with -c option. Number of parameters for -smbaddr must be the same as number of devices selected by -c.

Example:

b57diag -c 0 3 6 -smbaddr 6c-6e -t abcd.

b57diag -c 0 -e b57kia -pasf asf.bin -smbaddr 66 -t abcd

**-mfct <filename>    updates the PCI SSID and SVID**

Use this option to modify the PCI SSID and SVID stored in NVRAM. The new SSID and SVID are specified in the file <filename>. The format of <filename> is specified in the “EEPROM.TXT format” section below.

**-sil      suppresses warning messages of the “-firm” and “-firmall” commands**

The warning message, “Boot code file and device type incompatible produced by the “-firm” and “-firmall” commands is not displayed when the “-sil” command line parameter is entered before the “-firm” and “-firmall” commands. The “-firm” and “-firmall” commands are described below.

**-rf <x>              selects a reference device to use during external loopback**

Use this option with the external loopback test D3 to select a reference card. During the external loopback test the physical loopback can be provided with a RJ45 loopback plug or a CAT5 cable connected to a reference device (which loops back the data).

**-lbe g:h:t            define the number of packets to use during external loopback**

Selects the number of packets to send during the external loopback test D3 (g ,h ,t are the gigabit, hundred megabit, and ten megabit packet counts respectively) (default values are g=2000, h=1000, t=600). It is mandatory that all packets counts are entered with this command line option.

**-ckdev <xy>            executes diagnostic only x=chip rev and y=metal rev**

If <xy> does not match the device being accessed diagnostics will not execute. Use this option on a production line to verify that only chips of a specific rev are run with diagnostics and all others fail.

**-l200 <m200>            controls the number of 200 packets to send**

This option is used for the D4, "1G False Carrier loopback test. During the test packets are sent out in groups of 200. The user specified <m200> value determines the number of 200 packet groups to send out. The default value is <m200> =50 or 50 groups of 200 packets.

**-disppkts            display the packet count during the progress of the test**

This option is used for the D4, "1G False Carrier loopback test. On long-term test this option is entered to display the packet count every 20,000 packets to let a user know the test is operating normally and not frozen.

**-dispgbend            display the packet count at the end of the test**

This option is used for the D4, "1G False Carrier loopback test. This option is entered to print out the final packet count at the end of the test.

**-errlimit <lim>            controls the number of errors seen before the test fails**

This option is used for the D4, "1G False Carrier loopback test. The default value is <lim> =3.

**-pktf0            packet data pattern is alternating 64 1's and 0's**

This option is used for the D4, "1G False Carrier loopback test. The default packet data is an incrementing value per byte. With this option the packet data alternate between 64 1's and 0's that are aligned on a 64-bit PCI bus for maximum transitions.

**-dids            display chip information such as the PCI DID and VID**

The feature outputs the following information: PCI DID, VID, SDID & SVID; MAC address, Firmware revision, PXE, PXESpd, WOL, ASF, MBA, Bond Rev. This information can be used to verify the setup of a chip after a firmware upgrade.

**-elog <filename> an error log file <filename> used to log diagnostics failures**

The feature is used to log the failure information of the diagnostics run. When an error occurs if the file <filename> exists it is appended to and if it does not exist it is created. If there are no diagnostics failures then the file is not appended or created. A user abort of the diagnostics testing is log to the error log file <filename>.

**-firm <filename> used for field upgrade of bootcode firmware**

The feature is used to execute a field upgrade of bootcode firmware. The bootcode firmware is programmed into a/the device/s of a system if there is a match of the PCI DID, VID, SDID & SVID of firmware and device. For legacy bootcode, only VPD data block and Phase 1 and Phase 2 bootcode will be updated by default. For Selfboot firmware and Hardware Selfboot firmware, only patches will be updated.

**-firmall <filename> used for field upgrade for entire NVRAM image**

The feature is used to execute a field upgrade of entire NVRAM image. The new NVRAM image is programmed into a/the device/s of a system if there is a match of the PCI DID, VID, SDID & SVID of firmware and device. Media Manufact Region and Media VPD Block in NVRAM will be preserved. If the Advance Firmware, such as ASF and IPMI, is present, the Configuration Block of the firmware will also be preserved. Three additional command line parameters, “-updateasfcfg”, “-updatesecfg”, “-updateiscsicfg” and “-sil” can be used along with “-firmall” command.

**-updateasfcfg update the advance firmware configuration block of NVRAM with that of the NVRAM image provided by “-firmall” command**

When “-updateasfcfg” command line parameter is entered before “-firmall” command, the Advance Firmware Configuration Block of the NVRAM will not be preserved. It will get updated by the input file of “-firmall” command. The “-firmall” command is described above.

**-updatesecfg update the Media Manufact Region and Media VPD Block, of NVRAM with that of the NVRAM image provided by “-firmall” command**

When “-updatesecfg” command line parameter is entered before “-firmall” command, the Media Manufact Region and Media VPD Block of the NVRAM will not be preserved. They will get updated by the input file of “-firmall” command. The “-firmall” command is described above.

**-updateiscsicfg update the ISCSI firmware configuration block of NVRAM with that of the NVRAM image provided by “-firmall” command**



When “-updateiscsicfg” command line parameter is entered before “-firmall” command, the ISCSI Firmware Configuration Block of the NVRAM will not be preserved. It will get updated by the input file of “-firmall” command. The “-firmall” command is described above.

**-pipmi <filename> used for field program of IPMI firmware**

The feature is used to execute a field upgrade of NVRAM to add IPMI firmware. The firmware is programmed into a/the device/s specified by “-c” option switch.

**-uipmi <filename> used for field program of IPMI firmware**

The feature is used to execute a field upgrade of IPMI firmware. The firmware is programmed into a/the device/s specified by “-c” option switch if IPMI firmware is originally loaded in NVRAM

**-lbspd <spd> selects the line speeds to run test D3 (external loopback)**

The external loopback test, by default, runs line speeds of 10/100/1000 Mbits/s. The user can select to run any combination of the line speeds to run the external loopback via this option. The <spd> parameter specifies the operational speed by using t/h/g for 10/100/1000 Mbits/s respectively. To run test D3 at a 100 Mbits/s line rate the user would enter “b57diag -t abcd -T d3 -lbspd h”. To run test D3 at a 100 Mbits/s and 1 Gbit/s the user would enter “b57diag -t abcd -T d3 -lbspd hg”.

**-hlb <spd> puts the UUT into the host loopback mode**

All data sent to the device on the line side will be looped back to the line. This setup can be used to test a specific device with an external tester or can be used with the -tr option to perform a requestor-response test. The optional <spd> parameter specifies the operational speed by using 10/100/1000 for 10/100/1000 Mbits/s respectively. The default value is 1000 Mbits/s.

**-tr <pkts> <spd> the UUT transmits and receives data**

The UUT will transmit the number of packets specified by <pkts>. The default number of packets transmitted is 1,000,000. The UUT will receive all packets sent to it via the line side. Counts of the transmitted and received packets will be presented to the user. This setup can be used to test a specific device with an external tester or can be used with the -hlb option to perform a requestor-response test. The optional <spd> parameter specifies the operational speed by using 10/100/1000 for 10/100/1000 Mbits/s respectively. The default value is 1000 Mbits/s.

**-c <num> specify UUT device number**

When more than one device is in the system, the devices are numbered starting from zero. For example, if there are three devices detected, the devices are numbered as 0,1, and 2. In this case, by entering the parameter -c 2 will select the last found device as the default UUT.

In manufacture testing mode, by default, all devices are tested; however, if this option is used, only that selected device is tested.

Example: `-c 2`

**-l <file>        log file**

All diagnostic output can be saved in a log file. Type log file name is specified by this option. The default is no log file.

Example: `-l mylogfile.txt`

**-w <value>        enable WOL programming in manufacture mode**

After a successful manufacturing testing, the program will program WOL to either enable or disable mode (<value> = 1/0). By default, the WOL is programmed as disable. Entering value=1 will enable WOL.

When `-f` is entered, software uses eeprom.bin's content for WOL setting.

When `-w 1` is entered with `-f`, software forces WOL enabled.

**-x <value>        enable PXE in manufacture mode**

After a successful manufacturing testing, the program will program PXE to either enable or disable mode (<value> = 1/0). By default, the PXE is programmed as disable. Entering value=1 will enable PXE.

When `-f` is entered, software uses eeprom.bin's content for PXE setting.

When `-x 1` is entered with `-f`, software forces PXE enabled.

**-t <id>            disable test**

**-T <id>            enable test**

A certain test is enabled or disabled by default. User can overwrite the enabling status by those options. The test id must start with a letter 'A', 'B', 'C', or 'D' to indication the group and followed by test numbers. Each digit of number represents the sub-test number. For example, if the user wants to disable test A1 and A3. The option `-t A13` should be entered. If no test numbers entered, all tests in that group are selected. For the tests not specified, the default setting will be used. To enable or disable all tests, use the wildcard character "\*" by itself.

Example: `-t A15BC1 -T C4 -t D2`

This disables A1, A5, B1, B2, B3, B4, B5, B6, C1, D2 and enables C4

Example: `-t * -T d3`

This disables all tests, then enables just the D3 test.

Default Settings:

Enabled Tests:

- A1. Indirect Register Test
- A2. Control Register Test
- A3. Interrupt Test
- A4. Built In Self Test
- A5. PCI Cfg Register Test
- B1. Scratch Pad Test
- B2. BD SRAM Test
- B3. DMA SRAM Test
- B4. MBUF SRAM Test
- B5. MBUF SRAM via DMA Test
- C1. NVRAM Test
- C2. CPU Test
- C3. DMA Test
- C4. MII Test
- C5. VPD Test
- C6. ASF Hardware Test
- C7. ROM Expansion Test
- C8. CPU Fetch Test
- D1. Mac Loopback Test (not enabled on BCM5761)
- D2. Phy Loopback Test
- D5. MII Miscellaneous Test
- D6. MSI Test
- D7. E-Switch Test

Disabled Tests:

- B6. External SRAM Test
- D3. RJ45 Loopback Test
- D4. 1G False Carrier Test

For BCM5761 devices the following tests are available:

Enabled Tests:

- D8. SADB Test
- F3. Indirect APE Peripheral Register Test
- F4. APE Peripheral Register Test
- F5. APE Scratchpad Memory Test
- F6. APE Shared Memory
- F7. APE Shared Mem Indirect Access Test
- G1. APE CPU Memory Test
- G2. APE CPU Packet Test
- G5. APE CPU Event Register Test
- G6. APE CPU Mutex Register Test
- G8. APE CPU GRC Reset Test

Disabled Tests:

- D9. IPSec Loopback Test
- F1. Indirect APE Ctrl Register Test
- F2. APE Control Register Test
- F8. APE Mutex Register Test
- F9. APE Timers Test
- G3. APE CPU SMBus Loopback Test
- G4. APE CPU GPIO Register Test
- G9. APE USB Test (UEFI Only)

**-I <num>      iteration number**

Use this option to specify the number of times the tests to be run. The default is run one time. A number zero indicates loop forever. A control-C or control-break key can be used to break the loop. Any error detected will also stop testing after reporting the error.

Example: -I 5  
Run tests five times.

**-ver            display current version number**

If this option is entered, it displays the software version number/silkscreen revision and then exits the program.

**-e <code>      Encryption Code**

This option is required to use option -geneep, -f, -m, -n, -mac and -s.

**-geneep <file> Generate eeprom.bin file from eeprom.txt**

A password is needed to run this option. With this option, it updates the specified eeprom binary file with the specifications defined in eeprom.txt. Please see Section 6.0 EEPROM.TXT format for detailed argument description.

**-bus <bus:dev:func> Test UUT location**

If only bus number has been specified, the program will test all the UUTs at the specified bus number.

Example: -bus 2.

If bus number and device number have been specified, the program will test all the UUTs with the specified bus number and device number.

Example: -bus 2:4

If bus number, device number and function number have been specified, the program will only test the UUT with the specified bus number, device number and function number.

Example: -bus 2:4:1

This option should NOT be used along with -c option.

**-dpmi            Use DPMI memory allocation**

Use DPMI memory allocation method to allocate memory instead of malloc() or free()

**-f <file>        Program eeprom.bin**

The program programs the content of the specified file into EEPROM before testing.

**-m                Program MAC address**

If this option is entered, the program will prompt user for a new MAC address to be enter/scan before testing starts.

**-mac <mac address>            Program MAC address from command line**

If this option is entered, the program will retrieve MAC address right after the -mac option is entered. The mac address has to be entered in hex and as shown in the following example:

b57diag -mac 001018010203

**-fmac <file>        Program MAC address from a file**

If this option is entered, the program will retrieve MAC address from the specified file before starts testing. If the test passes, the MAC address from the specified file will be incremented; if not, it will stay unchanged. The text file which contains the MAC address range has the following format and the numbers are in hexadecimal:

mac\_addr\_pref = xxxxxx    => Which is the prefix of the MAC address.

mac\_addr\_start = xxxxxx   => Which is the start of the address range.

mac\_addr\_end = xxxxxx    => Which is the end of the address range.

Example:

mac\_addr\_pref = 001018

mac\_addr\_start = 000100

mac\_addr\_end = 000FFF

Working in conjunction with -f <file> option, this -fmac option is equivalent to option -m.

**-n Run program in Manufacturing Loop mode.**

With this option, the `-I`, iteration number option, is ignored. The program will run in manufacturing loop mode. Power on/off is supported. After each test, the program will prompt user to exchange the UUT before starts another testing.

**-s Skip eeprom programming process.**

With this option, the program will skip the eeprom programming process. However, it will check for the eeprom content and print a warning message if the content is not valid.

**The `-m` and `-f` combination will create the following behavior:**

**With both `-f` and `-m`:**

Program will not validate the eeprom content and go ahead to prompt user for the MAC address. It programs MAC address and EEPROM content and then checks the validity of eeprom content at the end of programming.

```
Loading EEPROM content from eeprom.bin: passed
Programming EEPROM from eeprom.bin....: passed
Checking EEPROM content.....: passed
```

**`-f` only:**

Program will check the validity of eeprom. If it is not valid, it will act as a), `-f -m` option. If it is good, it saves the MAC address from eeprom, program new eeprom binary file content into EEPROM and then restores the original MAC address. It checks the validity of eeprom content once more at the end of programming.

```
Checking EEPROM content.....: passed
Loading EEPROM content from <file>....: passed
Programming EEPROM from <file>.....: passed
Checking EEPROM content.....: passed
```

or

```
Checking EEPROM content.....: invalid
Loading EEPROM content from <file>....: passed
Programming EEPROM from <file>.....: passed
Checking EEPROM content.....: passed
```

**-m only:**

Program will check the validity of EEPROM. If it is not valid, it will act as a), -f -m option. If it is good, the program will prompt the user for a new MAC address and program the MAC address only. It checks the validity of EEPROM content once more at the end of programming.

```
Checking EEPROM content.....: passed
Programming MAC address.....: passed
Checking EEPROM content.....: passed
```

or

```
Checking EEPROM content.....: invalid
Loading EEPROM content from <file>....: passed
Programming EEPROM from <file>.....: passed
Checking EEPROM content.....: passed
```

d) no -m and -f options

Program will check the validity of EEPROM. If it is not valid, it will act as a), -f -m option. If it is good, it proceeds to normal diagnostics.

```
Checking EEPROM content.....: passed
```

or

```
Checking EEPROM content.....: invalid
Loading EEPROM content from eeprom.bin: passed
Programming EEPROM from eeprom.bin....: passed
Checking EEPROM content.....: passed
```

**-pasf <file>            Program ASF firmware**

This option needs to follow the -e and -c options. User has an option to program the ASF firmware from a single bin file that combined all 3 pieces of ASF bin files or from a specified text file that contain the file names of all 3 pieces of ASF bin files.

The program will detect input file type. If it is a single bin file, software will start programming. If the input file is a text file that contains the file name of the 3 pieces of ASF bin files, the program will retrieve the ASF firmware filenames from the specified text file then it will start programming. A sample asf.txt is provided and it has the following format: asf\_eep\_init = asfeinit.bin

asf\_eep\_cpua = asfecpua.bin

asf\_eep\_cpub = asfecpub.bin

**-ppxe <file>      Program PXE firmware**

This option needs to follow the `-e` and `-c` options. If this option is entered, the program will retrieve the PXE firmware filename from the command line then it will start programming.

Example:

```
b57diag -e <code> -c 0 -ppxe b57pxe.bin
```

**-mba <value>      Enable/Disable Multiple Boot Agent**

A value of 1 will enable Multiple Boot Agent and a 0 will disable.

Example:

```
b57diag -mba 0      : Disabling mba.
```

**-mbap <value>    Select Multiple Boot Agent Protocol**

value = 0    : Selecting PXE

value = 1    : Selecting RPL

value = 2    : Selecting BOOTP

Example:

```
b57diag -mbap 0
```

**-mbas <value>    Select Multiple Boot Agent Speed**

value = 0    : Selecting Auto

value = 1    : Selecting 10HD

value = 2    : Selecting 10FD

value = 3    : Selecting 100HD

value = 4    : Selecting 100FD

Example:

```
b57diag -mbas 0    : To select Auto speed mode.
```

**-pxes <value>    Select Multiple Boot Agent Speed**

value = 0    : Selecting Auto



value = 1 : Selecting 10HD  
value = 2 : Selecting 10FD  
value = 3 : Selecting 100HD  
value = 4 : Selecting 100FD

Example:

b57diag -mbas 0 : To select Auto speed mode.

**-disableeswitch    Disable E-Switch.**

Act like engineering mode "eswitch -f" command. Disable the E-Switch on the device that support eswitch, such as 5756. Without this option, PHY loopback test, external loopback test (pkttest -e), and carrier test are performed on both laptop mode and docking mode. With this option, above tests are performed on the default port.

**-h                    : High Resolution (80x50) Video Mode**  
**-p                    : Print on error**  
**-q                    : Quick diagnostic mode**  
**-asf <value>        : Enable/Disable (value = 1/0) ASF in manufacture mode**  
**-ipmi <value>        : Enable/Disable (value = 1/0) IPMI in manufacture mode**  
**-com <value>        : enable com port, value(1..4)**  
**-errctrl <c>         : On Error -> a:abort w:wait l:loop c:cont s:skip**  
**-ems <size>         : Enter external memory size in HEX to test**  
**-findref             : Detect reference device**  
**-lbm <n>             : Option to set mac loopback packets**  
**-lbp <n>             : Option to set phy loopback packets**  
**-lbe <n:n:n>         : Option to set external loopback packets, Format:  
                         <1000Mbps:100Mbps:10Mbps>**  
**-npol                : Select Negative Link Polarity in TBI test**  
**-ref                 : Run test with reference device**  
**-fail2               : On failure offset the failed message printout**  
**-ctpm                : Clears TPM data stored in NVRAM**  
**-optmem             : Option to allocate memory for optimization .  
                         ( For use with DOS B57diag on an EFI BIOS).**



## 5 Recover device from corrupt NVAM

When vendor id and/or device id have been modified and B57diag can longer find the device, the following steps can be followed to recover the missing device.

1. Use a PCI scan tool to scan all PCI devices in the system. Find out the vendor id and device id of the missing device.
2. Run B57diag with `-seldev` option switch to detect the missing device.  
e.g. `a:\b57diag -b57eng -seldev 14e411677`.
3. In B57diag engineering mode, clear the Magic value in the NVRAM.  
e.g. `0:>sewrite 0 0`
4. Reprogram the Bootcode firmware with “seprg” command.
5. Reboot system if needed.

## 6 EEPROM.TXT format

A set of commands is defined to allow user to change EEPROM.BIN content. To update EEPROM.BIN, user must enter `-e <code>` -genEEP options at the command prompt. A password must be entered to run this option. The 5704, Dual MAC, device use one single eeprom.bin to on both MAC channel configuration. Most of the configurations are shared expect the following commands:

PXE  
PXE\_SPEED  
WOL  
ASF

The WOL and ASF setting cannot be enabled on both channel at the same time. For example, if the primary WOL is already enabled, and the user try to enable secondary device's WOL, the primary's WOL setting will be disabled with the following message:

\*\* Warning, primary device WOL is disabled

By default, all commands configure the primary channel until the command MAC is used to select other channel.

### Syntax:

<Command> = <Argument>

xx           8-bit hex number  
xxxx        16-bit hex number  
xxxxxxxx    32-bit hex number  
d           decimal number ranges from 0 to 255  
string(n)   string of maximum size n.  
cc          2 bytes character  
n1..n2      a number ranges from n1 to n2.

#### For Boot Code:

MAC	= {0, 1}*
MAC_PREFIX	= xx:xx:xx
MAC_ADDRESS	= xx:xx:xx:xx:xx:xx
POWER_DISSIPATED	= d:d:d:d
POWER_CONSUMED	= d:d:d:d
SYSTEM_VENDOR_ID	= xxxx
SYSTEM_DEVICE_ID	= xxxx
SUBSYSTEM_VENDOR_ID	= xxxx
SUBSYSTEM_DEVICE_ID	= xxxx
PXE	= {enable, disable}
PXE_SPEED	= {auto, 10hd, 10fd, 100hd, 100fd, 1000fd}
WOL	= {enable, disable}
CABLE_SENSE	= {enable, disable}
PRODUCT_NAME	= string (48)

**PART\_NUMBER** = string (16)  
**ENGINEERING\_CHANGE** = string (10)  
**MANUFACTURING\_ID** = string (4)  
**ASSET\_TAG** = string (16)  
**FORCE\_PCI** = {enable, disable}  
**VOLTAGE\_SOURCE** = {1.3, 1.8}  
**LED\_MODE** = {mac\_mode, triple\_link /phy\_mode1, link\_speed/phy\_mode2,  
shared\_traffic, shasta\_mac, wireless\_combo}  
**MAX\_PCI\_RETRY** = {0..7, auto}  
**ASF** = {enable, disable}  
**DUAL\_MAC\_MODE** = {normal, mac0, mac1, xbar, swap, swapxbar} \*\*  
normal: Ch.0 and Ch. 1 enableds  
macb: Ch.0 enabled, Ch.1 disabled  
maca: Ch.0 disabled, Ch.1 enabled  
xbar: Both MACs shares one function in PCI configuration space  
swapxbar :  
swap: swap between Ch.0 and Ch1.  
**MBA\_BOOT\_PROTOCOL** = { pxe, rpl, bootp, iscsi}  
**MBA\_BOOTSTRAP\_TYPE** = {auto, bbs, int18, int19}  
**MBA\_DELAY\_TIME** = {0..15}  
**EXPANSION\_ROM\_SIZE** = {64K, 128K, 256K, 512K, 1M, 2M, 4M, 8M, 16M}  
**DESIGN\_TYPE** = {nic, lom}  
**VENDOR\_SPECIFIC0** = string (16)  
**VENDOR\_SPECIFIC1** = string (16)  
**REVERSE\_NWAY** = {yes, no}  
**WOL\_LIMIT\_10** = {yes, no}  
**FIBER\_WOL\_CAPABLE** = {yes, no}  
**CLOCK\_RUN\_SETTING** = {enable, disable}  
**DISABLE\_POWER\_SAVING** = {yes, no}  
**HIDE\_MBA\_SETUP\_PROMPT** = {enable, disable}  
**MBA\_SETUP\_HOT\_KEY** = {Ctrl-S, Ctrl-B}  
**ENABLE\_AUTO\_POWERDOWN** = {yes, no}  
**CAPACITIVE\_COUPLING** = {enable, disable}  
**PRI\_SMB\_ADDR** = {Hex(value)}  
**SEC\_SMB\_ADDR** = {Hex(value)}  
**L1ASPM\_Debounce\_En** = {enable, disable}

\*This should only be used to select port for dual ports devices.

\*\* This is for dual port devices only.

#### For Selfboot Firmware:

**MAC\_ADDRESS** = xx:xx:xx:xx:xx:xx  
**SYSTEM\_DEVICE\_ID** = xxxx  
**SUBSYSTEM\_VENDOR\_ID** = xxxx  
**SUBSYSTEM\_DEVICE\_ID** = xxxx  
**WOL** = {enable, disable}  
**WOL\_LIMIT\_10** = {yes, no}  
**DESIGN\_TYPE** = {nic, lom}  
**ENABLE\_AUTO\_POWERDOWN** = {yes, no}  
**REVERSE\_NWAY** = {yes, no}  
**DISABLE\_POWER\_SAVING** = {yes,no}  
**CABLE\_SENSE** = {enable, disable}  
**LED\_MODE** = {mac\_mode, phy\_mode1, phy\_mode2,  
shared\_traffic, shasta\_mac, wireless\_combo}  
**POWER\_DISSIPATED\_CONSUMED** = d0:d1:d2:d3:d4 \*

```

PCIE_POWER_BUDGETING_DATA = x0:x1:x2:x3:x4 **
PRODUCT_NAME               = {string (48)} ***
VPDR_SYSTEM_DEFAULT        = 1 ****
PART_NUMBER                = {string (16)}*****
ENGINEERING_CHANGE         = {string (10)}*****
SERIAL_NUMBER              = {string (16)}****
MANUFACTURING_ID           = {string (4)}*****
VENDOR_SPECIFIC0           = {string (16)}*****
    
```

\* If “d0” is 0, system default values will be selected. If “d0” is 1, suggested values will be used. If “d0” is 2, user defined values will be used. “d1” = Power Dissipated value in D0 state. “d2” = Power Dissipated value in D3 state. “d3” = Power Consumed value in D0 state. “d4” = Power Consumed value in D3 state. Power Dissipated suggested value in D0 state is 100. Power Dissipated suggested value in D3 state is 10. Power Consumed suggested value in D0 state is 100. Power Consumed suggested value in D3 state is 10.

\*\* If “x0” is 0, system default value will be selected. If “x0” is 1, suggested default values will be used. If “x0” is 2, user defined values will be used. “x1” = Power Budgeting Data 0 | Power Budgeting Data 1. “x2” = Power Budgeting Data 2 | Power Budgeting Data 3. “x3” = Power Budgeting Data 4 | Power Budgeting Data 5. “x4” = Power Budgeting Data 6 | Power Budgeting Data 7.

\*\*\* User defined values will be used.  
Suggested Product Name is “Broadcom NetXtreme Gigabit Ethernet Controller”

\*\*\*\* If “VPDR\_SYSTEM\_DEFAULT = 1” has been entered, System default value for all VPD-R data will be selected.

\*\*\*\*\*User defined values will be used.  
Suggested Part Number is “BCM957xx”  
Suggested Engineering Change Number is “106679-15”  
Suggested Serial Number is “0123456789”  
Suggested Manufacturing ID is “14e4”  
Suggested Vendor Specific Data is “”

Option f has been combined with option e in secfg for Selfboot Firmware.

**0:>secfg**

- 1. MAC Address .....: 001018000000**
- 2. Device Id .....: 1693**
- 3. Sub Vendor Id .....: 14E4**
- 4. Sub Device Id .....: 1693**
- 5. Wake on LAN .....: Disabled**
- 6. WoL Speed Limit 10 .....: Disabled**
- 8. LED mode .....: Disabled**
- 9. VPD Prod. Name and VPD-R Data .....: Disabled**
- a. Super-airplane Mode .....: Disabled**

- b. MBA Configuration
- c. NIC Design .....: LOM
- d. L1 PLL Disable .....: Yes
- e. Clkreq .....: No
- f. PCIE Link Polarity Fix Dis .....: No
- g. L1ASPM Debounce En .....: Yes
  
- x. Save & exit

----- Custom VPD Data (option 9)-----

**HW Selfboot VPD Data**

- 1. Part Number : BCM957xx
- 2. Engineering Change : 106679-15
- 3. Serial Number : 0123456789
- 4. Manufacturing ID : 14e4
- 5. Read only VPD Vendor Specific Data (V0) :
- 6. Product Name : Broadcom NetXtreme Gigabit Ethernet Controller

E	Custom VPD Data	Reference
Description: VPD Product description string.		

e-1	Part Number	Reference
Description: VPD part number..		

e-2	Engineering Change	Reference
Description: VPD engineering change.		

e-3	Serial Number	Reference
Description: VPD serial number.		

e-4	Manufacturing ID	Reference
Description: VPD manufacturing ID.		

e-5	Read only VPD Vendor Specific Data (V0)	Reference
Description: VPD V0 value. Data field provided for the customer.		

e-6	Product Name	Reference
Description: VPD Product description string.		

Example 1:

```
; This comment line
// This also can be used as comment line

// Blank line is also allowed

// This is Broadcom's MAC prefix
MAC_PREFIX = 00:10:18
POWER DISSIPATED = 10:0:0:100
POWER_CONSUMED = 10:0:0:100
SUBSYSTEM_VENDOR_ID = 14e4
SUBSYSTEM_DEVICE_ID = 1644
pxe = disable
PXE_Speed = 100fd
WOL = enable
Product_name = Broadcom Gigabit Ethernet Controller
PART_NUMBER = BCM95700A6
ENGINEERING_CHANGE = 106679-15
MANUFACTURING_ID = 14e4
Asset_Tag = XYZ1234567
DUAL_MAC_MODE = normal
MBA_BOOT_PROTOCOL = pxe
MBA_BOOTSTRAP_TYPE = bbs
MBA_DELAY_TIME = 6
EXPANSION_ROM_SIZE = 128K
DESIGN_TYPE = nic
; select other channel
MAC = 1
PXE = enable
```

*Example 2:*

```
; A. Super-airplane Mode (enable/disable)
super_airplane_mode = disable
; C. NIC Design (nic/lom)
nic_design = lom
; D. L1 PLL Disable (yes/no)
l1_pll_disable = no
; E. Clkreq (enable/disable)
ClkReq = disable
; F. PCIE Link Polarity Fix Dis (yes/no)
pcie_link_polarity_fix_dis = no
; G. L1ASPM Debounce En (yes/no)
L1ASPM_Debounce_En = no
; MBA 1. VLAN Enable (yes/no)
vlan_enable = yes
; MBA 2. VLAN ID (0..4095)
vlan_ID = 4095
; MBA 8. LINK SPEED (auto/10hd/10fd/100hd/100fd)
link_speed = 10fd
```



## 7 User Interface Commands

The commands are summarized in the following groups: vpd, nvram, cpu, dma, packet, mii, mem, test, power, irq, mac, misc, bridge and uart.

Note: For UEFI version, not all tests are supported. Refer to online help for more information.

### Command Group 'vpd'

vpdwrite	Write VPD Memory
vpdread	Read VPD Memory
vpdinfo	Show VPD Information

### Command Group 'nvram'

semode	Same as flshmode command
seread	Read NVRAM
sewrite	Write NVRAM
secfg	Configure NVRAM
seprg	Program NVRAM
uprg	Program userblock
upgfrm	Upgrade PXE or Boot Code from a File
sever	Display Serial NVRAM Version
sechksum	Check/Update Serial NVRAM checksum
sedump	Dump NVRAM content to a file
flshmode	Configure NVRAM mode
flshread	Same as seread
chkpxe	Check PXE code image
dir	Display file directory in NVRAM
iscsiprg	Program iSCSI from a File
erase	Erase file from directory in NVRAM
pxeprg	Upgrade PXE from a File
flshdev	Select flash device to access
bitbang	Generate BitBang pattern to SEEPROM bus
seclock	Set NVRAM config1 content value after reset
setorture	NVRAM reset torture test
seinit	Initialize NVRAM block
searb	Set/Report current NVRAM arbitration bit
seprotect	Set GPIO pin for NVRAM write protect
selclock	Set Legacy EEPROM clock value (bit 16-24 of EEPROM addr reg)
semap	Display NVRAM usage
setwol	Enable/Disable WOL
setpxe	Enable/Disable PXE
secomp	Compare EEPROM content against the file
dreset	Double reset test for EEPROM debug
userblock	Create a userblock in NVRAM
setmba	Enable Multiple Boot Agent
segenrc	Append a CRC to the specified block
secfghwsb1	Configure HW SelfBoot NVRAM
secfghwsb2	Configure HW SelfBoot NVRAM Group 2
secfghwsb3	Configure HW SelfBoot NVRAM Group 3
secfgsb1	Configure SelfBoot NVRAM
secfgsb2	Configure SelfBoot NVRAM Group 2
secfgsb3	Configure SelfBoot NVRAM Group 3
secfg1	Configure NVRAM Group 1
secfg2	Configure NVRAM Group 2
secfg3	Configure NVRAM Group 3

secfg4	Configure NVRAM Group 4
secfg5	Configure NVRAM Group 5
dirw	Write a directory entry to NVRAM
tpm	Write a directory entry for the TPM Block
umpcfg	Configure UMP in NVRAM
setipmi	Enable/Disable IPMI Passthrough Firmware
setump	Enable/Disable UMP
seotp	Configure OTP
otpchk	Check patch in OTP
defragment	Defragment NVRAM

**Command Group 'cpu'**

loadfw	Load Firmware to Tx/Rx CPUs
cpudtt	Dump Debugging Trace of TX CPU
cpudrt	Dump Debugging Trace of RX CPU
cputrace	Toggles CPU trace mode
haltcpu	Halt CPU
loadbootcode	Execute bootcode from file instead of NVRAM
disasm	Disassemble MIP instructions
step	Step MIP instructions
go	Start CPU
showgpr	Toggles showing CPU GPR mode
pc	Set current CPU program counter
breakpoint	Set current CPU break point
select	Select current CPU
u	Alias of disasm command
trap	Trap CPU memory
cpuinfo	Display CPU information
cpualive	Probe CPU if it is alive
cpumemstatus	Display internal CPU Memory test status
cpumemstart	Start internal CPU Memory test
cpumemstop	Stop internal CPU Memory test
cpumemset	Set internal CPU Memory test location
cpursttest	CPU Reset Test
cpufetch	CPU instruction fetch logic test
romcodetest	ROM Code Test

**Command Group 'dma'**

dmaw	DMA from NIC to Host Memory
dmар	DMA from Host to NIC Memory
dma_h	Dump DMA Entries
dma_d	Dump DMA Entries with Decode
dma_alloc	Allocate number of DMAD

**Command Group 'packet'**

mac1pk	Configure MAC loopback
blast	Blast Packets in Poll Mode
nicstats	Display NIC Statistics
ringindex	Dump Ring Index
phyctrl	Force Speeds/Duplex
txpkt	Transmit Packet
statusblk	Dump Status Block
stsbk	Dump Statistics Block
txcfg	Configure protocol packets for transmission
txsrcip	Configure source IP address for transmission
txdstip	Configure destination IP address for transmission
txsrcport	Configure source port address for transmission
txdstport	Configure destination port address for transmission
txtype	Configure packet type for transmission

txprot	Configure packet protocol field for transmission
txipprot	Configure packet IP protocol for transmission
rxcfg	Configure Rx parameters
tprot	Blast with TPROT Packets
qstat	Get a quick NIC statistic
drvrcfg	Configure driver parameters
irt	Test an individual register
macmrd	Test an individual register
miimrd	Test an individual register
miiloop	Test an MII/MII EXP/ MII TAP register
pcimrd	Test an individual register
inband	Force inband

### Command Group 'mii'

mwrite	Write PHY registers via MII Management interface
mread	Read PHY registers via MII Management interface
msreadlc	Read PHY Shadow registers 0x1C via MII Management interface
msreadl8	Read PHY Shadow registers 0x18 via MII Management interface
readexp	Read PHY Expansion Registers via MII Management interface
sdwrite	Write SERDES registers via MII Management interface
sdread	Read SERDES registers via MII Management interface
mdev	Select current PHY to be accessed
miimode	Select Auto Mode of MII Access
lbertram	Load data to PHY BIST RAM
dbertram	Dump PHY BIST RAM
bertstats	Dump PHY BIST Statistics
rm	Read MII Registers
mrloop	Loop on MII read
phymse	PHY mean square error
initphy	Initialize PHY
tapdump	Read PHY internal TAP values
m2dev	Select current PHY to be accessed

### Command Group 'mem'

memsearch	Search for a data pattern in memory
read	Read Memory
write	Write Memory
poll	Poll Memory
setbit	Read-Modify Memory by ORing with <bits>
clearbit	Read-Modify Memory by ANDing with ~<bits>
readbit	Read-Modify Memory by ored with <bits>
cread	Read PCI configuration space of specified device
cwrite	Write PCI configuration space of specified device
pcird	Read PCI Configuration Word
pciwr	Write PCI Configuration dWord

### Command Group 'test'

vpctest	Run VPD Memory Test
regtest	Run Register Test
miitest	Run MII Memory Test
msi	Run MSI Test Manually
memtest	Run Memory Test
setest	Run NVRAM Test
bist	Run BIST
nictest	Run a set of NIC Tests
intrtest	Run Interrupt Test
pktttest	Run Packet Tests

<code>cputest</code>	Run CPU Test
<code>dmatest</code>	Run DMA Test
<code>teste</code>	Enable Test
<code>testd</code>	Disable Test
<code>asftest</code>	ASF Test
<code>bustest</code>	PCI Bus Test
<code>sramtest</code>	SRAM test
<code>msitest</code>	Run MSI Test
<code>romtest</code>	ROM Test
<code>gpiotest</code>	Do GPIO test
<code>cpudiag</code>	Run diagnostic from internal CPU
<code>pcicfgtest</code>	Run PCI Config. Reg. Test
<code>petest</code>	Perform parity error test on a bridge
<code>errctrl</code>	Configure Error Control Setting
<code>sedvt</code>	Perform NVRAM dvt test
<code>miimisctest</code>	Run MII Misc. Tests
<code>cpugprtest</code>	Run CPU GPR test
<code>carrierptest</code>	Run False Carrier Test
<code>wseq</code>	Run Sequential Write Test
<code>umplb</code>	UMP Loopback Test
<code>dmashasta</code>	Run DMA Test On Shasta Family Of Devices
<code>apetest</code>	Run APE Self Tests

**Command Group 'power'**

<code>pmdcfg</code>	Dump Power Management Info
<code>pmpcfg</code>	Add/Del Pattern
<code>pmpd</code>	Power Down MAC

**Command Group 'irq'**

<code>intr</code>	Dump Interrupt Info
<code>intrctrl</code>	Mask/Unmask Current Interrupt
<code>intt</code>	Interrupt Tracer

**Command Group 'mac'**

<code>mbuf</code>	Dump Content of MBUFs
<code>loaddrv</code>	Load Driver
<code>unloaddrv</code>	Unload Driver
<code>machalt</code>	Halt MAC Controller
<code>ftq</code>	Dump FTQ
<code>addmc</code>	Add Multicast MAC
<code>delmc</code>	Delete Multicast MAC
<code>txmacdes</code>	Program Destination address to UUT
<code>txmacsrc</code>	Program Source address to UUT
<code>chkldrv</code>	Check to see if driver is loaded
<code>vlantag</code>	Display/Clear vlan tag information

**Command Group 'misc'**

<code>reg</code>	Register wizard
<code>exit</code>	Exit the System
<code>debug</code>	Debug functions
<code>gpiowrite</code>	Write a Value into GPIO pin
<code>gpioread</code>	Read GPIO Value
<code>pxecpy</code>	Load PXE Code to MBUF Memory
<code>device</code>	Show or Switch Device
<code>version</code>	Display Program Version
<code>help</code>	Display the Commands Available
<code>?</code>	Alternate Help Command
<code>radix</code>	Change System Radix
<code>apeaccmode</code>	Select between BAR1 Mode and Indirect Access Mode

nolog	Close the Current Logfile
log	Open Logfile
pciinit	Initialize PCI Configuration Registers
pciscan	Scan for All PCI Devices
dos	Execute DOS command
diagcfg	Configure Diagnostics
reset	Reset Chip
sadbtest	SADB test for IPSec supported devices
sadbutil	Utilities for SADB table, only valid for IPSec supported devices
quit	Exit the System
cls	Clear Screen
loop	Loop on a command
dbmode	Set DEBUG Mode to On or Off Mode
new	Display new commands available
sleep	Suspend command execution
fillpattern	Fill WOL matching pattern into Misc. Memory Location
inp	Input port
outp	Output port
switch	Alternate 'device' command
do	Execute commands from a file
txfill	Fill tx buffer with pattern and packet length (14-9018)
wbuf	Write tx rx buffer with specified data at offset
rbuf	Read tx rx bistin bistout bistex buffer
cpbuf	Copy the content of rx buffer into tx buffer
echo	Echo <string> to the screen
pause	Pause for user to hit a key
q	Exit the System, alias name for quit
verbose	Change verbose settings
beep	Create a beep sound
var	Display current variables
meminfo	Report memory information
delvar	Delete local variables
regdump	Dump register content to a file
regcomp	Compare register content to a file
regrestore	Restore register content from a file
showcurrcard	Dump Current Card Info
showglobal	Dump Global Info
nvsize	Programs the NVRAM and TPM size in Kbyte to NVRAM
aspm	Configure PCI-E Link power state operation
ledcfg	Configure the LED Mode (Shasta Specific)
ledblink	Blinks the LED
nmbuf	Select No. of Mbuf
umpecho	UMP Echo Test
dids	Display detail device information for all devices
serial	Compares the serial no. to defaults or the provided values
power	Verify the operation of the power capability (575x Cx specific)
binchksum	Run Checksum test on input firmware file
eswitchtest	Run E-Switch Timer Test
eswitch	Switch E-Switch between Docking and Laptop
sbfcfg	Self-boot file config.
linkintrtest	Link Changed Interrupt Test
ipv6extld	Load IPv6 Extension Header File
ipv6exten	Enable IPv6 Extension Headers (1=En, 0=Dis)
cfgpagesize	Configure Flash page size to 256B or 512B
cpmu	CPMU tests
mdio	MDIO tests
wait	delay a bit
ipsecsupport	To check if the device support IPSec.

### Command Group 'man'

asfcfg	Configure Management Firmware in NVRAM (AKA mancfcg)
mancfcg	Configure Management Firmware in NVRAM or configuration file
asfprg	Program ASF firmware into NVRAM (3 bin files)
smbusrun	Operate SMBUS in Auto or BitBang mode
smbusmode	Set SMBUS in Auto or BitBang mode
setasf	Enable/Disable ASF Firmware (AKA setman)
setman	Enable/Disable Management Firmware
asfmailbox	Create or Read ASF Mailboxes
asfeng	Enable/Disable ASF engineering mode
apeinfo	Display APE-related information
apectl	Control APE
apeping	send APE event to ping host from APE
aperead	Read APE local view data thru APE
apeotpkey	Print APE OTP key value
apelog	Create, view, and manage NVRAM-based APE Logs

### Command Group 'bridge'

readbr	Read bridge's configuration space
writebr	Write to bridge's configuration space
findbridge	Find all PCI bridges in the system
bridge	Switch to specified bridge
pere	Enable parity error response on a bridge
perd	Disable parity error response on a bridge
peclr	Clear parity error on bridge
pechk	Clear parity error on a bridge

### Command Group 'uart'

uart	Disable/Enable/Access/Test Virtual UART
------	---

## 8 Special Instruction

1. Mac register test:

Unload MAC driver before running test.

2. Memory test:

Unload MAC driver before running test.

3. DMA test:

Unload MAC driver before running test.

4. TX RX packets:

TX sides need to be configured (txcfg).

RX sides need to be configured (rxcfg).

Configure MAC and PHY loop back.

Call txpkt to transmit packets.

5. The following tests need to setup test configuration before running.

To setup test configuration, run "diagcfg". Diag config can be saved in system for future use.

Test:

Memory test

NIC test

6. Unload driver before power down NIC card.

7. Load driver after power up NIC card.

8. Blast Test:

Load MAC driver before running test.

## 9 Same System Send/Respond Test

During a send/respond test two Broadcom Ethernet devices in the same system are connected via a CAT5 (RJ45) cable and data is transmitted and received across the cable. In the above setup one device is the device under test (DUT) and the other is the reference device. Any Broadcom device that is supported by diagnostics can be used as the DUT or the reference device.

### 9.1 Identifying a reference device

A reference device needs to be identified in order to test a DUT against the reference device. The user can use the “b57diag –dids” command to display all the devices in the system. Each device is assigned a number that is displayed on the screen in the first column just before the device type. A user selects the number of the device that is to be used as the reference and uses this number to test the DUT as described in the next section.

### 9.2 Testing the DUT against the reference device

After identifying the reference device, as described in the previous section, the user can test the DUT by running “b57diag –rf 0 –c 1 –t abcd –T d3”. Each of the previous options are described in the table below (details of each option are provided in the “Command line option parameters” section above).

-rf 0	Identifies the reference device as device 0.
-c 1	Identifies the DUT as device 1.
-t abcd	Disables all tests (Disable all tests in group a, b, c, and d).
-T d3	Perform the external loopback test that will transmit data from the DUT to the reference device, via the CAT5 cable, then the reference device will receive the data and transmit it back, via the CAT5 cable, to the DUT which will verify the received data.

### 9.3 Other Options for the Send/Respond test

The following options can be used with the send/respond test (details of each option are provided in the “Command line option parameters” section above).

-lbspd t/h/g	Selects any combination of line speed to run the loopback test (by default runs all line speeds).
-lbe g:h:t	Selects the number of packets to send during the loopback test (g, h & t are the 1000/100/10 megabit packet counts respectively) (default values are g=2000, h=1000, t=600). It is mandatory that all packets counts are entered with this command line switch.



## 10 Test and Functions Description

### 10.1 vpdwrite

**Command:** vpdwrite

**Description:** Write data to VPD storage.

**Syntax:** vpdwrite <start[-end | len] value> | <filename>

File format:

Address range: 0x00 – 0xFF

num\_bytes: 256 (max)

If only one argument is entered, filename is assumed. Otherwise, 'start [len] value' format must be used.

### 10.2 vpdread

**Command:** vpread

**Description:** Read data from VPD storage

**Syntax:** vpdread start[-end | len]

Address range : 0x00 – 0xFF

num\_byte : 256 (max)

### 10.3 vpdinfo

**Command:** vpdinfo

**Description:** Show VPD Information

**Syntax:** vpdinfo

### 10.4 semode

**Command:** semode

**Description:** Configure NVRAM Mode.

**Syntax:** semode [auto | passthru | legacy | new]

**Example:**

1. Set Serial EEPROM mode to Auto.

```
0:> semode auto
```

2. Display Current mode

```
0:> semode
```

```
Current mode : New NVRAM Access, Auto  
device: Flash – ATMEL (with buffer)
```

## 10.5 seread

**Command:** seread

**Description:** Read NVRAM

**Syntax:** seread start[-end | len]

**Note:** Use “semode” command to configure the interface mode.

### Example:

1. Set number base to hex, then read and display serial eeprom locations from 0x00 to 0x20

```
0:> radix 16  
0:> seread 0-20  
*** Dump Serial EEPROM (Auto Mode) ***  
000000: 669955aa 08000000 00000069 00000200 d97b07d0 00000000 00000000 00000000  
  
000020: 00000000
```

2. Set number base to hex then read location 0x18 of serial eeprom.

```
0:> radix 16  
0:> seread 18 1  
*** Dump Serial EEPROM (Auto Mode) ***  
000018: 000000ff
```

## 10.6 sewrite

**Command:** sewrite

**Description:** Write NVRAM

**Syntax:** sewrite start[-end | len] data

**Note:** Use “semode” command to configure the interface mode.

### Example:

1. Set number base to hex, write 0x55AA to serial eeprom from locations 0x30 to 0x35

```
0:> radix 16  
0:> sewrite 30-35 55AA  
*** Write Serial EEPROM (Auto Mode) ***
```

2. Set number base to hex, write 0x2 to serial eeprom location 0x25

```
0:> radix 16
0:> sewrite 25 2
*** Write Serial EEPROM (Auto Mode) ***
```

## 10.7 secfg

**Command:** secfg

**Description:** Configure Boot Code or Selfboot firmware.

All option is for Boot Code only. If selected program with defaults (-f=1), eeprom.bin must be found in the same directory of b57diag.exe.

**Syntax:** secfg

**Options:**

-v<n> : verbose level (0,1,2) (def=0)

-f : force to program with defaults

**Example: (For Boot Code).**

- 1 Program Serial EEPROM with defaults value and set verbose level to 0.

```
0:> secfg
Reading current NVRAM ... OK
Validating content...

1. MAC Address : 00:10:18:04:1a:36
2. Power Dissipated (D3:D2:D1:D0) : 10:0:0:100
3. Power Consumed (D3:D2:D1:D0) : 10:0:0:100
4. Vendor ID : 14E4
5. Vendor Device ID : 1653
6. Subsystem Vendor ID : 14E4
7. Subsystem Device ID : 1653
8. PXE (does not apply to PXE in system ROM)
   { Enable(1), Disable(2) } : Disable
9. PXE Link Speed { Auto(0),10HD(1),10FD(2)
   100HD(3),100FD(4) } : Auto
10. Magic Packet WoL { Enable(1), Disable(2) } : Disable
11. Product Name : Broadcom NetXtreme Gigabit Ethernet Controller
12. Part Number : BCM95705A50
13. Engineering Change : 106679-15
14. Serial Number : 0123456789
15. Manufacturing ID : 14e4
16. Asset Tag : XYZ01234567
17. Part Revision : A0
18. Voltage { 1.3V(0), 1.8V(1) } : 1.8V
19. Force PCI Mode { Enable(1),Disable(2) } : Disable
20. PHY Type { Copper(1),Fiber(2) } : Copper
21. Led Mode { Mac Mode(0), Phy Model (1), Phy Model2 (2),
   Shasta Mac Mode (3), SharedTraffic (4),
   Wireless Combo (5)} : Phy Model1
22. PHY ID ([PHY#2][PHY#3]) : 00206160
23. Max PCI Retry {0-7, 8=auto} : 8
24. Advanced firmware feature (ASF/IPMI/UMP)
   { Enable(1), Disable(2) } : Disable
```

```

25. Dual MAC mode {Normal(0), MAC_B only(1)
   MAC_A only(2), XBAR(3), swap(4), swapXBAR(7)}: 0
26. MBA Boot Protocol {PXE(0), RPL(1), BOOTP(2),
   iSCSI(3)} : 0
27. MBA Bootstrap Type
   {Auto(0), BBS(1), Int18(2), Int19(3)}: 0
28. MBA Delay Time (0-15) : 0
29. Expansion ROM size (does not apply to PXE in system ROM)
   { 64k(0), 128k(1), 256k(2),512k(3), 1M(4), 2M(5),
   4M(6), 8M(7), 16M(8) } : 0
30. Design Type: {NIC(0), LOM(1)} : NIC
31. Read only VPD Vendor Specific Data (V0) :
32. Read/Write VPD Vendor Specific Data (V1) :
33. Reversed Nway {No(0), Yes(1)} : No
34. Limit WoL Speed to 10 {No(0), Yes(1)} : No
35. Fiber WoL Capable {No(0), Yes(1)} : No
36. Clock-Run Setting {Disable(0), Enable(1)} : Disable
37. Enable PHY Auto Powerdown {No(0), Yes(1)} : No
38. Disable PowerSaving capability {No(0),Yes(1)}: No
39. Hide MBA Setup Prompt {Disable(0), Enable(1)}: Disabled
40. MBA Setup Hot Key {Ctrl-S(0), Ctrl-B(1)} : 0
41. Capacitive Coupling (5705 only)
   {Disable(0), Enable(1)} : Disabled
42. SERDES TX Drvr Pre-Emp - Primary (5704 only) : 0
43. SERDES TX Drvr Pre-Emp - Second (5704 only) : 0
44. SERDES TX Drvr Pre-Emp - ENABLE (5704 only)
   {Disable(0), Enable(1)} : Disabled
46. Encoded Hot Plug Power Value 1 & 2 For NIC : 00000000
47. Encoded Hot Plug Power Value 3 & 4 For NIC : 00000000
48. Encoded Hot Plug Power Value 5 & 6 For NIC : 00000000
49. Encoded Hot Plug Power Value 7 & 8 For NIC : 00000000
50. Hot Plug Power {Disable(0), Enable(1)} : Disabled
51. Pri. Port SMB Address (ASF/IPMI) : A4
52. Sec. Port SMB Address (IPMI) : A6
53. Cable Sense/Low Power Energy Detect {Disable(0), Enable(1)} : Disabled
54. GPIO 0 Config {Input(0), Output Hi(1),
   Output Lo(2)} : Input
55. GPIO 2 Config {Input(0), Output Hi(1),
   Output Lo(2)} : Input
56. L1ASPM Debounce En {Disable(0), Enable(1)} : Disabled
57. Link Aware Mode {Disable(0), Enable(1)} : Enabled
58. Link Speed Power {Dis able(0), Enable(1)} : Enabled

```

Enter your choice (option=paramter/save/cancel) ->

**Description of above parameters:**

To enter the ‘secfg’ menu shown above a user type in ‘secfg’ at the diagnostics command line. A menu is present to the user showing the ‘secfg’ options shown above. A user can change the ‘secfg’ parameter by entering the option number, followed by an equal sign, and then followed by the parameter selected. To enable ASF (option 24) a user would enter “24=1” at the diagnostics ‘secfg’ command line, which would look like the following:  
 “Enter your choice (option=paramter/save/cancel) -> 24=1”.

The tables below provide details for each parameter of the ‘secfg’ options. The top right of each table has the word “Configure” or “Reference”. “Configure” signifies that the option is used to configure the operating characteristics of the device or default values of registers in the device. “Reference” signifies that the option is use for display purposes and does not affect the operating characteristics of the device or default values of registers in

the device.

1	MAC Address	Configure
Description: MAC address of the device.		
2	Power Dissipated (D3:D2:D1:D0)	Reference
Description: Power dissipated in D3-0 states.		
3	Power Consumed (D3:D2:D1:D0)	Reference
Description: Power consumed in D0-2 states.		
4	Vendor ID	Configure
Description: PCI Vendor ID. Default 0x14e4.		
5	Device ID	Configure
Description: PCI Device ID.		
6	Subsystem Vendor ID	Configure
Description: PCI Subsystem PCI Vendor ID.		
7	Subsystem Device ID	Configure
Description: PCI Subsystem PCI Device ID.		
8	PXE	Configure
Description: Enable the Pre-Boot Execution Environment (PXE) by using this option. This option does not apply to PXE in system ROM. Options 8,9, 26 - 29 and 39-40 configure the boot protocol behavior.		
Enable(1)	When PXE is enable the expansion ROM enable bit in the PCI configuration space is set for system boot up. The user has to make sure the PXE code is loaded into the non-volatile memory by using the "loadpxe" command.	
Disable(2)	When PXE is disable the expansion ROM enable bit in the PCI configuration space is cleared for system boot up. The PXE code does NOT have to been load into the non-volatile memory of the device.	
9	PXE Link Speed	Configure
Description: Pre-Boot Execution Environment line configuration is specified using the parameters below. Options 8,9, 26 - 29 and 39-40 configure the boot protocol behavior.		
Auto(0)	PXE auto detects the link configuration.	
10HD(1)	PXE uses a 10 Mbits/s, half duplex line configuration.	
10FD(2)	PXE uses a 10 Mbits/s, full duplex line configuration.	
100HD(3)	PXE uses a 100 Mbits/s, half duplex line configuration.	
100FD(4)	PXE uses a 100 Mbits/s, full duplex line configuration.	
10	Magic Packet WoL	Configure
Description: A system can be configured to power-on when a Magic Packet is received.		

Enable(1)	The device will assert the pme signal, to power on the system, when a magic packet is received.	
Disable(2)	Magic packets are ignored.	
Note: A driver can setup the WoL behavior of a device and the value programmed into this location is ignored.		
11	Product Name	Reference
Description: VPD Product description string.		
12	Part Number	Reference
Description: VPD part number..		
13	Engineering Change	Reference
Description: VPD engineering change.		
14	Serial Number	Reference
Description: VPD serial number.		
15	Manufacturing ID	Reference
Description: VPD manufacturing ID.		
16	Asset Tag	Reference
Description: VPD asset tag.		
17	Part Revision	Reference
Description: VPD part revision.		
18	Voltage (5701 only)	Configure
Description: Controls the voltage source level for the 5701 controller.		
1.3V(0)	Selects a 1.3V source.	
1.8V(1)	Selects a 1.8V source.	
19	Force PCI Mode	Configure
Description: PCI bus operational mode configuration.		
Enable(1)	When enabled the device uses PCI mode, instead of PCI-X, independent of the capabilities of the slot the device is plugged into.	
Disable(2)	When disabled the device uses the PCI mode of the slot the device is plugged into; if the device is capable of operating in the required mode.	
20	PHY Type {option no longer supported}	Configure
Description: PHY line type configuration.		
Copper(1)	The communication medium is copper.	
Fiber(2)	The communication medium is fiber.	

21	Led Mode	Configure												
Description: A device can be configured to use one LED to indicate speed and activity or three LEDs. Use a devices data sheet to verify the LED modes supported by a devices and for exceptions to the LED modes described below.														
Mac Mode (0)	The LEDs are controlled by the value in the Mac LED Control Register.													
Phy Mode1 (1)	Three LEDs are used for 10/100/1000 Mbits/s and each is driven individually by the device.													
Phy Mode2 (2)	One LED is used for 10/100/1000 Mbits/s and is connected as described below.													
	5700/01 – The Link10 line will indicate link for all speeds. Link100 and Link1000 will encode the line speed as show in the following table:													
	<table border="1"> <thead> <tr> <th>Link 100</th> <th>Link 1000</th> <th>Speed</th> </tr> </thead> <tbody> <tr> <td align="center">0</td> <td align="center">0</td> <td align="center">10 Mbits/s</td> </tr> <tr> <td align="center">1</td> <td align="center">0</td> <td align="center">100 Mbits/s</td> </tr> <tr> <td align="center">0</td> <td align="center">1</td> <td align="center">1000 Mbits/s</td> </tr> </tbody> </table>	Link 100	Link 1000	Speed	0	0	10 Mbits/s	1	0	100 Mbits/s	0	1	1000 Mbits/s	
	Link 100	Link 1000	Speed											
0	0	10 Mbits/s												
1	0	100 Mbits/s												
0	1	1000 Mbits/s												
For all other 57xx devices the three link lines operate in an open drain configuration and can be tied together with a pull up resistor to control a LED.														
SharedTraffic (3)	Mode specific to the 575x and 572x chips. The link LED performs a dual role: solid when there is a link and blinks when there is activity.													
Shasta Mac Mode (4)	Mode specific to the 575x and 572x chips. The link LED blinks only when traffic is for Shasta.													
Wireless Combo (5)	Mode specific to the 575x and 572x chips. When link is lost the LEDs are driven by inputs pins that are connected to the LED signals of a wireless link.													
22	PHY ID	Reference												
Description: Only for display purposes not used in software.														
[PHY#2][PHY#3]														
23	Max PCI Retry	Configure												
Description: The maximum number of time to retry an aborted PCI operation.														
0-7	The number of MAX PCI retries is force by the users configuration.													
8=auto	The MAX PCI retry field is selected dynamically by firmware based on PCI bus type detected.													
24	Advanced firmware feature (ASF/IPMI/UWP)	Configure												
Description: Advanced firmware feature such as Alert Standard Format (ASF) or IPMI is enabled by this option.														
Enable(1)	If the ASF/IPMI/UWP code is loaded in NVRAM it is loaded and executed by the device.													
Disable(2)	No ASF/IPMI/UWP functionality is provided when disable.													
25	Dual MAC mode	Configure												
Description: This option is only valid for a dual port device such as a 5704. The physical ports will be named MAC_A an MAC_B for the discussion below.														
Normal(0)	MAC_A and MAC_B are available and are supported as PCI function 0 & 1.													
MAC_B only(1)	MAC_B is available and is supported as PCI function 0. MAC_A is disable.													

MAC_A only(2)	MAC_A is available and is supported as PCI function 0. MAC_B is disable.
XBAR mode(3)	MAC_A and MAC_B are available and are supported as PCI function 0. MAC_A is the primary device and MAC_B is mapped in the space of MAC_A. To the OS and BOIS this looks like one port. The driver will trunk the two ports together as one logical port which allows for load balancing, failover/recovers, turbo-teaming, etc.
swap(4)	Physical port MAC_A and MAC_B are available and are supported as PCI function 1 and 0 respectively.
swapXBAR(7)	MAC_A and MAC_B are available and are supported as PCI function 0. MAC_B is the primary device and MAC_A mapped in the space of MAC_B. To the OS and BOIS this looks like one port. The driver will trunk the two ports together as one logical port which allows for load balancing, failover/recovers, turbo-teaming, etc.

26	MBA Boot Protocol	Configure
Description: Select a Multiple Boot Agent. Options 8,9, 26 - 29 and 39-40 configure the boot protocol behavior.		
PXE(0)	PreBoot Execution Environment (PXE) is the boot protocol.	
RPL(1)	Remote Program Load (RPL) is the boot protocol.	
BOOTP(2)	Boot Protocol (BOOTP) is the boot protocol.	
ISCSI(3)	ISCSI is the boot protocol.	

27	MBA Bootstrap Type	Configure
Description: The BIOS bootstrap methods listed below are supported. Options 8,9, 26 - 29 and 39-40 configure the boot protocol behavior.		
Auto(0)	Automatically configured use one of the methods below.	
BBS(1)	A BIOS that supports the BIOS Boot Specification (BBS) can initiate the bootstrap method via the expansion ROMs Bootstrap Entry Vector.	
Int18(2)	A INT18 is used to initiate the bootstrap method.	
Int19(3)	A INT19 is used to initiate the bootstrap method.	

28	MBA Delay Time	Configure
Description: The amount of time the MBA boot message is displayed, in order to give a user the option to enter the BOOT parameter configuration screen.		
0-15	The number of seconds the MBA banner is displayed.	

29	Expansion ROM size	Configure
Description: This value determines the size of the expansion ROM. This option does not apply to PXE in system ROM. Options 8,9, 26 - 29 and 39-40 configure the boot protocol behavior.		
64k(0)	Advertises 64k expansion ROM size.	
128k(1)	Advertises 128k expansion ROM size.	
256k(2)	Advertises 256k expansion ROM size.	
512k(3)	Advertises 512k expansion ROM size.	
1M(4)	Advertises 1M expansion ROM size.	
2M(5)	Advertises 2M expansion ROM size.	
4M(6)	Advertises 4M expansion ROM size.	
8M(7)	Advertises 8M expansion ROM size.	
16M(8)	Advertises 16M expansion ROM size.	

30	Design Type	Configure
----	-------------	-----------



Description: Select NIC or LOM based Ethernet controller.		
NIC(0)	Option selected for a plug in network interface card.	
LOM(1)	Option selected for a LAN on motherboard.	
Note: The firmware and the drivers use this to determine the operational characteristics of a device. For example, on a LOM GPIO2 is tied to the SEEPROM write protect pin and on a NIC GPIO 1 and 2 is used to switch between main and auxiliary power for wake on LAN functionality.		
31	Read only VPD Vendor Specific Data (V0)	Reference
Description: VPD V0 value. Data field provided for the customer.		
32	Read/Write VPD Vendor Specific Data (V1)	Reference
Description: VPD V1 value. Data field provided for the customer.		
33	Reversed Nway	Configure
Description: Nway Negotiation.		
0	(default) Auto-negotiation is done from 1000->100->10 Mbits/s.	
1	Auto-negotiation is done from 10->100->1000 Mbits/s.	
Note: A user could set the chip for option 1 when running a laptop on battery power. The chip would auto-negotiate starting at a lower speed and lower power. The chip would only go to higher line rates and higher power if the lower line rates were unavailable.		
34	Limit WoL Speed to 10	Configure
Description: Limit Wake on LAN (WoL) line speed.		
No(0)	10 or 100 Mbits/s is used for WoL.	
Yes(1)	Only 10 Mbits/s is used for WoL.	
35	Fiber WoL Capable	Configure
Description: Fiber Wake on LAN (WoL) Capable enable.		
No(0)	Disable Fiber WoL.	
Yes(1)	Enable Fiber WoL.	
36	Clock-Run Setting	Configure
Description: Enable Clock-Run on mini-PCI/cardbus systems. This parameter is valid only for the 5705 A0-A2.		
Enable(1)	Sets the clock mode register bit 22 which will assert (active low) the clock run signal on the bus prior to any PCI configuration space activity.	
Disable(0)	The above is not performed.	
37	Enable PHY Auto Powerdown	Configure
Description: Enable GPHY auto-power down when there is no link present (to conserve power).		
No(0)	The PHY will not auto power down.	
Yes(1)	The PHY will auto power down when there is no link.	
38	Disable Power Saving capability	Configure
Description: Disable Power Saving capability setup by option 33 above. When disable a device will use Nway negotiation.		

No(0)	Power-saving capability active.
Yes(1)	Power-saving capability inactive.

39	Hide MBA Setup Prompt	Configure
Description: During the MBA boot, the MBA setup prompt is displayed to provide the user the option to setup and configure various MBA parameters. Options 8,9, 26 - 29 and 39-40 configure the boot protocol behavior.		
Disable (0)	Hide the MBA Setup Prompt. The user is NOT give the option to change the MBA boot parameters.	
Enable (1)	Show the MBA Setup Prompt. The user is given the option to change the MBA boot parameters.	

40	MBA Setup Hot Key	Configure
Description: Hot Key used to enter the MBA Setup. Options 8,9, 26 - 29 and 39-40 configure the boot protocol behavior.		
Ctrl-S (0)	MBA Setup entered via ctrl-s.	
Ctrl-B (1)	MBA Setup entered via ctrl-b.	

41	Capacitive Coupling	Configure
Description: Enable/Disable the PHY to operate with capacitors as line isolators for the 5705 family.		
Disable (0)	Disable capacitive coupling.	
Enable (1)	Enable capacitive coupling.	

42	SERDES TX Drvr Pre-Emp – Primary (5704 only)	Configure
Description: The TX driver pre-emphasis value that is used for the primary device of a 5704 if it is enabled by option 44 below.		

43	SERDES TX Drvr Pre-Emp – Secondary (5704 only)	Configure
Description: The TX driver pre-emphasis value that is used for the secondary device of a 5704 if it is enabled by option 44 below.		

44	SERDES TX Drvr Pre-Emp – ENABLE (5704 only)	Configure
Description: Enable/Disable the setup of the TX driver pre-emphasis defined in options 42 and 43 above.		
Disable (0)	Disable TX driver per-emphasis.	
Enable (1)	Enable TX driver per-emphasis.	

45	Reserved. Option no longer in use.	Reference
----	------------------------------------	-----------

46	Encoded Hot Plug Power 1 & 2 For NIC	Configure
----	--------------------------------------	-----------

<p>Description:                  Encoded Hot plug power values 1 &amp; 2 for a NIC device that is used only if enabled in option 50 below. The following description applies to options 46-49.</p> <p>This option is only valid for 5787 and 575xCx devices. The power values need to be setup by the OEM / manufacture based on actual measurements. Default values, contained in the bootcode / firmware, are loosely based on the in house Broadcom NIC card.</p> <p>15-0 Power Budget data 1</p> <p>    7-0 Base Power in .1 Watt</p> <p>        For example, 1.4 Watt should have value 14.</p> <p>    9-8 PM State</p> <p>        00 D0</p> <p>        01 D1</p> <p>        10 D2</p> <p>        11 D3</p> <p>    12-10 Type</p> <p>        000 PME Aux</p> <p>        001 Auxiliary</p> <p>        010 Idle</p> <p>        011 Sustained</p> <p>        111 Maximum</p> <p>    13-15 Power Rail</p> <p>        000 12V</p> <p>        001 3.3V</p> <p>        010 1.8V</p> <p>        111 Thermal</p> <p>31-16 Power Budget data 2</p> <p>See Power Budget Data 1 for detail, Bit number add 16.</p>	
---	--

47	Encoded Hot Plug Power 3 & 4 For NIC	Configure
Description: Encoded Hot plug power values 3 & 4 for a NIC device that is used only if enabled in option 50 below. Look at option 46 for details.		

48	Encoded Hot Plug Power 5 & 6 For NIC	Configure
Description: Encoded Hot plug power values 5 & 6 for a NIC device that is used only if enabled in option 50 below. Look at option 46 for details.		

49	Encoded Hot Plug Power 7 & 8 For NIC	Configure
Description: Encoded Hot plug power values 7 & 8 for a NIC device that is used only if enabled in option 50 below. Look at option 46 for details.		

50	Hot Plug Power	Configure
Description: Enable/Disable the hot plug power values defined in options 46-49.		
Disable (0)	Disable hot plug power feature.	

Enable (1)	Enable hot plug power feature.
------------	--------------------------------

51	Pri. Port SMB Address (ASF/IPMI)	Configure
Description: SMB address of the primary port on a device.		

52	Sec. Port SMB Address (IPMI)	Configure
Description: SMB address of the secondary port on a device (only valid for dual port devices).		

53	Cable Sense	Configure
Description: Enables a low power mode (IDDQ) of the chip which allows the presence of a cable to be detected by the chip (valid for 5752M/5755M/5787M/5787FM devices). Once Cable Sense is enabled GPIO3 will be an Energy Detect Output Pin and will be high if a cable (with a signal present) is plugged into the RJ45 and low otherwise.		
Disable (0)	Disable cable sense.	
Enable (1)	Enable cable sense.	

54	GPIO 0 Input/Output Config	Configure
Description: Setting GPIO 0 input/output configuration (5714 LOM, 5715 LOM 5755M LOM only)		
Input (0)	Config as input	
Output HI (1)	Config as output HI	
Output LO (2)	Config as output LOW	

55	GPIO 2 Input/Output Config	Configure
Description: Setting GPIO 2 input/output configuration (5714 LOM, 5715 LOM 5755M LOM only)		
Input (0)	Config as input	
Output HI (1)	Config as output HI	
Output LO (2)	Config as output LOW	

56	L1ASPM Debounce En	Configure
Description: Enable L1ASPM Debounce for PCI-E Mobile Devices only		
Disable(0)	Disable	
Enable (1)	Enable	

57	Link Aware Mode	Configure
Description: Enable Link Aware Mode for Taishan, Caesar_II and Soledad Devices only		
Disable(0)	Disable	
Enable (1)	Enable	

58	Link Speed Power	Configure
Description: Enable Link Speed Power for Taishan, Caesar_II and Soledad Devices only		
Disable(0)	Disable	
Enable (1)	Enable	

Example: (For Selfboot firmware).

0:>secfg

1. MAC Address.....: 001018000000
2. Device Id.....: 1693
3. Sub Vendor Id.....: 14E4
4. Sub Device Id.....: 1693
5. Wake on LAN.....: Disabled
6. WoL Speed Limit 10...: Disabled
7. LOM/NIC design.....: NIC
8. Phy. Auto PowerDown.: Disabled
9. Reverse Nway.....: Disabled
  - a. Disable PowerSaving.: Disabled
  - b. LED mode.....: Phy1 Mode
  - c. Custom PCI power....: Yes (consumed D0:100.D3:10 dissipated D0:100.D3:10)
  - d. Custom PCIE power...: Yes (3.3v, Sustained, D0, 1.2 Watt...)
  - e. Custom VPD-R Data...: Yes
  - g. Cable Sense/Low Power Energy Detect .....: Disabled
  - h. MBA CFG Data
    - i. User Defined Data.....: No
    - j. Link Aware Mode.....: Enabled
    - k. Link Speed Power.....: Enabled

x. Save & exit

----- LED Mode (option b)-----

-> b

Selfboot LED Mode selection

0. Mac Mode
1. Phy1 Mode
2. Phy2 Mode
3. Shared Traffic/Lik LED Mode
4. Shasta MAC mode
5. Wireless Combo Mode

----- Custom PCI power (option c)-----

-> c

Selfboot PCI Power Consumption/Dissipate value

1. Use system default
2. Costom define

-> 2

- D0 State Power Comsumed decimal value (0-255) (100)->100  
D3 State Power Comsumed decimal value (0-255) (10)->10  
D0 State Power Dissipated decimal value (0-255) (100)->100

**D3 State Power Dissipated decimal value (0-255) (10)->10**

----- Custom PCIE power (option d)-----

-> d

**Selfboot PCIE Power Budgeting Data**

1. Use system default
2. Costom define

-> 2

0. Data0: 3.3v, Sustained, D0, 1.2 Watt
1. Data1: 3.3v, Maximum, D0, 1.4 Watt
2. Data2: 3.3v, Maximum, D3, 0.6 Watt
3. Data3: 3.3v, PME Aux, D3, 0.4 Watt
4. Data4: Unused
5. Data5: Unused
6. Data6: Unused
7. Data7: Unused

----- Custom VPD Prod.Name (option e)-----

-> e

**Selfboot VPD R**

1. Use system default
2. Costom define

-> 2

1. Part Number : BCM957xx
2. Engineering Change : 106679-15
3. Serial Number : 0123456789
4. Manufacturing ID : 14e4
5. Read only VPD Vendor Specific Data (V0) :
6. Product Name :

**Description of above parameters:**

To enter the 'secfg' menu shown above a user type in 'secfg' for selfboot firmware at the diagnostics command line. A menu is present to the user showing the 'secfg' options shown above. A user can change the 'secfg' parameter by entering the option number.

The tables below provide details for each parameter of the 'secfg' options for selfboot firmware. The top right of each table has the word "Configure" or "Reference". "Configure" signifies that the option is used to configure the operating characteristics of the device or default values of registers in the device. "Reference" signifies that the option is use for display purposes and does not affect the operating characteristics of the device or

default values of registers in the device.

1	MAC Address	Configure
Description: MAC address of the device.		
2	Device ID	Configure
Description: PCI Device ID.		
3	Subsystem Vendor ID	Configure
Description: PCI Subsystem PCI Vendor ID.		
4	Subsystem Device ID	Configure
Description: PCI Subsystem PCI Device ID.		
5	Magic Packet WoL	Configure
Description: A system can be configured to power-on when a Magic Packet is received.		
Enable(1)	The device will assert the pme signal, to power on the system, when a magic packet is received.	
Disable(2)	Magic packets are ignored.	
Note: A driver can setup the WoL behavior of a device and the value programmed into this location is ignored.		
6	Limit WoL Speed to 10	Configure
Description: Limit Wake on LAN (WoL) line speed.		
No(0)	10 or 100 Mbits/s is used for WoL.	
Yes(1)	Only 10 Mbits/s is used for WoL.	
7	Design Type	Configure
Description: Select NIC or LOM based Ethernet controller.		
NIC(0)	Option selected for a plug in network interface card.	
LOM(1)	Option selected for a LAN on motherboard.	
Note: The firmware and the drivers use this to determine the operational characteristics of a device. For example, on a LOM GPIO2 is tied to the SEEPROM write protect pin and on a NIC GPIO 1 and 2 is used to switch between main and auxiliary power for wake on LAN functionality.		
8	Enable PHY Auto Powerdown	Configure
Description: Enable GPHY auto-power down when there is no link present (to conserve power).		
No(0)	The PHY will not auto power down.	
Yes(1)	The PHY will auto power down when there is no link.	
9	Reversed Nway	Configure
Description: Nway Negotiation.		
0	(default) Auto-negotiation is done from 1000->100->10 Mbits/s.	
1	Auto-negotiation is done from 10->100->1000 Mbits/s.	

<p>Note:                  A user could set the chip for option 1 when running a laptop on battery power. The chip would auto-negotiate starting at a lower speed and lower power. The chip would only go to higher line rates and higher power if the lower line rates were unavailable.</p>
--

a	Disable Power Saving capability	Configure
Description: Disable Power Saving capability setup by option 33 above. When disable a device will use Nway negotiation.		
No(0)	Power-saving capability active.	
Yes(1)	Power-saving capability inactive.	

b	Led Mode	Configure												
Description: A device can be configured to use one LED to indicate speed and activity or three LEDs. Use a devices data sheet to verify the LED modes supported by a devices and for exceptions to the LED modes described below.														
Mac Mode (0)	The LEDs are controlled by the value in the Mac LED Control Register.													
Phy Mode1 (1)	Three LEDs are used for 10/100/1000 Mbits/s and each is driven individually by the device.													
Phy Mode2 (2)	One LED is used for 10/100/1000 Mbits/s and is connected as described below.  5700/01 – The Link10 line will indicate link for all speeds. Link100 and Link1000 will encode the line speed as show in the following table: <table border="1" style="margin-left: 40px;"> <thead> <tr> <th>Link 100</th> <th>Link 1000</th> <th>Speed</th> </tr> </thead> <tbody> <tr> <td align="center">0</td> <td align="center">0</td> <td align="center">10 Mbits/s</td> </tr> <tr> <td align="center">1</td> <td align="center">0</td> <td align="center">100 Mbits/s</td> </tr> <tr> <td align="center">0</td> <td align="center">1</td> <td align="center">1000 Mbits/s</td> </tr> </tbody> </table> For all other 57xx devices the three link lines operate in an open drain configuration and can be tied together with a pull up resistor to control a LED.		Link 100	Link 1000	Speed	0	0	10 Mbits/s	1	0	100 Mbits/s	0	1	1000 Mbits/s
Link 100	Link 1000	Speed												
0	0	10 Mbits/s												
1	0	100 Mbits/s												
0	1	1000 Mbits/s												
SharedTraffic (3)	Mode specific to the 575x and 572x chips. The link LED performs a dual role: solid when there is a link and blinks when there is activity.													
Shasta Mac Mode (4)	Mode specific to the 575x and 572x chips. The link LED blinks only when traffic is for Shasta.													
Wireless Combo (5)	Mode specific to the 575x and 572x chips. When link is lost the LEDs are driven by inputs pins that are connected to the LED signals of a wireless link.													

c	Custom PCI Power	Reference
Description: Self boot PCI Power Consumption/Dissipate for D0 and D3 state.		

d	Custom PCIE Power	Reference
Description: Self boot PCIE Power Budgeting Data		

e-1	Part Number	Reference
Description: VPD part number..		

e-2	Engineering Change	Reference
Description: VPD engineering change.		



e-3	Serial Number	Reference
Description: VPD serial number.		

e-4	Manufacturing ID	Reference
Description: VPD manufacturing ID.		

e-5	Read only VPD Vendor Specific Data (V0)	Reference
Description: VPD V0 value. Data field provided for the customer.		

e-6	Product Name	Reference
Description: VPD Product description string.		

g	Cable Sense	Configure
Description: Enables a low power mode (IDDQ) of the chip which allows the presence of a cable to be detected by the chip (valid for 5752M/5755M/5787M/5787FM devices). Once Cable Sense is enabled GPIO3 will be an Energy Detect Output Pin and will be high if a cable (with a signal present) is plugged into the RJ45 and low otherwise.		
Disable (0)		Disable cable sense.
Enable (1)		Enable cable sense.

h	MBA Configuration	Configure
Description: MBA Configuration		
VLAN Enable (1)		Enabled / Disabled
VLAN ID (2)		VLAN ID
MBA Hide Banner (3)		Enabled / Disabled
MBA Hot Key (4)		Ctrl-B / Ctrl-S
MBA Protocol (5)		PXE / RPL / BOOTP / ISCSI
MBA Config Timeout (6)		MBA Config Timeout value
MBA Bootstrap (7)		Auto / BBS / INT18 / INT19
MBA Link Speed (8)		Auto /10HD / 10FD / 100HD / 100FD
Remove MBA Config (9)		0: No 1:Yes

**Example: (For Hardware Selfboot firmware).**

**0:>secfg**

- 1. MAC Address.....: 001018123456**
- 2. Device Id.....: 1713**
- 3. Sub Vendor Id.....: 14E4**
- 4. Sub Device Id.....: 9713**
- 5. WOL.....: Disabled**
- 6. WOL Speed Limit 10...: Disabled**
- 7. WOL Auto.....: Disabled**

- 8. LED mode.....: PHY MODE 2
- 9. VPD Prod.Name and VPD-R Data: Yes
  - A. Super-airplane Mode: Enabled
  - B. MBA Configuration
  - C. NIC Design.....: NIC
  - D. L1 PLL Disable ..... No
  - E. Clkreq.....: Yes
  - F. PCIE Link Polarity Fix Disable: Yes (5906 A2 only)
  - G. L1ASPM Debounce Enable: Disable (5906M only)

x. Save & exit

----- WOL Auto (option 7)-----

-> 7

HW Selfboot Driver WOL selection

- 0. Disable
- 1. Magic Packet WOL Enable

----- LED Mode (option 8)-----

-> 8

HW Selfboot LED Mode selection

- 0. Mac Mode
- 1. Phy1 Mode
- 2. Phy2 Mode
- 3. Shared Traffic/Lik LED Mode

----- VPD Data (option 9)-----

-> 9

HW Selfboot VPD Data

- 1. Part Number : BCM95906
- 2. Engineering Change : 106679-15
- 3. Serial Number : 0123456789
- 4. Manufacturing ID : 14e4
- 5. Read only VPD Vendor Specific Data (V0) :
- 6. Product Name : Broadcom NetLink Fast Ethernet Controller

x. Save & exit

----- MBA Configuration (option B)-----

-> B

HW Selfboot MBA Data

- 1. VLAN Enable : Disabled

- 2. VLAN ID : 0
- 3. MBA HIDE BANNER : Disabled
- 4. MBA HOT KEY : Ctrl-B
- 5. MBA PROTOCOL : PXE
- 6. MBA CONFIG TIME OUT : 0
- 7. MBA BOOTSTRAP : AUTO
- 8. MBA LINK SPEED : AUTO

x. Save & exit

**Description of above parameters:**

To enter the 'secfg' menu shown above a user type in 'secfg' for hardware selfboot firmware at the diagnostics command line. A menu is present to the user showing the 'secfg' options shown above. A user can change the 'secfg' parameter by entering the option number.

The tables below provide details for each parameter of the 'secfg' options for hardware selfboot firmware. The top right of each table has the word "Configure" or "Reference". "Configure" signifies that the option is used to configure the operating characteristics of the device or default values of registers in the device. "Reference" signifies that the option is use for display purposes and does not affect the operating characteristics of the device or default values of registers in the device.

1	MAC Address	Configure
Description: MAC address of the device.		
2	Device ID	Configure
Description: PCI Device ID.		
3	Subsystem Vendor ID	Configure
Description: PCI Subsystem PCI Vendor ID.		
4	Subsystem Device ID	Configure
Description: PCI Subsystem PCI Device ID.		
5	WoL	Configure
Description: A system can be configured to power-on when a Magic Packet is received.		
Enable	The device will assert the pme signal, to power on the system, when a magic packet is received.	
Disable	Magic packets are ignored.	
Note: A driver can setup the WoL behavior of a device and the value programmed into this location is ignored.		
6	Limit WoL Speed to 10	Configure

Description: Limit Wake on LAN (WoL) line speed.	
No	10 or 100 Mbps/s is used for WoL.
Yes	Only 10 Mbps/s is used for WoL.

7	Wol Auto Setting	Configure
Description: Select WOL Auto Setting		
Disable (0)	WOL is Disabled	
Magic Packet WOL (1)	Option selected for Magic Packet WOL.	

8	Led Mode	Configure
Description: A device can be configured to use one LED to indicate speed and activity or three LEDs. Use a device's data sheet to verify the LED modes supported by a device and for exceptions to the LED modes described below.		
Mac Mode (0)	The LEDs are controlled by the value in the Mac LED Control Register.	
Phy Mode1 (1)	Dual Link LED configuration. Two LEDs are used to indicate 10/100 Mbps.	
Phy Mode2 (2)	Link/Speed LED configuration. One LED is used to indicate link status of all 10/100 Mbps.	
SharedTraffic (3)	Shared Traffic/Link LED Mode. The Link LED performs dual role: Solid Green, when Link up. Blink when there is an activity.	

9	VPD Data	Configure
Description: VPD Product Name and Read Data.		
Part Number (1)	VPD Part Number	
Engineering Change (2)	VPD Engineering Change	
Serial Number (3)	VPD Serial Number	
Manufacturing ID (4)	VPD Manufacturing ID	
Read only VPD Data (5)	VPD Read only V0 Data	
Product Name (6)	VPD Product Name	

A	Super Airplane Mode	Configure
Description: Super Airplane mode		
Disable	Disable Super Airplane Mode	
Enable	Enable Super Airplane Mode	

B	MBA Configuration	Configure
Description: MBA Configuration		
VLAN Enable (1)	Enabled / Disabled	
VLAN ID (2)	VLAN ID	
MBA Hide Banner (3)	Enabled / Disabled	
MBA Hot Key (4)	Ctrl-B / Ctrl-S	
MBA Protocol (5)	PXE / RPL / BOOTP / ISCSI	
MBA Config Timeout (6)	MBA Config Timeout value	
MBA Bootstrap (7)	Auto / BBS / INT18 / INT19	
MBA Link Speed (8)	Auto / 10HD / 10FD / 100HD / 100FD	

C	Design Type	Configure
Description: Select NIC or LOM based Ethernet controller.		
NIC(1)	Option selected for a plug in network interface card.	
LOM(0)	Option selected for a LAN on motherboard.	

D	L1 PLL Powerdown Disable	Configure
Description: Disable / Enable L1 PLL Powerdown		
Disable(0)	Disable L1 PLL Powerdown	
Enable(1)	Enable L1 PLL Powerdown	

E	Clock Request	Configure
Description: Disable / Enable Clock Request		
Disable(0)	Disable Clock Request	
Enable(1)	Enable Clock Request	

## 10.8 seprg

**Command:** seprg

**Description:** Program NVRAM and management firmware

**Syntax:** seprg <file\_name>

Input file need to be found in the same location as b57diag.exe.

### Options:

- d Do not perform device check
- f<string> filename
- l<HEX> length in bytes (Default = size of input file)
- m Do not restore original MAC address
- n Do not restore original all data at Configuration Block in OTP
- o<HEX> offset of serial nvram (def=00000000)
- s Do not restore original Serial Number
- a Program ASF/management firmware
- b As APE UPDATE entry (for -a with APE)

-k<DEC>	pad to kilobytes (for -a with APE)
-u	Program UMP firmware
-c	Skip image CRC check
-p	Restore NVRAM Config data between Bootcode and Selfboot fw *

**Example:**

1. Program NVRAM with contents of input file seprg.bin

```
0:> seprg seprg.bin
```

```
0:>seprg asffirmware.bin -a
```

2. Program NVRAM with APE DASH firmware dashfw.rom and pad up to 300kilo bytes.

```
0:>seprg -a -k300 dashfw.rom
```

3. Program NVRAM with APE DASH firmware dashfw.rom as APE UPDATE image and pad up to 300kilo bytes. Padding and backup image is needed to enable out-of-band DASH firmware update.

```
0:>seprg -a -b -k300 dashfw.rom
```

\*-p option will restore NVRAM configuration data between Bootcode and selfboot firmware. The configuration data that will be restored are listed below.

- a. Wol\_enable
- b. Wol\_Limit\_10
- c. Driver\_wol\_enable
- d. Lom\_design
- e. Phy\_auto\_power\_down
- f. Reverse\_nway
- g. Disable\_power\_saving
- h. Phy\_led\_mode
- i. Shasta\_ext\_led\_mode
- j. Cable\_sense

## 10.9 upgfrm

**Command:** upgfrm

**Description:** Upgrade ISCSI, PXE, Boot Code and Selfboot Firmware from a File. This command reads code from a file and program into iscsi, pxe or boot area. Both parameter, the programming target '-p', 'b' or '-i' and filename, must be specified. In case of ISCSI firmware, ISCSI Boot Firmware will be updated by default. ISCSI CFG Block will only be updated if there is no ISCSI CFG Block present in NVRAM. ISCSI CFG Program will NOT be programmed by this command. In case of Selfboot Firmware, this command will support Format 0 to Format 1 and Format1 to Format1 update. Format0 to Format0 and Format1 to Format0 update are NOT supported by this command. User will need to use "seprg" command. For legacy bootcode, only VPD data block and Phase 1 and Phase 2 bootcode will be updated by default. If -v is entered, only Phase 1 and Phase 2 bootcode will be updated. For Selfboot firmware and Hardware Selfboot firmware, only patches will be updated.

**Syntax:** upgfrm <p | b | I | v> filename

```
upgfrm [-f<filename>] [-b | -p | -i] | [-d] | [-v]
```

### Options:

-b	Upgrade boot code/Selfboot firmware
-d	Do not perform device check
-f<string>	Input file
-p	Upgrade PXE code
-i	Upgrade ISCSI code
-r	Do not perform chip rev. check for Selfboot
-n	Set PFN when loading LOM PXE code
-v	Preserve VPD block data in legacy bootcode upgrade

## 10.10 sever

**Command:** sever

**Description:** Display Serial NVRAM Version

**Syntax:** sever

## 10.11 sechksum

**Command:** sechksum

**Description:** Check/Update Serial NVRAM checksum

**Syntax:** sechksum

**Options:**

-v<DEC>      verbose level (0,1) (def=1)

## 10.12 sedump

**Command:** sedump

**Description:** Dump NVRAM content to a file. The value stored at location 0xC in the NVRAM is the starting address of the code that the chip loads and executes. The code starting address is 0x200 for NVRAM pages aligned on 256 byte boundaries and 0x2F8 for the Atmel device that has a NVRAM page size of 264. If “sedump” is used to generate an image file for a NVRAM/EEPROM burner then “-a” should be used, for an Atmel device, if one wants to preserve the 0x2F8 code starting address that is automatically translated to 0x200 by software without the “-a” option. More details can be found in application note 570X-AN700-R.

**Syntax:** sedump -f<string> [Len]

**Options:**

-a<string>      no Atmel Flash address translation  
-f<string>      string is the filename  
-l<DEC>      length in decimal, use 0 for entire NVRAM image (def=8192)  
-s              skip crc checking for legacy bootcode

## 10.13 asfcfg

**Command:** asfcfg

**Description:** Configure ASF in NVRAM

**Syntax:** asfcfg [filename]

If [filename] is provided the file is used to configure the ASF configuration information in NVRAM. If asfcfg is entered by it self a menu is present the user and the options of the items listed in the menu are described below.

0	Save to file
---	--------------



Description: Use this menu option save the network adapter's ASF configuration table to a binary file.	
---	--

1	ASF Settings	
Description: Provides the sub menu with the options described below.		
	0	Return to previous menu.
	1	Enable/Disable Remote Management Control Protocol (RMCP)
	2	Enable/Disable the transmission of Platform Event Trap (PET) messages.
	3	Enable/Disable transmission of periodic system heartbeat messages.
	4	Interval in seconds between transmitted heartbeat messages.
	5	PET messages (except the system heartbeat) are retransmitted three times. This entry allows the user to specify, in seconds, the interval of time between these transmissions.
	6	Client IP address used for communication.
	7	Management console IP address to which communication is directed.
	8	Gateway IP address used when client and management console are on different subnets.
	9	Subnet mask used to determine client and management console subnets.
	10	Configures link speed for ASF operation when the OS is not present.
	11	Unique value used to by ASF. Usually setup by an OS based configuration utility, based on the SMBIOS system information structure.
	12	Configure the polling interval, in seconds, which legacy SMBus devices are monitored (such as the chassis intrusion sensor).
	13	Specifies, in seconds, the amount of time to wait before polling the first legacy SMBus device.
	14	Send 'ASD Ready' SMBus Msg
	15	Enable/Disable the system to wake up from a low power states when the system receives ASF or RMCP traffic.
	16	Feature is not currently supported.
	17	SNMP community name.

2	ASF Alert Info
3	ASF Alert Data for Legacy Devices
4	ASF Remote Control Data
5	ASF Capabilities Supported
6	ASF SMBus Addresses

Description:

These menu options may be used to override values in the network adapter's ASF Configuration Table that correlate to the following records in the system's "ASF! Description Table":

ASF\_INFO  
 ASF\_ALRT  
 ASF\_RCTL  
 ASF\_RMCP  
 ASF\_ADDR

The values in these records are normally copied from the system's "ASF! Description Table" (using the ACPI System Description Table architecture) into the ASF Configuration Table in the network adapter's non-volatile memory by the Broadcom ASF Configuration Utility (ASFConfig). See the Alert Standard Format (ASF) Specification v2.0 for detailed descriptions of these records.

7	Reserved
Description: Reserved	

8	SMBus Init Data
Description: Not to be used by customers. For Broadcom internal use only.	

9	Save and Exit
Description: Saves the modified ASF Configuration Table to the network adapter's non-volatile memory and exists.	

10	Exit without Saving
Description: Exits without saving any changes to the ASF Configuration Table.	

## 10.14 mancfg

**Command:** mancfg

**Description:** Configure management firmware in NVRAM. For device without APE, this command is same as asfcfg command.

**Syntax:** mancfg [-d | -a | -s | -i filename | -e filename]

**Options:**

- d            Verbose description of content of APE\_CFG
- a            Import ASF\_SDT record from host PC memory
- s            Import SMBIOS record from host PC memory
- i            Import APE\_CFG from file instead of NVRAM

-e Export APE\_CFG to specified file

Note: mancfg command can be run in command mode or interactive mode. When option is specified, command mode is used. For details of interactive mode, please refer to bmcfg, and mancfg documentation. This command is work in progress, will be updated later.

## 10.15 asfmbox

**Command:** asfmbox

**Description:** Create ASF Off-Line Mailboxes in NVRAM.

**Syntax:** asfmbox [-c | -v | -d | -t]

### Options:

-c Create specified number of ASF Mailboxes  
-v View status of all ASF Mailboxes  
-d Dump contents of specified ASF Mailbox  
-t Test ASF Mailbox NVRAM block

## 10.16 flshmode

**Command:** flshmode

**Description:** Configure ASF in NVRAM

**Syntax:** flshmode [auto | passthru | legacy | new]

## 10.17 flshread

**Command:** flshread

**Description:** Same as seread

**Syntax:** flshread start[-end | len]

## 10.18 chkpxe

**Command:** chkpxe

**Description:** Check PXE code image

**Syntax:** chkpxe <filename>

### 10.19 dir

**Command:** dir

**Description:** display file directory in NVRAM

**Syntax:** dir

### 10.20 erase

**Command:** erase

**Description:** erase file from directory in NVRAM

**Syntax:** erase <entry> | all

**Options:**

-y do not ask for conformation

### 10.21 pxeprg

**Command:** pxeprg

**Description:** Program PXE firmware into NVRAM. This command reads PXE code from a file and program into NVRAM

**Syntax:** pxeprg <filename>

### 10.22 flshdev

**Command:** flshdev

**Description:** Select flash device to access

**Syntax:** flshdev seeprom | atmelflash | sstflash

### 10.23 bitbang

**Command:** bitbang

**Description:** Generate bitbang pattern to seeprom bus. Enter hex numbers to be written to seeprom bus. For serial eeprom, use 's' for start, 'p' for stop condition, 'x' with '1..9' specifies how many residual bits to send. For flash, use 's' for chipSelect, 'p' for cancel chipSelect 'r' with a number specifies how many bytes to read.

**Syntax:** bitbang <data>

## 10.24 seclock

**Command:** seclock

**Description:** set NVRAM config1 content value after reset.

**Syntax:** seclock <32bitNewValue>

## 10.25 setorture

**Command:** setorture

**Description:** NVRAM reset torture test

**Syntax:** setorture

**Options:**

-c	Continue on Error
-i<DEC>	Initial Delay in ms (def=0)
-m<DEC>	Maximum Delay in ms (def=3000)
-n<DEC>	iteration (def=1)
-p	Power reset
-r	Random delay
-s<DEC>	Delay incremental time (us) (def=1)

## 10.26 seinit

**Command:** seinit

**Description:** Initialize NVRAM block.

**Syntax:** seinit

**Options:**

-i<DEC>	do not restore NVRAM clock default value (def=0)
---------	--

## 10.27 searb

**Command:** searb

**Description:** set/report current NVRAM arbitration bit. Valid arbitraion number is 0..3.

**Syntax:** searb [n]

## 10.28 seprotect

**Command:** seprotect

**Description:** set gpio pin for NVRAM write protect. Use 'none' to disable write protect feature.

**Syntax:** seprotect [0|1|2|none]

## 10.29 selclock

**Command:** selclock

**Description:** set Legacy EEPROM clock value (bit 16-24 of EEPROM addr reg)

**Syntax:** selclock <8bitNewValue>

## 10.30 semap

**Command:** semap

**Description:** Display NVRAM usage

**Syntax:** semap

## 10.31 setwol

**Command:** setwol

**Description:** Enable/Disable WOL

**Syntax:** setwol [e/d]

**Options:**

-d            Disable WOL

-e            Enable WOL

## 10.32 setpxe

**Command:** setpxe

**Description:** Enable/Disable PXE

**Syntax:** setpxe

**Options:**

- d            Disable PXE
- e            Enable PXE
- s<DEC>     Specify PXE Speed (def=0), 0. auto, 1. 10HD, 2. 10FD, 3. 100HD, 4. 100FD

**10.33 setasf**

**Command:** setasf

**Description:** Enable/Disable ASF

**Syntax:** setasf

**Options:**

- d            Disable ASF
- e            Enable ASF

**10.34 secomp**

**Command:** secomp

**Description:** compare eeprom content against the file

**Syntax:** secomp

**Options:**

- c            continue on error
- f<string>   filename
- l<HEX>     length (def=00000000)
- o<HEX>     offset (def=00000000)

**10.35 dreset**

**Command:** dreset

**Description:** double reset test for EEPROM debug

**Syntax:** dreset

### 10.36 userblock

**Command:** userblock

**Description:** create a userblock in NVRAM

**Syntax:** userblock

### 10.37 setmba

**Command:** setmba

**Description:** Enable Multiple Boot Agent

**Syntax:** setmba

**Options:**

- d            Disable MBA
- e<DEC>    Enable MBA Protocol (def=0), 0. PXE, 1. RPL, 2. BOOTP
- s<DEC>    Specify MBA Speed (def=0), 0. auto, 1. 10HD, 2. 10FD, 3. 100HD, 4. 100FD

### 10.38 segencrc

**Command:** segencrc

**Description:** Append a CRC to the specified block

**Syntax:** segencrc <start-end> | <start length>

### 10.39 secfg1

**Command:** secfg1

**Description:** Configure NVRAM Group 1

**Syntax:** secfg1

**Options:**

- m            Mac address xx:xx:xx:xx:xx:xx
- r            Power Dissipated D3:D2:D1:D0
- c            Power Consumed D3:D2:D1:D0
- v            Vendor ID



-d	Vendor Device ID
-s	SubSystem Vendor ID
-i	SubSystem Device ID
-p	PXE 1:Enable 2:Disable
-x	PXE SP 0:Auto 1:10HD 2:10FD 3:100HD 4:100FD 6:1000FD (TBI)
-w	Magic Packet WOL 1:Enable 2:Disable

#### 10.40 secfg2

**Command:** secfg2

**Description:** Configure NVRAM Group 2

**Syntax:** secfg2

**Options:**

-p	Product Name
-n	Part Number
-e	Engineering Change
-s	Serial Number
-m	Manufacturing ID
-a	Asset Tag
-r	Part Rev
-v	Voltage
-f	Force PCI Mode 1:Enable 2:Disable

#### 10.41 secfg3

**Command:** secfg3

**Description:** Configure NVRAM Group 3

**Syntax:** secfg3

**Options:**

- l Led Mode 0:Mac 1:Phy1 2:Phy2 3:S Traffic 4:Shasta Mac 5:Wireless Combo
- r Max PCI Retry 0-7, 8:Auto
- a ASF 1:Enable 2:Disable
- d Dual Mac 0:Norm 1:MacB 2:MacA 3:XBAR 4:Swap 7:SwapXBAE
- m MBA Boot Protocol 0:PXE 1:RPL 2:BOOTP
- b MBA Bootstrap Type 0:Auto 1:BBS 2:Int18 3:Int19
- t MBA Delay Time 0-15
- e Exp ROM 0:64K 1:128K 2:256K 3:512K 4:1M 5:2M 6:4M 7:8M 8:16M
- n Design Type 0:NIC 1:LOM
- m MBA Boot Protocol 0:PXE 1:RPL 2:BOOTP 3:ISCSI

**10.42 secfg4**

**Command:** secfg4

**Description:** Configure NVRAM Group 4

**Syntax:** secfg4

**Options:**

- v Read VPD Vendor Data V0
- w Read/Write VPD Vendor Data V1
- n Reversed Nway 0:No 1:Yes
- s Limit WOL Speed to 10 0:No 1:Yes
- f Fiber WOL Capable 0:No 1:Yes
- c Clock-Run 0:Disable 1:Enable
- p Enable Phy Auto Power Down 0:No 1:Yes
- d Disable Power Saving 0:No 1:Yes
- h Hide MBA Setup Prompt 0:Disable 1:Enable

- k MBA Setup Hot Key 0:Ctrl-S 1:Ctrl-B
- i Capacitive Coupling 0:Disable 1:Enable

### 10.43 secfg5

**Command:** secfg5

**Description:** Configure NVRAM Group 5

**Syntax:** secfg5

**Options:**

- i Capacitive Coupling 0:Dis 1:En
- e SERDES Tx Drvr Pre-Emp - Primary
- t SERDES Tx Drvr Pre-Emp - Secondary
- z SERDES Tx Drvr Pre-Emp - ENABLE
- u Encoded Hot Plug Power Value 1 & 2 For NIC
- d Encoded Hot Plug Power Value 3 & 4 For NIC
- r Encoded Hot Plug Power Value 5 & 6 For NIC
- q Encoded Hot Plug Power Value 7 & 8 For NIC
- h Hot Plug Power For NIC - Enable
- a ASF/IPMI SMB Address for Pri. Port
- b IPMI SMB Address for Sec. Port
- c Cable Sense - Enable
- x GPIO 0 Config 0:Input 1:OutputH 2: OutputL (5714LOM and 5715 LOM only)
- y GPIO 2 Config 0:Input 1:OutputH 2: OutputL (5714LOM and 5715 LOM only)
- f L1ASPM\_Debounce\_En 0:Dis 1:En (PCIE Mobile devices only)

### 10.44 setipmi

**Command:** setipmi

**Description:** Enable/Disable IPMI

**Syntax:** setipmi

**Options:**

-d            Disable IPMI  
-e            Enable IPMI

## 10.45 loadfw

**Command:** loadfw

**Description:** Load Firmware to Tx/RX CPUs

**Syntax:** loadfw

**Options:**

-b<HEX>     set breakpoint (0=off) (def=00000000)  
-d<HEX>     data pattern to be used for memory init. (def=00000000)  
-f<string>   filename (for bcmediag compatibility)  
-I            don't initialize memory before loading, def=yes  
-m            don't turn on CPU Trace (def=on)  
-s            don't start cpu  
-t            load to tx cpu, def=rxpcu  
-v            disable verbose

## 10.46 cpudtt

**Command:** cpudtt

**Description:** Read and display TX CPU trace (not valid for 5705)

**Syntax:** cpudtt <begin\_addr>[- end\_addr | num\_bytes ]

Address range: 0x00 – 0x80

**Example:**

1. Read and display TX CPU trace from location 0x00 to 0x04.

```
0:> cpudtt 0-5  
000 t0000002f c0000000 00000000 00000000 00000000 00000000
```

```
001 t00000000 00000000 00000000 00000000 00000000 00000000
002 t00000000 00000000 00000000 00000000 00000000 00000000
003 t00000000 00000000 00000000 00000000 00000000 00000000
004 t00000000 00000000 00000000 00000000 00000000 00000000
```

2. Read and display 4 locations of TX CPU trace from start from location 0x00.

```
0:> cpudtt 0 5
000 MainCpuB t0000002f c0000000 00000000 00000000 00000000 00000000
001 t00000000 00000000 00000000 00000000 00000000 00000000
002 t00000000 00000000 00000000 00000000 00000000 00000000
003 t00000000 00000000 00000000 00000000 00000000 00000000
```

## 10.47 cpudrt

**Command:** cpudrt

**Description:** Read and display RX CPU trace (not valid for 5705)

**Syntax:** cpudrt <begin\_addr>[- end\_addr | num\_bytes]

Address range: 0x00 – 0x80

**Example:**

1. Read and display RX CPU trace from location 0x00 to 0x04.

```
0:> cpudrt 0-5
000 MainCpuA t00000030 164414e4 e1000004 00000000 164414e4 00000000
001 *BUpCpuA t00000032 00000000 08000034 00440400 00001c40 00000000
002 *BUpCpuA t00000001 00000001 08000034 00440000 00000000 00000000
003 t00000000 00000000 00000000 00000000 00000000 00000000
004 t00000000 00000000 00000000 00000000 00000000 00000000
```

2. Read and display 4 locations of RX CPU trace from start from location 0x00.

```
0:> cpudrt 0 5
000 t00000030 164414e4 e1000004 00000000 164414e4 00000000
001 t00000032 00000000 08000034 00440400 00001c40 00000000
002 t00000001 00000001 08000034 00440000 00000000 00000000
003 t00000000 00000000 00000000 00000000 00000000 00000000
```

## 10.48 cputrace

**Command:** cputrace

**Description:** toggles cpu trace mode

**Syntax:** cputrace [1|0]

## 10.49 haltcpu

**Command:** haltcpu

**Description:** Halt CPU

**Syntax:** haltcpu

## 10.50 loadbootcode

**Command:** loadbootcode

**Description:** execute bootcode from file instead of NVRAM

**Syntax:** loadbootcode <filename>

**Options:**

-d<HEX>	data pattern to be used for memory init. (def=00000000)
-f<string>	filename
-i	don't initialize memory before loading, def=yes
-m	don't turn on CPU Trace (def=on)
-s	don't start cpu
-t	load to tx cpu, def=rxcpu
-v	disable verbose

## 10.51 disasm

**Command:** disasm

**Description:** Disassemble MIP instructions

**Syntax:** disasm [address [line]]

## 10.52 step

**Command:** step

**Description:** Step MIP instructions

**Syntax:** step

## 10.53 go

**Command:** go

**Description:** start CPU

**Syntax:** go

### 10.54 showgpr

**Command:** showgpr

**Description:** toggles showing cpu gpr mode

**Syntax:** showgpr 1|0

### 10.55 pc

**Command:** pc

**Description:** set current CPU program counter

**Syntax:** pc

### 10.56 breakpoint

**Command:** breakpoint

**Description:** set current CPU breakpoint

**Syntax:** breakpoint

### 10.57 select

**Command:** select

**Description:** select current CPU

**Syntax:** select <r|t>

### 10.58 u

**Command:** u

**Description:** Alias of disasm command

**Syntax:** u [address [line]]

### 10.59 trap

**Command:** trap

**Description:** trap cpu memory

**Syntax:** trap <low> <high> | off

### 10.60 **cpuinfo**

**Command:** cpuinfo

**Description:** display cpu information

**Syntax:** cpuinfo [r|t]

### 10.61 **cpugprstatus**

**Command:** cpugprstatus

**Description:** Display internal CPU GPR test status

**Syntax:** cpugprstatus

### 10.62 **cpugprrestart**

**Command:** cpugprrestart

**Description:** Restart internal CPU GPR test

**Syntax:** cpugprrestart

### 10.63 **cpufetch**

**Command:** cpufetch

**Description:** Test CPU Instruction fetch logic with 1.1V and 1.3 V setting

**Syntax:** cpufetch

### 10.64 **romcodetest**

**Command:** romcodetest

**Description:** execute ROM Code Test

**Syntax:** romcodetest

**Options:**

-f<string> filename

-i<HEX> iteration (def=00000001)

-v verbose



## 10.65 apeinfo

**Command:** apeinfo

**Description:** Display APE CPU related info. When no option is given, displays memory usage for each Nucleus tasks and memory pool. Only one option can be specified at a time, information related to specified option will be displayed.

**Syntax:** apeinfo

**Options:**

-c	Show APE CPMU related info
-s1/s2	Show APE SMBus block 1 or 2 info
-r	Show APE control registers
-f	Show receive management filters
-m	Show APE mutex and shared memory info

## 10.66 apectl

**Command:** apectl

**Description:** The command is used to control the Application Processor Engine (APE) and its firmware (in applicable devices, e.g. BCM5761). Used for debugging. One event option has to be specified.

**Syntax:** apectl

**Options:**

-n	send APE NOP event (are you alive?)
-r	send APE reset event (graceful)
-f<HEX>	turn rx mgmt filter all(-f2)/on(-f1)/off(-f0)
-w<HEX>	set WFI mode sleep(-w1),off(-w0),deepsleep(-w2)
-t	set APE date/time
-e<HEX>	send APE custom event ddeess (dd=data, ee=event id, ss=source)
-R	reset APE block (ungraceful)
-C	reset APE CPU (ungraceful)

-h                   halt APE (graceful)  
-u                   un-halt APE  
-H                   halt (hold in reset) APE CPU (ungraceful)  
-K                   kick start APE (A0 workaround)

Log Control:

-a                   operate on Audit Log instead of Event Log  
-l<string>         add log entry (text only)  
-c                   clear log

Note: "apectl -r" can be used to restart APE manually when firmware or config changes.

### 10.67 apeping

**Command:** apeping

**Description:** Send APE event to ping host from APE. The host can be IPv4, IPv6 address or host name for a DNS lookup.

**Syntax:** apeping <host>

### 10.68 aperead

**Command:** aperead

**Description:** Read APE local view data thru APE.

**Syntax:** apectl start[-end | len]

### 10.69 apeotpkey

**Command:** apeotpkey

**Description:** S

**Syntax:** apeotpkey

### 10.70 apelog

**Command:** apelog

**Description:** The command is used to create, query, and control the APE Logs in NVRAM(in applicable devices, e.g. BCM5761)

**Option:**

- a operate on Audit Log instead of Event Log
- c<DEC> create NVRAM storage for log (of specified length) (def=3232)
- q query log status
- v view log records (in hex format)
- i view log records (in decoded format)
- d dump log storage (in Byte/ASCII format)
- l<string> add log entry (text only)
- C clear log

**10.71 dmaw**

**Command:** dmaw

**Description:** DMA from NIC to Host Memory

**Syntax:** dmaw

**Options:**

- 3 Force to use 32-bit bus
- 4 Allocate 4k-aligned buffers
- a<HEX> NIC address to DMA data from (def=00000000)
- b byte swap
- c Continuously dma data
- d<HEX> delay poll dma done polling (def=00000000)
- f<string> filename
- h Use high priority DMA Write
- l<HEX> Length of DATA in bytes to DMA (def=00000100)
- n<DEC> iteration
- o<HEX> Buffer offset (def=00000000)

-p<HEX> Pattern of Data. 0 - byte increment ; 1- byte decrement  
 2 - FF's ; 3 - 00's ; 4- AA 55 ... ; 5 - 55 AA ...  
 6 - FFFFFFFF 00000000 FFFFFFFF 00000000  
 7 - FFFFFFFFFFFFFFFF 0000000000000000  
 FFFFFFFFFFFFFFFF  
 8 - FFFFFFFFFFFFFFFF0000000000000000...  
 9 - 00000000000000000000000000000000 FFFFFFFF...  
 a - Word Increment ; b - Dword Increment  
 c - Word Decrement ; d - Dword Decrement  
 e - ffffffff00000000 00000000ffffffbf  
 f - 00000000ffffffbf fffffffbf00000000  
 10 - 64-bit-pattern 0000000000000000 64-bit-pattern ...  
 11 - 64-bit-pattern fffffffbf00000000 64-bit-pattern ... (def=00000000)

-q<HEX> low 32-bit of 64-bit pattern (def=FFFFFFFF)

-v<DEC> Verbose (1..2) (def=2)

-w word swap

-x<HEX> high 32-bit of 64-bit pattern (def=FFFFFFFF)

-K<HEX> DMA write to absolute address and hang the system  
 (def=00000000)

**Example:**

1. Setup DMA NIC Memory to HOST memory. Using low priority DMA Read and disable byte swap and enable detail display.

```
1:> dmaw -a=0 -l=10
Device 1
Host Address : 0x0068bb38
NIC Address  : 0x00000000
Length       : 0x0010
Priority      : Low
Byte Swap    : No
Word Swap    : No

Dev 1: DMA SRAM 00000000 to Host 0068BB38
```

**10.72 dmar**

**Command:** dmar

**Description:** DMA from Host to NIC Memory

**Syntax:** dmar

**Options:**

-w	word swap
-3	Force to use 32-bit
-4	Allocate 4k-aligned buffers
-a<HEX>	NIC address to DMA data to (def=00000000)
-b	Byte Swap
-c	Continuously dma
-d<HEX>	delay poll dma done polling (def=00000000)
-f<string>	File name of file that contains <length,patterns ....>
-h	Use high priority DMA Read
-l<HEX>	Length of DATA to do DMA (def=00000100)
-n<DEC>	iteration
-o<HEX>	Buffer offset (def=00000000)
-p<HEX>	Pattern of Data 0 - byte increment ; 1- byte decrement 2 - FF's ; 3 - 00's ; 4- AA 55 ... ; 5 - 55 AA ... 6 - FFFFFFFF 00000000 FFFFFFFF 00000000 7 - FFFFFFFF FFFFFFFF 00000000 00000000 FFFFFFFF FFFFFFFF 8 - FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00000000 00000000... 9 - 00000000 00000000 00000000 00000000 FFFFFFFF FFFFFFFF... a - Word Increment ; b - Dword Increment c - Word Decrement ; d - Dword Decrement (def=00000000)
-q<HEX>	low 32-bit of 64-bit pattern (def=FFFFFFF)
-v<DEC>	Verbose (1..2) (def=2)
-w	Word Swap
-x<HEX>	high 32-bit of 64-bit pattern (def=FFFFFFF)
-K<HEX>	DMA read from absolute address (def=00000000)

**Example:**

1. Sup DMA host memory to NIC memory. Using low priority DMA Read and disable byte swap.

```
1:> dmar -a=0 -l=100
Device 1
Host Address : 0x0068bb38
NIC Address  : 0x00000000
Length       : 0x0100
Priority      : Low
Byte Swap    : No
Word Swap    : No
```

```
Dev 1: DMA Host 0068BB38 to SRAM 00000000
```

### 10.73 dma\_h

**Command:** dma\_h

**Description:** Dump DMA Entries

**Syntax:** dma\_h <start> <end>

### 10.74 dma\_d

**Command:** dma\_d

**Description:** Dump DMA Entries with Decode

**Syntax:** dma\_d <start> <end>

### 10.75 dma\_alloc

**Command:** dma\_alloc

**Description:** Allocate number of DMAD (non-BCM5705 family only)

**Syntax:** dma\_alloc <dma count>

### 10.76 maclpk

**Command:** maclpk

**Description:** Configure MAC loopback, 0 to disable, otherwise enable MAC loopback

**Syntax:** maclpk <n>

0 to disable. Otherwise enable

**Example:**

1. Driver must be loaded before configure.

```
0:> loaddrv
```

2. Enable MAC loop back.

```
0:> mcaclpb 1
Enabling MAC loopback ... OK
```

2. Disable MAC loop back.

```
0:> maclpb 0
Disabling MAC loopback ... OK
```

## 10.77 blast

**Command:** blast

**Description:** Blast Packets in Poll Mode and display statistics. Load MAC driver before running the test.

**Syntax:** blast

### Options:

- a<DEC> IP total length (def=0)
- c<DEC> Number of Tx buffer (def=100)
- d<DEC> Interpacket GAP in microseconds (def=0)
- e<DEC> Upper Limit of Tx buffer in incremental packet size (def=1514)
- f<string> Sniffer file containing contents of Tx packets
- g<DEC> Rx Threshold (def=5)
- h Enable Host Loopback
- I Increment length
- j Regenerate CRC-32 in host loopback mode
- k Applies CRC-32 check on Rx path
- l<DEC> Length of Tx packet (def=60)
- m Generate TPROT packets
- n<DEC> Number of packets to be transmitted (def=0)
- o<DEC> Number of Rx Rings (def=1)
- p Send protocol packets configured with txcfg command.
- q Use software CRC-32 on Tx Path
- r Enable Rx

- s Stop on Failure
- t Enable Tx
- w<DEC> Low watermark max RxFrame value (0-65535)
- x Check length of received packet
- z<DEC> Tx Threshold (def=5)
- P Pause
- R Enable RS232 statistic update
- S TCP segment test
- T<DEC> Packet Type, 0:None, 1:Eth2, 2:802.3, 4:SNAP
- W Check DMA\_Write error status

**Example:**

1. Load MAC driver and enable transmission.

```
0:packet> loaddrv
Reinitializing PCI Configuration Space
Bus Number      : 1
Device/Funtion  : 11/0
Base Address    : 0xfb010000
IRQ             : 9
Bringing up MAC driver ... OK
PHY calculated ID: 60008162
BCM5702/03 Internal Phy Rev#2
Configuring BCM54xx ... Done
Determining Link Speed ... 1000Base-T Full Duplex
0:packet> blast -t
PageUP/PageDN to scroll. Ins/Del toggles refresh. ESC to exit
```

	Total	Rate
	=====	=====
Txed Packets (Ring#0) :	1007609	507523
Txed Packets (Ring#1) :	0	0
Txed Packets (Ring#2) :	0	0
Txed Packets (Ring#3) :	0	0
Tx Packets Enqed (Ring#0) :	0	0
Tx Packets Enqed (Ring#1) :	0	0
Tx Packets Enqed (Ring#2) :	0	0
Tx Packets Enqed (Ring#3) :	0	0
Rxed Packets (Ring#00) :	0	0
Rxed Packets (Ring#01) :	0	0
Rxed Packets (Ring#02) :	0	0
Rxed Packets (Ring#03) :	0	0
Rxed Packets (Ring#04) :	0	0
Rxed Packets (Ring#05) :	0	0
Rxed Packets (Ring#06) :	0	0
Rxed Packets (Ring#07) :	0	0
Rxed Packets (Ring#08) :	0	0
Rxed Packets (Ring#09) :	0	0

PageUP/PageDN to scroll. Ins/Del toggles refresh. ESC to exit



## 10.78 nicstats

**Command:** nicstats

**Description:** Display NIC Statistics

**Syntax:** nicstats <-c>

-c : Clear Statistics

**Example:** Load driver if driver is not loaded.

```
0:> loaddrv
Reinitializing PCI Configuration Space
Bus Number      : 1
Device/Funtion  : 11/0
Base Address    : 0xfb010000
IRQ             : 9
Bringing up MAC driver ... OK
PHY calculated ID: 60008162
BCM5702/03 Internal Phy Rev#2
Configuring BCM54xx ... Done
Determining Link Speed ... 1000Base-T Full Duplex
0:> nicstats
```

	Total	Rate
	=====	=====
Txed Packets (Ring#0) :	0	0
Txed Packets (Ring#1) :	0	0
Txed Packets (Ring#2) :	0	0
Txed Packets (Ring#3) :	0	0
Tx Packets Enqed (Ring#0) :	0	0
Tx Packets Enqed (Ring#1) :	0	0
Tx Packets Enqed (Ring#2) :	0	0
Tx Packets Enqed (Ring#3) :	0	0
Rxed Packets (Ring00) :	0	0
Rxed Packets (Ring01) :	0	0
Rxed Packets (Ring02) :	0	0
Rxed Packets (Ring03) :	0	0
Rxed Packets (Ring04) :	0	0
Rxed Packets (Ring05) :	0	0
Rxed Packets (Ring06) :	0	0
Rxed Packets (Ring07) :	0	0
Rxed Packets (Ring08) :	0	0
Rxed Packets (Ring09) :	0	0

PageUP/PageDN to scroll. Ins/Del toggles refresh. ESC to exit

Total	Rate	
	=====	=====
Rxed Packets (Ring#10) :	0	0
Rxed Packets (Ring#11) :	0	0
Rxed Packets (Ring#12) :	0	0
Rxed Packets (Ring#13) :	0	0
Rxed Packets (Ring#14) :	0	0
Rxed Packets (Ring#15) :	0	0
Rxed CRC-32 Errors :	0	0
Out of Memory :	0	0
Too Many Frag Pkt :	0	0

PageUP/PageDN to scroll. Ins/Del toggles refresh. ESC to exit

CHIP Statistics

```

=====
ifHCInOctets      :      0  etherStatsFragments      :      0
ifHCInUcastPkts  :      0  ifHCInMulticastPkts    :      0
ifHCInBroadcastPkts :      0  d3StatsFCSErrors      :      0
d3StatsAlignmentErrors :      0  xonPauseFramesReceived :      0
xoffPauseFramesReceived:      0  macControlFramesReceived:      0
xoffStateEntered  :      0  dot3StatsFramesTooLong :      0
etherStatsJabbers :      0  etherStatsUndersizePkts :      0
inRangeLengthError :      0  outRangeLengthError    :      0
etherStatsPkts64Octets :      0  etherStatsPkts65-127   :      0
etherStatsPkts128-255 :      0  etherStatsPkts256-511   :      0
etherStatsPkts512-1023 :      0  etherStatsPkts1024-1522 :      0
etherStatsPkts1523-2047:      0  etherStatsPkts2048-4095 :      0
etherStatsPkts4096-8191:      0  etherStatsPkts8192-9022 :      0
ifHCOutOctets    :      0  etherStatsCollisions    :      0
outXonSent       :      0  outXoffSent             :      0
flowControlDone  :      0  d3StatsIntlMacTxErrors  :      0
d3StatsSingleColFrames :      0  d3StatsMultipleColFrames:      0
dt3StatsDeferredTx :      0  d3StatsExcessiveCol    :      0
=====
    
```

PageUP/PageDN to scroll. Ins/Del toggles refresh. ESC to exit

CHIP Statistics

```

=====
d3StatsLateCol    :      0  d3Collided2Times      :      0
d3Collided3Times :      0  d3Collided4Times      :      0
d3Collided5Times :      0  d3Collided6Times      :      0
d3Collided7Times :      0  d3Collided8Times      :      0
d3Collided9Times :      0  d3Collided10Times     :      0
d3Collided11Times :      0  d3Collided12Times     :      0
d3Collided13Times :      0  d3Collided14Times     :      0
d3Collided15Times :      0  ifHCOutUcastPkts     :      0
d3StatsCarSenseErrors :      0  ifOutDiscards         :      0
COSIfHCInPkts[00] :      0  COSIfHCInPkts[01]    :      0
COSIfHCInPkts[02] :      0  COSIfHCInPkts[03]    :      0
COSIfHCInPkts[04] :      0  COSIfHCInPkts[05]    :      0
COSIfHCInPkts[06] :      0  COSIfHCInPkts[07]    :      0
COSIfHCInPkts[08] :      0  COSIfHCInPkts[09]    :      0
COSIfHCInPkts[10] :      0  COSIfHCInPkts[11]    :      0
COSIfHCInPkts[12] :      0  COSIfHCInPkts[13]    :      0
COSIfHCInPkts[14] :      0  COSIfHCInPkts[15]    :      0
COSFrmsDxDueToFilters :      0  nicDmaWriteQueueFull  :      0
nicDmaWrHiPQFull  :      0  nicNoMoreRxBDs       :      0
=====
    
```

PageUP/PageDN to scroll. Ins/Del toggles refresh. ESC to exit

PageUP/PageDN to scroll. Ins/Del toggles refresh. ESC to exit

CHIP Statistics

```

=====
ifInDiscards      :      0  ifInErrors             :      0
nicRecvThresholdHit :      0  nicDmaReadQueueFull    :      0
COSIfHCOutPkts[00] :      0  COSIfHCOutPkts[01]    :      0
COSIfHCOutPkts[02] :      0  COSIfHCOutPkts[03]    :      0
COSIfHCOutPkts[04] :      0  COSIfHCOutPkts[0]     :      0
Rxed Packets (Ring#05) :      0  0                      :      0
Rxed Packets (Ring#06) :      0  0                      :      0
Rxed Packets (Ring#07) :      0  0                      :      0
Rxed Packets (Ring#08) :      0  0                      :      0
=====
    
```

Rxed Packets (Ring#09) : 0 0

PageUP/PageDN to scroll. Ins/Del toggles refresh. ESC to exit

## 10.79 ringIndex

**Command:**

**Description:** Dump Ring Index. Load Mac driver before running.

**Syntax:** ringindex t|r

**Options:**

- n<DEC> Number of Rx Ring to dump (Default=1) (def=1)
- r Dump Rx Ring Index
- t Dump Tx Ring Index

**Example:**

1 Load MAC driver and display TX and RX Ring Index.

```
0:> loaddrv
Bus Number      : 1
Device/Funtion  : 11/0
Base Address    : 0xfb010000
IRQ             : 9
Bringing up MAC driver ... OK
PHY calculated ID: 60008162
BCM5702/03 Internal Phy Rev#2
Configuring BCM54xx ... Done
Determining Link Speed ... 1000Base-T Full Duplex
0:> ringindex rt
```

	Mailbox	RBDI	RBDC	HC	StsBlk	Driver
	=====	=====	=====	=====	=====	=====
RxStdPidx	100	100	100	---	---	100
RxStdCidx	---	---	---	000	000	000
RetRPidx#00	---	---	---	000	---	---
RetRCidx#00	000	---	---	---	---	000

  

	Mailbox	SBDI	SBDSEL	HC	StsBlk	Driver
	=====	=====	=====	=====	=====	=====
SendHostPidx#00	000	000	---	---	---	000
SendHostCidx#00	---	---	000	000	000	000
SendHostPidx#01	000	000	---	---	---	000
SendHostCidx#01	---	---	000	000	000	000
SendHostPidx#02	000	000	---	---	---	000
SendHostCidx#02	---	---	000	000	000	000
SendHostPidx#03	000	000	---	---	---	000
SendHostCidx#03	---	---	000	000	000	000

## 10.80 phyctrl

**Command:** phyctrl

**Description:** Configure Phy Speed

**Syntax:** phyctrl

**Options:**

- f<string> file contains initialization scripts
- h force half duplex
- r reset PHYs
- s<HEX> 0:10 Mbps, 1:100 Mbps, 2:1000 Mbps, 3 - Auto (def=00000002)

## 10.81 txpkt

**Command:** txpkt

**Description:** Transmit Packets. Driver must be loaded.

**Syntax:** txpkt

**Options:**

- a<DEC> Specify number of IP fragment count (def=0)
- b<DEC> Burst length (def=0)
- c Clear Statistics (always on: kept for bcmediag compatibility)
- d<DEC> Interpacket delay in microseconds (def=txcfg)
- e Insert raw checksum into the packet
- f<DEC> Max number of fragments (def=1)
- g<HEX> Tx Flags (def=txcfg)
- h<DEC> Specifies IP total length (Default: correct IP length)
- I Incremental length
- j Random number of fragments
- k Use random packet length

- l<DEC> Start packet length (def=txcfg)
- m Use multiple Tx ring test (def=txcfg)
- n<DEC> Number of packet to transmit (def=0)
- o<HEX> Buffer Offset (def=00000000)
- p<DEC> Number of Tx rings to use in multiple ring test (def=txcfg)
- q<DEC> Number of Packets per ring (def=txcfg)
- r<DEC> Tx ring number (def=txcfg)
- v Insert fragment count and fragment size into the packet
- x Display Statistics (always on: kept for bcmediag compatibility)
- u Standard Fragmentation
- w<DEC> Low watermark max RxFrame value (0-65535)
- z<DEC> Minimum fragment size, -1=disable (def=-1)
- A Use static buffer
- B<HEX> Begin Cipher ID for the IPsec test. (This command only valid for the IPsec supported devices).
- E<HEX> End Cipher ID for the IPsec test. (only valid for IPsec supported devices). Combine this options with -B<HEX> options can specify the testing range of IPsec algorithm.
- G<HEX> IPsec random test.
- L Don't initialize packets
- R Enable RS232 statistic update
- S TCP segmentation test -V<DEC> Create IPV4 or IPV6 IPS packets. (def= V4)
  - 1: Random mix Non-IPsec/IPsec packets
  - 2: Random GCM/GMAC salt.
  - 4:Random encryption/authentication keys.
  - 7: Random all above.

-H<DEC> Percentage of the IPSec packets if Random mix Non-IPSec/IPSec packets.

## 10.82 statusblk

**Command:** statusblk

**Description:** Display Status Block

**Syntax:** statusblk

**Example:**

```
0:> statusblk

***** STATUS Block @ 0x0027c040 *****
Status : 0x0000
Rx Standard CIdx : 0      Rx Jumbo CIdx : 0      Rx Mini CIdx : 0
Rx PIdx[00] : 0          Send CIdx[00] : 0
Rx PIdx[01] : 0          Send CIdx[01] : 0
Rx PIdx[02] : 0          Send CIdx[02] : 0
Rx PIdx[03] : 0          Send CIdx[03] : 0
Rx PIdx[04] : 0          Send CIdx[04] : 0
Rx PIdx[05] : 0          Send CIdx[05] : 0
Rx PIdx[06] : 0          Send CIdx[06] : 0
Rx PIdx[07] : 0          Send CIdx[07] : 0
Rx PIdx[08] : 0          Send CIdx[08] : 0
Rx PIdx[09] : 0          Send CIdx[09] : 0
Rx PIdx[10] : 0          Send CIdx[10] : 0
Rx PIdx[11] : 0          Send CIdx[11] : 0
Rx PIdx[12] : 0          Send CIdx[12] : 0
Rx PIdx[13] : 0          Send CIdx[13] : 0
Rx PIdx[14] : 0          Send CIdx[14] : 0
Rx PIdx[15] : 0          Send CIdx[15] : 0
```

## 10.83 stsbk

**Command:** stsbk

**Description:** Display Statistics Block.

**Syntax:** stsbk

**Example:**

```
0:> stsbk
***** STATISTICS Block @ 0x0027c0c0 *****
ifHCInOctets : 0 etherStatsFragments : 0
ifHCInUcastPkts : 0 ifHCInMulticastPkts : 0
ifHCInBroadcastPkts : 0 d3StatsFCSErrors : 0
d3StatsAlignmentErrors : 0 xonPauseFramesReceived : 0
xoffPauseFramesReceived : 0 macControlFramesReceived : 0
xoffStateEntered : 0 dot3StatsFramesTooLong : 0
etherStatsJabbers : 0 etherStatsUndersizePkts : 0
inRangeLengthError : 0 outRangeLengthError : 0
etherStatsPkts64Octets : 0 etherStatsPkts65-127 : 0
etherStatsPkts128-255 : 0 etherStatsPkts256-511 : 0
etherStatsPkts512-1023 : 0 etherStatsPkts1024-1522 : 0
etherStatsPkts1523-2047 : 0 etherStatsPkts2048-4095 : 0
etherStatsPkts4096-8191 : 0 etherStatsPkts8192-9022 : 0
```

```

ifHCOutOctets          :          0 etherStatsCollisions      :          0
outXonSent             :          0 outXoffSent              :          0
flowControlDone        :          0 d3StatsInt1MacTxErrors :          0
d3StatsSingleColFrames :          0 d3StatsMultipleColFrames:          0
dt3StatsDeferredTx     :          0 d3StatsExcessiveCol    :          0
d3StatsLateCol         :          0 d3Collided2Times      :          0
d3Collided3Times       :          0 d3Collided4Times      :          0
d3Collided5Times       :          0 d3Collided6Times      :          0
d3Collided7Times       :          0 d3Collided8Times      :          0
d3Collided9Times       :          0 d3Collided10Times     :          0
d3Collided11Times      :          0 d3Collided12Times     :          0
d3Collided13Times      :          0 d3Collided14Times     :          0
d3Collided15Times      :          0 ifHCOutUcastPkts      :          0
d3StatsCarSenseErrors  :          0 ifOutDiscards         :          0
COSIfHCInPkts[00]     :          0 COSIfHCInPkts[01]    :          0
COSIfHCInPkts[02]     :          0 COSIfHCInPkts[03]    :          0
COSIfHCInPkts[04]     :          0 COSIfHCInPkts[05]    :          0
COSIfHCInPkts[06]     :          0 COSIfHCInPkts[07]    :          0
COSIfHCInPkts[08]     :          0 COSIfHCInPkts[09]    :          0
COSIfHCInPkts[10]     :          0 COSIfHCInPkts[11]    :          0
COSIfHCInPkts[12]     :          0 COSIfHCInPkts[13]    :          0
COSIfHCInPkts[14]     :          0 COSIfHCInPkts[15]    :          0
COSFrmsDxDueToFilters :          0 nicDmaWriteQueueFull  :          0
nicDmaWrHiPQFull      :          0 nicNoMoreRxBDs       :          0
ifInDiscards          :          0 ifInErrors            :          0
nicRecvThresholdHit   :          0 nicDmaReadQueueFull  :          0
COSIfHCOutPkts[00]    :          0 COSIfHCOutPkts[01]   :          0
COSIfHCOutPkts[02]    :          0 COSIfHCOutPkts[03]   :          0
COSIfHCOutPkts[04]    :          0 COSIfHCOutPkts[05]   :          0
COSIfHCOutPkts[06]    :          0 COSIfHCOutPkts[07]   :          0
COSIfHCOutPkts[08]    :          0 COSIfHCOutPkts[09]   :          0
COSIfHCOutPkts[10]    :          0 COSIfHCOutPkts[11]   :          0
COSIfHCOutPkts[12]    :          0 COSIfHCOutPkts[13]   :          0
COSIfHCOutPkts[14]    :          0 COSIfHCOutPkts[15]   :          0
nicDmaRdHPQueueFull  :          0 nicSendDataCompQueueFull:          0
nicRingSetSdPIIdx    :          0 nicRingStatusUpdate   :          0
nicInterrupts         :          0 nicAvoidedInterrupts  :          0
nicSendThresholdHit   :          0
Phy CRC counter       :          0

```

## 10.84 txcfg

**Command:** txcfg

**Description:** Configure protocol packets for transmission

**Syntax:** txcfg

**Example:**

```

0:> txcfg
 1. Source MAC                : 10:11:12:13:14:15
 2. Destination MAC          : 00:01:02:03:04:05
 3. Length (14-65535)        : 1514
 4. Packet Type {Non(0),EthV2(1),802.3(2),SNAP(3)}: Ethernet II
 5. Protocol Field {Non(0),IP(1),ARP(2),BRM(3)}  : IP
 6. Source IP                 : 10.2.1.1
 7. Destination IP           : 10.2.1.2
 8. IP Protocol Field { UDP(17), TCP(6) }        : UDP
   80. Source Port            : 100
   81. Destination Port      : 200
 9. IP Option Length (32-bit Words)              : 0
10. TCP Option Length (32-bit Words)             : 0
11. Pattern { As-is(0), Inc(1), Random(2), 0s(3), FFs(4),
    AA55(5), 55AA(6),IP_Iden-Inc(7),Load from file(8)
    00ff8(9) 00ff16(10), 00ff32(11), 00ff(12)}  : Increment (00,01,02 ...)

```

```
12. IP Checksum Offload{ YES(1), NO(0) }      : NO
13. TCP/UDP Checksum Offload { YES(1), NO(0) } : NO
14. TCP/UDP Pseudo Checksum Only { YES(1), NO(0) } : NO
15. Insert VLAN Tag { YES(1), NO(0) }        : NO
16. VLAN Tag                                  : 1
17. Random IP header field { YES(1), NO(0) }  : NO
18. Random TCP/UDP header field { YES(1), NO(0) } : NO
0. Exit
```

Enter your choice (option=parameter) ->

## 10.85 rxcfg

**Command:** rxcfg

**Description** Configure RX parameters.

**Syntax:** rxcfg

**Example:**

```
0:> rxcfg
1. Host Loopback { Enable(1), Disable(0) }    : Disable
2. Modify Rx Packet { Enable(1), Disable(0) }  : Disable
3. Dump Rx Packet { None(1),Hex(2), Decode(3) } : None
4. Dump Rx Length                              : 64
5. Tx Fragment Length                          : 1518
6. Tx Flags                                    : 0000
7. Tx VLAN Tag                                : 0000
8. Tx Ring Number                             : 0
9. Tx Generate CRC { Enable(1), Disable(0) }   : Enable
10. Capture Rx Packet { Enable(1), Disable(0) } : Enable
11. Rx Mask
0. Exit
```

Enter your choice (option=parameter) ->

## 10.86 tprot

**Command:** tprot

**Description:** Blast with TPROT Packets. This command is same as command 'blast -trm'

**Syntax:** tprot

**Options:**

-d<DEC> Interpacket gap in microseconds (def=10)

## 10.87 qstat

**Command:** qstat

**Description:** Get a quick NIC statistic. [qstat string] used to select specific statistic.

**Syntax:** qstat [qstat string]



**Options:**

- c Clear statistic
- l List all qstat string

**10.88 drvrcfg**

**Command:** drvrcfg

**Description:** configure driver parameters

**Syntax:** drvrcfg

**Options:**

- a<DEC> Turn on/off autolink capability (def=0)
- q<DEC> Configure Rx ring size (def=0)
- r<DEC> Turn on/off rxflow capability (def=0)
- t<DEC> Turn on/off txflow capability (def=0)
- x<DEC> Configure Tx ring size (def=0)

**10.89 irt**

**Command:** irt

**Description:** Test an individual register. Test an individual register with a specified number of reads/writes.

**Syntax:** irt

**Options:**

- n<DEC> Number of read/write accesses (def=1)
- r<string> Register offset

**10.90 macmrd**

**Command:** macmrd

**Description:** Test an individual register. Test an individual register with a specified number of reads/writes.

**Syntax:** macmrd

**Options:**

- d<DEC> Delay in uS (def=0)
- n<DEC> Number of read/write accesses (def=1)
- r<string> Register offset

**10.91 miimrd**

**Command:** miimrd

**Description:** Test an individual register. Test an individual register with a specified number of reads/writes.

**Syntax:** miimrd

**Options:**

- d<DEC> Delay in uS (def=0)
- n<DEC> Number of read/write accesses (def=1)
- r<string> Register offset

**10.92 pcimrd**

**Command:** pcimrd

**Description:** Test an individual register. Test an individual register with a specified number of reads/writes.

**Syntax:** pcimrd

**Options:**

- d<DEC> Delay in uS (def=0)
- n<DEC> Number of read/write accesses (def=1)
- r<string> Register offset

**10.93 inband**

**Command:** inband

**Description:** Force inband

**Syntax:** inband

**Options:**

- s<HEX> Standard inband only(5701 GPHY) 0:disable, 1:enable (def=0)
- x<HEX> Extend inband, 0:disable Tx/Rx, 1:enable Tx, 2:enable Rx, 3:enable Tx/Rx. (def=0)

### 10.94 mwrite

**Command:** mwrite

**Description:** Write PHY registers via MII Management interface

**Syntax:** mwrite <addr > <data>

Address range: 0x00 – 0x1F

**Example:**

1. Write 0x15 to MII register 2

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

### 10.95 mread

**Command:** mread

**Description:** Read PHY registers via MII Management interface

**Syntax:** mread <begin>[-<end> | <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

```
0:> mread 0 5
00: 1100 7949 0020 6051 01e1
```

## 10.96 mdev

**Command:** mdev

**Description:** Select current PHY to be accessed. The default device ID is 0x01. If no parameter is entered, it displays the current PHY address setting. This controls the internal **phy\_device2** setting in the diagnostic.

**Syntax:** mdev [<phy\_id>]

**Example:**

```
0:> mdev 1
Phy Address = 0x1
```

## 10.97 m2dev

**Command:** m2dev

**Description:** Select current PHY to be accessed. The default device ID is 0x01. If no parameter is entered, it displays the current PHY address setting. This controls the internal **phy\_device1** setting in the diagnostic.

**Syntax:** m2dev [<phy\_id>]

**Example:**

```
0:> m2dev 1
Phy Address = 0x1
```

## 10.98 miimode

**Command:** miimode

**Description:** Select Auto Mode of MII Access. 0:disable, 1:enable

**Syntax:** miimode <1|0>

**Example:**

```
0:> miimode 0
Setting MII auto mode to OFF
0:> miimode 1
Setting MII auto mode to ON
0:> miimode
```

## 10.99 lbertram

**Command:** lbertram

**Description:** Load data to PHY BIST RAM

**Syntax:** lbertram [filename]

**Options:**

- c<DEC> channel number (def=0)
- e enable BIST
- f<string> File name containing BIST data

### 10.100 dbertram

**Command:** dbertram

**Description:** Dump PHY BIST RAM

**Syntax:** dbertram

**Options:**

- b<HEX> Begin of BIST RAM (def=00000000)
- c<DEC> channel number (def=0)
- e<HEX> End of BIST RAM (def=000000FF)
- r Dump Rx BIST RAM
- t Dump Tx BIST RAM

### 10.101 bertstats

**Command:** bertstats

**Description:** Dump PHY BIST statistics

**Syntax:** bertstats

### 10.102 rm

**Command:** rm

**Description:** Read MII Registers

**Syntax:** rm

### 10.103 mrloop

**Command:** mrloop

**Description:** loop on MII read. This is special test routine for MII read. It loops on MII register read until user abort or if value is zero.

**Syntax:** mrloop <addr>

### 10.104 phymse

**Command:** phymse

**Description:** PHY mean square error.

**Syntax:** phymse

**Options:**

-p           Polling continuously at 100ms interval

### 10.105 initphy

**Command:** initphy

**Description:** Initialize phy.

**Syntax:** initphy

### 10.106 tapdump

**Command:** tapdump

**Description:** Read PHY internal TAP values

**Syntax:** tapdump <dfc> | <echo> | <sd> | <mse>

### 10.107 memsearch

**Command:** memsearch

**Description:** Search a Data Pattern in Memory. The default, begin address = 0, and len = 0x20000. The data pattern must be specified.

**Syntax:** memsearch begin[-end | len] data

### 10.108 read

**Command:** read

**Description:** Read Memory

**Syntax:** read [!#\*\$~^!ImSsxX]<begin> [-end | len]

! = Configuration space (32)

S = Configuration space (16)

X = Configuration space (8)  
# = Registers (32) (default)  
\* = SRAM (32)  
\$ = NVRAM (SEEPROM/FLASH) (32)  
m = MII registers (16)  
~ = VPD Access (32)  
I = indirect access of host memory (32)  
^ = internal scratchpad (32)  
l = direct access of host memory (32)  
s = direct access of host memory (16)  
x = direct access of host memory (8)

**Example:**

1. Read from Configuration space

```
0:> read !10  
000010: f4000004
```

2. Read from Register

```
0:> read #10  
000010: f4000004
```

3. Read from SRAM

```
0:> read *10  
000010: 00010001
```

4. Read from internal scratchpad

```
0:> read ^00  
000000: 000312ae
```

## 10.109 write

**Command:** write

**Description:** Write Memory

**Syntax:** write [!#\*\$~^lImSsxX]<address> [-end | len] data

! = Configuration space (32)

S = Configuration space (16)

X = Configuration space (8)

# = Registers (32) (default)

\* = SRAM (32)

\$ = NVRAM (SEEPROM/FLASH) (32)

m = MII registers (16)

~ = VPD Access (32)

I = indirect access of host memory (32)

^ = internal scratchpad (32)

l = direct access of host memory (32)

s = direct access of host memory (16)

x = direct access of host memory (8)

**Example:**

1. Write to configuration space.

```
0:> write !10 f4000004
```

2. Write to register.

```
0:> write #10 f4000004
```

3. Write to SRAM

```
0:> write *10 10001
```

4. Write to internal scratchpad

```
0:> write ^10 f4000004
```

### 10.110 poll

**Command:** poll

**Description:** poll Memory

**Syntax:** poll [!#\*\$~^!ImSsxX]<addr> [[!#\*\$~^!ImSsxX]<addr>...]

! = Configuration space (32)

S = Configuration space (16)

X = Configuration space (8)



# = Registers (32) (default)  
\* = SRAM (32)  
\$ = NVRAM (SEEPROM/FLASH) (32)  
m = MII registers (16)  
~ = VPD Access (32)  
I = indirect access of host memory (32)  
^ = internal scratchpad (32)  
l = direct access of host memory (32)  
s = direct access of host memory (16)  
x = direct access of host memory (8)

**Options:**

-n<DEC>    Number of changes to print before stop (def=0)

**10.111 setbit**

**Command:** setbit

**Description:** Read-Modify Memory by ored with <bits>

**Syntax:** setbit [!#\*\$~^lImSsxX]<addr> <bit#> [<bit#>] ...

! = Configuration space (32)  
S = Configuration space (16)  
X = Configuration space (8)  
# = Registers (32) (default)  
\* = SRAM (32)  
\$ = NVRAM (SEEPROM/FLASH) (32)  
m = MII registers (16)  
~ = VPD Access (32)  
I = indirect access of host memory (32)

^ = internal scratchpad (32)

l = direct access of host memory (32)

s = direct access of host memory (16)

x = direct access of host memory (8)

### 10.112 clearbit

**Command:** clearbit

**Description:** Read-Modify Memory by anded with ~<bits>

**Syntax:** clearbit [!#\*\$~^lImSsxX]<addr> <bit#> [<bit#>] ...

! = Configuration space (32)

S = Configuration space (16)

X = Configuration space (8)

# = Registers (32) (default)

\* = SRAM (32)

\$ = NVRAM (SEEPROM/FLASH) (32)

m = MII registers (16)

~ = VPD Access (32)

I = indirect access of host memory (32)

^ = internal scratchpad (32)

l = direct access of host memory (32)

s = direct access of host memory (16)

x = direct access of host memory (8)

### 10.113 readbit

**Command:** readbit

**Description:** Read the bit specified by <bit#>

**Syntax:** readbit [!#\*\$~^lImSsxX]<addr> <bit#>

! = Configuration space (32)  
S = Configuration space (16)  
X = Configuration space (8)  
# = Registers (32) (default)  
\* = SRAM (32)  
\$ = NVRAM (SEEPROM/FLASH) (32)  
m = MII registers (16)  
~ = VPD Access (32)  
I = indirect access of host memory (32)  
^ = internal scratchpad of host memory (32)  
l = direct access of host memory (32)  
s = direct access of host memory (16)  
x = direct access of host memory (8)

### 10.114 cread

**Command:** cread

**Description:** Read PCI configuration Space of specified device. default - 32 bits read, S - 16 bits read, X - 8 bits read

**Syntax:** cread <bus> <dev> <func> [S|X]<begin> [-end | len]

### 10.115 cwrite

**Command:** cwrite

**Description:** Write PCI configuration space of specified device. default - 32 bits read, S - 16 bits read, X - 8 bits read

**Syntax:** cwrite <bus> <dev> <func> [S|X]<begin> [-end | len] data

### 10.116 pcird

**Command:** pcird

**Description:** Read PCI configuration word.

**Syntax:** pcird <bus> <dev> <func> <addr>

**Example:** pcird 2 1 0 8 : pcird 2 1 0 8 : Read PCI device 1,  
function 0 on bus 2.

### 10.117 pciwr

**Command:** pcird

**Description:** Write PCI configuration word.

**Syntax:** pciwr <bus> <dev> <func> <addr> <dword>

**Example:** pciwr 2 1 0 8 9 : Write PCI device 1, function 0 on  
bus 2 with a 9.

### 10.118 vpdtest

**Command:** vpdtest

**Description:** Run VPD Memory Test. Write designed pattern to VPD storage. Then read back and compare with designed pattern.

**Syntax:** vpdtest

**Options:**

- d Force destructive test
- n<DEC> iteration
- p<DEC> Pattern to test (def=0)  
0 - Increment; 1 - Decrement ;1 - 0's 2 - FF's ; 3 - AA55; 4 - 55AA
- r Random address test
- w Force write test enable

### 10.119 regtest

**Command:** regtest

**Description:** MAC registers read/write test. Driver must be unloaded.

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

**Options:**

- i Also run indirect memory test
- n<DEC> iteration (The default iteration is 1. 0 means run forever)
- r<DEC> repeat count for each register test (def=1)

-I Do not perform reset before test

### 10.120 miitest

**Command:** miitest

**Description:** Run MII Memory Test. PHY registers read write test

**Syntax:** miitest [iteration]

**Options:**

-n<DEC> iteration (The default iteration is 1. 0 means run forever)

### 10.121 msi

**Command:** msi

**Description:** Run MSI Test Manually

**Syntax:** msi

**Options:**

-c<HEX> message count (2 to powered of c) (def=00000003)

-d option removed, kept for bmediag compatibility

-I initializing MSI block

-o<DEC> offset (def=0)

### 10.122 memtest

**Command:** memtest

**Description:** Test memory blocks such as scratch pad, BD sram, DMA sram, Mbuf, external SRAM. Running “diagcfg” can configure memory block ranges. See “diagcfg” for detail. Driver must be unloaded.

**Syntax:** memtest [iteration]

**Options:**

-b Test BD SRAM

-c Test MBUF special

-d Test DMA SRAM

-e Test External Memory

- m Test MBUF SRAM
- n<DEC> iteration (The default iteration is 1. 0 means run forever)
- p Test CPU GPRs
- s Test Scratch Pad
- x Test MBUF SRAM via DMA

### 10.123 setest

**Command:** setest

**Description:** Run NVRAM Test

**Syntax:** setest [iteration]

**Options:**

- e extensive test
- d<HEX> ending offset (with -e option) (def=FFFFFFFF)
- n<DEC> iteration
- q quiet mode
- r read only test
- s<HEX> start offset (with -e option) (def=00000000)

### 10.124 bist

**Command:** bist

**Description:** Run BIST. The default iteration is 1. 0 means run forever.

**Syntax:** bist [iteration]

### 10.125 nictest

**Command:** nictest

**Description:** Run a set of NIC Tests. NIC test can include memory test, serial eeprom test, interrupt test, packet exchange, MAC registers test, Mii registers test, cpu test, dma test. This test can to be configured by running “diagcfg”. See “diagcfg” for details. If a “test list” is not entered below then a set of default tests are run.

**Syntax:** nictest [test list]

abcd -- runs all tests

b -- runs all test in group B

a3 b1 -- runs test a3 and b1 only

a124b2 -- runs test a1,a2,a4 and b2

**Options:**

-e run NVRAM verification also

-n<DEC> iteration

### 10.126 intrtest

**Command:** intrtest

**Description:** Interrupt Test

**Syntax:** intrtest [iteration]

**Options:**

-n<DEC> iteration (The default iteration is 1. 0 means run forever.)

### 10.127 pkttest

**Command:** pkttest

**Description:** Perform MAC and/or PHY loopback test. This test will send 100 packets in incremental length and check for contents of loopbacked packets.

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

**Options:**

-e run external loopback test

-m run mac loopback test

-n<DEC> iteration (The default iteration is 1. 0 means run forever.)

-p run phy loopback test

### 10.128 cputest

**Command:** cputest

**Description:** TX / RX CPU Test. This test needs an input CPU file in the same location as b57diag.exe. The default file name is cpu.bin or cpu05.bin unless specified by -f option.

**Syntax:** cputest [iteration]

**Options:**

- f<string> input filename
- n<DEC> iteration (The default iteration is 1. 0 means run forever)

### 10.129 dmatest

**Command:** dmatest

**Description:** DMA Test

**Syntax:** dmatest [iteration]

**Options:**

- 4 Allocate 4k-aligned buffers
- a<HEX> NIC address (def=00002100)
- d Display DMA info.
- f Force to use 32-bit bus
- l<HEX> Length of DATA to do DMA (def=00000400)
- n<DEC> iteration
- o<HEX> Buffer offset (def=00000000)

### 10.130 teste

**Command:** teste

**Description:** The command enables tests. It effects nictest, regtest, pkttest, and memtest commands. The test must starts with test group alpha (a-d). If no number is entered, all tests in that group are enabled.

**Syntax:** teste [<tests> [<tests>...]]



**Example:**   **teste a12bc**    -- Enable test a1, a2, all tests in group b and c  
              **teste ab cd**    -- Enables all tests  
              **teste**            -- Display enabled tests

### 10.131 testd

**Command:** testd

**Description:** The command disables tests. It effects nictest, regtest, pkttest, and memtest commands. The test must starts with test group alpha (a-d). If no number is entered, all tests in that group are disabled.

**Syntax:** testd [<tests> [<tests>...]]

**Example:**   **testd a12bc**    -- Disable test a1, a2, and all tests in group b and c.  
              **testd ab cd**    -- Disables all tests.  
              **testd**            -- Display disabled tests.

### 10.132 asftest

**Command:** asftest

**Description:** ASF Test

**Syntax:** asftest

**Options:**

      -n<DEC>    iteration

### 10.133 bustest

**Command:** bustest

**Description:** PCI Bus Test

**Syntax:** bustest

**Options:**

      -a<HEX>    NIC address to DMA data to. (def=00002100)

      -d<HEX>    delay poll dma done polling (def=00000000)

      -e<DEC>    End of test case (def=259)

      -g            Insert debugging information

      -h<DEC>    Maxmum length (def=1024)

- i<DEC>    Number of transactions per pattern (def=10)
- l<DEC>    Minimum length (def=256)
- n<DEC>    iteration
- o<DEC>    Number of consecutive patterns (def=1)
- p<DEC>    DMA priority (def=0)
- s<DEC>    Start of test case (def=0)
- t<DEC>    Transient fixed pattern (def=0)
- v<DEC>    Verbose level (0..2) (def=1)
- L            Loop

There are total 260 test cases (258 unique tests cases) which are described as follows:

Test case#	Pattern
=====	=====
0	ffffffff ffffffff 00000000 00000000
1	ffffffff ffffffffe 00000000 00000000
2	ffffffff ffffffffd 00000000 00000000
.	.
.	.
.	.
64	7ffffffff ffffffff 00000000 00000000
65	00000000 00000000 ffffffff ffffffff
66	00000000 00000000 ffffffff ffffffffe
67	00000000 00000000 ffffffff fffffffd
.	.
.	.
.	.
129	00000000 00000000 7ffffffff ffffffff
130	00000000 00000000 ffffffff ffffffff (repeat)
131	00000000 00000001 ffffffff ffffffff
132	00000000 00000002 ffffffff ffffffff
.	.
.	.
.	.
194	80000000 00000000 ffffffff ffffffff

```

195          ffffffff ffffffff 00000000 00000000 (repeat)
196          ffffffff ffffffff 00000000 00000001
197          ffffffff ffffffff 00000000 00000002
.
.
.
259          ffffffff ffffffff 80000000 00000000
    
```

If you run bustest command without any parameters, it will perform DMA testing on all 260 patterns with 10 iterations per pattern and different data length in each iteration. First eight bytes of data are used to store the following info for debug:

```

byte 0-4 : length
byte 5-6 : iteration#
byte 6-7 : test case#
    
```

### 10.134 sramtest

**Command:** sramtest

**Description:** SRAM Test

**Syntax:** sramtest <begin> [ <len> |<-end>]

### 10.135 msitest

**Command:** msitest

**Description:** MSI Test

**Syntax:** msitest

**Options:**

```

-c<HEX>    message count (2 to powered of c) (def=00000003)
-I          initializing MSI block
-n<DEC>    iteration
    
```

### 10.136 romtest

**Command:** romtest

**Description:** ROM Test

**Syntax:** romtest

**Options:**

-n<DEC> iteration

**10.137 gpiotest**

**Command:** gpiotest

**Description:** do GPIO test

**Syntax:** gpiotest

**Options:**

-n<DEC> iteration

**10.138 cpudiag**

**Command:** cpudiag

**Description:** run diagnostic from internal CPU

**Syntax:** cpudiag

**Options:**

-b Test BD SRAM (0x0000-0x0fff and 0x4000-0x7fff)

-d Test DMA SRAM (0x2000-0x3fff)

-m Test MBUF SRAM (0x8000-0x00000005)

-n<DEC> Iteration

-r Register Test

-T Test with Tx CPU

-R Test with Rx CPU

**10.139 pcicfgtest**

**Command:** pcicfgtest

**Description:** Run PCI Config. Reg. Test

**Syntax:** pcicfgtest

**Options:**

- I Do not perform reset before test
- r<DEC> repeat count for each register test (def=1)
- n<DEC> iteration

### 10.140 petest

**Command:** petest

**Description:** Perform parity error test on a bridge

**Syntax:** petest <bridge>

### 10.141 errctrl

**Command:** errctrl

**Description:** Configure Error Control Setting

**Syntax:** errctrl [w|c|a|l|s]

- w - Wait on Error  
Program will pause and wait for user's action (eng. default)
- c - Continue on Error  
Program will continue even if the error is detected
- a - Abort on Error (Manufacturing default)  
Program stops
- l - Loop on Error  
Program will retry the same test
- s - Skip on Error  
Program will skip the rest of the present test

### 10.142 sedvt

**Command:** sedvt

**Description:** Perform NVRAM dvt test. When 'init' subcommand is entered, the NVRAM is initialized into pseudo random pattern. The original content is DESTROYED.

**Syntax:** sedvt [init]

**Options:**

- a Access test
- e Erase with reset
- f force
- l<HEX> size (def=00000100)
- n<DEC> iteration (def=0)

- p pause
- r Read Test with reset
- s skip checking entire NVRAM
- w Read/Write Test with reset

### 10.143 miimisctest

**Command:** miimisctest

**Description:** Run MII Misc. Tests.

**Syntax:** miimisctest

**Options:**

- n<DEC> iteration

### 10.144 cpugprtest

**Command:** cpugprtest

**Description:** Run CPU GPR test.

**Syntax:** cpugprtest

**Options:**

- r run rx\_cpu only
- t run tx cpu only
- u run Address Up

### 10.145 dmashasta

**Command:** dmashasta

**Description:** DMA Test

**Syntax:** dmashasta

**Options:**

- r Disable the read DMA test
- w Disable the write DMA test

- e Disable the chip reset execution before each DMA test
- l<HEX> Number of BD sot DMA on read (0x1-0x79) (def=0xA)

### 10.146 binchksum

**Command:** binchksum

**Description:** Verify the checksum of each piece of firmware in the input file that contains a complete NVRAM image.

**Syntax:** binchksum -f<filename>

### 10.147 pmdcfg

**Command:** pmdcfg

**Description:** Display Power Management Info

**Syntax:** pmdcfg

**Options:**

- a<HEX> 0 to add a pattern; otherwise delete (def=00000000)
- l<HEX> length (def=00000000)
- p<HEX> pattern. 0 - Increment; 1 - Increment (def=00000000)

### 10.148 pmpd

**Command:** pmpd

**Description:** Power Down MAC. Input file wol.txt should be found in the same location of b57diag.exe. The input file contains patterns. If the file name is not specified, data zero will be used.

**Syntax:** pmpd [filename]

**Options:**

- a<HEX> 1 enables ACPI Packet Match (def=00000000)
- c<HEX> 0 to add a pattern; otherwise delete (def=00000000)
- f<string> File name which contains patterns
- m<HEX> 1 enables Magic MAC detection (def=00000000)

- o<HEX>   offset (def=00000000)
- v<HEX>   Versbose level (default=0) (def=00000000)

### 10.149 intr

**Command:** intr

**Description:** Dump Interrupt Info

**Syntax:** intr

**Example:**

```
0:> intr
Interrupt Count      : 48337
IPC MASK             : 0xb8 0x0c
IPC IS1 IS2          : 0x00 0x00
IPC IRR1 IRR2        : 0x18 0x00
IPC ILCR1 ILCR2      : 0x20 0x0e
Worst Intr. Latency  : 54476 CPU clocks/50 uS
```

### 10.150 intrctrl

**Command:** intrctrl

**Description:** Control Interrupt Controller

**Syntax:** intrctrl u|m

- u : unmask current interrupt
- m : mask current interrupt

**Example:**

1. Mask current interrupt  
0:irq> intrctrl m  
Masking Interrupt 10
2. Unmask current interrupt  
0:irq> intrctrl u  
Unmasking Interrupt 10

### 10.151 intt

**Command:** intt

**Description:** Interrupt Tracer. This is special function to monitor interrupt functions.

**Syntax:** intt

### 10.152 mbuf

**Command:** mbuf



**Description:** Dump Content of Mbufs. The display command must be specified by -c option or 'chain', 'info', 'cluster', 'hdr', and 'ckhdr'.

**Syntax:** mbuf [chain|info|cluster|hdr|ckhdr]

**Options:**

-c<HEX> command

- 0 - displays a Mbuf
- 1 - displays a Mbuf chain
- 2 - displays general Mbuf information
- 3 - displays Mbuf Cluster
- 4 - Check Mbuf header corruption
- 5 - Dump all Mbuf headers

-m<HEX> display mode, 0: decode, 1: in hex (def=00000000)

-n<DEC> Mbuf number to display/decode (def=256)

-w Mbuf workaround

**Example:**

1. Display Mbuf chain.

```
0:> mbuf chain
->143->144->145->146->147->148->149->14a->14b->14c->14d->14e->14f->150
->151->152->153->154->155->156->157->158->159->15a->15b->15c->15d->15e
->15f->160->161->162->163->164->165->166->167->168->169->16a->16b->16c
->16d->16e->16f->170->171->172->173->174->175->176->177->178->179->17a
```

## 10.153 loaddrv

**Command:** loaddrv

**Description:** Load Driver

**Syntax:** loaddrv

**Options:**

-4<HEX> Enable 4k-aligned memory (def=00000000)

-o<HEX> Allocate memory with specified offset (def=00000000)

-j Allocate memory for Jumbo packet

-t Allocate memory for TCP Segmentation

- f Force Link to Max Speed
- z Allocate memory for optimization (i.e. DOS with EFI BIOS)

**Example:**

```
0:> loaddrv
Reinitializing PCI Configuration Space
Bus Number      : 1
Device/Funtion  : 11/0
Base Address     : 0xfb010000
IRQ             : 9
Bringing up MAC driver ... OK
PHY calculated ID: 60008162
BCM5702/03 Internal Phy Rev#2
Configuring BCM54xx ... Done
Determining Link Speed ... 1000Base-T Full Duplex
```

### 10.154 unloaddrv

**Command:** unloaddrv

**Description:** Unload NIC driver

**Syntax:** unloaddrv

**Example:**

```
0:> unloaddrv

Unloading MAC driver ... OK
```

### 10.155 machalt

**Command:** machalt

**Description:** Halt MAC controller

**Syntax:** machalt

**Example:**

```
0:> machalt
Halting MAC ... OK
```

### 10.156 ftq

**Command:** ftq

**Description:** Dump FTQ

**Syntax:** ftq

**Example:**

```
0:> ftq
```

```
***** Dump FTQ Peak/Write (Control,Full Counter, Write/Peak) *****
DMA Read FTQ (1)      : 00000000 00000000 20000000
DMA High Read FTQ (2) : 00000000 00000000 60002160
DMA Write FTQ (6)     : 00000000 00000000 20000000
DMA High Write FTQ (7) : 00000000 00000000 20000000
DMA Complete Dx FTQ (3) : 00000000 00000000 20000000
Send BD Comp. FTQ (4) : 00000000 00000000 20000000
Send Data Init FTQ (5) : 00000000 00000000 20000000
Send Data Comp. FTQ (9) : 00000000 00000000 20000000
Rx BD Complete FTQ (13) : 00000000 00000000 60002160
Rx Data Complete FTQ (16) : 00000000 00000000 20000000
S/W Type 1 FTQ (8)    : 00000000 00000000 20000000
Host Coalescing FTQ (10) : 00000000 00000000 2000:00000000
MAC TX FTQ (11)      : 00000000 00000000 2000:00000000
Mbuf Cluster Free FTQ (12) : 00000000 00000000 2000:00000000
RX List Placement FTQ (14) : 00000000 00000000 2000:00000000
RX Data Initiator FTQ (15) : 00000000 00000000 2000:00000000

S/W Type 2 FTQ (17)      : 00000000 00000000 2000:00000000
```

### 10.157 addmc

**Command:** addmc

**Description:** Add Multicast MAC

**Syntax:** addmc <xx:xx:xx:xx:xx:xx>

**Example:**

```
0:> addmc FF:FF:00:0A:00:00
```

### 10.158 delmc

**Command:** delmc

**Description:** Delete Multicast MAC

**Syntax:** delmc <xx:xx:xx:xx:xx:xx>

**Example:**

```
0:> delmc FF:FF:00:0A:00:00
```

### 10.159 txmacdes

**Command:** txmacdes

**Description:** Program Destination address to UUT

**Syntax:** txmacdes <xx:xx:xx:xx:xx:xx>

### 10.160 txmacsrc

**Command:** txmacsrc

**Description:** Program Source address to UUT

**Syntax:** txmacsrc <xx:xx:xx:xx:xx:xx>

### 10.161 chklddrv

**Command:** chklddrv

**Description:** Check to see if driver is loaded. Returns 1 if driver is loaded, returns 0 otherwise.

**Syntax:** chklddrv

### 10.162 vlantag

**Command:** vlantag

**Description:** Display/Clear vlanTag information.

**Syntax:** vlantag

**Options:**

-c clear vlanTag info

### 10.163 reg

**Command:** reg

**Description:** register wizard. This command allows user to view edit registers.

**Syntax:** reg [pci|mii] [offset]

### 10.164 exit

**Command:** exit

**Description:** Exit System

**Syntax:** exit

### 10.165 debug

**Command:** debug

**Description:** Display debugs information

**Syntax:** debug <n>

1: Dump TX / RX Stats

- 2: Dump Clock Scale info
- 3: Clear worst interrupt latency
- 4: Toggle indirect access flag
- 5: Toggle PCI-X workaround

**Example:**

- 1. Display debug information.

```
0:> debug 1
Tx Packets Enqueued      :      0
Tx Packet Complete      :      0
Tx Packet Complete Error :      0
Rx Packets               :      0
Rx Unknown Packets      :      0
Rx Bad Packets          :      0
Rx Good Packets         :      0
```

### 10.166 gpiowrite

**Command:** gpiowrite

**Description:** Write a Value into GPIO pin

**Syntax:** gpiowrite <GPIO\_num> <value>

Valid value for <GPIO\_num> is 0-3, <value> is 0 or 1.

**Example:**

- 1. Write 1 to GPIO#1 Pin

```
0:> gpiowrite 1 1
Writing 1 to GPIO#1
```

### 10.167 gpioread

**Command:** gpioread

**Description:** Read GPIO Value

**Syntax:** gpioread

**Example:**

- 1. Read GPIO Pins

```
0:> gpioread
GPIO#0 : 1
GPIO#1 : 1
```

GPIO#2 : 0  
GPIO#3 : 0

### 10.168 pxeepy

**Command:** pxeepy

**Description:** Load PXE Code to MBUF Memory. The file name must be specified in the parameter.

**Syntax:** pxeepy <file>

**Options:**

-f<string> filename

### 10.169 device

**Command:** device

**Description:** Show or Switch Device. If no parameter is entered, it will display all device available.

**Syntax:** device <dev>

**Options:**

-n<HEX> Device Number (def=00000000)

-r Remove all current devices and re-scan available devices

-s Silent mode - do not display devices

### 10.170 version

**Command:** version

**Description:** Display Program Version

**Syntax:** version

### 10.171 help

**Command:** help

**Description:** Enter command group for the list of available commands. If no parameter is entered, all commands are displayed. Example: help vpd. For each command help, type the command and then '?'. Example: memtest ?

**Syntax:** help [vpd|nvram|cpu|dma|packet|mii|mem|test|power|irq|mac|misc]

### 10.172 ?

**Command:** ?

**Description:** Alternate Help Command. This is same command as 'help' command.

**Syntax:** ? [vpd|nvram|cpu|dma|packet|mii|mem|test|power|irq|mac|misc]

### 10.173 radix

**Command:** radix

**Description:** Change System Radix. Radix must be 2-16. Radix used for number entry. 16 means enter number in hex, and 10 means in decimal.

**Syntax:** radix <2 | 8 | 10 | 16>

### 10.174 nolog

**Command:** nolog

**Description:** Close the Current Logfile

**Syntax:** nolog

### 10.175 log

**Command:** log

**Description:** Save all output to log file

**Syntax:** log

**Options:**

-f<string> filename (for bcmediag compatibility only)

-a Append to existing file

### 10.176 pciinit

**Command:** pciinit

**Description:** Initialize PCI configuration registers

**Syntax:** pciinit

## 10.177 pciscan

**Command:** 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

```

## 10.178 dos

**Command:** dos

**Description:** Execute DOS command. If no parameter is entered, DOS shell is entered.

**Syntax:** dos <dos command>

This command is not supported in UEFI version. Use the Shell command instead.

## 10.179 Shell

**Command:** shell

**Description:** Execute a UEFI shell command. If no parameter is entered, the command terminates. This command only works if the applications is run from the UEFI shell prompt. This command is not supported in the DOS version. Use the DOS command instead.

**Syntax:** shell <shell command>

## 10.180 diagcfg

**Command:** diagcfg

**Description:** Configure diagnostics parameter for Memory tests and Manufacturing test (NIC test).

**Syntax:** diagcfg

**Example:**



0:> diagcfg

Diagnostics Configuration Menu

1. Memory Test Configuration Menu
2. Test Configuration Menu
3. Driver Configuration Menu
4. Abort On Failure is enabled
5. Save Configuration

Enter your choice or ESC to exit -> 1

Memory Test Configuration Menu

1. SRAM BD1 Start (0x00000000-0x00000fff) : 00000000
2. SRAM BD1 End (0x00000000-0x00000fff) : 00000fff
3. SRAM BD2 Start (0x00004000-0x00007fff) : 00004000
4. SRAM BD2 End (0x00004000-0x00007fff) : 00007fff
5. SRAM DMA Start (0x00002000-0x00003fff) : 00002000
6. SRAM DMA End (0x00002000-0x00003fff) : 00003fff
7. SRAM MBUF Start (0x00008000-0x00015fff) : 00008000
8. SRAM MBUF End (0x00008000-0x00015fff) : 00000000
9. SRAM SPAD Start (0x00030000-0x00037fff) : 00030000
10. SRAM SPAD End (0x00030000-0x00037fff) : 00037fff
11. Ext. SRAM Start (0x00020000-0x00ffffff) : 00020000
12. Ext. SRAM End (0x00020000-0x00ffffff) : 00ffffff
13. MBUF Bank (1 - Odd ; 2 - Even ; 3 - Both) : 3
0. Exit to previous menu

Enter your choice (option=paramter) -> 0

Diagnostics Configuration Menu

1. Memory Test Configuration Menu
2. Test Configuration Menu
3. Driver Configuration Menu
4. Abort On Failure is enabled
5. Save Configuration

Enter your choice or ESC to exit -> 2

Test Configuration Menu

- A1. Indirect Register.....: Enabled
- A2. Control Register.....: Enabled
- A3. Interrupt.....: Enabled
- A4. Built In Self.....: Enabled
- A5. PCI Cfg Register.....: Enabled
- B1. Scratch Pad.....: Enabled
- B2. BD SRAM.....: Enabled
- B3. DMA SRAM.....: Enabled
- B4. MBUF SRAM.....: Enabled
- B5. MBUF SRAM via DMA.....: Enabled
- B6. External SRAM.....: Disabled
- B7. CPU GPR.....: Enabled
- C1. NVRAM.....: Enabled
- C2. CPU.....: Enabled
- C3. DMA.....: Enabled
- C4. MII.....: Enabled

C5. VPD.....: Enabled  
C6. ASF Miscellaneous.....: Enabled  
C7. Expansion ROM.....: Enabled  
D1. MAC Loopback.....: Enabled  
D2. PHY Loopback.....: Enabled  
D3. External Loopback.....: Disabled  
D5. MII Miscellaneous.....: Enabled  
D6. MSI.....: Enabled

Enter test number to toggle or ESC to exit ->

Diagnostics Configuration Menu

1. Memory Test Configuration Menu
2. Test Configuration Menu
3. Driver Configuration Menu
4. Abort On Failure is enabled
5. Save Configuration

Enter your choice or ESC to exit -> 3

Driver Configuration Menu

1. Rx Coalescing Ticks	: 1000
2. Rx Coalescing Ticks During Intr	: 0
3. Rx Coalescing Frames	: 1
4. Rx Coalescing Frames During Intr	: 0
5. Tx Coalescing Ticks	: 1000
6. Tx Coalescing Ticks During Intr	: 0
7. Tx Coalescing Frames	: 1
8. Tx Coalescing Frames During Intr	: 0
9. Statistics Coalescing Ticks	: 1000000
10. Tx Packet Descriptor Count	: 50
11. Rx Standard Packet Count	: 100
12. Rx Jumbo Packet Count	: 50
13. Enable Mini Ring {Yes(1),No(0)}	: 1
14. Mini Ring Packet Size (64-512)	: 64
15. External Memory Exists {Yes(1), No(0)}	: 0
16. MBUF Base	: 0x008000
17. MBUF Length	: 0x018000
18. Tx Flow Control { Enable(1),Disable(2) }	: Disable
19. Rx Flow Control { Enable(1),Disable(2) }	: Disable
20. Auto Link Speed { Enable(1),Disable(2) }	: Enable
21. Send Ring Size { 32, 64, 128, 256, 512 }	: 512
22. Rx Ring Size { 32, 64, 128, 256, 512 }	: 512
0. Exit to previous menu	

Enter your choice (option=paramter) -> 0

Diagnostics Configuration Menu

1. Memory Test Configuration Menu
2. Test Configuration Menu
3. Driver Configuration Menu
4. Abort On Failure is enabled
5. Save Configuration

Enter your choice or ESC to exit ->

### 10.181 reset

**Command:** reset

**Description:** Reset Chip

**Syntax:** reset

**Options:**

- c Simulate cold reset
- t Display time from reset to firmware invert signature
- w Wait for firmware signature

### 10.182 quit

**Command:** quit

**Description:** Exit System

**Syntax:** quit

### 10.183 smbus

**Command:** smbus

**Description:** ASF terminal. Run this command to access SMBus, the parameter a1 and a2 are one byte value to specify the NIC SMBus Addresses. It sets to 0 by default.

**Syntax:** smbus [a1 a2]

**Options:**

- s Run in slave mode

### 10.184 cls

**Command:** cls

**Description:** Clear Screen.

**Syntax:** cls

### 10.185 loop

**Command:** loop

**Description:** loop on command.

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

### 10.186 dbmode

**Command:** dbmode

**Description:** Set DEBUG Mode to On or Off Mode.

**Syntax:** dbmode on|off

### 10.187 asf

**Command:** asf

**Description:** run asf monitor program with option to Load asf firmware image. This routine loads firmware images into CPU memory and execute the RXCPU. The default files names are asfinit.bin, asfcgua.bin, and asfcpub.bin, which can be over written by parameters.

**Syntax:** asf [init\_img [rx\_img [tx\_img]]]

**Options:**

- l Load firmware only
- m Enter asf mode only
- w Simulate warm boot

### 10.188 new

**Command:** new

**Description:** Display new command available. The default parameter for [n] is 10

**Syntax:** new [n]

### 10.189 asfprg

**Command:** asfprg

**Description:** Program asf firmware into NVRAM. The default files names are asfinit.bin, asfcgua.bin, and asfcpub.bin, which can be over written by parameters.

**Syntax:** asfprg [init\_img [rx\_img [tx\_img]]]

**Options:**

- v<HEX> verbose level (0,1,2) (def=00000001)

### 10.190 sleep

**Command:** sleep

**Description:** delay execution for specified length of milliseconds. Can be used in script files to delay program execution.

**Syntax:** sleep [milliseconds]

### 10.191 fillpattern

**Command:** fillpattern

**Description:** Fill WOL matching pattern into Misc. Memory Location.

**Syntax:** fillpattern [filename]

**Options:**

- e<HEX> (end address + 1) of the first block (def=00020000)
- f<string> filename which contains data pattern
- o<HEX> sram first block offset to be loaded (def=00000000)
- s<HEX> sram second block offset to be loaded (def=00000000)

### 10.192 inp

**Command:** inp

**Description:** input port (not supported for UEFI)

**Syntax:** inp <addr>

**Options:**

- l long word size
- w word size

### 10.193 outp

**Command:** outp

**Description:** input port (not supported for UEFI)

**Syntax:** outp <addr> <value>

**Options:**

- l long word size
- w word size

### 10.194 switch

**Command:** switch

**Description:** Alternate 'device' command. If no parameter is entered, it will display all device available.

**Syntax:** switch <dev>

**Options:**

- n<HEX> Device Number (def=00000000)
- r Remove all current devices and re-scan available devices
- s Silent mode - do not display devices

### 10.195 do

**Command:** do

**Description:** Excute commands from a file.

**Syntax:** do <filename> [with <parameter1>, ...]

**Options:**

- c continue on error
- e echo command
- p<DEC> pause between each command. If a value is entered, it delays for # of ms (def=0)

### 10.196 txfill

**Command:** txfill

**Description:** Fill tx buffer with pattern and packet length (14-9018).

**Syntax:** txfill [-f=<file>] [-x=<load length>] [-p=<pattern>] [-l=<packet length>]

**Options:**

- f<string> filename

- l<DEC> packet length in bytes (14-9018) (def=1514)
- p<DEC> pattern selection (0-8) (def=0)
- x<DEC> length to load in bytes (default to EOF)

Pattern:

0. Use buffer as is
1. Increment data
2. Random
3. all 0
4. all FF
5. AA55
6. 55AA
7. IP\_Iden-Inc
8. Load from file
9. 8 bytes of 0 and f
10. 16 bytes of 0 and f
11. 32 bytes of 0 and f
12. 64 bytes of 0 and f

### 10.197 wbuf

**Command:** wbuf

**Description:** Write tx|rx buffer with specified data at offset. Only works with static buffer selection -A.

**Syntax:** wbuf tx|rx <offset> <data>

### 10.198 rbuf

**Command:** rbuf

**Description:** Read tx|rx|bistin|bistout|bistex buffer. Read tx|rx|bistin|bistout|bistex buffer with specified at offset with a specified length. Only works with static buffer selection -A.

**Syntax:** rbuf tx|rx|bistin|bistout|bistex <offset> <len>

### 10.199 cpbuf

**Command:** cpbuf

**Description:** Copy the content of rx buffer into tx buffer. Only works with static buffer selection -A.

**Syntax:** cpbuf <offset> <length>

### 10.200 echo

**Command:** echo

**Description:** echo <string> to screen.

**Syntax:** echo <string>

### 10.201 pause

**Command:** pause

**Description:** Pause for user to hit a key. If no parameter is entered, 'press any key to continue...' will be displayed

**Syntax:** pause < message>

### 10.202 q

**Command:** q

**Description:** Exit System

**Syntax:** q

### 10.203 verbose

**Command:** verbose

**Description:** change verbose setting

**Syntax:** verbose

**Options:**

- c toggles CONSOLE
- e toggles ERROR
- i toggles IO
- d toggles DEBUG
- p toggles PRINTER
- w toggles WARNING



- r toggles Interrupt Verbose
- f toggles flush per line to enable dynamic tracking

### 10.204 beep

**Command:** beep

**Description:** Create a beep sound. The default to beep once. If parameter 'n' is entered, it beeps n times

**Syntax:** beep [<n> | on | off]

### 10.205 var

**Command:** var

**Description:** Display current variables

**Syntax:** var

### 10.206 meminfo

**Command:** meminfo

**Description:** report the memory information

**Syntax:** meminfo

### 10.207 delvar

**Command:** delvar

**Description:** Delete local variables

**Syntax:** delvar

### 10.208 regdump

**Command:** regdump

**Description:** Dump register content to a file

**Syntax:** regdump

**Options:**

- c PCI Config Reg.

-f<string> filename  
-m MII Registers  
-r<DEC> Mac Registers (def=1)

### 10.209 regcomp

**Command:** regcomp

**Description:** Compare register content to a file. This command may be used together with regdump to find out any register got changed.

**Syntax:** regcomp <filename>

**Options:**

-c PCI Config Reg.  
-f<string> filename  
-m MII Registers  
-r<DEC> Mac Registers (def=1)

### 10.210 regrestore

**Command:** regrestore

**Description:** Restore register content from a file. This command may be used together with regdump to restore register got changed.

**Syntax:** regrestore <filename>

**Options:**

-c PCI Config Reg.  
-f<string> filename  
-m MII Registers  
-r<DEC> Mac Registers (def=1)

### 10.211 nvsize

**Command:** nvsize

**Description:** Programs the NVRAM and TPM size in Kbyte to NVRAM. If [NVRAM] & [TPM] options are entered, they are programmed to NVRAM. If [NVRAM] & [TPM] options are NOT entered, calculated values are programmed. Use a zero value for [NVRAM] & [TPM] to erase the programmed values.

**Syntax:** nvsize [NVRAM] [TPM]

**Options:**

- d Display Present Size Programmed Into NVRAM
- D Specify [NVRAM] & [TPM] size in decimal (dflt. hex)

## 10.212 aspm

**Command:** aspm

**Description:** Configure the PCI-E Link power state operation at the root complex and the device under test. Power state option described below.

0 ASPM disable

s ASPM L0s enabled

1 ASPM L1 enabled

a ASPM L0s and L1 enabled

**Syntax:** aspm [-e 0|s|1|a] [-r 0|s|1|a] [-x 0|1] [-d] [-i]

**Options:**

- r<string> Configure root complex device
- e<string> Configure endpoint (Broadcom) device (dut)
- x<DEC> Enable ExtendedSync mode for root complex and endpoint (def=0)
- d Display present ASPM State
- i Ignore all previous ASPM setting after a chip reset

## 10.213 dids

**Command:** dids

**Description:** The feature outputs the following information: PCI DID, VID, SDID & SVID; MAC address, Firmware revision, PXE, PXESpd, WOL, ASF, MBA, Bond Rev. This information can be used to verify the setup of a chip after a firmware upgrade.

**Syntax:** dids

### 10.214 serial

**Command:** serial

**Description:** Compares the serial number to defaults or the provided values. This PCI capability testing is only applicable to Shasta C stepping (i.e. BCM5752Cx) family.

**Syntax:** serial <dwordLo> <dwordHi>

### 10.215 power

**Command:** power

**Description:** Verify the power function of PCI capability. This PCI capability testing is only applicable to Shasta C stepping (i.e. BCM5752Cx) family.

**Syntax:** power

### 10.216 readbr

**Command:** readbr

**Description:** Read a 32-bit value from bridge's configuration space register address.

**Syntax:** readbr <register address>

### 10.217 writebr

**Command:** writebr

**Description:** Write a 32-bit value to bridge's configuration space register address.

**Syntax:** writebr <register address> <value>

### 10.218 findbridge

**Command:** findbridge

**Description:** Find all bridges in the system.

**Syntax:** findbridge

### 10.219 bridge

**Command:** bridge

**Description:** Switch to specified bridge.

**Syntax:** bridge

### 10.220 pere

**Command:** pere

**Description:** Enable parity error response on a bridge. Defaults to current bridge.

**Syntax:** pere <bridge>

### 10.221 perd

**Command:** perd

**Description:** Disable parity error response on a bridge. Defaults to current bridge.

**Syntax:** perd <bridge>

### 10.222 peclr

**Command:** peclr

**Description:** Clear parity error on bridge. Defaults to current bridge.

**Syntax:** peclr <bridge>

### 10.223 pechk

**Command:** pechk

**Description:** Check parity error on bridge. Defaults to current bridge.

**Syntax:** pechk <bridge>

### 10.224 iscsiprg

**Command:** iscsiprg

**Description:** Program ISCSI firmware into NVRAM. This command reads ISCSI code from a file and program into NVRAM. There are 2 types of ISCSI firmware image. One only contains ISCSI Boot Firmware. The other also contains ISCSI CFG Block and ISCSI CFG Program. ISCSI Boot Firmware will be programmed by default. ISCSI CFG Block will be programmed if either there is no ISCSI CFG Block present in NVRAM or “-c” option has been entered. ISCSI CFG Program will be programmed only when “-p” option has been entered.

**Syntax:** iscsiprg [-f<filename>] [-c] [-p] [-v | -b]

**Options:**

-f<filename> ISCSI firmware file name.

- c Forced to program ISCSI CFG FW
- p Forced to program ISCSI CFG Program.
- v Support IPv6 protocol
- b Support IPv4 and IPv6 protocols

## 10.225 umpecho

**Command:** umpecho

**Description:** Enable/Disable UMP Echo Test function in UMP Firmware. It requires either ump14a.bin/ump14b.bin test firmware or UMP Firmware. Options that will work with test firmware are '-o', '-c', '-i' and '-a'. Options that will work with UMP firmware are '-e' and '-d'.

**Syntax:** umpecho [-o | -c | -i | -a] | [-e] | [-d]

### Options:

- o Running UMP Echo Test with test firmware
- c Debug display of SRAM address 0xC00
- i Debug display of CPU code loading
- a Debug prompt after CPU code loading
- e Enable New UMP Echo Test in UMP Firmware
- d Disable New UMP Echo Test in UMP Firmware

## 10.226 umpcfg

**Command:** umpcfg

**Description:** Configure UMP in NVRAM

**Syntax:** umpcfg

1	<b>UMP Settings</b>	
Description: Provides the sub menu with the options described below.		
	0	Return to previous menu.
	1	Enable/Disable SetLink
	2	Enable/Disable RDISTallTimer
	3	Set RDISTallTimerValue
	4	DisableHostHashTable
	5	Enable/Disable HostEchoControl

	6	Enable/Disable Exceed_375ma_rule
	7	Link Speed
		0 : Return to previous menu 1 : Speed 10/100 (default) 2 : Speed ALL 3 : Speed 10 4 : Speed 100 5 : Speed 1000 6 : Duplex 7 : Auto/Force Mode 8 : Pause Capability
2		Save and Exit
<b>Description:</b> Saves the modified UMP Configuration Table to the network adapter's non-volatile memory and exists.		
3		Exit without Saving
<b>Description:</b> Exits without saving any changes to the UMP Configuration Table.		

### 10.227 setipmi

**Command:** setipmi

**Description:** Enable/Disable IPMI Passthrough Firmware

**Syntax:** setipmi

**Options:**

- e Enable IPMI Passthrough Firmware
- d Disable IPMI Passthrough Firmware

### 10.228 setump

**Command:** setump

**Description:** Enable/Disable UMP

**Syntax:** setump

**Options:**

- d Disable UMP
- e Enable UMP

### 10.229 seotp

**Command:** seotp

**Description:** Configure OTP

**Syntax:** seotp

**Options:**

-s            Show OTP status  
-e            Update OTP bits  
-f<string>   filename

### 10.230 otpchk

**Command:** otpchk

**Description:** Check patch in OTP

**Syntax:** otpchk

**Options:**

-m            Check MAC and Config  
-l<HEX>      Iteration (def=000000001)  
-f<string>   filename

### 10.231 defragment

**Command:** defragment

**Description:** Defragment NVRAM data.

**Syntax:** defragment

### 10.232 secfgsb1

**Command:** secfgsb1

**Description:** Configure Selfboot NVRAM Group 1

**Syntax:** secfgsb1

**Options:**

-m            Mac address xx:xx:xx:xx:xx:xx  
-v            Vendor ID



- z Vendor Device ID
- s SubSystem Vendor ID
- i SubSystem Device ID
- w Magic Packet WOL 1: Enable 2: Disable
- o Limit WOL Speed to 10 0: No 1: Yes
- l Design Type 0: NIC 1: LOM
- p Enable Phy Auto Power Down 0: No 1: Yes
- r Reversed Nway 0: No 1: Yes
- A Disable Power Saving 0: No 1: Yes
- B Led Mode 0: Mac 1:Phy1 2:Phy2 3:S Traffic 4:Shasta Mac 5:Wireless Combo
- C PCI Power Consumption/Dissipate 1:Default 2:Custom define
- c PCI Power Consumption/Dissipate Value D0:D3/D0:D3
- G Cable Sense 0: No 1: Yes

### 10.233 secfgsb2

**Command:** secfgsb2

**Description:** Configure Selfboot NVRAM Group 2

**Syntax:** secfgsb2

**Options:**

- D PCIE Pwr Consumption/Dissipate 1: Default 2: Custom define
- d PCIE Pwr Consumption/Dissipate Data;  
Data: Pwr Rail: Type: State: Base Power  
Select Data = 0-7  
Pwr Rail = 0:12V; 1: 3.3V; 2: 1.8V; 7: Thermal; 99: Invalidate  
Type = 0: PME 1: Aux 2: Idle 3: Sustained 7: Max  
Pwr Mgt St = 0: D0 1: D1 2: D2 3: D3

Base Power = X in 0.1 Watt

### 10.234 secfgsb3

**Command:** secfgsb3

**Description:** Configure Selfboot NVRAM Group 3

**Syntax:** secfgsb3

**Options:**

-E Product Name 1: Default 2: Custom define

### 10.235 secfghwsb1

**Command:** secfghwsb1

**Description:** Configure Hardware Selfboot NVRAM Group 1

**Syntax:** secfghwsb1

**Options:**

-m Mac address xx:xx:xx:xx:xx:xx

-z Vendor Device ID

-s SubSystem Vendor ID

-i SubSystem Device ID

-w WOL

-o Limit WOL Speed to 10 0: No 1: Yes

-a Wol Auto 0:Dis 1: Magic 2: Interest 3: Magic & Interest

-B Led Mode 0: Mac 1:Phy1 2:Phy2 3:S Traffic

-G Super Airplane Mode 0: No 1: Yes

-H Clkreq 0:No 1:Yes

-I PCIE Link Polarity Fix Dis 0:No 1:Yes

-n Design Type 0:LOM 1:NIC

- I L1 PLL Powerdown Disable 0:No 1:Yes
- J L1 ASPM Debounce 0:Dis 1:En (PCI-E Mobile Devices Only)

### 10.236 secfghwsb2

**Command:** secfghwsb2

**Description:** Configure Hardware Selfboot NVRAM Group 2

**Syntax:** secfghwsb2

**Options:**

- e Engineering Change
- s Serial Number
- m Manufacturing ID
- v Rd VPD Vendor Data V0

### 10.237 secfghwsb3

**Command:** secfghwsb2

**Description:** Configure Hardware Selfboot NVRAM Group 2

**Syntax:** secfghwsb2

**Options:**

- h Hide MBA Setup Prompt 0: Dis 1: En
- k MBA Setup Hot Key 0: Ctrl-S 1: Ctrl-B
- m MBA Boot Protocol 0: PXE 1: RPL 2: BOOTP 3: iSCSI
- v Rd VPD Vendor Data V0
- b MBA Bootstrap Type 0: Auto 1: BBS 2: Int18 3: Int19
- t MBA Delay Time 0-15
- e Vlan 0: Dis 1:En
- i VLAN ID
- s MBA Link Speed 0: Auto 1: 10HD 2: 10FD 3: 100HD 4: 100FD

## 10.238 eswitch

**Command:** eswitch

**Description:** Configure E-Switch on the device that support eswitch, such as 5756 and 5761E.

**Syntax:** eswitch

**Options:**

- d Change the port to Docking mode
- l Change the port to Laptop mode
- e Enable E-switch functionality.
- f Disable E-switch functionality. This option will prevent PHY loopback test, external loopback test (pkttest -e), and carrier test from being performed on both laptop mode and docking mode. By setting this option the tests are performed only on the default port.
- m internal engineering manual switch and MAC test

## 10.239 sbfcfg

**Command:** sadbtest

**Description:** Self-boot file configuration

**Syntax:** sbfcfg <masterfile> [<upgradefile>]

## 10.240 linkintrtest

**Command:** linkintrtest

**Description:** Link Changed Interrupt Test

**Syntax:** linkintrtest

## 10.241 ipv6extld

**Command:** ipv6extld

**Description:** Load IPv6 Extension Header File

**Syntax:** ipv6extld <filename> (default filename is ipv6ext.txt)

### 10.242 ipv6exten

**Command:** ipv6exten

**Description:** Enable IPv6 Extension Headers

**Syntax:** ipv6exten <0 | 1> (0 = disable, 1 = enable)

### 10.243 cfgpagesize

**Command:** cfgpagesize

**Description:** Configure Flash page size to 256 bytes or 512 bytes.

**Syntax:** cfgpagesize -v (verbose)

### 10.244 cpmu

**Command:** cpmu

**Description:** CPMU tests

**Syntax:** cpmu

**Options:**

- t timeout timer test
- p prescaler timer test, -v -p
- v<HEX> value (def = 00000003)
- r<HEX> reference (def = 00020000)

### 10.245 mdio

**Command:** mdio

**Description:** MDIO tests

**Syntax:** mdio

**Options:**

- v<HEX> read MII times (def = 00000200)
- t<HEX> available tests:
  - 1: read MII registers with core clock override.
  - 2: write/read/compare MII register with core clock override.
  - 4: read MII registers without core clock override.

8: write/read/compare MII register without core clock override.  
f: all above.

### 10.246 wait

**Command:** wait

**Description:** Delay for a bit.

**Syntax:** wait

**Options:**

-m<DEC>      wait milli seconds (default=1000)

-u<DEC>      wait micro seconds (default=1000)

### 10.247 ipsecsupport

**Command:** ipsecsupport

**Description:** Check device for IPsec support.

**Syntax:** ipsecsupport

### 10.248 sadbtest

**Command:** sadbtest

**Description:** Invoke the Security Association Database (SADB) test. Valid only for IPsec supported devices.

**Syntax:** sadbtest

### 10.249 sadbutil

**Command:** sadbutil

**Description:** Utilities for the Security Association Database (SADB) table. Valid only for IPsec supported devices.

**Syntax:** sadbutil

**Options:**

-a              Add one SA entry

-d              Delete one SA entry

- e Delete all SA entries
- u Dump one SA entry
- U Dump all SA entries
- t Dump internal IPsec transmitted packets digest log for debug purpose
- r Dump internal IPsec received packets' digest log for debug purpose
- s Dump internal IPsec statistics counters
- c Clear internal IPsec statistics counters

### 10.250 apetest

**Command:** apetest

**Description:** Run APE CPU Self Tests and utility functions. Only valid for APE enabled NetXtreme controllers.

**Syntax:** apetest -t<1-9> or other command option.

**Options:**

- t<1-9> Run APE CPU Self Test
- n<#> Test iterations (0 = loop forever)
- l Load APE diagnostic firmware
- u Unload APE diagnostic firmware
- v Get APE diagnostic firmware version
- r Reset APE diagnostic interface
- d Display APE diagnostic result string
- G<#> Set APE GPIO output pin <0-6>
- g<#> Clear APE GPIO output pin <0-6>

## 11 ERROR MESSAGES

ERROR ID	ERROR NO.	ERROR STRING
NO_ERROR	0	
ERR_IND_REG_ERR	1	Got XXXX @ XXXX. Expected XXXX
ERR_CHIP_RUNNING	2	Cannot perform task while chip is running
ERR_BAD_NIC	3	Invalid NIC device
ERR_READ_ONLY_CLEAR	4	Read only bit X got changed after writing zero at offset XXXX.
ERR_READ_ONLY_SET	5	Read only bit X got changed after writing one at offset XXXX
ERR_READ_WRITE_NOT_CLEAR	6	R/W bit X did not get cleared after writing zero at offset XXXX
ERR_READ_WRITE_NOT_SET	7	R/W bit X did not get set after writing one at offset XXXX
ERR_BIST	8	BIST failed
ERR_INTERRUPT	9	Could not generate interrupt
CMD_ABORT	10	Aborted by user
ERR_DMA_TXDATA	11	Tx DMA:Got XXXX @ XXXX. Expected XXXX
ERR_DMA_RXDATA	12	Rx DMA:Got XXXX @ XXXX. Expected XXXX
ERR_TXDMA	13	Tx DMA failed
ERR_RXDMA	14	Rx DMA failed
ERR_MEM	15	Data error, got XXXX at XXXX, expected XXXX
ERR_MEM2	16	Second read error, got XXXX at XXXX, expected XXXX
ERR_EEP_WRITE	17	Failed writing NVRAM at XXXX
ERR_EEP_READ	18	Failed reading NVRAM at XXXX
ERR_EEP_DATA	19	NVRAM data error, got XXXX at XXXX, expected XXXX
ERR_FILE_OPEN	20	Cannot open file <filename>
ERR_BAD_CPU_CFG	21	Invalid CPU image file <filename>
ERR_IMAGE_SIZE	22	Invalid CPU image size XXXX
ERR_MALLOC	23	Cannot allocate memory for size XXXX
ERR_CPU_RESET	24	Cannot reset Rx   Tx CPU
ERR_CPU_NO_RESP	25	Rx   Tx CPU does not respond
ERR_CPU_TEST	26	Rx   Tx CPU test failed
ERR_DMA_RANGE	27	Invalid Test Address Range Valid NIC address is XXXX- and exclude XXXX-XXXX
ERR_DMA_DATA	28	DMA:Got XXXX @ XXXX. Expected XXXX SRAM data=XXXX @ XXXX
ERR_PHY_ID	29	Unsupported PhyId XXXX:XXXX
ERR_PHY_TOO_MANY_REG	30	Too many registers specified in the file, max is XXXX
ERR_VPD_WRITE	31	Cannot write to VPD address XXXX
ERR_VPD_DATA	32	VPD data error, got XXXX @ XXXX, expected XXXX
ERR_NO_LINK	33	No good link! Check Loopback plug
ERR_DATA_TX	34	Cannot TX Packet!
ERR_DATA_TX_MISSING	35	Requested to Tx XXXX. Only XXXX is transmitted
ERR_DATA_RX_MISSING	36	Expected XXXX packets. Only XXXX good packet(s) have been received. XXXX unknown packets have been received. XXXX packets have been missing.
ERR_INVALID_TEST	37	XXXX is an invalid Test
ERR_EEPROM_CHECKSUM	38	NVRAM checksum error
ERR_READING_WOL_PXE	39	Error in reading WOL/PXE
ERR_READING_WOL_PXE	40	Error in writing WOL/PXE
ERR_NO_EXT_SRAM	41	No external memory detected
ERR_DMA_LEN	42	DMA buffer XXXX is too large, size must be less than XXXX
ERR_FILE_TOO_BIG	43	File size XXXX is too big, available space is XXXX
ERR_INVALID	44	Invalid <string>
ERR_WRITE	45	Failed writing XXXX to XXXX
CMD_QUIT	46	
ERR_CPU_MEM_ERR	47	Rx   Tx CPU access error @ XXXX, expected XXXX but got XXXX
ERR_ENDIF	48	
ERR_ROM_D_DATA	49	ROM disable error, data returned while disabled
ERR_CHIP_NOT_RUNNING	50	Cannot perform task while chip is not running. (need driver)
ERR_NO_REG_DEF	51	Cannot open register define file or content is bad
ERR_ASF_RST	52	ASF Reset bit did not self-cleared
ERR_ASF_ATTN_LOC	53	ATTN_LOC XXXX cannot be mapped to Rx   Tx CPU event bit XXXX
ERR_ASF_RST_VAL	54	Register is not cleared to zero after reset
ERR_ASF_PA_TIMER	55	Cannot start poll_ASF Timer
ERR_ASF_PA_CLEAR	56	poll_ASF bit did not get reset after acknowledged
ERR_ADF_NO_STAMP	57	Timestamp Counter is not counting
ERR_ADF_NO_TIMER	58	Timer is not working
ERR_ASF_EVENT	59	Cannot clear bit X in Rx   Tx CPU event register
ERR_EEP_FILESIZE	60	Invalid file size, expected XXXX but only can read XXXX bytes
ERR_MAGIC_VALUE	61	Invalid magic value in XXXX, expected XXXX but found XXXX
ERR_EEP_FMT	62	Invalid manufacture revision, expected X but found X



ERR_EEP_BOOTVER	63 Invalid Boot Code revision, expected XXXX.XXXX but found XXXX.XXXX
ERR_EEP_CANNOT_WRITE	64 Cannot write to NVRAM
ERR_EEP_CANNOT_READ	65 Cannot read from NVRAM
ERR_BAD_CHECKSUM	66 Invalid Checksum
ERR_BAD_MAGIC_VALUE	67 Invalid Magic Value
ERR_MAC	68 Invalid MAC address, expected XX-XX-XX-XX-XX-XX
ERR_BUS	69 Slot error, expected an UUT to be found at location XX:XX:00
ERR_SPEC_MEM	70 Adjacent memory has been corrupted while testing block XXXX-XXXX Got XXXX @ address XXXX. Expected XXXX
ERR_NOT_SUPPORT	71 The function is not Supported in this chip
ERR_BAD_CRC	72 Packets received with CRC error
ERR_MII_ERR_BITS_SET	73 MII error bits set: XXXX
ERR_INIT_MAC	74 CPU does not initialize MAC address register correctly
ERR_FW_FILE_FORMAT	75 Invalid firmware file format
ERR_RESET_TX_CPU	76 Resetting TX CPU Failed
ERR_RESET_RX_CPU	77 Resetting RX CPU Failed
ERR_INVALID_MAC_ADDR	78 Invalid MAC address
ERR_MAC_REG	79 Mac address registers are not initialized correctly
ERR_BOOTCHECKSUM	80 NVRAM Bootstrap checksum error
ERR_VPD_READONLY	81 Write operation changed VPD read only data from XXXX to XXXX at XXXX
ERR_VPD_READ	82 Cannot read data from VPD address XXXX
ERR_MEM_READ	83 Memory read and compare error
ERR_MEM_WRITE	84 Memory write error (no longer in use)
ERR_PXE_PGM	85 PXE Programming Error
ERR_PXE_VFY	86 PXE Verification Error
ERR_EXT_MEM_EXE_TIMEOUT	87 Cannot execute code from external memory, pc=XXXX
ERR_EXT_MEM_SIZE	88 External memory size detection error
ERR_RESET_TIMEOUT	89 Reset Time
ERR_MSI_ERR_NOTCLEAR	90 MSI Error bits are not cleared after reset
ERR_MSI_DATA	91 MSI expected XXXX, but read XXXX at XXXX
ERR_MEM_INIT	92 mem pool initialization failed
ERR_MEM_UNINIT	93 mem pool un-initialization failed
ERR_PCI_REGS_WIDTH	94 Read/Write PCI regs width XXXX affects wider than expected at offset XXXX
ERR_LINK_STATUS	95 Link status error in auto-polling mode
ERR_PHY_INTERRUPT	96 Phy interrupt did not happen
ERR_EEP_BIT_BANG	97 EEprom test fails in bit-bang mode at address XXXXXXXX
ERR_ROM_SIZE	98 ROM size error. Expected XXXX but read XXXX at ROM Bar (0x30) register with XXXX written to ROM size reg.(0x88)
ERR_ROM_DATA	99 Data Error. Expected XXXX but read XXXX at XXXX
ERR_ROM_ENABLE	100 Expansion ROM Desired bit is not set after loading firmware
ERR_GPIO	101 GPIOXX Error, write=XXXX, read=XXXX
ERR_GPIO5704	102 Dev:XXXX Expected GPIO 0/1/2 = XXXX/XXXX/XXXX, but read as XXXX/XXXX/XXXX
ERR_BIST_NOT_DONE	103 Bist test did not complete internally
ERR_BIST_DATA_MISCOMP	104 Bist data miscompared at bit: XXXX out: XXXX exp: XXXX
ERR_CPU_NO_RESPONSE	105 No Response from firmware
ERR_CPU_ERR_CODE	106 Rx   Tx CPU returned result XXXX, key = XXXX
CMD_LOOP	107
CMD_SKIP	108
CMD_ELSE	109
CMD_ELSEIF	110
CMD_BREAK	111
CMD_ENDWHILE	112
ERR_BYTE	113 Byte access error: expected XX at XXXX but got XX
ERR_WORD	114 Wrđ acc err: exp XXXX at XXXX but got XXXX (need flshdg5x.bin v2.3 or newer)
ERR_NO_LINK_DOWN	115 No link down found
ERR_MISMATCHED_DEVICEID	116 bootcode Image file belongs to XXXX family, does not match with board XXXX
ERR_INVALID_DATA_SIZE	117 Invalid data size
ERR_MAC_ADDR_ENDED	118 Runs out of Mac Address
ERR_ILLEGAL_MAC_ADDR	119 Illegal Mac Address
ERR_BIST_DATA_INVALID	120 Invalid bist data from buffer at XXXX
ERR_INVALID_BOND_ID	121 Invalid bond id XXXX
ERR_BAD_CPU_RESET	122 CPU reset failed, register 5034 is XXXX
ERR_INCORRECT_VERSION	123 Incorrect version
ERR_MISMATCH_CFG_VERSION	124 Mismatched CFG and FW Image version
ERR_NOT_SUPP_CFG_BW	125 Current ASF_T_VERSION is not backward compatible
ERR_POST_1G_LB	126 1G Tx/Rx Lines Have A Short/Open

ERR_DRIVER	127	Unsupported driver version VX.XX (need bcm57diag v9.0.0 or tg3d 3.661 or newer)
ERR_TXDMA_OVERFLOW	128	TxDMA Overflow
ERR_RXDMA_OVERFLOW	129	RxDMA Overflow
ERR_DRIVER_BAD_STATUS	130	Driver returned error status=XXXX on ioctl=X
ERR_INVALID_HANDLE	131	Invalid Handle
ERR_SOCKET	132	Socket open error XXXX
ERR_SIOCGIFINDEX	133	ioctl(): SIOCGIFINDEX failed
ERR_BIND	134	bind() failed:
ERR_SETSOCKOPT	135	setsockopt() failed:XXXX
ERR_FCNTL	136	fcntl() failed:XXXX
ERR_SEND	137	send() failed (cnt=XXXX): XXXX
ERR_RECEIVE	138	recvfrom() failed (cnt=XXXX): XXXX
ERR_FALSE_CARRIER	139	Error! False Carrier detected during the test
ERR_INV_OPT	140	Invalid Options
ERR_INV_DEV	141	Found Rv = XXXX, Expected Rv = XXXX due to -ckdev value
ERR_IRQ	142	Invalid IRQ XX
ERR_TIMEOUT	143	Timeout
ERR_PKT_DATA	144	Packet data error at offset XXXX, expected XX but received XX
ERR_MAC_ZERO	145	Zero Mac Address in Mac Register
ERR_MAC_MIS_REG	146	Mac Address MisMatch: Got XX-XX-XX-XX-XX-XX.
ERR_CPU_ACC	147	Rx   Tx CPU failed on XXXX bit access to address XXXX
ERR_CPU_MEM	148	Rx   Tx CPU failed on memory pattern XXXX at address XXXX
ERR_CPU_INSTR	149	Rx   Tx CPU Instruction test failed
ERR_BUS_LINK	150	PCI-E Bus Link Error
ERR_ASF_FILE_VER	151	ASF File versions Error
ERR_ASF_VS_DEV	152	ASF VS Device Error
ERR_MISMATCHED_DEVICEID_1	153	bootcode Image file belongs to unknown family, does not match with board XXXX
ERR_MISMATCHED_DEVICEID_2	154	bootcode Image file belongs to XXXX family, does not match with unknown board
ERR_CPU_FETCH	155	CPU Fetch Test Error: Breakpoint reads XXXX
ERR_UMPLB	156	UMP loopback failed, Total Good packet XXXX
ERR_UMPCTRL	157	Error: UMPCtrl 0x5F0 = XXXX
ERR_REG_TABLE	158	Error: Unable to create Reg Table
ERR_FW_IMAGE_SIZE	159	Error: Firmware image size (XXXX) larger than NVRAM size (XXXX)
ERR_MISS_PXE	160	Error: PXE firmware cannot be found in NVRAM
ERR_MISS_ASF	161	Error: ASF firmware cannot be found in NVRAM
ERR_NVRAM_DIR	162	Error: NVRAM Directory
ERR_MISS_UMP	163	Error: UMP firmware cannot be found in NVRAM
ERR_CPU_MEM2	164	Rx   Tx CPU memory error @ XXXX, expected XXXX but got XXXX
ERR_SERIAL1	165	PCIE Serial Device Number Capability Not Found.
ERR_SERIAL2	166	PCIE Serial Device Number Match Failure.
ERR_POWER1	167	PCIE Power Budget Capability Not Found.
ERR_POWER2	168	PCIE Power Budget Error (ad=XXXX, saw=XXXX, exp=XXXX).
ERR_UMP_VS_DEV	169	UMP VS Device Error
ERR_HM_BR_LANES	170	M Bridge Lanes Error: XXXX PCI-E lanes are actually linked and running, expected XXXX.
ERR_BRIDGE	171	Invalid bridge.
ERR_NO_LINK10	172	No good link at 10 Mbits/s! Check Loopback plug
ERR_NO_LINK100	173	No good link at 100 Mbits/s! Check Loopback plug
ERR_NO_LINK1000	174	No good link at 1000 Mbits/s! Check Loopback plug
ERR_DMA_SHASTA	175	Failed! Saw=XXXX @ XXXX, Exp=XXXX from XXXX
ERR_SECFG_CONT	176	SECFG Config Error. Line XXXX
ERR_SECFG_INVALID	177	Invalid Argument in SECFG Config File at Line XXXX
ERR_NV_SELFBOOT	178	NVRAM is SELFBOOT
ERR_PART_NUM	179	Error Part Number Check Failed
ERR_UMPECHO	180	UMP Echo - Debug Mode
ERR_IMAGE_CHECKSUM	181	Firmware image checksum error
ERR_INVALID_SMB	182	Invalid SMB Address
ERR_LOM_CFG	183	LOM CONF Bit mismatch in Reg 178 and Reg 7C04
ERR_ISCSI_PGM	184	iSCSI Programming Error
ERR_BAD_CHIP_REV	185	Invalid Chip Rev.
ERR_NO_CPU	186	Device has no internal CPU
ERR_BAD_PARITY_VALUE	187	Invalid Parity.
ERR_BAD_VPD_CHKSUM	188	Invalid VPD Checksum
ERR_ESWITCH_CTRL_TO	189	Halting E Switch Ctrl time out. XXXX
ERR_ESWITCH_EVENT	190	ESWITCH event not set. XXXX
ERR_ESWITCH_TO	191	E Switch time out. XXXX
ERR_ESWITCH_ARBIT_TO	192	Halting E Switch Ctrl Arbitor time out. XXXX
ERR_ESWITCH_INTR	193	ESWITCH interrupt not set. XXXX
ERR_ESWITCH_STATUS	194	ESWITCH status word not set. XXXX

ERR_SADB_ENTRY	195 SADB Entry error at SA Indx XXXX with pattern <string>
ERR_SADB_CORRUPT	196 SADB Data corrupted: Got XXXX at SADB index XXXX word index XXXX. Expected = XXXX
ERR_SADB_DEL_ALL	197 SADB delete all error
ERR_SADB_DEL	198 SADB delete SA index XXXX error
ERR_SADB_INTR	199 SADB interrupt not set. <string> index XXXX
ERR_SADB_ENTRY_TEST	200 SADB Entry Test error at SA Indx XXXX
ERR_SHAL_NOT_SUPPORTED	201 SHA1 not supported
ERR_DIR_CHECKSUM,	202 Directory Checksum Error
ERR_APE_REGISTER_RANGE,	203 Invalid APE register range
ERR_APE_Mutex_RELEASE,	204 Can not release APE Mutex Grant register
ERR_APE_Mutex_REQUEST_GRANT	205 APE Mutex register was not granted (Mutex registers 0xXXXX=XXXX; 0xXXXX=XXXX).
ERR_APE_TIMER_RESET	206 APE Timer reset value error, XXXX.
ERR_APE_TIMER_RUN	207 APE Timer counting error, XXXX.
ERR_APECPU_LOAD	208 APE firmware load error: XXXX
ERR_APECPU_TIMEOUT	209 APE CPU test timed out: XXXX
ERR_APECPU_MEM	210 APE CPU Memory Test error: XXXX
ERR_APECPU_ETH	211 APE CPU Ethernet Test error: XXXX
ERR_APECPU_SMB	212 APE CPU SMBus Test error: XXXX
ERR_APECPU_GPIO	213 APE CPU GPIO Test error: XXXX
ERR_APECPU_EVENT	214 APE CPU Event Test error: XXXX
ERR_APECPU_Mutex	215 APE CPU Mutex Test error: XXXX
ERR_APECPU_TIMER	216 APE CPU Timer Test error: XXXX
ERR_APECPU_GRC	217 APE CPU GRC Test error: XXXX
ERR_POST_100_LB	218 Tx/Rx Lines Have A Short/Open
ERR_DATA_TX_MISSING_IPSEC	219 Requested Tx of XXXX IPSEC Pkt(s) with Cipher <string>. Only XXXX transmitted
ERR_DATA_RX_MISSING_IPSEC	220 Expected Rx of XXXX IPSEC Pkt(s) with Cipher <string>. Only XXXX received
ERR_RX_HW_CRYPTODONE_CNT	221 Got HW_CRYPTODONE_CNT = XXXX, Expected XXXX
ERR_USER_BLOCK_NOT_AVAILABLE	222 User block not available
ERR_CPU_Mutex_ERROR	223 Rx   Tx CPU test APE Mutex error, test XXXX
ERR_CPU_REG_ERROR	224 Rx   Tx CPU test APE reg error, test XXXX
ERR_CPU_BIN_FILE_MISMATCH	225 Soledad Rev B0 and up should use cpusd.bin version 4.0 or up.
ERR_USB_NOT_DETECTED	226 USB device was not detected. Please check cable connection.
ERR_USB_DATA_ERROR	227 USB device error during data transfer.
ERR_SELFBOOTII_BAD_ECC,	228 SelfbootII CPD #XXXX ECC got corrupted.
ERR_SELFBOOTII_BAD_CPD,	229 SelfbootII CPD #XXXX got corrupted.
ERR_SELFBOOTII_FULL,	230 OTP is full.
ERR_SELFBOOTII_EMPTY,	231 OTP is empty.
ERR_SELFBOOTII_OTP_CORRUPTED	232 Dismatched when compared to file <filename>