Release Notice
C200 Series Diagnostic Database V4.0

Document No. 760-001030-005

February 1991

CONVEX Computer Corporation

© 1990 CONVEX Computer Corporation

This document is copyrighted. All rights are reserved. CONVEX Computer Corporation (CONVEX) grants that this document may be copied, duplicated, reproduced, translated, stored electronically, or reduced to machine-readable form, provided that such duplications are for internal use only and that they display the CONVEX copyright notice.

Although the material contained herein has been carefully reviewed, CONVEX Computer Corporation does not warrant it to be free of errors or omissions. CONVEX reserves the right to make corrections, updates, revisions or changes to the information contained herein. CONVEX does not warrant the material described herein to be free of patent infringement.

UNLESS PROVIDED OTHERWISE IN WRITING WITH CONVEX COMPUTER CORPORATION (CONVEX), THE SOFTWARE DESCRIBED HEREIN IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES. THE ABOVE EXCLUSION MAY NOT BE APPLICABLE TO ALL PURCHASERS BECAUSE WARRANTY RIGHTS CAN VARY FROM STATE TO STATE. IN NO EVENT WILL CONVEX BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING ANY LOST PROFITS OR LOST SAVINGS, ARISING OUT OF THE USE OR INABILITY TO USE THIS SOFTWARE. CONVEX WILL NOT BE LIABLE EVEN IF IT HAS BEEN NOTIFIED OF THE POSSIBILITY OF SUCH DAMAGE BY THE PURCHASER OR ANY THIRD PARTY.

CONVEX, and C200 Series are trademarks of CONVEX Computer Corporation.

UNIX is a trademark of AT&T Bell Laboratories.

Table of Contents

1 Release Notice
1. Introduction 1-1
2. Contents of This Distribution 1-1
3. Notes and Warnings 1-1
4. Enhancements 1-2
5. Fixes 1-3
6. Known Software Problems 1-3
7. Known Documentation Problems 1-3
8. New Documentation 1-3

Appendices

A Installing C200 Series Diagnostic Database V4.0 ..... A-1
Warnings A-1
Installation Procedure A-1
B Files list B-1
C Documentation Update C-1
Overview C-1

fs1 Field Service Script *fs*
Revision: V4.0 ..... fs1-1
Warnings fs1-1
Introduction fs1-1
Purpose fs1-1
How to use *fs* fs1-3

fs2 Field Service Script *fs*
Directories and Files in */hw/field* fs2-1
Data files fs2-1
The *bin* directory fs2-2
The *dbase* directory fs2-3
Executables fs2-3
The *ptests* directory fs2-3
The *scripts* directory fs2-4

1

Release Notice

1. Introduction

This document describes release V4.0 of the C200 Series Diagnostic Database. 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 the Diagnostic 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 fixes to the current documentation.
- Section 8 contains information on new documentation.
- Appendix A contains instructions for installing this release on a CONVEX Service Processor Unit.
- Appendix B contains a listing of the expected output during installation.
- Appendix C contains documentation for the field service scripts.

The C200 Series Diagnostic Database consists of microcode files, scan-ring definition files, interactive scan scripts, and various other files required by programs and diagnostic utilities which execute under the UNIX Version 7 operating system of the Service Processor Unit. These files 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 the C200 Series Diagnostic Database consists of this document and the distribution media for the software. The specific contents of the software are described in the following table:

C200 Series Diagnostics Database (QIC Tape)

ITEM	QTY	TYPE	PART NUMBER	DESCRIPTION	FORMAT
1.	1	QIC.	760-001315-222	Diagnostic Database, V4.0	Installsw

3. Notes and Warnings

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

- This release supercedes all previous releases of the C200 Series Diagnostic Database.

- Diagnostic Database V4.0 is compatible with certain revisions of CONVEX hardware. Consult the current C200 Series CPU Logic Configurator for applicability of this database to the existing hardware configuration. Functionality on hardware configurations other than those specified by the Logic Configurators is not guaranteed. The following table lists the applicable configurators for the different machines in the C200 Series.

C200 Series Logic Configurator Specification

Logic Configurator	PART NUMBER
CX Configurator	400-000100-979
C2XX Shipping Configurator	400-000100-981

- C200 Series System Diagnostics V3.4 (760-001015-214) is a prerequisite for C200 Series Diagnostics Database V4.0. There will be some diagnostics failures in diagnostics versions older than V3.4 System Diagnostics when running this database. Refer to the Known Software Problems section for details.

- Installation of the C200 Series Diagnostic Database V4.0 tape (760-001315-222) saves the currently installed microcode files in */mnt/usr/ucode.old* and installs the new microcode files in */mnt/usr/ucode*. If necessary, the original microcode files can be restored from the *ucode.old* directory.

- The installation of this release results in the deletion of the following directories prior to loading the V4.0 files from the release tape:

  — */mnt/usr/scn*
  — */hw/cputest*
  — */hw/field*

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

4. Enhancements

- This release contains the new scan ring definitions for the 1230 and 2230 MCM3 memory boards.

- This release contains the new scan ring definitions for the rev 3 3224 PI2 peripheral interface adapter board. This version of the PI2 board fixes a performance problem associated with multiple PI2s in a single system.

- The board configuration file, DB_cop was modified to support the 1230 and 2230 MCM3 memory boards, as well as the 3224 PI2 board

- This release contains the following C200 Series microcode file revisions:

us.200.wcs	11.2
sr.wcs	11.2
vd.200.wcs	10.1
ua.wcs	10.10
um.wcs	10.6
ul.wcs	10.3

- The iscn scripts in the */hw/cputest* directory were modified to recognize and support the MCM3 memory boards.

5. Fixes

None.

6. Known Software Problems

System Diagnostics Expected Failures

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

7. Known Documentation Problems

- Not applicable.

8. New Documentation

- Not applicable.

A

Installing C200 Series Diagnostic Database V4.0

Warnings

This release supercedes all previous releases of the C200 Series Diagnostic Database.

Installation of the C200 Series Diagnostic Database V4.0 tape saves the currently installed microcode files in */mnt/usr/ucode.old* and installs new microcode in */mnt/usr/ucode*.

The installation of this release results in the deletion of the following directories prior to loading the V4.0 files from the release tape:

— */mnt/usr*
— */hw/cputest*

Installation Procedure

1. If SPU UNIX is already booted, go to step 6.

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

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

4. The SPU UNIX bootstrap routine will prompt with:

   SPU UNIX boot
   :

   You should enter a carriage return <CR> in response to the prompt. SPU UNIX 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. If steps 3-5 were skipped, then execute */etc/fsck* manually and correct all errors before proceeding:

(spu)> /etc/fsck -y

If errors are detected in the root file system, then reboot SPU UNIX via */etc/reboot*:

(spu)> /etc/reboot -n

6. Place the C200 Series Diagnostic Database V4.0 tape (760-001315-222) in the cartridge tape unit and enter the following command:

   (spu)> /etc/installsw -i

   When installing this release of the Diagnostic Database, **.diaginit** will run automatically as part of the installation. During the execution of **.diaginit**, the following message may be seen:

   SPU_UNIX: trap hard error

   If this message appears execute **.diaginit -f**, when the **installsw** command finishes.

   The installation of this release requires about 5 minutes. A log of all tar operations is saved in */tmp/installsw.tar*. Once installation has been successfully completed, */tmp/installsw.tar* should be removed as follows:

   (spu)> rm /tmp/installsw.tar

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

8. 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

9. This completes the installation of C200 Series Diagnostic Database V4.0

B
Files list

The following is the *installsw* output from making the V4.0 Diagnostic Database tape.

DATE: Fri Mar 1 01:27:32 CST 1991

```
a /tmp/install1 1 blocks
a /tmp/install2 2 blocks
a usr/lib/DB_cop 5 blocks
a usr/lib/DB_idc 7 blocks
a usr/lib/RLL_2_7 1 blocks
a usr/scn/edc_rev1 55 blocks
a usr/scn/edc_rev2 56 blocks
a usr/scn/idc_rev1 296 blocks
a usr/scn/pia_rev1 48 blocks
a usr/scn/pia_rev2 48 blocks
a usr/scn/dcu_rev1 42 blocks
a usr/scn/pi2_rev1 100 blocks
a usr/scn/pi2_rev2 101 blocks
a usr/scn/mcm_rev1 91 blocks
a usr/scn/mcm_rev2 95 blocks
a usr/scn/mcm2_rev1 95 blocks
a usr/scn/cue_rev1 126 blocks
a usr/scn/fse_rev1 14 blocks
a usr/scn/efu_rev1 215 blocks
a usr/scn/efu_rev2 218 blocks
a usr/scn/lpia_rev1 5 blocks
a usr/scn/lpia_rev2 5 blocks
a usr/scn/asp_rev1 38 blocks
a usr/scn/opia_rev1 12 blocks
a usr/scn/opia_rev2 12 blocks
a usr/scn/lpi2_rev1 10 blocks
a usr/scn/lpi2_rev2 10 blocks
a usr/scn/lmcm_rev1 14 blocks
a usr/scn/lmcm_rev2 14 blocks
a usr/scn/lcue_rev1 8 blocks
a usr/scn/cuo_rev1 39 blocks
a usr/scn/opi2_rev1 39 blocks
a usr/scn/opi2_rev2 39 blocks
a usr/scn/iop_rev1 18 blocks
a usr/scn/cxm_rev1 62 blocks
a usr/scn/vpdc_rev1 177 blocks
a usr/scn/vpc_rev1 77 blocks
a usr/scn/tli_rev1 164 blocks
a usr/scn/ipp_rev1 32 blocks
a usr/scn/vpd_rev1 29 blocks
a usr/scn/hsp_rev1 8 blocks
a usr/scn/cpx_rev1 74 blocks
a usr/scn/cpx_rev2 74 blocks
```

a usr/scn/cpx_rev3 77 blocks
a usr/scn/sfu_rev1 88 blocks
a usr/scn/lcpx_rev1 7 blocks
a usr/scn/lcpx_rev2 7 blocks
a usr/scn/lcpx_rev3 7 blocks
a usr/scn/viop_rev1 18 blocks
a usr/scn/opi2_rev3 39 blocks
a usr/scn/lpi2_rev3 10 blocks
a usr/scn/pi2_rev3 101 blocks
a usr/scn/lmcm3_rev1 23 blocks
a usr/scn/lmcm3_rev2 23 blocks
a usr/scn/mcm3_rev1 161 blocks
a usr/scn/mcm3_rev2 161 blocks
a usr/scn/tmcm3_rev1 32 blocks
a usr/scn/tmcm3_rev2 33 blocks
a usr/ucode/us.200.wcs 321 blocks
a usr/ucode/sr.wcs 25 blocks
a usr/ucode/vd.200.wcs 17 blocks
a usr/ucode/ua.wcs 16 blocks
a usr/ucode/um.wcs 16 blocks
a usr/ucode/ul.wcs 10 blocks
a usr/ucode/UCODE_REV 1 blocks
a cputest/as_tune 1 blocks
a cputest/asp_func 27 blocks
a cputest/asp_scr 31 blocks
a cputest/clock 3 blocks
a cputest/cpx_func 1 blocks
a cputest/cue_func 16 blocks
a cputest/cue_scr 11 blocks
a cputest/cuo_func 17 blocks
a cputest/dcu_func 10 blocks
a cputest/dcu_scr 15 blocks
a cputest/edc_func 8 blocks
a cputest/efu_func 23 blocks
a cputest/fastdmp_iscn 7 blocks
a cputest/fu_mmqueue 14 blocks
a cputest/halt_off 1 blocks
a cputest/hang 8 blocks
a cputest/hard 15 blocks
a cputest/hwdump 1 blocks
a cputest/hwdump_fast 1 blocks
a cputest/hwdump_iscn 8 blocks
a cputest/hwdump_vpscn_a 1 blocks
a cputest/hwdump_vpscn_b 1 blocks
a cputest/hwdump_vpscn_c 1 blocks
a cputest/hwdump_vpscn_d 1 blocks
a cputest/ipp_func 14 blocks
a cputest/ipp_scr 7 blocks
a cputest/load 5 blocks
a cputest/mbus 1 blocks
a cputest/mcm_func 39 blocks
a cputest/mcm_scr 21 blocks
a cputest/mdat_iscn 3 blocks
a cputest/patch.spin 3 blocks
a cputest/patch.uc 9 blocks
a cputest/pi2_func 40 blocks
a cputest/ri_test 39 blocks
a cputest/sfu_func 11 blocks
a cputest/sfu_ram 24 blocks

a cputest/sfu_scr 30 blocks
a cputest/sys 1 blocks
a cputest/tipr 2 blocks
a cputest/ustk 2 blocks
a cputest/xfer.scr 27 blocks
a cputest/hwlog.cpu 7 blocks
a cputest/hwlog.dfw 6 blocks
a cputest/hwlog.hang 6 blocks
a cputest/hwlog.memory 1 blocks
a cputest/hwlog.ringdump 1 blocks
a cputest/hwlog.sh 6 blocks
a cputest/hwlog.vpscn 1 blocks
a cputest/hwlog.xedclip 1 blocks
a field/INSTALL/ptests 1 blocks
a field/IO/ptests 1 blocks
a field/bin/3.2_BUILD_IN 34 blocks
a field/bin/3.1_BUILD_IN 33 blocks
a field/bin/marg 26 blocks
a field/bin/mem_ans 39 blocks
a field/bin/build_input 34 blocks
a field/bin/check.old 233 blocks
a field/bin/io_check 33 blocks
a field/bin/pix_check 34 blocks
a field/bin/3.1_build_in 33 blocks
a field/bin/dcu_edc_check 33 blocks
a field/bin/vers_check 12 blocks
a field/bin/check_cpus 33 blocks
a field/ptests/cpu4030/DEF/test 1 blocks
a field/ptests/cpu4030/DEF/genans 4 blocks
a field/ptests/cpu4030/DEF/ans 1 blocks
a field/ptests/cpu4030/FF.CH/test 1 blocks
a field/ptests/cpu4030/FF.CH/genans 4 blocks
a field/ptests/cpu4030/FF.CH/ans 1 blocks
a field/ptests/cpu4030/CH/test 1 blocks
a field/ptests/cpu4030/CH/genans 4 blocks
a field/ptests/cpu4231/DEF/subs 1 blocks
a field/ptests/cpu4231/DEF/test 1 blocks
a field/ptests/cpu4231/DEF/genans 4 blocks
a field/ptests/cpu4231/DEF/ctest.def 1 blocks
a field/ptests/cpu4231/DEF/restore 1 blocks
a field/ptests/cpu4231/DEF/setup 6 blocks
a field/ptests/cpu4231/DEF/ctest_3.3 1 blocks
a field/ptests/cpu4231/DEF/ctest.def_3.3 1 blocks
a field/ptests/cpu4231/DEF/ans 1 blocks
a field/ptests/cpu4231/SEG4/subs 1 blocks
a field/ptests/cpu4231/SEG4/test 1 blocks
a field/ptests/cpu4231/SEG4/genans 4 blocks
a field/ptests/cpu4231/SEG4/setup 4 blocks
a field/ptests/cpu4231/SEG4/ctest.seg4 1 blocks
a field/ptests/cpu4231/SEG4/restore 1 blocks
a field/ptests/cpu4231/SEG4/ans 1 blocks
a field/ptests/cpu4231/CH.SEG4/test 1 blocks
a field/ptests/cpu4231/CH.SEG4/genans 4 blocks
a field/ptests/cpu4231/CH.SEG4/setup 4 blocks
a field/ptests/cpu4231/CH.SEG4/restore 1 blocks
a field/ptests/cpu4231/CH.SEG4/ctest.CH.SEG4 1 blocks
a field/ptests/cpu4231/CH.SEG4/ans 1 blocks
a field/ptests/cpu4232/DEF/test 1 blocks
a field/ptests/cpu4232/DEF/genans 4 blocks

a field/ptests/cpu4232/DEF/ans 1 blocks
a field/ptests/cpu4232/CH.FF/test 1 blocks
a field/ptests/cpu4232/CH.FF/genans 4 blocks
a field/ptests/cpu4232/CH/test 1 blocks
a field/ptests/cpu4232/CH/genans 4 blocks
a field/ptests/cpu4233/AM/test 1 blocks
a field/ptests/cpu4233/AM/genans 4 blocks
a field/ptests/cpu4233/BM/test 1 blocks
a field/ptests/cpu4233/BM/genans 4 blocks
a field/ptests/cpu4233/CM/test 1 blocks
a field/ptests/cpu4233/CM/genans 4 blocks
a field/ptests/cpu4233/DM/test 1 blocks
a field/ptests/cpu4233/DM/genans 4 blocks
a field/ptests/cpu4233/DM/AM 1 blocks
a field/ptests/cpu4233/DM/BM 1 blocks
a field/ptests/cpu4233/DM/CM 1 blocks
a field/ptests/cpu4233/FF.A/test 1 blocks
a field/ptests/cpu4233/FF.A/genans 4 blocks
a field/ptests/cpu4041/CH.16/test 1 blocks
a field/ptests/cpu4041/CH.16/genans 4 blocks
a field/ptests/cpu4041/CH.16/ans 1 blocks
a field/ptests/cpu4041/CL4.FF/test 1 blocks
a field/ptests/cpu4041/CL4.FF/class 1 blocks
a field/ptests/cpu4041/CL4.FF/genans 4 blocks
a field/ptests/cpu4041/CL4.FF/ans 1 blocks
a field/ptests/cpu4041/CL1.FF/test 1 blocks
a field/ptests/cpu4041/CL1.FF/class 1 blocks
a field/ptests/cpu4041/CL1.FF/genans 4 blocks
a field/ptests/cpu4041/CL1.FF/ans 1 blocks
a field/ptests/cpu4041/CL2VXS.FF/subs 1 blocks
a field/ptests/cpu4041/CL2VXS.FF/test 1 blocks
a field/ptests/cpu4041/CL2VXS.FF/genans 4 blocks
a field/ptests/cpu4041/CL2VXS.FF/ans 1 blocks
a field/ptests/cpu4041/FF.16/test 1 blocks
a field/ptests/cpu4041/FF.16/genans 4 blocks
a field/ptests/cpu4041/MCM.FF/test 1 blocks
a field/ptests/cpu4041/MCM.FF/subs 1 blocks
a field/ptests/cpu4041/MCM.FF/genans 4 blocks
a field/ptests/cpu4041/MCM.FF/C2A.FF 1 blocks
a field/ptests/cpu4041/MCM.FF/C2B.FF 1 blocks
a field/ptests/cpu4041/MCM.FF/CL1.FF 1 blocks
a field/ptests/cpu4041/MCM.FF/CL2VXS.FF 1 blocks
a field/ptests/cpu4041/MCM.FF/CL3VXV.FF 1 blocks
a field/ptests/cpu4041/MCM.FF/CL4.FF 1 blocks
a field/ptests/cpu4041/MCM.FF/FF.16 1 blocks
a field/ptests/cpu4041/CH.128/test 1 blocks
a field/ptests/cpu4041/CH.128/genans 4 blocks
a field/ptests/cpu4041/CL3VXV.FF/test 1 blocks
a field/ptests/cpu4041/CL3VXV.FF/genans 4 blocks
a field/ptests/cpu4041/CL3VXV.FF/subs 1 blocks
a field/ptests/cpu4041/CL3VXV.FF/ans 1 blocks
a field/ptests/cpu4041/C2A.FF/subs 1 blocks
a field/ptests/cpu4041/C2A.FF/genans 4 blocks
a field/ptests/cpu4041/C2A.FF/test 1 blocks
a field/ptests/cpu4041/C2A.FF/ans 1 blocks
a field/ptests/cpu4041/C2B.FF/genans 4 blocks
a field/ptests/cpu4041/C2B.FF/test 1 blocks
a field/ptests/cpu4041/C2B.FF/subs 1 blocks
a field/ptests/cpu4041/C2B.FF/ans 1 blocks

a field/ptests/cpu4241/CH.16/test 1 blocks
a field/ptests/cpu4241/CH.16/genans 4 blocks
a field/ptests/cpu4241/CH.16/setup 4 blocks
a field/ptests/cpu4241/CH.16/restore 1 blocks
a field/ptests/cpu4241/CH.16/ctest.ch.16 1 blocks
a field/ptests/cpu4241/CH.16/ans 1 blocks
a field/ptests/cpu4241/FF.16/test 1 blocks
a field/ptests/cpu4241/FF.16/genans 4 blocks
a field/ptests/cpu4241/FF.16/restore 1 blocks
a field/ptests/cpu4241/FF.16/ctest.ff.16 1 blocks
a field/ptests/cpu4241/FF.16/setup 4 blocks
a field/ptests/cpu4241/ASST.FF/test 1 blocks
a field/ptests/cpu4241/ASST.FF/genans 4 blocks
a field/ptests/cpu4241/ASST.FF/subs 1 blocks
a field/ptests/cpu4241/ASST.FF/setup 4 blocks
a field/ptests/cpu4241/ASST.FF/restore 1 blocks
a field/ptests/cpu4241/ASST.FF/ctest.asst.ff 1 blocks
a field/ptests/cpu4241/ASST.FF/ans 1 blocks
a field/ptests/cpu4241/CH.128/test 1 blocks
a field/ptests/cpu4241/CH.128/genans 4 blocks
a field/ptests/cpu4241/CH.128/restore 1 blocks
a field/ptests/cpu4241/CH.128/setup 4 blocks
a field/ptests/cpu4241/CH.128/ctest.ch.128 1 blocks
a field/ptests/spu4000/DEF/test 1 blocks
a field/ptests/spu4000/DEF/ans 1 blocks
a field/ptests/io4000/DEF/test 1 blocks
a field/ptests/io4000/DEF/ans 1 blocks
a field/ptests/io4000/DEF/genans 1 blocks
a field/ptests/io4000/DEF/subs 1 blocks
a field/ptests/io5000/DEF/test 1 blocks
a field/ptests/io5000/DEF/ans 1 blocks
a field/ptests/io5000/DEF/genans 1 blocks
a field/ptests/pia4000/DEF/ans 1 blocks
a field/ptests/pia4000/DEF/genans 1 blocks
a field/ptests/pia4000/DEF/test 1 blocks
a field/ptests/pia4000/DEF/subs 1 blocks
a field/ptests/pi2_4000/DEF/ans 1 blocks
a field/ptests/pi2_4000/DEF/genans 1 blocks
a field/ptests/pi2_4000/DEF/test 1 blocks
a field/ptests/idc4000/DEF/ans 1 blocks
a field/ptests/idc4000/DEF/genans 1 blocks
a field/ptests/idc4000/DEF/test 1 blocks
a field/ptests/mem4100/C.A/test 1 blocks
a field/ptests/mem4100/C.A/subs 1 blocks
a field/ptests/mem4100/C.A/genans 3 blocks
a field/ptests/mem4100/C.B/test 1 blocks
a field/ptests/mem4100/C.B/subs 1 blocks
a field/ptests/mem4100/C.B/genans 3 blocks
a field/ptests/mem4100/C.C/test 1 blocks
a field/ptests/mem4100/C.C/subs 1 blocks
a field/ptests/mem4100/C.C/genans 3 blocks
a field/ptests/mem4100/C.D/test 1 blocks
a field/ptests/mem4100/C.D/subs 1 blocks
a field/ptests/mem4100/C.D/genans 3 blocks
a field/ptests/mem4100/DEF/test 1 blocks
a field/ptests/mem4100/DEF/genans 3 blocks
a field/ptests/mem4100/DEF/ans 1 blocks
a field/ptests/mem4100/IO/test 1 blocks
a field/ptests/mem4100/IO/subs 1 blocks

a field/ptests/mem4100/IO/genans 3 blocks
a field/scripts/summary 2 blocks
a field/scripts/wr_report 3 blocks
a field/scripts/beep 2 blocks
a field/scripts/grep.spu 36 blocks
a field/scripts/stat_update 2 blocks
a field/scripts/stat_init 2 blocks
a field/scripts/syscon 1 blocks
a field/dbase/fename 1 blocks
a field/dbase/site 1 blocks
a field/dbase/avail 1 blocks
a field/dbase/ptests 2 blocks
a field/dbase/date_stamp 1 blocks
a field/CPU/ptests 1 blocks
a field/MEMORY/ptests 1 blocks
a field/CPU-QUICK/ptests 1 blocks
a field/test/p0r0_4241 35 blocks
a field/test/p0rN_4241 2 blocks
a field/test/cpu4231.rnn 542 blocks
a field/test/p0r0_4231 23 blocks
a field/test/p0rN_4231 30 blocks
a field/test/support_4231 35 blocks
a field/test/cpu4241.t 513 blocks
a field/test/cpu4241.rnn 4014 blocks
a field/test/cpu4241.x00 2814 blocks
a field/test/cpu4231.t 513 blocks
a field/test/list_diag_size 2 blocks
a field/SPRINT/ptests 1 blocks
a field/FS_REV 1 blocks
a field/fs 50 blocks
a field/current 1 blocks
a field/ctest 1 blocks
a field/README 20 blocks
a field/doit 3 blocks
a field/Options 5 blocks
a field/summary 2 blocks
a field/scroll 1 blocks
a /tmp/v4.0ddb.rel 91 blocks

** Installsw Header File Copy **
Product: Diagnostics Data Base, Version: V4.0
Release date: Feb 07, 1991
Directories: /mnt/usr, /hw/cputest, /hw/field
SPU tape device is /dev/rmt1

C

Documentation Update

Overview

The following documentation updates the *CONVEX Diagnostics Utilities Manual (C200 Series)*. These updates will be included in a future revision of this manual.

fs1

Field Service Script *fs*

Revision: V4.0

This release of documentation corresponds to revision V4.0 data base release.

Warnings

This release of the C200 Series Field Service Script is dependent on V5.1 SPU UNIX or greater, and Systems diagnostics V4.0 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 C200 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, spu Unix 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.