



Deconft Tool

Revision: 0.13
Date: 2/8/2007



REVISION HISTORY

REV	DATE	AUTHOR	DESCRIPTION OF CHANGE
0.1	10/10/2006		Initial draft.
0.3	10/13/2006	Kamalesh	Modified for native TCL command changes
0.4	10/20/2006	Prasad	<ul style="list-style-type: none">- Added streaming commands- More information or result reporting.- PLX PCI Bridge Access Tool
0.5	11/03/2006	Kamalesh	Added 'dmtest' command to the 'deconft' tool
0.6	11/12/2006	Kamalesh	<ul style="list-style-type: none">- Removed plxpci tool description- Added bcpci tool description- Modified 'bcdpci' and 'bcwrpci' in deconft tool to take optional size parameter.
0.7	11/16/2006	Kamalesh	<ul style="list-style-type: none">- Added EEPROM test support- Added Appendix for 'Deconf' modes
0.8	11/20/2006	Kamalesh	Input video format support in Decplay command.
0.9	11/28/2006	Kamalesh	FPGA-EPROM Copy engine test command added
0.10	11/04/2006	Kamalesh	Note added for EPROM Scrambler implementation in FPGA
0.11	11/06/2006	Kamalesh	EPROM burn, load bootloader and decoder reset command added

Broadcom Corporation
16215 Alton Parkway
P.O. Box 15703
Irvine, CA 92619-7013

© Copyright 2006 by Broadcom Corporation
All rights reserved
Printed in the U.S.A.





Table of Contents

1.	INTRODUCTION	5
2.	SCOPE.....	5
3.	APPLICABLE DOCUMENTS	5
4.	STATUS RETURN CODES	5
5.	Command List	5
6.	Format	7
6.1.	Command Usage	7
6.2.	Result reporting.....	7
6.3.	Status Definitions:.....	7
7.	Description	8
7.1.	help.....	8
7.2.	devopen	9
7.3.	devclose.....	9
7.4.	getver.....	10
7.5.	hwinfo	10
7.6.	rdpci	11
7.7.	wrpci	11
7.8.	rdreg	12
7.9.	wrreg	12
7.10.	rdfpga	13
7.11.	wrfpga	13
7.12.	rdmem	14
7.13.	wrmem	15
7.14.	draminit	16
7.15.	mtest	17
7.16.	fwinit	18
7.17.	devinit	19
7.18.	tdma.....	19
7.19.	fwcmd	20
7.20.	decplay	21
7.21.	decplays.....	22
7.22.	verbose	23
7.23.	dmtest	24
7.24.	rdeprom	25
7.25.	wreprom	26
7.26.	eptest	27
7.27.	epinit	28
7.28.	epcopy	28
7.29.	epburn	29
7.30.	loadbl.....	29
7.31.	decrst	30
8.	NOTE.....	31
9.	PCI Device Access Tool.....	32



9.1.	Description (Not Supported in this version).....	32
9.2.	Command Description	32
10.	Appendix – A (‘deconf’ modes)	35
10.1.	Diagnostic mode	35
10.2.	Playback mode	35
10.3.	Monitor mode.....	35



1. INTRODUCTION

This document covers syntax of the commands supported in the deconft tool. The tool will be used in the TCL environment for scripting and testing. The commands will be used at the testing place for checking correctness of the BCM70010 High Definition Video Decoder product.

PCI Read/Write Access Tool is also explained in this document.

2. SCOPE

This document covers the deconft command usage on the TCL shell prompt. And this document also covers PCI Read-Write access tool description.

3. APPLICABLE DOCUMENTS

None

4. STATUS RETURN CODES

All deconft commands return two parameters to the TCL interface/shell,

1. Return code – This parameter tells the command successful (0) or failure (!0) status code.
2. Result – This parameter gives the result of the command.

5. Command List

This section tabulates all the commands supported in deconft tool. All these commands are prefixed with “bc” to avoid conflicts with TCL original commands.

Command	Description
help	Command help
devopen	Open Device
devclose	Close Device
getver	Get Software Version
hwinfo	Get Adapter Information
pciconfig	Get PCI Config Information
pcird	Read PCI configuration at given offset
pciwr	Write PCI configuration at given offset
rdreg	Read DEC register at given offset
wrreg	Write DEC register at given offset
rdfpga	Read FPGA-MMIO register at given offset
wrfpga	Write FPGA-MMIO register at given offset
rdmem	Read DEC memory
wrmem	Write DEC memory at given offset
draminit	Initialize DRAM with specific pattern
mtest	Test full DRAM Memory with different patterns
fwinit	Download Firmware to Device
devinit	Initialize Device
tdma	TX-DMA (HostMemory to DRAM) test
fwcmd	Issue FW CMD
decplay	Play and Capture Video
decplays	Play and Capture Video through AES Enabled path





6. Format

The format followed in the document is listed as below

6.1. Command Usage

- [] - Optional Argument
- < > - Argument value to be entered
- Blue font** – Command to be entered at the shell
- Italics* - Information Messages

6.2. Result reporting.

All the command results are reported in the following format with status as the first element in the list. Global TCL error will be set to an error status only if the TCL library interfaces fail. Actual command specific status is communicated as the second element of the result string.

Status { command specific information separated by space }

6.3. Status Definitions:

```
typedef enum _BC_STATUS {  
    BC_STS_SUCCESS           = 0,  
    BC_STS_INV_ARG           = 1,  
    BC_STS_BUSY              = 2,  
    BC_STS_NOT_IMPL          = 3,  
    BC_STS_PGM_QUIT          = 4,  
    BC_STS_NO_ACCESS         = 5,  
    BC_STS_INSUFF_RES        = 6,  
    BC_STS_IO_ERROR          = 7,  
    BC_STS_NO_DATA           = 8,  
    BC_STS_VER_MISMATCH      = 9,  
    BC_STS_TIMEOUT           = 10,  
    BC_STS_FW_CMD_ERR        = 11,  
    BC_STS_DEC_NOT_OPEN      = 12,  
    BC_STS_ERR_USAGE         = 13,  
    BC_STS_IO_USER_ABORT     = 14,  
    BC_STS_IO_XFR_ERROR      = 15,  
    BC_STS_DEC_NOT_STARTED   = 16,  
    BC_STS_FWHEX_NOT_FOUND   = 17,  
    BC_STS_FMT_CHANGE        = 18,  
    BC_STS_HIF_ACCESS        = 19,  
  
    BC_STS_ERROR             = -1  
} BC_STATUS;
```





7. Description

The following sections describe deconft commands.

7.1. help

7.1.1. Description

To display list of deconf command supported and its usage.

7.1.2. Usage

help [<deconf_cmd_name>]

where,

deconf_cmd_name – This option is used to know the syntax of command usage (optional).

7.1.3. TCL Command equivalent

bchelp [<deconf_cmd_name>]

7.1.4. Snap shot

```
% bchelp
help          : Command help
getver        : Get Software Version
hwinfo        : Get Adapter Information
pciconfig     : Get PCI Config Information
rdreg         : Read DEC register
wrreg         : Write DEC register
rdfpga        : Read FPGA-MMIO register
wrfpga        : Write FPGA-MMIO register
rdmem         : Read DEC memory
wrmem         : Write DEC memory
draminit      : Initialize DRAM with specific pattern
fwinit        : Download Firmware to Device
devinit       : Initialize Device
tdma          : TX-DMA (HostMemory to DRAM) test
fwcmd         : Issue FW CMD
decplay       : Play and Capture Video
quit          : Quit deconf
```

% bchelp devinit

```
devinit <SkipFWDownload> <EnableFWInit>
```

```
%
```



7.2. devopen

7.2.1. Description

To open the Decoder device.

This command is present only in “deconft” tool and it is not present in “deconf” console application.

7.2.2. Usage

devopen

7.2.3. TCL Command equivalent

bcdevopen

7.2.4. Snap shot

```
% bcdevopen
0 {}
%
```

7.3. devclose

7.3.1. Description

To close the Decoder device.

This command is present only in “deconft” tool and it is not present in “deconf” console application.

7.3.2. Usage

devclose

7.3.3. TCL Command equivalent

bcdevclose

7.3.4. Snap shot

```
% bcdevclose
0 {}
%
```



7.4. getver

7.4.1. Description

To get software version information.

7.4.2. Usage

getver

7.4.3. TCL Command equivalent

bcgetver

7.4.4. Snap shot

```
% bcgetver  
0 {Version: Driver[0.30.0] Lib [0.27.1] Chip [00040700] FPGA [240000fb] Deconf [0.2.0]}  
%
```

7.5. hwinfo

7.5.1. Description

To get Adapter information.

7.5.2. Usage

hwinfo

7.5.3. TCL Command equivalent

bchwinfo

7.5.4. Snap shot

```
% bchwinfo  
0 {VendorID: 0x00001610 DeviceID: 0x000014E4 Revision: 0x00000000}  
%
```



7.6. rdpci

7.6.1. Description

To read PCI Configuration Register value of the FPGA's PCI interface.

7.6.2. Usage

rdpci <offset> [<size>]

Where,

Offset – Configuration space offset value

Size – Optional parameter. 1 – byte; 2 – 16-bit value; 4 – 32-bit value; other – 32-bit value

7.6.3. TCL Command equivalent

bcrdpci

7.6.4. Snap shot

```
% bcrdpci 0
0 {0x00000000 : 0x161014E4}
% bcrdpci 0 1
0 {0x00000000 : 0x000000E4}
% bcrdpci 0 2
0 {0x00000000 : 0x000014E4}
%
```

7.7. wrpci

7.7.1. Description

To write to PCI Configuration Register of the FPGA's PCI interface.

7.7.2. Usage

wrpci <offset> <value> [<size>]

Where,

Offset – Configuration space offset value

Size – Optional parameter. 1 – byte; 2 – 16-bit value; 4 – 32-bit value; other – 32-bit value

7.7.3. TCL Command equivalent

bcwrpci

7.7.4. Snap shot

```
% bcwrpci 0 0 1
0 {}
%
```



7.8. rdreg

7.8.1. Description

To read from Decoder registers

7.8.2. Usage

rdreg <reg_addr>

7.8.3. TCL Command equivalent

bcrdreg <reg_addr>

7.8.4. Snap shot

```
% bcrdreg 0
0 {0x00000000 : 0x00000000}
% bcrdreg 4
0 {0x00000004 : 0x00000000}
%
```

7.9. wrreg

7.9.1. Description

To write to Decoder register

7.9.2. Usage

wrreg <reg_addr> <32_bit_value>

7.9.3. TCL Command equivalent

bcwrreg <reg_addr> <32_bit_value>

7.9.4. Snap shot

```
% bcwrreg 0 0x3
0 { }
% bcrdreg 0
0 {0x00000000 : 0x00000003}
%
```



7.10. rdfpga

7.10.1. Description

To read from FPGA-MMIO register.

7.10.2. Usage

rdfpga <reg_addr>

7.10.3. TCL Command equivalent

bcrdfpga <reg_addr>

7.10.4. Snap shot

```
% bcrdfpga 404
0 {0x00000404 : 0xFFFFFFFF}
%
```

7.11. wrfpga

7.11.1. Description

To write to FPGA-MMIO register

7.11.2. Usage

wrfpga <reg_addr> <32_bit_value>

7.11.3. TCL Command equivalent

bcwrfpga <reg_addr> <32_bit_value>

7.11.4. Snap shot

```
% bcwrfpga 404 0
0 { }
%
```



7.12. rdmem

7.12.1. Description

To read from Decoder DRAM memory.

Note:

- This command returns list of memory location values without offset information
- This command will return maximum of 2048 bytes of data.
- This command should be run after DRAM initialization (bcdraminit) command.

7.12.2. Usage

rdmem <off> <size> [<filename>]

where,

off – Offset from the base address of DRAM memory
size – Number of bytes to be read (hex value)
filename – File name for dumping purpose (optional)

7.12.3. TCL Command equivalent

bcrdmem <off> <size> [<filename>]

7.12.4. Snap shot

```
% bcrdmem 0 400000 dump.bin
0 { }
%
% bcrdmem 0 40
0 {0xEFFDEFA7 0xFFFFFFFF 0xEFFDEFA7 0xFFFFFFFF
0xEFFDEFA7 0xFFFFFFFF 0xEFFDEFA7 0xFFFFFFFF
0xEFFDEFA7 0xFFFFFFFF 0xEFFDEFA7 0xFFFFFFFF
0xEFFDEFA7 0xFFFFFFFF 0xEFFDEFA7 0xFFFFFFFF}
%
```



7.13. wrmem

7.13.1. Description

To write to Decoder DRAM memory.

Note:

- This command should be run after DRAM initialization (bcdraminit) command.

7.13.2. Usage

wrmem <off> <32_bit_value>

where,

off – Offset from the base address of DRAM memory

7.13.3. TCL Command equivalent

bcwrmem <off> <32_bit_value>

7.13.4. Snap shot

```
% bcwrmem 0 0xabcdef12}
0 { }
% bcrdmem 0
0 {0x00000000 : 0xABCDEF12}
%
```



7.14. draminit

7.14.1. Description

To initialize the DRAM with given pattern.

7.14.2. Usage

draminit <FillPattern> [<Start Offset> [<Size>]]

where,

FillPattern – Hex value to loaded into DRAM memory

Start Offset – Optional; by default, value is 0

Size – Size in bytes (Hex); Optional; by default, FULL

7.14.3. TCL Command equivalent

bcdraminit <FillPattern> [<Start Offset> [<Size>]]

7.14.4. Snap shot

```
% bcdraminit 12345679 0 100
0 { }
%
```



7.15. mtest

7.15.1. Description

To test full DRAM memory.

Note:

- This command should be run after DRAM initialization (bcdraminit) command.

7.15.2. Usage

mtest <type> <pattern>

Where,

type: 1 - Short Memory Test
2 - Long Memory Test

pattern: Valid only if short memory test selected

7.15.3. TCL Command equivalent

bcmtest <type> <pattern>

Snap shot

```
% bcmtest 1 0x5a5a5a5a
0 {}
% bcmtest 2
0 {}
```



7.16. fwinit

7.16.1. Description

To download firmware image to the device.

Note:

- This command should be run after DRAM initialization (bcdraminit) command.

7.16.2. Usage

fwinit [<StreamFile> <VdecOutFile> <VdecInFile>]

where,

All files are optional.

By default,

StreamFile - stream.hex
VdecOutFile - vdec_outer.hex
VdecInFile - vdec_inner.hex

7.16.3. TCL Command equivalent

bcfwinit [<StreamFile> <VdecOutFile> <VdecInFile>]

7.16.4. Snap shot

```
% bcfwinit stream.hex vdec_outer.hex vdec_inner.hex
0 {}
%
```



7.17. devinit

7.17.1. Description

To initialize the hardware device

7.17.2. Usage

devinit [<SkipFWDownload> [<EnableFWInit>]]

where,

SkipFWDownload : 0 – Don't skip (default), 1 – Skip

EnableFWInit : 0 – Disable (default), 1 – Initialize

7.17.3. TCL Command equivalent

bcdevinit [<SkipFWDownload> [<EnableFWInit>]]

7.17.4. Snap shot

```
% bcdevinit 0 1
0 { }
%
```

7.18. tdma

7.18.1. Description

To test TX-DMA path (Host memory to Decoder DRAM).

Note:

- This command should be run after DRAM initialization (bcdraminit) command.

7.18.2. Usage

tdma <size> <pattern>

where,

size – Size in bytes (Hex)

pattern – 32 bit value

7.18.3. TCL Command equivalent

bctdma <size> <pattern>

7.18.4. Snap shot

```
% bctdma 100 87654321
0 { }
%
```



7.19. fwcmd

7.19.1. Description

To issue firmware command to the device.

Note:

- This command should be run after firmware initialization (bcfwinit) command.

7.19.2. Usage

fwcmd <cmd> [<args>]

where,

cmd – Sub-command (command to the firmware)

1 – Selftest

2 – Version

args – parameters for sub-command

For Selftest sub-command,

1 – Short Memory test

2 – Long Memory test

3 – Short Register test

4 – Long Register test

5 – Decode Loop-back test

else – Not allowed

7.19.3. TCL Command equivalent

bcfwcmd <cmd> [<args>]

7.19.4. Snap shot

```
% bcfwcmd 1 3
0 { }
% bcfwcmd 2
0 {FW Version: 0x00080004 0x00080003 0x00000000}
%
```





7.20. decplay

7.20.1. Description

Play and capture video to a file. This command read data from input file, DMA compressed data to decoder and captures decoded output data. Both file capturing and CRC computation are optional.

Note:

- This command should be run after complete device initialization (bcdevinit 0 1).

7.20.2. Usage

decplay <VideoInFile> <OutFile> <Iterations> <FrameNo> <CRCFile> <IFmt> <OFmt>

where,

VideoInFile:	Video Clip File name.
OutFile:	Filename to dump decoded Output.
Iterations:	Number of Iterations to run the video Clip (decimal).
FrameNo:	Starting frame number to be captured to the file (decimal).
CRCFile:	Output File to dump decoded picture CRC.
IFmt:	Input format (0=H.264, 1=MPEG, 2=VC1)
OFmt:	Format to Capture (0=YV12)

7.20.3. Result

**Status {FramesCaptured FramesCapturedToFile FramesDropped height width FrameRate
Progressive/Interlaced}**

FramesCaptured	Total number of frames captured.
FramesCapturedToFile	Number of frames dumped to file.
FramesDropped	Number of frames dropped due to errors.
Frame height	Pixel height
Frame width	Pixel width
FrameRate	Frame Rate (30, 60 ... or unknown)
Progressive/Interlaced	1 (progressive) 0 (interlaced)

7.20.4. Snap shot

```
% bcdecplay bigship.264 out.yuv 1 30 crc.txt 0 0
0 {0x000000AB 0x00000000 0x00000000 0x000002D0 0x00000500 Unknown 0x00000001}
%
```

NOTE:

- If the files ([bigship.264](#), [out.yuv](#)) are not in the same directory, then directory separator '\' should be put twice.
Ex: `bcdecplay c:\\streamfiles\\bigship.264 c:\\outfiles\\out.yuv 1 720 30`
- Only YV12 (0) capture format is supported in this tool version.
- FrameNo parameter gives the starting point for frame capture and the tool will capture 30 frames starting from frame number "FrameNo".
- When output filenames (OutFile and CRCFile) are entered as "0", file dumping will not be done. This can be used when logging of data is not needed.



7.21. decplays

7.21.1. Description

Play and capture video through AES Enabled path. This command read data from input file, DMA compressed data to decoder and captures decoded output data. Both file capturing and CRC computation are optional.

Note:

- This command should be run after complete device initialization (bcdevinit 0 1).

7.21.2. Usage

decplays <VideoInFile> <OutFile> <Iterations> <FrameNo> <CRCFile> <IFmt> <OFmt>

where,

VideoInFile:	Video Clip File name.
OutFile:	Filename to dump decoded Output.
Iterations:	Number of Iterations to run the video Clip (decimal).
FrameNo:	Starting frame number to be captured to the file (decimal).
CRCFile:	Output File to dump decoded picture CRC.
IFmt:	Input format (0=H.264, 1=MPEG, 2=VC1)
OFmt:	Format to Capture (0=YV12)

7.21.3. Result

**Status {FramesCaptured FramesCapturedToFile FramesDropped height width FrameRate
Progressive/Interlaced}**

FramesCaptured	Total number of frames captured.
FramesCapturedToFile	Number of frames dumped to file.
FramesDropped	Number of frames dropped due to errors.
Frame height	Pixel height
Frame width	Pixel width
FrameRate	Frame Rate (30, 60 ... or unknown)
Progressive/Interlaced	1 (progressive) 0 (interlaced)

7.21.4. Snap shot

```
% bcdecplays bigship.264 out.yuv 1 30 crc.txt 0 0
0 {0x000000AB 0x00000000 0x00000000 0x000002D0 0x00000500 Unknown 0x00000001}
%
```

NOTE:

- If the files ([bigship.264](#), [out.yuv](#)) are not in the same directory, then directory separator '\' should be put twice.
Ex: [bcdecplay c:\\streamfiles\\bigship.264 c:\\outfiles\\out.yuv 1 720 30](#)
- Only YV12 (0) capture format is supported in this tool version.
- FrameNo parameter gives the starting point for frame capture and the tool will capture 30 frames starting from frame number "FrameNo".
- When output filenames (OutFile and CRCFile) are entered as "0", file dumping will not be done. This can be used when logging of data is not needed.



7.22. verbose

7.22.1. Description

To turn-on or turn-off the debug information of the command. This feature would help in debugging.

Note:

- This command is not listed in the help command display

7.22.2. Usage

verbose <option>

where,

Option : “tinfo” – verbose turn-on, “off” – verbose turn-off

7.22.3. TCL Command equivalent

bcverbose <option>

7.22.4. Snap shot

```
% bcverbose tinfo
0 { }
% bcverbose off
0 { }
%
```



7.23. dmtest

7.23.1. Description

To test the DDR memory through scratch pad DMA of 7411.

Note:

- This command should be run after DRAM initialization (bcdraminit) command.

7.23.2. Usage

dmtest <sub-cmd> <start address> <end address> <pattern filename>

where,

sub-cmd : “read” – Read DMA burst test, “write” – Write DMA burst test
start address : DDR Test Start Address
end address : DDR Test End Address
pattern filename : Filename with pattern (ASCII Format)

7.23.3. TCL Command equivalent

bcdmtest <sub-cmd> <start address> <end address> <pattern filename>

7.23.4. Snap shot

```
% bcdmtest read 0 3FFFFFF pattern.txt
0 { }
% bcdmtest write 0 3FFFFFF pattern.txt
0 { }
%
```

NOTE:

- Size of memory test is aligned to 512-byte boundary
- Pattern file is a text mode file with 32-bit hexadecimal values separated by a space





7.24. rdeprom

7.24.1. Description

To read the EPROM memory contents.

7.24.2. Usage

rdeprom <offset> <length>

where,

offset : EPROM address offset

length : Number of locations to read (Hex bytes; Max 512 bytes)

7.24.3. TCL Command equivalent

bcrdeprom <offset> <length>

7.24.4. Snap shot

```
% bcrdeprom 0 100
0 {0x00 0x00 0x1F 0x38 0x62 0xA0 0x00 0x00 0x00 0x00 0x1F 0x38 0x10 0x00 0x00 0x00
0x00 0x00 0x1F 0x38 0x16 0x00 0x00 0x00 0x00 0x00 0x1F 0x38 0x97 0x8C 0x00 0x00 0
x00 0x00 0x1F 0x38 0xE2 0x8C 0x00 0x00 0x00 0x00 0x1F 0x38 0x2D 0x8D 0x00 0x00 0x0
0 0x00 0x1F 0x38 0x78 0x8D 0x00 0x00 0x00 0x00 0x1F 0x38 0xC3 0x8D 0x00 0x00 0x01
0x80 0xBF 0x1F 0xFF 0xFF 0xFF 0x7F 0xFF 0xFF 0xFF 0x7F 0xFF 0xFF 0xFF 0x7F 0x80 0x
FD 0xFF 0x27 0xFF 0xFF 0xFF 0x7F 0x01 0x80 0xBF 0x1F 0xFF 0xFF 0xFF 0x7F 0xFF 0xFF
0xFF 0x7F 0xFF 0xFF 0xFF 0x7F 0x80 0xFD 0xFF 0x27 0xFF 0xFF 0xFF 0x7F 0x20 0x00 0
x00 0x00 0x40 0x00 0x80 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x0
0 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x80
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x
00 0x00 0x00 0x00 0x10 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x02 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0
x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x20 0x00 0x20 0x00 0x0
0 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x20 0x00 0x00 0x00 0x
00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00}
%
```

NOTE:

- This command is available only in '47x' FPGA versions
- This command is not supported for future FPGA versions (48 onwards). This is due to the implementation of scrambler logic in the FPGA



7.25. wreprom

7.25.1. Description

To write to EPROM memory locations.

7.25.2. Usage

wreprom <offset> <value> [<length>]

where,

offset : EPROM offset

value : Value to be written

length : Number of bytes to be written with '*value*'. (optional: default = 4).

Length should be multiple of 4 bytes.

7.25.3. TCL Command equivalent

bcwreprom <offset> <value> [<length>]

7.25.4. Snap shot

```
% bcwreprom 0 aa 10
0 { }
% bcrdeprom 0 10
0 {0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA}
%
```



7.26. eptest

7.26.1. Description

To test the EEPROM memory.

7.26.2. Usage

eptest <pattern> <sequence>

where,

pattern : Pattern to be used for testing

sequence :

- 1 – Incremental pattern starting from '*pattern*' value
- 2 – Same '*pattern*' is used to test the complete memory.
- 3 – Toggle pattern using the '*pattern*' value.

7.26.3. TCL Command equivalent

bc eptest <pattern> <sequence>

7.26.4. Snap shot

```
% bceptest 0 1
0 { }
% bcrdeprom 0 10
0 {0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F}
% bceptest aa 2
0 { }
% bcrdeprom 0 10
0 {0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA}
% bceptest 0 3
0 { }
% bcrdeprom 0 10
0 {0x00 0xFF 0x00 0xFF 0x00 0xFF 0x00 0xFF 0x00 0xFF 0x00 0xFF 0x00 0xFF 0x00 0xFF}
%
```

NOTE:

- This command work only in '0x2Fx' FPGA versions
- This command is not supported for future FPGA versions (0x30x onwards). This is due to the implementation of scrambler logic in the FPGA



7.27. epinit

7.27.1. Description

To initialize the EPROM memory with data from a binary file.

7.27.2. Usage

epinit <offset> <filename>

where,

offset : EPROM offset to start initializing

filename : Binary file. Length will be aligned to 4 bytes and written to EPROM.

7.27.3. TCL Command equivalent

bcepinit <offset> <filename>

7.27.4. Snap shot

```
% bcepinit 0 env.bin
0 { }
%
```

7.28. epcopy

7.28.1. Description

To test the copy engine of the FPGA.

Copy engine will copy EPROM data from the specified offset and length to DRAM. This is used for loading of bootstrap code to boot the hardware.

NOTE: This feature can be tested on 0x2FA and upward version of FPGA.

7.28.2. Usage

epcopy <offset> <len>

where,

offset : EPROM offset to start copy

len : Number of bytes (in hex). Length should be multiple of 4 bytes.

7.28.3. TCL Command equivalent

bcepcopy <offset> <len>

7.28.4. Snap shot

```
% bcepcopy 0 0x100
0 { }
%
```



7.29. epburn

7.29.1. Description (Not Supported in this version moved to burnsig.exe)

To copy the bootloader code to EPROM.

NOTE: This feature can be tested on 0x30B and upward version of FPGA.

7.29.2. Usage

epburn <file>

where,

file : Bootloader binary file (Length is aligned to 4-byte boundary and copied to EPROM).

7.29.3. TCL Command equivalent

bcepburn <file>

7.29.4. Snap shot

```
% bcepburn boot.bin
0 { }
%
```

7.30. loadbl

7.30.1. Description

To load the bootloader code from the EPROM to DRAM using copy engine logic of FPGA.

This command is also used to load the security key from EPROM to DRAM at the pre-defined location along with loading of bootloader.

NOTE: This feature can be tested on 0x30B and upward version of FPGA.

7.30.2. Usage

loadbl <epromOffset> <len>

where,

offset : EPROM address offset

len: Number of locations

7.30.3. TCL Command equivalent

bcloadbl <epromOffset> <len>

7.30.4. Snap shot

```
% bcloadbl 40 0x200
0 { }
% bcloadbl 30 0x220
0 { }
%
```



7.31. decrst

7.31.1. Description

This command is used to reset the decoder chip.

NOTE: This feature can be tested on 0x30B and upward version of FPGA.

7.31.2. Usage

decrst

7.31.3. TCL Command equivalent

bcdecrst

7.31.4. Snap shot

```
% bcdecrst
0 { }
%
```



8. NOTE

1. All deconf commands should be prefixed with “bc” string to run on TCL shell.
2. “devopen” command should be run before running any operation on the hardware device.
3. Run “devclose” command before closing the TCL shell to close the device.
4. “bcverbose” command is not listed in the command list (Hidden command).





9. PCI Device Access Tool

9.1. Description (Not Supported in this version)

PCI Device Access Tool (bcpci.exe) is used to access the configuration space of the any PCI Device in the System.

9.2. Command Description

9.2.1. PCI Device Scan

9.2.1.1. Description

This command is used to scan all the PCI devices in the System.

9.2.1.2. Command Usage

```
bcpci scan
```

9.2.1.3. Return Values

This command list all the devices in the System.

9.2.1.4. Snap shot

```
P:\> plxpci fd

b : d : f   VenID:DevID

0 : 0 : 0   3340:8086
0 : 1 : 0   3341:8086
0 : 29 : 0   24C2:8086
0 : 29 : 1   24C4:8086
0 : 29 : 2   24C7:8086
0 : 29 : 7   24CD:8086
0 : 30 : 0   2448:8086
0 : 31 : 0   24CC:8086
0 : 31 : 1   24CA:8086
0 : 31 : 5   24C5:8086
0 : 31 : 6   24C6:8086
1 : 0 : 0   4C66:1002
1 : 0 : 1   4C66:1002

P:\>
```



9.2.2. PCI Configuration Space Read

9.2.2.1. Description

This command is used to read the value at a given offset of the PCI Device (Identified by bus, device and function number) configuration space. 32-bit value is read from the given offset.

9.2.2.2. Command Usage

`bcpci read <busno> <devno> <funno> <offset>`

9.2.2.3. Return Values

This command returns the value if the command usage is correct otherwise command returns an error string indicating the mistake in the usage.

9.2.2.4. Snap shot

```
P:\> bcpci read 2 1 1 0
0x71131217
P:\> bcpci read
BC_ERROR: Less no. of arguments
P:\>
```

9.2.3. PCI Configuration Space Write

9.2.3.1. Description

This command is used to write the given value at a given offset of the PCI Device (Identified by bus, device and function number) configuration space. 32-bit value is written at the given offset.

9.2.3.2. Command Usage

`bcpci write <busno> <devno> <funno> <offset> <value>`

9.2.3.3. Return Values

This command simply return if the command usage is correct otherwise command returns an error string indicating the mistake in the usage.

9.2.3.4. Snap shot

```
P:\> bcpci write 1 1 1 3c 0

P:\> bcpci write
BC_ERROR: Less no. of arguments
P:\>
```



9.2.4. PCI Configuration Space Dump

9.2.4.1. Description

This command is used to show/dump a range of configuration space values starting from the given offset of the PCI Device (Identified by bus, device and function number).

9.2.4.2. Command Usage

```
bcpci show <busno> <devno> <funno> <offset> <bytes>
```

Where,

offset – Starting location in the configuration space

bytes – Number of bytes to dump/show (decimal and multiple of 4)

9.2.4.3. Return Values

This command return 'bytes' length of 32-bit values separated by a space if the command usage is correct otherwise command returns an error string indicating the mistake in the usage.

9.2.4.4. Snap shot

```
P:\> bcpci show 0 0 0 12  
0x27748086 0x20900106 0x06000000  
P:\>
```

Note:

- PortTalk Driver (PortTalk.sys) should be installed before accessing the commands.
- Or PortTalk Driver (PortTalk.sys) is in the same directory of bcpci tool
- All bcpci commands are 32-bit accesses to configuration space.





10. Appendix – A ('deconf' modes)

This section describes the list of modes supported from 'deconf' console application.

10.1. Diagnostic mode

This mode is used for debugging the hardware. To run the deconf application in this mode, enter the following command at the console prompt.

```
C:\> deconf -Diag
```

In this mode, device name (DOZER, LINK etc) will appear as the prompt name.

```
DOZER> help
```

Note:

- 'deconf' (TCL Version) will always open in diagnostics mode.

10.2. Playback mode

This mode is used to run in playback mode (to simulate the application usage directly). To run the deconf application in this mode, enter the following command at the console prompt.

```
C:\> deconf -Play <VideoFile> <Outfile (0=nofiledump)> <Iterations> <frm_off> <CRCFile> <I Fmt> <OFmt>
```

10.3. Monitor mode

This mode is used for logging all the statistical information when running decoder (i.e, decplay command or any other playback application). To run the deconf application in this mode, enter the following command at the prompt.

```
C:\> deconf -Monitor
```

In this mode, "DMON" will appear as the prompt name.

```
DMON> getver
```

List of monitor parameters supported in this mode,

- Driver buffer list count
- Output frame information
 - Frame rate
 - Frames captured
 - Frames dropped
 - Frames repeated
 - Frames with no PIB
- Input sample information
 - Sample count
 - Sample size captured
 - Throughput
- Pause and resume counts (Not implemented)
- Interrupt counts (Not implemented)



List of commands specific to monitor mode,

10.3.1. getstat

10.3.1.1. Description

To start the statistic capture.

10.3.1.2. Usage

getstat <frequency>

where,

frequency : Period of stat display (default : 2sec)

10.3.1.3. Snap shot

```
DMON>getstat
Stat Dump
-----

Ready List Length      : 0
Free List Length       : 0

Output Frame Rate      : 0
  Frame Captured Count : 0
  Frame Drop Count     : 0
  Repeated Frame Count : 0
  Missed PIB           : 0

Input (count) (size)   : 0 0 KB
  Throughput (MB/s)    : 0.00

Pause Count            : 0
Interrupt              : 0

Press 's' or 'S' to stop:
```

NOTE:

- Pause Count and Interrupt are not implemented





10.3.2. rststat

10.3.2.1. Description

To reset all the statistic capture counts to zero.

10.3.2.2. Usage

rststat

10.3.2.3. Snap shot

```
DMON>rststat
DMON>
```

NOTE:

1. The above modes can be run at the same time so that user can capture the statistics of the above listed parameters in real time.
2. Restrictions
 - a. Only one instance of each mode can be run simultaneously.
 - b. Either Diagnostics or playback mode is allowed to run simultaneously.
3. Playback mode is invoked when running the decoder with DirectShow or using deconf –Play mode

