



Broadcom NetXtreme Ethernet Adapter Diagnostic User's Guide

B57diag 5.02 · Date 10/23/02

**Prepared by: Tak Tomita
Updated by: Paul Nguyen
Updated by: Carlton Joseph**

Copyright © 2000, 2001 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
16125 Alton Parkway
Irvine, CA 92619-7013
www.broadcom.com

TABLE OF CONTENTS

1	INTRODUCTION	1
2	PREREQUISITES	2
3	DIAGNOSTIC TESTS	3
3.1	TEST NAMES	3
3.2	ERROR CODES.....	3
3.3	TEST DESCRIPTIONS.....	6
3.3.1	A1. Indirect Register Test	6
3.3.2	A2. Control Register Test	6
3.3.3	A3. Interrupt Test	7
3.3.4	A4. BIST.....	7
3.3.6	B1. Scratch Pad Test	7
3.3.7	B2. BD SRAM Test.....	8
3.3.8	B3. DMA SRAM Test.....	8
3.3.9	B4. MBUF SRAM Test	8
3.3.10	B5. MBUF SRAM via DMA Test	8
3.3.11	B6. External SRAM Test.....	9
3.3.12	C1. EEPROM Test.....	9
3.3.13	C2. CPU Test.....	9
3.3.14	C3. DMA Test.....	9
3.3.15	C4. MII Test.....	10
3.3.16	C5. VPD Test.....	10
3.3.17	C6. ASF Test.....	11
3.3.19	D1. Mac Loopback Test	11
3.3.20	D2. Phy Loopback Test	11
3.3.21	D3. RJ45 Loopback Test.....	12
3.3.23	D5. MSI Test.....	12
4	COMMAND LINE OPTION PARAMETERS	13
5	EEPROM.TXT FORMAT	21
6	USER INTERFACE COMMANDS.....	24
7	SPECIAL INSTRUCTION	29
8	TEST AND FUNCTIONS DESCRIPTION.....	30
8.1	VPDWRITE.....	30
8.2	VPDREAD	30
8.3	SEMODE.....	30
8.4	SERead.....	31
8.5	SEWRITE	31
8.6	SECFG.....	32
8.7	SEPRG.....	33
8.8	UPGFRM.....	34
8.9	SEVER.....	34
8.10	SECHKSUM	34
8.11	SEDUMP	35

8.12	ASFCFG.....	35
8.13	FLSHMODE	35
8.14	FLSHREAD	35
8.15	CHKPXE.....	35
8.16	DIR	36
8.17	ERASE	36
8.18	PXEPRG	36
8.19	FLSHDEV.....	36
8.20	BITBANG	36
8.21	SECLOCK	37
8.22	SETORTURE	37
8.23	SEINIT	37
8.24	SEARB	37
8.25	SEPROTECT	38
8.26	SELCLOCK.....	38
8.27	SEMAP	38
8.28	SETWOL	38
8.29	SETPXE.....	38
8.30	SETASF.....	39
8.31	SECOMP	39
8.32	DRESET.....	39
8.33	USERBLOCK	40
8.34	SETMBA.....	40
8.35	LOADFW	40
8.36	CPUDRT	41
8.37	CPUDTT	41
8.38	CPUTRACE	42
8.39	HALTCPU.....	42
8.40	LOADBOOTCODE	42
8.41	DISASM	42
8.42	STEP	43
8.43	GO.....	43
8.44	SHOWGPR.....	43
8.45	PC.....	43
8.46	SELECT	43
8.47	BREAKPOINT	43
8.48	U.....	44
8.49	TRAP	44
8.50	CPUINFO.....	44
8.51	CPUGPRSTATUS	44
8.52	CPUGPRRESTART	44
8.53	DMAW	44
8.54	DMAR.....	46
8.55	DMA_H.....	47
8.56	DMA_D.....	47
8.57	MACLPK.....	48
8.58	NICSTATS.....	48
8.59	RINGINDEX	50
8.60	BLAST	51
8.61	PHYCTRL	53
8.62	TXPKT	54
8.63	STATUSBLK	55
8.64	STSBLK	56
8.65	TXCFG.....	57

8.66	RXCFG.....	58
8.67	TPROT	58
8.68	QSTAT	59
8.69	DRVRCFG.....	59
8.70	IRT	59
8.71	MACMRD	60
8.72	MIIMRD	60
8.73	PCIMRD	60
8.74	MWRITE.....	61
8.75	MREAD	61
8.76	MDEV.....	61
8.77	MIIMODE.....	62
8.78	LBERTRAM	62
8.79	DBERTRAM	62
8.80	BERTSTATS.....	63
8.81	RM.....	63
8.82	MRLOOP	63
8.83	PHYMSE.....	63
8.84	INITPHY	64
8.85	MEMSEARCH.....	64
8.86	READ.....	64
8.87	WRITE	65
8.88	POLL	66
8.89	SETBIT	67
8.90	CLEARBIT	67
8.91	READBIT	68
8.92	CREAD	69
8.93	CWRITE.....	69
8.94	VPDTEST	69
8.95	REGTEST	69
8.96	MIITEST	70
8.97	MSI	70
8.98	MEMTEST	70
8.99	SETEST	71
8.100	BIST	71
8.101	NICTEST	72
8.102	INTRTEST	72
8.103	PKTTEST	72
8.104	CPUTEST	73
8.105	DMATEST	73
8.106	TESTE	74
8.107	TESTD.....	74
8.108	ASFTEST	74
8.109	BUSTEST	74
8.110	SRAMTEST	76
8.111	MSITEST	76
8.112	ROMTEST	77
8.113	GPIOTEST	77
8.114	CPUDIAG.....	77
8.115	PCICFGTEST	78
8.116	PETEST	78
8.117	ERRCTRL	78
8.118	SEDVT	79
8.119	MIIMISCTEST	79

8.120	CPUGPRTEST	79
8.121	PMDCFG.....	80
8.122	PMPD	80
8.123	INTR.....	81
8.124	INTRCTRL	81
8.125	INTT.....	81
8.126	MBUF.....	81
8.127	LOADDRV	82
8.128	UNLOADDRV.....	83
8.129	MACHALT	83
8.130	FTQ.....	83
8.131	ADDMC.....	84
8.132	DELMC.....	84
8.133	TXMACDES.....	84
8.134	TXMACSRC.....	84
8.135	CHKLDDRV.....	85
8.136	VLANTAG	85
8.137	REG.....	85
8.138	DEBUG.....	85
8.139	GPIOWRITE	86
8.140	GPIOREAD	86
8.141	PXECPY	86
8.142	DEVICE	87
8.143	VERSION.....	87
8.144	HELP	87
8.145	?	87
8.146	RADIX	88
8.147	NOLOG	88
8.148	LOG.....	88
8.149	PCIINIT	88
8.150	PCISCAN.....	88
8.151	DOS.....	89
8.152	DIAGCFG.....	89
8.153	RESET	92
8.154	SMBUS	92
8.155	CLS.....	92
8.156	LOOP	92
8.157	DBMODE	92
8.158	ASF.....	93
8.159	NEW	93
8.160	ASFPRG.....	93
8.161	SLEEP	93
8.162	FILLPATTERN	94
8.163	INP.....	94
8.164	OUTP.....	94
8.165	SWITCH.....	95
8.166	DO.....	95
8.167	TXFILL.....	95
8.168	WBUF	96
8.169	RBUF.....	96
8.170	CPBUF	96
8.171	ECHO	96
8.172	PAUSE	97
8.173	VERBOSE	97

8.174	BEEP.....	97
8.175	VAR.....	97
8.176	MEMINFO	98
8.177	DELVAR	98
8.178	REGDUMP	98
8.179	REGCOMP	98
8.180	REGRESTORE	99
8.181	READBR	99
8.182	WRITEBR.....	99
8.183	WRITEBR.....	99
8.184	FINDBRIDGE	100
8.185	BRIDGE.....	100
8.186	PERE	100
8.187	PERD	100
8.188	PECLR.....	100
8.189	PECHK.....	100
8.190	EXIT	101
8.191	QUIT	101
9	ERROR MESSAGES	102

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 BCM5700-5705 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.

OS: Dos 6.22 or Win95/98 DOS.

Software: b57diag.exe

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

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

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

cpu.bin or cpu05.bin (CPU Instruction test)

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

flshdiag.bin or flashdg05.bin

config.sys

himem.sys

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', and 'D'.

3.1 Test Names

Group A.

- A1. Indirect Register Test
- A2. Control Register Test
- A3. Interrupt Test
- A4. BIST
- A5. PCI Cfg 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

Group C.

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

Group D.

- D1. Mac Loopback Test
- D2. Phy Loopback Test
- D3. RJ45 Loopback Test
- D4. MII Miscellaneous Test
- D5. MSI Test

3.2 Error Codes

Code	Message
------	---------

- | | |
|---|--|
| 1 | Got 0x%08X @ 0x%08X. Expected 0x%08X |
| 2 | Cannot perform task while chip is running |
| 3 | Invalid NIC device |
| 4 | Read only bit %s got changed after writing zero at offset 0x%X |
| 5 | Read only bit %s got changed after writing one at offset 0x%X |

6 Read/Write bit %s did not get cleared after writing zero at offset 0x%X
7 Read/Write bit %s did not get set after writing one at offset 0x%X
8 BIST failed
9 Could not generate interrupt
10 Aborted by user
11 Tx DMA:Got 0x%08X @ 0x%08X. Expected 0x%08X
12 Rx DMA:Got 0x%08X @ 0x%08X. Expected 0x%08X
13 Tx DMA failed
14 Rx DMA failed
15 Data error, got 0x%08X at 0x%08X, expected 0x%08X
16 Second read error, got 0x%08X at 0x%08X, expected 0x%08X
17 Failed writing NVRAM at 0x%04X
18 Failed reading NVRAM at 0x%04X
19 NVRAM data error, got 0x%08X at 0x%04X, expected 0x%08X
20 Cannot open file %s
21 Invalid CPU image file %s
22 Invalid CPU image size %d
23 Cannot allocate memory for size %d
24 Cannot reset %cX CPU
25 /* no longer in use --- Available */
26 %cx CPU test failed
27 Invalid Test Address Range\nValid NIC address is 0x%08X-0x%08X and exclude
0x%08X-0x%08X
28 DMA:Got 0x%08X @ 0x%08X. Expected 0x%08X
29 Unsupported PhyId %04X:%04X
30 Too many registers specified in the file, max is %d
31 Cannot write to VPD address %04X
32 VPD data error, got %08X @ 0x%04X, expected %08X
33 No good link! Check Loopback plug
34 Cannot TX Packet!
35 Requested to Tx %d. Only %d is transmitted
36 Expected %d packets. Only %d good packet(s) have been received\n%d unknown packets
have been received.\n%d bad packets have been received.
37 %c%d is an invalid Test
38 NVRAM checksum error
39 Error in reading WOL/PXE
40 Error in writing WOL/PXE
41 No external memory detected
42 DMA buffer %04X is too large, size must be less than %04X
43 File size %d is too big, max is %d
44 Invalid %s
45 Failed writing 0x%x to 0x%x
46
47 %s CPU access error @ %08X, expected %08X but got %08X
48
49 ROM disable error, data returned while disabled
50 Cannot perform task while chip is not running. (need driver)
51 Cannot open register define file or content is bad

- 52 ASF Reset bit did not self-cleared
- 53 ATTN_LOC %d cannot be mapped to %cX CPU event bit %d
- 54 %s Register is not cleared to zero after reset
- 55 Cannot start poll_ASF Timer
- 56 poll_ASF bit did not get reset after acknowledged
- 57 Timestamp Counter is not counting
- 58 %s Timer is not working
- 59 Cannot clear bit %s in %cX CPU event register
- 60 Invalid %s file size, expected %d but only can read %d bytes
- 61 Invalid magic value in %s, expected %08x but found %08x
- 62 Invalid manufacture revision, expected %c but found %c
- 63 Invalid Boot Code revision, expected %d.%d but found %d.%d
- 64 Cannot write to NVRAM
- 65 Cannot read from NVRAM
- 66 Invalid Checksum
- 67 Invalid Magic Value
- 68 Invalid MAC address, expected %02X-%02X-%02X-%02X-%02X-%02X
- 69 Slot error, expected an UUT to be found at location %02X:%02X:00
- 70 Adjacent memory has been corrupted while testing block 0x%08x-0x%08x\nGot 0x%08x @
address 0x%08x. Expected 0x%08x
- 71 The function is not Supported in this chip
- 72 Packets received with CRC error
- 73 MII error bits set: %04x
- 74 CPU does not initialize MAC address register correctly
- 75 Invalid firmware file format
- 76 Resetting TX CPU Failed
- 77 Resetting RX CPU Failed
- 78 Invalid MAC address
- 79 Mac address registers are not initialized correctly
- 80 NVRAM Bootstrap checksum error
- 81 Write operation changed VPD read only data from %08X to %08X at %04X
- 82 Cannot read data from VPD address %04X
- 83 Memory read and compare error
- 84 Memory write error /* no longer in use */
- 85 PXE Programming Error
- 86 PXE Verification Error
- 87 Cannot execute code from external memory, pc=%08X
- 88 External memory size detection error
- 89 Reset Time
- 90 MSI Error bits are not cleared after reset
- 91 MSI expected %04X, but read %04X at %08X
- 92 mem pool initialization failed
- 93 mem pool un-initialization failed
- 94 Read/Write PCI regs width %d affects wider than expected at offset 0x%X
- 95 Link status error in auto-polling mode
- 96 Phy interrupt did not happen
- 97 EEprom test fails in bit-bang mode at address %X

98 ROM size error\nExpected %08X but read %08X at\nROM Bar (0x30) register with %d
written to ROM size reg.(0x88)
99 Data Error\nExpected %08X but read %08X at %08X
100 Expansion ROM Desired bit is not set after loading firmware
101 GPIO%d Error, write=%d, read=%d
102 Dev:%d Expected GPIO 0/1/2 = %d/%d/%d, but read as %d/%d/%d
103 Bist test did not complete internally
104 Bist data miscompared at bit: %d out: %d exp: %d
105 No Response from firmware
106 %s CPU returned result %d, key = %d
107
108
109
110
111
112
113 Byte access error: expected %02x at %08x but got %02x
114 Word access error: expected %04x at %08x but got %04x
115 No link down found
116 bootcode Image file belongs to %d family, does not match with board %d
117 Invalid data size
118 Runs out of Mac Address
119 Illegal Mac Address
120 Invalid bist data from buffer at %d
121 Invalid bond id

*Blank lines indicate an unused error code.

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: Test Enabled

3.3.2 A2. Control Register Test

Command: regtest

Function: Each Register specified in the configuration contents read only bit and read/write bit defines. The test writing zero and one into the test bits to insure the read only bits are not changed, and read/write bits are changed accordingly.

Default: Test 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: Due to the intermittent failure, this test is currently disabled by default

3.3.5 A5. PCI Cfg Register Test

Command: pcicfg

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

3.3.6 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.

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 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

3.3.7 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

3.3.8 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

3.3.9 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

3.3.10 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

3.3.11 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

3.3.12 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

3.3.13 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

3.3.14 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

3.3.15 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: Test 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 exist, 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

3.3.16 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: Disabled

3.3.17 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_ATTN bit. By changing ASF_ATTN 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.

3.3.18 C7. ROM Expansion Test

Command: romtest

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

3.3.19 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

3.3.20 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.

Default: Enabled

3.3.21 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.

Default: Disabled

3.3.22 D4. 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

3.3.23 D5. MSI Test

Command: msitest

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

Default: Disabled

4 Command line option parameters

When user 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.

-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 Mb/s respectively. The default value is 1000 Mb/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 Mb/s respectively. The default value is 1000 Mb/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 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. The 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 `eprom.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 `eprom.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.

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

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

Default Setting:

Enabled Tests:

- A1. Indirect Register Test
- A2. Control Register Test
- A3. BIST
- A4. Interrupt 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 Test
- C7. ROM Expansion Test

D1. Mac Loopback Test
D2. Phy Loopback Test
D4. MII Miscellaneous Test
D5. MSI Test

Disabled Tests:

B6. External SRAM Test
D3. RJ45 Loopback Test

-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, 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 5.0 EEPROM.TXT format for detailed argument description.

-bus <num> Test UUT location

If this option is specified, the program will only test all the UUTs at the specified bus number. This option is ignored if -c option is entered.

-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 before starts testing. Prompt user for the MAC address. With this option, user must enter/scan the MAC address before testing. The program also checks for the file "mac_pref.txt" This is a text file should contain six digits of ASCII MAC three-byte-prefix address. Any of the following formats are supported:

Example:

001018

00 10 18

00

10

18

If this file exists, user has option to enter/scan 7 digit hex number. The first digit will be ignored and the last 6 digits will be used for the lower part of MAC address. Combine with the prefix, it creates 6 byte (12 digit) hex number. If this file does not exist, the whole 12-digit number must be entered for the MAC address. For readability, when entering MAC address, a space character is allowed between each byte. For example, any of the following examples are valid.

001018010203

00 10 18 01 02 03

1010203 (currently, the scanner uses this format)

-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. If this option is entered, 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.

-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
-com <value> : enable com port, value(1..4)
-errctrl <c> : Action on Error --> a:abort, w:wait, l:loop, c:cont.
-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

5 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.

MAC_PREFIX	= xx:xx:xx
POWER DISSIPATED	= d:d:d:d
POWER_CONSUMED	= d:d:d:d
SUBSYSTEM_VENDOR_ID	= xxxx
SUBSYSTEM_DEVICE_ID	= xxxx
PXE	= {enable, disable}
PXE_SPEED	= {auto, 10hd, 10fd, 100hd, 100fd}
WOL	= {enable, disable}
PRODUCT_NAME	= string (48)
PART_NUMBER	= string (16)
ENGINEERING_CHANGE	= string (10)
MANUFACTURING_ID	= string (4)
ASSET_TAG	= string (16)
VOLTAGE_SOURCE	= {1.3, 1.8}

FORCE_PCI	= {enable, disable}
PART_REVISION	= cc
LED_MODE	= {Triple_link, link_speed} or {phy_mode1, phy_mode2}
PHY_TYPE	= {Copper, Fiber}
PHY_ID	= xxxxxxxx
MAX_PCI_RETRY	= {0..7, auto}
ASF	= {enable, disable}
DUAL_MAC_MODE	= {normal, mac0, mac1, xbar}
	normal: Ch.0 and Ch. 1 enabled
	mac0: Ch.0 enabled, Ch.1 disabled
	mac0: Ch.0 disabled, Ch.1 enabled
	xbar: Both MACs shares one function in PCI configuration space
MBA_BOOT_PROTOCOL	= { pxe, rpl, bootp}
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}
MAC	= {0, 1}
VENDOR_SPECIFIC0	= string (16)
VENDOR_SPECIFIC1	= string (16)
POWER_SAVING_MODE	= {normal, low}
WOL_LIMIT_10	= {yes, no}
FIBER_WOL_CAPABLE	= {yes, no}

All reserved words are not case sensitive. A ‘;’, ‘//’ can be used at the beginning of line as comment.

Example:

```
; 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

6 User Interface Commands

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

Command Group vpd

vpdwrite	Write VPD Memory
vpdread	Read VPD Memory

Command Group nvram

semode	same as flshmode command
seread	Read NVRAM
sewrite	Write NVRAM
secfg	Configure NVRAM
seprg	Program NVRAM
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
asfcfg	Configure ASF in NVRAM
flshmode	Configure NVRAM mode
flshread	same as seread
chkpxe	Check PXE code image
dir	display file directory in NVRAM
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
setasf	Enable/Disable ASF
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

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 PC
breakpoint	set current CPU break point
select	select current CPU
u	Alias of disasm command
trap	trap cpu memory
cpuinfo	display cpu information
cpugprstatus	Display internal CPU GPR test status
cpugprrestart	Restart internal CPU GPR test

Command Group dma

dmaw	DMA from NIC to Host Memory
dmr	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

macbpk	Configure MAC loopback
nicstats	Display NIC Statistics
ringindex	Dump Ring Index
blast	Blast Packets in Poll Mode
phyctrl	Configure Speeds/Duplex
txpkt	Transmit Packet
statusblk	Dump Status Block
stsbk	Dump Statistics Block
txcfg	Configure protocol packets 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
pcimrd	Test an individual register

Command Group mii

mwrite	Write PHY registers via MII Management interface
mread	Read PHY 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

Command Group mem

memsearch	Search a Data Pattern in Memory
read	Read Memory
write	Write Memory
poll	poll Memory
setbit	Read-Modify Memory by ored with <bits>
clearbit	Read-Modify Memory by anded 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

Command Group test

vpdtest	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
cputest	Run CPU Test
dmatest	Run DMA Test
teste	Enable Test
testd	Disable Test
asftest	ASF Test
bustest	PCI Bus Test
sramtest	sram test
msitest	Run MSI Test
romtest	ROM Test
gpiotest	do GPIO test
cpudiag	run diagnostic from internal CPU
pcicfgtest	Run PCI Config. Reg. Test
petest	Perform parity error test on a bridge
errctrl	Configure Error Control Setting
sedvt	Perform NVRAM dvt test
miimisctest	Run MII Misc. Tests
cpugprtest	Run CPU GPR test

Command Group power

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

Command Group irq

intr	Dump Interrupt Info
------	---------------------

intrctrl	Mask/Unmask Current Interrupt
intt	Interrupt Tracer

Command Group mac

mbuf	Dump Content of Mbufs
loaddrv	Load Driver
unloaddrv	Unload Driver
machalt	Halt MAC Controller
ftq	Dump FTQ
addmc	Add Multicast MAC
delmc	Delete Multicast MAC
txmacdes	Program Destination address to UUT
txmacsrc	Program Source address to UUT
chklddrv	Check to see if driver is loaded
vlangtag	Display/Clear vlanTag information

Command Group misc

reg	register wizard
exit	Exit the System
debug	debug functions
gpiorwrite	Write a Value into GPIO pin
gpiorread	Read GPIO Value
pxecpy	Load PXE Code to Mbuf Memory
device	Show or Switch Device
version	Display Program Version
help	Display the Commands Available
?	Alternate Help Command
radix	Change System Radix
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
quit	Exit the System
smbus	ASF terminal
cls	Clear Screen
loop	loop on command
dbmode	Set DEBUG Mode to On or Off Mode
asf	run asf monitor program with option to Load asf firmware image
new	Display new command available
asfprg	Program asf firmware into NVRAM
sleep	suspense process for Excute command from a file
fillpattern	Fill WOL matching pattern into Misc. Memory Location
inp	input port
outp	output port
switch	Altinate 'device' command
do	Excute command 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 screen
pause	Pause for user to hit a key
q	Exit the System, alias name for quit
verbose	change verbose setting
beep	Create a beep sound
var	Display current variables
meminfo	report the 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

Command Group bridge

readbr	Read bridge's configuration space
writebr	Write bridge's configuration space
findbridge	Find all 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

7 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.

8 Test and Functions Description

8.1 vpdwrite

cmd: 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.

Example:

8.2 vpdread

cmd: vpread (Not support in A0)

Description: Read data from VPD storage

Syntax: vpdread start[-end | len]

Address range : 0x00 – 0xFF

num_byte : 256 (max)

Example:

8.3 semode

cmd: semode

Description: Configure Serial EEPROM to either Auto (I²C) or Manual (Bit-Bang) Mode.

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

Example:

1. Set Serial EEPROM mode to Auto (I²C).

0:> semode auto

2. Set Serial EEPROM mode to Bit-Bang.

0:> semode bitbang

3. Display Current mode

0:> semode

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

8.4 seread

cmd: seread

Description: Read NVRAM

Syntax: seread start[-end | len]

Options:

-a : force auto read

-m : force manual bit-bang read

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

8.5 sewrite

cmd: sewrite

Description: Write NVRAM

Syntax: sewrite start[-end | len] data

Options:

-a : force auto write

-m : force manual bit-bang write

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) ***
```

8.6 secfg

cmd: secfg

Description: Configure NVRAM

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:

- 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 (D0:D1:D2:D3) : 10:0:0:100
3. Power Consumed (D0:D1:D2:D3) : 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 { Enable(1), Disable(2) } : Disable
9. PXE Link Speed { Auto(0), 100HD(1), 100FD(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 { Phy Mode1 (1),Phy Mode2 (2)} : Phy Mode1
22. PHY ID ([PHY#2][PHY#3]) : 00206160
23. Max PCI Retry {0-7, 8=auto} : 8
24. ASF { Enable(1), Disable(2) } : Disable
25. Dual MAC mode {Normal(0), MAC_B only(1)
    MAC_A only(2), XBAR mode(3), swaped(4)} : 0
26. MBA Boot Protocol {PXE(0), RPL(1), BOOTP(2)} : 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 { 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. PowerSaving(TM) Mode {Normal(0), Low(1)} : Normal
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
```

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

8.7 seprg

cmd: seprg

Description: Program NVRAM

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
-o<HEX>	offset of serial nvram (def=00000000)

-s Do not restore original Serial Number

Example:

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

0:> seprg seprg.bin

8.8 upgfrm

cmd: upgfrm

Description: Upgrade PXE or Boot Code from a File. This command reads code from a file and program into pxe or boot area. Both parameter, the programming target 'pxe' or 'boot' and filename, must be specified.

Syntax: upgfrm <pxe | boot> filename

Options:

-b	Upgrade boot code
-d	Do not perform device check
-f<string>	Input file
-p	Upgrade PXE code

8.9 sever

cmd: sever

Description: Display Serial NVRAM Version

Syntax: sever

8.10 sechksum

cmd: sechksum

Description: Check/Update Serial NVRAM checksum

Syntax: sechksum

Options:

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

8.11 sedump

cmd: sedump

Description: Dump NVRAM content to a file

Syntax: sedump <filename> [HexLen]

Options :

- a<string> no Atmel Flash address translation
- f<string> filename
- l<DEC> file length, use 0 for entire NVRAM image (def=8192)

8.12 asfcfg

cmd: asfcfg

Description: Configure ASF in NVRAM

Syntax: asfcfg [filename]

8.13 flshmode

cmd: flshmode

Description: Configure ASF in NVRAM

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

8.14 flshread

cmd: flshread

Description: Same as seread

Syntax: flshread start[-end | len]

8.15 chkpxe

cmd: chkpxe

Description: Check PXE code image

Syntax: chkpxe <filename>

8.16 dir

cmd: dir

Description: display file directory in NVRAM

Syntax: dir

8.17 erase

cmd: erase

Description: erase file from directory in NVRAM

Syntax: erase <entry> | all

Options:

-y do not ask for conformation

8.18 pxeprg

cmd: pxeprg

Description: display file directory in NVRAM. This command reads PXE code from a file and program into NVRAM

Syntax: pxeprg <filename>

8.19 flshdev

cmd: flshdev

Description: Select flash device to access

Syntax: flshdev seeprom | atmelflash | sstflash

8.20 bitbang

cmd: 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>

8.21 seclock

cmd: seclock

Description: set NVRAM config1 content value after reset.

Syntax: seclock <32bitNewValue>

8.22 setorture

cmd: 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)

8.23 seinit

cmd: seinit

Description: Initialize NVRAM block.

Syntax: seinit

Options:

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

8.24 searb

cmd: searb

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

Syntax: searb [n]

8.25 seprotect

cmd: seprotect

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

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

8.26 selclock

cmd: selclock

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

Syntax: selclock <8bitNewValue>

8.27 semap

cmd: semap

Description: Display NVRAM usage

Syntax: semap

8.28 setwol

cmd: setwol

Description: Enable/Disable WOL

Syntax: setwol [e/d]

Options:

-d	Disable WOL
-e	Enable WOL

8.29 setpxe

cmd: 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

8.30 setasf

cmd: setasf

Description: Enable/Disable ASF

Syntax: setasf

Options:

-d	Disable ASF
-e	Enable ASF

8.31 secomp

cmd: 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)

8.32 dreset

cmd: dreset

Description: double reset test for EEPROM debug

Syntax: dreset

8.33 userblock

cmd: userblock

Description: create a userblock in NVRAM

Syntax: userblock

8.34 setmba

cmd: 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 |

8.35 loadfw

cmd: 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

8.36 cpudrt

cmd: 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
```

8.37 cpudtt

cmd: 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
```

8.38 cputrace

cmd: cputrace

Description: toggles cpu trace mode

Syntax: cputrace [1|0]

8.39 haltcpu

cmd: haltcpu

Description: Halt CPU

Syntax: haltcpu

8.40 loadbootcode

cmd: 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=rxpcu
-v	disable verbose

8.41 disasm

cmd: disasm

Description: Disassemble MIP instructions

Syntax: disasm [address [line]]

8.42 step

cmd: step

Description: Step MIP instructions

Syntax: step

8.43 go

cmd: go

Description: start CPU

Syntax: go

8.44 showgpr

cmd: showgpr

Description: toggles showing cpu gpr mode

Syntax: showgpr 1|0

8.45 pc

cmd: pc

Description: set current CPU PC

Syntax: pc

8.46 select

cmd: select

Description: select current CPU

Syntax: select <r|t>

8.47 breakpoint

cmd: breakpoint

Description: set current CPU breakpoint

Syntax: breakpoint

8.48 u

cmd: u

Description: Alias of disasm command

Syntax: u [address [line]]

8.49 trap

cmd: trap

Description: trap cpu memory

Syntax: trap <low> <high> | off

8.50 cpuinfo

cmd: cpuinfo

Description: display cpu information

Syntax: cpuinfo [r|t]

8.51 cpugprstatus

cmd: cpugprstatus

Description: Display internal CPU GPR test status

Syntax: cpugprstatus

8.52 cpugprrestart

cmd: cpugprrestart

Description: Restart internal CPU GPR test

Syntax: cpugprrestart

8.53 dmaw

cmd: 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 - FFFFFFFFFFFFFFFFFF 0000000000000000 FFFFFFFFFFFFFFFFFF 8 - FFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFF 0000000000000000... 9 - 00000000000000000000000000000000 FFFFFFFFFFFFFFFF... 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 ffffffffbf 64-bit-pattern ... (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 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

8.54 dmar

cmd: 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=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 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

8.55 dma_h

cmd: dma_h

Description: Dump DMA Entries

Syntax: dma_h <start> <end>

8.56 dma_d

cmd: dma_d

Description: Dump DMA Entries with Decode

Syntax: dma_d <start> <end>

8.57 maclpk

cmd: 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:> mcac1pb 1
Enabling MAC loopback ... OK
```

2. Disable MAC loop back.

```
0:> mac1pb 0
Disabling MAC loopback ... OK
```

8.58 nicstats

cmd: 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  d3StatsInt1MacTxErrors    :      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    :
Rxd Packets (Ring#05) :      0  0
Rxd Packets (Ring#06) :      0  0
Rxd Packets (Ring#07) :      0  0
Rxd Packets (Ring#08) :      0  0
Rxd Packets (Ring#09) :      0  0

```

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

8.59 ringIndex

cmd:

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

8.60 blast

cmd: 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

8.61 phyctrl

cmd: 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)

8.62 txpkt

cmd: 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
- L Don't initialize packets
- R Enable RS232 statistic update
- S TCP segmentation test

8.63 statusblk

cmd: 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

8.64 stsblk

cmd: stsblk

Description: Display Statistics Block.

Syntax: stsblk

Example:

```
0:> stsblk
***** STATISTICS Block @ 0x0027c0c0 *****
ifHCInOctets           :           0   etherStatsFragments      :
0
ifHCInUcastPkts        :           0   ifHCInMulticastPkts      :
0
ifHCInBroadcastPkts    :           0   d3StatsFCSErrors        :
0
d3StatsAlignmentErrors :           0   xonPauseFramesReceived  :
0
xonPauseFramesReceived :           0   macControlFramesReceived:
0
xonStateEntered        :           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
d3StatsDeferredTx       :           0   d3StatsExcessiveCol     :           0
d3StatsLateCol          :           0   d3Collided2Times        :
0
d3Collided3Times       :           0   d3Collided4Times        :           0
```

```

d3Collided5Times      :      0    d3Collided6Times      :      0
d3Collided7Times      :      0    0    d3Collided8Times      :
0
d3Collided9Times      :      0    d3Collided10Times     :      0
d3Collided11Times     :      0    d3Collided12Times     :      0
d3Collided13Times     :      0    d3Collided14Times     :      0
d3Collided15Times     :      0    ifHCOutUcastPkts      :      0
d3StatsCarSenseErrors :      0    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    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    0    COSIfHCOutPkts[13]    :
0
COSIfHCOutPkts[14]    :      0    COSIfHCOutPkts[15]    :      0
nicDmaRdHPQueueFull   :      0    0    nicSendDataCompQueueFull:
0
nicRingSetSdPIDx      :      0    nicRingStatusUpdate   :      0
nicInterrupts          :      0    nicAvoidedInterrupts   :      0
nicSendThresholdHit    :      0
Phy CRC counter        :      0

```

8.65 txcfg

cmd: 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) ->

8.66 rxcfg

cmd: 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) ->

8.67 tprot

cmd: 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)

8.68 qstat

cmd: 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

8.69 drvrcfg

cmd: 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)

8.70 irt

cmd: 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

8.71 macmrd

cmd: 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

8.72 miimrd

cmd: 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

8.73 pcimrd

cmd: 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

8.74 mwrite

cmd: 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
```

8.75 mread

cmd: 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
```

8.76 mdev

cmd: mdev

Description: Select current PHY to be accessed. The default device ID is 0x01. If no parameter is entered, it displays current phy address setting.

Syntax: mdev [<phy_id>]

Example:

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

8.77 miimode

cmd: 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
```

8.78 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

8.79 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

8.80 bertstats

Command: bertstats

Description: Dump PHY BIST statistics

Syntax: bertstats

8.81 rm

Command: rm

Description: Read MII Registers

Syntax: rm

8.82 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>

8.83 phymse

Command: phymse

Description: PHY mean square error.

Syntax: phymse

Options:

-p Polling continuously at 100ms interval

8.84 initphy

Command: initphy

Description: Initialize phy.

Syntax: initphy

8.85 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

8.86 read

cmd: read

Description: Read Memory

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

! = Configuration space (32)

S = Configuration space (16)

X = Configuration space (8)

= Registers (32) (default)

* = SRAM (32)

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

m = MII registers (16)

~ = VPD Access (32)

I = indirect access (32)

^ = internal scratchpad (32)

l = direct access (32)

s = direct access (16)

x = direct access (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
```

8.87 write

cmd: write

Description: Write Memory

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

! = Configuration space (32)

S = Configuration space (16)

X = Configuration space (8)

= Registers (32) (default)

* = SRAM (32)

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

m = MII registers (16)

~ = VPD Access (32)

I = indirect access (32)

^ = internal scratchpad (32)

l = direct access (32)

s = direct access (16)

x = direct access (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
```

8.88 poll

cmd: poll

Description: poll Memory

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

! = Configuration space (32)

S = Configuration space (16)

X = Configuration space (8)

= Registers (32) (default)

* = SRAM (32)

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

m = MII registers (16)

~ = VPD Access (32)

I = indirect access (32)

^ = internal scratchpad (32)

l = direct access (32)

s = direct access (16)

x = direct access (8)

Options:

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

8.89 setbit

cmd: 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 (32)

^ = internal scratchpad (32)

l = direct access (32)

s = direct access (16)

x = direct access (8)

8.90 clearbit

cmd: 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 (EEPROM/FLASH) (32)

m = MII registers (16)

~ = VPD Access (32)

I = indirect access (32)

^ = internal scratchpad (32)

l = direct access (32)

s = direct access (16)

x = direct access (8)

8.91 readbit

cmd: 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 (EEPROM/FLASH) (32)

m = MII registers (16)

~ = VPD Access (32)

I = indirect access (32)

^ = internal scratchpad (32)

l = direct access (32)

s = direct access (16)

x = direct access (8)

8.92 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]

8.93 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

8.94 vpdtest

cmd: 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

8.95 regtest

cmd: 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

8.96 miitest

cmd: 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)

8.97 msi

cmd: 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 bcmediag compatibility
- I initializing MSI block
- o<DEC> offset (def=0)

8.98 memtest

cmd: 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

8.99 setest

cmd: 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)

8.100 bist

cmd: bist

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

Syntax: bist [iteration]

8.101 nictest

cmd: 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

8.102 intrtest

cmd: intrtest

Description: Interrupt Test

Syntax: intrtest [iteration]

Options:

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

8.103 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

8.104 cputest

cmd: 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)

8.105 dmatest

cmd: 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)

8.106 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

8.107 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.

8.108 asftest

cmd: asftest

Description: ASF Test

Syntax: asftest

Options:

 -n<DEC> iteration

8.109 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>	Maximum 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 fffffffe 00000000 00000000
2	ffffffff ffffffff 00000000 00000000
.	.
.	.
.	.
64	7ffffffff ffffffff 00000000 00000000
65	00000000 00000000 ffffffff ffffffff
66	00000000 00000000 ffffffff fffffffe

```
67          00000000 00000000 ffffffff ffffffff
.
.
.
129         00000000 00000000 7fffffff 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#
```

8.110 sramtest

cmd: sramtest

Description: SRAM Test

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

8.111 msitest

cmd: 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

8.112 romtest

cmd: romtest

Description: ROM Test

Syntax: romtest

Options:

- n<DEC> iteration

8.113 gpiotest

cmd: gpiotest

Description: do GPIO test

Syntax: gpiotest

Options:

- n<DEC> iteration

8.114 cpudiag

cmd: 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

8.115 pcicfgtest

cmd: 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

8.116 petest

cmd: petest

Description: Perform parity error test on a bridge

Syntax: petest <bridge>

8.117 errctrl

cmd: errctrl

Description: Configure Error Control Setting

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

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

8.118 sedvt

cmd: 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

8.119 miimisctest

cmd: miimisctest

Description: Run MII Misc. Tests.

Syntax: miimisctest

Options:

-n<DEC>	iteration
---------	-----------

8.120 cpugprtest

cmd: cpugprtest

Description: Run CPU GPR test.

Syntax: cpugprtest

Options:

- r run rx_cpu only
- t run tx cpu only
- u run Address Up

8.121 pmdcfg

cmd: 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)

8.122 pmpd

cmd: 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)

8.123 intr

cmd: 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
```

8.124 intrctrl

cmd: 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
```

8.125 intt

cmd: intt

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

Syntax: intt

8.126 mbuf

cmd: 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
```

8.127 loaddrv

cmd: 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

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
```

8.128 unloaddrv

cmd: unloaddrv

Description: Unload NIC driver

Syntax: unloaddrv

Example:

```
0:> unloaddrv
```

```
Unloading MAC driver ... OK
```

8.129 machalt

cmd: machalt

Description: Halt MAC controller

Syntax: machalt

Example:

```
0:> machalt
```

```
Halting MAC ... OK
```

8.130 ftq

cmd: 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
```

8.131 addmc

cmd: addmc

Description: Add Multicast MAC

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

Example:

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

8.132 delmc

cmd: delmc

Description: Delete Multicast MAC

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

Example:

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

8.133 txmacdes

cmd: txmacdes

Description: Program Destination address to UUT

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

8.134 txmacsrc

cmd: txmacsrc

Description: Program Source address to UUT

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

8.135 chklddrv

cmd: chklddrv

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

Syntax: chklddrv

8.136 vlantag

cmd: vlantag

Description: Display/Clear vlanTag information.

Syntax: vlantag

Options:

-c clear vlanTag info

8.137 reg

cmd: reg

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

Syntax: reg [pci|mii] [offset]

8.138 debug

cmd: 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
```

8.139 gpiowrite

cmd: gpiowrite

Description: Write a Value into GPIO pin

Syntax: gpiowrite <GPIO_num> <value>

Valid value for <GPIO_num> is 0-2, <value> is 0 or 1.

Example:

1. Write 1 to GPIO#1 Pin

```
0:> gpiowrite 1 1

Writing 1 to GPIO#1
```

8.140 gpioread

cmd: gpioread

Description: Read GPIO Value

Syntax: gpioread

Example:

1. Read GPIO Pins

```
0:> gpioread
GPIO#0 : 1
GPIO#1 : 1
GPIO#2 : 0
```

8.141 pxecpy

cmd: pxecpy

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

Syntax: pxecpy <file>

Options:

-f<string> filename

8.142 device

cmd: 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

8.143 version

cmd: version

Description: Display Program Version

Syntax: version

8.144 help

cmd: 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]

8.145 ?

cmd: ?

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

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

8.146 radix

cmd: 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>

8.147 nolog

cmd: nolog

Description: Close the Current Logfile

Syntax: nolog

8.148 log

cmd: log

Description: Save all output to log file

Syntax: log

Options:

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

-a Append to existing file

8.149 pciinit

cmd: pciinit

Description: Initialize PCI configuration registers

Syntax: pciinit

8.150 pciscan

cmd: pciscan

Description: Scan for all PCI Devices

Syntax: pciscan

Example:

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

8.151 dos

cmd: dos

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

Syntax: dos <dos command>

8.152 diagcfg

cmd: diagcfg

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

Syntax: diagcfg

Example:

```
0:misc> 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 |

D4. MII Miscellaneous.....: Enabled
D5. 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 ->

8.153 reset

cmd: 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

8.154 smbus

cmd: 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

8.155 cls

cmd: smbus

Description: Clear Screen.

Syntax: cls

8.156 loop

cmd: loop

Description: loop on command.

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

8.157 dbmode

cmd: dbmode

Description: Set DEBUG Mode to On or Off Mode.

Syntax: dbmode on|off

8.158 asf

cmd: 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 asfcgub.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

8.159 new

cmd: new

Description: Display new command available. The default parameter for [n] is 10

Syntax: new [n]

8.160 asfprg

cmd: asfprg

Description: Program asf firmware into NVRAM. The default files names are asfinit.bin, asfcgua.bin, and asfcgub.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)

8.161 sleep

cmd: sleep

Description: suspense process for Excute command from a file

Syntax: sleep [milliseconds]

8.162 fillpattern

cmd: 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)

8.163 inp

cmd: inp

Description: input port

Syntax: inp <addr>

Options:

- l long word size
- w word size

8.164 outp

cmd: outp

Description: input port

Syntax: outp <addr> <value>

Options:

- l long word size
- w word size

8.165 switch

cmd: 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

8.166 do

cmd: 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)

8.167 txfill

cmd: 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

8.168 wbuf

cmd: wbuf

Description: Write tx|rx buffer with specified data at offset. Only works with static buffer selection -A.

Syntax: wbuf tx|rx <offset> <data>

8.169 rbuf

cmd: 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>

8.170 cpbuf

cmd: cpbuf

Description: Copy the content of rx buffer into tx buffer. Only works with static buffer selection -A.

Syntax: cpbuf <offset> <length>

8.171 echo

cmd: echo

Description: echo <string> to screen.

Syntax: echo <string>

8.172 pause

cmd: 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>

8.173 verbose

cmd: 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

8.174 beep

cmd: 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]

8.175 var

cmd: var

Description: Display current variables

Syntax: var

8.176 meminfo

cmd: meminfo

Description: report the memory information

Syntax: meminfo

8.177 delvar

cmd: delvar

Description: Delete local variables

Syntax: delvar

8.178 regdump

cmd: 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)

8.179 regcomp

cmd: regcomp

Description: Compare register content to a file. his 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)

8.180 regrestore

cmd: 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)

8.181 readbr

cmd: readbr

Description: Read bridge's configuration space.

Syntax: readbr

8.182 writebr

cmd: writebr

Description: Write bridge's configuration space.

Syntax: writebr

8.183 writebr

cmd: writebr

Description: Write bridge's configuration space.

Syntax: writebr

8.184 findbridge

cmd: findbridge

Description: Find all bridges in the system.

Syntax: findbridge

8.185 bridge

cmd: bridge

Description: Switch to specified bridge.

Syntax: bridge

8.186 pere

cmd: pere

Description: Enable parity error response on a bridge. Defaults to current bridge.

Syntax: pere <bridge>

8.187 perd

cmd: perd

Description: Disable parity error response on a bridge. Defaults to current bridge.

Syntax: perd <bridge>

8.188 peclr

cmd: peclr

Description: Clear parity error on bridge. Defaults to current bridge.

Syntax: peclr <bridge>

8.189 pechk

cmd: pechk

Description: Check parity error on bridge. Defaults to current bridge.

Syntax: pechk <bridge>

8.190 exit

cmd: exit

Description: Exit System

Syntax: exit

8.191 quit

cmd: quit

Description: Exit System

Syntax: quit

9 ERROR MESSAGES

```
/* NO_ERROR                                0 */ "",
/* ERR_IND_REG_ERR                         1 */ "Got 0x%08X @ 0x%08X. Expected 0x%08X",
/* 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 %s got changed after writing zero
at offset 0x%X",

/* ERR_READ_ONLY_SET                     5 */ "Read only bit %s got changed after writing one
at offset 0x%X",
/* ERR_READ_WRITE_NOT_CLEAR              6 */ "Read/Write bit %s did not get cleared after
writing zero at offset 0x%X",
/* ERR_READ_WRITE_NOT_SET                7 */ "Read/Write bit %s did not get set after writing
one at offset 0x%X",
/* ERR_BIST                              8 */ "BIST failed",
/* ERR_INTERRUPT                         9 */ "Could not generate interrupt",

/* ERR_ABORT                            10 */ "Aborted by user",
/* ERR_DMA_TXDATA                       11 */ "Tx DMA:Got 0x%08X @ 0x%08X. Expected 0x%08X",
/* ERR_DMA_RXDATA                       12 */ "Rx DMA:Got 0x%08X @ 0x%08X. Expected 0x%08X",
/* ERR_TXDMA                            13 */ "Tx DMA failed",
/* ERR_RXDMA                            14 */ "Rx DMA failed",

/* ERR_MEM                              15 */ "Data error, got 0x%08X at 0x%08X, expected
0x%08X",
/* ERR_MEM2                             16 */ "Second read error, got 0x%08X at 0x%08X,
expected 0x%08X",
/* ERR_EEP_WRITE                         17 */ "Failed writing NVRAM at 0x%04X",
/* ERR_EEP_READ                         18 */ "Failed reading NVRAM at 0x%04X",
/* ERR_EEP_DATA                         19 */ "NVRAM data error, got 0x%08X at 0x%04X, expected
0x%08X",

/* ERR_FILE_OPEN                        20 */ "Cannot open file %s",
/* ERR_BAD_CPU_CFG                      21 */ "Invalid CPU image file %s",
/* ERR_IMAGE_SIZE                       22 */ "Invalid CPU image size %d",
/* ERR_MALLOC                           23 */ "Cannot allocate memory for size %d",
/* ERR_CPU_RESET                        24 */ "Cannot reset %cX CPU",

/* ERR_CPU_RELEASE                      25 */ "", /* no longer in use --- Available */
/* ERR_CPU_TEST                         26 */ "%cx CPU test failed",
/* ERR_DMA_RANGE                        27 */ "Invalid Test Address Range\nValid NIC address is
0x%08X-0x%08X and exclude 0x%08X-0x%08X",
/* ERR_DMA_DATA                         28 */ "DMA:Got 0x%08X @ 0x%08X. Expected 0x%08X",
/* ERR_PHY_ID                           29 */ "Unsupported PhyId %04X:%04X",

/* ERR_PHY_T00_MANY_REG                 30 */ "Too many registers specified in the file, max is
%d",
/* ERR_VPD_WRITE                        31 */ "Cannot write to VPD address %04X",
/* ERR_VPD_DATA                         32 */ "VPD data error, got %08X @ 0x%04X, expected
%08X",
/* 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 %d. Only %d is transmitted",
```

```
/* ERR_DATA_RX_MISSING      36 */ "Expected %d packets. Only %d good packet(s) have
been received\n%d unknown packets have been received.\n%d bad packets have been
received.",
/* ERR_INVALID_TEST         37 */ "%c%d 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 %04X is too large, size must be less
than %04X",
/* ERR_FILE_TOO_BIG         43 */ "File size %d is too big, max is %d",
/* ERR_INVALID              44 */ "Invalid %s",

/* ERR_WRITE                45 */ "Failed writing 0x%x to 0x%x",
/* CMD_QUIT                 46 */ "",
/* ERR_CPU_MEM_ERR          47 */ "%s CPU access error @ %08X, expected %08X but
got %08X",
/* 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 %d cannot be mapped to %cX CPU event
bit %d",
/* ERR_ASF_RST_VAL         54 */ "%s 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 */ "%s Timer is not working",
/* ERR_ASF_EVENT            59 */ "Cannot clear bit %s in %cX CPU event register",

/* ERR_EEP_FILESIZE        60 */ "Invalid %s file size, expected %d but only can
read %d bytes",
/* ERR_MAGIC_VALUE         61 */ "Invalid magic value in %s, expected %08x but
found %08x",
/* ERR_EEP_FMT              62 */ "Invalid manufacture revision, expected %c but
found %c",
/* ERR_EEP_BOOTVER         63 */ "Invalid Boot Code revision, expected %d.%d but
found %d.%d",
/* 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 %02X-%02X-%02X-
%02X-%02X-%02X",
/* ERR_BUS                 69 */ "Slot error, expected an UUT to be found at
location %02X:%02X:00",
```

```
/* ERR_SPEC_MEM          70 */ "Adjacent memory has been corrupted while testing
block 0x%08x-0x%08x\nGot 0x%08x @ address 0x%08x. Expected 0x%08x",
/* 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: %04x",
/* 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
%08X to %08X at %04X",
/* ERR_VPD_READ           82 */ "Cannot read data from VPD address %04X",
/* 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=%08X",
/* 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 %04X, but read %04X at %08X",
/* 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 %d affects wider than
expected at offset 0x%X",

/* 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
%X",
/* ERR_ROM_SIZE           98 */ "ROM size error\nExpected %08X but read %08X
at\nROM Bar (0x30) register with %d written to ROM size reg.(0x88)",
/* ERR_ROM_DATA           99 */ "Data Error\nExpected %08X but read %08X at
%08X",

/* ERR_ROM_ENABLE         100 */ "Expansion ROM Desired bit is not set after
loading firmware",
/* ERR_GPIO               101 */ "GPIO%d Error, write=%d, read=%d",
/* ERR_GPIO5704           102 */ "Dev:%d Expected GPIO 0/1/2 = %d/%d/%d, but read
as %d/%d/%d",
/* ERR_BIST_NOT_DONE      103 */ "Bist test did not complete internally",
/* ERR_BIST_DATA_MISCOMP  104 */ "Bist data miscompared at bit: %d out: %d exp:
%d",

/* ERR_CPU_NO_RESPONSE    105 */ "No Response from firmware",
/* ERR_CPU_ERR_CODE       106 */ "%s CPU returned result %d, key = %d",
/* ERR_LOOP               107 */ "",
```

```
/* ERR_SKIP                108 */ "",
/* CMD_ELSE,               109 */ "",

/* CMD_ELSEIF,            110 */ "",
/* CMD_BREAK,             111 */ "",
/* CMD_ENDWHILE,          112 */ "",
/* ERR_BYTE,              113 */ "Byte access error: expected %02x at %08x but got
%02x",
/* ERR_WORD,              114 */ "Word access error: expected %04x at %08x but got
%04x",

/* ERR_NO_LINK_DOWN       115 */ "No link down found",
/* ERR_MISMATCHED_DEVICEID 116 */ "bootcode Image file belongs to %d family, does
not match with board %d",
/* 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 %d",
/* ERR_INVALID_BOND_ID    121 */ "Invalid bond id",
```