BCM5700

Diagnostic User's Guide

Revision 0.2 • Date 09/05/00

Prepared by: Austin Hui

Broadcom Corporation
16125 Alton Parkway

Irvine, CA 92619-7013

www.broadcom.com

TABLE OF CONTENTS

1. Introduction

This document provides the information on how to use the engineering diagnostic on Broadcom Gigabit Ethernet adapter, in particular to check out the functionality of the BCM5700 and its related components.

2. Prerequisites

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

OS: Dos 6.22

Software: bcmediag.exe, dos4gw.exe, and cpu.bin.

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

ctrlreg.txt (mac registers test input file)

miireg.txt (mii registers test input file)

vpdwrite.txt (write vpd input file)

seprg.txt (program SEEPROM input file)

physcript.txt (Phy speed select)

wol.txt (Power Down MAC pattern input file)

firmware.bin (TX & RX CPUs Firmware file)

eeprom.bin (Serial EEPROM config input file)

cpu.bin (CPU test)

Output File List:

The following files will be generated in run time.

diagcfg.bin

bcmediag.log

3. Test List

Test:	cmd:
VPD test	vpdtest (Not for A0)
Mac register test	regtest
SEEPROM test	setest
CPU test	cputest
DMA test	dmatest
MII test	miitest
Memory test	memtest
Interrupt test	intrtest
MSI test	msi (Not for A0, A1)
NIC test	nictest

4. Functions List

Functions:	cmd:
VPD read write	vpdread, vpdwrite
SEEPROM mode configuration	semode
SEEPROM read write	seread, sewrite
SEEPROM program	seprg
SEEPROM configuration	secfg
RX CPU trace display	cpudrt
TX CPU trace display	cpudtt
DMA read write	dmar, dmaw
DMA entries display	dma_h, dma_d
TX packet configuration	txcfg
TX packet	txpkt
RX configuration	rxcfg
Statistics block display	stsblk
Status block display	statusblk
Chip reset	reset
Phy speed configuration	phyctrl
MAC internal loop back	maclpb
MII read write	mread, mwrite
Phy select	mdev
MII access modes select	miimode
Generic memory read	read
Generic memory write	write
Power management info	pmdcfg

PM add/del pattern	pmpcfg
MAC power down	pmpd
Poll ISR	poll
Interrupt info display	intr
Halt MAC controller	machalt
Add/Del Multicast MAC	addmc, delmc
Display FTQ	ftq
Display MBUFs	mbuf
Load MAC driver	loaddrv
Unload MAC driver	unloaddrv
Load TX RX CPU firmware	loadfw
Display NIC statistics	nicstats
Display debug info	debug
PCI scan	pciscan
Diagnostics configuration	diagcfg
Log output to file	log
Close log file	nolog
Number base setting	radix
Exit application	exit, quit
Go up one level of menu	up
Blast Packets in Poll Mode	blast
Control Output of GPIO Pin	gpiowrite
Get Input of GPIO Pin	gpioread
Repeat a command in x iteration	loop
Display Diagnostics Version	version

Dump Ring Index	ringindex
Enter Dos shell	dos
Copy PXE code to MBUF memory	pxecpy
Initialize PCI configuration registers	pciinit
Control Interrupt Controller	intrctrl

5. 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.

6. Test and Functions Description

6.1 vpdwrite

cmd: vpdwrite (Not support in A0, A1)

Description: Write data to VPD storage.

Syntax: vpdwrite [-f <file>] <begin_addr> [- end_addr | num_bytes]

File format:

Address range: 0x00 – 0xFF

num_bytes: 256 (max)

Example:

6.2 vpdread

cmd: vpread (Not support in A0)

Description: Read data from VPD storage

Syntax: vpdread <begin_addr> [- end_addr | num_bytes]

Address range : 0x00 – 0xFF

num_byte : 256 (max)

Example:

6.3 vpdtest

cmd: vpdtest (Not support in A0, A1)

Description: Write designed pattern toVPD storage. Then read back and compare with designed pattern.

Syntax: vpdtest <-n=a |-p=b |-c=c> | <-?> for help

-a : Number of iterations (0 means forever)

-b : Pattern to test.

0 - Increment (def);

1 - Decrement ;1 - 0's

2 - FF's

3 - AA55

4 - 55AA

-c : 1 means corruptive test ; 0 means otherwise

Example:

6.4 semode

cmd: semode

Description: Configure Serial EEPROM to either Auto (I^2C) or Manual (Bit-Bang) Mode.

Syntax: semode <auto> | <man> | <> for help

Example:

1. Set Serial EEPROM mode to Auto (I^2C).

0:main> semode auto

2. Set Serial EEPROM mode to Manual (Bit-Bang).

0:main> semode man

3. Display Help.

0:main> semode
Usage : semode [auto | man]

6.5 seread

cmd: seread

Description: Read content of designated block of Serial EEPROM.

Syntax: seread <begin_addr> [- end_addr | num_bytes]

Address range: 0x00 – 0x1000

Num_bytes: 4097

Example:

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

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

6.6 sewrite

cmd: sewrite

Description: Write data to designated block of Serial EEPROM.

Syntax: sewrite <begin_addr> [- end_addr | value]

Address range: 0x0000 – 0x1000

Example:

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

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

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

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

6.7 seprg

cmd: seprg

Description: Program designated Serial EEPROM block via an input file. Input file need to be found in the same location as bcmediag.exe.

Syntax: seprg <file_name> | <> for help

File_name:

Example:

1. Program Serial EEPROM via input file seprg.txt

0:main> seprg seprg.txt

2. Display Help

```
0:main> seprg
Usage : seprg <file>
```

6.8 secfg

cmd: secfg

Description: Configure Serial EEPROM content.

If selected program with defaults (-f=1), eeprom.bin must be found in the same directory of bcmediag.exe.

Syntax: secfg <-v=a | -f =b> | <-?> for help

-a : Verbose Level

-b : 1 means force to program with defaults

Example:

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

```
0:main> secfg -v=0 -f=1
Reading current serial eeprom ... OK

1. MAC Address                      : 00:00:00:00:00:00
2. Part Number                      : BCM5700
3. Revision                         : P0
4. Power Dissipated (D0:D1:D2:D3)   : 0:0:0:0
5. Power Consumed (D0:D1:D2:D3)     : 0:0:0:0
0. Exit
20. Save and Exit


Enter your choice (option=paramter) :0
```

2. Set verbose level = 1 to display detail of the Serial EEPROM configuration.

```
0:main> secfg -v=1 -f=0
Reading current serial eeprom ... OK

*****************************************************
Magic Number   : 0x669955aa
Boot Code Info (start,length,offset): 0x08000000,0x69,0x0200
Code Directory (start,length,offset)
Dir#0 : 0x0,-4,0x000000ff    Dir#1 : 0x1,8,0x00000000
Dir#2 : 0x0,0,0x00000000    Dir#3 : 0x0,0,0x00000000
Dir#4 : 0x0,0,0x00000000    Dir#5 : 0x0,0,0x00000000
Dir#6 : 0x0,0,0x00000000    Dir#7 : 0x0,0,0x00000000
*****************************************************

1. MAC Address                      : 00:00:00:00:00:00
2. Part Number                      : BCM5700
3. Revision                         : P0
```

```
4. Power Dissipated (D0:D1:D2:D3)   : 0:0:0:0
5. Power Consumed (D0:D1:D2:D3)     : 0:0:0:0
0. Exit
20. Save and Exit

Enter your choice (option=paramter) : 0
```

3. Display Help.

```
0:main> secfg -?
Usage : secfg -v=a -f =b
   a : Verbose Level
   b : 1 means force to program with defaults.
```

6.9 setest

cmd: setest

Description: Serial EEPROM read write test

Syntax: setest <-n=a> | <-?> for help

-a : Number of iterations (0 means forever)

Example:

1. Run Serial EEPROM read write test.

```
0:seeprom> setest -n=1
Iteration   : 1
Writing to serial EEPROM ... OK
Reading and Verifying serial EEPROM ... OK
```

2. Display Help.

```
0:main> setest -?
Usage : setest -n=a
    a : Number of iterations (0 means forever)
```

6.10 flshmode

cmd: flshmode

Description: Configure Serial Flash/EEPROM to either Auto (I^2C) or Manual (Bit-Bang) Mode and display the current setting.

Syntax: flshmode [auto | bitbang | passthru | nopass]

Example:

1. Display help and the current setting.

```
0:main> flshmode
Usage: flshmode [ auto | bitbang | passthru | nopass]
         Current mode: auto; no passthru
         Type: Flash with no buffer
```

2. Set Serial flash mode to auto.

```
0:main> flshmode
Usage : flshmode auto
```

6.11 flshread

cmd: flshread

Description: Read content of designated block of Serial Flash/EEPROM.

Syntax: flshread <begin_addr> [- end_addr | num_bytes]

All parameters are in the current radix form. See radix command for details. One type of flashes has "holes" in their addressing space. When "num_bytes" parameter is used, the command skips the holes as necessary to display as many bytes as specified. When "-end_addr" is used, only the bytes (except for those in the holes) within the range will be displayed.

Example:

1. Set number base to hex, then read and display 0x48 (72) bytes of the serial flash content from location from 0xe0. Notice the address skip to bypass the "hole."

```
0:main> radix 16
0:main> flshread e0 48
*** Dump Serial Flash (Auto Mode) ***
0000e0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 6ea64608
000100: 82250042 726f6164
000200: 3c1cc000 24020060 af80007c af820070 24020050 af820078 af80680c 8f826804
```

2. Set number base to hex then read content from locations 0xe0 to 0x220. Again, the hole is skipped.

```
0:main> radix 16
0:main> flshread e0-220
*** Dump Serial Flash (Auto Mode) ***
0000e0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 6ea64608
000100: 82250042 726f6164
000200: 3c1cc000 24020060 af80007c af820070 24020050 af820078 af80680c 8f826804
```

6.12 flshwrite

cmd: flshwrite

Description: Write data to designated block of Serial Flash/EEPROM.

Syntax: flshwrite <begin_addr> [- end_addr][value]

If "value" is not given, it is assumed to be zero. The value will not be written on locations that are not addressable (holes).

Example:

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

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

3. Set number base to hex, write 0x0 to serial flash/EEPROM location 0x25

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

6.13 flshprg

cmd: flshprg

Description: Program designated Serial Flash/EEPROM block via an input file. The input file needs to be found in the same location as bcmediag.exe.

Syntax: flshprg –f=<file_name> [-o=<offset>][-l=<length>]

The <file_name> is the name of the input file. The <offset> is the starting address where the content of the file is loaded, and the default is 0. The <length> is the number of bytes loaded from the file (must be less than the file size), and the default is the size of the file.

Example:

1. Program Serial Flash/EEPROM via input file flshprg.txt

```
0:main> flshprg –f=flshprg.txt
```

6.14 cpudrt

cmd: cpudrt

Description: Read and display RX CPU trace

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:main> 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:main> 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
```

6.15 cpudtt

cmd: cpudtt

Description: Read and display TX CPU trace

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:main> 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:main> 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
```

6.16 cputest

cmd: cputest

Description: TX / RX CPU Test. This test need cpu.bin in the same location as bcmediag.exe.

Syntax: cputest <-n=a | -f=filename> | <-?> for help

-n : number of iterations

-f : cpu.bin

Example:

1. Running CPU Test two times.

```
0:main> cputest    -n=2
```

```
Iteration   : 1
Testing CPU ... OK
Iteration   : 2
Testing CPU ... OK
```

2. Display Help.

```
0:main> cputest -?
Usage : cputest -n=a -f=filename
   -n : number of iterations
   -f : firmware file
```

6.17 dmar

cmd: dmar

Description: Setup DMA Host Memory to NIC memory

Syntax: dmar: <-a=address | -l=length | -p=pattern | -h | -b | -w | -f> | <-?> for help

-a : NIC address to DMA data to

-l : Length of DATA in bytes to DMA

-p : Pattern of Data. 0 - increment

1- decrement

2 - FF's

3 - 00's

4 - AA 55...

5 - 55 AA...

6 - FFFFFFFF 000000000 FFFFFFFF 00000000

7 - FFFFFFFFFFFFFFFF 0000000000000000 FFFFFFFFFFFFFFFF

8 - FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF 0000000000000...

-h : Use high priority DMA Read.

-b : Byte Swap

-w : Word Swap

-f : force to 32-bit bus

Example:

1. Get valid NIC address, then set up DMA host memory to NIC memory. Using high priority DMA Read and enable byte swap.

```
0:main> dmar

Valid NIC address is 0x00000000-0x0001ffff and exclude 0x00002000-0x000020c0
0:main> dmar -a=0 -l=100 -p=4 -h -b  -w -6
Host Address :  0x001422a0
NIC Address  :  0x00000000
Length       :  0x0100
Priority     :  High
Byte Swap    :  Yes
Word Swap    :  Yes
DMAing from Host memory to NIC memory ... OK
```

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

```
0:main> dmar -a=0 -l=100
Host Address :  0x001422a0
NIC Address  :  0x00000000
Length       :  0x0100
Priority     :  Low
Byte Swap    :  No
Word Swap    :  No
```

3. Display Help.

```
0:main> dmaw -?
Usage : dmar -a=address -l=length -p=pattern -h -b -w -f
   -a : NIC address to DMA data to.
   -l : Length of DATA in bytes to DMA.
   -p : Pattern of Data. 0 - byte increment ; 1- byte decrement
        2 - FF's ; 3 - 00's ; 4- AA 55 ... ; 5 - 55 AA ...
        6 - FFFFFFFF 000000000 FFFFFFFF 00000000
        7 - FFFFFFFFFFFFFFFF 0000000000000000 FFFFFFFFFFFFFFFF
        8 - FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF 0000000000000...
        9 - 00000000000000000000000000000000 FFFFFFFFFFFF...
        a - Word Increment ; b - Dword Increment
        c - Word Decrement ; d - Dword Decrement
   -h : Use high priority DMA Read.
   -b : Byte Swap
   -w : Word Swap
   -f : Force to use 32-bit
```

6.18 dmaw

cmd: dmaw

Description: Setup DMA NIC Memory to HOST memory

Syntax; dmaw <-a=address | -l=length | -h | -b | -w> | <-? > for help

-a : NIC address to DMA data from

-l : Length of DATA in bytes to DMA

-h : Use high priority DMA Write

-b : Byte Swap

-w : Word Swap

-d : Dump content of Host Memory

-f : Force to use 32-bit bus

Example:

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

```
0:main> dmaw -a=1 -l=10 -h -b-d -f
Host Data :
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

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

```
0:main> dr  maw -a=0 -l=10
Host Address :  0x003421f8
NIC Address  :  0x00000000
Length       :  0x0010
Priority     :  Low
Byte Swap    :  No
Word Swap    :  No
DMAing from NIC memory to Host memory ... OK
Host Data :
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

3. Display Help.

```
0:main> dmaw -?
Usage : dmaw -a=address -l=length -h -b -w
   -a : NIC address to DMA data from.
   -l : Length of DATA in bytes to DMA.
   -h : Use high priority DMA Write.
   -b : Byte Swap
       -w : Word Swap
   -d : Dump content of Host Memory
   -f : Force to use 32-bit bus
```

6.19 dma_h

cmd: dma_h

Description: Display DMA entries in HEX

Address range: 0x2000 – 0x3FFFF

Syntax: dma_h <begin_addr> [- end_addr | num_bytes]

Example:

1. Read DMA content start from location 0x2000 to 0x2040.

```
0:main> dma_h 0  2000 2  -2040
002000:  00000000  00000000  00000000  00000000  00000000  00000000  00000000
00000000
002020:  00000000  003421f8  00000000  10070010  00020004  deadbeef  deadbeef
deadbeef
002040: 00000000
```

2. Read 40 bytes DMA content start from location 0x2000.

```
0:main> dma_h 2000 40
002000:  00000000  00000000  00000000  00000000  00000000  00000000  00000000
00000000
002020: 00000000
```

6.20 dma_d

cmd: dma_d

Description: Display DMA entries with decode

Syntax: dma_d <begin_addr> [- end_addr | num_bytes]

Example:

1. Read and decode DMA content from location 0x2000 to 0x2040.

```
0:main> dma_d 2000 -2040

**** DMA entry @ 0x2000 ****
Host Address : 00000000:00000000
NIC Address  : 0x00000000
Complete Q   : Unknown
Source   Q   : Unknown
Length       : 0
Flags        : 0x00000000
Opaque Data  : 0x00000000 0x00000000 0x00000000

**** DMA entry @ 0x2020 ****
Host Address : 00000000:003421F8
NIC Address  : 0x00000000
Complete Q   : Rx Data Complete Q
Source   Q   : DMA High Priority WQ
Length       : 16
Flags        : 0x00020004
Opaque Data  : 0xdeadbeef 0xdeadbeef 0xdeadbeef
```

2. Read and decode 40 bytes DMA content start from location 0x2000.

```
0:main> dma_d 2000 40

**** DMA entry @ 0x2000 ****
Host Address : 00000000:00000000
NIC Address  : 0x00000000
Complete Q   : Unknown
Source   Q   : Unknown
Length       : 0
Flags        : 0x00000000
Opaque Data  : 0x00000000 0x00000000 0x00000000
```

6.21 dmatest

cmd: dmatest

Description: DMA Test

Syntax: dmatest <-n=iteration | –l | –a | –f> | <-?> for help

-n : Number of iteration. 0 means forever

-l : Length of data to do DMA

-a : NIC address

-f : Force to use 32-bit bus

Example:

1. Run DMA test.

```
0:main> dmatest
**** Testing low priority DMA ***
DMAing HOST (@0x003421f8) to NIC (@0x02100) length = 0x400  ... OK
DMAing NIC (@0x02100) to HOST (@0x003421f8) length = 0x400  ... OK
Checking data contents ... OK
**** Testing High priority DMA ***
DMAing HOST (@0x003421f8) to NIC (@0x00002100) length = 0x400  ... OK
DMAing NIC (@0x02100) to HOST (@0x003421f8) with length = 0x400  ... OK
Checking data contents ... OK
```

2. Display Help.

```
0:main> dmatest    -?
Usage : dmatest -n=iteration -l=length -a=NIC_address -f
   -n : Number of interation. 0 means forever
   -l : Length of data to do DMA
   -a : NIC address
   -f : Force to use 32-bit bus
```

6.22 txcfg

cmd: txcfg

Description: Configure transmits packet protocol

Syntax: txcfg

Example:

```
0:main> txcfg
 1. Source MAC                                  : 00:01:02:03:04:05
 2. Destination MAC                             : 10:11:12:13:14:15
 3. Length (14-1518)                            : 1514
 4. Packet Type { EthV2(1), 802.3(2), SNAP(3)}  : Ethernet II
 5. Protocol Field { IP(800) }                  : 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 { Inc(1), Random(2), 0s(3), FFs(4)       : Increment (00,01,02
...)
12. IP Checksum Offload{ YES(1), NO(2) }             : NO
13. TCP/UDP Checksum Offload { YES(1), NO(2) }       : NO
14. TCP/UDP Pseudo Checksum Only { YES(1), NO(2) }: NO
15. Insert VLAN Tag { YES(1), NO(2) }                : NO
16. VLAN Tag                                         : 1
 0. Exit
Enter your choice (option=paramter) :
```

6.23 txpkt

cmd: txpkt

Description: Transmit Packets. Driver must be loaded.

Syntax: txpkt <-n=a | -f=b | -r=c | -i=d | -s=e | -d=f | -c=g | -m=h | -p=i | -q=j> | <-?> for help

-a : Number of packets to TX (0 means forever)

-b : Fragment size

-c : TX ring number

-d : 1 means use incremental length ; 0 means otherwise

-e : Start packet length

-f : Interpacket delay

-g : TX Flags

-h : 1 means multiple TX ring test; 0 means otherwise

-i : Number of TX rings to use in multiple ring test

-j : Number of Packets per ring

Example:

1. Transmitting one packet. Driver must be loaded.

```
0:main> loaddrv
Reinitializing PCI Configuration Space
Bus Number       : 0
Device/Funtion   : 14/0
Base Address     : 0xf4000004
IRQ              : 10
Bringing up MAC driver ... OK
```

```
0:main> txpkt -n=1
 ***************** Blasting Packets *****************
Packets to Transmit : 1           Tx Ring Number    : 0
Source MAC          : 00:01:02:03:04:05
Destination MAC     : 10:11:12:13:14:15
Packet Type         : Ethernet II
Pattern             : Increment (00,01,02 ...)
Interpacket Delay   : 0           Fragment Size          : 1514
Incremental Length  : Y           Start Length           : 60
IP Checksum Offload : N           TCP/UDP Chksum Offload : N
Pseudo-Chksum Only  : N
VLAN Tag Insertion  : N           VLAN Tag               : 1

 ************************************************
Transmitting packet  :               1 (   60 bytes)
```

2. Display Help.

```
0:main> txpkt -?
Usage : txpkt -n=a -f=b -r=c -i=d -s=e -d=f -c=g
              -m=h -p=i -q=j
   a : Number of packets to tx (0 means forever)
   b : Fragment size
   c : Tx ring number
   d : 1 means use incremental length ; 0 means otherwiese
   e : Start packet length
   f : Interpacket delay
   g : Tx Flags
   h : 1 means multiple Tx ring test; 0 means otherwise
   i : Number of Tx rings to use in multiple ring test
   j : Number of Packets per ring
```

6.24 rxcfg

cmd: rxcfg

Description Configure RX parameters.

Syntax: rxcfg

Example:

```
0:main> rxcfg
 1. Host Loopback { Enable(1), Disable(2) }       : Disable
 2. Modify Rx Packet { Enable(1), Disable(2) }    : 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(2) }     : Enable
10. Capture Rx Pacpket { Enable(1), Disable(2) }  : Enable
 0. Exit
```

6.25 stsblk

cmd: stsblk

Description: Display Statistics Block.

Syntax: stsblk

Example:

```
0:main> stsblk
************** STATISTICS Block @ 0x0027c0c0 *************
ifHCInOctets                    :           0   etherStatsFragments     :           0
ifHCInUcastPkts                 :           0   ifHCInMulticastPkts     :           0
ifHCInBroadcastPkts             :           0   d3StatsFCSErrors        :           0
d3StatsAlignmentErrors          :           0   xonPauseFramesReceived  :           0
xoffPauseFramesReceived         :           0   macControlFramesReceived:           0
xoffStateEntered                :           0   dot3StatsFramesTooLong  :           0
etherStatsJabbers               :           0   etherStatsUndersizePkts :           0
inRangeLengthError              :           0   outRangeLengthError     :           0
etherStatsPkts64Octets          :           0   etherStatsPkts65-127    :           0
etherStatsPkts128-255           :           0   etherStatsPkts256-511   :           0
etherStatsPkts512-1023          :           0   etherStatsPkts1024-1522 :           0
etherStatsPkts1523-2047         :           0   etherStatsPkts2048-4095 :           0
etherStatsPkts4096-8191         :           0   etherStatsPkts8192-9022 :           0
ifHCOutOctets                   :           0   etherStatsCollisions    :           0
outXonSent                      :           0   outXoffSent             :           0
flowControlDone             :           0   d3StatsIntlMacTxErrors  :           0
d3StatsSingleColFrames      :           0   d3StatsMultipleColFrames:           0
dt3StatsDeferredTx          :           0   d3StatsExcessiveCol     :           0
d3StatsLateCol              :           0   d3Collided2Times        :           0
d3Collided3Times            :           0   d3Collided4Times        :           0
d3Collided5Times            :           0   d3Collided6Times        :           0
d3Collided7Times            :           0   d3Collided8Times        :           0
d3Collided9Times            :           0   d3Collided10Times       :           0
d3Collided11Times           :           0   d3Collided12Times       :           0
d3Collided13Times           :           0   d3Collided14Times       :           0
d3Collided15Times           :           0   ifHCOutUcastPkts        :           0
d3StatsCarSenseErrors       :           0   ifOutDiscards           :           0
COSIfHCInPkts[00]           :           0   COSIfHCInPkts[01]       :           0
COSIfHCInPkts[02]           :           0   COSIfHCInPkts[03]       :           0
COSIfHCInPkts[04]           :           0   COSIfHCInPkts[05]       :           0
COSIfHCInPkts[06]           :           0   COSIfHCInPkts[07]       :           0
COSIfHCInPkts[08]           :           0   COSIfHCInPkts[09]       :           0
COSIfHCInPkts[10]           :           0   COSIfHCInPkts[11]       :           0
COSIfHCInPkts[12]           :           0   COSIfHCInPkts[13]       :           0
COSIfHCInPkts[14]           :           0   COSIfHCInPkts[15]       :           0
COSFrmsDxDueToFilters       :           0   nicDmaWriteQueueFull    :           0
nicDmaWrHiPQFull            :           0   nicNoMoreRxBDs          :           0
ifInDiscards                :           0   ifInErrors              :           0
nicRecvThresholdHit         :           0   nicDmaReadQueueFull     :           0
COSIfHCOutPkts[00]          :           0   COSIfHCOutPkts[01]      :           0
COSIfHCOutPkts[02]          :           0   COSIfHCOutPkts[03]      :           0
COSIfHCOutPkts[04]          :           0   COSIfHCOutPkts[05]      :           0
COSIfHCOutPkts[06]          :           0   COSIfHCOutPkts[07]      :           0
COSIfHCOutPkts[08]          :           0   COSIfHCOutPkts[09]      :           0
COSIfHCOutPkts[10]          :           0   COSIfHCOutPkts[11]      :           0
COSIfHCOutPkts[12]          :           0   COSIfHCOutPkts[13]      :           0
COSIfHCOutPkts[14]          :           0   COSIfHCOutPkts[15]      :           0
nicDmaRdHPQueueFull         :           0   nicSendDataCompQueueFull:           0
nicRingSetSdPIdx            :           0   nicRingStatusUpdate     :           0
nicInterrupts               :           0   nicAvoidedInterrupts    :           0
nicSendThresholdHit         :           0
```

6.26 statusblk

cmd: statusblk

Description: Display Status Block

Syntax: statusblk

Example:

```
0:main> 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
```

6.27 reset

cmd: reset

Description: Reset Chip

Syntax: reset

Example:

```
0:main> reset
Resetting Chip ... OK
```

6.28 phyctrl

cmd: phyctrl

Description: Configure Phy Speed

Syntax: phyctrl <-s=a | -m=b | -i=c | -f=d> | <-?> for help

-a : 0 - 10 Mbps.

1 - 100 Mbps.

2 - 1000 Mbps.

3 - Auto Negotiation.

-b : 0 - full duplex.

1 - half duplex.

-c : 1 - reset PHYs

-d : file contains initialization scripts

Example:

1. Configure Phy into Auto Negotiation, full duplex mode.

```
0:main> physctrl -s=3 -m=0 -i=1
Resetting PHY ... OK
PHY ID          : 0x0020 - 0x6051
PHY Description : BCM5401 Rev#1
Initializing registers (work-around for BCM5401) ... Done
Configure MAC and PHY to ... MII/Full/Auto Negotiation mode
```

2. Display Help.

```
0:main>phyctrl -?
Usage : phyctrl -s=a -m=b -i=c -f=d
   a  : 0 - 10 Mbps.
        1 - 100 Mbps.
        2 - 1000 Mbps.
        3 - Auto Negotiation.
   b  : 0 - full duplex.
        1 - half duplex.
   c  : 1 - reset PHYs
   d  : file contains initialization scripts
```

6.29 maclpb

cmd: maclpb (B0 only)

Description: Enable or Disable MAC loop back

Syntax: maclpb <-m=a> | <-?> for help

-a : 0 means disable. Otherwise enable

Example:

1. Driver must be loaded before configure.

```
0:main> loaddrv
Reinitializing PCI Configuration Space
Bus Number       : 0
Device/Funtion   : 14/0
Base Address     : 0xf4000004
IRQ              : 10
Bringing up MAC driver ... OK
```

2. Enable MAC loop back.

```
0:main> mcaclpb -m=1
Enabling MAC loopback ... OK
```

2. Disable MAC loop back.

```
0:main> maclpb -m=0
Disabling MAC loopback ... OK
```

3. Display Help.

```
0:main> maclpb -?
Usage : maclpk -m=a
   a : 0 means disable. Otherwise enable.
```

6.30 mread

cmd: mread

Description: Read PHY registers via MII

Syntax: mread <begin_addr> <- end_addr >

Address range: 0x00 – 0x1F

Example:

1. Read MII register 0

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

2 Read MII registers 0 to 10

```
0:main> mread 0-10
00:  1100 7949 0020 6051 01e1 0000 0004 2001
08:  0000 0300 0000 0000 0000 0000 0000 3000
10:  0002
```

4. Read 5 MII registers start from register 0

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

6.31 mwrite

cmd: mwrite

Description: Write PHY registers via MII

Syntax: mwrtie <begin_addr> <- end_addr > <value>

Address range: 0x00 – 0x1F

Example:

1. Write 0x15 to MII register 2

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

2. Write 0x15 to MII register start from register 2 to 10

```
0:main> mwrite 2-10 15
```

6.32 mdev

cmd: mdev

Description: Current Phy Selection. The default device ID is 0x01.

Syntax: mdev <phy_id>

Example:

```
0:main> mdev 1
```

6.33 miimode

cmd: miimode

Description: MII auto or manual mode select

Syntax: miimode <auto> | <manu>

Example:

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

6.34 miitest

cmd: miitest

Description: PHY registers read write test

Syntax: miitest <-n=a> | <-?> for help

-a : number of iterations

Example:

1. Running MII Test.

```
0:main> miitest -n=1
PHY ID          : 0x0020 - 0x6051
PHY Description : BCM5401 Rev#1
Iteration   : 1
Testing 15 MII Registers ... OK
```

2. Display Help.

```
0:main> miitest -?
Usage : miitest -n=a
   a : Number of iterations (0 means forever)
```

6.35 read

cmd: read

Description: Generic Memory Read

Syntax: read [@#*^]<begin_addr> [- end_addr | num_bytes]

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

= Registers (default) (address range: 0x00 – 0x1FFFF)

* = SRAM (address range: 0x00 –0x1FFFF)

^ = internal scratchpad (address range: 0x3000 – 0x37FFFF)

d = direct access

Example:

1. Read from Configuration space

```
0:main> read @10
000010: f4000004
```

2. Read from Register

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

3. Read from SRAM

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

4. Read from internal scratchpad

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

6.36 write

cmd: write

Description: Generic Memory Write

Syntax: write [@#*$%^]<begin_addr> [- end_addr] <value>

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

= Registers (default) (address range: 0x00 – 0x1FFFF)

* = SRAM (address range: 0x00 –0x1FFFF)

^ = Internal scratchpad (address range: 0x3000 – 0x37FFFF)

d = direct access

Example:

1. Write to configuration space.

```
0:main> write @10 f4000004
```

2. Write to register.

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

3. Write to SRAM

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

4. Write to internal scratchpad

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

6.37 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 <-n=a | -s | -b | -d | -m | -e> | <-?> for help

-n : Number of iterations (0 means forever)

-s : Test Scratch Pad (0x30000-0x37fff)

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

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

-m : Test MBUF SRAM (0x8000-0x1ffff)

-e : Test External Memory (0x20000-0xXXXXXXX)

-x: Test MBUF SRAM via DMA

-c: Test MBUF SRAM (special test) *

* -c option is useful to detect memory problem in A1 silicon. Write once and read the adjacent memory location.

Example:

1. Unload driver if driver loaded, run memory test 1 time. Scratch Pad, DMA blocks had been selected in this example.

```
0:main> unloaddrv
Unloading MAC driver ... OK
0:main> memtest -n=1 -s -d
Putting Rx and Tx CPU to halt
Iteration  : 1

******** Testing DMA SRAM 0x00002000-0x00003fff ********
Pattern Test 0x00000000 ... OK
Alternate Pattern Test 0x00000000 (0xffffffff) ... OK
Pattern Test 0xffffffff ... OK
Alternate Pattern Test 0xffffffff (0x00000000) ... OK
Pattern Test 0xaa55aa55 ... OK
Alternate Pattern Test 0xaa55aa55 (0x55aa55aa) ... OK
Pattern Test 0x55aa55aa ... OK
Alternate Pattern Test 0x55aa55aa (0xaa55aa55) ... OK
Address Test ... OK
Walking One Test ... OK

******** Testing Scratch Pad 0x00030000-0x00037fff ********
Pattern Test 0x00000000 ... OK
Alternate Pattern Test 0x00000000 (0xffffffff) ... OK
Pattern Test 0xffffffff ... OK
Alternate Pattern Test 0xffffffff (0x00000000) ... 0:main>
```

2. Display Help.

```
0:main> memtest -?
Usage : memtest -n=a -s -b -d -m -e
   -n : Number of iterations (0 means forever)
   -s : Test Scratch Pad (0x30000-0x37fff)
   -b : Test BD SRAM (0x0000-0x0fff and 0x4000-0x7fff)
   -d : Test DMA SRAM (0x2000-0x3fff)
   -m : Test MBUF SRAM (0x8000-0x1ffff)
   -e : Test External Memory (0x20000-0xXXXXXXX)
   -x:  Test MBUF SRAM via DMA
   -c:  Test MBUF SRAM (special test)
```

6.38 pmdcfg

cmd: pmdcfg

Description: Display Power Management Info

Syntax: pmdcfg

Example:

```
0:main> pmdcfg
PMCSR          : 0x2100
PM Capability  : 0xc002
Data Scale     : 1
D0 Power Consumed    (00) : 0x00
D1 Power Consumed    (01) : 0x00
D2 Power Consumed    (02) : 0x00
D3 Power Consumed    (03) : 0x00
D0 Power Dissipated  (04) : 0x00
D1 Power Dissipated  (05) : 0x00
D2 Power Dissipated  (06) : 0x00
```

```
D3 Power Dissipated  (07) : 0x00
Common Power Cons.   (08) : 0x00
Reserved             (09) : 0x00
Reserved             (10) : 0x00
Reserved             (11) : 0x00
Reserved             (12) : 0x00
Reserved             (13) : 0x00
Reserved             (14) : 0x00
Reserved             (15) : 0x00
```

6.39 pmpd

cmd: pmpd

Description: Power Down MAC. Input file wol.txt should be found in the same location of bcmediag.exe.

Syntax: pmpd <-f=a | -a=b | -o=c | -m=d | -i=e | -v=f> | <-?> for help

-a : File name which contains patterns

-b : 0 to add a pattern; 1 means otherwise

-c : offset

-d : 1 enables Magic MAC detection

-e : 1 enables ACPI Packet Match

-f : Verbose level

Example:

1. Power down MAC

```
0:main> pmpd -a=0 -o=0 -m=1 -i=1 -v=1

Wake-On-Lan Patterns :

Halting MAC ... OK
Programming patterns to H/W ... OK
Programming ACPI Registers Buf @ 0x00000800 offset = 0 length = 128.
Enable ACPI Pattern Match
Enable Magic MAC detection
2. Display Help
      0:main> pmpd -?
Usage : pmpd -f=a -a=b -o=c -m=d -i=e -v=f
      a  : File name which contains patterns
      b  : 0 to add a pattern; 1 means otherwise.
      c  : offset
      d  : 1 enables Magic MAC detection
      e  : 1 enables ACPI Packet Match
      f      : Versbose level
```

6.40 poll

cmd: poll

Description: Poll ISR and display

Syntax: poll

Example:

```
0:main> poll
                                    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
```

6.41 intr

cmd: intr

Description: Display Interrupt Info

Syntax: intr

Example:

```
0:main> intr
Interrupt Count : 3
IPC MASK        : 0xb8 0x0b
IPC IS1 IS2     : 0x00 0x00
IPC IRR1 IRR2   : 0x00 0x00
IPC ILCR1 ILCR2 : 0x00 0x0e
```

6.42 intrtest

cmd: intrtest

Description: Interrupt Test

Syntax: intrtest <-n=a> | <-?> for help

-a : number iterations

Example:

1. Running interrupt test 2 times.

```
0:main> intrtest -n=2
Iteration   : 1
Testing Interrupt ... OK
Iteration   : 2
Testing Interrupt ... OK
```

2. Display Help

```
0:main> intrtest -?
Usage : intrtest -n=a
   a : Number of iterations (0 means forever)
```

6.43 machalt

cmd: machalt

Description: Halt MAC controller

Syntax: machalt

Example:

```
0:main> machalt
```

6.44 addmc

cmd: addmc

Description: Add Multicast MAC

Syntax: addmc address0,....,address5

Example:

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

6.45 delmc

cmd: delmc

Description: Delete Multicast MAC

Syntax: delmc address0,....,address5

Example:

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

6.46 ftq

cmd: ftq

Description: Display FTQ info

Syntax: ftq

Example:

```
0:main> 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
```

6.47 mbuf

cmd: mbuf

Description: Display Content of MBUFs

Syntax: mbuf <-c=a | -m=b | -n=c> | <-?> for help

-a : 0 - displays a MBUF

1 - displays a MBUF chain

2 - displays general MBUF information

3 - displays MBUF Cluster

-b : 0 - decode a MBUF

1 - display MBUF in hex

-c : MBUF number to display/decode

Example:

1. Display MBUF.

```
0:main> mbuf -c=0
***** MBUFS 0x100 @ 0x00008000 *****
00008080 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
Chain  : 0    Frame : 0   Next MBUF : 0x0101   Len : 0
Next Frame Pointer : 0x00000000
Data:
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000
```

2. Display MBUF chain.

```
0:main> mbuf -c=1
->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
```

3. Displays general MBUF information.

```
0:main> mbugf -c=2
MBUF Pool Base     : 0x8000
MBUF Pool Length   : 0x18000
MBUF Free Head/Tail: 0x0143/0x03ff
MBUF Free          : 701
MBUF Chains        : 0
MBUF Crosslinks    : 0
```

4. Display Help.

```
0:main> mbuf -?
Usage : mbuf -c=a -m=b -n=c
   a : 0 - displays a MBUF
       1 - displays a MBUF chain
       2 - displays general MBUF information
             3 - displays MBUF Cluster
   b : 0 - decode a MBUF
       1 - display MBUF in hex
   c :  MBUF number to display/decode
```

6.48 loaddrv

cmd: loaddrv

Description: Load NIC driver

Syntax: loaddrv

Syntax: loaddrv <-b | <-?> for help

-b : option to allocate small buffer without straddling a 4k boundary

Example:

```
0:main> loaddrv
Reinitializing PCI Configuration Space
```

```
Bus Number       : 0
Device/Funtion   : 14/0
Base Address     : 0xf4000004
IRQ              : 10
Bringing up MAC driver ... OK
```

6.49 unloaddrv

cmd: unloaddrv

Description: Unload NIC driver

Syntax: unloaddrv

Example:

```
0:main> unloaddrv

Unloading MAC driver ... OK
```

6.50 loadfw

cmd: loadfw

Description: Load Firmware to TX & RX CPUs

Syntax: loadfw <-f=filename | -r | -t> | <-?> for help

-f : firmware file

-r : Load firmware to RX-CPU

-t : Load firmware to TX-CPU

Example:

```
0:main> loadfw -?
Usage : loadfw -f=filename -r -t
   -f : firmware file
   -r : Load firmware to RX-CPU
   -t : Load firmware to TX-CPU
```

6.51 nictest

cmd: nictest

Description: NIC test includes memory test, serial eeprom test, interrupt test, packet exchange, MAC registers test, Mii registers test, cpu test, dma test. This test need to be configured by running "diagcfg". See "diagcfg" for detail.

Syntax: nictest <-n=a> | <-?> for help

-a : number interations

Example:

1. Run "diagcfg" to configure NIC test (Manufacturing test) parameter. Only DMA test and CPU test are covered in this example.

```
0:main> diagcfg
             Diagnostics Configuration Menu
   1. Memory Test Configuration Menu
   2. Manufacturing Test Configuration Menu
   3. Driver Configuration Menu
   4. Abort On Failure {Yes(1), No(0)}          : Yes
   5. Verbose Level (0..6)                      : 2


Enter your choice (option=paramter/save/cancel) :2
Manufacturing Test Configuration Menu

 1. Register Test { Enable(1), Disable(2) }              : Disable
 2. Memory Test { Enable(1), Disable(2) }                : Diable
 3. Serial EEPROM Test { Enable(1), Disable(2) }         : Disable
 4. Interrupt Test { Enable(1), Disable(2) }             : Disable
 5. CPU Test { Enable(1), Disable(2) }                   : Enable
 6. DMA Test { Enable(1), Disable(2) }                   : Enable
 7. VPD Test { Enable(1), Disable(2) }                   : Disable
 8. MII Test { Enable(1), Disable(2) }                   : Disable
 9. Packet Tx/Rx { Enable(1), Disable(2) }               : Disable
 0. Exit to previous menu

Enter your choice (option=paramter) : 0
```

2. After making selection, type "save" to save setting. Then run NIC Test.

```
0:main> nictest
<<<< DMA Test >>>>
**** Testing low priority DMA ***
DMAing HOST (@0x003421f8) to NIC (@0x02100) length = 0x400  ... OK
DMAing NIC (@0x02100) to HOST (@0x003421f8) length = 0x400  ... OK
Checking data contents ... OK
**** Testing High priority DMA ***
DMAing HOST (@0x003421f8) to NIC (@0x00002100) length = 0x400  ... OK
DMAing NIC (@0x02100) to HOST (@0x003421f8) with length = 0x400  ...
OK
Checking data contents ... OK
    >>>> Passed <<<<
    <<<< CPU Test >>>>
Testing CPU ... OK
    >>>> Passed <<<<
*********************************************************************
      Test Result         Passed : 1      Failed : 0
*********************************************************************
Register Test             : Not Tested
Memory Test               : Passed
DMA Test                  : Passed
Serial EEPROM Test        : Not Tested
MII Test                  : Not Tested
Interrupt Test            : Not Tested
CPU Test                  : Passed
VPD Test                  : Not Tested
Packet Tx/Rx Test         : Not Tested
3. Display Help.
```

```
0:main> nictest -?
Usage : nictest -n=a
   a : number interations.
0:main> nictstest –c=1
```

6.52 nicstats

cmd: nicstats

Description: Display NIC test statistics

Syntax: nicstats <-c>

-c : Clear Statistics

Example: Load driver if driver is not loaded.

```
0:main> loaddrv
Reinitializing PCI Configuration Space
Bus Number       : 0
Device/Funtion   : 14/0
Base Address     : 0xf4000004
IRQ              : 10
Bringing up MAC driver ... OK
0:main> nicstats
                                    Total                Rate
                                 ==========      ==========
Txed Packets (Ring#0)     :             0                0
Txed Packets (Ring#1)     :             0                0
Txed Packets (Ring#2)     :             0                0
Txed Packets (Ring#3)     :             0                0
Tx Packets Enqed (Ring#0) :             0                0
Tx Packets Enqed (Ring#1) :             0                0
Tx Packets Enqed (Ring#2) :             0                0
Tx Packets Enqed (Ring#3) :             0                0
Rxed Packets (Ring00)     :             0                0
Rxed Packets (Ring01)     :             0                0
Rxed Packets (Ring02)     :             0                0
Rxed Packets (Ring03)     :             0                0
Rxed Packets (Ring04)     :             0                0
Rxed Packets (Ring05)     :             0                0
Rxed Packets (Ring06)     :             0                0
Rxed Packets (Ring07)     :             0                0
Rxed Packets (Ring08)     :             0                0
Rxed Packets (Ring09)     :             0                0

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

Total                   Rate
                     ==========      ==========
Rxed Packets (Ring#10)    :             0                0
Rxed Packets (Ring#11)    :             0                0
Rxed Packets (Ring#12)    :             0                0
Rxed Packets (Ring#13)    :             0                0
Rxed Packets (Ring#14)    :             0                0
Rxed Packets (Ring#15)    :             0                0
Rxed CRC-32 Errors        :             0                0
```

```
Out of Memory            :                0                      0
Too Many Frag Pkt        :                0                      0


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

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


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

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


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

PageUP/PageDN to scroll. Ins/Del toggles refresh. ESC to exit
                       CHIP Statistics
=============================================================================
ifInDiscards           :          0   ifInErrors              :          0
nicRecvThresholdHit    :          0   nicDmaReadQueueFull     :          0
COSIfHCOutPkts[00]     :          0   COSIfHCOutPkts[01]      :          0
COSIfHCOutPkts[02]     :          0   COSIfHCOutPkts[03]      :          0
COSIfHCOutPkts[04]     :          0   COSIfHCOutPkts[0
```

```
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
```

6.53 regtest

cmd: regtest

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

Syntax: regtest < -n=a> | <-?> for help

-a : number of iterations

Example:

1. Running register test.

```
0:main> unloaddrv
Unloading MAC driver ... OK
0:main> regtest    -n=1
Iteration   : 1
Testing 483 Registers ... OK
```

2. Display Help.

```
0:main> regtest -?
Usage : regtest -n=a
   a : number of iterations
```

6.54 debug

cmd: debug

Description: Display debugs information

Syntax: debug <-c=a> | <-?> for help

-a= 1 : Dump TX / RX Stats

Example:

1. Display debug information.

```
0:main> debug -c=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
2.  Display Help.
0:main> debug -?
Usage : debug -c=a
   1 : Dump Tx/Rx Stats
```

6.55 pciscan

cmd: pciscan

Description: Scan for all PCI Devices

Syntax: pciscan

Example:

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

6.56 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. Manufacturing Test Configuration Menu
 3. Driver Configuration Menu
 4. Abort On Failure {Yes(1), No(0)}          : Yes
 5. Verbose Level (0..6)                      : 2

Enter your choice (option=paramter/save/cancel) :

                  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 (0x00008000-0x0001ffff)   : 00008000
 8. SRAM MBUF End (0x00008000-0x0001ffff)     : 0001ffff
 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
 0. Exit to previous menu

Enter your choice (option=paramter) :



             Diagnostics Configuration Menu

 1. Memory Test Configuration Menu
 2. Manufacturing Test Configuration Menu
 3. Driver Configuration Menu
 4. Abort On Failure {Yes(1), No(0)}          : Yes
 5. Verbose Level (0..6)                      : 2

Enter your choice (option=paramter/save/cancel) :

             Manufacturing Test Configuration Menu

 1. Register Test { Enable(1), Disable(2) }              : Enable
 2. Memory Test { Enable(1), Disable(2) }                : Enable
 3. Serial EEPROM Test { Enable(1), Disable(2) }         : Enable
 4. Interrupt Test { Enable(1), Disable(2) }             : Enable
 5. CPU Test { Enable(1), Disable(2) }                   : Enable
 6. DMA Test { Enable(1), Disable(2) }                   : Enable
 7. VPD Test { Enable(1), Disable(2) }                   : Disable
 8. MII Test { Enable(1), Disable(2) }                   : Enable
 9. Packet Tx/Rx { Enable(1), Disable(2) }               : Disable
 0. Exit to previous menu

Enter your choice (option=paramter) :



             Diagnostics Configuration Menu

 1. Memory Test Configuration Menu
 2. Manufacturing Test Configuration Menu
 3. Driver Configuration Menu
 4. Abort On Failure {Yes(1), No(0)}          : Yes
 5. Verbose Level (0..6)                      : 2

Enter your choice (option=paramter/save/cancel) :

                 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. Queue Rx Packets                    : 1
14. Tx Copy Buffer Size                 : 64
15. External Memory Exists              : 0
16. MBUF Base                           : 0x008000
17. MBUF Length                         : 0x018000
18. MBUF WorkAround                     : 0
 0. Exit to previous menu
```

6.57 log

cmd: log

Description: Save all output to log file (ttp.log)

Syntax: log

Example:

```
0:main> log
started logfile 'bcmediag.log'
```

6.58 nolog

cmd: nolog

Description: Closes Log file

Syntax: nolog

Example:

```
0:main> nolog
logfile closed
```

6.59 radix

cmd: radix

Description: Set the base of input number.

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

Example: Set base of input number to hex

```
0:main> radix 16
0:main> radix
current input radix = 16
```

6.60 up

cmd: up

Description: Go up one level of menu

Syntax: up

Example:

```
0:main> up
```

6.61 exit

cmd: exit

Description: Exit diagnostic

Syntax: exit

Example:

```
0:main> exit
```

6.62 blast

cmd: blast

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

Syntax: blast -l=<length> -t -r -h -c=<num_buf> -k -s

-l : Length of Tx packet (Default = 64)

-t : Enable Tx

-r : Enable Rx

-h : Enable Host Loopback *

-c : Number of Tx buffer (Default = 100)

-k : Applies CRC-32 check on Rx path

-s : Stop on Failure

*Workaround A1 memory Problem: In order to use –h option do the followings.

1. loaddrv
2. mbuf –w=1
3. blast –h

4. Start SmartBit to inject traffic. Watch for CRC Error indication.

Example:

1. Load MAC driver and enable transmission.

```
0:packet> loaddrv
PHY ID          : 0x0020 - 0x6051
PHY Description : BCM5401 Rev#1
Configuring BCM5401 ... Done
Reinitializing PCI Configuration Space
Bus Number      : 0
Device/Funtion  : 14/0
Base Address    : 0xf4000004
IRQ             : 10
Bringing up MAC driver ... OK
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
```

2. Display Help.

```
0:packet> blast -?
Usage : blast -l=<length> -t -r -h -c=<num_buf> -k -s
   -l : Length of Tx packet (Default = 64)
   -t : Enable Tx
   -r : Enable Rx
   -h : Enable Host Loopback
   -c : Number of Tx buffer (Default = 100)
   -k : Applies CRC-32 check on Rx path
   -s : Stop on Failure
```

6.63 gpiowrite

cmd: gpiowrite

Description: Control Output of GPIO Pin

Syntax: gpiowrite -n=<GPIO_num> -v=<1 | 0>

-n : Specifies GPIO pin (Default = 0)

-v : Value to be written to GPIO pins (Default = 0)

Example:

1. Write 1 to GPIO#1 Pin

```
0:main> gpiowrite -n=1 =  -v=1

Writing 1 to GPIO#1
```

2. Display Help.

```
0:main> gpiowrite -?
Usage : gpiowrite -n=<GPIO_num> -v=<1 | 0>
   -n : Specifies GPIO pin (Default = 0)
   -v : Value to be written to GPIO pins (Default = 0)
```

6.64 gpioread

cmd: gpioread

Description: Get Input of GPIO Pin

Syntax: gpioread

Example:

1. Read GPIO Pins

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

2. Display Help.

```
0:main> gpioread -?
Usage : giopread
```

6.65 loop

cmd: loop

Description: Repeat a command in x iteration

Syntax: loop -n=<iteration> -c=<command>

n : number of iterations

c : command to execute

Example:

1. Repeat "gpioread" 5 times

```
0:misc> loop -n=5 -c=gpioread
Iteration   : 1
GPIO#0 : 1
GPIO#1 : 1
GPIO#2 : 0
Iteration   : 2
GPIO#0 : 1
GPIO#1 : 1
GPIO#2 : 0
Iteration   : 3
GPIO#0 : 1
GPIO#1 : 1
GPIO#2 : 0
Iteration   : 4
GPIO#0 : 1
GPIO#1 : 1
GPIO#2 : 0
Iteration   : 5
GPIO#0 : 1
GPIO#1 : 1
GPIO#2 : 0
```

2. Display Help

```
0:misc> loop -?
Usage : loop -n=<iteration> -c=<command>
   n : number of iterations
   c : command to execute
```

6.66 version

cmd: version

Description: Display Diagnostics Version

Syntax: version

Example:

```
0:misc> version
BCM5700 Engineering Diagnostics (BCM5700) v1.03
```

6.67 ringIndex

cmd:

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

Syntax: ringindex –r –t

Example:

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

```
0:main> loaddrv
```

```
PHY ID          : 0x0020 - 0x6051
PHY Description : BCM5401 Rev#1
Configuring BCM5401 ... Done
Bringing up MAC driver ... OK
0:main> rinh  gindex -r -t

                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
```

2. Display Help.

```
0:main> ringindex -?
Usage : ringindex -r -t
   -r : Dump Rx Ring Index
   -t : Dump Tx Ring Index
```

6.68 dos

cmd: dos

Description: Enter to Dos shell

Syntax: dos

Example:

```
0:main> dos
```

6.69 pxecpy

cmd: pxecpy

Description: Copy PXE code to MBUF memory

Syntax: pxecpy –f=<file>

Example:

```
0:main> pxecpy -?
Usage : pxecpy –f=<file>
   -f : file name
```

6.70 quit

cmd: quit

Description: Exit diagnostic

Syntax: quit

Example:

```
0:main> quit
```

6.71 pciinit

cmd: pciinit

Description: Initialize PCI configuration registers

Syntax: pciinit

Example:

```
0:misc> pciinit
Initializing PCI Configuration Space
Bus Number       : 0
Device/Funtion   : 14/0
Base Address     : 0xf4000004
IRQ              : 10
Broadcom 5700 NIC is detected
```

6.72 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
```

3. Display Help.

```
0:irq> intrctrl -?
Usage : intrctrl -u -m
   u : unmask current interrupt
   m : mask current interrupt
```

6.73 upgfrm

Description: Upgrade boot code firmware or PXE.

Syntax : upgfrm –b –p –f=<filename>

-b : Upgrade boot code firmware

-p : Upgrade PXE code.

-f : Specifies input filename.

Examples:

1. Upgrade boot code firmware from eeprom.bin
   Upgfrm –b -f=eeprom.bin
2. Upgrade PXE code from b57pxe.bin
   Upgfrm –p –f=b57pxe.bin

6.74 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 –m –p –n=<iteration>

-m : Perform MAC loopback test.
-p : Perform PHY loopback test.
-n : Specificies number of iteration.

Examples:
1. Perform MAC and PHY loopback in 3 iterations.
   pkttest –p –m –n=2