



PERQ
Systems
Corporation

**PERQ FAULT DICTIONARY:
THE KEY TO THE
PERQ DIAGNOSTIC DISPLAY**

March 1984

This manual is for use with POS Release G.5 and subsequent releases until further notice.

Copyright(C) 1983, 1984
PERQ Systems Corporation
2600 Liberty Avenue
P. O. Box 2600
Pittsburgh, PA 15230
(412) 355-0900

This document is not to be reproduced in any form or transmitted in whole or in part, without the prior written authorization of PERQ Systems Corporation.

The information in this document is subject to change without notice and should not be construed as a commitment by PERQ Systems Corporation. The company assumes no responsibility for any errors that may appear in this document.

PERQ Systems Corporation will make every effort to keep customers apprised of all documentation changes as quickly as possible. The Reader's Comments card is distributed with this document to request users' critical evaluation to assist us in preparing future documentation.

PERQ and PERQ2 are trademarks of PERQ Systems Corporation.

PERQ FAULT DICTIONARY

<u>Display</u>	<u>Description</u>
000	Boot never got going, StackReset doesn't work or other major problem in the processor board (or clock).
001	Simple Branches fail.
002	Main Data Path Failure.
003	Dual Address failure on Registers.
004	Y Ram Failure.
005	Const/Carry Propagate failure.
006	ALU failure.
007	Conditional Branch failure.
008	Looping failure.
009	Control Store (or Write Control Store) failure.
010	Hung in Disk Boot.
011	Memory Data Error.
012	Memory Address Error.
013	Disk never became ready.
014	Couldn't boot from either disks.
015 - 020	Bad Interrupts Reading Floppy Disk Data.

030	VFY Hung.
050	Bad Error Message from VFY.
051	Empty stack bit not working.
052	Could not load TOS.
053	Push did not work.
054	Stack Empty did not go off.
055	Data error in push.
056	Empty or Full set when that is not the case.
057	Data error in bit 15 of the stack.
058	Stack empty set when the stack is full.
059	Data error on stack.
060	Data error after POP. Bit 14.
061	Data error after POP. Bit 13.
062	Data error after POP. Bit 12.
063	Data error after POP. Bit 11.
064	Data error after POP. Bit 10.
065	Data error after POP. Bit 9.
066	Data error after POP. Bit 8.
067	Data error after POP. Bit 7.
068	Data error after POP. Bit 6.
069	Data error after POP. Bit 5.

070 Data error after POP. Bit 4.
071 Data error after POP. Bit 3.
072 Data error after POP. Bit 2.
073 Empty wrong.
074 Data error after POP. Bit 1.
075 Data error after POP. Bit 0.
076 Empty not set after all pops.
077 Call test failed.
078 Odd didn't jump on a 1.
079 Odd jumped on a 0.

080 Byte sign didn't jump on 200.
081 Byte sign jumped on 0.
082 C19 didn't jump when it should have.
083 BCP[3] didn't jump when it should have.
084 C19 jumped when it shouldn't have.
085 BCP[3] jumped when it shouldn't have.
086 GTR didn't jump.
087 GTR jumped when it shouldn't have.
088 GEQ didn't jump.
089 GEQ jumped when it shouldn't have.

090 LSS didn't jump when it should have.
091 LSS jumped when it shouldn't have.

- 092 LEQ didn't jump.
- 093 LEQ jumped when it shouldn't have.
- 094 GEQ didn't jump on equal.
- 095 LEQ didn't jump on equal.
- 096 Carry didn't jump when it should have.
- 097 Carry jumped when it shouldn't have.
- 098 Overflow didn't jump when it should have.
- 099 Overflow jumped when it shouldn't have.

- 100 And-Not ALU function failed.
- 101 Or ALU function failed.
- 102 Or-Not ALU function failed.
- 103 And ALU function failed.
- 104 Or-Not ALU function failed.
- 105 Not-A ALU function failed.
- 106 Not-B ALU function failed.
- 107 Xor ALU function failed.
- 108 Xnor ALU function failed.
- 109 OldCarry-Add ALU function failed.

- 110 OldCarry-Sub ALU function failed.
- 111 OldCarry-Add /w No OldCarry failed.
- 112 Fetch error on Force Bad Parity.
- 113 Unexpected Parity error.

- 114 No parity errors on force bad parity.
- 115 Wrong address on force bad parity.
- 116 Upper 4 bit test failed.
- 117 MDX test failed.
- 118 Stack upper bits test failed.
- 119 Store/Fetch test failed.
- 120 Unexpected refill.
- 121 BPC test failed.
- 122 Fetch4 test failed.
- 123 Fetch4R test failed.
- 124 Store4 test failed.
- 125 Fetch2 test failed.
- 126 Store2 test failed.
- 127 NextOp test failed.
- 128 Fetch/Store overlap failed.
- 129 Bad interrupt Loc 4.
- 130 Bad interrupt Loc 14.
- 131 Bad interrupt Loc 20.
- 132 Bad interrupt Loc 30.
- 133 Data error on memory sweep.
- 134 Address error on memory sweep.
- 135 Field didn't work.
- 136 Dispatch did not jump.

- 137 Wrong Dispatch target.
- 138 Data error on inverted memory sweep.
- 139 Address error on inverted memory sweep.
- 150 Sysb not loaded correctly or hung.
- 151 Sysb did not complete.
- 152 Illegal Boot Key.
- 153 Hard Disk Restore Failure.
- 154 No such boot.
- 155 No interpreter for that key.
- 156 Interpreter file is empty.
- 157 Disk Error.
- 158 Floppy error.
- 159 Malformed Boot File.
- 160 CheckSum error in microcode.
- 161 CheckSum error in QCode.
- 162 - 168 Bad interrupts.
- 169 Not used
- 170 No ACK from keyboard; on PERQ2 workstations only
- 171 Wrong disk type for this Sysb; on PERQ2 workstations only
- 198 QCode interpreter microcode not entered correctly.
- 199 System not entered - calls or assignments don't work.
- 200 System entered, InitMemory to be called.

- 201 InitMemory entered.
- 203 SAT and SIT pointers set.
- 204 StackSegment number set.
- 205 Reading the BootBlock.
- 206 System version number set.
- 207 Head of free-segment-number list set.
- 208 First system segment number set.
- 209 System boot disk set.

- 210 System boot character set.
- 211 Boot block read.
- 212 Default heap segment number set.
- 213 First used segment number set.
- 214 Before setting freelists of data segments.
- 215 Before trying to allocate a segment number.
- 216 Temporary segment number allocated.
- 217 Ready to enter loop to find memory size.
- 218 Exited from memory size loop.
- 219 Restored mangled word.

- 220 Released temporary segment number.
- 221 Boot file has wrong size.
- 222 Modified the location of I/O segment.
- 223 Adjusted free memory.

- 224 Freelists of data segments set.
- 225 Set screen segment.
- 226 Header buffer allocated for swapping.
- 227 Status buffer allocated for swapping.
- 228 SwappingAllowed set false.
- 229 All boot-loaded segments set UnSwappable (if booted from floppy), InitMemory complete, ready to return to System.

- 230 Starting to increase number of segments allowed (because memory is larger than 1/4 megabyte).
- 231 Changed maximum of SITSeg.
- 232 Changed size of SITSeg.
- 233 Changed maximum of SATSeg.
- 234 Changed size of SATSeg.
- 235 Created new unallocated segment numbers.
- 236 Finished InitMemory.

- 300 InitIO to be called.
- 301 InitIO entered.

- 310 Device Table allocated, calling InitDeviceTable.
- 311 InitDeviceTable entered.
- 312 Allocating the hard disk control block.
- 313 Allocating the EIO Disk Control Block.
- 314 Allocating the pointer's control block.

- 315 Allocating the timer's control block.
- 316 Calling Video - Setup Device Table.
- 331 Video setup device table entered.
- 332 Screen control blocks and display lists allocated.
- 333 Video device table setup complete.
- 350 ScreenInit complete, sending device table to microcode.
- 358 Configuration module initialization to be called.
- 360 StartIO to microcode complete, allocating 280 messages.
- 370 Messages allocated, calling Vid_Initialize.
- 371 Vid_Initialize entered, calling InitTablet.
- 372 InitTablet complete, calling InitCursor.
- 373 InitCursor complete, enabling video interrupts.
- 380 Vid_Initialize complete, calling Key_Initialize.
- 381 Key_Initialize entered, allocating status buffer.
- 382 Status buffer allocated, allocating circular buffers.
- 383 Circular buffer allocated, enabling keyboard interrupts.
- 390 Key_Initialize complete, calling Dsk_Initialize.
- 391 Dsk_initialize entered.

- 392 Disk interrupts enabled, allocating temporary buffers.
- 393 Buffers allocated, calling LocateDskHeads.
- 394 LocateDskHeads entered, about to search for track zero.
- 395 Track zero located.
- 396 LocateDskHeads complete, calling FindSize.
- 397 FindSize entered, about to seek to a 24MByte sector.
- 398 Disk size determined.
- 399 FindSize complete, disposing temporary buffers.
- 400 Dsk_Initialize complete, calling Flp_Initialize.
- 401 Flp_Initialize entered, allocating Floppy status buffer.
- 402 Status buffer allocated, allocating Floppy control block.
- 403 Floppy control block allocated, initializing variables.
- 404 Variables initialized, enabling Floppy interrupts.
- 410 Flp_Initialize complete, calling GPB_Initialize.
- 411 GPB_Initialize entered.
- 412 Allocating the GPIBs High Volume buffer.
- 413 Allocating the GPIBs Status buffer.
- 414 Allocating the GPIBs circular buffer.
- 415 Enabling GPIB interrupts.

- 416 Sensing to see if the GPIB is there.
- 420 - 427 Talking to the GPIB.
- 430 GPB_Initialize complete, calling RS2_Initialize.
- 431 RS2_Initialize entered.
- 432 Allocating an RS232 high volume buffer.
- 433 Allocating an RS232 circular buffer.
- 434 Allocating an RS232 status buffer.
- 435 Enabling RS232 interrupts.
- 436 Allocating temporary buffers.
- 437 Sensing to see if the RS232 is there.
- 438 Disposing of temporary buffers.
- 440 RS232 devices initialization complete.
- 441 Ptr-initialize entered.
- 442 Allocating the pointer's status buffer.
- 443 Enabling pointer interrupts.
- 444 Sensing to see if the pointer is there.
- 445 Turning on the pointer.
- 446 Determining if the pointer is connected.
- 447 Turning off the pointer.
- 450 Ptr_Initialize complete, calling Clk_Initialize.
- 451 Clk_Initialize entered.

- 452 Allocating the clock's status buffer.
- 453 Buffer allocated, enabling Clock interrupts.
- 454 Allocating temporary buffers.
- 455 Sensing to see if the clock is there.
- 456 Disposing of temporary buffers.

- 460 Clk_Initilize complete, calling Z80_Initialize.
- 461 Z80_Initialize entered.
- 462 Allocating the Z80's high volume buffer.
- 463 Allocating the Z80's status buffer.
- 464 Enabling the Z80.
- 465 Allocating temporary buffers.
- 466 Sensing to see if the Z80 is there.
- 467 Disposing of temporary buffers.

- 470 Z80 device initialization complete.

- 499 About to exit InitIO.

- 500 InitIO complete, InitStream to be called.

- 600 InitStream complete, FSInit to be called.

- 700 FSInit complete.

- 800 Command file and Console opened, InitExceptions to be called.

- 810 InitExceptions complete.
- 820 System version number set.
- 822 Current 60 Hz. clock value read.
- 824 60 Hz time reference set, TimeStamp time reference to be set.
- 900 FSSetUpSystem to be called.
- 950 FSSetUpSystem complete.
- 951 About to enable swapping (if booted from hard disk).
- 952 FSLocalLookup and EnableSwapping complete.
- 960 Calling Ethernet initialization.
- 961 E10Init entered.
- 962 Ethernet device table initialization complete.
- 963 EtherSeg created.
- 964 Buffers allocated from EtherSeg.
- 965 EtherSeg made unmoveable.
- 966 Exiting E10Init
- 969 Ethernet initialization complete.
- 970 Loading Z80 from ZBoot file.
- 979 Z80 load complete.
- 980 Loading double precision microcode files.

999 System fully initialized, system title line to
be printed.