USER'S GUIDE

VX305x PBIT

SD.DT.G51-0e - July 2016

POSSIBILITIES START HERE kontron

VX305x PBIT User's Guide

Lise-Meitner-Str. 3-5

86156 Augsburg

Germany

www.kontron.com

REVISION HISTORY

PUBLICATION TITLE:	VX305x PBIT User's Guide	
DOC. ID:	SD.DT.G51-0e	
Revision	Brief Description of Changes	Date of Issue
0e	Initial Issue	07-2016

SYMBOLS

The following symbols may be used in this manual:

⚠DANGER	DANGER indicates a hazardous situation which, if not avoided, will result in death or serious injury.
⚠WARNING	WARNING indicates a hazardous situation which, if not avoided, could result in death or serious injury.
⚠CAUTION	CAUTION indicates a hazardous situation which, if not avoided, may result in minor or moderate injury.
NOTICE	NOTICE indicates a property damage message.
	**Electric Shock!** This symbol and title warn of hazards due to electrical shocks (> 60V) when touching products or parts of them. Failure to observe the precautions indicated and/or prescribed by the law may endanger your life/health and/or result in damage to your material. Please refer also to the "High-Voltage Safety Instructions" portion below in this section.
	**ESD Sensitive Device!** This symbol and title inform that the electronic boards and their components are sensitive to static electricity. Care must therefore be taken during all handling operations and inspections of this product in order to ensure product integrity at all times.
	**HOT Surface!** Do NOT touch! Allow to cool before servicing.
	This symbol indicates general information about the product and the user manual. This symbol also indicates detail information about the specific product configuration.
	This symbol precedes helpful hints and tips for daily use.

FOR YOUR SAFETY

Your new Kontron product was developed and tested carefully to provide all features necessary to ensure its compliance with electrical safety requirements. It was also designed for a long fault-free life. However, the life expectancy of your product can be drastically reduced by improper treatment during unpacking and installation. Therefore, in the interest of your own safety and of the correct operation of your new Kontron product, you are requested to conform with the following guidelines.

High Voltage Safety Instructions

As a precaution, in case of danger, the power connector is the product's main disconnect device and must be easily accessible.

⚠CAUTION

Warning!
All operations on this device must be carried out by sufficiently skilled personnel only.

Caution, Electric Shock!
Before installing a not hot-swappable Kontron product into a system always ensure that your mains power is switched off. This applies also to the installation of piggybacks. Serious electrical shock hazards can exist during all installation, repair and maintenance operations with this product. Therefore, always unplug the power cable and any other cables which provide external voltages before performing work.
Earth ground connection to vehicle's chassis or a central grounding point shall remain connected. The earth ground cable shall be the last disconnected or the first connected during operations of cabling.

Special Handling and Unpacking Instructions

ESD Sensitive Device!
Electronic boards and their components are sensitive to static electricity. Therefore, care must be taken during all handling operations and inspections of this product, in order to ensure product integrity at all times

Do not handle this product out of its protective enclosure while it is not used for operational purposes unless it is otherwise protected.

Whenever possible, unpack or pack this product only at EOS/ESD safe work stations. Where a safe work station is not guaranteed, it is important for the user to be electrically discharged before touching the product with his/her hands or tools. This is most easily done by touching a metal part of your system housing.

It is particularly important to observe standard anti-static precautions when changing piggybacks, ROM devices, jumper settings etc. If the product contains batteries for RTC or memory backup, ensure that the board is not placed on conductive surfaces, including anti-static plastics or sponges. They can cause short circuits and damage the batteries or conductive circuits on the board.

GENERAL INSTRUCTIONS ON USAGE

In order to maintain Kontron's product warranty, this product must not be altered or modified in any way. Changes or modifications to the device, which are not explicitly approved by Kontron and described in this manual or received from Kontron's Technical Support as a special handling instruction, will void your warranty.

This device should only be installed in or connected to systems that fulfill all necessary technical and specific environmental requirements. This applies also to the operational temperature range of the specific board version, which must not be exceeded. If batteries are present, their temperature restrictions must be taken into account.

In performing all necessary installation and application operations, please follow only the instructions supplied by the present manual.

Keep all the original packaging material for future storage or warranty shipments. If it is necessary to store or ship the board, please re-pack it as nearly as possible in the manner in which it was delivered.

Special care is necessary when handling or unpacking the product. Please consult the special handling and unpacking instruction.

ENVIRONMENTAL PROTECTION STATEMENT

This product has been manufactured to satisfy environmental protection requirements where possible. Many of the components used (structural parts, printed circuit boards, connectors, batteries, etc.) are capable of being recycled.

Final disposition of this product after its service life must be accomplished in accordance with applicable country, state, or local laws or regulations.

Environmental protection is a high priority with Kontron.
Kontron follows the DEEE/WEEE directive.
You are encouraged to return our products for proper disposal.

The Waste Electrical and Electronic Equipment (WEEE) Directive aims to:

- reduce waste arising from electrical and electronic equipment (EEE)
- make producers of EEE responsible for the environmental impact of their products, especially when they become waste
- encourage separate collection and subsequent treatment, reuse, recovery, recycling and sound environmental disposal of EEE
- improve the environmental performance of all those involved during the lifecycle of EEE

TRADEMARKS

This document may include names, company logos and trademarks, which are registered trademarks and, therefore, proprietary to their respective owners.

Table Of Contents

1 / PBIT Overview

This document describes the PowerOn Built In Test (PBIT) for Kontron VX305x boards.

The PBIT is an optional product available under the VX305x EFI BIOS shell environment. It is implemented as a binary executable located in the system Flash and included in the BIOS shell application. The PBIT configuration such as tests list and tests result is stored in the VX305x OS EEPROM.

The PBIT includes among others the following services:

- It offers a list of tests that can be added or removed from a run list by command according to the desired trade-off between time to boot, coverage rate and system dependent configuration.
- It also offers a system test that can quickly spot any configuration change.
- It can be run automatically (when booting firmware) or in an interactive mode (at EFI BIOS Shell firmware prompt).
- Tests configuration and results are stored in the OS EEPROM and can also be accessed and reconfigured under Operating System such as Linux. See Chapter 3 page 46.
- Simplified test result is also available in a 8-bit PLD register (register 0x2).

1.1 Related Documents

Hardware:

- VX305x 3U VPX User's Guide CA.DT.B25

BIOS:

- VX305x AMI BIOS User Reference Manual SD.DT.G50

1.2 PBIT Installation and Activation

The PBIT software comes pre–installed in the system Flash, along with the EFI BIOS firmware, on the VX305x boards.

The PBIT can be activated on any VX305x. Please contact Kontron support team for more information.

To install a new BIOS version including a new PBIT version please refer to the VX305x BIOS User's Reference Manual - SD.DT.G50.

1.3 PBIT Configuration

The PBIT must be configured first by an EFI shell command line.

The PBIT is presented as a list of tests to be executed. Each test is focused on a specific device of the VX305x.

The list of tests to be executed can be displayed and modified by using the EFI Shell command "**kdiag**" (see section 1.5 page 7).

1.3.1 Configure the PBIT by command line

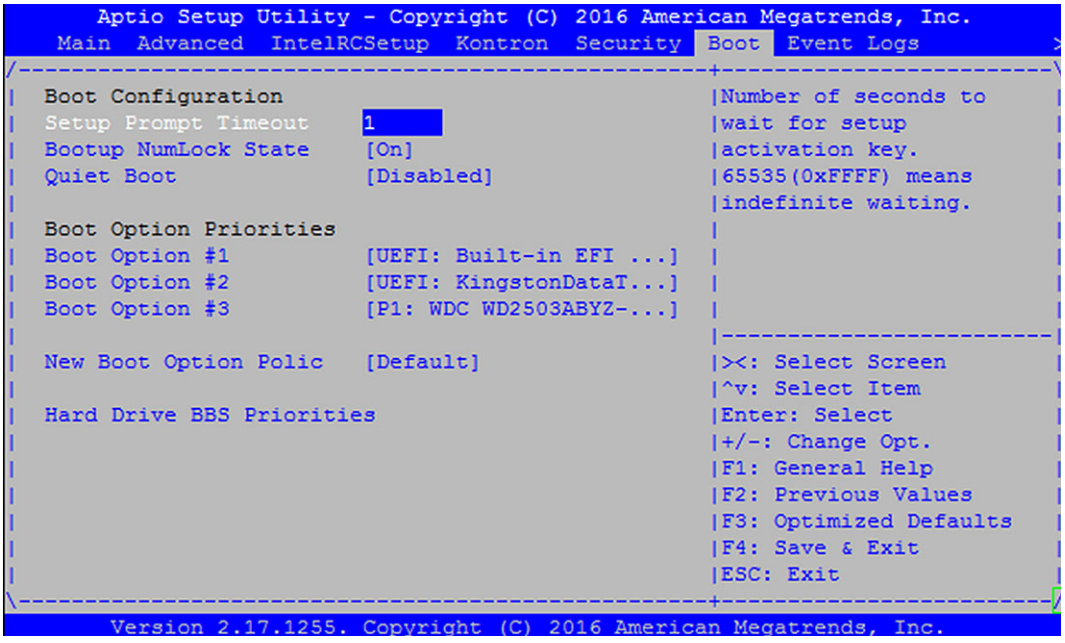
The following explains how to configure and execute the PBIT by a command line.

- Select "**Built-in EFI Shell**" as the first boot device:

  Enter the BIOS setup by pressing the <F2> keyboard key and select the Boot menu.

  Select "**Built-in EFI Shell**" as the Boot Option #1 (use key <+> or <->).

  Then, in the Save & Exit menu select «**Saving Changes and Reset**».



  After reset, the EFI shell prompt is displayed, allowing to enter the PBIT commands.

- Verify the PBIT version:

```
VX3058> kdiag version
PBIT VERSION 1.5 ID16150
```

- Launch the PBIT manually for verification:

```
VX3058> kdiag run
PBIT "mem_data" (fast,simple)  PASSED
PBIT "mem_addr" (fast,simple)  PASSED
PBIT "mem_pattern1" (slow,simple)  PASSED
PBIT "mem_pattern2" (slow,simple)  PASSED
PBIT "mem_pattern3" (slow,simple)  PASSED
PBIT "mem_pattern4" (slow,simple)  PASSED
PBIT "core" (fast,simple)  AES disabled, 8 core board PASSED
PBIT "tpm" (fast,simple)   TPM NOT EQUIPPED PASSED
PBIT "pcie_vpx_sw" (fast,simple)  PASSED
PBIT "m2_bottom" (fast,simple)   M2S2 is SSD-SATA module, TEST BYPASSED PASSED
PBIT "m2_top" (fast,simple)  PASSED
PBIT "serial" (fast,simple)  PASSED
PBIT "rtc" (fast,simple)  PASSED
PBIT "sysflash" (fast,simple)  PASSED
PBIT "cpld" (fast,simple)   PASSED
PBIT "fram" (fast,simple)  PASSED
PBIT "ether_loop0" (fast,simple)  PASSED
PBIT "ether_loop1" (fast,simple)  PASSED
PBIT "ether_loop2" (fast,simple)  PASSED
PBIT "ether_loop3" (fast,simple)  PASSED
PBIT "hwmon" (fast,simple)  PASSED
PBIT "sata0_controler" (fast,simple)  PASSED
PBIT "sata1_controler" (fast,simple)   DISABLED PASSED
PBIT "vpd" (fast,simple)  PASSED
PBIT "eeprom" (fast,simple)  PASSED
PBIT "ehci_controller" (fast,simple)  PASSED
PBIT "xhci_controller" (fast,simple)  PASSED
PBIT "system" (fast,simple)
SYSTEM INFOS NOT SAVED
PASSED
```

- Configure the “system” test:

By default the test named “system” is not “ready”. The end user should record the system configuration when the system is ready for this. All the equipments must be connected such as other CPU boards, XMC, VPX switch boards, Ethernet connections to external switches, USB devices, SATA devices, BIOS setup configuration. See section 2.10 page 23 “PBIT System Test” for more details.

To record and then activate this test , run the following:

```
VX3058> kdiag system_learn

   Seg  Bus  Dev  Func
   ---  ---  ---  ----
    00   00   00    00 ==> Bridge Device - Host/PCI bridge
              Vendor 8086 Device 6F00 Prog Interface 0
    00   00   01    00 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 6F02 Prog Interface 0
    00   00   02    00 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 6F04 Prog Interface 0
    00   00   02    02 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 6F06 Prog Interface 0
    00   00   03    00 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 6F08 Prog Interface 0
    00   00   03    02 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 6F0A Prog Interface 0
    00   00   03    03 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 6F0B Prog Interface 0
```

```
00   00   05    00 ==> Base System Peripherals - Other system peripheral
          Vendor 8086 Device 6F28 Prog Interface 0
00   00   05    01 ==> Base System Peripherals - Other system peripheral
          Vendor 8086 Device 6F29 Prog Interface 0
00   00   05    02 ==> Base System Peripherals - Other system peripheral
          Vendor 8086 Device 6F2A Prog Interface 0
00   00   05    04 ==> Base System Peripherals - PIC
          Vendor 8086 Device 6F2C Prog Interface 20
00   00   14    00 ==> Serial Bus Controllers - USB
          Vendor 8086 Device 8C31 Prog Interface 30
00   00   1C    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 8086 Device 8C10 Prog Interface 0
00   00   1D    00 ==> Serial Bus Controllers - USB
          Vendor 8086 Device 8C26 Prog Interface 20
00   00   1F    00 ==> Bridge Device - PCI/ISA bridge
          Vendor 8086 Device 8C54 Prog Interface 0
00   00   1F    02 ==> Mass Storage Controller - Serial ATA controller
          Vendor 8086 Device 8C02 Prog Interface 1
00   00   1F    03 ==> Serial Bus Controllers - System Management Bus
          Vendor 8086 Device 8C22 Prog Interface 0
00   02   00    00 ==> Base System Peripherals - Other system peripheral
          Vendor 8086 Device 6F50 Prog Interface 0
00   02   00    01 ==> Base System Peripherals - Other system peripheral
          Vendor 8086 Device 6F51 Prog Interface 0
00   02   00    02 ==> Base System Peripherals - Other system peripheral
          Vendor 8086 Device 6F52 Prog Interface 0
00   02   00    03 ==> Base System Peripherals - Other system peripheral
          Vendor 8086 Device 6F53 Prog Interface 0
00   03   00    00 ==> Network Controller - Ethernet controller
          Vendor 8086 Device 15AB Prog Interface 0
00   03   00    01 ==> Network Controller - Ethernet controller
          Vendor 8086 Device 15AB Prog Interface 0
00   04   00    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 10B5 Device 8725 Prog Interface 0
00   04   00    01 ==> Base System Peripherals - Other system peripheral
          Vendor 10B5 Device 87D0 Prog Interface 0
00   04   00    02 ==> Base System Peripherals - Other system peripheral
          Vendor 10B5 Device 87D0 Prog Interface 0
00   04   00    03 ==> Base System Peripherals - Other system peripheral
          Vendor 10B5 Device 87D0 Prog Interface 0
00   04   00    04 ==> Base System Peripherals - Other system peripheral
          Vendor 10B5 Device 87D0 Prog Interface 0
00   05   00    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 10B5 Device 8725 Prog Interface 0
00   05   01    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 10B5 Device 8725 Prog Interface 0
00   05   08    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 10B5 Device 8725 Prog Interface 0
00   07   00    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 111D Device 808C Prog Interface 0
00   08   02    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 111D Device 808C Prog Interface 0
00   08   04    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 111D Device 808C Prog Interface 0
00   08   06    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 111D Device 808C Prog Interface 0
00   08   08    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 111D Device 808C Prog Interface 0
00   08   0C    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 111D Device 808C Prog Interface 0
00   08   10    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 111D Device 808C Prog Interface 0
```

```
    00   08   14    00 ==> Bridge Device - PCI/PCI bridge
              Vendor 111D Device 808C Prog Interface 0
    00   11   00    00 ==> Network Controller - Ethernet controller
              Vendor 8086 Device 1533 Prog Interface 0
    00   12   00    00 ==> Network Controller - Ethernet controller
              Vendor 8086 Device 1533 Prog Interface 0
    00   13   00    00 ==> Display Controller - VGA/8514 controller
              Vendor 126F Device 0750 Prog Interface 0
CPLD Version 0x2
DRAM size 8 GB
System Controller
Geographical Address 1
XMC not present
SATA PORT 0 WDC WD2503ABYZ (251.0GB)
SATA PORT 1 WDC WD2502ABYS (251.0GB)
Device detected on SMBUS0, address = 0x30
BIOS Setup Checksum : 138318
BIOS Version : ID16130
ETH0 connected device speed 1000Mb/s
ETH2 connected device speed 1000Mb/s
      2 Drives, 1 Hub
MassStorage Device name (0) : UFD 3.0 Silicon-Pow
MassStorage Device name (1) : KingstonDataTravele
USB Port 0 connected
USB Port 3 connected

Number of System Test Elements detected : 57
DRAM area [ 0x350377C0  0x35038560 ] will be stored in EEPROM
Storing system infos...

Storing system configuration...
```

- Check the PBIT results:

```
VX3058> kdiag stat
Status of PBITs configured to run from command line :
 PASSED : mem_data(1) (fast,simple)
 PASSED : mem_addr(2) (fast,simple)
 PASSED : mem_pattern1(6) (slow,simple)
 PASSED : mem_pattern2(7) (slow,simple)
 PASSED : mem_pattern3(8) (slow,simple)
 PASSED : mem_pattern4(9) (slow,simple)
 PASSED : core(10) (fast,simple)
 PASSED : tpm(11) (fast,simple)
 PASSED : pcie_vpx_sw(13) (fast,simple)
 PASSED : m2_bottom(14) (fast,simple)
 PASSED : m2_top(15) (fast,simple)
 PASSED : serial(16) (fast,simple)
 PASSED : rtc(20) (fast,simple)
 PASSED : sysflash(22) (fast,simple)
 PASSED : cpld(24) (fast,simple)
 PASSED : fram(40) (fast,simple)
 PASSED : ether_loop0(55) (fast,simple)
 PASSED : ether_loop1(56) (fast,simple)
 PASSED : ether_loop2(57) (fast,simple)
 PASSED : ether_loop3(58) (fast,simple)
 PASSED : hwmon(61) (fast,simple)
 PASSED : sata0_controler(68) (fast,simple)
 PASSED : sata1_controler(69) (fast,simple)
 PASSED : vpd(70) (fast,simple)
```

```
 PASSED : eeprom(71) (fast,simple)
 PASSED : ehci_controller(85) (fast,simple)
 PASSED : xhci_controller(86) (fast,simple)
 PASSED : system(89) (fast,simple)

RUN    : 28
PASSED : 28
FAILED : 0
NOT_RUN: 0
```

1.3.2 Configure the PBIT to run automatically

The PBIT uses BIOS environment variables to run automatically at the end of the BIOS boot and before the Operating System boot:

- Configure the PBIT to be launched at boot time:

The automatic start is activated using the environment variable "**BootCmd**".

```
VX3058> set BootCmd "kdiag run"
```

The delay before executing the **BootCmd** is given by the variable **BootDelay** which is expressed in seconds.

Default value is "**5**". Value "**0**" is possible.

```
VX3058> set BootDelay 1
```

- Verify:

```
VX3058> set
BootCmd = "kdiag run"
BootDelay = 1
    path = .\;FS0:\efi\tools\;FS0:\efi\boot\;FS0:\
debuglasterror = 0x0
lasterror = 0x0
profiles = ;Install;Debug1;Driver1;network1;
uefishellsupport = 3
uefishellversion = 2.1
uefiversion = 2.40
```

- Then reset the system:

```
VX3058> reset
```

The PBIT will be launched automatically. When finished, the BIOS boots from the next valid device in the boot list.

To stop under the EFI shell after the PBIT execution, define the variable named "**StopEfiShell**":

```
VX3058> set StopEfiShell 1
```

1.4 Synthetic PBIT Result

A 8-bit synthetic PBIT result can be read in the CPLD register 0x2.

This register is accessible under the Operating System using the CPLD OS facility or a direct memory I/O access at address 0x802.

Under the BIOS EFI shell use the following command:

```
VX3058> kpld -r 2
READ : @0x2 = 0x1
```

The 8-bit register 0x2 content is the following (reset value=0):

TEST FAIL NUMBER 1.. 128							RUN
7	6	5	4	3	2	1	0

Bit 0: 0 = NOT RUN 1 = ALL RUN

Bit 1..7: if All 0 => No FAILED test, if NOT 0 then indicates ID number of first failing test

- Examples:

0x00 => PBIT not run

0x03 => ALL Tests run and Test 1 FAILED

0x61 => ALL Tests run and Test 48 FAILED (48 = 0x30, 0x30 << 1 = 0x60)

0x01 => ALL Tests run and PASSED

To identify a PBIT test by its number , use the command "**kdiag [PBITnumber]**"
Example:

```
VX3058> kdiag 16
 serial (16) - Checks the serial line COM2
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
```

This register is set to 0 at each hardware reset. It can safely be written to 0 at any time.

1.5 PBIT Tests List

The PBIT tests list comes in two parts: a default list of selected tests and a list of additional not selected tests.

This can be changed by the user to fulfill his specific coverage and execution time requirements. The "**kdiag**" command displays the 2 lists. Note: the initial default tests list can be restored with the "**kdiag default**" command

1.5.1 Selected Tests List

The default selected tests list contains all the diagnostics that can be run without any specific equipment. All the tests have been designed to be safe for the system containing a VX305x. No signal on any connector will be modified during the default tes t execution.

The command "**kdiag**" displays the default tests list to run:

```
VX3058> kdiag
PBITs configured to run from command line :
 mem_data (1) - Checks Memory/ECC data lines
   capabilities : fast,simple,allresets
   run mode 1   : fast,simple,allresets
 mem_addr (2) - Checks Memory/ECC address lines
   capabilities : fast,simple,allresets
   run mode 1   : fast,simple,allresets
 mem_pattern1 (6) - Checks Memory/ECC using pattern 0xFFFFFFFF
   capabilities : slow/fast,simple,allresets
   run mode 1   : slow,simple,allresets
 mem_pattern2 (7) - Checks Memory/ECC using pattern 0x55555555
   capabilities : slow/fast,simple,allresets
   run mode 1   : slow,simple,allresets
 mem_pattern3 (8) - Checks Memory/ECC using pattern 0xAAAAAAAA
   capabilities : slow/fast,simple,allresets
   run mode 1   : slow,simple,allresets
 mem_pattern4 (9) - Checks Memory/ECC using pattern 0x00000000
   capabilities : slow/fast,simple,allresets
   run mode 1   : slow,simple,allresets
 core (10) - Checks Core count and AES disabled
   capabilities : fast,simple,allresets
   run mode 1   : fast,simple,allresets
 tpm (11) - Checks TPM access
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 pcie_vpx_sw (13) - Checks the PCI-Express backplane VPX switch
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 m2_bottom (14) - Checks M.2 bottom module access
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 m2_top (15) - Checks M.2 top module access
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 serial (16) - Checks the serial line COM2
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 rtc (20) - Checks the RTC time
   capabilities : fast,simple,allresets
   run mode 1   : fast,simple,allresets
 sysflash (22) - Checks the BIOS rescue in system flash
   capabilities : fast,simple,allresets
   run mode 1   : fast,simple,allresets
 cpld (24) - Checks PLD, GeoAddress, watchdog
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 fram (40) - Checks F-RAM device.
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 ether_loop0 (55) - Checks ethernet interface 0 in loopback mode
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 ether_loop1 (56) - Checks ethernet interface 1 in loopback mode
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 ether_loop2 (57) - Checks ethernet interface 2 in loopback mode
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
```

```
 ether_loop3 (58) - Checks ethernet interface 3 in loopback mode
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 hwmon (61) - Checks hardware monitoring, temperature and voltage sensors
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 sata0_controler (68) - Checks SATA controller 0 (D31:F2)
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 sata1_controler (69) - Check SATA controller 1 (D31:F5)
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 vpd (70) - Checks VPD data required for board operation.
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 eeprom (71) - Checks User EEPROM (0xA2)
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 ehci_controller (85) - Check EHCI controller
   capabilities : fast,simple,allresets
   run mode 1   : fast,simple,allresets
 xhci_controller (86) - Check xHCI controller
   capabilities : fast,simple,allresets
   run mode 1   : fast,simple,allresets
 system (89) - Checks system configuration SETUP,PCIe,SATA,USB,ETH stability
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
```

To run the default PBIT, enter the command:

```
VX3058> kdiag run
```

1.5.2 Not Selected Tests List

The second part of the list includes all the tests not currently selected for execution. These tests appear at the end of the "**kdiag**" command after the message "**Other PBITs available but not yet configured**":

```
Other PBITs available but not yet configured :

 mem_bitflip (3) - Checks Mem/ECC using bit-flip pattern ((1 << (offset % 64))
   capabilities : slow/fast,simple,allresets
 mem_addrpat (4) - Checks Memory/ECC using address pattern (offset)
   capabilities : slow/fast,simple,allresets
 mem_addrpat2 (5) - Checks Memory/ECC using address pattern (~offset)
   capabilities : slow/fast,simple,allresets
 smbus0 (26) - Check SMBUS0 between PLD and backplane
   capabilities : fast,simple/complex,allresets
 smbus1 (27) - Check SMBUS1 between PLD and backplane
   capabilities : fast,simple/complex,allresets
 ether_link0 (50) - Checks the link status on ethernet interface 0.
   capabilities : fast,simple/complex,allresets
 ether_link1 (51) - Checks the link status on ethernet interface 1.
   capabilities : fast,simple/complex,allresets
 ether_link2 (52) - Checks the link status on ethernet interface 2.
   capabilities : fast,simple/complex,allresets
```

```
 ether_link3 (53) - Checks the link status on ethernet interface 3.
   capabilities : fast,simple/complex,allresets
 sata0_dev_see (62) - Checks if device is present on SATA0
   capabilities : fast,simple,allresets
 sata1_dev_see (63) - Checks if device is present on SATA1
   capabilities : fast,simple,allresets
 sata2_dev_see (64) - Checks if device is present on SATA2
   capabilities : fast,simple,allresets
 sata3_dev_see (65) - Checks if device is present on SATA3
   capabilities : fast,simple,allresets
 sata4_dev_see (66) - Checks if device is present on SATA4 (M.2 Top slot)
   capabilities : fast,simple,allresets
 sata5_dev_see (67) - Checks if device is present on SATA5 (M.2 Bottom slot)
   capabilities : fast,simple,allresets
 usb0_dev_see (80) - Check if a USB device is present on USB2 Front Panel
   capabilities : fast,simple,allresets
 usb1_dev_see (81) - Check if a USB device is present on USB2 port1 (RTM USB C)
   capabilities : fast,simple,allresets
 usb2_dev_see (82) - Check if a USB device is present on USB2 port2 (RTM USB D)
   capabilities : fast,simple,allresets
 usb3_dev_see (83) - Check if a USB device is present on USB2 port 0 or USB3 port 1 (RTM USB A-B)
   capabilities : fast,simple,allresets
 faultytest (98) - A dummy test that returns FAIL
   capabilities : fast,simple,allresets
 hangtest (99) - A dummy test that will hang
   capabilities : fast,simple,allresets

Use "help kdiag" to get more info.
```

The unselected tests list contains:

- Memory tests to complete the default memory tests. Long execution time.
- SMBus tests: I2C tests on the backplane, reserved for complex test with specific external equipment. Do not use it.
- External equipment dependent tests to check Ethernet link, XMC presence, connected SATA and USB Devices. Only use them when it is necessary to verify a device presence.
- Utility tests: "**faultytest**" helps you to test the error reporting mechanism. "**hangtest**" is useful for watchdog recovery checking.

All the USB device detection tests are limited to USB Mass storage devices. They will not detect USB keyboard or mouse. Use system test number 89 if you want to check USB mouse or keyboard presence.

1.6 PBIT Execution Time

The default PBIT (see section 1.5.1 page 7) runs in about 30 seconds for a 8-GB DRAM board.

Below is the execution time for each test and for a board with 8-GB DRAM.

Tests with a significant duration compared to the other tests are highlighted in red.

TEST NAME	TEST ID NUMBER	EXECUTION TIME
mem_data (fast,simple)	1	< 350 ms
mem_addr (fast,simple)	2	800 ms
mem_bitflip (slow,simple)	3	5.5 s
mem_addrpat (slow,simple)	4	4.5 s
mem_addrpat2 (slow,simple)	5	4.9 s
mem_pattern1 (slow,simple)	6	4.6 s
mem_pattern2 (slow,simple)	7	4.8 s
mem_pattern3 (slow,simple)	8	4.8 s
mem_pattern4 (slow,simple)	9	4.8 s
core_dmi (fast,simple)	10	< 200 ms
pcie_vpx_sw (fast,simple)	13	< 200 ms
serial (fast,simple)	16	300 ms
rtc (fast,simple)	20	< 200 ms
ether_link0 (fast,simple)	50	< 200 ms if PASSED but 3s if FAILED
ether_link1 (fast,simple)	51	< 200 ms if PASSED but 3s if FAILED
ether_link2 (fast,simple)	52	< 200 ms if PASSED but 3s if FAILED
ether_link3 (fast,simple)	53	< 200 ms if PASSED but 3s if FAILED
ether_loop0 (fast,simple)	55	800 ms (KR)
ether_loop1 (fast,simple)	56	800 ms (KR)
ether_loop2 (fast,simple)	57	2.6 s
ether_loop3 (fast,simple)	58	2.6 s
hwmon (fast,simple)	61	< 300 ms
sata0_dev_see (fast,simple)	62	< 300 ms
sata1_dev_see (fast,simple)	63	< 300 ms
sata2_dev_see (fast,simple)	64	< 300 ms
sata3_dev_see (fast,simple)	65	< 300 ms
sata4_dev_see (fast,simple)	66	< 300 ms
sata5_dev_see (fast,simple)	67	< 300 ms
sata0_controler (fast,simple)	68	< 300 ms
sata1_controler (fast,simple)	69	< 300 ms
vpd (fast,simple)	70	< 350 ms
usb0_dev_see (fast,simple)	80	< 300 ms
usb1_dev_see (fast,simple)	81	< 300 ms
usb2_dev_see (fast,simple)	82	< 300 ms
usb3_dev_see (fast,simple)	83	< 300 ms
ehci_controller (fast,simple)	85	< 300 ms
xhci_controller (fast,simple)	86	< 300 ms

2 / PBIT Command Line Reference Guide

The PBIT are configured and executed using the EFI Shell command «**kdiag**».

The following section describes the various "**kdiag**" command parameters.

2.1 On-line Help

At EFI Shell prompt enter the command "**help kdiag**" to display the usage messages.

The command formats are:

- [ ] meaning optional parameters
- | meaning a OR choice between possible parameters
- ... meaning an undetermined number of repeated previous parameters

```
VX3058> help kdiag
Kontron board diagnostics command
Print list of PBITs and infos about them :
  kdiag [<PBITname>|<PBITnum> ...]
    <PBITname>|<PBITnum> ...
             list of PBIT(s) to display. All if the list is empty.
             PBIT(s) are referenced using their name or their number.
Run PBIT(s) from command line :
  kdiag run [loop <count>] [<PBITname>|<PBITnum> ...]
    loop <count>
             run PBIT(s) <count> times instead of once
    <PBITname>|<PBITnum> ...
             list of PBIT(s) to run. All if the list is empty.
             PBIT(s) are referenced using their name or their number.
Print PBIT(s) status :
  kdiag stat [<PBITname>|<PBITnum> ...]
    <PBITname>|<PBITnum> ...
             list of PBIT(s) to display. All if the list is empty.
             PBIT(s) are referenced using their name or their number.
Clear PBIT(s) status :
  kdiag clrstat|clrallstat [<PBITname>|<PBITnum> ...]
    clrstat    : Reset status to NOTRUN
    clrallstat : Reset status to NOTRUN and clear the (FAILED once) flag
    <PBITname>|<PBITnum> ...
             list of PBIT(s) to clear. All if the list is empty.
             PBIT(s) are referenced using their name or their number.
Restore default PBIT configuration :
  kdiag default
Delete all PBITs from configuration :
  kdiag deleteall
Configure PBIT(s) :

  kdiag  cfg <cfgarg> ... [<PBITname>|<PBITnum>] ...
   cfg <cfgarg> : Configure one or several PBIT(s).
                  <cfgarg> is either :
                  - "delete" to delete PBIT(s) from the list of configured PBITs
                  - "default" to configure PBIT(s) with a default run mode
```

```
                  - a comma separated list of runflags defining a PBIT
                    run mode; for example : fast,complex.
                  valid runflags are :
                  - "SPEED" flags (can NOT be mixed)
                    slow          : run in slow mode (full testing)
                    fast          : run in fast mode (fast testing)
                  - "CONFIG" flags (can NOT be mixed)
                    simple        : run in simple mode (no external hardware)
                    complex       : run in complex mode (needs external hardware)
                  - "HALT" flags (can NOT be mixed)
                    haltonfail    : halt immediately (hang) if test fails
                    promptonfail  : halt at Firmware prompt (no OS boot) if test fails
                  - "RESET" flags (can be mixed together)
                    normalreset   : run after a normal reset
                    poweronreset  : run after a power-on reset
                    allresets     : run after all resets listed above
    [<PBITname>|<PBITnum>] ...
                  list of PBIT(s) to configure.
                  PBIT(s) are referenced using their name or their number.
                  All configured tests if the list is empty
Toggle PBIT running log information:
  kdiag silent
Toggle PBIT Bypassed:
  kdiag bypass
Record System Configuration for system Test:
  kdiag system_learn or kdiag learn system
Edit System Configuration for system Test:
 (cf documentation for details)
  kdiag system_edit or kdiag edit system
Clear System Configuration for system Test:
  kdiag system_clear or kdiag clear system
Display PBIT version :
  kdiag version
```

2.2 Display the List of Selected Tests

To display the selected tests list and their configuration use the command:

```
"kdiag [<PBITname>|<PBITnum> ...]".
```

Running the command "**kdiag**" with no argument prints the list of the tests that are selected to run with the command "**kdiag run**" and also the other tests not selected by default.

```
VX3058> kdiag
PBITs configured to run from command line :
 mem_data (1) - Checks Memory/ECC data lines
   capabilities : fast,simple,allresets
   run mode 1   : fast,simple,allresets
 mem_addr (2) - Checks Memory/ECC address lines
   capabilities : fast,simple,allresets
   run mode 1   : fast,simple,allresets
 mem_pattern1 (6) - Checks Memory/ECC using pattern 0xFFFFFFFF
   capabilities : slow/fast,simple,allresets
   run mode 1   : slow,simple,allresets
 mem_pattern2 (7) - Checks Memory/ECC using pattern 0x55555555
   capabilities : slow/fast,simple,allresets
   run mode 1   : slow,simple,allresets
```

```
mem_pattern3 (8) - Checks Memory/ECC using pattern 0xAAAAAAAA
  capabilities : slow/fast,simple,allresets
  run mode 1   : slow,simple,allresets
mem_pattern4 (9) - Checks Memory/ECC using pattern 0x00000000
  capabilities : slow/fast,simple,allresets
  run mode 1   : slow,simple,allresets
core (10) - Checks Core count and AES disabled
  capabilities : fast,simple,allresets
  run mode 1   : fast,simple,allresets
tpm (11) - Checks TPM access
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,allresets
pcie_vpx_sw (13) - Checks the PCI-Express backplane VPX switch
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,allresets
m2_bottom (14) - Checks M.2 bottom module access
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,allresets
m2_top (15) - Checks M.2 top module access
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,allresets
serial (16) - Checks the serial line COM2
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,allresets
rtc (20) - Checks the RTC time
  capabilities : fast,simple,allresets
  run mode 1   : fast,simple,allresets
sysflash (22) - Checks the BIOS rescue in system flash
  capabilities : fast,simple,allresets
  run mode 1   : fast,simple,allresets
cpld (24) - Checks PLD, GeoAddress, watchdog
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,allresets
fram (40) - Checks F-RAM device.
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,allresets
ether_loop0 (55) - Checks ethernet interface 0 in loopback mode
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,allresets
ether_loop1 (56) - Checks ethernet interface 1 in loopback mode
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,allresets
ether_loop2 (57) - Checks ethernet interface 2 in loopback mode
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,allresets
ether_loop3 (58) - Checks ethernet interface 3 in loopback mode
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,allresets
hwmon (61) - Checks hardware monitoring, temperature and voltage sensors
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,allresets
sata0_controler (68) - Checks SATA controller 0 (D31:F2)
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,allresets
sata1_controler (69) - Check SATA controller 1 (D31:F5)
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,allresets
vpd (70) - Checks VPD data required for board operation.
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,allresets
```

```
 eeprom (71) - Checks User EEPROM (0xA2)
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 ehci_controller (85) - Check EHCI controller
   capabilities : fast,simple,allresets
   run mode 1   : fast,simple,allresets
 xhci_controller (86) - Check xHCI controller
   capabilities : fast,simple,allresets
   run mode 1   : fast,simple,allresets
 system (89) - Checks system configuration SETUP,PCIe,SATA,USB,ETH stability
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets

Other PBITs available but not yet configured :

 mem_bitflip (3) - Checks Mem/ECC using bit-flip pattern ((1 << (offset % 64))
   capabilities : slow/fast,simple,allresets
 mem_addrpat (4) - Checks Memory/ECC using address pattern (offset)
   capabilities : slow/fast,simple,allresets
 mem_addrpat2 (5) - Checks Memory/ECC using address pattern (~offset)
   capabilities : slow/fast,simple,allresets
 smbus0 (26) - Check SMBUS0 between PLD and backplane
   capabilities : fast,simple/complex,allresets
 smbus1 (27) - Check SMBUS1 between PLD and backplane
   capabilities : fast,simple/complex,allresets
 ether_link0 (50) - Checks the link status on ethernet interface 0.
   capabilities : fast,simple/complex,allresets
 ether_link1 (51) - Checks the link status on ethernet interface 1.
   capabilities : fast,simple/complex,allresets
 ether_link2 (52) - Checks the link status on ethernet interface 2.
   capabilities : fast,simple/complex,allresets
 ether_link3 (53) - Checks the link status on ethernet interface 3.
   capabilities : fast,simple/complex,allresets
 sata0_dev_see (62) - Checks if device is present on SATA0
   capabilities : fast,simple,allresets
 sata1_dev_see (63) - Checks if device is present on SATA1
   capabilities : fast,simple,allresets
 sata2_dev_see (64) - Checks if device is present on SATA2
   capabilities : fast,simple,allresets
sata3_dev_see (65) - Checks if device is present on SATA3
   capabilities : fast,simple,allresets
 sata4_dev_see (66) - Checks if device is present on SATA4 (M.2 Top slot)
   capabilities : fast,simple,allresets
 sata5_dev_see (67) - Checks if device is present on SATA5 (M.2 Bottom slot)
   capabilities : fast,simple,allresets
 usb0_dev_see (80) - Check if a USB device is present on USB2 Front Panel
   capabilities : fast,simple,allresets
 usb1_dev_see (81) - Check if a USB device is present on USB2 port1 (RTM USB C)
   capabilities : fast,simple,allresets
 usb2_dev_see (82) - Check if a USB device is present on USB2 port2 (RTM USB D)
   capabilities : fast,simple,allresets
 usb3_dev_see (83) - Check if a USB device is present on USB2 port 0 or USB3 port 1 (RTM USB A-B)
   capabilities : fast,simple,allresets
 faultytest (98) - A dummy test that returns FAIL
   capabilities : fast,simple,allresets
 hangtest (99) - A dummy test that will hang
   capabilities : fast,simple,allresets

Use "help kdiag" to get more info.
```

2.3 Execute the PBIT from the Command Line

To run the PBIT selected tests list from the command line, enter:

```
PBIT "mem_data" (fast,simple)  PASSED
PBIT "mem_addr" (fast,simple)  PASSED
PBIT "mem_pattern1" (slow,simple)  PASSED
PBIT "mem_pattern2" (slow,simple)  PASSED
PBIT "mem_pattern3" (slow,simple)  PASSED
PBIT "mem_pattern4" (slow,simple)  PASSED
PBIT "core" (fast,simple)  AES disabled, 8 core board PASSED
PBIT "tpm" (fast,simple)   TPM NOT EQUIPPED PASSED
PBIT "pcie_vpx_sw" (fast,simple)  PASSED
PBIT "m2_bottom" (fast,simple)   M2S2 is SSD-SATA module, TEST BYPASSED PASSED
PBIT "m2_top" (fast,simple)  PASSED
PBIT "serial" (fast,simple)  PASSED
PBIT "rtc" (fast,simple)  PASSED
PBIT "sysflash" (fast,simple)  PASSED
PBIT "cpld" (fast,simple)   PASSED
PBIT "fram" (fast,simple)  PASSED
PBIT "ether_loop0" (fast,simple)  PASSED
PBIT "ether_loop1" (fast,simple)  PASSED
PBIT "ether_loop2" (fast,simple)  PASSED
PBIT "ether_loop3" (fast,simple)  PASSED
PBIT "hwmon" (fast,simple)  PASSED
PBIT "sata0_controler" (fast,simple)  PASSED
PBIT "sata1_controler" (fast,simple)   DISABLED PASSED
PBIT "vpd" (fast,simple)  PASSED
PBIT "eeprom" (fast,simple)  PASSED
PBIT "ehci_controller" (fast,simple)  PASSED
PBIT "xhci_controller" (fast,simple)  PASSED
PBIT "system" (fast,simple)
PCI...  CPLD... SATA...  SMBUS...  BIOS...  ETH...  USB...
PASSED
```

To run a single test or a limited list of tests, enter:

```
VX3058> kdiag run <PBIT number | PBIT name ...>
```

- For example:

```
VX3058> kdiag run mem_data core
PBIT "mem_data" (fast,simple)  PASSED
PBIT "core" (fast,simple)  AES disabled, 8 core board PASSED
```

That is equivalent to:

```
VX3058> kdiag run 1 10
PBIT "mem_data" (fast,simple)  PASSED
PBIT "core" (fast,simple)  AES disabled, 8 core board PASSED
```

The “PBIT number” is displayed by the “**kdiag**” command (with no parameter) or with the "**kdiag <PBIT name>**" command.

2.4 Execute the PBIT in Loop Mode

To run the PBIT in loop mode, enter:

```
VX3058 > kdiag run loop <count>
```

<count> is the number of loop to execute.

To run a single test in loop mode, enter:

```
VX3058> kdiag run loop <count> <PBIT number | PBIT name ...>
```

- Example: running 10 times the test ether_loop0 number 55, enter:

```
VX3058> kdiag run loop 10 55
```

It is equivalent to enter:

```
VX3058> kdiag run loop 10 ether_loop0
```

2.5 Get the PBIT Results

To get the PBIT results, use the "**kdiag stat**" command:

```
VX3058> kdiag stat
Status of PBITs configured to run from command line :
 PASSED : mem_data(1) (fast,simple)
 PASSED : mem_addr(2) (fast,simple)
 PASSED : mem_pattern1(6) (slow,simple)
 PASSED : mem_pattern2(7) (slow,simple)
 PASSED : mem_pattern3(8) (slow,simple)
 PASSED : mem_pattern4(9) (slow,simple)
 PASSED : core(10) (fast,simple)
 PASSED : tpm(11) (fast,simple)
 PASSED : pcie_vpx_sw(13) (fast,simple)
 PASSED : m2_bottom(14) (fast,simple)
 PASSED : m2_top(15) (fast,simple)
 PASSED : serial(16) (fast,simple)
 PASSED : rtc(20) (fast,simple)
 PASSED : sysflash(22) (fast,simple)
 PASSED : cpld(24) (fast,simple)
 PASSED : fram(40) (fast,simple)
 PASSED : ether_loop0(55) (fast,simple)
 PASSED : ether_loop1(56) (fast,simple)
 PASSED : ether_loop2(57) (fast,simple)
 PASSED : ether_loop3(58) (fast,simple)
 PASSED : hwmon(61) (fast,simple)
 PASSED : sata0_controler(68) (fast,simple)
 PASSED : sata1_controler(69) (fast,simple)
 PASSED : vpd(70) (fast,simple)
 PASSED : eeprom(71) (fast,simple)
 PASSED : ehci_controller(85) (fast,simple)
 PASSED : xhci_controller(86) (fast,simple)
 PASSED : system(89) (fast,simple)

RUN    : 28
PASSED : 28
FAILED : 0
NOT_RUN: 0
```

2.6 Clear the PBIT Results

Upon failure of any test, a specific "**FAILED ONCE**" flag is set. This flag is kept even if this test is successfully PASSED later. This feature has been designed to keep track of intermittent failures.

- To clear the PBIT results enter:

```
VX3058> kdiag clrstat
```

- To clear all the PBIT history including the "**FAILED ONCE**" flags, enter:

```
VX3058> kdiag clrallstat
```

2.7 Configure the PBIT Tests List to Execute

The list of tests to execute can be modified with the "**kdiag**" command.

Each test can be added, removed and configured with a specific run mode. If no run mode is specified then the default run mode (fast,simple) is applied.

2.7.1 Run mode parameters

The possible specific run modes are defined with the following test flags:

- **HALT** flag (can NOT be mixed):

  haltonfail: halt immediately (hang) if test fails

  promptonfail: halt at BIOS prompt (no OS boot) if test fails

  This flag offers the possibility to halt all test execution when an error is detected.

- **SPEED** flag (can NOT be mixed)

  slow: run in slow mode (full testing)

  fast: run in fast mode (fast testing)

  In the current PBIT version, no test implements a difference between fast and slow modes.

- **CONFIG** flag (can NOT be mixed)

  simple: run in simple mode (no external hardware)

  complex: run in complex mode (needs external hardware)

  Complex mode requires external devices to the VX305x and is reserved for the factory tests.

- **RESET** flag

  normalreset: run after a normal reset (means a board warm boot).

  poweronreset: run after a power-on reset (PBIT with this flag will run only when board is powered ON or cold boot)

  allresets: run after all resets listed above

2.7.2 Adding a Test to the Current Run List

To add a test to the run list, enter:

```
VX3058> kdiag cfg <cfgarg> ... <PBITname>|<PBITnum> ...
```

"**cfgarg**" allows to choose the test run mode.

Use the keyword "**default**" to set the default mode (typically fast and simple)

To add test 16 (serial test) with default mode, enter:

```
VX3058> kdiag cfg default 16
```

To add this test with the promptonfail flag (and the other default flags), enter:

```
VX3058> kdiag cfg promptonfail 16
```

To add this test with the complex and promptonfail flags enter:

```
VX3058> kdiag cfg complex,promptonfail 16
```

Execute the command "**kdiag**" or "**kdiag 16**" to check the configuration:

```
VX3058> kdiag 16
 serial (16) - Checks the serial line COM2
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,complex,promptonfail,allresets
```

2.7.3 Removing a Test from the Current Run List

For example, to remove test number 16, enter:

```
VX3058> kdiag cfg delete 16
```

Verify:

```
VX3058> kdiag 16
PBIT "16" is not configured to run from command line
```

To remove all the tests from the current run list, enter:

```
VX3058> kdiag deleteall
```

Don't forget to add some tests to the current list before running kdiag:

```
VX3058> kdiag run
WARNING: No PBIT will be run because run list is empty
```

2.7.4 Set a RUN mode parameter to All the Tests of the Current Run List

The command **kdiag cfg** "**runflag**" allows to set a "**runflag**" (HALT, CONFIG, RESET described in section 2.7.1 page 18) to all current run list tests.

For example to set default run mode for all tests of current run list:

```
VX3058> kdiag cfg default
Configuration of current tests in asked mode:
command,fast,simple,allresets
List of test to configure
--> mem_data (1) - Checks Memory/ECC data lines
--> mem_addr (2) - Checks Memory/ECC address lines
--> mem_pattern1 (6) - Checks Memory/ECC using pattern 0xFFFFFFFF
--> mem_pattern2 (7) - Checks Memory/ECC using pattern 0x55555555
--> mem_pattern3 (8) - Checks Memory/ECC using pattern 0xAAAAAAAA
--> mem_pattern4 (9) - Checks Memory/ECC using pattern 0x00000000
--> core (10) - Checks Core count and AES disabled
--> tpm (11) - Checks TPM access
--> pcie_vpx_sw (13) - Checks the PCI-Express backplane VPX switch
--> m2_bottom (14) - Checks M.2 bottom module access
--> m2_top (15) - Checks M.2 top module access
--> serial (16) - Checks the serial line COM2
--> rtc (20) - Checks the RTC time
```

```
--> sysflash (22) - Checks the BIOS rescue in system flash
--> cpld (24) - Checks PLD, GeoAddress, watchdog
--> fram (40) - Checks F-RAM device.
--> ether_loop0 (55) - Checks ethernet interface 0 in loopback mode
--> ether_loop1 (56) - Checks ethernet interface 1 in loopback mode
--> ether_loop2 (57) - Checks ethernet interface 2 in loopback mode
--> ether_loop3 (58) - Checks ethernet interface 3 in loopback mode
--> hwmon (61) - Checks hardware monitoring, temperature and voltage sensors
--> sata0_controler (68) - Checks SATA controller 0 (D31:F2)
--> sata1_controler (69) - Check SATA controller 1 (D31:F5)
--> vpd (70) - Checks VPD data required for board operation.
--> eeprom (71) - Checks User EEPROM (0xA2)
--> ehci_controller (85) - Check EHCI controller
--> xhci_controller (86) - Check xHCI controller
--> system (89) - Checks system configuration SETUP,PCIe,SATA,USB,ETH stability
```

For example to set promptonfail, poweronreset run mode for all tests of current run list:

```
VX3058> kdiag cfg promptonfail,poweronreset
Configuration of current tests in asked mode:
command,fast,simple,promptonfail,poweronreset
List of test to configure
--> mem_data (1) - Checks Memory/ECC data lines
--> mem_addr (2) - Checks Memory/ECC address lines
--> mem_pattern1 (6) - Checks Memory/ECC using pattern 0xFFFFFFFF
--> mem_pattern2 (7) - Checks Memory/ECC using pattern 0x55555555
--> mem_pattern3 (8) - Checks Memory/ECC using pattern 0xAAAAAAAA
--> mem_pattern4 (9) - Checks Memory/ECC using pattern 0x00000000
--> core (10) - Checks Core count and AES disabled
--> tpm (11) - Checks TPM access
--> pcie_vpx_sw (13) - Checks the PCI-Express backplane VPX switch
--> m2_bottom (14) - Checks M.2 bottom module access
--> m2_top (15) - Checks M.2 top module access
--> serial (16) - Checks the serial line COM2
--> rtc (20) - Checks the RTC time
--> sysflash (22) - Checks the BIOS rescue in system flash
--> cpld (24) - Checks PLD, GeoAddress, watchdog
--> fram (40) - Checks F-RAM device.
--> ether_loop0 (55) - Checks ethernet interface 0 in loopback mode
--> ether_loop1 (56) - Checks ethernet interface 1 in loopback mode
--> ether_loop2 (57) - Checks ethernet interface 2 in loopback mode
--> ether_loop3 (58) - Checks ethernet interface 3 in loopback mode
--> hwmon (61) - Checks hardware monitoring, temperature and voltage sensors
--> sata0_controler (68) - Checks SATA controller 0 (D31:F2)
--> sata1_controler (69) - Check SATA controller 1 (D31:F5)
--> vpd (70) - Checks VPD data required for board operation.
--> eeprom (71) - Checks User EEPROM (0xA2)
--> ehci_controller (85) - Check EHCI controller
--> xhci_controller (86) - Check xHCI controller
--> system (89) - Checks system configuration SETUP,PCIe,SATA,USB,ETH stability
```

Verify with **kdiag** command, the run mode is promptonfail, poweronreset for all tests in the run list:

```
VX3058> kdiag
PBITs configured to run from command line :
 mem_data (1) - Checks Memory/ECC data lines
   capabilities : fast,simple,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
```

```
 mem_addr (2) - Checks Memory/ECC address lines
   capabilities : fast,simple,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
 mem_pattern1 (6) - Checks Memory/ECC using pattern 0xFFFFFFFF
   capabilities : slow/fast,simple,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
 mem_pattern2 (7) - Checks Memory/ECC using pattern 0x55555555
   capabilities : slow/fast,simple,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
 mem_pattern3 (8) - Checks Memory/ECC using pattern 0xAAAAAAAA
   capabilities : slow/fast,simple,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
 mem_pattern4 (9) - Checks Memory/ECC using pattern 0x00000000
   capabilities : slow/fast,simple,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
 core (10) - Checks Core count and AES disabled
   capabilities : fast,simple,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
 tpm (11) - Checks TPM access
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
 pcie_vpx_sw (13) - Checks the PCI-Express backplane VPX switch
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
 m2_bottom (14) - Checks M.2 bottom module access
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
 m2_top (15) - Checks M.2 top module access
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
 serial (16) - Checks the serial line COM2
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
 rtc (20) - Checks the RTC time
   capabilities : fast,simple,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
 sysflash (22) - Checks the BIOS rescue in system flash
   capabilities : fast,simple,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
 cpld (24) - Checks PLD, GeoAddress, watchdog
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
fram (40) - Checks F-RAM device.
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
 ether_loop0 (55) - Checks ethernet interface 0 in loopback mode
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
 ether_loop1 (56) - Checks ethernet interface 1 in loopback mode
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
 ether_loop2 (57) - Checks ethernet interface 2 in loopback mode
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
 ether_loop3 (58) - Checks ethernet interface 3 in loopback mode
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
 hwmon (61) - Checks hardware monitoring, temperature and voltage sensors
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
```

```
sata0_controler (68) - Checks SATA controller 0 (D31:F2)
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,promptonfail,poweronreset
sata1_controler (69) - Check SATA controller 1 (D31:F5)
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,promptonfail,poweronreset
vpd (70) - Checks VPD data required for board operation.
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,promptonfail,poweronreset
eeprom (71) - Checks User EEPROM (0xA2)
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,promptonfail,poweronreset
ehci_controller (85) - Check EHCI controller
  capabilities : fast,simple,allresets
  run mode 1   : fast,simple,promptonfail,poweronreset
xhci_controller (86) - Check xHCI controller
  capabilities : fast,simple,allresets
  run mode 1   : fast,simple,promptonfail,poweronreset
system (89) - Checks system configuration SETUP,PCIe,SATA,USB,ETH stability
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,promptonfail,poweronreset
```

Verify for test 13:

```
VX3058> kdiag 13
 pcie_vpx_sw (13) - Checks the PCI-Express backplane VPX switch
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
```

The command will not include the test numbers greater than 95 (hangtest, faulty) because these tests are tool tests for PBIT validation.

2.7.5 Restore the Default Run List

To restore the default run tests list, enter:

```
VX3058> kdiag default
```

2.8 Run the PBIT in Silent Mode

To avoid the PBIT to display test messages during execution, use the toggle command:

```
VX3058> kdiag silent
PBIT set in silent mode
```

To disable the silent mode, re-execute the same command:

```
VX3058> kdiag silent
PBIT silent mode removed
```

1. In this mode error messages are displayed anyway.
2. To prevent any output messages to the serial line, use the BIOS setup configuration (Serial Line Console Redirection).

2.9 Display the PBIT Version

To display the PBIT version, enter:

```
VX3058> kdiag version
PBIT VERSION 1.5 ID16150
```

2.10 PBIT System Test

2.10.1 Recording the System Configuration

The command "**kdiag system_learn**" or "**kdiag learn system**" is used to record the current system configuration. It might be run when the system configuration is the correct one to be recorded.

It records:

- All detected PCI devices (the list is visible with the BIOS shell command "**pci**"),
- PCIe devices infos (vendorID, deviceID, ClassCode) from detected PCI devices,
- PCIe link width and speed for PCI/PCI bridge devices,
- SATA information (port connected, disk name, disk size),
- USB information (ort connected, keyboard, mice, mass storage device name(s)),
- Ethernet information (port link up, speed),
- BIOS information: BIOS ID and checksum (checksum of BIOS setup),
- Information from CPLD registers: CPLD Version, DRAM size, system controller status, geographical address, XMC presence,
- SMBUS connected device's addresses on backplane SMBUS0 and SMBUS1.

Here is an example of the command on a VX3058 board:

```
VX3058> kdiag learn system
   Seg  Bus  Dev  Func
   ---  ---  ---  ----
    00   00   00    00 ==> Bridge Device - Host/PCI bridge
              Vendor 8086 Device 6F00 Prog Interface 0
    00   00   01    00 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 6F02 Prog Interface 0
    00   00   02    00 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 6F04 Prog Interface 0
    00   00   02    02 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 6F06 Prog Interface 0
    00   00   03    00 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 6F08 Prog Interface 0
    00   00   03    02 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 6F0A Prog Interface 0
    00   00   03    03 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 6F0B Prog Interface 0
    00   00   05    00 ==> Base System Peripherals - Other system peripheral
              Vendor 8086 Device 6F28 Prog Interface 0
    00   00   05    01 ==> Base System Peripherals - Other system peripheral
              Vendor 8086 Device 6F29 Prog Interface 0
```

```
00   00   05    02 ==> Base System Peripherals - Other system peripheral
          Vendor 8086 Device 6F2A Prog Interface 0
00   00   05    04 ==> Base System Peripherals - PIC
          Vendor 8086 Device 6F2C Prog Interface 20
00   00   14    00 ==> Serial Bus Controllers - USB
          Vendor 8086 Device 8C31 Prog Interface 30
00   00   1D    00 ==> Serial Bus Controllers - USB
          Vendor 8086 Device 8C26 Prog Interface 20
00   00   1F    00 ==> Bridge Device - PCI/ISA bridge
          Vendor 8086 Device 8C54 Prog Interface 0
00   00   1F    02 ==> Mass Storage Controller - Serial ATA controller
          Vendor 8086 Device 8C02 Prog Interface 1
00   00   1F    03 ==> Serial Bus Controllers - System Management Bus
          Vendor 8086 Device 8C22 Prog Interface 0
00   02   00    00 ==> Base System Peripherals - Other system peripheral
          Vendor 8086 Device 6F50 Prog Interface 0
00   02   00    01 ==> Base System Peripherals - Other system peripheral
          Vendor 8086 Device 6F51 Prog Interface 0
00   02   00    02 ==> Base System Peripherals - Other system peripheral
          Vendor 8086 Device 6F52 Prog Interface 0
00   02   00    03 ==> Base System Peripherals - Other system peripheral
          Vendor 8086 Device 6F53 Prog Interface 0
00   03   00    00 ==> Network Controller - Ethernet controller
          Vendor 8086 Device 15AB Prog Interface 0
00   03   00    01 ==> Network Controller - Ethernet controller
          Vendor 8086 Device 15AB Prog Interface 0
00   04   00    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 10B5 Device 8725 Prog Interface 0
00   04   00    01 ==> Base System Peripherals - Other system peripheral
          Vendor 10B5 Device 87D0 Prog Interface 0
00   04   00    02 ==> Base System Peripherals - Other system peripheral
          Vendor 10B5 Device 87D0 Prog Interface 0
00   04   00    03 ==> Base System Peripherals - Other system peripheral
          Vendor 10B5 Device 87D0 Prog Interface 0
00   04   00    04 ==> Base System Peripherals - Other system peripheral
        Vendor 10B5 Device 87D0 Prog Interface 0
00   05   00    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 10B5 Device 8725 Prog Interface 0
00   05   01    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 10B5 Device 8725 Prog Interface 0
00   05   08    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 10B5 Device 8725 Prog Interface 0
00   07   00    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 111D Device 808C Prog Interface 0
00   08   02    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 111D Device 808C Prog Interface 0
00   08   04    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 111D Device 808C Prog Interface 0
00   08   06    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 111D Device 808C Prog Interface 0
00   08   08    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 111D Device 808C Prog Interface 0
00   08   0C    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 111D Device 808C Prog Interface 0
00   08   10    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 111D Device 808C Prog Interface 0
00   08   14    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 111D Device 808C Prog Interface 0
00   11   00    00 ==> Network Controller - Ethernet controller
          Vendor 8086 Device 1533 Prog Interface 0
00   12   00    00 ==> Network Controller - Ethernet controller
          Vendor 8086 Device 1533 Prog Interface 0
```

```
CPLD Version 0x3
DRAM size 8 GB
System Controller
Geographical Address 1
XMC not present
SATA PORT 0 WDC WD2503ABYZ (251.0GB)
SATA PORT 1 WDC WD2502ABYS (251.0GB)
SATA PORT 4 TS32GMTS400    (32.0GB)
SATA PORT 5 TS32GMTS400    (32.0GB)
Device detected on SMBUS0, address = 0x30
BIOS Setup Checksum : 138322
BIOS Version : ID16133
ETH0 connected device speed 1000Mb/s
ETH2 connected device speed 1000Mb/s
ETH3 connected device speed 1000Mb/s
      4 Drives, 1 Hub
MassStorage Device name (0) : KingstonDataTravele
MassStorage Device name (1) : KingstonDataTravele
MassStorage Device name (2) : UFD 3.0 Silicon-Pow
MassStorage Device name (3) : KingstonDataTravele
USB Port 0 connected
USB Port 1 connected
USB Port 2 connected
USB Port 3 connected

Number of System Test Elements detected : 60
DRAM area [ 0x354F0BC0  0x354F1960 ] will be stored in EEPROM
Storing system infos...

Storing system configuration...
```

2.10.2 Testing the Current System Configuration

The test named "**system**"(89) is used to read the current system configuration and check it against the recorded configuration. The command used is "**kdiag run system**" or "**kdiag system_run**". By default, the system test prints errors only, and details level is NORMAL.

The following is an example with several detected failures on a VX3058 board:

```
VX3058> kdiag run system
PBIT "system" (fast,simple)
PCI...  CPLD... SATA...  SMBUS...  BIOS...  ETH...  USB...
ERR)PCI 00:1F:02 DeviceID  (R)0x8C02                (D)0x8C00
ERR)PCI 00:1F:02 ClassCode (R)010601                (D)010601
ERR)PCI Bus:Dev:Func       (R)02:00:00              (D)00:1F:05
ERR)PCI 02:00:00 DeviceID  (R)0x6F50                (D)0x8C08
ERR)PCI 02:00:00 ClassCode (R)088000                (D)088000
ERR)PCI Bus:Dev:Func       (R)02:00:01              (D)02:00:00
ERR)PCI 02:00:01 DeviceID  (R)0x6F51                (D)0x6F50
ERR)PCI Bus:Dev:Func       (R)02:00:02              (D)02:00:01
ERR)PCI 02:00:02 DeviceID  (R)0x6F52                (D)0x6F51
ERR)PCI Bus:Dev:Func       (R)02:00:03              (D)02:00:02
ERR)PCI 02:00:03 DeviceID  (R)0x6F53                (D)0x6F52
ERR)PCI Bus:Dev:Func       (R)03:00:00              (D)02:00:03
ERR)PCI 03:00:00 DeviceID  (R)0x15AB                (D)0x6F53
ERR)PCI 03:00:00 ClassCode (R)020000                (D)020000
```

```
ERR)PCI Bus:Dev:Func       (R)03:00:01               (D)03:00:00
ERR)PCI Bus:Dev:Func       (R)04:00:00               (D)03:00:01
ERR)PCI 04:00:00 VendorID  (R)0x10B5                 (D)0x8086
ERR)PCI 04:00:00 DeviceID  (R)0x8725                 (D)0x15AB
ERR)PCI 04:00:00 ClassCode (R)060400                 (D)060400
ERR)PCI 04:00:00 LinkStat  (R)Link is UP             (D)LinkStat not available
ERR)PCI 04:00:00 Wdth/Spd  (R)x8 / 8GT/s             (D)Bandwidth not available
ERR)PCI Bus:Dev:Func       (R)04:00:01               (D)04:00:00
ERR)PCI 04:00:01 DeviceID  (R)0x87D0                 (D)0x8725
ERR)PCI 04:00:01 ClassCode (R)088000                 (D)088000
ERR)PCI 04:00:01 LinkStat  (R)LinkStat not available (D)Link is UP
ERR)PCI 04:00:01 Wdth/Spd  (R)Bandwidth not available(D)x8 / 8GT/s
ERR)PCI Bus:Dev:Func       (R)04:00:02               (D)04:00:01
ERR)PCI Bus:Dev:Func       (R)04:00:03               (D)04:00:02
ERR)PCI Bus:Dev:Func       (R)04:00:04               (D)04:00:03
ERR)PCI Bus:Dev:Func       (R)05:00:00               (D)04:00:04
ERR)PCI 05:00:00 DeviceID  (R)0x8725                 (D)0x87D0
ERR)PCI 05:00:00 ClassCode (R)060400                 (D)060400
ERR)PCI 05:00:00 Wdth/Spd  (R)Undefined              (D)Bandwidth not available
ERR)PCI Bus:Dev:Func       (R)05:01:00               (D)05:00:00
ERR)PCI 05:01:00 LinkStat  (R)Link is UP             (D)Link not UP
ERR)PCI 05:01:00 Wdth/Spd  (R)x4 / 5GT/s             (D)Undefined
ERR)PCI Bus:Dev:Func       (R)05:08:00               (D)05:01:00
ERR)PCI 05:08:00 LinkStat  (R)Link not UP            (D)Link is UP
ERR)PCI 05:08:00 Wdth/Spd  (R)Undefined              (D)x4 / 5GT/s
ERR)PCI Bus:Dev:Func       (R)07:00:00               (D)05:08:00
ERR)PCI 07:00:00 VendorID  (R)0x111D                 (D)0x10B5
ERR)PCI 07:00:00 DeviceID  (R)0x808C                 (D)0x8725
ERR)PCI 07:00:00 LinkStat  (R)Link is UP             (D)Link not UP
ERR)PCI 07:00:00 Wdth/Spd  (R)x4 / 5GT/s             (D)Undefined
ERR)PCI Bus:Dev:Func       (R)08:02:00               (D)07:00:00
ERR)PCI 08:02:00 LinkStat  (R)Link not UP            (D)Link is UP
ERR)PCI 08:02:00 Wdth/Spd  (R)Undefined              (D)x4 / 5GT/s
ERR)PCI Bus:Dev:Func       (R)08:04:00               (D)08:02:00
ERR)PCI Bus:Dev:Func       (R)08:06:00               (D)08:04:00
ERR)PCI Bus:Dev:Func       (R)08:08:00               (D)08:06:00
ERR)PCI Bus:Dev:Func       (R)08:0C:00               (D)08:08:00
ERR)PCI Bus:Dev:Func       (R)08:10:00               (D)08:0C:00
ERR)PCI Bus:Dev:Func       (R)08:14:00               (D)08:10:00
ERR)PCI Bus:Dev:Func       (R)11:00:00               (D)08:14:00
ERR)PCI 11:00:00 VendorID  (R)0x8086                 (D)0x111D
ERR)PCI 11:00:00 DeviceID  (R)0x1533                 (D)0x808C
ERR)PCI 11:00:00 ClassCode (R)020000                 (D)020000
ERR)PCI 11:00:00 Wdth/Spd  (R)Bandwidth not available(D)Undefined
ERR)PCI Bus:Dev:Func       (R)12:00:00               (D)11:00:00
ERR)PCI Bus:Dev:Func       (R)No Device              (D)12:00:00
ERR)PCI 00:00:00 VendorID  (R)No Device              (D)0x8086
ERR)PCI 00:00:00 DeviceID  (R)No Device              (D)0x1533
ERR)PCI 00:00:00 ClassCode (R)No Device              (D)No Device
FAILED
```

(R) is for Recorded

(D) is for Detected

2.10.3 Editing the System Configuration

A menu is available to edit the system configuration. This menu allows to IGNORE specific items.

For example it is possible to ignore checking SATA disk size or USB keyboard connected ...

The menu also allows to configure a specific level of debug when the system test is executed. To enter **system edit** menu enter:

```
VX3058> kdiag  edit system
```

Equivalent to:

```
VX3058> kdiag  system_edit
```

The menu is displayed only if a system configuration has been previously recorded with the "**kdiag learn system**" or "**kdiag system_learn**" command. On the contrary a message informs that no configuration has been recorded yet:

```
SYSTEM INFOS NOT SAVED
```

Once edited the system edition gives you a prompt and is organized with sublevel menus. Help is available from any submenu by typing '**?**'.

Two mains features can be distinguished:

- the **Ignore** menu to bypass specific item verification
- the **print debug setting** menu to modify the debug level and manage the scroll limit for it.

2.10.3.1 Edit System Items (to Ignore Specific Test)

By default, all the items of the system test are checked. The system edit menu allows user to choose which items will be ignored during the next system test executions.

It is possible to ignore all or part of specific elements of the system test structures. For example an USB port, a SATA port, an Ethernet interface, SMBUS devices, information from the CPLD as the System Controller Status etc.

The following shows a learn/edit/run system session ignoring the System Controller status information from CPLD, the SATA port 4 information, all the SMBUS1 device addresses, all information for PCI device 28, the ClassCode information for PCI device 16, checksum BIOS information and the Ethernet interfaces 2 and 3:

```
VX3058> kdiag learn system
   Seg  Bus  Dev  Func
   ---  ---  ---  ----
    00   00   00    00 ==> Bridge Device - Host/PCI bridge
              Vendor 8086 Device 6F00 Prog Interface 0
    00   00   01    00 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 6F02 Prog Interface 0
    00   00   02    00 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 6F04 Prog Interface 0
    00   00   02    02 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 6F06 Prog Interface 0
    00   00   03    00 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 6F08 Prog Interface 0
    00   00   03    02 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 6F0A Prog Interface 0
    00   00   03    03 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 6F0B Prog Interface 0
    00   00   05    00 ==> Base System Peripherals - Other system peripheral
              Vendor 8086 Device 6F28 Prog Interface 0
```

```
 00   00   05    01 ==> Base System Peripherals - Other system peripheral
           Vendor 8086 Device 6F29 Prog Interface 0
 00   00   05    02 ==> Base System Peripherals - Other system peripheral
           Vendor 8086 Device 6F2A Prog Interface 0
 00   00   05    04 ==> Base System Peripherals - PIC
           Vendor 8086 Device 6F2C Prog Interface 20
 00   00   14    00 ==> Serial Bus Controllers - USB
           Vendor 8086 Device 8C31 Prog Interface 30
 00   00   1D    00 ==> Serial Bus Controllers - USB
           Vendor 8086 Device 8C26 Prog Interface 20
 00   00   1F    00 ==> Bridge Device - PCI/ISA bridge
           Vendor 8086 Device 8C54 Prog Interface 0
 00   00   1F    02 ==> Mass Storage Controller - Serial ATA controller
           Vendor 8086 Device 8C02 Prog Interface 1
 00   00   1F    03 ==> Serial Bus Controllers - System Management Bus
           Vendor 8086 Device 8C22 Prog Interface 0
 00   02   00    00 ==> Base System Peripherals - Other system peripheral
           Vendor 8086 Device 6F50 Prog Interface 0
 00   02   00    01 ==> Base System Peripherals - Other system peripheral
           Vendor 8086 Device 6F51 Prog Interface 0
 00   02   00    02 ==> Base System Peripherals - Other system peripheral
           Vendor 8086 Device 6F52 Prog Interface 0
 00   02   00    03 ==> Base System Peripherals - Other system peripheral
           Vendor 8086 Device 6F53 Prog Interface 0
 00   03   00    00 ==> Network Controller - Ethernet controller
           Vendor 8086 Device 15AB Prog Interface 0
 00   03   00    01 ==> Network Controller - Ethernet controller
           Vendor 8086 Device 15AB Prog Interface 0
 00   04   00    00 ==> Bridge Device - PCI/PCI bridge
           Vendor 10B5 Device 8725 Prog Interface 0
 00   04   00    01 ==> Base System Peripherals - Other system peripheral
           Vendor 10B5 Device 87D0 Prog Interface 0
 00   04   00    02 ==> Base System Peripherals - Other system peripheral
           Vendor 10B5 Device 87D0 Prog Interface 0
 00   04   00    03 ==> Base System Peripherals - Other system peripheral
           Vendor 10B5 Device 87D0 Prog Interface 0
 00   04   00    04 ==> Base System Peripherals - Other system peripheral
           Vendor 10B5 Device 87D0 Prog Interface 0
 00   05   00    00 ==> Bridge Device - PCI/PCI bridge
           Vendor 10B5 Device 8725 Prog Interface 0
 00   05   01    00 ==> Bridge Device - PCI/PCI bridge
           Vendor 10B5 Device 8725 Prog Interface 0
 00   05   08    00 ==> Bridge Device - PCI/PCI bridge
           Vendor 10B5 Device 8725 Prog Interface 0
 00   07   00    00 ==> Bridge Device - PCI/PCI bridge
           Vendor 111D Device 808C Prog Interface 0
 00   08   02    00 ==> Bridge Device - PCI/PCI bridge
           Vendor 111D Device 808C Prog Interface 0
 00   08   04    00 ==> Bridge Device - PCI/PCI bridge
           Vendor 111D Device 808C Prog Interface 0
 00   08   06    00 ==> Bridge Device - PCI/PCI bridge
           Vendor 111D Device 808C Prog Interface 0
 00   08   08    00 ==> Bridge Device - PCI/PCI bridge
           Vendor 111D Device 808C Prog Interface 0
00   08   0C    00 ==> Bridge Device - PCI/PCI bridge
           Vendor 111D Device 808C Prog Interface 0
 00   08   10    00 ==> Bridge Device - PCI/PCI bridge
           Vendor 111D Device 808C Prog Interface 0
 00   08   14    00 ==> Bridge Device - PCI/PCI bridge
           Vendor 111D Device 808C Prog Interface 0
 00   11   00    00 ==> Network Controller - Ethernet controller
           Vendor 8086 Device 1533 Prog Interface 0
```

```
     00   12   00    00 ==> Network Controller - Ethernet controller
             Vendor 8086 Device 1533 Prog Interface 0
CPLD Version 0x3
DRAM size 8 GB
System Controller
Geographical Address 1
XMC not present
SATA PORT 0 WDC WD2503ABYZ (251.0GB)
SATA PORT 1 WDC WD2502ABYS (251.0GB)
SATA PORT 4 TS32GMTS400    (32.0GB)
SATA PORT 5 TS32GMTS400    (32.0GB)
Device detected on SMBUS0, address = 0x30
BIOS Setup Checksum : 138322
BIOS Version : ID16133
ETH0 connected device speed 1000Mb/s
ETH2 connected device speed 1000Mb/s
ETH3 connected device speed 1000Mb/s
      4 Drives, 1 Hub
MassStorage Device name (0) : KingstonDataTravele
MassStorage Device name (1) : KingstonDataTravele
MassStorage Device name (2) : UFD 3.0 Silicon-Pow
MassStorage Device name (3) : KingstonDataTravele
USB Port 0 connected
USB Port 1 connected
USB Port 2 connected
USB Port 3 connected

Number of System Test Elements detected : 60
DRAM area [ 0x354F0BC0  0x354F1960 ] will be stored in EEPROM
Storing system infos...

Storing system configuration...
VX3058> kdiag edit system

           <<  KONTRON SYSTEM PBIT : EDIT MODE  >>

Edit by feature, choose: PCI CPLD SATA ETH USB SMBUS BIOS
Other available cmds: `s' for settings, `?' for help, `q' to quit edit mode

--SystemEdit>>cpld

--SystemEdit-CPLD>>?

p : print Recorded values
0,..,4 : edit CPLD element
ignoreall : ignore all CPLD infos
default : restore default CPLD config (remove flags)
? : help
q : go back to main menu

--SystemEdit-CPLD>>p


Recorded CPLD config:

0. CPLD Version           0x3
1. CPLD Dram Size         8GB
2. CPLD SysCon            System Controller
3. CPLD Geo Address       1
4. CPLD XMC Presence      XMC not present
--SystemEdit-CPLD>>2
```

```
--SystemEdit-CPLD-SysCon>>?

p : print Recorded value
i : set ignore flag
r : remove flag
? : help
q : go back to CPLD menu

--SystemEdit-CPLD-SysCon>>i

Ignore flag has been set
*-SystemEdit-CPLD-SysCon>>q

*-SystemEdit-CPLD>>p


Recorded CPLD config:

0. CPLD Version           0x3
1. CPLD Dram Size         8GB
2. CPLD SysCon            System Controller      (i)
3. CPLD Geo Address       1
4. CPLD XMC Presence      XMC not present
*-SystemEdit-CPLD>>

           <<  KONTRON SYSTEM PBIT : EDIT MODE  >>

Edit by feature, choose: PCI CPLD (*) SATA ETH USB SMBUS BIOS
Other available cmds: `s´ for settings, `?´ for help, `q´ to quit edit mode

*-SystemEdit>>sata

*-SystemEdit-SATA>>p

0. SATA Port 0            WDC WD2503ABYZ (251.0GB
1. SATA Port 1            WDC WD2502ABYS (251.0GB
2. SATA Port 2            Not Connected (0.0GB)
3. SATA Port 3            Not Connected (0.0GB)
4. SATA Port 4            TS32GMTS400    (32.0GB)
5. SATA Port 5            TS32GMTS400    (32.0GB)
*-SystemEdit-SATA>>4

*-SystemEdit-SATA-Port 4>>i

Ignore flag has been set
*-SystemEdit-SATA-Port 4>>

*-SystemEdit-SATA>>p

0. SATA Port 0            WDC WD2503ABYZ (251.0GB
1. SATA Port 1            WDC WD2502ABYS (251.0GB
2. SATA Port 2            Not Connected (0.0GB)
3. SATA Port 3            Not Connected (0.0GB)
4. SATA Port 4            TS32GMTS400    (32.0GB)(i)
5. SATA Port 5            TS32GMTS400    (32.0GB)
*-SystemEdit-SATA>>
```

```
           <<  KONTRON SYSTEM PBIT : EDIT MODE  >>

Edit by feature, choose: PCI CPLD (*) SATA (*) ETH USB SMBUS BIOS
Other available cmds: `s´ for settings, `?´ for help, `q´ to quit edit mode

*-SystemEdit>>smbus

*-SystemEdit-SMBUS>>?

p : print Recorded values
0,..,1 : edit SMBUS element
ignoreall : ignore all SMBUS infos
default : restore default SMBUS config (remove flags)
? : help
q : go back to main menu

*-SystemEdit-SMBUS>>p

0. SMBUS0
SMBUS0 Device 0        0x30
SMBUS0 Device 1        No Device
SMBUS0 Device 2        No Device
SMBUS0 Device 3        No Device
SMBUS0 Device 4        No Device
SMBUS0 Device 5        No Device
SMBUS0 Device 6        No Device
SMBUS0 Device 7        No Device
SMBUS0 Device 8        No Device
SMBUS0 Device 9        No Device
1. SMBUS1
SMBUS1 Device 0        No Device
SMBUS1 Device 1        No Device
SMBUS1 Device 2        No Device
SMBUS1 Device 3        No Device
SMBUS1 Device 4        No Device
SMBUS1 Device 5        No Device
SMBUS1 Device 6        No Device
SMBUS1 Device 7        No Device
SMBUS1 Device 8        No Device
SMBUS1 Device 9        No Device
*-SystemEdit-SMBUS>>1

*-SystemEdit-SMBUS-SMBUS1>>?

p : print Recorded values
0,..,9 : edit SMBUS1 element
ignoreall : ignore all SMBUS1 infos
default : restore default SMBUS1 config (remove flags)
? : help
q : go back to SMBUS menu

*-SystemEdit-SMBUS-SMBUS1>>ignoreall

SMBUS1 entirely ignored
*-SystemEdit-SMBUS-SMBUS1>>

*-SystemEdit-SMBUS>>p

0. SMBUS0
SMBUS0 Device 0        0x30
```

```
SMBUS0 Device 1        No Device
SMBUS0 Device 2        No Device
SMBUS0 Device 3        No Device
SMBUS0 Device 4        No Device
SMBUS0 Device 5        No Device
SMBUS0 Device 6        No Device
SMBUS0 Device 7        No Device
SMBUS0 Device 8        No Device
SMBUS0 Device 9        No Device
1. SMBUS1 (ignored)
SMBUS1 Device 0        No Device             (i)
SMBUS1 Device 1        No Device             (i)
SMBUS1 Device 2        No Device             (i)
SMBUS1 Device 3        No Device             (i)
SMBUS1 Device 4        No Device             (i)
SMBUS1 Device 5        No Device             (i)
SMBUS1 Device 6        No Device             (i)
SMBUS1 Device 7        No Device             (i)
SMBUS1 Device 8        No Device             (i)
SMBUS1 Device 9        No Device             (i)
*-SystemEdit-SMBUS>>

           <<  KONTRON SYSTEM PBIT : EDIT MODE  >>

Edit by feature, choose: PCI CPLD (*) SATA (*) ETH USB SMBUS (*) BIOS
Other available cmds: `s´ for settings, `?´ for help, `q´ to quit edit mode

--SystemEdit>>pci

--SystemEdit-PCI>>?

pa : print all Recorded PCI Device infos
pb : print all Recorded PCI/PCI Bridge infos
p1 : print Recorded VendorID only
p2 : print Recorded DeviceID only
p3 : print Recorded ClassCode only
p4 : print Recorded Linkstatus only (only for PCI/PCI bridge)
p5 : print Recorded Link Speed/Width only (only for PCI/PCI bridge)
p : print Recorded values
0,..,49 : edit PCI element
ignoreall : ignore all PCI infos
default : restore default PCI config (remove flags)
? : help
q : go back to main menu

--SystemEdit-PCI>>p

0.  PCI 00:00:00 Bridge Device/Host/PCI bridge
1.  PCI 00:01:00 Bridge Device/PCI/PCI bridge
2.  PCI 00:02:00 Bridge Device/PCI/PCI bridge
3.  PCI 00:02:02 Bridge Device/PCI/PCI bridge
4.  PCI 00:03:00 Bridge Device/PCI/PCI bridge
5.  PCI 00:03:02 Bridge Device/PCI/PCI bridge
6.  PCI 00:03:03 Bridge Device/PCI/PCI bridge
7.  PCI 00:05:00 Base System Peripherals/Other system peripheral
8.  PCI 00:05:01 Base System Peripherals/Other system peripheral
9.  PCI 00:05:02 Base System Peripherals/Other system peripheral
10. PCI 00:05:04 Base System Peripherals/PIC
11. PCI 00:14:00 Serial Bus Controllers/USB
12. PCI 00:1D:00 Serial Bus Controllers/USB
13. PCI 00:1F:00 Bridge Device/PCI/ISA bridge
```

```
14. PCI 00:1F:02 Mass Storage Controller/Serial ATA controller
15. PCI 00:1F:03 Serial Bus Controllers/System Management Bus
16. PCI 02:00:00 Base System Peripherals/Other system peripheral
17. PCI 02:00:01 Base System Peripherals/Other system peripheral
18. PCI 02:00:02 Base System Peripherals/Other system peripheral
19. PCI 02:00:03 Base System Peripherals/Other system peripheral
20. PCI 03:00:00 Network Controller/Ethernet controller
21. PCI 03:00:01 Network Controller/Ethernet controller
22. PCI 04:00:00 Bridge Device/PCI/PCI bridge
23. PCI 04:00:01 Base System Peripherals/Other system peripheral
24. PCI 04:00:02 Base System Peripherals/Other system peripheral
25. PCI 04:00:03 Base System Peripherals/Other system peripheral
26. PCI 04:00:04 Base System Peripherals/Other system peripheral
27. PCI 05:00:00 Bridge Device/PCI/PCI bridge
28. PCI 05:01:00 Bridge Device/PCI/PCI bridge
29. PCI 05:08:00 Bridge Device/PCI/PCI bridge
30. PCI 07:00:00 Bridge Device/PCI/PCI bridge
31. PCI 08:02:00 Bridge Device/PCI/PCI bridge
32. PCI 08:04:00 Bridge Device/PCI/PCI bridge
33. PCI 08:06:00 Bridge Device/PCI/PCI bridge
34. PCI 08:08:00 Bridge Device/PCI/PCI bridge
35. PCI 08:0C:00 Bridge Device/PCI/PCI bridge
36. PCI 08:10:00 Bridge Device/PCI/PCI bridge
37. PCI 08:14:00 Bridge Device/PCI/PCI bridge
38. PCI 11:00:00 Network Controller/Ethernet controller
39. PCI 12:00:00 Network Controller/Ethernet controller
40. PCI No Device No Device/No Device
41. PCI No Device No Device/No Device
42. PCI No Device No Device/No Device
43. PCI No Device No Device/No Device
44. PCI No Device No Device/No Device
45. PCI No Device No Device/No Device
46. PCI No Device No Device/No Device
47. PCI No Device No Device/No Device
48. PCI No Device No Device/No Device
49. PCI No Device No Device/No Device
--SystemEdit-PCI>> 28

--SystemEdit-PCI-05:01:00>>ignoreall

PCI 05:01:00 entirely ignored
*-SystemEdit-PCI-05:01:00>>p

0. VendorId               0x10B5 (i)
1. DeviceId               0x8725 (i)
2. ClassCode              060400 (i)
   Prog Interface         0
   BaseClass              Bridge Device
   SubClass               PCI/PCI bridge
3. LinkStatus             Link is UP (i)
4. Link Speed/Width       x4 / 5GT/s (i)
*-SystemEdit-PCI-05:01:00>>

*-SystemEdit-PCI>>

           <<  KONTRON SYSTEM PBIT : EDIT MODE  >>

Edit by feature, choose: PCI (*) CPLD (*) SATA (*) ETH (*) USB SMBUS (*) BIOS (*)
Other available cmds: `s´ for settings, `?´ for help, `q´ to quit edit mode

--SystemEdit>>pci
```

```
--SystemEdit-PCI>>16

--SystemEdit-PCI-02:00:00>>?

p : print Recorded values
0,..,4 : edit PCI 02:00:00 element
ignoreall : ignore all PCI 02:00:00 infos
default : restore default PCI 02:00:00 config (remove flags)
? : help
q : go back to main menu

--SystemEdit-PCI-02:00:00>>2

--SystemEdit-PCI-02:00:00-ClassCode>>?

p : print Recorded value
i : set ignore flag
r : remove flag
? : help
q : go back to PCI 02:00:00 menu

--SystemEdit-PCI-02:00:00-ClassCode>>i

Ignore flag has been set
*-SystemEdit-PCI-02:00:00-ClassCode>>

*-SystemEdit-PCI-02:00:00>>p

0. VendorId               0x8086
1. DeviceId               0x6F50
2. ClassCode              088000 (i)
   Prog Interface         0
   BaseClass              Base System Peripherals
   SubClass               Other system peripheral
*-SystemEdit-PCI-02:00:00>>

*-SystemEdit-PCI>>

           <<  KONTRON SYSTEM PBIT : EDIT MODE  >>

Edit by feature, choose: PCI (*) CPLD (*) SATA (*) ETH USB SMBUS (*) BIOS
Other available cmds: `s´ for settings, `?´ for help, `q´ to quit edit mode

*-SystemEdit>>bios

*-SystemEdit-BIOS>>?

p : print Recorded values
0,..,1 : edit BIOS element
ignoreall : ignore all BIOS infos
default : restore default BIOS config (remove flags)
? : help
q : go back to main menu

*-SystemEdit-BIOS>>p

0. BIOS Checksum          138322
1. BIOS Version           ID16133
```

```
*-SystemEdit-BIOS>>0

*-SystemEdit-BIOS-Checksum>>i

Ignore flag has been set
*-SystemEdit-BIOS-Checksum>>p

138322                  (i)
*-SystemEdit-BIOS-Checksum>>

*-SystemEdit-BIOS>>p

0. BIOS Checksum          138322                 (i)
1. BIOS Version           ID16133
*-SystemEdit-BIOS>>

           <<  KONTRON SYSTEM PBIT : EDIT MODE  >>

Edit by feature, choose: PCI (*) CPLD (*) SATA (*) ETH USB SMBUS (*) BIOS (*)
Other available cmds: `s´ for settings, `?´ for help, `q´ to quit edit mode

*-SystemEdit>>eth

*-SystemEdit-ETH>>p

0. ETH Port 0             Link UP speed 1000 Mb/s
1. ETH Port 1             Not Connected
2. ETH Port 2             Link UP speed 1000 Mb/s
3. ETH Port 3             Link UP speed 1000 Mb/s
*-SystemEdit-ETH>>2

*-SystemEdit-ETH-Port 2>>i

Ignore flag has been set
*-SystemEdit-ETH-Port 2>>

*-SystemEdit-ETH>>p

0. ETH Port 0             Link UP speed 1000 Mb/s
1. ETH Port 1             Not Connected
2. ETH Port 2             Link UP speed 1000 Mb/s(i)
3. ETH Port 3             Link UP speed 1000 Mb/s
*-SystemEdit-ETH>>3

*-SystemEdit-ETH-Port 3>>i

Ignore flag has been set
*-SystemEdit-ETH-Port 3>>

*-SystemEdit-ETH>>p

0. ETH Port 0             Link UP speed 1000 Mb/s
1. ETH Port 1             Not Connected
2. ETH Port 2             Link UP speed 1000 Mb/s(i)
3. ETH Port 3             Link UP speed 1000 Mb/s(i)
*-SystemEdit-ETH>>
```

```
          <<  KONTRON SYSTEM PBIT : EDIT MODE  >>

Edit by feature, choose: PCI (*) CPLD (*) SATA (*) ETH (*) USB SMBUS (*) BIOS (*)
Other available cmds: `s´ for settings, `?´ for help, `q´ to quit edit mode

*-SystemEdit>>q
Quit Edit menu

Modifications have been made.
Do you want to save your configuration before leaving? (y/n)
*-SystemEdit-Save>>y
VX3058> kdiag run system
PBIT "system" (fast,simple,promptonfail)
PCI (*) CPLD (*) SATA (*) SMBUS (*) BIOS (*) ETH (*) USB...
PASSED
```

System stats are displayed when we print recorded values with command '**p**' (**[LAST FAILED]** and/or **[FAILED ONCE]** labels).

2.10.3.2 Print Debug Settings

By default, the system test prints errors only, and detail level is normal. It is possible to modify these general parameters in this menu.

- **Three print levels are available:**
  - Errors only,
  - Errors and Infos only,
  - Debug lvl (print all).

  Each printed message in system test corresponds to one of these levels and is easily identifiable thanks to 3 labels:
  - Error: **ERR)**
  - Info: **INF)**
  - Debug: **DBG)**

- **Scroll limit for `Debug lvl` mode ONLY:**

  If Debug mode is selected, a scroll limit can be set in order to stop scrolling during system test, depending on the chosen limit: 0 is no limit (do not stop scrolling by default), max for scroll limit is 30 lines.

  WARNING: WATCH- DOG WILL BE STOPPED during "system" test execution IF SCROLL LIMIT IS NOT 0. Then, at the next system run, use space (scroll to the limit) or Enter (scroll line by line) to make test results scroll.

- **Three detail levels are available (for experts because not very useful for user): synthetic mode, normal mode and detailed mode.**
  - Synthetic mode does not display errors and information results but gives the number of errors found by type and the total.

    Example with SYNTHETIC MODE:

    ```
    VX3058> kdiag run system
    PBIT "system" (fast,simple)
    PCI (*) CPLD (*) SATA (*) SMBUS (i) BIOS (*) ETH (*) USB...
    ERR)PCI Results : 51 ERRORS
    ERR)USB Results : 3 ERRORS
    ERR)TOTAL : 54 ERRORS
    FAILED
    ```

  - Normal mode displays system test results depending on the print level. The classcode value will be printed for PCI devices.

- Detailed mode allows to print more details for PCI devices:

  It prints classcode value but also prog interface nb, BaseClass string, SubClass string which are deducted from classcode value.

- Clear System Stats: used to clear system stats (see section 2.10.4 page 38).

The following shows a system_edit session setting print level to Debug Level, a scroll limit of 10 lines and detail level set to Detailed mode:

```
VX3058> kdiag  system_edit

           <<  KONTRON SYSTEM PBIT : EDIT MODE  >>

Edit by feature, choose: PCI (*) CPLD (*) SATA (*) ETH (*) USB SMBUS (*) BIOS (*)
Other available cmds: `s´ for settings, `?´ for help, `q´ to quit edit mode

--SystemEdit>>s

p : Set Print Level
Current Print Lvl is: ERRORS ONLY
d : Set Detail Level
Current Details Lvl is: NORMAL
c : Clear System Stats
--SystemEdit-generalSettings>>p

--SystemEdit-generalSettings-printLvl>>?

p : print current print Level
e : ERRORS ONLY mode
i : ERRORS & INFOS ONLY mode
d : DEBUG LVL mode (print all)
q : return to general settings menu
--SystemEdit-generalSettings-printLvl>>d

Print Lvl DEBUG PRINT ALL set, scroll limit is 24 by default
Set Scroll limit (0 = no limit , max = 30)
*-SystemEdit-generalSettings-printLvl-Scroll>>10

Scroll Limit set to 10 line(s)
*-SystemEdit-generalSettings-printLvl>>q

p : Set Print Level
Current Print Lvl is: DEBUG PRINT ALL
d : Set Detail Level
Current Details Lvl is: NORMAL
c : Clear System Stats
*-SystemEdit-generalSettings>>d

*-SystemEdit-generalSettings-detailsLvl>>?

p : print Current Details Lvl
s : synthethic results
n : normal infos
d : detailed
q : return to general settings menu
*-SystemEdit-generalSettings-detailsLvl>>p
```

```
Current Details Lvl is: NORMAL
*-SystemEdit-generalSettings-detailsLvl>>d

Details Lvl DETAILED set
*-SystemEdit-generalSettings-detailsLvl>>p

Current Details Lvl is: DETAILED
*-SystemEdit-generalSettings-detailsLvl>>q

p : Set Print Level
Current Print Lvl is: DEBUG PRINT ALL
d : Set Detail Level
Current Details Lvl is: DETAILED
c : Clear System Stats
*-SystemEdit-generalSettings>>c  q

           <<  KONTRON SYSTEM PBIT : EDIT MODE  >>

Edit by feature, choose: PCI (*) CPLD (*) SATA (*) ETH (*) USB SMBUS (*) BIOS (*)
Other available cmds: `s´ for settings, `?´ for help, `q´ to quit edit mode

*-SystemEdit>>q
Quit Edit menu

Modifications have been made.
Do you want to save your configuration before leaving? (y/n)
*-SystemEdit-Save>>y
```

Use important commands '**?**' and '**p**' to get help and to see recorded values respectively. Note that return (just Enter) is equivalent to 'q' (go back to previous menu). Ignored elements are marked with the **(i)** label. Stats are shown through **[LAST FAILED]** and/or **[FAILED ONCE]** labels.

If the system configuration is edited again, then the excluded information will be marked **IGNORED** and will not participate to the test. If **IGNORED** items changed after the **system_learn** phase, the system test will not fail because the tests corresponding to the items are bypassed.

Now, it is possible to remove **IGNORED** flags one by one (browse menu and type 'r' to remove ignore flag of an element) or remove all flags at once of an object or type (to be placed in the right place in the menu and type 'default' for a group of elements) in order to test items again.

2.10.4 Clearing System Stats

To clear **[LAST FAILED]** and **[FAILED ONCE]** labels of system test items, enter the **kdiag system menu** section general settings and use the following command:

```
VX3058> kdiag system_edit

           <<  KONTRON SYSTEM PBIT : EDIT MODE  >>

Edit by feature, choose: PCI (*) CPLD (*) SATA (*) ETH (*) USB SMBUS (*) BIOS (*)
Other available cmds: `s´ for settings, `?´ for help, `q´ to quit edit mode

--SystemEdit>>s

p : Set Print Level
Current Print Lvl is: DEBUG PRINT ALL
d : Set Detail Level
Current Details Lvl is: DETAILED
c : Clear System Stats
--SystemEdit-generalSettings>>c
```

```
System stats have been cleared

p : Set Print Level
Current Print Lvl is: DEBUG PRINT ALL
d : Set Detail Level
Current Details Lvl is: DETAILED
c : Clear System Stats
--SystemEdit-generalSettings>>

           <<  KONTRON SYSTEM PBIT : EDIT MODE  >>

Edit by feature, choose: PCI (*) CPLD (*) SATA (*) ETH (*) USB SMBUS (*) BIOS (*)
Other available cmds: `s´ for settings, `?´ for help, `q´ to quit edit mode

--SystemEdit>>q
Quit Edit menu

No modifications, quit without saving..
VX3058> kdiag run system
PBIT "system" (fast,simple,promptonfail)
Size of one STES = 0x2
Total size = 0xD
Size of global control = 0xC

PCI (*) CPLD (*) SATA (*) SMBUS (*) BIOS (*) ETH (*) USB...
CRC Recorded 0x9 different from CRC Detected 0x9
==>System Configuration area changed

DBG)PCI Bus:Dev:Func       (R)00:00:00              (D)00:00:00
DBG)PCI 00:00:00 VendorID  (R)0x8086                (D)0x8086
DBG)PCI 00:00:00 DeviceID  (R)0x6F00                (D)0x6F00
DBG)PCI 00:00:00 ClassCode (R)060000                (D)060000
ProgInterface          (R)0                      (D)0
BaseClass              (R)Bridge Device          (D)Bridge Device
SubClass               (R)Host/PCI bridge        (D)Host/PCI bridge
DBG)PCI Bus:Dev:Func       (R)00:01:00              (D)00:01:00
DBG)PCI 00:01:00 VendorID  (R)0x8086                (D)0x8086
DBG)PCI 00:01:00 DeviceID  (R)0x6F02                (D)0x6F02
DBG)PCI 00:01:00 ClassCode (R)060400                (D)060400
ProgInterface          (R)0                      (D)0
BaseClass              (R)Bridge Device          (D)Bridge Device
SubClass               (R)PCI/PCI bridge         (D)PCI/PCI bridge
DBG)PCI 00:01:00 LinkStat  (R)Link not UP           (D)Link not UP
DBG)PCI 00:01:00 Wdth/Spd  (R)Undefined             (D)Undefined
DBG)PCI Bus:Dev:Func       (R)00:02:00              (D)00:02:00
DBG)PCI 00:02:00 VendorID  (R)0x8086                (D)0x8086
DBG)PCI 00:02:00 DeviceID  (R)0x6F04                (D)0x6F04
DBG)PCI 00:02:00 ClassCode (R)060400                (D)060400
ProgInterface          (R)0                      (D)0
BaseClass              (R)Bridge Device          (D)Bridge Device
SubClass               (R)PCI/PCI bridge         (D)PCI/PCI bridge
DBG)PCI 00:02:00 LinkStat  (R)Link is UP            (D)Link is UP
DBG)PCI 00:02:00 Wdth/Spd  (R)x1 / 2.5GT/s          (D)x1 / 2.5GT/s
DBG)PCI Bus:Dev:Func       (R)00:02:02              (D)00:02:02
DBG)PCI 00:02:02 VendorID  (R)0x8086                (D)0x8086
DBG)PCI 00:02:02 DeviceID  (R)0x6F06                (D)0x6F06
DBG)PCI 00:02:02 ClassCode (R)060400                (D)060400
ProgInterface          (R)0                      (D)0
BaseClass              (R)Bridge Device          (D)Bridge Device
SubClass               (R)PCI/PCI bridge         (D)PCI/PCI bridge
DBG)PCI 00:02:02 LinkStat  (R)Link is UP            (D)Link is UP
DBG)PCI 00:02:02 Wdth/Spd  (R)x1 / 2.5GT/s          (D)x1 / 2.5GT/s
```

```
DBG)PCI Bus:Dev:Func       (R)00:03:00              (D)00:03:00
DBG)PCI 00:03:00 VendorID  (R)0x8086                (D)0x8086
DBG)PCI 00:03:00 DeviceID  (R)0x6F08                (D)0x6F08
DBG)PCI 00:03:00 ClassCode (R)060400                (D)060400
ProgInterface          (R)0                      (D)0
BaseClass              (R)Bridge Device          (D)Bridge Device
SubClass               (R)PCI/PCI bridge         (D)PCI/PCI bridge
DBG)PCI 00:03:00 LinkStat  (R)Link is UP            (D)Link is UP
DBG)PCI 00:03:00 Wdth/Spd  (R)x8 / 8GT/s            (D)x8 / 8GT/s
DBG)PCI Bus:Dev:Func       (R)00:03:02              (D)00:03:02
DBG)PCI 00:03:02 VendorID  (R)0x8086                (D)0x8086
DBG)PCI 00:03:02 DeviceID  (R)0x6F0A                (D)0x6F0A
DBG)PCI 00:03:02 ClassCode (R)060400                (D)060400
ProgInterface          (R)0                      (D)0
BaseClass              (R)Bridge Device          (D)Bridge Device
SubClass               (R)PCI/PCI bridge         (D)PCI/PCI bridge
DBG)PCI 00:03:02 LinkStat  (R)Link is UP            (D)Link is UP
DBG)PCI 00:03:02 Wdth/Spd  (R)x1 / 2.5GT/s          (D)x1 / 2.5GT/s
DBG)PCI Bus:Dev:Func       (R)00:03:03              (D)00:03:03
DBG)PCI 00:03:03 VendorID  (R)0x8086                (D)0x8086
DBG)PCI 00:03:03 DeviceID  (R)0x6F0B                (D)0x6F0B
DBG)PCI 00:03:03 ClassCode (R)060400                (D)060400
ProgInterface          (R)0                      (D)0
BaseClass              (R)Bridge Device          (D)Bridge Device
SubClass               (R)PCI/PCI bridge         (D)PCI/PCI bridge
DBG)PCI 00:03:03 LinkStat  (R)Link is UP            (D)Link is UP
DBG)PCI 00:03:03 Wdth/Spd  (R)x1 / 2.5GT/s          (D)x1 / 2.5GT/s
DBG)PCI Bus:Dev:Func       (R)00:05:00              (D)00:05:00
DBG)PCI 00:05:00 VendorID  (R)0x8086                (D)0x8086
DBG)PCI 00:05:00 DeviceID  (R)0x6F28                (D)0x6F28
DBG)PCI 00:05:00 ClassCode (R)088000                (D)088000
ProgInterface          (R)0                      (D)0
BaseClass              (R)Base System Peripherals(D)Base System Peripherals
SubClass               (R)Other system peripheral(D)Other system peripheral
DBG)PCI Bus:Dev:Func       (R)00:05:01              (D)00:05:01
DBG)PCI 00:05:01 VendorID  (R)0x8086                (D)0x8086
DBG)PCI 00:05:01 DeviceID  (R)0x6F29                (D)0x6F29
DBG)PCI 00:05:01 ClassCode (R)088000                (D)088000
ProgInterface          (R)0                      (D)0
BaseClass              (R)Base System Peripherals(D)Base System Peripherals
SubClass               (R)Other system peripheral(D)Other system peripheral
DBG)PCI Bus:Dev:Func       (R)00:05:02              (D)00:05:02
DBG)PCI 00:05:02 VendorID  (R)0x8086                (D)0x8086
DBG)PCI 00:05:02 DeviceID  (R)0x6F2A                (D)0x6F2A
DBG)PCI 00:05:02 ClassCode (R)088000                (D)088000
ProgInterface          (R)0                      (D)0
BaseClass              (R)Base System Peripherals(D)Base System Peripherals
SubClass               (R)Other system peripheral(D)Other system peripheral
DBG)PCI Bus:Dev:Func       (R)00:05:04              (D)00:05:04
DBG)PCI 00:05:04 VendorID  (R)0x8086                (D)0x8086
DBG)PCI 00:05:04 DeviceID  (R)0x6F2C                (D)0x6F2C
DBG)PCI 00:05:04 ClassCode (R)080020                (D)080020
ProgInterface          (R)20                     (D)20
BaseClass              (R)Base System Peripherals(D)Base System Peripherals
SubClass               (R)PIC                    (D)PIC
DBG)PCI Bus:Dev:Func       (R)00:14:00              (D)00:14:00
DBG)PCI 00:14:00 VendorID  (R)0x8086                (D)0x8086
DBG)PCI 00:14:00 DeviceID  (R)0x8C31                (D)0x8C31
DBG)PCI 00:14:00 ClassCode (R)0C0330                (D)0C0330
ProgInterface          (R)30                     (D)30
```

```
BaseClass              (R)Serial Bus Controllers (D)Serial Bus Controllers
SubClass               (R)USB                    (D)USB
DBG)PCI Bus:Dev:Func       (R)00:1D:00               (D)00:1D:00
DBG)PCI 00:1D:00 VendorID  (R)0x8086                 (D)0x8086
DBG)PCI 00:1D:00 DeviceID  (R)0x8C26                 (D)0x8C26
DBG)PCI 00:1D:00 ClassCode (R)0C0320                 (D)0C0320
ProgInterface          (R)20                     (D)20
BaseClass              (R)Serial Bus Controllers (D)Serial Bus Controllers
SubClass               (R)USB                    (D)USB
DBG)PCI Bus:Dev:Func       (R)00:1F:00               (D)00:1F:00
DBG)PCI 00:1F:00 VendorID  (R)0x8086                 (D)0x8086
DBG)PCI 00:1F:00 DeviceID  (R)0x8C54                 (D)0x8C54
DBG)PCI 00:1F:00 ClassCode (R)060100                 (D)060100
ProgInterface          (R)0                      (D)0
BaseClass              (R)Bridge Device          (D)Bridge Device
SubClass               (R)PCI/ISA bridge         (D)PCI/ISA bridge
DBG)PCI Bus:Dev:Func       (R)00:1F:02               (D)00:1F:02
DBG)PCI 00:1F:02 VendorID  (R)0x8086                 (D)0x8086
DBG)PCI 00:1F:02 DeviceID  (R)0x8C02                 (D)0x8C02
DBG)PCI 00:1F:02 ClassCode (R)010601                 (D)010601
ProgInterface          (R)1                      (D)1
BaseClass              (R)Mass Storage Controller(D)Mass Storage Controller
SubClass               (R)Serial ATA controller  (D)Serial ATA controller
DBG)PCI Bus:Dev:Func       (R)00:1F:03               (D)00:1F:03
DBG)PCI 00:1F:03 VendorID  (R)0x8086                 (D)0x8086
DBG)PCI 00:1F:03 DeviceID  (R)0x8C22                 (D)0x8C22
DBG)PCI 00:1F:03 ClassCode (R)0C0500                 (D)0C0500
ProgInterface          (R)0                      (D)0
BaseClass              (R)Serial Bus Controllers (D)Serial Bus Controllers
SubClass               (R)System Management Bus  (D)System Management Bus
DBG)PCI Bus:Dev:Func       (R)02:00:00               (D)02:00:00
DBG)PCI 02:00:00 VendorID  (R)0x8086                 (D)0x8086
DBG)PCI 02:00:00 DeviceID  (R)0x6F50                 (D)0x6F50
INF)PCI 02:00:00 ClassCode ignored
DBG)PCI Bus:Dev:Func       (R)02:00:01               (D)02:00:01
DBG)PCI 02:00:01 VendorID  (R)0x8086                 (D)0x8086
DBG)PCI 02:00:01 DeviceID  (R)0x6F51                 (D)0x6F51
DBG)PCI 02:00:01 ClassCode (R)088000                 (D)088000
ProgInterface          (R)0                      (D)0
BaseClass              (R)Base System Peripherals(D)Base System Peripherals
SubClass               (R)Other system peripheral(D)Other system peripheral
DBG)PCI Bus:Dev:Func       (R)02:00:02               (D)02:00:02
DBG)PCI 02:00:02 VendorID  (R)0x8086                 (D)0x8086
DBG)PCI 02:00:02 DeviceID  (R)0x6F52                 (D)0x6F52
DBG)PCI 02:00:02 ClassCode (R)088000                 (D)088000
ProgInterface          (R)0                      (D)0
BaseClass              (R)Base System Peripherals(D)Base System Peripherals
SubClass               (R)Other system peripheral(D)Other system peripheral
DBG)PCI Bus:Dev:Func       (R)02:00:03               (D)02:00:03
DBG)PCI 02:00:03 VendorID  (R)0x8086                 (D)0x8086
DBG)PCI 02:00:03 DeviceID  (R)0x6F53                 (D)0x6F53
DBG)PCI 02:00:03 ClassCode (R)088000                 (D)088000
ProgInterface          (R)0                      (D)0
BaseClass              (R)Base System Peripherals(D)Base System Peripherals
SubClass               (R)Other system peripheral(D)Other system peripheral
DBG)PCI Bus:Dev:Func       (R)03:00:00               (D)03:00:00
DBG)PCI 03:00:00 VendorID  (R)0x8086                 (D)0x8086
DBG)PCI 03:00:00 DeviceID  (R)0x15AB                 (D)0x15AB
DBG)PCI 03:00:00 ClassCode (R)020000                 (D)020000
```

```
ProgInterface          (R)0                     (D)0
BaseClass              (R)Network Controller    (D)Network Controller
SubClass               (R)Ethernet controller   (D)Ethernet controller
DBG)PCI Bus:Dev:Func       (R)03:00:01              (D)03:00:01
DBG)PCI 03:00:01 VendorID  (R)0x8086                (D)0x8086
DBG)PCI 03:00:01 DeviceID  (R)0x15AB                (D)0x15AB
DBG)PCI 03:00:01 ClassCode (R)020000                (D)020000
ProgInterface          (R)0                     (D)0
BaseClass              (R)Network Controller    (D)Network Controller
SubClass               (R)Ethernet controller   (D)Ethernet controller
DBG)PCI Bus:Dev:Func       (R)04:00:00              (D)04:00:00
DBG)PCI 04:00:00 VendorID  (R)0x10B5                (D)0x10B5
DBG)PCI 04:00:00 DeviceID  (R)0x8725                (D)0x8725
DBG)PCI 04:00:00 ClassCode (R)060400                (D)060400
ProgInterface          (R)0                     (D)0
BaseClass              (R)Bridge Device         (D)Bridge Device
SubClass               (R)PCI/PCI bridge        (D)PCI/PCI bridge
DBG)PCI 04:00:00 LinkStat  (R)Link is UP            (D)Link is UP
DBG)PCI 04:00:00 Wdth/Spd  (R)x8 / 8GT/s            (D)x8 / 8GT/s
DBG)PCI Bus:Dev:Func       (R)04:00:01              (D)04:00:01
DBG)PCI 04:00:01 VendorID  (R)0x10B5                (D)0x10B5
DBG)PCI 04:00:01 DeviceID  (R)0x87D0                (D)0x87D0
DBG)PCI 04:00:01 ClassCode (R)088000                (D)088000
ProgInterface          (R)0                     (D)0
BaseClass              (R)Base System Peripherals(D)Base System Peripherals
SubClass               (R)Other system peripheral(D)Other system peripheral
DBG)PCI Bus:Dev:Func       (R)04:00:02              (D)04:00:02
DBG)PCI 04:00:02 VendorID  (R)0x10B5                (D)0x10B5
DBG)PCI 04:00:02 DeviceID  (R)0x87D0                (D)0x87D0
DBG)PCI 04:00:02 ClassCode (R)088000                (D)088000
ProgInterface          (R)0                     (D)0
BaseClass              (R)Base System Peripherals(D)Base System Peripherals
SubClass               (R)Other system peripheral(D)Other system peripheral
DBG)PCI Bus:Dev:Func       (R)04:00:03              (D)04:00:03
DBG)PCI 04:00:03 VendorID  (R)0x10B5                (D)0x10B5
DBG)PCI 04:00:03 DeviceID  (R)0x87D0                (D)0x87D0
DBG)PCI 04:00:03 ClassCode (R)088000                (D)088000
ProgInterface          (R)0                     (D)0
BaseClass              (R)Base System Peripherals(D)Base System Peripherals
SubClass               (R)Other system peripheral(D)Other system peripheral
DBG)PCI Bus:Dev:Func       (R)04:00:04              (D)04:00:04
DBG)PCI 04:00:04 VendorID  (R)0x10B5                (D)0x10B5
DBG)PCI 04:00:04 DeviceID  (R)0x87D0                (D)0x87D0
DBG)PCI 04:00:04 ClassCode (R)088000                (D)088000
ProgInterface          (R)0                     (D)0
BaseClass              (R)Base System Peripherals(D)Base System Peripherals
SubClass               (R)Other system peripheral(D)Other system peripheral
DBG)PCI Bus:Dev:Func       (R)05:00:00              (D)05:00:00
DBG)PCI 05:00:00 VendorID  (R)0x10B5                (D)0x10B5
DBG)PCI 05:00:00 DeviceID  (R)0x8725                (D)0x8725
DBG)PCI 05:00:00 ClassCode (R)060400                (D)060400
ProgInterface          (R)0                     (D)0
BaseClass              (R)Bridge Device         (D)Bridge Device
SubClass               (R)PCI/PCI bridge        (D)PCI/PCI bridge
DBG)PCI 05:00:00 LinkStat  (R)Link not UP           (D)Link not UP
DBG)PCI 05:00:00 Wdth/Spd  (R)Undefined             (D)Undefined
INF)PCI 05:01:00           ignored
DBG)PCI Bus:Dev:Func       (R)05:08:00              (D)05:08:00
DBG)PCI 05:08:00 VendorID  (R)0x10B5                (D)0x10B5
DBG)PCI 05:08:00 DeviceID  (R)0x8725                (D)0x8725
DBG)PCI 05:08:00 ClassCode (R)060400                (D)060400
```

```
ProgInterface          (R)0                    (D)0
BaseClass              (R)Bridge Device        (D)Bridge Device
SubClass               (R)PCI/PCI bridge       (D)PCI/PCI bridge
DBG)PCI 05:08:00 LinkStat  (R)Link not UP          (D)Link not UP
DBG)PCI 05:08:00 Wdth/Spd  (R)Undefined            (D)Undefined
DBG)PCI Bus:Dev:Func       (R)07:00:00             (D)07:00:00
DBG)PCI 07:00:00 VendorID  (R)0x111D               (D)0x111D
DBG)PCI 07:00:00 DeviceID  (R)0x808C               (D)0x808C
DBG)PCI 07:00:00 ClassCode (R)060400               (D)060400
ProgInterface          (R)0                    (D)0
BaseClass              (R)Bridge Device        (D)Bridge Device
SubClass               (R)PCI/PCI bridge       (D)PCI/PCI bridge
DBG)PCI 07:00:00 LinkStat  (R)Link is UP           (D)Link is UP
DBG)PCI 07:00:00 Wdth/Spd  (R)x4 / 5GT/s           (D)x4 / 5GT/s
DBG)PCI Bus:Dev:Func       (R)08:02:00             (D)08:02:00
DBG)PCI 08:02:00 VendorID  (R)0x111D               (D)0x111D
DBG)PCI 08:02:00 DeviceID  (R)0x808C               (D)0x808C
DBG)PCI 08:02:00 ClassCode (R)060400               (D)060400
ProgInterface          (R)0                    (D)0
BaseClass              (R)Bridge Device        (D)Bridge Device
SubClass               (R)PCI/PCI bridge       (D)PCI/PCI bridge
DBG)PCI 08:02:00 LinkStat  (R)Link not UP          (D)Link not UP
DBG)PCI 08:02:00 Wdth/Spd  (R)Undefined            (D)Undefined
DBG)PCI Bus:Dev:Func       (R)08:04:00             (D)08:04:00
DBG)PCI 08:04:00 VendorID  (R)0x111D               (D)0x111D
DBG)PCI 08:04:00 DeviceID  (R)0x808C               (D)0x808C
DBG)PCI 08:04:00 ClassCode (R)060400               (D)060400
ProgInterface          (R)0                    (D)0
BaseClass              (R)Bridge Device        (D)Bridge Device
SubClass               (R)PCI/PCI bridge       (D)PCI/PCI bridge
DBG)PCI 08:04:00 LinkStat  (R)Link not UP          (D)Link not UP
DBG)PCI 08:04:00 Wdth/Spd  (R)Undefined            (D)Undefined
DBG)PCI Bus:Dev:Func       (R)08:06:00             (D)08:06:00
DBG)PCI 08:06:00 VendorID  (R)0x111D               (D)0x111D
DBG)PCI 08:06:00 DeviceID  (R)0x808C               (D)0x808C
DBG)PCI 08:06:00 ClassCode (R)060400               (D)060400
ProgInterface          (R)0                    (D)0
BaseClass              (R)Bridge Device        (D)Bridge Device
SubClass               (R)PCI/PCI bridge       (D)PCI/PCI bridge
DBG)PCI 08:06:00 LinkStat  (R)Link not UP          (D)Link not UP
DBG)PCI 08:06:00 Wdth/Spd  (R)Undefined            (D)Undefined
DBG)PCI Bus:Dev:Func       (R)08:08:00             (D)08:08:00
DBG)PCI 08:08:00 VendorID  (R)0x111D               (D)0x111D
DBG)PCI 08:08:00 DeviceID  (R)0x808C               (D)0x808C
DBG)PCI 08:08:00 ClassCode (R)060400               (D)060400
ProgInterface          (R)0                    (D)0
BaseClass              (R)Bridge Device        (D)Bridge Device
SubClass               (R)PCI/PCI bridge       (D)PCI/PCI bridge
DBG)PCI 08:08:00 LinkStat  (R)Link not UP          (D)Link not UP
DBG)PCI 08:08:00 Wdth/Spd  (R)Undefined            (D)Undefined
DBG)PCI Bus:Dev:Func       (R)08:0C:00             (D)08:0C:00
DBG)PCI 08:0C:00 VendorID  (R)0x111D               (D)0x111D
DBG)PCI 08:0C:00 DeviceID  (R)0x808C               (D)0x808C
DBG)PCI 08:0C:00 ClassCode (R)060400               (D)060400
ProgInterface          (R)0                    (D)0
BaseClass              (R)Bridge Device        (D)Bridge Device
SubClass               (R)PCI/PCI bridge       (D)PCI/PCI bridge
DBG)PCI 08:0C:00 LinkStat  (R)Link not UP          (D)Link not UP
DBG)PCI 08:0C:00 Wdth/Spd  (R)Undefined            (D)Undefined
```

```
DBG)PCI Bus:Dev:Func       (R)08:10:00              (D)08:10:00
DBG)PCI 08:10:00 VendorID  (R)0x111D                (D)0x111D
DBG)PCI 08:10:00 DeviceID  (R)0x808C                (D)0x808C
DBG)PCI 08:10:00 ClassCode (R)060400                (D)060400
ProgInterface          (R)0                      (D)0
BaseClass              (R)Bridge Device          (D)Bridge Device
SubClass               (R)PCI/PCI bridge         (D)PCI/PCI bridge
DBG)PCI 08:10:00 LinkStat  (R)Link not UP           (D)Link not UP
DBG)PCI 08:10:00 Wdth/Spd  (R)Undefined             (D)Undefined
DBG)PCI Bus:Dev:Func       (R)08:14:00              (D)08:14:00
DBG)PCI 08:14:00 VendorID  (R)0x111D                (D)0x111D
DBG)PCI 08:14:00 DeviceID  (R)0x808C                (D)0x808C
DBG)PCI 08:14:00 ClassCode (R)060400                (D)060400
ProgInterface          (R)0                      (D)0
BaseClass              (R)Bridge Device          (D)Bridge Device
SubClass               (R)PCI/PCI bridge         (D)PCI/PCI bridge
DBG)PCI 08:14:00 LinkStat  (R)Link not UP           (D)Link not UP
DBG)PCI 08:14:00 Wdth/Spd  (R)Undefined             (D)Undefined
DBG)PCI Bus:Dev:Func       (R)11:00:00              (D)11:00:00
DBG)PCI 11:00:00 VendorID  (R)0x8086                (D)0x8086
DBG)PCI 11:00:00 DeviceID  (R)0x1533                (D)0x1533
DBG)PCI 11:00:00 ClassCode (R)020000                (D)020000
ProgInterface          (R)0                      (D)0
BaseClass              (R)Network Controller     (D)Network Controller
SubClass               (R)Ethernet controller    (D)Ethernet controller
DBG)PCI Bus:Dev:Func       (R)12:00:00              (D)12:00:00
DBG)PCI 12:00:00 VendorID  (R)0x8086                (D)0x8086
DBG)PCI 12:00:00 DeviceID  (R)0x1533                (D)0x1533
DBG)PCI 12:00:00 ClassCode (R)020000                (D)020000
ProgInterface          (R)0                      (D)0
BaseClass              (R)Network Controller     (D)Network Controller
SubClass               (R)Ethernet controller    (D)Ethernet controller
DBG)CPLD Version           (R)0x3                   (D)0x3
DBG)CPLD Dram Size         (R)8GB                   (D)8GB
INF)CPLD SysCon            ignored
DBG)CPLD Geo Address       (R)1                     (D)1
DBG)CPLD XMC Presence      (R)XMC not present       (D)XMC not present
DBG)SATA Port 0            (R)WDC WD2503ABYZ (251.0GB)(D)WDC WD2503ABYZ (251.0GB)
DBG)SATA Port 1            (R)WDC WD2502ABYS (251.0GB)(D)WDC WD2502ABYS (251.0GB)
DBG)SATA Port 2            (R)Not Connected (0.0GB)  (D)Not Connected (0.0GB)
DBG)SATA Port 3            (R)Not Connected (0.0GB)  (D)Not Connected (0.0GB)
INF)SATA Port 4            ignored
DBG)SATA Port 5            (R)TS32GMTS400    (32.0GB)(D)TS32GMTS400    (32.0GB)
DBG)SMBUS0 Device 0        (R)0x30                  (D)0x30
INF)BIOS Checksum          ignored
DBG)BIOS Version           (R)ID16133               (D)ID16133
DBG)ETH Port 0             (R)Link UP speed 1000 Mb/s(D)Link UP speed 1000 Mb/s
DBG)ETH Port 1             (R)Not Connected         (D)Not Connected
INF)ETH Port 2             ignored
INF)ETH Port 3             ignored
DBG)USB Drive´s Count      (R)4 Drives              (D)4 Drives
DBG)USB Kbd´s Count        (R)0 Keyboard            (D)0 Keyboard
DBG)USB Mouse´s Count      (R)0 Mouse               (D)0 Mouse
DBG)USB Hub´s Count        (R)1 Hub                 (D)1 Hub
DBG)USB Point´s Count      (R)0 Point               (D)0 Point
DBG)USB Ccid´s Count       (R)0 SmartCard Reader    (D)0 SmartCard Reader
DBG)USB MassStorage(0)     (R)KingstonDataTravele   (D)KingstonDataTravele
DBG)USB MassStorage(1)     (R)KingstonDataTravele   (D)KingstonDataTravele
DBG)USB MassStorage(2)     (R)UFD 3.0 Silicon-Pow   (D)UFD 3.0 Silicon-Pow
DBG)USB MassStorage(3)     (R)KingstonDataTravele   (D)KingstonDataTravele
```

```
DBG)USB Port 0              (R)Connected              (D)Connected
DBG)USB Port 1              (R)Connected              (D)Connected
DBG)USB Port 2              (R)Connected              (D)Connected
DBG)USB Port 3              (R)Connected              (D)Connected
DBG)USB Port 4              (R)Not Connected          (D)Not Connected
PASSED
```

You must exit the menu so that settings changes are taken into account.

2.11 Bypass the PBIT Tests

The following toggle command can be used to avoid the PBIT to run the tests:

```
VX3058> kdiag bypass
PBIT BYPASS set
```

To disable the bypass mode, re-execute the same command:

```
VX3058> kdiag bypass
PBIT BYPASS removed
```

This is useful to bypass the PBIT tests if they are configured to run automatically.

This feature is also accessible thru the **kdiag** Linux utility to enable/disable the tests under OS at the next boot.

3 / PBIT and OS Interfaces

The PBIT synthetic result and the PBIT detailed results are accessible under the Operating System delivered with the VX305x board.

The following gives you an overview of the facilities available under Linux Board Support Package.

Please refer to the appropriate OS Release Notes document for more details.

3.1 Linux

3.1.1 Linux Synthetic PBIT Result

The synthetic PBIT result stored in the CPLD register 0x2 is accessible under Linux with the "**cpldtool**" facility.

Use the "**-a**" option and check the register 0x2 value.

- Example:

```
# cpldtool -a
VX3058 detected
Reg 0x0 - CPLD_ID = 0x03
        CPLD_ID=0x0
        CPLD_Debug=0x0
        CPLD_Version=0x3
Reg 0x1 - PCB_ID = 0x46
Reg 0x2 - FIRM_POST = 0x00
        PBIT_FAIL=0x0
        PBIT_RUN=0x0
Reg 0x3 - PWON_STATUS = 0x00
        DPOST=0x0
        BOARD_CLASS=0x0
        POST_RTC=0x0
Reg 0x4 - PWR_RST_CONFIG = 0x02
        PWRON_MODE=0x0
        PEXRST_Dis=0x0
        Alert_inhib=0x0
        PEXRST_Ctl=0x1
Reg 0x7 - SERIAL_LINES_CTL = 0x04
        SERIAL_MODE=0x0
```

3.1.2 Linux Detailed PBIT Result

The detailed PBIT results are accessible under Linux with the "**sysvartool**" facility.

Use the "**-A pbit -l**" options to display the PBIT results.

- Example:

```
# sysvartool -A pbit -l
VX3058 detected
POSTs configured to run from command line:
       mem_data: PASSED
       mem_addr: PASSED
   mem_pattern1: PASSED
   mem_pattern2: PASSED
   mem_pattern3: PASSED
   mem_pattern4: PASSED
           core: PASSED
```

```
    pcie_vpx_sw: PASSED
         serial: PASSED
            rtc: PASSED
    ether_loop0: PASSED
    ether_loop1: PASSED
    ether_loop2: PASSED
    ether_loop3: PASSED
          hwmon: PASSED
sata0_controler: PASSED
sata1_controler: PASSED
            vpd: PASSED
ehci_controller: PASSED
xhci_controller: PASSED

PASSED  : 20
FAILED  : 0
NOT RUN : 0
TOTAL   : 20
```

3.1.3 Linux Kdiag Utility

The Linux "**kdiag**" utility can be used to configure the PBIT directly under OS.

It is distributed in the BIT-6-VX305x product including the PBIT Firmware and the Linux BIT for VX305x.

BIT-6-VX305x ID16133 includes the Linux BIT V2.3 ID16130 that supports the following features:

- Listing and modifying the PBIT test list to be run at the next boot
- Getting and clearing the PBIT status
- Editing the system menu and choose which items will be ignored during the next system test executions
- Bypassing the PBIT tests at the next boot

The Linux **kdiag** command syntax is similar to the PBIT **kdiag** command used under the BIOS EFI shell:

```
# kdiag help
kdiag    - perform board diagnostics
  ----- Usage -----
Print list of PBITs and infos about them :
  kdiag <PBITname>|<PBITnum> ...
    <PBITname>|<PBITnum> ...
             list of PBIT(s) to display. All if the list is empty.
             PBIT(s) are referenced using their name or their number.
Print PBIT(s) status :
  kdiag stat [oside] [<PBITname>|<PBITnum> ...]
    <PBITname>|<PBITnum> ...
             list of PBIT(s) to display. All if the list is empty.
             PBIT(s) are referenced using their name or their number.
oside (VX6060 board only): print PBIT status of peer CPU side
Clear PBIT(s) status :
  kdiag [oside] clrstat|clrallstat [<PBITname>|<PBITnum> ...]
    clrstat    : Reset status to NOTRUN
    clrallstat : Reset status to NOTRUN and clear the (FAILED once) flag
    <PBITname>|<PBITnum> ...
             list of PBIT(s) to clear. All if the list is empty.
             PBIT(s) are referenced using their name or their number.
    oside (VX6060 board only): clear PBIT status of peer CPU side
Delete all PBITs from configuration :
  kdiag [oside] deleteall
    oside (VX6060 board only): delete all PBITs for peer CPU side
```

```
Configure PBIT(s) :
  kdiag [oside] cfg <cfgarg> ... <PBITname>|<PBITnum> ...
    cfg <cfgarg> : Configure one or several PBIT(s).
                   <cfgarg> is either :
                   - "delete" to delete PBIT(s) from the list of configured PBITs
                   - "default" to configure PBIT(s) with a default run mode
                   - a comma separated list of runflags defining a PBIT
                     run mode; for example : fast,complex.
                   valid runflags are :
                   - "SPEED" flags (can NOT be mixed)
                     slow          : run in slow mode (full testing)
                     fast          : run in fast mode (fast testing)
                   - "CONFIG" flags (can NOT be mixed)
                     simple        : run in simple mode (no external hardware)
                     complex       : run in complex mode (needs external hardware)
                   - "HALT" flags (can NOT be mixed)
                     haltonfail    : halt immediately (hang) if test fails
                     promptonfail  : halt at U-Boot prompt (no OS) if test fails
                   - "RESET" flags (can be mixed together)
                     normalreset   : run after a normal reset
                     poweronreset  : run after a power-on reset
                     allresets     : run after all resets listed above
[<PBITname>|<PBITnum>] ...
                   list of PBIT(s) to configure.
                   PBIT(s) are referenced using their name or their number.
                   All missing tests if the list is empty (but the cfgarg is default).
    oside (VX6060 board only): Configure PBIT(s) for peer CPU side
Edit System Configuration for system Test:
(cf documentation for details)
  kdiag system_edit or kdiag edit system
Clear System Configuration for system Test:
  kdiag system_clear or kdiag clear system
Display kdiag command version :
  kdiag version
Toggle PBIT Bypassed:
  kdiag bypass
```

For example:

```
# kdiag version
kdiag command Version:  V2.3    16130
# kdiag
PBITs configured to run from command line :
 mem_data (1) -    capabilities : fast,simple,allresets
   run mode 1   : fast,simple,allresets
 mem_addr (2) -    capabilities : fast,simple,allresets
   run mode 1   : fast,simple,allresets
 mem_pattern1 (6) -    capabilities : slow/fast,simple,allresets
   run mode 1   : slow,simple,allresets
 mem_pattern2 (7) -    capabilities : slow/fast,simple,allresets
   run mode 1   : slow,simple,allresets
 mem_pattern3 (8) -    capabilities : slow/fast,simple,allresets
   run mode 1   : slow,simple,allresets
 mem_pattern4 (9) -    capabilities : slow/fast,simple,allresets
   run mode 1   : slow,simple,allresets
 core (10) -    capabilities : fast,simple,allresets
   run mode 1   : fast,simple,allresets
 tpm (11) -    capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
```

```
 pcie_vpx_sw (13) -    capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 m2_bottom (14) -    capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 m2_top (15) -    capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 serial (16) -    capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 rtc (20) -    capabilities : fast,simple,allresets
   run mode 1   : fast,simple,allresets
 sysflash (22) -    capabilities : fast,simple,allresets
   run mode 1   : fast,simple,allresets
 cpld (24) -    capabilities : fast,simple/complex,allresets
   run mode 1   : fast,complex,allresets
 fram (40) -    capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 ether_loop0 (55) -    capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 ether_loop1 (56) -    capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 ether_loop2 (57) -    capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 ether_loop3 (58) -    capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 hwmon (61) -    capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 sata0_controler (68) -    capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 sata1_controler (69) -    capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 vpd (70) -    capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 eeprom (71) -    capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets
 ehci_controller (85) -    capabilities : fast,simple,allresets
   run mode 1   : fast,simple,allresets
 xhci_controller (86) -    capabilities : fast,simple,allresets
   run mode 1   : fast,simple,allresets
 system (89) -    capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,allresets


Other PBITs available but not yet configured :

 mem_bitflip (3) -    capabilities : slow/fast,simple,allresets
 mem_addrpat (4) -    capabilities : slow/fast,simple,allresets
 mem_addrpat2 (5) -    capabilities : slow/fast,simple,allresets
 smbus0 (26) -    capabilities : fast,simple/complex,allresets
 smbus1 (27) -    capabilities : fast,simple/complex,allresets
 ether_link0 (50) -    capabilities : fast,simple/complex,allresets
 ether_link1 (51) -    capabilities : fast,simple/complex,allresets
 ether_link2 (52) -    capabilities : fast,simple/complex,allresets
 ether_link3 (53) -    capabilities : fast,simple/complex,allresets
 sata0_dev_see (62) -    capabilities : fast,simple,allresets
 sata1_dev_see (63) -    capabilities : fast,simple,allresets
 sata2_dev_see (64) -    capabilities : fast,simple,allresets
 sata3_dev_see (65) -    capabilities : fast,simple,allresets
 sata4_dev_see (66) -    capabilities : fast,simple,allresets
 sata5_dev_see (67) -    capabilities : fast,simple,allresets
 usb0_dev_see (80) -    capabilities : fast,simple,allresets
 usb1_dev_see (81) -    capabilities : fast,simple,allresets
 usb2_dev_see (82) -    capabilities : fast,simple,allresets
```

```
 usb3_dev_see (83) -    capabilities : fast,simple,allresets
 faultytest (98) -    capabilities : fast,simple,allresets
 hangtest (99) -    capabilities : fast,simple,allresets

Use ´ kdiag help´ to get more info.

# kdiag stat
Display status
Status of PBITs configured to run from command line :
 PASSED : mem_data (fast,simple)
 PASSED : mem_addr (fast,simple)
 PASSED : mem_pattern1 (slow,simple)
 PASSED : mem_pattern2 (slow,simple)
 PASSED : mem_pattern3 (slow,simple)
 PASSED : mem_pattern4 (slow,simple)
 PASSED : core (fast,simple)
 PASSED : tpm (fast,simple)
 PASSED : pcie_vpx_sw (fast,simple)
 PASSED : m2_bottom (fast,simple)
 PASSED : m2_top (fast,simple)
 PASSED : serial (fast,simple)
 PASSED : rtc (fast,simple)
 PASSED : sysflash (fast,simple)
 FAILED : cpld (fast,complex)
 PASSED : fram (fast,simple)
 PASSED : ether_loop0 (fast,simple)
 PASSED : ether_loop1 (fast,simple)
 PASSED : ether_loop2 (fast,simple)
 PASSED : ether_loop3 (fast,simple)
 PASSED : hwmon (fast,simple)
 PASSED : sata0_controler (fast,simple)
 PASSED : sata1_controler (fast,simple)
 PASSED : vpd (fast,simple)
 PASSED : eeprom (fast,simple)
 PASSED : ehci_controller (fast,simple)
 PASSED : xhci_controller (fast,simple)
 PASSED : system (fast,simple)

RUN    : 28
PASSED : 27
FAILED : 1
NOT_RUN: 0

# kdiag edit system
           <<  KONTRON SYSTEM PBIT : EDIT MODE  >>

Edit by feature, choose:PCI CPLD SATA ETH USB SMBUS BIOS
Other available cmds: `s´ for settings, `?´ for help,  `q´ to quit edit mode

--SystemEdit>>pci

--SystemEdit-PCI>>?

pa : print all Recorded PCI device infos
pb : print all Recorded PCI/PCI Bridge infos
p1 : print Recorded VendorID only
p2 : print Recorded DeviceID only
p3 : print Recorded ClassCode only
p4 : print Recorded Linkstatus only (only for PCI/PCI bridge)
p5 : print Recorded Link Speed/Width only (only for PCI/PCI bridge)
p : print Recorded values
0,..,49 : edit PCI element
```

```
ignoreall : ignore all PCI infos
default : restore default PCI config (remove flags)
? : help
q : go back to main menu
--SystemEdit-PCI>>pa

0. PCI 00:00:00
VendorID                0x8086
DeviceID                0x6f00
ClassCode               060000
1. PCI 00:01:00
VendorID                0x8086
DeviceID                0x6f02
ClassCode               060400
LinkStatus              Link not UP
Link Width/Speed        Undefined
2. PCI 00:02:00
VendorID                0x8086
DeviceID                0x6f04
ClassCode               060400
LinkStatus              Link is UP
Link Width/Speed        x1 / 2.5GT/s
3. PCI 00:02:02
VendorID                0x8086
DeviceID                0x6f06
ClassCode               060400
LinkStatus              Link is UP
Link Width/Speed        x1 / 2.5GT/s
4. PCI 00:03:00
VendorID                0x8086
DeviceID                0x6f08
ClassCode               060400
LinkStatus              Link is UP
Link Width/Speed        x8 / 8GT/s
5. PCI 00:03:02
VendorID                0x8086
DeviceID                0x6f0a
ClassCode               060400
LinkStatus              Link is UP
Link Width/Speed        x1 / 2.5GT/s
6. PCI 00:03:03
VendorID                0x8086
DeviceID                0x6f0b
ClassCode               060400
LinkStatus              Link is UP
Link Width/Speed        x1 / 2.5GT/s
7. PCI 00:05:00
VendorID                0x8086
DeviceID                0x6f28
ClassCode               088000
--SystemEdit-PCI>>pb

1. PCI 00:01:00 :
VendorID                0x8086
DeviceID                0x6f02
ClassCode               060400
LinkStatus              Link not UP
Link Width/Speed        Undefined
2. PCI 00:02:00 :
VendorID                0x8086
DeviceID                0x6f04
ClassCode               060400
```

```
LinkStatus              Link is UP
Link Width/Speed        x1 / 2.5GT/s
3. PCI 00:02:02 :
VendorID                0x8086
DeviceID                0x6f06
ClassCode               060400
LinkStatus              Link is UP
Link Width/Speed        x1 / 2.5GT/s
4. PCI 00:03:00 :
VendorID                0x8086
DeviceID                0x6f08
ClassCode               060400
LinkStatus              Link is UP
Link Width/Speed        x8 / 8GT/s
5. PCI 00:03:02 :
VendorID                0x8086
DeviceID                0x6f0a
ClassCode               060400
LinkStatus              Link is UP
Link Width/Speed        x1 / 2.5GT/s
6. PCI 00:03:03 :
VendorID                0x8086
DeviceID                0x6f0b
ClassCode               060400
LinkStatus              Link is UP
Link Width/Speed        x1 / 2.5GT/s
22. PCI 04:00:00 :
VendorID                0x10b5
DeviceID                0x8725
ClassCode               060400
LinkStatus              Link is UP
Link Width/Speed        x8 / 8GT/s
27. PCI 05:00:00 :
VendorID                0x10b5
DeviceID                0x8725
ClassCode               060400
LinkStatus              Link not UP
Link Width/Speed        Undefined
28. PCI 05:01:00 :
VendorID                0x10b5
DeviceID                0x8725
ClassCode               060400
LinkStatus              Link not UP
Link Width/Speed        Undefined
29. PCI 05:08:00 :
VendorID                0x10b5
DeviceID                0x8725
ClassCode               060400
LinkStatus              Link not UP
Link Width/Speed        Undefined
--SystemEdit-PCI>>p1

0.  PCI 00:00:00 VendorID  0x8086
1.  PCI 00:01:00 VendorID  0x8086
2.  PCI 00:02:00 VendorID  0x8086
3.  PCI 00:02:02 VendorID  0x8086
4.  PCI 00:03:00 VendorID  0x8086
5.  PCI 00:03:02 VendorID  0x8086
6.  PCI 00:03:03 VendorID  0x8086
7.  PCI 00:05:00 VendorID  0x8086
8.  PCI 00:05:01 VendorID  0x8086
9.  PCI 00:05:02 VendorID  0x8086
```

```
10. PCI 00:05:04 VendorID  0x8086
11. PCI 00:14:00 VendorID  0x8086
12. PCI 00:1d:00 VendorID  0x8086
13. PCI 00:1f:00 VendorID  0x8086
14. PCI 00:1f:02 VendorID  0x8086
15. PCI 00:1f:03 VendorID  0x8086
16. PCI 02:00:00 VendorID  0x8086
17. PCI 02:00:01 VendorID  0x8086
18. PCI 02:00:02 VendorID  0x8086
19. PCI 02:00:03 VendorID  0x8086
20. PCI 03:00:00 VendorID  0x8086
21. PCI 03:00:01 VendorID  0x8086
22. PCI 04:00:00 VendorID  0x10b5
23. PCI 04:00:01 VendorID  0x10b5
24. PCI 04:00:02 VendorID  0x10b5
25. PCI 04:00:03 VendorID  0x10b5
26. PCI 04:00:04 VendorID  0x10b5
27. PCI 05:00:00 VendorID  0x10b5
28. PCI 05:01:00 VendorID  0x10b5
29. PCI 05:08:00 VendorID  0x10b5
30. PCI 09:00:00 VendorID  0x8086
31. PCI 0a:00:00 VendorID  0x8086
32. PCI 00:00:00 VendorID  No Device
33. PCI 00:00:00 VendorID  No Device
34. PCI 00:00:00 VendorID  No Device
35. PCI 00:00:00 VendorID  No Device
36. PCI 00:00:00 VendorID  No Device
37. PCI 00:00:00 VendorID  No Device
38. PCI 00:00:00 VendorID  No Device
39. PCI 00:00:00 VendorID  No Device
40. PCI 00:00:00 VendorID  No Device
41. PCI 00:00:00 VendorID  No Device
42. PCI 00:00:00 VendorID  No Device
43. PCI 00:00:00 VendorID  No Device
44. PCI 00:00:00 VendorID  No Device
45. PCI 00:00:00 VendorID  No Device
46. PCI 00:00:00 VendorID  No Device
47. PCI 00:00:00 VendorID  No Device
48. PCI 00:00:00 VendorID  No Device
49. PCI 00:00:00 VendorID  No Device
--SystemEdit-PCI>>p2

0.  PCI 00:00:00 DeviceID  0x6f00
1.  PCI 00:01:00 DeviceID  0x6f02
2.  PCI 00:02:00 DeviceID  0x6f04
3.  PCI 00:02:02 DeviceID  0x6f06
4.  PCI 00:03:00 DeviceID  0x6f08
5.  PCI 00:03:02 DeviceID  0x6f0a
6.  PCI 00:03:03 DeviceID  0x6f0b
7.  PCI 00:05:00 DeviceID  0x6f28
8.  PCI 00:05:01 DeviceID  0x6f29
9.  PCI 00:05:02 DeviceID  0x6f2a
10. PCI 00:05:04 DeviceID  0x6f2c
11. PCI 00:14:00 DeviceID  0x8c31
12. PCI 00:1d:00 DeviceID  0x8c26
13. PCI 00:1f:00 DeviceID  0x8c54
14. PCI 00:1f:02 DeviceID  0x8c02
15. PCI 00:1f:03 DeviceID  0x8c22
16. PCI 02:00:00 DeviceID  0x6f50
17. PCI 02:00:01 DeviceID  0x6f51
18. PCI 02:00:02 DeviceID  0x6f52
19. PCI 02:00:03 DeviceID  0x6f53
```

```
20. PCI 03:00:00 DeviceID  0x15ab
21. PCI 03:00:01 DeviceID  0x15ab
22. PCI 04:00:00 DeviceID  0x8725
23. PCI 04:00:01 DeviceID  0x87d0
24. PCI 04:00:02 DeviceID  0x87d0
25. PCI 04:00:03 DeviceID  0x87d0
26. PCI 04:00:04 DeviceID  0x87d0
27. PCI 05:00:00 DeviceID  0x8725
28. PCI 05:01:00 DeviceID  0x8725
29. PCI 05:08:00 DeviceID  0x8725
30. PCI 09:00:00 DeviceID  0x1533
31. PCI 0a:00:00 DeviceID  0x1533
32. PCI 00:00:00 DeviceID  No Device
33. PCI 00:00:00 DeviceID  No Device
34. PCI 00:00:00 DeviceID  No Device
35. PCI 00:00:00 DeviceID  No Device
36. PCI 00:00:00 DeviceID  No Device
37. PCI 00:00:00 DeviceID  No Device
38. PCI 00:00:00 DeviceID  No Device
39. PCI 00:00:00 DeviceID  No Device
--SystemEdit-PCI>>p3

0.  PCI 00:00:00 ClassCode 060000
1.  PCI 00:01:00 ClassCode 060400
2.  PCI 00:02:00 ClassCode 060400
3.  PCI 00:02:02 ClassCode 060400
4.  PCI 00:03:00 ClassCode 060400
ERROR result Input = 0
5.  PCI 00:03:02 ClassCode 060400
6.  PCI 00:03:03 ClassCode 060400
7.  PCI 00:05:00 ClassCode 088000
8.  PCI 00:05:01 ClassCode 088000
9.  PCI 00:05:02 ClassCode 088000
ERROR result Input = 0
10. PCI 00:05:04 ClassCode 080020
11. PCI 00:14:00 ClassCode 0c0330
12. PCI 00:1d:00 ClassCode 0c0320
13. PCI 00:1f:00 ClassCode 060100
14. PCI 00:1f:02 ClassCode 010601
ERROR result Input = 0
15. PCI 00:1f:03 ClassCode 0c0500
16. PCI 02:00:00 ClassCode 088000
17. PCI 02:00:01 ClassCode 088000
18. PCI 02:00:02 ClassCode 088000
19. PCI 02:00:03 ClassCode 088000
ERROR result Input = 0
20. PCI 03:00:00 ClassCode 020000
21. PCI 03:00:01 ClassCode 020000
22. PCI 04:00:00 ClassCode 060400
23. PCI 04:00:01 ClassCode 088000
24. PCI 04:00:02 ClassCode 088000
ERROR result Input = 0
25. PCI 04:00:03 ClassCode 088000
26. PCI 04:00:04 ClassCode 088000
27. PCI 05:00:00 ClassCode 060400
28. PCI 05:01:00 ClassCode 060400
29. PCI 05:08:00 ClassCode 060400
ERROR result Input = 0
30. PCI 09:00:00 ClassCode 020000
31. PCI 0a:00:00 ClassCode 020000
32. PCI 00:00:00 ClassCode No Device
33. PCI 00:00:00 ClassCode No Device
```

```
34. PCI 00:00:00 ClassCode No Device
--SystemEdit-PCI>>p4

1.  PCI 00:01:00 LinkStat  Link not UP
2.  PCI 00:02:00 LinkStat  Link is UP
3.  PCI 00:02:02 LinkStat  Link is UP
4.  PCI 00:03:00 LinkStat  Link is UP
5.  PCI 00:03:02 LinkStat  Link is UP
6.  PCI 00:03:03 LinkStat  Link is UP
22. PCI 04:00:00 LinkStat  Link is UP
27. PCI 05:00:00 LinkStat  Link not UP
28. PCI 05:01:00 LinkStat  Link not UP
29. PCI 05:08:00 LinkStat  Link not UP
--SystemEdit-PCI>>p5

1.  PCI 00:01:00 Wdth/Spd  Undefined
2.  PCI 00:02:00 Wdth/Spd  x1 / 2.5GT/s
3.  PCI 00:02:02 Wdth/Spd  x1 / 2.5GT/s
4.  PCI 00:03:00 Wdth/Spd  x8 / 8GT/s
5.  PCI 00:03:02 Wdth/Spd  x1 / 2.5GT/s
6.  PCI 00:03:03 Wdth/Spd  x1 / 2.5GT/s
22. PCI 04:00:00 Wdth/Spd  x8 / 8GT/s
27. PCI 05:00:00 Wdth/Spd  Undefined
28. PCI 05:01:00 Wdth/Spd  Undefined
29. PCI 05:08:00 Wdth/Spd  Undefined
--SystemEdit-PCI>>
           <<  KONTRON SYSTEM PBIT : EDIT MODE  >>

Edit by feature, choose:PCI CPLD SATA ETH USB SMBUS BIOS
Other available cmds: `s´ for settings, `?´ for help,  `q´ to quit edit mode

--SystemEdit>>cpld

--SystemEdit-CPLD>>?

p : print Recorded values
0,..,4 : edit CPLD element
ignoreall : ignore all CPLD infos
default : restore default CPLD config (remove flags)
? : help
q : go back to main menu
--SystemEdit-CPLD>>p

0. CPLD Version           0x3
1. CPLD Dram Size         8GB
2. CPLD SysCon            System Controller
3. CPLD Geo Address       1
4. CPLD XMC Presence      XMC not present
--SystemEdit-CPLD>>
           <<  KONTRON SYSTEM PBIT : EDIT MODE  >>

Edit by feature, choose:PCI CPLD SATA ETH USB SMBUS BIOS
Other available cmds: `s´ for settings, `?´ for help,  `q´ to quit edit mode

--SystemEdit>>

# kdiag bypass
PBIT BYPASS set
# kdiag bypass
PBIT BYPASS removed
```

4 / Use Cases

The following "Use Cases" are covered by PBIT:

- EVAL (platform EVALuation)
- DEVELOP (application or system DEVELOPment)
- DEPLOY (system production at the factory and DEPLOYment in the field)
- MAINTAIN (specific use case to MAINTAIN the system once deployed or to help repairing it)
- MANUFACTURING (used mostly by Kontron)

4.1 PBIT Features and Benefits

The following describes the PBIT features and benefits according to targeted Use Cases.

4.1.1 EVAL

PBIT can easily be evaluated from the BIOS, thanks to its interactive mode. Access to the PBIT operator interface and to the results can be done through the console serial line or through the HDMI screen associated with a USB mouse and keyboard.

Following the detailed sections of this manual, all the PBIT commands can be experimented.

4.1.2 DEVEL

The PBIT tests list can be modified in order to try various coverage/execution time trade-offs.

The "`system`" test can be used at the end of the development, to capture a complex peripheral configuration (PCIe, USB, SATA, CPLD infos, BIOS SETUP) which becomes the validated final system configuration for the deployment.

According to the complete system test policy, various PBIT results collection methods can be selected at application design time. The Operating System access to the PBIT results gives most of the information needed by a control and monitoring application to take a decision on the system following the boot steps.

The PBIT synthetic result register featured by the board control unit (cPLD) can be used from a management unit such as Kontron CMB or from other boards linked to the same SMB bus on the backplane. From this result, the Control Unit can alter the control registers to modify the behaviour of the board under control.

4.1.3 DEPLOY

PBIT software implements unrivalled features which make Kontron product deployment easier and cheaper. The PBIT code is located in the same device as the BIOS code. It is deployed along with the BIOS (see Appendix A of the BIOS User's Guide). The PBIT settings (and results) are located in their dedicated non volatile EEPROM device. To copy these settings, boot the Operating System and record the OS EEPROM content from the offset address 0x2000 up to 0x3FFF and store it in a file. Then use this file to copy back data to the new board OS EEPROM to configure.

One of the most useful features of PBIT for deployments is implemented in the "system" test. As this test scans all the possible I/O ports and devices to compare with a pre-recorded system reference, it can signal any change in the expected I/O devices availability. This can cover I/O device failures, or more subtle system alterations (for example a storage device left unconnected). The "system" test results are organized to pinpoint the faulty I/O port easily, before attempting a complete system boot and having to manage the corresponding cascading failures due to this incorrect peripheral configuration.

4.1.4 MAINTAIN

The following PBIT features can be very useful in the context of long term maintenance and troubleshooting.

The PBIT results record the behaviour of the last PBIT run (see section 2.5 page 17). In addition to these results, each test implements a dedicated "FAILED ONCE" flag which has to be reset separately (see section 2.6 page 18). In situations where a system is regularly re-started, possibly under the control of a higher level control device, this long term flag can be used by maintenance teams to search for the root cause of previously aborted boots.

Updating the BIOS and the PBIT code is done at the same time, using binary images containing both software, thanks to the same storage device being used by both.

PBIT extension: Thanks to its modular approach, and to the EFI execution environment. PBIT can easily be modified and expanded to match very specific application use case. Please contact GSS-MAR-Toulon@kontron.com to open a feature request discussion with Kontron.

4.1.5 MANUFACTURING

Thanks to the wide range of services the PBIT software can offer, PBIT is a powerful tool used during the manufacturing process of the boards.

Specific features have been implemented for that purpose.

Appendix A - List of Abbreviations

CPLD	Complex Programmable Logic Device
OS	Operating System
PBIT	Power on Built In Test
PMC	PCI Mezzanine Card
XMC	PCI Express Mezzanine Card

About Kontron

Kontron, a global leader in embedded computing technology and trusted advisor in IoT, works closely with its customers, allowing them to focus on their core competencies by offering a complete and integrated portfolio of hardware, software and services designed to help them make the most of their applications.

With a significant percentage of employees in research and development, Kontron creates many of the standards that drive the world's embedded computing platforms; bringing to life numerous technologies and applications that touch millions of lives. The result is an accelerated time-to-market, reduced total-cost-of-ownership, product longevity and the best possible overall application with leading-edge, highest reliability embedded technology

Kontron is a listed company. Its shares are traded in the Prime Standard segment of the Frankfurt Stock Exchange and on other exchanges under the symbol "KBC".
For more information, please visit: **www.kontron.com**

CORPORATE OFFICES

EUROPE, MIDDLE EAST & AFRICA

Lise-Meitner-Str. 3-5
86156 Augsburg
Germany
Tel.: + 49 821 4086-0
Fax: + 49 821 4086-111
info@kontron.com

NORTH AMERICA

14118 Stowe Drive
Poway, CA 92064-7147
USA
Tel.: + 1 888 294 4558
Fax: + 1 858 677 0898
info@us.kontron.com

ASIA PACIFIC

1~2F, 10 Building, No. 8 Liangshuihe 2nd Street,
Economical & Technological Development Zone,
Beijing, 100176, P.R. China
Tel.: + 86 10 63751188
Fax: + 86 10 83682438
info@kontron.cn