BROADCOM®

Broadcom NetXtreme Ethernet Adapter

Linux Diagnostic User's Guide

4/24/07

Prepared by: Tak Tomita
Updated by: Paul Nguyen, Austin Hui, Carlton Joseph

Broadcom Corporation
16125 Alton Parkway

Irvine, CA 92619-7013

www.broadcom.com

TABLE OF CONTENTS

1   Introduction

This document provides details of the Broadcom Linux Diagnostic. In order to run the Broadcom Linux diagnostic the following requirements must be met.

1. A Power PC system that uses Linux SuSE Enterprise Server 8.0 or an Intel x86 based machine running Redhat 8.0.
2. The system has a LOM 5704S or and 5704S NIC card.
3. A command shell (sh, csh, tcsh, bash, etc) is required to interface with the text based diagnostics tool.
4. The Broadcom driver version 7.1.11 or greater has be loaded, and the device has to be in the "up" state, for all devices that need to be tested via the diagnostics.
5. The b57ldiagp (PPC) or b57ldiagi (x86) executable has to be run as root.
6. Before running any of the diagnostic tests, on a new device with a blank NVRAM, the user must program the NVRAM of the device using the "seprg" command. Not programming the device's NVRAM will cause diagnostics to run very slow.

NOTES – The diagnostics online help takes precedence over this document.

2   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'.

2.1   Test Names

Group A. Register Tests

> A1. Indirect Register Access
> A2. Control Registers
> A3. Interrupt
> A4. PCI Cfg Registers
> A5. MII Registers

Groupt B. Memory Tests

> B1. Scratch Pad
> B2. BD SRAM
> B3. MBUF SRAM
> B4. CPU General Purpose Registers

Group C. Miscellaneous Tests

> C1. NVRAM
> C2. CPU
> C3. ASF

Group D. Driver Associated Tests

> D1. Mac Loopback
> D2. Phy Loopback
> D2. External Loopback

2.2   Test Descriptions

2.2.1   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

2.2.2   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.

2.2.3   A3. Interrupt Test

Command: intrtest

Function: This test verifies the interrupt functionality. It enables interrupt and waits for interrupt to occur.

Default: Enabled

2.2.4   A4. PCI Cfg Register Test

Command: nictest a4

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

2.2.5   A5. 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.

2.2.6   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 is test is repeated for entire test range.

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

Default: Enabled

2.2.7 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

2.2.8 B4. MBUF SRAM Test

Command: memtest -m

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

Default: Enabled

2.2.9 B6. CPU General Purpose Register

Command: memtest -p

Function: It tests SRAM locations, use by the CPU as General Purposes Registers, by performing the tests described in test B1. The Scratch Pad Test.

Default: Enabled

2.2.10 C1. NVRAM Test

Command: nictest c1

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

Default: Enabled

2.2.11 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

2.2.12 C3. ASF Test

Command: nictest c3

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.

2.2.13 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

2.2.14 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.

Default: Enabled

2.2.15 D2. External Loopback Test

Command: pkttest -e

Function: This test is same as D1. Mac Loopback Test except, the data is routed back via a loopback RJ45 connector

Default: Enabled

3   Command line option parameters

When user invokes the diagnostic program, optional parameters can be used to configure the operation of the program. This section summarizes the options. At present only one option is available.

-do <filename>   executes a script file of <filename>

Use the -do option to execute a script file containing diagnostics commands. The "User Commands" section below describes the diagnostics commands. The b57diag_scripts.pdf provides detail of the script file.

4   User Commands

The user commands are subdivided into the following groups: vpd, nvram, mem, test and misc.

Command Group vpd
vpdwrite	Write VPD Memory
vpdread	Read VPD Memory

Command Group nvram
secfg	Configure NVRAM
dir	Display file directory in NVRAM
semap	Display NVRAM usage
sechksum	Check/Update Serial NVRAM checksum
seread	Read NVRAM
sewrite	Write NVRAM
erase	Erase file from directory in NVRAM
seprg	Program NVRAM
upgfrm	Upgrade PXE or Boot Code from a File
pxeprg	Upgrade PXE from a File
setwol	Enable/Disable WOL
setpxe	Enable/Disable PXE
setmba	Enable Multiple Boot Agent
setasf	Enable/Disable ASF
asfprg	Program asf firmware into NVRAM
sedump	Dump NVRAM content to a file
secomp	compare eeprom content against the file
asfcfg	Configure ASF in NVRAM

Command Group cpu
halt	Halt CPU
disasm	Disassemble MIP instructions
u	Alias of disasm command
step	Step MIP instructions
go	start CPU
breakpoint	set current CPU breakpoint
bp	Alternate command for breakpoint

Command Group mem
read	Read Memory
write	Write Memory

Command Group test
errctrl	Configure Error Control Setting
nictest	Run a set of NIC Tests
diagcfg	Diagnostics Configuration
teste	Enable Test
testd	Disable Test
memtest	Run Memory Test
miitest	Run MII Memory Test
intrtest	Run Interrupt Test
regtest	Run Register Test
asftest	ASF Test
pkttest	Run Packet Tests

cputest	Run CPU Test

Command Group misc

quit	Exit the System
exit	Exit the System
help	Display the Commands Available
?	Alternate Help Command
loop	Loop on command
log	Open Logfile
nolog	Close the Current Logfile
version	Display Program Version
device	Show or Change Active Device
var	Display current variables
do	Excute command from a file
delvar	Delete all local variables
verbose	change verbose setting
shell	Execute system shell command
reset	Device Reset
dinfo	Display Driver Information
suspend	Suspend Driver
resume	Suspend Driver

Command Group asf

asf	run asf monitor program with option to Load asf firmware image

5 Test and Functions Description

5.1 vpdread

cmd: vpread

Description: Read data from VPD storage

Syntax: vpdread start[-end | len]

Address range : 0x00 – 0xFF

num_byte : 256 (max)

5.2 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.

5.3 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:

-f : force to program with defaults

Example:

1   Program Serial EEPROM with default values.

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

 1. MAC Address                                    : 00:10:18:04:1a:36
 2. Power Dissipated (D3:D2:D1:D0)                 : 10:0:0:100
 3. Power Consumed (D3:D2:D1:D0)                   : 10:0:0:100
 4. Vendor ID                                      : 14E4
 5. Vendor Device ID                               : 1653
 6. Subsystem Vendor ID                            : 14E4
 7. Subsystem Device ID                            : 1653
 8. PXE { Enable(1), Disable(2) }                  : Disable
 9. PXE Link Speed { Auto(0),10HD(1),10FD(2)
                     100HD(3),100FD(4)}            : Auto
10. Magic Packet WoL { Enable(1), Disable(2) }     : Disable
11. Product Name : Broadcom NetXtreme Gigabit Ethernet Controller
12. Part Number                                    : BCM95705A50
13. Engineering Change                             : 106679-15
14. Serial Number                                  : 0123456789
15. Manufacturing ID                               : 14e4
16. Asset Tag                                      : XYZ01234567
17. Part Revision                                  : A0
18. Voltage  { 1.3V(0), 1.8V(1) }                  : 1.8V
19. Force PCI Mode { Enable(1),Disable(2) }        : Disable
20. PHY Type { Copper(1),Fiber(2) }                : Copper
21. Led Mode { 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(3), swap(4), swapXBAR(7)}: 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. Reversed Nway {No(0), Yes(1)}                  : No
34. Limit WoL Speed to 10 {No(0), Yes(1)}          : No
35. Fiber WoL Capable {No(0), Yes(1)}              : No
36. Clock-Run Setting {Disable(0), Enable(1)}      : Disable
37. Enable PHY Auto Powerdown {No(0), Yes(1)}      : No
38. Disable PowerSaving capability {No(0),Yes(1)}: No
39. Hide MBA Setup Prompt {Disable(0), Enable(1)}: Disabled
40. MBA Setup Hot Key {Ctrl-S(0),Ctrl-B(1)}        : 0

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

Description of above parameters:

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

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

1	MAC Address	`Configure`
`Description:`		
MAC address of the device.		

2	Power Dissipated (D3:D2:D1:D0)	`Reference`
`Description:`		
Power dissipated in D3-0 states.		

3	Power Consumed (D3:D2:D1:D0)	`Reference`
`Description:`		
Power consumed in D0-2 states.		

4	Vendor ID	`Configure`
`Description:`		
PCI Vendor ID. Default 0x14e4.		

5	Device ID	`Configure`
`Description:`		
PCI Device ID.		

6	Subsystem Vendor ID	`Configure`
`Description:`		
PCI Subsystem PCI Vendor ID.		

7	Subsystem Device ID	`Configure`
`Description:`		
PCI Subsystem PCI Device ID.		

8	PXE	`Configure`
`Description:`		
Enable the Pre-Boot Execution Environment (PXE) by using this option. Options 8,9, 26 - 29 and 39-40 configure the boot protocol behavior.		
Enable(1)	When PXE is enable the expansion ROM enable bit in the PCI configuration space is set for system boot up. The user has to make sure the PXE code is loaded into the non-volatile memory by using the "loadpxe" command.	
Disable(2)	When PXE is disable the expansion ROM enable bit in the PCI configuration space is cleared for system boot up. The PXE code does NOT have to been load into the non-volatile memory of the device.	

9	PXE Link Speed	`Configure`
`Description:`		
Pre-Boot Execution Environment line configuration is specified using the parameters below. Options 8,9, 26 - 29 and 39-40 configure the boot protocol behavior.		
Auto(0)	PXE auto detects the link configuration.	
10HD(1)	PXE uses a 10 Mbits/s, half duplex line configuration.	
10FD(2)	PXE uses a 10 Mbits/s, full duplex line configuration.	
100HD(3)	PXE uses a 100 Mbits/s, half duplex line configuration.	

100FD(4)	PXE uses a 100 Mbits/s, full duplex line configuration.

`10` Magic Packet WoL	`Configure`
`Description:` A system can be configured to power-on when a Magic Packet is received.	
Enable(1)	The device will assert the pme signal, to power on the system, when a magic packet is received.
Disable(2)	Magic packets are ignored.
`Note:` A driver can setup the WoL behavior of a device and the value programmed into this location is ignored.	

`11` Product Name	`Reference`
`Description:` VPD Produce description string.	

`12` Part Number	`Reference`
`Description:` VPD part number..	

`13` Engineering Change	`Reference`
`Description:` VPD engineering change.	

`14` Serial Number	`Reference`
`Description:` VPD serial number.	

`15` Manufacturing ID	`Reference`
`Description:` VPD manufacturing ID.	

`16` Asset Tag	`Reference`
`Description:` VPD asset tag.	

`17` Part Revision	`Reference`
`Description:` VPD part revision.	

`18` Voltage	`Configure`
`Description:` Device voltage source.	
1.3V(0)	Selects a 1.3V source.
1.8V(1)	Selects a 1.8V source.

`19` Force PCI Mode	`Configure`
`Description:` PCI bus operational mode configuration.	
Enable(1)	When enabled the device uses PCI mode, instead of PCI-X, independent of the capabilities of the slot the device is plugged into.
Disable(2)	When disabled the device uses the PCI mode of the slot the device is plugged into; if the device is capable of operating in the required mode.

`20` PHY Type {option no longer supported}	`Configure`
`Description:` PHY line type configuration.	
Copper(1)	The communication medium is copper.

Fiber(2)	The communication medium is fiber.

`21`	Led Mode	`Configure`

`Description:`
A device can be configured to use one LED to indicate speed and activity or three LEDs. Use a devices data sheet to verify the LED modes supported by a devices and for exceptions to the LED modes described below.

Phy Mode1 (1)	Three LEDs are used for 10/100/1000 Mbits/s and each is driven individually by the device.
Phy Mode2 (2)	One LED is used for 10/100/1000 Mbits/s and is connected as described below. 5700/01 – The Link10 line will indicate link for all speeds. Link100 and Link1000 will encode the line speed as show in the following table: <table><tr><td>`Link 100`</td><td>`Link 1000`</td><td>`Speed`</td></tr><tr><td>`0`</td><td>`0`</td><td>`10 Mbits/s`</td></tr><tr><td>`1`</td><td>`0`</td><td>`100 Mbits/s`</td></tr><tr><td>`0`</td><td>`1`</td><td>`1000 Mbits/s`</td></tr></table> For all other 57xx devices the three link lines operate in an open drain configuration and can be tied together with a pull up resistor to control a LED.

`22`	PHY ID	`Reference`

`Description:`
Only for display purposes not used in software.

[PHY#2][PHY#3]	

`23`	Max PCI Retry	`Configure`

`Description:`
The maximum number of time to retry an aborted PCI operation.

0-7	The number of MAX PCI retries is force by the users configuration.
8=auto	The MAX PCI retry field is selected dynamically by firmware based on PCI bus type detected.

`24`	ASF	`Configure`

`Description:`
The functionality of Alert Standard Format (ASF) is enabled by this option.

Enable(1)	If the ASF code is loaded in NVRAM it is loaded and executed by the device.
Disable(2)	No ASF functionality is provided when disable.

`25`	Dual MAC mode	`Configure`

`Description:`
This option is only valid for a dual port device such as a 5704. The physical ports will be named MAC_A an MAC_B for the discussion below.

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

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

26	MBA Boot Protocol	Configure

Description:
Select a Multiple Boot Agent. Options 8,9, 26 - 29 and 39-40 configure the boot protocol behavior.

PXE(0)	PreBoot Execution Environment (PXE) is the boot protocol.
RPL(1)	Remote Program Load (RPL) is the boot protocol.
BOOTP(2)	Boot Protocol (BOOTP) is the boot protocol.

27	MBA Bootstrap Type	Configure

Description:
The BIOS bootstrap methods listed below are supported. Options 8,9, 26 - 29 and 39-40 configure the boot protocol behavior.

Auto(0)	Automatically configured use one of the methods below.
BBS(1)	A BIOS that supports the BIOS Boot Specification (BBS) can initiate the bootstrap method via the expansion ROMs Bootstrap Entry Vector.
Int18(2)	A INT18 is used to initiate the bootstrap method.
Int19(3)	A INT19 is used to initiate the bootstrap method.

28	MBA Delay Time	Configure

Description:
The amount of time the MBA boot message is displayed, in order to give a user the option to enter the BOOT parameter configuration screen.

0-15	The number of seconds the MBA banner is displayed.

29	Expansion ROM size	Configure

Description:
This value determines the size of the expansion ROM. Options 8,9, 26 - 29 and 39-40 configure the boot protocol behavior.

64k(0)	Advertises 64k expansion ROM size.
128k(1)	Advertises 128k expansion ROM size.
256k(2)	Advertises 256k expansion ROM size.
512k(3)	Advertises 512k expansion ROM size.
1M(4)	Advertises 1M expansion ROM size.
2M(5)	Advertises 2M expansion ROM size.
4M(6)	Advertises 4M expansion ROM size.
8M(7)	Advertises 8M expansion ROM size.
16M(8)	Advertises 16M expansion ROM size.

30	Design Type	Configure

Description:
Select NIC or LOM based Ethernet controller.

NIC(0)	Option selected for a plug in network interface card.
LOM(1)	Option selected for a LAN on motherboard.

Note:
The firmware and the drivers use this to determine the operational characteristics of a device. For example, on a LOM GPIO2 is tied to the SEEPROM write protect pin and on a NIC GPIO 1 and 2 is used to switch between main and auxiliary power for wake on LAN functionality.

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

32	Read/Write VPD Vendor Specific Data (V1)	Reference
Description:		
VPD V1 value. Data field provided for the customer.		

33	Reversed Nway	Configure
Description:		
Nway Negotiation.		
0	(default) Auto-negotiation is done from 1000->100->10 Mbits/s.	
1	Auto-negotiation is done from 10->100->1000 Mbits/s.	
Note:		
A user could set the chip for option 1 when running a laptop on battery power. The chip would auto-negotiate starting at a lower speed and lower power. The chip would only go to higher line rates and higher power if the lower line rates were unavailable.		

34	Limit WoL Speed to 10	Configure
Description:		
Limit Wake on LAN (WoL) line speed.		
No(0)	10 or100 Mbits/s is used for WoL.	
Yes(1)	Only 10 Mbits/s is used for WoL.	

35	Fiber WoL Capable	Configure
Description:		
Fiber Wake on LAN (WoL) Capable enable.		
No(0)	Disable Fiber WoL.	
Yes(1)	Enable Fiber WoL.	

36	Clock-Run Setting	Configure
Description:		
Enable Clock-Run on mini-PCI/cardbus systems. This parameter is valid only for the 5705 A0-A2.		
Enable(1)	Sets the clock mode register bit 22 which will assert (active low) the clock run signal on the bus prior to any PCI configuration space activity.	
Disable(0)	The above is not performed.	

37	Enable PHY Auto Powerdown	Configure
Description:		
Enable GPHY auto-power down when there is no link present (to conserver power).		
No(0)	The PHY will not auto power down.	
Yes(1)	The PHY will auto power down when there is no link.	

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

39	Hide MBA Setup Prompt	Configure
Description:		
During the MBA boot, the MBA setup prompt is displayed to provide the user the option to setup and configure various MBA parameters. Options 8,9, 26 - 29 and 39-40 configure the boot protocol behavior.		

Disable (0)	Hide the MBA Setup Prompt. The user is NOT give the option to change the MBA boot parameters.
Enable (1)	Show the MBA Setup Prompt. The user is given the option to change the MBA boot parameters.

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

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

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

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

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

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

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

```
Description:
```
Encoded Hot plug power values 1 & 2 for a NIC device that is used only if enabled in option 50 below. The following description applies to options 46-49.

This option is only valid for 5787 and 575xCx devices. The power values need to be setup by the OEM / manufacture based on actual measurements. Default values, contained in the bootcode / firmware, are loosely based on the in house Broadcom NIC card.

15-0   Power Budget data 1
   7-0   Base Power in .1 Watt
       For example, 1.4 Watt should have value 14.
   9-8   PM State
       00   D0
       01   D1
       10   D2
       11   D3
   12-10   Type
       000   PME Aux
       001   Auxiliary
       010   Idle
       011   Sustained
       111   Maximum
   13-15   Power Rail
       000   12V
       001   3.3V
       010   1.8V
       111   Thermal
31-16   Power Budget data 2
See Power Budget Data 1 for detail, Bit number add 16.

47	Encoded Hot Plug Power 3 & 4 For NIC	Configure

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

48	Encoded Hot Plug Power 5 & 6 For NIC	Configure

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

49	Encoded Hot Plug Power 7 & 8 For NIC	Configure

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

50	Hot Plug Power	Configure

```
Description:
```
Enable/Disable the hot plug power values defined in options 46-49.

Disable (0)	Disable hot plug power feature.
Enable (1)	Enable hot plug power feature.

51	Pri. Port SMB Address (ASF/IPMI)	Configure

```
Description:
```
SMB address of the primary port on a device.

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

5.4 dir

cmd: dir

Description: display file directory in NVRAM

Syntax: dir

5.5 semap

cmd: semap

Description: Display NVRAM usage

Syntax: semap

5.6 sechksum

cmd: sechksum

Description: Check/Update Serial NVRAM checksum

Syntax: sechksum

Options:

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

5.7 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
```

5.8   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) ***
```

5.9   erase

cmd: erase

Description: erase file from directory in NVRAM

Syntax: erase <entry> | all

Options:

-y   do not ask for conformation

5.10 seprg

cmd: seprg

Description: Program NVRAM

Syntax: seprg <file_name>

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

or

0:> seprg -f\root\seprg.bin

5.11 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

5.12 pxeprg

cmd: pxeprg

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

Syntax: pxeprg <filename>

5.13 setwol

cmd: setwol

Description: Enable/Disable WOL

Syntax: setwol [e/d]

Options:

        -d   Disable WOL

        -e   Enable WOL

5.14 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

5.15 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

5.16 setasf

cmd: setasf

Description: Enable/Disable ASF

Syntax: setasf

Options:

-d	Disable ASF
-e	Enable ASF

5.17 asfprg

cmd: asfprg

Description: Program asf firmware into NVRAM. The default files names are asfinit.bin, asfcpua.bin, and asfcpub.bin, which can be over written by parameters.

Syntax: asfprg [init_img [rx_img [tx_img]]]

Options:

-v\<HEX\>   verbose level (0,1,2) (def=00000001)

5.18 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 all data in NVRAM (def=8192)

5.19 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)

5.20 asfcfg

cmd: asfcfg

Description: Configure ASF in NVRAM

Syntax: asfcfg [filename]

5.21 halt

cmd: halt

Description: Halt CPU

Syntax: halt

5.22 disasm

cmd: disasm

Description: Disassemble MIP instructions

Syntax: disasm [address [line]]

5.23 u

cmd: u

Description: Alias of disasm command

Syntax: u [address [line]]

5.24 step

cmd: step

Description: Step MIP instructions

Syntax: step

5.25 go

cmd: go

Description: start CPU

Syntax: go

5.26 breakpoint

cmd: breakpoint

Description: set current CPU breakpoint

Syntax: breakpoint

5.27 bp

cmd: bp

Description: set current CPU breakpoint

Syntax: bp

5.28 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 (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)

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
   ```

5.29 write

cmd: write

Description: Write Memory

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

! = Configuration space (32)

S = Configuration space (16)

X = Configuration space (8)

= Registers (32) (default)

* = SRAM (32)

$ = NVRAM (SEEPROM/FLASH) (32)

m = MII registers (16)

~ = VPD Access (32)

I = indirect access (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
```

5.30 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
```

5.31 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

5.32 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 ->
```

5.33 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

5.34 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.

5.35 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

5.36 miitest

cmd: miitest

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

Syntax: miitest [itereation]

Options:

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

5.37 intrtest

cmd: intrtest

Description: Interrupt Test

Syntax: intrtest [iteration]

Options:

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

5.38 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

5.39 asftest

cmd: asftest

Description: ASF Test

Syntax: asftest

Options:

-n<DEC>   iteration

5.40 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

5.41 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)

5.42 quit

cmd: quit

Description: Exit System

Syntax: quit

5.43 exit

cmd: exit

Description: Exit System

Syntax: exit

5.44 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]

5.45 ?

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]

5.46 loop

cmd: loop

Description: loop on command.

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

5.47 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

5.48 nolog

cmd: nolog

Description: Close the Current Logfile

Syntax: nolog

5.49 version

cmd: version

Description: Display Program Version

Syntax: version

5.50 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

5.51 var

cmd: var

Description: Display current variables

Syntax: var

5.52 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)

5.53 delvar

cmd: delvar

Description: Delete local variables

Syntax: delvar

5.54 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

5.55 shell

cmd: dos

Description: Execute a shell command.

Syntax: shell <shell command>

5.56 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

5.57 dinfo

cmd: dinfo

Description: Prints out driver information.

Syntax: dinfo

5.58 suspend

cmd: suspend

Description: Suspends the driver

Syntax: suspend

5.59 resume

cmd: resume

Description: resumes the driver.

Syntax: resume

5.60 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, asfcpua.bin, and asfcpub.bin, which can be over written by parameters.

Syntax: asf [init_img [rx_img [tx_img]]]

Options:

-l	Load firmware only
-m	Enter asf mode only
-w	Simulate warm boot

6 ERROR MESSAGES

```
static u08 * errorMsg[] = {
/* 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",

/* CMD_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_NO_RESP            25 */ "%cx CPU does not respond",
/* 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%02X @ 0x%08X. Expected 0x%02X\nSRAM
data=0x%02X @ 0x%08X",
/* ERR_PHY_ID                 29 */ "Unsupported PhyId %04X:%04X",

/* ERR_PHY_TOO_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 Regsiter 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 acknowleged",
/* ERR_ADF_NO_STAMP          57 */ "Timestamp Counter is not counding",
/* 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_TIMOUT    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_MISCMP      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",
/* CMD_LOOP                  107 */ "",
/* CMD_SKIP                  108 */ "",
/* CMD_ELSE,                 109 */ "",

/* CMD_ELSEIF,               110 */ "",
/* CMD_BREAK,                111 */ "",
/* CMD_ENDWHILE,             112 */ "",
/* ERR_BYTE,                 113 */ "Byte access error: expected %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",
/* ERR_BAD_CPU_RESET         122 */ "CPU reset failed, register 5034 is 0x%x",
/* ERR_INCORRECT_VERSION     123 */ "Incorrect version",
/* ERR_MISMATCH_CFG_VERSION  124 */ "Mismatched CFG and FW Image version",
/* ERR_NOT_SUPP_CFG_BW       125 */ "Not support ASF_T_VERSION backward compatible",
/* ERR_POST_1G_LB            126 */ "1G Tx/Rx Lines Have A Short/Open",
/* ERR_DRIVER                127 */ "Driver %d.%d.%d or later is required to run this
function",
/* ERR_TXDMA_OVERFLOW        128 */ "TxDMA Overflow",
/* ERR_RXDMA_OVERFLOW        129 */ "RxDMA Overflow",
/* ERR_DRIVER_BAD_STATUS     130 */ "Driver returned error status",
/* ERR_INVALID_HANDLE        131 */ "Invalid Handle",
/* ERR_SOCKET                132 */ "Socket open error%d %s",
/* ERR_SIOCGIFINDEX          133 */ "ioctl(): SIOCGIFINDEX failed",
/* ERR_BIND                  134 */ "bind() failed: %s",
/* ERR_SETSOCKOPT            135 */ "setsockopt() failed:%d %s",
/* ERR_FCNTL                 136 */ "fcntl() failed:%d %s",
/* ERR_SEND                  137 */ "send() failed (cnt=%d): %d %s",
/* ERR_RECEIVE               138 */ "recvfrom() failed (cnt=%d): %d %s",
/* ERR_FALSE_CARRIER         139 */ "Error! False Carrier Detected During The Test",
/* ERR_INV_OPT               140 */ "Invalid Options",
/* ERR_INV_DEV               141 */ "Found Rv = %s, Expected Rv = %s due to -ckdev value",
/* ERR_IRQ                   142 */ "Invalid IRQ %d",
/* ERR_TIMEOUT               143 */ "Timeout",
/* ERR_PKT_DATA              144 */ "Packet data error at offset %d, expected %02X but
recevied %02X",
/* ERR_MAC_ZERO              145 */ "Zero Mac Address in Mac Register\n",
/* ERR_MAC_MIS_REG           146 */ "Mac Address MisMatch: Got %02X-%02X-%02X-%02X-%02X-
%02X.\n",

};
```