Release Notice
CONVEX C200/C3200 Series Processor Diagnostics and Database V5.0

Document No. 760-001130-007

February 1992

CONVEX Computer Corporation
Richardson, Texas, U.S.A.

Table of Contents

Appendices

1

Release Notice

1. Introduction

This document is intended to enhance and clarify the existing permanent documentation for this product with information that is up-to-the-minute, or was developed too late for inclusion in the permanent documentation. Always refer to this release notice before reporting questions or problems with CONVEX C200/C3200 Series Processor Diagnostics and Database. Your questions may be answered here. Fixes and workarounds are listed here that may save you time in rediscovering known problems.

The remaining sections in this document describe the contents of this release:

- Section 2 describes the contents of this distribution.
- Section 3 contains notes and warnings about the use of the software.
- Section 4 contains enhancements to the previous functionality.
- Section 5 describes fixes for previously reported problems.
- Section 6 describes known software problems.
- Section 7 contains known hardware problems.
- Section 8 contains known documentation problems.
- Section 9 contains description of new documentation.
- Appendix A contains instructions for installing this release on a CONVEX C200/C3200 Series Service Processor Unit.
- Appendix B contains a list of the files contained on the release tape.

Processor Diagnostics consists of various functional-level test programs and diagnostic utilities that execute under the CONVEX SPU OS V6.0 operating system of the Service Processor Unit. All programs MUST execute in the offline diagnostic environment of CONVEX SPU OS and are mutually exclusive with the operation of ConvexOS on the main processors. These programs are the property of CONVEX Computer Corporation and are intended for use only by CONVEX Field Service.

2. Contents of This Distribution

The distribution package for this release of Processor Diagnostics consists of this document, distribution media for the software, and documentation. The specific contents of the software and documentation distribution are described in the following tables:

CONVEX C200/C3200 Series Processor Diagnostics Media

ITEM	QTY	TYPE	PART NUMBER	DESCRIPTION	FORMAT
1.	1	QIC.	760-001015-220	Processor Diagnostics and Database, V5.0	Installsw

If you do not already have the Processor Diagnostics Documentation, you will receive the Release Package.

CONVEX C200 Series Processor Diagnostics Documentation

CONVEX C200 Series Release Package

ITEM	QTY	TYPE	PART NUMBER	DESCRIPTION
1.	1	Manl.	760-000430-000	CONVEX C200 Diagnostics Master Index, 4th ed.
2.	1	Manl.	760-000550-203	CONVEX C200 Processor Diagnostics Manual, 3rd ed.
3.	1	Manl.	760-000599-200	CONVEX C200 Processor Diagnostics Manual Addendum, 3rd ed. rev 1
4.	1	Manl.	760-000730-000	PBUS I/O System Diagnostics Manual, 3rd ed.
5.	1	Manl.	760-000830-000	CONVEX C200 Diagnostics Utilities Manual, 3rd ed.
6.	1	Manl.	760-000830-001	CONVEX C200 Diagnostics Utilities Manual Addendum, 3rd ed. rev 1

CONVEX C200/C3200 Series Update Package

- Should include item 6 in release package.

3. Notes and Warnings

This section contains generally useful information or words of caution about the product.

- This release supercedes all previous versions of CONVEX C200/C3200 Series System Diagnostics.
- This release of CONVEX C200/C3200 Series Processor Diagnostics and Database requires the V1.0 release of the CONVEX C200/C3200 Series IO Diagnostics (760-004215-000) and the V5.1 (or later) release of CONVEX SPU OS (760-001215-203).
- The following table lists the hardware configurators required to support the release of System Diagnostics:

CONVEX C200/C3200 Series Logic Configurator Specification

Logic Configurator	PART NUMBER
CX Configurator	400-000100-979
C32XX Shipping Configurator	400-000100-981

Board revision requirements

Board	Part Number	Rev.	Expected subtest failures
IPP	410-001207-200	J or older	cpu4231 -s 31,42,44 cpu4233 -s 1010
		K or newer	none related to the IPP
DCU	410-001208-200	K or older	cpu4231 -s 323-324 cpu4241 -s 2505-2606 cpu4241 -s 4505-4606
		L or newer	none related to the DCU
EDC	410-001219-200	D or older	cpu4231 -s 323-324,400 cpu4241 -s 2505-2606 cpu4241 -s 4505-4606
		E or newer	none related to the EDC

- This release should only be installed by a qualified CONVEX Field Service representative. Please see Appendix A for installation details.

4. Enhancements

Utilities

The following enhancements have been made to utility programs in this release:

- The following utilities were modified as part of splitting the IO portion of the utilities from the CPU portion. This was done to facilitate the maintenance of the various utilities and files that are shared resources between the two groups.
  - hard_logger - functionality for logging IO hard errors is handled by io_hard_logger which is called from hard_logger.
  - margin - functionality for margining the ccu clocks is handled by io_margin which is called by margin.
  - scn_util - addition of ccu tokens to the DB_cop file are interpreted. The DB_cop file is broken into two pieces DB_cop.proc and DB_cop.io which are combined by the install process.
  - sysreset - initialization of all IO devices is handled bu io_sysreset which is called from sysreset.

Test Programs

The following enhancements have been made to test programs in this release:

- No enhancements have been made to the test programs.

Hardware/Microcode

The following enhancements have been made to the hardware/microcode in this release:

- No enhancements have been made to the hardware/microcode.

5. Fixes

Utilities

- No utility upgrades are included in this release.

Test Programs

- cpu4232 - the following subtests were modified to test the new microcode: st_540, st_550, st_640, st_650, st_1540, st_1550, st_1640, st_1650.
- mem4100 - Fixed the problem of being unable to loop on the test under certain conditions.
- spu4000 - TLI and HiPPI are now supported.
- misc4000 - modified to support the pia.

Hardware/Microcode

The following fixes have been made to the hardware/microcode in this release:

- The us.200.wcs and the sr.wcs files have been updated. These new revisions implement a more precise algorithm for the calculation of sine and cosine.
- This release contains the following CONVEX C200/C3200 Series microcode file revisions:

us.200.wcs	11.6
sr.wcs	11.3
vd.200.wcs	10.1
ua.wcs	10.10
um.wcs	10.6
ul.wcs	10.3

6. Known Software Problems

At the time this release notice was prepared, this section contains the known problems with this release of Processor Diagnostics software. Please refer to this list prior to reporting a problem in order to ensure that it has not been previously reported. Serious problems include workarounds if they are known.

Utilities

- None

Test Programs

- The diagnostic shell (dshell) incorrectly prints the description of some tests when it is invoked by entering

  dshell test

  This problem does not affect the functionality of dshell or any of the tests, regardless of whether the aforementioned description is correct or incorrect.

- The test **cpu4233** will very infrequently (about 1 time in 25 runs) fail subtest 521 when run on a C240 with 4 cpus under test. The subtest will fail with a cpu timeout. This is a result of a bug in the diagnostic software, and not an indication that the subtest actually failed. A work-around is to run the test again.

- The test **cpu4231** will not run on a C230 or C240 with less than 128MB of main memory. The following message will be displayed:

  ```
  LCPU_014:get_start_address: Not enough physical memory
  ```

  A work-around is to run the test on one or two heads at a time.

- When running the dshell from a directory other than /mnt/test you will see the message:

  ```
  xxx.t Unable to determine SW revision
  ```

  This is not an error but indicates that the version program was unable to find the file xxx.t (for example cpu4030.t or pi2_4000.t) in the current directory.

9. Known Hardware Problems

Version of diags	Board	Part Number	Expected Rev.	subtest failures
any	any	n/a		cpu4231 -s 42,44,700,701 in ring 0 cpu4241 -s 1005 in ring 4
any	IPP	410-001207-200	J or older K or newer	cpu4231 -s 31,42,44 cpu4233 -s 1010 none related to the IPP
V3.4 or newer	DCU	410-001208-200	K or older L or newer	cpu4231 -s 323-324 cpu4241 -s 2505-2606 cpu4241 -s 4505-4606 none related to the DCU
V3.4 or newer	EDC	410-001219-200	D or older E or newer	cpu4231 -s 323-324,400 cpu4241 -s 2505-2606 cpu4241 -s 4505-4606 none related to the EDC

8. Known Documentation Problems

There are no known documentation problems for this release.

9. New Documentation

Included below is a description of how hard and soft errors are handled and logged by the CONVEX SPU software on a CONVEX C3200 Series machine.

9.0 Error Logging

This chapter describes how errors are processed for CONVEX C3200 and CONVEX C3400 class systems. Error handling is almost identical between the system types, with the only difference being with respect to the soft errors which can occur on a CONVEX C3400 CPU. There are two utilities which perform the task of logging errors on these systems: hard_logger and errintd.

The hard_logger utility is used by diagnostics to report the hard and soft errors that are monitored, and it is invoked by errintd to report hard errors and fatal soft errors that occur during ConvexOS. The hard_logger stops clocks and (by default) scans each board in the system to determine what errors are present in the system. It displaying it's information to stdout, and also maintains a last invocation file and a history file. When invoked by errintd, output is usually time stamped and duplicated in /mnt/errlog by prtlog.

The errintd utility runs on the CONVEX SPU when ConvexOS is running and monitors the system for a variety of conditions, including hard, soft, and environmental errors. All hard and some soft errors result in an OS crash, whereas memory soft errors, certain special soft errors, and environmental errors are tolerable and allow the OS to continue running. Anytime errintd reports a non-fatal soft error, that error is then re-enabled. Also, for any soft error that errintd reports, there is a corresponding error message in hard_logger.

9.1 Error Processing During ConvexOS

Whenever ConvexOS is running on a system, the diagnostic utility errintd runs in background on the CONVEX SPU, monitoring for specific error conditions. The errintd utility sets up an interrupt handler for hard, soft, and environmental errors; then enables these particular interrupts. If one of these interrupts occurs, errintd processes it accordingly.

All environmental errors are warning messages, as the System Monitor Board will remove power without warning if a serious environmental condition is detected. By their nature, hard errors are fatal errors, but certain soft errors are also considered fatal and result in ungraceful system shutdown.

When an environmental error occurs for the first time, a message is displayed and the event remembered, as the error signal remains as long as the error condition is present. At regular intervals, errintd checks to see if the environmental error interrupt remains. If it is, a "still present" message is displayed, with any changes noted from the previous error. If no error remains, an "all ok" message is displayed.

When a hard error occurs, errintd does not process the error itself, but calls the hard_logger utility to scan the system and determine the cause. Normally the hard_logger output that goes to stdout will also be captured by prtlog and copied into the file /mnt/errlog. However, the hard_logger utility also generates it's own last invocation file /mnt/hardlog and history file /mnt/hardlog.history, as prtlog may have died due to the hard error.

There are three basic kinds of soft errors that can occur on a CONVEX C3200 or CONVEX C3400 system: single bit memory errors, CONVEX C3400 CPU board (jcpu) soft errors, and fatal soft errors. When a soft error occurs, errintd determines the type, and acts accordingly.

Memory system soft errors are logged in the file /mnt/softlog, with each entry denoting the failing part, as possible. If they begin to occur very frequently, memory soft errors are throttled, as it is possible for a hardware failure to cause an excessive number of failures (and thus messages), which could fill up the log if they were allowed to continue. The errintd utility maintains a memory resident database of soft memory errors, which is only updated periodically (every 5 minutes), and only if a soft memory error has occurred since the last update. After logging, the soft error is re-enabled.

Fatal soft errors are Peripheral Interface Adaptor and CONVEX C3200 series CPU Utility board soft errors. These are treated identically to hard errors, and result in ungraceful system shutdown.

9.2 Error Processing During Diagnostics

Error monitoring while executing a diagnostic test or utility is dependent on the specific test or utility. Nevertheless, there is a great amount of consistency. During diagnostics, all unexpected errors are considered fatal errors. All tests (cpu4XXX.t, mem4100.t, spu4000.t, ...) invoke hard_logger whenever a hard or soft error occurs. Likewise, although most low level utilities by their nature ignore hard and soft errors (or more commonly have them disabled), ones that expect the system to be in a rational state and behave normally do monitor for hard and soft errors and call hard_logger if they occur. A good example of a utility that does this is mminit. Environmental error monitoring is not done for diagnostic tests or utilities.

A

Installing CONVEX C200/C3200 Series Processor Diagnostics and Database V5.0

Warnings

This section contains generally useful information or words of caution about the product.

- This release supercedes all previous versions of CONVEX C200/C3200 Series System Diagnostics.
- This release of CONVEX C200/C3200 Series Processor Diagnostics and Database requires the V1.0 release of the CONVEX C200/C3200 Series IO Diagnostics (760-004215-000) and the V5.1 (or later) release of CONVEX SPU OS (760-001215-203).
- The following table lists the hardware configurators required to support the release of System Diagnostics:

CONVEX C200/C3200 Series Logic Configurator Specification

Logic Configurator	PART NUMBER
CX Configurator	400-000100-979
C32XX Shipping Configurator	400-000100-981

Board revision requirements

Board	Part Number	Rev.	Expected subtest failures
IPP	410-001207-200	J or older	cpu4231 -s 31,42,44 cpu4233 -s 1010
		K or newer	none related to the IPP
DCU	410-001208-200	K or older	cpu4231 -s 323-324 cpu4241 -s 2505-2606 cpu4241 -s 4505-4606
		L or newer	none related to the DCU
EDC	410-001219-200	D or older	cpu4231 -s 323-324,400 cpu4241 -s 2505-2606 cpu4241 -s 4505-4606
		E or newer	none related to the EDC

- This release should only be installed by a qualified CONVEX Field Service representative.

Installation Procedure

1. If CONVEX SPU OS is already booted, go to step 5.

2. Place the front panel key switch in the *local* position and depress the system reset button to boot CONVEX SPU OS.

3. The soft front panel menu will be displayed. Change the mode to diagnostics and continue the boot process by entering the following commands at the **(fp)>** prompt:

   (fp)> set mode=diagnostic (or sm=d)
   (fp)> boot (or b)

4. The CONVEX SPU OS bootstrap routine will prompt with:

   CONVEX SPU OS boot
   :

   You should enter a carriage return <CR> in response to the prompt. CONVEX SPU OS will now boot and prompt with **(spu)>** when boot is complete.

   NOTE: A file system check is performed during the boot procedure. If errors are detected in the file system, they will be corrected if possible. If it is not possible to automatically correct the errors, then you will be requested to execute */etc/fsck* manually to correct these errors before proceeding.

5. Verify CONVEX SPU OS V5.1 or later is installed :

   (spu)> more /UNIX_REV

   If the file */UNIX_REV* is not present, then install CONVEX SPU OS V5.1 or later.

6. Place the CONVEX C200/C3200/C3400 IO Diagnostics V1.0 tape (760-004215-000) in the cartridge tape unit and install the tape per the instructions in the CONVEX C200/C3200/C3400 IO Diagnostics V1.0 Release Notice (760-006730-000).

7. Place the CONVEX C200/C3200 Procesor Diagnostics and Database V5.0 tape (760-001015-220) in the cartridge tape unit and enter the following command:

   (spu)> /etc/installsw -i

   The installation of this release requires about 17 minutes.

8. A log of all tar operations is saved in */tmp/installsw.tar*. should be removed as follows:

   (spu)> rm /tmp/installsw.tar

9. After installation is complete, remove the tape from the cartridge tape unit.

10. If the desired mode of operation is diagnostic mode, then this step may be skipped. Otherwise, return to the soft front panel via the */etc/reboot* command:

    (spu)> /etc/reboot

    Change the mode of operation setting to the *desired-mode*. Use the soft front panel *help* command if you need assistance.

(fp)> set mode=*desired-mode*

Place the front panel key switch in the *secure* position and enter the *boot* command to reboot the system:

(fp)> boot (or b)

11. This completes the installation of CONVEX C200/C3200 Processor Diagnostics and Database. Note that the machine is automatically set up to the maximum configuration. All installed CPU's will be enabled and initalized to a CONVEX C200/C3200 configuration.

B
Files list

The following is the *installsw* output from making the V5.0 Processor Diagnostics and Database tape.

a /tmp/install1 1 blocks
a /tmp/install2 4 blocks
a mnt/bin/CPU/mminit.x00 42 blocks
a mnt/bin/commreg 304 blocks
a mnt/bin/config_chk 202 blocks
a mnt/bin/cop 324 blocks
a mnt/bin/cpureg 300 blocks
a mnt/bin/cpuvreg 344 blocks
a mnt/bin/cs 343 blocks
a mnt/bin/dcache 307 blocks
a mnt/bin/dshell 88 blocks
a mnt/bin/.diaginit 7 blocks
a mnt/bin/dump_winq 227 blocks
a mnt/bin/errintd 365 blocks
a mnt/bin/hard_logger 484 blocks
a mnt/bin/hwlog 205 blocks
a mnt/bin/icache 351 blocks
a mnt/bin/initall 3 blocks
a mnt/bin/iscn 539 blocks
a mnt/bin/man 45 blocks
a mnt/bin/map 301 blocks
a mnt/bin/margin 206 blocks
a mnt/bin/mcm3_config 424 blocks
a mnt/bin/memld 383 blocks
a mnt/bin/mkdiag_db 202 blocks
a mnt/bin/disable_cpu 1 blocks
a mnt/bin/enable_cpu 1 blocks
a mnt/bin/rita_perr 1 blocks
a mnt/bin/secure 1 blocks
a mnt/bin/val_perr 1 blocks
a mnt/bin/mm 505 blocks
a mnt/bin/mm_sniff 92 blocks
a mnt/bin/mmap 83 blocks
a mnt/bin/mminit 733 blocks
a mnt/bin/pte_cache 298 blocks
a mnt/bin/pup 213 blocks
a mnt/bin/RC_Talk 25 blocks
a mnt/bin/RC_Xfer 57 blocks
a mnt/bin/reset_cpus 165 blocks
a mnt/bin/scn_ring 306 blocks
a mnt/bin/scn_util 599 blocks
a mnt/bin/scnlink 113 blocks
a mnt/bin/sfpread 49 blocks
a mnt/bin/sp2util 174 blocks
a mnt/bin/syshalt 162 blocks

a mnt/bin/sysreset 591 blocks
a mnt/bin/version 43 blocks
a mnt/bin/x 28 blocks
a mnt/bin/lib/security_clear/mm_purge_ccu 2 blocks
a mnt/bin/lib/security_clear/mm_purge_spu 12 blocks
a mnt/bin/lib/security_clear/purge_cpu 33 blocks
a mnt/bin/lib/security_clear/purge_hsp 66 blocks
a mnt/bin/lib/security_clear/purge_iop 61 blocks
a mnt/bin/lib/security_clear/purge_viop 59 blocks
a mnt/bin/lib/security_clear/purge_patterns 1 blocks
a mnt/bin/security_clear 689 blocks
a mnt/test/CPU/mem4100.x00 29 blocks
a mnt/test/CPU/mem4100.x01 29 blocks
a mnt/test/CPU/cpu4010.rnn 92 blocks
a mnt/test/CPU/p0r0_4010.1 3 blocks
a mnt/test/CPU/p0r0_4010.2 4 blocks
a mnt/test/CPU/cpu4030.rnn 422 blocks
a mnt/test/CPU/p0r0_4030 3 blocks
a mnt/test/CPU/p0rN_4030 2 blocks
a mnt/test/CPU/pccarry_4030 9 blocks
a mnt/test/CPU/wrapl_4030 1 blocks
a mnt/test/CPU/wrapu_4030 9 blocks
a mnt/test/CPU/cpu4040.rnn 125 blocks
a mnt/test/CPU/p0r0_4040.1 2 blocks
a mnt/test/CPU/p0r0_4040.2 3 blocks
a mnt/test/CPU/cpu4041.rnn 1617 blocks
a mnt/test/CPU/p0r0_4041 12 blocks
a mnt/test/CPU/p0rN_4041 2 blocks
a mnt/test/CPU/cpu4041.x00 1159 blocks
a mnt/test/CPU/cpu4241.rnn 4021 blocks
a mnt/test/CPU/p0r0_4241 36 blocks
a mnt/test/CPU/p0rN_4241 2 blocks
a mnt/test/CPU/cpu4241.x00 2823 blocks
a mnt/test/CPU/cpu4241.x04 2823 blocks
a mnt/test/CPU/cpu4231.rnn 732 blocks
a mnt/test/CPU/p0r0_4231 23 blocks
a mnt/test/CPU/p0rN_4231 30 blocks
a mnt/test/CPU/st_323_spt 1 blocks
a mnt/test/CPU/support_4231 34 blocks
a mnt/test/CPU/segment7 1 blocks
a mnt/test/CPU/cpu4232.rnn 1400 blocks
a mnt/test/CPU/p0r0_4232 8 blocks
a mnt/test/CPU/p0rN_4232 2 blocks
a mnt/test/CPU/cpu4233.rnn 641 blocks
a mnt/test/CPU/p0r0_4233 45 blocks
a mnt/test/CPU/p0rN_4233 41 blocks
a mnt/test/CPU/support_4233 1 blocks
a mnt/test/tables/cpu4030_cl.o 1 blocks
a mnt/test/tables/cpu4030_st.o 20 blocks
a mnt/test/tables/cpu4041_cl.o 2 blocks
a mnt/test/tables/cpu4041_st.o 35 blocks
a mnt/test/tables/cpu4041_tm.o 8 blocks
a mnt/test/tables/cpu4047_cl.o 1 blocks
a mnt/test/tables/cpu4047_st.o 12 blocks
a mnt/test/tables/cpu4231_cl.o 2 blocks
a mnt/test/tables/cpu4231_st.o 16 blocks
a mnt/test/tables/cpu4231_tm.o 4 blocks
a mnt/test/tables/cpu4232_cl.o 2 blocks
a mnt/test/tables/cpu4232_st.o 8 blocks

a mnt/test/tables/cpu4232_tm.o 3 blocks
a mnt/test/tables/cpu4233_cl.o 3 blocks
a mnt/test/tables/cpu4233_st.o 30 blocks
a mnt/test/tables/cpu4233_tm.o 4 blocks
a mnt/test/tables/cpu4241_cl.o 2 blocks
a mnt/test/tables/cpu4241_st.o 90 blocks
a mnt/test/tables/cpu4241_tm.o 17 blocks
a mnt/test/tables/cpu4331_cl.o 2 blocks
a mnt/test/tables/cpu4331_st.o 17 blocks
a mnt/test/tables/cpu4331_tm.o 4 blocks
a mnt/test/tables/cpu4332_cl.o 2 blocks
a mnt/test/tables/cpu4332_st.o 8 blocks
a mnt/test/tables/cpu4332_tm.o 3 blocks
a mnt/test/tables/cpu4333_cl.o 3 blocks
a mnt/test/tables/cpu4333_st.o 24 blocks
a mnt/test/tables/cpu4333_tm.o 4 blocks
a mnt/test/mem4100.t 1048 blocks
a mnt/test/spu4000.t 774 blocks
a mnt/test/misc4000.t 720 blocks
a mnt/test/cpx4000.t 661 blocks
a mnt/test/cpu4XXX.t 845 blocks
a mnt/test/cpu4030.t link to mnt/test/cpu4XXX.t
a mnt/test/cpu4041.t link to mnt/test/cpu4XXX.t
a mnt/test/cpu4231.t link to mnt/test/cpu4XXX.t
a mnt/test/cpu4232.t link to mnt/test/cpu4XXX.t
a mnt/test/cpu4233.t link to mnt/test/cpu4XXX.t
a mnt/test/cpu4241.t link to mnt/test/cpu4XXX.t
a mnt/test/pia4000.t 787 blocks
a mnt/test/pi2_4000.t 469 blocks
a mnt/test/pi2_4000.x00 282 blocks
a mnt/test/cpu4010.t 806 blocks
a mnt/test/cpu4040.t 803 blocks
a mnt/man/cat1/Intro.1 3 blocks
a mnt/man/cat1/Intro.1d 4 blocks
a mnt/man/cat1/adb.1 35 blocks
a mnt/man/cat1/awk.1 15 blocks
a mnt/man/cat1/cat.1 2 blocks
a mnt/man/cat1/cd.1 2 blocks
a mnt/man/cat1/chmod.1 6 blocks
a mnt/man/cat1/cmp.1 3 blocks
a mnt/man/cat1/commreg.1d 10 blocks
a mnt/man/cat1/config_chk.1d 4 blocks
a mnt/man/cat1/cop.1d 15 blocks
a mnt/man/cat1/cp.1 2 blocks
a mnt/man/cat1/cpureg.1d 7 blocks
a mnt/man/cat1/cpuvreg.1d 7 blocks
a mnt/man/cat1/cs.1d 15 blocks
a mnt/man/cat1/date.1 8 blocks
a mnt/man/cat1/dcache.1d 7 blocks
a mnt/man/cat1/dd.1 8 blocks
a mnt/man/cat1/df.1 2 blocks
a mnt/man/cat1/diaginit.1d 6 blocks
a mnt/man/cat1/diskfmt.1d 11 blocks
a mnt/man/cat1/dshell.1d 18 blocks
a mnt/man/cat1/echo.1 2 blocks
a mnt/man/cat1/errintd.1d 10 blocks
a mnt/man/cat1/fs.1d 12 blocks
a mnt/man/cat1/grep.1 9 blocks
a mnt/man/cat1/hard_logger.1d 7 blocks

a mnt/man/cat1/icache.1d 10 blocks
a mnt/man/cat1/initall.1d 4 blocks
a mnt/man/cat1/ipcs_server.1d 2 blocks
a mnt/man/cat1/iscn.1d 2 blocks
a mnt/man/cat1/kermit.1 37 blocks
a mnt/man/cat1/kill.1 3 blocks
a mnt/man/cat1/lf.1 0 blocks
a mnt/man/cat1/ll.1 0 blocks
a mnt/man/cat1/ln.1 3 blocks
a mnt/man/cat1/ls.1 14 blocks
a mnt/man/cat1/man.1d 5 blocks
a mnt/man/cat1/map.1d 4 blocks
a mnt/man/cat1/margin.1d 11 blocks
a mnt/man/cat1/memld.1d 5 blocks
a mnt/man/cat1/mkdiag_db.1d 9 blocks
a mnt/man/cat1/mkdir.1 2 blocks
a mnt/man/cat1/mm.1d 25 blocks
a mnt/man/cat1/mm_sniff.1d 4 blocks
a mnt/man/cat1/mminit.1d 22 blocks
a mnt/man/cat1/more.1 3 blocks
a mnt/man/cat1/mt.1 6 blocks
a mnt/man/cat1/mv.1 3 blocks
a mnt/man/cat1/clear.1 2 blocks
a mnt/man/cat1/od.1 8 blocks
a mnt/man/cat1/proctype.1 2 blocks
a mnt/man/cat1/prof.1 4 blocks
a mnt/man/cat1/ps.1 7 blocks
a mnt/man/cat1/pte_cache.1d 7 blocks
a mnt/man/cat1/pup.1d 3 blocks
a mnt/man/cat1/pwd.1 1 blocks
a mnt/man/cat1/reset.1 3 blocks
a mnt/man/cat1/rm.1 4 blocks
a mnt/man/cat1/scn_ring.1d 4 blocks
a mnt/man/cat1/scn_util.1d 12 blocks
a mnt/man/cat1/scnlink.1d 7 blocks
a mnt/man/cat1/sfpread.1d 3 blocks
a mnt/man/cat1/sh.1 48 blocks
a mnt/man/cat1/sleep.1 2 blocks
a mnt/man/cat1/sort.1 10 blocks
a mnt/man/cat1/sp2util.1d 19 blocks
a mnt/man/cat1/stty.1 6 blocks
a mnt/man/cat1/syshalt.1d 4 blocks
a mnt/man/cat1/sysreset.1d 7 blocks
a mnt/man/cat1/tail.1 3 blocks
a mnt/man/cat1/tar.1 11 blocks
a mnt/man/cat1/tee.1 2 blocks
a mnt/man/cat1/test.1 5 blocks
a mnt/man/cat1/time.1 2 blocks
a mnt/man/cat1/true.1 2 blocks
a mnt/man/cat1/uptime.1 2 blocks
a mnt/man/cat1/version.1d 10 blocks
a mnt/man/cat1/vi.1 7 blocks
a mnt/man/cat1/vp_scn.1d 12 blocks
a mnt/man/cat1/which.1 2 blocks
a mnt/man/cat1/x.1d 3 blocks
a mnt/man/cat1/xed.1 37 blocks
a mnt/man/cat1/secure.1d 4 blocks
a mnt/man/cat1/rita_perr.1d 6 blocks
a mnt/man/cat1/val_perr.1d 5 blocks

a mnt/man/cat1/mcm3_config.1d 4 blocks
a mnt/man/cat1/security_clear.1d 14 blocks
a mnt/man/cat1/csh.1 145 blocks
a mnt/man/cat1/less.1 62 blocks
a mnt/man/cat4/Intro.4 3 blocks
a mnt/man/cat4/ct.4 11 blocks
a mnt/man/cat4/dk.4 11 blocks
a mnt/man/cat4/mem.4 2 blocks
a mnt/man/cat4/mtio.4 10 blocks
a mnt/man/cat4/null.4 1 blocks
a mnt/man/cat4/tty.4 46 blocks
a mnt/man/cat4/wndw.4 13 blocks
a mnt/man/cat5/DB_cop.5d 3 blocks
a mnt/man/cat5/DB_diskfmt.5d 8 blocks
a mnt/man/cat5/b.out.5 16 blocks
a mnt/man/cat5/backup.5 14 blocks
a mnt/man/cat5/controllers.5d 2 blocks
a mnt/man/cat5/core.5 3 blocks
a mnt/man/cat5/dir.5 3 blocks
a mnt/man/cat5/environ.5 3 blocks
a mnt/man/cat5/filsys.5 19 blocks
a mnt/man/cat5/fstab.5 8 blocks
a mnt/man/cat5/group.5 2 blocks
a mnt/man/cat5/ioconfig.5d 20 blocks
a mnt/man/cat5/mtab.5 2 blocks
a mnt/man/cat5/passwd.5 3 blocks
a mnt/man/cat5/softlog.5d 9 blocks
a mnt/man/cat5/tar.5 8 blocks
a mnt/man/cat5/termcap.5 56 blocks
a mnt/man/cat5/ttys.5 3 blocks
a mnt/man/cat5/ttytype.5 2 blocks
a mnt/man/cat5/types.5 4 blocks
a mnt/man/cat7/Intro.7 1 blocks
a mnt/man/cat7/ascii.7 4 blocks
a mnt/man/cat7/hier.7 7 blocks
a mnt/man/cat8/Intro.8 1 blocks
a mnt/man/cat8/backup.8 10 blocks
a mnt/man/cat8/bootchk.8 2 blocks
a mnt/man/cat8/cleanup.8 2 blocks
a mnt/man/cat8/fasthalt.8 2 blocks
a mnt/man/cat8/format.8 14 blocks
a mnt/man/cat8/fsck.8 16 blocks
a mnt/man/cat8/getty.8 4 blocks
a mnt/man/cat8/init.8 5 blocks
a mnt/man/cat8/installsw.8 28 blocks
a mnt/man/cat8/mkfs.8 7 blocks
a mnt/man/cat8/mklost+found.8 2 blocks
a mnt/man/cat8/mknod.8 2 blocks
a mnt/man/cat8/mount.8 4 blocks
a mnt/man/cat8/pstat.8 11 blocks
a mnt/man/cat8/pwrdwn.8 2 blocks
a mnt/man/cat8/reboot.8 12 blocks
a mnt/man/cat8/sync.8 1 blocks
a mnt/man/cat8/update.8 2 blocks
a mnt/man/man.index 6 blocks
a mnt/man/whatis 18 blocks
a mnt/usr/ucode/sr.wcs 25 blocks
a mnt/usr/ucode/ua.wcs 16 blocks
a mnt/usr/ucode/ul.wcs 10 blocks

a mnt/usr/ucode/um.wcs 16 blocks
a mnt/usr/ucode/us.wcs 321 blocks
a mnt/usr/ucode/us.200.wcs 321 blocks
a mnt/usr/ucode/vd.wcs 17 blocks
a mnt/usr/ucode/vd.200.wcs 17 blocks
a mnt/usr/scn/lmcm3_rev1 23 blocks
a mnt/usr/scn/lmcm3_rev2 23 blocks
a mnt/usr/scn/lmcm_rev1 14 blocks
a mnt/usr/scn/lmcm_rev2 14 blocks
a mnt/usr/scn/mcm2_rev1 95 blocks
a mnt/usr/scn/mcm3_rev1 161 blocks
a mnt/usr/scn/mcm3_rev2 161 blocks
a mnt/usr/scn/mcm_rev1 91 blocks
a mnt/usr/scn/mcm_rev2 95 blocks
a mnt/usr/scn/tmcm3_rev1 32 blocks
a mnt/usr/scn/tmcm3_rev2 33 blocks
a mnt/usr/scn/lpi2_rev1 10 blocks
a mnt/usr/scn/lpi2_rev2 10 blocks
a mnt/usr/scn/opi2_rev1 39 blocks
a mnt/usr/scn/opi2_rev2 39 blocks
a mnt/usr/scn/pi2_rev1 100 blocks
a mnt/usr/scn/pi2_rev2 101 blocks
a mnt/usr/scn/lpia_rev1 5 blocks
a mnt/usr/scn/lpia_rev2 5 blocks
a mnt/usr/scn/opia_rev1 12 blocks
a mnt/usr/scn/opia_rev2 12 blocks
a mnt/usr/scn/pia_rev1 48 blocks
a mnt/usr/scn/pia_rev2 48 blocks
a mnt/usr/scn/pi2_rev3 101 blocks
a mnt/usr/scn/opi2_rev3 39 blocks
a mnt/usr/scn/lpi2_rev3 10 blocks
a mnt/usr/scn/asp_rev1 38 blocks
a mnt/usr/scn/cpx_rev1 74 blocks
a mnt/usr/scn/cpx_rev2 74 blocks
a mnt/usr/scn/cpx_rev3 77 blocks
a mnt/usr/scn/lcpx_rev1 7 blocks
a mnt/usr/scn/lcpx_rev2 7 blocks
a mnt/usr/scn/lcpx_rev3 7 blocks
a mnt/usr/scn/cue_rev1 126 blocks
a mnt/usr/scn/lcue_rev1 8 blocks
a mnt/usr/scn/cuo_rev1 39 blocks
a mnt/usr/scn/cxm_rev1 62 blocks
a mnt/usr/scn/dcu_rev1 42 blocks
a mnt/usr/scn/edc_rev1 55 blocks
a mnt/usr/scn/edc_rev2 56 blocks
a mnt/usr/scn/efu_rev1 215 blocks
a mnt/usr/scn/efu_rev2 218 blocks
a mnt/usr/scn/ipp_rev1 32 blocks
a mnt/usr/scn/sfu_rev1 88 blocks
a mnt/usr/scn/vpc_rev1 77 blocks
a mnt/usr/scn/vpd_rev1 29 blocks
a mnt/usr/scn/vpdc_rev1 177 blocks
a mnt/usr/lib/DB_cop_proc 5 blocks
a mnt/data/rt.dat 231 blocks
a hw/cputest/-MCS_Product_stamp 0 blocks
a hw/cputest/as_tune 1 blocks
a hw/cputest/asp_func 27 blocks
a hw/cputest/asp_scr 31 blocks
a hw/cputest/clock 3 blocks

a hw/cputest/cpx_func 1 blocks
a hw/cputest/cue_func 16 blocks
a hw/cputest/cue_scr 11 blocks
a hw/cputest/cuo_func 17 blocks
a hw/cputest/dcu_func 10 blocks
a hw/cputest/dcu_scr 15 blocks
a hw/cputest/edc_func 8 blocks
a hw/cputest/efu_func 23 blocks
a hw/cputest/fastdmp_iscn 7 blocks
a hw/cputest/fu_mmqueue 14 blocks
a hw/cputest/halt_off 1 blocks
a hw/cputest/hang 8 blocks
a hw/cputest/hard 15 blocks
a hw/cputest/hwdump 1 blocks
a hw/cputest/hwdump_fast 1 blocks
a hw/cputest/hwdump_iscn 8 blocks
a hw/cputest/hwdump_vp_scn 1 blocks
a hw/cputest/hwdump_vpscn_a 1 blocks
a hw/cputest/hwdump_vpscn_b 1 blocks
a hw/cputest/ipp_scr 7 blocks
a hw/cputest/hwdump_vpscn_c 1 blocks
a hw/cputest/hwdump_vpscn_d 1 blocks
a hw/cputest/hwlog.cpu 7 blocks
a hw/cputest/hwlog.dfw 6 blocks
a hw/cputest/hwlog.hang 6 blocks
a hw/cputest/hwlog.memory 1 blocks
a hw/cputest/hwlog.ringdump 1 blocks
a hw/cputest/hwlog.sh 6 blocks
a hw/cputest/hwlog.vpscn 1 blocks
a hw/cputest/hwlog.xedclip 1 blocks
a hw/cputest/ipp_func 14 blocks
a hw/cputest/load 5 blocks
a hw/cputest/mbus 1 blocks
a hw/cputest/mcm_func 40 blocks
a hw/cputest/mcm_scr 35 blocks
a hw/cputest/mcms_scr 31 blocks
a hw/cputest/mdat_iscn 3 blocks
a hw/cputest/patch.spin 3 blocks
a hw/cputest/patch.uc 9 blocks
a hw/cputest/pi2_func 40 blocks
a hw/cputest/ri_test 39 blocks
a hw/cputest/scn_rings 461 blocks
a hw/cputest/sfu_func 11 blocks
a hw/cputest/sfu_ram 24 blocks
a hw/cputest/sfu_scr 30 blocks
a hw/cputest/sys 1 blocks
a hw/cputest/tipr 2 blocks
a hw/cputest/ustk 2 blocks
a hw/cputest/xfer.scr 27 blocks
a hw/field/INSTALL/ptests 1 blocks
a hw/field/IO/ptests 1 blocks
a hw/field/bin/3.1_BUILD_IN 33 blocks
a hw/field/bin/3.2_BUILD_IN 34 blocks
a hw/field/bin/check_cpus 33 blocks
a hw/field/bin/dcu_edc_check 33 blocks
a hw/field/bin/io_check 33 blocks
a hw/field/bin/marg 26 blocks
a hw/field/bin/mem_ans 39 blocks
a hw/field/bin/pix_check 34 blocks

a hw/field/bin/vers_check 12 blocks
a hw/field/bin/check.old 233 blocks
a hw/field/bin/build_input 34 blocks
a hw/field/bin/3.1_build_in 33 blocks
a hw/field/ptests/cpu4030/DEF/test 1 blocks
a hw/field/ptests/cpu4030/DEF/genans 4 blocks
a hw/field/ptests/cpu4030/FF.CH/test 1 blocks
a hw/field/ptests/cpu4030/FF.CH/genans 4 blocks
a hw/field/ptests/cpu4030/CH/test 1 blocks
a hw/field/ptests/cpu4030/CH/genans 4 blocks
a hw/field/ptests/cpu4231/DEF/subs 1 blocks
a hw/field/ptests/cpu4231/DEF/test 1 blocks
a hw/field/ptests/cpu4231/DEF/genans 4 blocks
a hw/field/ptests/cpu4231/DEF/ctest.def 1 blocks
a hw/field/ptests/cpu4231/DEF/restore 1 blocks
a hw/field/ptests/cpu4231/DEF/setup 6 blocks
a hw/field/ptests/cpu4231/DEF/ctest_3.3 1 blocks
a hw/field/ptests/cpu4231/DEF/ctest.def_3.3 1 blocks
a hw/field/ptests/cpu4231/SEG4/subs 1 blocks
a hw/field/ptests/cpu4231/SEG4/test 1 blocks
a hw/field/ptests/cpu4231/SEG4/genans 4 blocks
a hw/field/ptests/cpu4231/SEG4/setup 4 blocks
a hw/field/ptests/cpu4231/SEG4/ctest.seg4 1 blocks
a hw/field/ptests/cpu4231/SEG4/restore 1 blocks
a hw/field/ptests/cpu4231/CH.SEG4/test 1 blocks
a hw/field/ptests/cpu4231/CH.SEG4/genans 4 blocks
a hw/field/ptests/cpu4231/CH.SEG4/setup 4 blocks
a hw/field/ptests/cpu4231/CH.SEG4/restore 1 blocks
a hw/field/ptests/cpu4231/CH.SEG4/ctest.CH.SEG4 1 blocks
a hw/field/ptests/cpu4232/DEF/test 1 blocks
a hw/field/ptests/cpu4232/DEF/genans 4 blocks
a hw/field/ptests/cpu4232/CH.FF/test 1 blocks
a hw/field/ptests/cpu4232/CH.FF/genans 4 blocks
a hw/field/ptests/cpu4232/CH/test 1 blocks
a hw/field/ptests/cpu4232/CH/genans 4 blocks
a hw/field/ptests/cpu4233/AM/test 1 blocks
a hw/field/ptests/cpu4233/AM/genans 4 blocks
a hw/field/ptests/cpu4233/BM/test 1 blocks
a hw/field/ptests/cpu4233/BM/genans 4 blocks
a hw/field/ptests/cpu4233/CM/test 1 blocks
a hw/field/ptests/cpu4233/CM/genans 4 blocks
a hw/field/ptests/cpu4233/DM/test 1 blocks
a hw/field/ptests/cpu4233/DM/genans 4 blocks
a hw/field/ptests/cpu4233/DM/AM 1 blocks
a hw/field/ptests/cpu4233/DM/BM 1 blocks
a hw/field/ptests/cpu4233/DM/CM 1 blocks
a hw/field/ptests/cpu4233/FF.A/test 1 blocks
a hw/field/ptests/cpu4233/FF.A/genans 4 blocks
a hw/field/ptests/cpu4041/CH.16/test 1 blocks
a hw/field/ptests/cpu4041/CH.16/genans 4 blocks
a hw/field/ptests/cpu4041/CL4.FF/test 1 blocks
a hw/field/ptests/cpu4041/CL4.FF/class 1 blocks
a hw/field/ptests/cpu4041/CL4.FF/genans 4 blocks
a hw/field/ptests/cpu4041/CL1.FF/test 1 blocks
a hw/field/ptests/cpu4041/CL1.FF/class 1 blocks
a hw/field/ptests/cpu4041/CL1.FF/genans 4 blocks
a hw/field/ptests/cpu4041/CL2VXS.FF/subs 1 blocks
a hw/field/ptests/cpu4041/CL2VXS.FF/test 1 blocks
a hw/field/ptests/cpu4041/CL2VXS.FF/genans 4 blocks

a hw/field/ptests/cpu4041/FF.16/test 1 blocks
a hw/field/ptests/cpu4041/FF.16/genans 4 blocks
a hw/field/ptests/cpu4041/MCM.FF/test 1 blocks
a hw/field/ptests/cpu4041/MCM.FF/subs 1 blocks
a hw/field/ptests/cpu4041/MCM.FF/genans 4 blocks
a hw/field/ptests/cpu4041/MCM.FF/C2A.FF 1 blocks
a hw/field/ptests/cpu4041/MCM.FF/C2B.FF 1 blocks
a hw/field/ptests/cpu4041/MCM.FF/CL1.FF 1 blocks
a hw/field/ptests/cpu4041/MCM.FF/CL2VXS.FF 1 blocks
a hw/field/ptests/cpu4041/MCM.FF/CL3VXV.FF 1 blocks
a hw/field/ptests/cpu4041/MCM.FF/CL4.FF 1 blocks
a hw/field/ptests/cpu4041/MCM.FF/FF.16 1 blocks
a hw/field/ptests/cpu4041/CH.128/test 1 blocks
a hw/field/ptests/cpu4041/CH.128/genans 4 blocks
a hw/field/ptests/cpu4041/CL3VXV.FF/test 1 blocks
a hw/field/ptests/cpu4041/CL3VXV.FF/genans 4 blocks
a hw/field/ptests/cpu4041/CL3VXV.FF/subs 1 blocks
a hw/field/ptests/cpu4041/C2A.FF/subs 1 blocks
a hw/field/ptests/cpu4041/C2A.FF/genans 4 blocks
a hw/field/ptests/cpu4041/C2A.FF/test 1 blocks
a hw/field/ptests/cpu4041/C2B.FF/genans 4 blocks
a hw/field/ptests/cpu4041/C2B.FF/test 1 blocks
a hw/field/ptests/cpu4041/C2B.FF/subs 1 blocks
a hw/field/ptests/cpu4241/CH.16/test 1 blocks
a hw/field/ptests/cpu4241/CH.16/genans 4 blocks
a hw/field/ptests/cpu4241/CH.16/setup 4 blocks
a hw/field/ptests/cpu4241/CH.16/restore 1 blocks
a hw/field/ptests/cpu4241/CH.16/ctest.ch.16 1 blocks
a hw/field/ptests/cpu4241/FF.16/test 1 blocks
a hw/field/ptests/cpu4241/FF.16/genans 4 blocks
a hw/field/ptests/cpu4241/FF.16/restore 1 blocks
a hw/field/ptests/cpu4241/FF.16/ctest.ff.16 1 blocks
a hw/field/ptests/cpu4241/FF.16/setup 4 blocks
a hw/field/ptests/cpu4241/ASST.FF/test 1 blocks
a hw/field/ptests/cpu4241/ASST.FF/genans 4 blocks
a hw/field/ptests/cpu4241/ASST.FF/subs 1 blocks
a hw/field/ptests/cpu4241/ASST.FF/setup 4 blocks
a hw/field/ptests/cpu4241/ASST.FF/restore 1 blocks
a hw/field/ptests/cpu4241/ASST.FF/ctest.asst.ff 1 blocks
a hw/field/ptests/cpu4241/CH.128/test 1 blocks
a hw/field/ptests/cpu4241/CH.128/genans 4 blocks
a hw/field/ptests/cpu4241/CH.128/restore 1 blocks
a hw/field/ptests/cpu4241/CH.128/setup 4 blocks
a hw/field/ptests/cpu4241/CH.128/ctest.ch.128 1 blocks
a hw/field/ptests/spu4000/DEF/test 1 blocks
a hw/field/ptests/spu4000/DEF/ans 1 blocks
a hw/field/ptests/io4000/DEF/test 1 blocks
a hw/field/ptests/io4000/DEF/ans 1 blocks
a hw/field/ptests/io4000/DEF/genans 1 blocks
a hw/field/ptests/io4000/DEF/subs 1 blocks
a hw/field/ptests/io5000/DEF/test 1 blocks
a hw/field/ptests/io5000/DEF/ans 1 blocks
a hw/field/ptests/io5000/DEF/genans 1 blocks
a hw/field/ptests/pia4000/DEF/ans 1 blocks
a hw/field/ptests/pia4000/DEF/genans 1 blocks
a hw/field/ptests/pia4000/DEF/test 1 blocks
a hw/field/ptests/pia4000/DEF/subs 1 blocks
a hw/field/ptests/pi2_4000/DEF/ans 1 blocks
a hw/field/ptests/pi2_4000/DEF/genans 1 blocks

a hw/field/ptests/pi2_4000/DEF/test 1 blocks
a hw/field/ptests/idc4000/DEF/ans 1 blocks
a hw/field/ptests/idc4000/DEF/genans 1 blocks
a hw/field/ptests/idc4000/DEF/test 1 blocks
a hw/field/ptests/mem4100/C.A/test 1 blocks
a hw/field/ptests/mem4100/C.A/subs 1 blocks
a hw/field/ptests/mem4100/C.A/genans 3 blocks
a hw/field/ptests/mem4100/C.A/ans 1 blocks
a hw/field/ptests/mem4100/C.B/test 1 blocks
a hw/field/ptests/mem4100/C.B/subs 1 blocks
a hw/field/ptests/mem4100/C.B/genans 3 blocks
a hw/field/ptests/mem4100/C.B/ans 1 blocks
a hw/field/ptests/mem4100/C.C/test 1 blocks
a hw/field/ptests/mem4100/C.C/subs 1 blocks
a hw/field/ptests/mem4100/C.C/genans 3 blocks
a hw/field/ptests/mem4100/C.D/test 1 blocks
a hw/field/ptests/mem4100/C.D/subs 1 blocks
a hw/field/ptests/mem4100/C.D/genans 3 blocks
a hw/field/ptests/mem4100/DEF/test 1 blocks
a hw/field/ptests/mem4100/DEF/genans 3 blocks
a hw/field/ptests/mem4100/DEF/ans 1 blocks
a hw/field/ptests/mem4100/IO/test 1 blocks
a hw/field/ptests/mem4100/IO/subs 1 blocks
a hw/field/ptests/mem4100/IO/genans 3 blocks
a hw/field/ptests/mem4100/IO/ans 1 blocks
a hw/field/scripts/summary 2 blocks
a hw/field/scripts/wr_report 3 blocks
a hw/field/scripts/beep 2 blocks
a hw/field/scripts/grep.spu 36 blocks
a hw/field/scripts/stat_update 2 blocks
a hw/field/scripts/stat_init 2 blocks
a hw/field/scripts/syscon 1 blocks
a hw/field/dbase/fename 1 blocks
a hw/field/dbase/site 1 blocks
a hw/field/dbase/avail 1 blocks
a hw/field/dbase/ptests 2 blocks
a hw/field/dbase/date_stamp 1 blocks
a hw/field/CPU/ptests 1 blocks
a hw/field/MEMORY/ptests 1 blocks
a hw/field/CPU-QUICK/ptests 1 blocks
a hw/field/test/p0r0_4241 35 blocks
a hw/field/test/p0rN_4241 2 blocks
a hw/field/test/cpu4231.rnn 542 blocks
a hw/field/test/p0r0_4231 23 blocks
a hw/field/test/p0rN_4231 30 blocks
a hw/field/test/support_4231 35 blocks
a hw/field/test/cpu4241.t 513 blocks
a hw/field/test/cpu4241.rnn 4014 blocks
a hw/field/test/cpu4241.x00 2814 blocks
a hw/field/test/cpu4231.t 513 blocks
a hw/field/test/list_diag_size 2 blocks
a hw/field/SPRINT/ptests 1 blocks
a hw/field/FS_REV 1 blocks
a hw/field/fs 51 blocks
a hw/field/current 1 blocks
a hw/field/ctest 1 blocks
a hw/field/README 20 blocks
a hw/field/doit 3 blocks
a hw/field/Options 5 blocks

```
a hw/field/summary 2 blocks
a hw/field/scroll 1 blocks
a doc/Readme 122 blocks
================================================================
** Installsw Header File Copy **
Product:      CONVEX C200/C3200 System Diagnostics and Database, Version: V5.0
Release date: February 12, 1992
Directories:    /mnt/bin, /mnt/test, /mnt/usr, /hw/cputest
================================================================
SPU tape device is /dev/rmt1
```

fs1

Field Service Script *fs*

Revision: V5.0

This release of documentation corresponds to revision V5.0 Convex CONVEX C200/C3200 Processor diagnostics and database release.

Warnings

This release of the CONVEX C200/C3200 Series Field Service Script is dependent on V5.1 CONVEX SPU UNIX or greater. It is also advised to run **.diaginit -f** before running the field service script.

Introduction

The field service script is called *fs* and it resides on the spu disk in */hw/field*. The program *fs* is an interactive test that is used to run basic diagnostics from a fixed database of scripts and diagnostic tests. The *fs* package will work on the CONVEX C200/C3200 series (except C210As). The following list shows the scripts available upon entering *fs*:

0) CPU

1) CPU-QUICK

2) INSTALL

3) IO

4) MEMORY

5) SPRINT

Purpose

The *fs* package provides some correlation between CONVEX's Field Service and CONVEX's Manufacturing and Test Operations. The *fs* package allows a group of diagnostics to be run under one script name. The same script, therefore, can be run by manufacturing as well as by field service.

In addition, the scripts are designed to functionally test each subsection of the system or to provide a test for a specified purpose. Some diagnostic tests from a script may be skipped if the system is not configured to run them.

The following table shows each subsection and a corresponding script with approximate times for that script. There may be variations in the times due to different configurations.

Field Service (fs) Scripts				
Subsection	Script	Approx. Time (Hrs.)		
		C210	C220	C240
Scalars/Vectors	SPRINT	0.25	0.50	0.75
Scalars/Vectors	CPU-QUICK	0.5	1.0	2.0
Scalars/Vectors	CPU	2.5	4.5	6.0
I/O	IO	1.5	1.5	3.5
Memory(MCMs)	MEMORY	1.5	3.5	5.0
All	INSTALL	7.0	8.0	9.0

NOTE 1: At least one of these scripts should be run prior to returning a failed field board.

NOTE 2: At installation, the **INSTALL** script should be run.

NOTE 3: For systems with DCUs assembly rev *l* and below, or systems with EDCs assembly rev *e* and below, V3.1 diagnostics of *cpu4231* and *cpu4241* are executed from */hw/field/test* instead of the current diagnostic found in */mnt/test*.

Upon returning a board to CONVEX manufacturing, the generated report should also be returned. The report generated is *report*. To get a hardcopy of the report, do the following:

1. **Turn on console printer (make sure it is on-line)**
2. **Enter, ^*Enter* or ^*PF4* on the keyboard. (^ = control key)**
3. *cat report*

This will give a printout of useful information to CONVEX Product Engineering about the diagnostics executed and a possible failed board.

How to use *fs*

After installing the tape, type the following to invoke *fs*:

1. *cd /hw/field*
2. *fs*
3. A menu will now appear with the following script options:

0) CPU

1) CPU-QUICK

2) INSTALL

3) IO

4) MEMORY

5) SPRINT

Enter the number corresponding to the desired script.

The next line will ask: *Enter FE name:*

Enter the FE name.

Then the script will ask for the customer name: *Enter Customer Name:*

Enter the Customer's site and system serial number.

Then the script will ask to log failures: *Stop on first failure? [y]*

Enter *n* to log failures or *y* to stop on first test failure.

The script will start to execute and check for failures. The screen will be split in two sections. The bottom section gives the status information and the top section is the active window. Note that the bottom section may have failure information. The failure shown is always the last failure that occurred. To get a summary of failures you can type *summary*. This will preen through the *failure* file and print out a summary of all the failures.

When *fs* is done it will write the output to *report* and print

End of Script

Enter ^C to Exit
type 'scroll' for 24 line scrolling
type 'cat report' for report details

Enter *^C* (control-C) to exit *fs*. Type *scroll* to get full screen scrolling.

NOTE 5: Use *scroll* anytime a full screen is desired.

NOTE 6: The bottom part of the screen will have two possible messages as follows:
FAILURES EXIST; SEE report file or
ALL DIAGS PASSED SO FAR

NOTE 7: To resume a script, type *fs resume*.

Margining

Margining is done manually before running *fs*.

fs2

Field Service Script *fs*

File/Directory Description

Directories and Files in */hw/field*

Each available script in *fs* has a corresponding directory of the same name as the script in this central directory. In each of these script directories there is a file called ***ptests***, which contains the margin and ptest definitions for the script. The ptest definition is a key word that describes the diagnostic and option(s) for that diagnostic. An example would be cpu4030/CH.FF. This ptest definition says to run cpu4030, CHained mode, Force Faults.

Data files

These are files that are generally created and/or used by the *fs* software.

ctest - this is what the *doit* script uses after it has invoked dshell. It simply contains all of the information necessary to run a ptest. The file has only one line of information of the form:

test "name of diagnostic" [-s sublist] < ptests/"name of ptest"/ans.

For example, if the ptest being run is cpu4231/DEF, which is cpu4030 subtests 1-31,33-2999, then ctest would be

test cpu4231 -s 1-31,33-2999 <ptests/cpu4231/DEF/ans.

current - contains the script being currently executed. In the above example it would contain *cpu4231/DEF.*

failure - this file is created whenever a failure of a diagnostic occurs. It contains the test and subtest of the failure and the cpu register dump.

log - This contains the output of a normal diagnostic run. This is what is checked by *fs* to determine whether a failure has occurred.

passes - This contains a list of all of the ptests that have passed.

report - This is the final report that is generated by *fs*. It contains the FE name, site, CONVEX SPU OS and diagnostic revisions, a cop.out listing, and any failure or pass information.

The *bin* directory

This directory houses all of the binary executables that are called outside of the main *fs* executable. At this time, the files are:

build_input, 3.1_build_in, 3.2_build_in, 3.1_BUILD_IN, 3.2_BUILD_IN - these are the answer-building binary files for all cpu diagnostics. The *build_input* command will build the appropriate answer file for each cpu diagnostic based on keywords. The *build_input* command is called by the *genans* script which resides in the directory *ptests/<name of diagnostic>/<diag option>* . Two files will reside in the bin directory at one time (*build_input and 3.1_build_in, or build_input and 3.2_build_in),* depending on what version of diagnostics are loaded (V3.1 and below, or V3.2 and above).

check_cpus - returns a code which determines how many cpus are installed in the machine. It is called by *genans* scripts.

dcu_edc_check - command that determines if down-rev'd EDCs or DCUs are in the system. It looks at cop.out and returns one of three codes:

0 - No boards installed, or EDC and DCU installed.

1 - Current system is past EDC rev e or DCU rev l.

2 - Current system is not past EDC rev e or DCU rev l.

marg - Quick margin command. Receives 4 letters corresponding to the following sequence: clk, -2V, -4.5V, +5V. (i.e "marg unln" will margin the clock to upper, -2V to nominal, -4.5V to lower, and +5 V to nominal). This can optionally be called by the *fs* software.

mem_ans - This is the mem4100 answer file generator. At the present time, it only recognizes cpu specification keywords.

io_check - Binary which searches cop.out to see if viops and/or iops are installed. Used to decide to run the io tests (io4000 and io5000).

pix_check - Binary file which searches cop.out to see if c240 pix's or piy's are installed or if pia's are installed. Used to decide if pia4000 and/or pi2_4000 needs to be run.

vers_check - Binary file used to check to see what version of diagnostics are loaded. It looks at */mnt/DIAG_REV* for the revision and returns the following:

0 - Error, could not find */mnt/DIAG_REV.*

1 - System diagnostics is V3.3 or below.

2 - Systems diagnostics is V3.4 or above.

These are used to determine whether to run known failures or not. (See Note 4 of Section 1).

The *dbase* directory

This is a directory that has the database files in it.

avail - a list of available scripts to run.

fename - The Field Engineer (FE) that is running the *fs* program.

site - the name of the system on which the test is running.

ptests - this is a list of the available ptests that exist.

date_stamp - date and time stamp for each diagnostic and script.

Executables

The following executables reside in the */hw/field* directory.

doit - This is the executable that calls the answer file generator, invokes dshell, runs the ptest, and runs the setup and restore scripts.

fs - The main executable for the whole software package.

scroll - Script that sets the screen scrolling region. Without an argument, it sets the screen to full screen scroll. An number (1-24) argument will set the scroll region to that number.

summary - Script that preens through the *failure* file and prints out a summary of the failures, if a *failure* file exists.

The *ptests* directory

This directory has all of the directories and subdirectories that correspond to all of the possible ptests to run. The directories under *ptests* are usually named after a diagnostic. These directories contain subdirectories which represent an option of the specified diagnostic. In each of these subdirectories we have the following files:

test - the diagnostic that this ptest runs.

ans - the static answer file that is used for this ptest.

genans (optional)- the script that will generate the answer file. The output of this script will be *ans*.

subs (optional) - the list of subtests that the ptest runs.

setup (optional) - this does any necessary setups to the system before the ptest is ran.

restore (optional) - this does anything that needs to be done after the ptest is completed.

The *scripts* directory

This contains the scripts (other than *doit* or *genans*) that are called in *fs*. At present, this directory contains the following:

beep - The script that prints the final message at the end of running a script.

wr_report - The script that prepares the *report* file when *fs* is executed.

stat_init - The script that initializes the bottom section of the screen.

stat_update - The script that updates the bottom section of the screen.

syscon - The script that greps for the software versions.