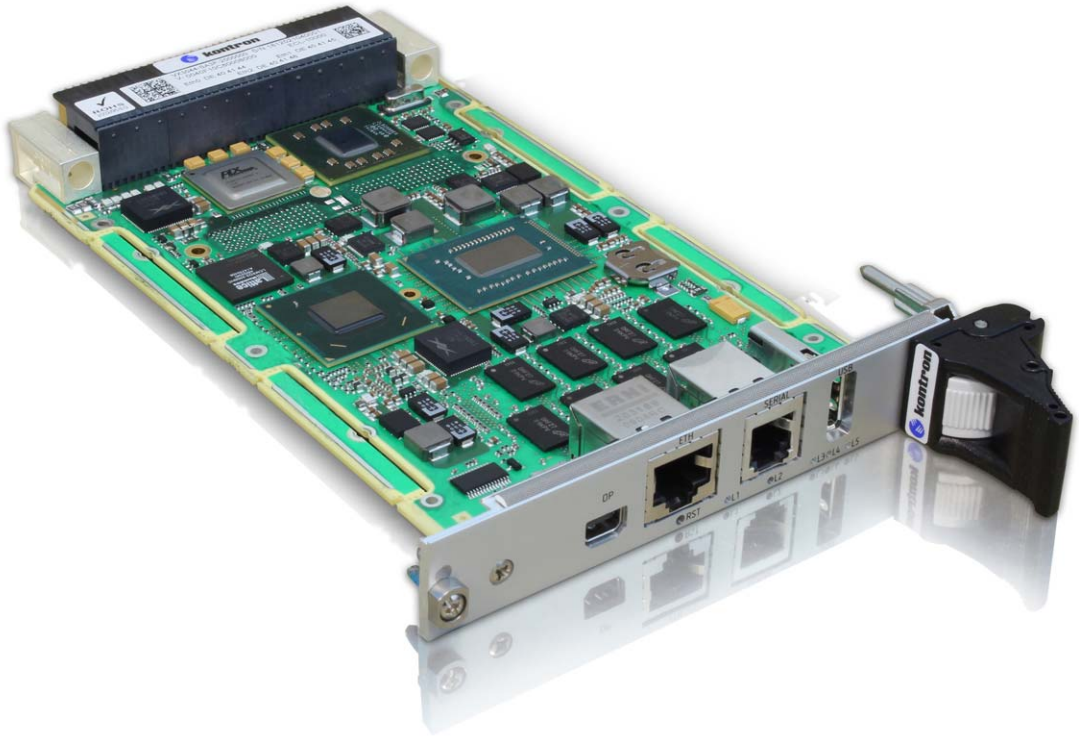
kontron

» VX304x «



PBIT User's Guide

SD.DT.G14-1e - October 2013

Revision History

Publication Title:	VX304x PBIT User's Guide	
Doc. ID:	SD.DT.G14-1e	
Rev.	Brief Description of Changes	Date of Issue
1e	Updated Sections: 1.3 - PBIT Configuration 1.5 - PBIT Test List 2.1 - On-Line Help 2.7.4 - Set a RUN mode parameter 2.9 - Display the PBIT Version 2.10 - PBIT System Test 4.1 - PBIT Features and Benefits	10-2013
0e	Initial Version	01-2013

Copyright © 2013 Kontron AG. All rights reserved. All data is for information purposes only and not guaranteed for legal purposes. Information has been carefully checked and is believed to be accurate; however, no responsibility is assumed for inaccuracies. Kontron and the Kontron logo and all other trademarks or registered trademarks are the property of their respective owners and are recognized. Specifications are subject to change without notice.

Proprietary Note

This document contains information proprietary to Kontron. It may not be copied or transmitted by any means, disclosed to others, or stored in any retrieval system or media without the prior written consent of Kontron or one of its authorized agents.

The information contained in this document is, to the best of our knowledge, entirely correct. However, Kontron cannot accept liability for any inaccuracies or the consequences thereof, or for any liability arising from the use or application of any circuit, product, or example shown in this document.

Kontron reserves the right to change, modify, or improve this document or the product described herein, as seen fit by Kontron without further notice.

Trademarks

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

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

Conventions

This guide uses several types of notice: Note, Caution, ESD.

Note: this notice calls attention to important features or instructions.


Caution: this notice alert you to system damage, loss of data, or risk of personal injury.


ESD: This banner indicates an Electrostatic Sensitive Device.

All numbers are expressed in decimal, except addresses and memory or register data, which are expressed in hexadecimal. The prefix `0x' shows a hexadecimal number, following the `C' programming language convention.

The multipliers `k', `M' and `G' have their conventional scientific and engineering meanings of $*10^3$, $*10^6$ and $*10^9$ respectively. The only exception to this is in the description of the size of memory areas, when `K', `M' and `G' mean $*2^{10}$, $*2^{20}$ and $*2^{30}$ respectively.

When describing transfer rates, `k' `M' and `G' mean $*10^3$, $*10^6$ and $*10^9$ *not $*2^{10}$ $*2^{20}$ and $*2^{30}$.*

In PowerPC terminology, multiple bit fields are numbered from 0 to n, where 0 is the MSB and n is the LSB. PCI and CompactPCI terminology follows the more familiar convention that bit 0 is the LSB and n is the MSB.

Signal names ending with an asterisk (*) or a hash (#) denote active low signals; all other signals are active high.

Signal names follow the PICMG 2.0 R3.0 CompactPCI Specification and the PCI Local Bus 2.3 Specification.

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


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.

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 on the previous page of this manual.

Table Of Contents

Chapter 1 - PBIT Overview

This document describes the PowerOn Built In Test (PBIT) for Kontron VX304x boards.

The PBIT is an optional product available under the VX304x 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 VX304x system 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 system EEPROM and can also be accessed and reconfigured under Operating System such as Linux or VxWorks . See Chapter 3 page 46.

> Simplified test result is also available in a 8-bit PLD register (register 0x2).

1.1 Related Documents

Hardware:

> VX304x 3U VPX User's Guide CA.DT.A98

> VX304x Hardware Release Notes CA.DT.A99

BIOS:

> VX304x AMI BIOS User Reference Manual SD.DT.F96

1.2 PBIT Installation and Activation

The PBIT software comes pre−installed in the system Flash, along with the EFI BIOS firmware, on the VX304x boards.

The PBIT can be activated on any VX304x. Please contact Kontron support team for more information.

To install a new BIOS version including a new PBIT version please refer to the VX304x BIOS User's Reference Manual - SD.DT.F96.

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 VX304x.

The list of tests to be executed can be displayed and modified by using the EFI Shell command "`kdiag`" (see section 1.5 page 9).

1.3.1 Configure the PBIT by command line

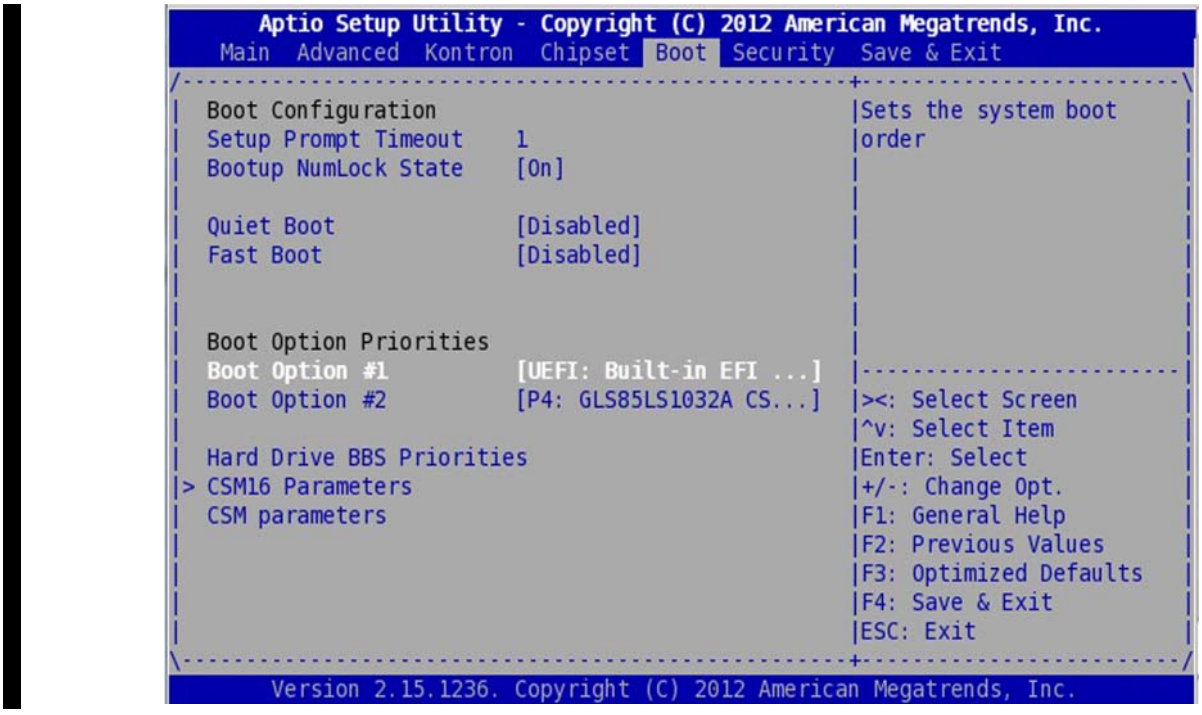
The following explains how to configure and execute the PBIT by a command line.

> 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:

```
VX304x> kdiag version
PBIT VERSION 1.7 ID13274
```

> Launch the PBIT manually for verification:

```
VX304x> 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_dmi" (fast,simple)  PASSED
PBIT "tpm" (fast,simple)  PASSED
PBIT "pcie_vpx_sw" (fast,simple)   PCIe Link is not UP behind B3/D9/F0 PASSED
PBIT "serial" (fast,simple)  PASSED
PBIT "rtc" (fast,simple)  PASSED
PBIT "sysflash" (fast,simple)  PASSED
PBIT "cpld" (fast,simple)  PASSED
PBIT "fnvram" (fast,simple)  PASSED
PBIT "ether_loop0" (fast,simple)  PASSED
PBIT "ether_loop1" (fast,simple)  PASSED
PBIT "ether_loop2" (fast,simple)  PASSED
PBIT "etherPM_loop" (fast,simple)   MOD-1BT NOT EQUIPPED 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 "xmc_check" (fast,simple)   XMC NOT EQUIPPED PASSED
PBIT "usb1_controller" (fast,simple)  PASSED
PBIT "usb2_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 32 “PBIT System Learn Test” for more details).

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

```
VX304x> kdiag system_learn

   Seg  Bus  Dev  Func
   ---  ---  ---  ----
    00   00   00    00 ==> Bridge Device - Host/PCI bridge
              Vendor 8086 Device 0154 Prog Interface 0
    00   00   01    00 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 0151 Prog Interface 0
```

```
00   00   01    01 ==> Bridge Device - PCI/PCI bridge
          Vendor 8086 Device 0155 Prog Interface 0
00   00   02    00 ==> Display Controller - VGA/8514 controller
          Vendor 8086 Device 0166 Prog Interface 0
00   00   14    00 ==> Serial Bus Controllers - USB
          Vendor 8086 Device 1E31 Prog Interface 30
00   00   19    00 ==> Network Controller - Ethernet controller
          Vendor 8086 Device 1502 Prog Interface 0
00   00   1A    00 ==> Serial Bus Controllers - USB
          Vendor 8086 Device 1E2D Prog Interface 20
00   00   1D    00 ==> Serial Bus Controllers - USB
          Vendor 8086 Device 1E26 Prog Interface 20
00   00   1F    00 ==> Bridge Device - PCI/ISA bridge
          Vendor 8086 Device 1E55 Prog Interface 0
00   00   1F    02 ==> Mass Storage Controller - UNDEFINED
          Vendor 8086 Device 1E03 Prog Interface 1
00   00   1F    03 ==> Serial Bus Controllers - System Management Bus
          Vendor 8086 Device 1E22 Prog Interface 0
00   00   1F    06 ==> Data Acquisition & Signal Processing Controllers - Ot
          Vendor 8086 Device 1E24 Prog Interface 0
00   01   00    00 ==> Network Controller - Ethernet controller
          Vendor 8086 Device 10FC Prog Interface 0
00   01   00    01 ==> Network Controller - Ethernet controller
          Vendor 8086 Device 10FC Prog Interface 0
00   02   00    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 10B5 Device 8725 Prog Interface 0
00   02   00    01 ==> Base System Peripherals - Other system peripheral
          Vendor 10B5 Device 87D0 Prog Interface 0
00   02   00    02 ==> Base System Peripherals - Other system peripheral
          Vendor 10B5 Device 87D0 Prog Interface 0
00   02   00    03 ==> Base System Peripherals - Other system peripheral
          Vendor 10B5 Device 87D0 Prog Interface 0
00   02   00    04 ==> Base System Peripherals - Other system peripheral
          Vendor 10B5 Device 87D0 Prog Interface 0
00   03   00    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 10B5 Device 8725 Prog Interface 0
00   03   08    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 10B5 Device 8725 Prog Interface 0
00   03   09    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 10B5 Device 8725 Prog Interface 0
00   06   00    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 111D Device 808C Prog Interface 0
00   07   02    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 111D Device 808C Prog Interface 0
00   07   04    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 111D Device 808C Prog Interface 0
00   07   06    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 111D Device 808C Prog Interface 0
00   07   08    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 111D Device 808C Prog Interface 0
00   07   0C    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 111D Device 808C Prog Interface 0
00   07   10    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 111D Device 808C Prog Interface 0
00   07   14    00 ==> Bridge Device - PCI/PCI bridge
          Vendor 111D Device 808C Prog Interface 0
```

```
CPLD Version 0x5
DRAM size 4 GB
System Controller
Geographical Address 1
XMC not present
SATA PORT 3 WDC WD800JD-75 (80.0GB)
SATA PORT 4 GLS85LS1032A C (32.0GB)
Device detected on SMBUS0, address = 0x30
Device detected on SMBUS0, address = 0xEC
BIOS Setup Checksum : 117679
BIOS Version : ID13274
ETH0 connected device speed 1000Mb/s
ETH1 connected device speed 1000Mb/s
      1 Drive, 1 Keyboard, 3 Hubs
MassStorage Device name (0) :  0.00
USB Port 1 connected
USB Port 2 connected

Number of System Test Elements detected : 91
DRAM area [ 0x15E7FC0  0x15E8C20 ] will be stored in EEPROM
Storing system infos...

Storing system configuration...
```

> Check the PBIT results:

```
 VX304x> kdiag stat
 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_dmi (fast,simple)
 PASSED : tpm (fast,simple)
 PASSED : pcie_vpx_sw (fast,simple)
 PASSED : serial (fast,simple)
 PASSED : rtc (fast,simple)
 PASSED : sysflash (fast,simple)
 PASSED: cpld (fast,simple)
 PASSED : fnvram (fast,simple)
 PASSED : ether_loop0 (fast,simple)
 PASSED : ether_loop1 (fast,simple)
 PASSED : ether_loop2 (fast,simple)
 PASSED : etherPM_loop (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 : xmc_check (fast,simple)
 PASSED : usb1_controller (fast,simple)
 PASSED : usb2_controller (fast,simple)
 PASSED : xHCI_controller (fast,simple)
 PASSED : system (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**”.

```
VX304x> set bootcmd "kdiag run"
```

The delay before executing the **bootcmd** is given by the variable **bootdelay** which is expressed in seconds.

Default value is “1”. Value “0” is possible.

```
VX304x> set bootdelay 1
```

> Verify:

```
VX304x> set
    bootcmd   : kdiag run
    bootdelay : 1
```

> Then reset the system:

```
VX304x> 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**”:

```
VX304x> 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:

```
VX304x> 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:

```
VX304x> kdiag 16
 serial (16) - Checks the serial line COM2
   capabilities : fast,simple/complex
   run mode 1   : fast,simple
```

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 VX304x. No signal on any connector will be modified during the default tes t execution.

The command “`kdiag`” displays the default tests list to run:

```
VX304x> 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_dmi (10) - Checks Core & DMI bus
   capabilities : fast,simple/complex,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 pciExpress backplane switch
   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
 fnvram (40) - Checks F-NVRAM device.
    capabilities : fast,simple/complex,allresets
    run mode 1   : fast,simple,allresets
 ether_loop0 (55) - Checks 82579 (Lewisville) Gigabit Ethernet in Loopback mode
    capabilities : fast,simple/complex,allresets
     run mode 1   : fast,simple,allresets
 ether_loop1 (56) - Checks 82599 10GEth Rear Interface 1 in Loopback mode
    capabilities : fast,simple/complex,allresets
    run mode 1   : fast,simple,allresets
ether_loop2 (57) - Checks 82599 10GEth Rear Interface 2 in Loopback mode
    capabilities : fast,simple/complex,allresets
    run mode 1   : fast,simple,allresets
 etherPM_loop (58) - Checks MOD-1BT Gigabit Rear Interface 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,allresets
    run mode 1   : fast,simple,allresets
 sata0_controler (68) - Checks sata0 controller
    capabilities : fast,simple/complex,allresets
    run mode 1   : fast,simple,allresets
 sata1_controler (69) - Check sata1 controller
    capabilities : fast,simple/complex,allresets
    run mode 1   : fast,simple,allresets
 vpd (70) - Checks VPD data required for board operation.
    capabilities : fast,simple,allresets
    run mode 1   : fast,simple,allresets
 eeprom (71) - Checks User EEPROM (0xA2)
    capabilities : fast,simple/complex,allresets
    run mode 1   : fast,simple,allresets
 xmc_check (76) - Checks XMC path & slot
    capabilities : fast,simple/complex,allresets
    run mode 1   : fast,simple,allresets
 usb1_controller (86) - Check first PCH usb controller
    capabilities : fast,simple,allresets
    run mode 1   : fast,simple,allresets
 usb2_controller (87) - Check second PCH usb controller
    capabilities : fast,simple,allresets
    run mode 1   : fast,simple,allresets
 xHCI_controller (88) - Check xHCI PCH usb 3.0 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:

```
VX304x> 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 % 32))
   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 is "UP" on 82579 (Lewisville) Gigabit Network.
   capabilities : fast,simple/complex,allresets
 ether_link1 (51) - Checks the link status is "UP" on 82599 10GEth Rear 1
   capabilities : fast,simple/complex,allresets
 ether_link2 (52) - Checks the link status is "UP" on 82599 10GEth Rear 2
   capabilities : fast,simple/complex,allresets
 etherPM_link (53) - Checks the link status is "UP" on MOD-1BT Gigabit Rear Interface
   capabilities : fast,simple/complex,allresets
 sata0_dev_see (63) - Checks if a SATA0 disk on rear P1 is present
```

```
    capabilities : fast,simple/complex,allresets
 sata1_dev_see (64) - Checks if a SATA1 disk on rear P1 is present
   capabilities : fast,simple/complex,allresets
 sata2_dev_see (65) - Checks if a SATA2 disk on rear P2 is present
   capabilities : fast,simple/complex,allresets
 sata3_dev_see (66) - Checks if a SATA3 disk on rear P2 is present
   capabilities : fast,simple/complex,allresets
 sata4_dev_see (67) - Checks if a SATA SDD on Board device is present
   capabilities : fast,simple/complex,allresets
 xmc_see (77) - Checks if device is present XMC slot
   capabilities : fast,simple/complex,allresets
 usb0_dev_see (80) - Check if a USB0 device on front panel is present
   capabilities : fast,simple/complex,allresets
 usb1_dev_see (81) - Check if a USB device on USB 3.0 port is present
   capabilities : fast,simple/complex,allresets
 usb2_dev_see (82) - Check if a USB2 device on rear P1 is present
   capabilities : fast,simple/complex,allresets
 usb3_dev_see (83) - Check if a USB3 device on rear P1 is present
   capabilities : fast,simple/complex,allresets
 usb4_dev_see (84) - Check if a USB4 device on rear P2 is present
   capabilities : fast,simple/complex,allresets
 usb5_dev_see (85) - Check if a USB5 device on rear P2 is present
   capabilities : fast,simple/complex,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 9) runs in about 20 seconds for a 4 GB DRAM board and 30 seconds for a 8 GB DRAM board without the system test configured.

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 appear in red.

Test name	Test ID number	Execution Time
mem_data (fast,simple)	1	500 ms
mem_addr (fast,simple)	2	5s
mem_bitflip (slow,simple)	3	4.7s
mem_addrpat (slow,simple)	4	4s
mem_addrpat2 (slow,simple)	5	4s
mem_pattern1 (slow,simple)	6	4s
mem_pattern2 (slow,simple)	7	4s
mem_pattern3 (slow,simple)	8	4s
mem_pattern4 (slow,simple)	9	4s
core_dmi (fast,simple)	10	<300 ms
tpm (fast,simple)	10	<300 ms
pcie_vpx_sw (fast,simple)	13	<300 ms
serial (fast,simple)	16	<300 ms
rtc (fast,simple)	20	<300 ms
sysflash (fast,simple)	22	<300 ms
cpld (fast,simple)	24	1.7 s
fnvram (fast,simple)	40	350 ms
ether_link0 (fast,simple)	50	<300 ms if OK but 3s if it failed
ether_link1 (fast,simple)	51	<300 ms if OK but 3s if it failed
ether_link2 (fast,simple)	52	<300 ms if OK but 3s if it failed
ether_loop0 (fast,simple)	55	1.2s
ether_loop1 (fast,simple)	56	3.4s
ether_loop2 (fast,simple)	57	3.4s
hwmon (fast,simple)	61	500 ms
sata0_dev_see (fast,simple)	63	<300 ms
Sata1_dev_see (fast,simple)	64	<300 ms
Sata2_dev_see (fast,simple)	65	<300 ms
Sata3_dev_see (fast,simple)	66	<300 ms
Sata4_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	<300 ms
eeprom (fast,simple)	71	400 ms

Test name	Test ID number	Execution Time
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
usb4_dev_see (fast,simple)	84	<300 ms
usb5_dev_see (fast,simple)	85	<300 ms
usb1_controller (fast,simple)	86	<300 ms
usb2_controller (fast,simple)	87	<300 ms
XHCI_controller (fast,simple)	88	<300 ms
system (fast,simple)*	89	1s to 10s. Configurable. About 1s if all tested Ethernet links are connected. 3s must be added for each Ethernet link unconnected.

* The execution time for the system test depends on the configuration and if the test is successful or not. In case of Ethernet Link not present on all interfaces the execution time may reach 9s (3s per Ethernet link not connected). To save execution time it is advised to remove test on unconnected Ethernet link with command "`kdiag system_edit`" .

Chapter 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.

Note: the command formats are

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

```
VX304x> help kdiag
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.


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 [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): clear PBIT status of peer CPU side

Restore default PBIT configuration :

  kdiag [oside] default

    oside (VX6060 board only): restore default PBIT for 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 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

    oside (VX6060 board only): Configure PBIT(s) for peer CPU side

Toggle PBIT running log information:

  kdiag silent


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.

```
VX304x> kdiag
PBITs configured to run from command line :
 mem_data (1) - Checks Memory/ECC data lines
   capabilities : fast,simple
   run mode 1   : fast,simple
 mem_addr (2) - Checks Memory/ECC address lines
   capabilities : fast,simple
   run mode 1   : fast,simple
 mem_pattern1 (6) - Checks Memory/ECC using pattern 0xFFFFFFFF
   capabilities : slow/fast,simple
   run mode 1   : slow,simple
 mem_pattern2 (7) - Checks Memory/ECC using pattern 0x55555555
   capabilities : slow/fast,simple
   run mode 1   : slow,simple
(...)
Other PBITs available but not yet configured :
 mem_bitflip (3) - Checks Mem/ECC using bit-flip pattern ((1 << (offset % 32))
   capabilities : slow/fast,simple
 mem_addrpat (4) - Checks Memory/ECC using address pattern (offset)
   capabilities : slow/fast,simple
 mem_addrpat2 (5) - Checks Memory/ECC using address pattern (~offset)
   capabilities : slow/fast,simple
 smbus0 (26) - Check SMBUS0 between PLD and backplane
   capabilities : fast,simple/complex
 smbus1 (27) - Check SMBUS1 between PLD and backplane
   capabilities : fast,simple/complex
 ether_link0 (50) - Checks the link status is "UP" on 82579 (Lewisville) Gigabit Network.
   capabilities : fast,simple/complex
 ether_link1 (51) - Checks the link status is "UP" on 82580 GigaEth Rear 1
   capabilities : fast,simple/complex
(...)

Use ´help kdiag´ to get more info.
VX304x>
```

2.3 Execute the PBIT from the Command Line

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

```
VX304x> 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_dmi" (fast,simple)  PASSED
PBIT "tpm" (fast,simple)  PASSED
PBIT "pcie_vpx_sw" (fast,simple)   PCIe Link is not UP behind B3/D9/F0 PASSED
PBIT "serial" (fast,simple)  PASSED
PBIT "rtc" (fast,simple)  PASSED
PBIT "sysflash" (fast,simple)  PASSED
PBIT "cpld" (fast,simple)  PASSED
PBIT "fnvram" (fast,simple)  PASSED
PBIT "ether_loop0" (fast,simple)  PASSED
PBIT "ether_loop1" (fast,simple)  PASSED
PBIT "ether_loop2" (fast,simple)  PASSED
PBIT "etherPM_loop" (fast,simple)   MOD-1BT NOT EQUIPPED 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 "xmc_check" (fast,simple)   XMC NOT EQUIPPED PASSED
PBIT "usb1_controller" (fast,simple)  PASSED
PBIT "usb2_controller" (fast,simple)  PASSED
PBIT "xHCI_controller" (fast,simple)  PASSED
PBIT "system" (fast,simple)
SYSTEM INFOS NOT SAVED
PASSED
```

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

```
VX304x> kdiag run <PBIT number | PBIT name ...>
```

> For example:

```
VX304x> kdiag run usb2_controller sysflash
PBIT "usb2_controller" (fast,simple)  PASSED
PBIT "sysflash" (fast,simple)  PASSED
```

That is equivalent to:

```
VX304x> kdiag run 87 22
PBIT "usb2_controller" (fast,simple)  PASSED
PBIT "sysflash" (fast,simple)  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:

```
VX304x > kdiag run loop <count>
```

with `<count>` being the number of loop to execute.

To run a single test in loop mode, enter:

```
VX304x> kdiag run loop <count> <PBIT number | PBIT name ...>
```

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

```
VX304x> kdiag run loop 10 55
```

It is equivalent to enter:

```
VX304x> kdiag run loop 10 ether_loop0
```

2.5 Get the PBIT Results

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

```
VX304x> kdiag stat
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_dmi (fast,simple)
 PASSED : tpm (fast,simple)
 PASSED : pcie_vpx_sw (fast,simple)
 PASSED : serial (fast,simple)
 PASSED : rtc (fast,simple)
 PASSED : sysflash (fast,simple)
 PASSED: cpld (fast,simple)
 PASSED : fnvram (fast,simple)
 PASSED : ether_loop0 (fast,simple)
 PASSED : ether_loop1 (fast,simple)
 PASSED : ether_loop2 (fast,simple)
 PASSED : etherPM_loop (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 : xmc_check (fast,simple)
 PASSED : usb1_controller (fast,simple)
 PASSED : usb2_controller (fast,simple)
 PASSED : xHCI_controller (fast,simple)
 PASSED : system (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:

```
VX304x> kdiag clrstat
```

> To clear all the PBIT history including the “FAILED ONCE” flags, enter:

```
VX304x> 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 VX304x 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:

```
VX304x> 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:

```
VX304x> kdiag cfg default 16
```

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

```
VX304x> kdiag cfg promptonfail 16
```

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

```
VX304x> kdiag cfg  complex,promptonfail 16
```

Execute the command “kdiag” or “kdiag 16” to check the configuration:

```
VX304x> kdiag 16
 serial (16) - Checks the serial line COM2
   capabilities : fast,simple/complex
   run mode 1   : fast,complex,promptonfail
```

2.7.3 Removing a Test from the Current Run List

For example, to remove test number 16, enter:

```
VX304x> kdiag cfg delete 16
```

Verify:

```
VX304x> kdiag 16
PBIT "16" is not configured to run from command line
```

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

```
VX304x> kdiag deleteall
```

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

```
VX304x> 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 23) to the all current run list test.

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

```
VX304x> 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_dmi (10) - Checks Core & DMI bus
--> tpm (11) - Checks TPM access
--> pcie_vpx_sw (13) - Checks the pciExpress backplane switch
--> 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
--> fnvram (40) - Checks F-NVRAM device.
--> ether_loop0 (55) - Checks 82579 (Lewisville) Gigabit Ethernet in Loopback mode
--> ether_loop1 (56) - Checks 82599 10GEth Rear Interface 1 in Loopback mode
--> ether_loop2 (57) - Checks 82599 10GEth Rear Interface 2 in Loopback mode
--> etherPM_loop (58) - Checks MOD-1BT Gigabit Rear Interface in Loopback mode
--> hwmon (61) - Checks hardware monitoring, temperature and voltage sensors
--> sata0_controler (68) - Checks sata0 controller
--> sata1_controler (69) - Check sata1 controller
--> vpd (70) - Checks VPD data required for board operation.
--> eeprom (71) - Checks User EEPROM (0xA2)
--> xmc_check (76) - Checks XMC path & slot
--> usb1_controller (86) - Check first PCH usb controller
--> usb2_controller (87) - Check second PCH usb controller
--> xHCI_controller (88) - Check xHCI PCH usb 3.0 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:

```
VX304x>  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_dmi (10) - Checks Core & DMI bus
--> tpm (11) - Checks TPM access
--> pcie_vpx_sw (13) - Checks the pciExpress backplane switch
--> 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
--> fnvram (40) - Checks F-NVRAM device.
--> ether_loop0 (55) - Checks 82579 (Lewisville) Gigabit Ethernet in Loopback mode
--> ether_loop1 (56) - Checks 82599 10GEth Rear Interface 1 in Loopback mode
--> ether_loop2 (57) - Checks 82599 10GEth Rear Interface 2 in Loopback mode
--> etherPM_loop (58) - Checks MOD-1BT Gigabit Rear Interface in Loopback mode
--> hwmon (61) - Checks hardware monitoring, temperature and voltage sensors
--> sata0_controler (68) - Checks sata0 controller
--> sata1_controler (69) - Check sata1 controller
--> vpd (70) - Checks VPD data required for board operation.
--> eeprom (71) - Checks User EEPROM (0xA2)
--> xmc_check (76) - Checks XMC path & slot
--> usb1_controller (86) - Check first PCH usb controller
--> usb2_controller (87) - Check second PCH usb controller
--> xHCI_controller (88) - Check xHCI PCH usb 3.0 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:

```
VX304x> 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_dmi (10) - Checks Core & DMI bus
   capabilities : fast,simple/complex,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 pciExpress backplane switch
   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
 fnvram (40) - Checks F-NVRAM device.
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
 ether_loop0 (55) - Checks 82579 (Lewisville) Gigabit Ethernet in Loopback mode
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
 ether_loop1 (56) - Checks 82599 10GEth Rear Interface 1 in Loopback mode
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
```

```
ether_loop2 (57) - Checks 82599 10GEth Rear Interface 2 in Loopback mode
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,promptonfail,poweronreset
etherPM_loop (58) - Checks MOD-1BT Gigabit Rear Interface 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,allresets
  run mode 1   : fast,simple,promptonfail,poweronreset
sata0_controler (68) - Checks sata0 controller
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,promptonfail,poweronreset
sata1_controler (69) - Check sata1 controller
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,promptonfail,poweronreset
vpd (70) - Checks VPD data required for board operation.
  capabilities : fast,simple,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
xmc_check (76) - Checks XMC path & slot
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,promptonfail,poweronreset
usb1_controller (86) - Check first PCH usb controller
  capabilities : fast,simple,allresets
  run mode 1   : fast,simple,promptonfail,poweronreset
usb2_controller (87) - Check second PCH usb controller
  capabilities : fast,simple,allresets
  run mode 1   : fast,simple,promptonfail,poweronreset
xHCI_controller (88) - Check xHCI PCH usb 3.0 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:

```
  VX304x> kdiag 13
   pcie_vpx_sw (13) - Checks the pciExpress backplane 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 Quickly configure all tests in specific mode

The fastest method to configure all PBIT in a specific mode (like `promptonfail`, `poweronreset` …) is to delete all tests with `kdiag deleteall` and add all test with `kdiag cfg` + chosen mode to set.

Example to configure all tests in `poweronreset` and `promptonfail` mode

```
VX304x> kdiag deleteall

VX304x> kdiag cfg promptonfail,poweronreset
Configuration of all PBITs in default mode only
List of test to configure
--> mem_data (1) - Checks Memory/ECC data lines
--> mem_addr (2) - Checks Memory/ECC address lines
--> mem_bitflip (3) - Checks Mem/ECC using bit-flip pattern ((1 << (offset % 32))
--> mem_addrpat (4) - Checks Memory/ECC using address pattern (offset)
--> mem_addrpat2 (5) - Checks Memory/ECC using address pattern (~offset)
--> 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_dmi (10) - Checks Core & DMI bus
--> tpm (11) - Checks TPM access
--> pcie_vpx_sw (13) - Checks the pciExpress backplane switch
--> 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
--> smbus0 (26) - Check SMBUS0 between PLD and backplane
--> smbus1 (27) - Check SMBUS1 between PLD and backplane
--> fnvram (40) - Checks F-NVRAM device.
--> ether_link0 (50) - Checks the link status is "UP" on 82579 (Lewisville) Gigabit
Network.
--> ether_link1 (51) - Checks the link status is "UP" on 82599 10GEth Rear 1
--> ether_link2 (52) - Checks the link status is "UP" on 82599 10GEth Rear 2
--> ether_loop0 (55) - Checks 82579 (Lewisville) Gigabit Ethernet in Loopback mode
--> ether_loop1 (56) - Checks 82599 10GEth Rear Interface 1 in Loopback mode
--> ether_loop2 (57) - Checks 82599 10GEth Rear Interface 2 in Loopback mode
--> hwmon (61) - Checks hardware monitoring, temperature and voltage sensors
--> sata0_dev_see (63) - Checks if a SATA0 disk on rear P1 is present
--> sata1_dev_see (64) - Checks if a SATA1 disk on rear P1 is present
--> sata2_dev_see (65) - Checks if a SATA2 disk on rear P2 is present
--> sata3_dev_see (66) - Checks if a SATA3 disk on rear P2 is present
--> sata4_dev_see (67) - Checks if a SATA SDD on Board device is present
--> sata0_controler (68) - Checks sata0 controller
--> sata1_controler (69) - Check sata1 controller
--> vpd (70) - Checks VPD data required for board operation.
--> eeprom (71) - Checks User EEPROM (0xA2)
--> xmc_check (76) - Checks XMC path & slot
--> xmc_see (77) - Checks if device is present XMC slot
```

```
--> usb0_dev_see (80) - Check if a USB0 device on front panel is present
--> usb1_dev_see (81) - Check if a USB device on USB 3.0 port is present
--> usb2_dev_see (82) - Check if a USB2 device on rear P1 is present
--> usb3_dev_see (83) - Check if a USB3 device on rear P1 is present
--> usb4_dev_see (84) - Check if a USB4 device on rear P2 is present
--> usb5_dev_see (85) - Check if a USB5 device on rear P2 is present
--> usb1_controller (86) - Check first PCH usb controller
--> usb2_controller (87) - Check second PCH usb controller
--> xHCI_controller (88) - Check xHCI PCH usb 3.0 controller
--> system (89) - Checks system configuration SETUP,PCIe,SATA,USB,ETH stability
```

Verify for test 3:

```
VX304x> kdiag 3
 mem_bitflip (3) - Checks Mem/ECC using bit-flip pattern ((1 << (offset % 32))
   capabilities : slow/fast,simple,allresets
   run mode 1   : fast,simple,promptonfail,poweronreset
```

Then don't forget to remove unwanted test with command `kdiag cfg delete <test_name>` or `<test_number>`

2.7.6 Restore the Default Run List

To restore the default run tests list, enter:

```
   VX304x> kdiag default
```

2.8 Run the PBIT in Silent Mode

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

```
VX304x> kdiag silent
PBIT set in silent mode
```

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

```
VX304x> 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:

```
VX304x> kdiag version
PBIT VERSION 1.7 ID 13274
```

2.10 PBIT System Test

2.10.1 Recording the system configuration:

The command "`kdiag system_learn`" is used to record the current system configuration. It must be run when the system configuration is the correct one to be recorded.

It records:

- All the 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 (Port 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.
- SMBUS connected devices addresses on backplanes SMBUS0 and SMBUS1.

```
VX304x> kdiag learn system
   Seg  Bus  Dev  Func
   ---  ---  ---  ----
    00   00   00    00 ==> Bridge Device - Host/PCI bridge
              Vendor 8086 Device 0154 Prog Interface 0
    00   00   01    00 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 0151 Prog Interface 0
    00   00   01    01 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 0155 Prog Interface 0
    00   00   02    00 ==> Display Controller - VGA/8514 controller
              Vendor 8086 Device 0166 Prog Interface 0
    00   00   14    00 ==> Serial Bus Controllers - USB
              Vendor 8086 Device 1E31 Prog Interface 30
    00   00   19    00 ==> Network Controller - Ethernet controller
              Vendor 8086 Device 1502 Prog Interface 0
    00   00   1A    00 ==> Serial Bus Controllers - USB
              Vendor 8086 Device 1E2D Prog Interface 20
    00   00   1D    00 ==> Serial Bus Controllers - USB
              Vendor 8086 Device 1E26 Prog Interface 20
    00   00   1F    00 ==> Bridge Device - PCI/ISA bridge
              Vendor 8086 Device 1E55 Prog Interface 0
    00   00   1F    02 ==> Mass Storage Controller - UNDEFINED
              Vendor 8086 Device 1E03 Prog Interface 1
    00   00   1F    03 ==> Serial Bus Controllers - System Management Bus
              Vendor 8086 Device 1E22 Prog Interface 0
```

```
      00   00   1F    06 ==> Data Acquisition & Signal Processing Controllers - Ot
                Vendor 8086 Device 1E24 Prog Interface 0
      00   01   00    00 ==> Network Controller - Ethernet controller
                Vendor 8086 Device 10FC Prog Interface 0
      00   01   00    01 ==> Network Controller - Ethernet controller
                 Vendor 8086 Device 10FC Prog Interface 0
      00   02   00    00 ==> Bridge Device - PCI/PCI bridge
                Vendor 10B5 Device 8725 Prog Interface 0
      00   03   00    00 ==> Bridge Device - PCI/PCI bridge
                Vendor 10B5 Device 8725 Prog Interface 0
      00   03   08    00 ==> Bridge Device - PCI/PCI bridge
                Vendor 10B5 Device 8725 Prog Interface 0
      00   03   09    00 ==> Bridge Device - PCI/PCI bridge
                Vendor 10B5 Device 8725 Prog Interface 0
      00   06   00    00 ==> Bridge Device - PCI/PCI bridge
                Vendor 111D Device 808C Prog Interface 0
      00   07   02    00 ==> Bridge Device - PCI/PCI bridge
                Vendor 111D Device 808C Prog Interface 0
      00   07   04    00 ==> Bridge Device - PCI/PCI bridge
                Vendor 111D Device 808C Prog Interface 0
      00   07   06    00 ==> Bridge Device - PCI/PCI bridge
                Vendor 111D Device 808C Prog Interface 0
      00   07   08    00 ==> Bridge Device - PCI/PCI bridge
                Vendor 111D Device 808C Prog Interface 0
      00   07   0C    00 ==> Bridge Device - PCI/PCI bridge
                Vendor 111D Device 808C Prog Interface 0
      00   07   10    00 ==> Bridge Device - PCI/PCI bridge
                Vendor 111D Device 808C Prog Interface 0
      00   07   14    00 ==> Bridge Device - PCI/PCI bridge
                Vendor 111D Device 808C Prog Interface 0
CPLD Version 0x5
DRAM size 4 GB
System Controller
Geographical Address 1
XMC not present
SATA PORT 3 WDC WD800JD-75 (80.0GB)
SATA PORT 4 GLS85LS1032A C (32.0GB)
Device detected on SMBUS0, address = 0x30
Device detected on SMBUS0, address = 0xEE
BIOS Setup Checksum : 117663
BIOS Version : ID13274
      1 Drive, 1 Keyboard, 3 Hubs
MassStorage Device name (0) : UFD 3.0 Silicon-Pow
USB Port 1 connected

Number of System Test Elements detected : 91
DRAM area [ 0x15E7FC0  0x15E8C20 ] will be stored in EEPROM
Storing system infos...

Storing system configuration...
```

2.10.2 Testing the current system configuration:

The test named "system" is used to read the current system configuration and check it against the recorded configuration. The command used is "kdiag run system" (equivalent to "kdiag system_run"). By default, the system test prints errors only, and details level is NORMAL. The following is an example showing several detected failures:

The following is an example showing several detected failures:

```
VX304x> kdiag run system
PBIT "system" (fast,simple)
PCI...  CPLD... SATA...  SMBUS...  BIOS...  ETH...  USB...
ERR)PCI Bus:Dev:Func       (R)03:00:00              (D)02:00:01
ERR)PCI 03:00:00 DeviceID  (R)0x8725                (D)0x87D0
ERR)PCI 03:00:00 ClassCode (R)060400                (D)060400
ERR)PCI 03:00:00 Wdth/Spd  (R)Undefined             (D)Bandwidth not available
ERR)PCI Bus:Dev:Func       (R)03:08:00              (D)02:00:02
ERR)PCI 03:08:00 DeviceID  (R)0x8725                (D)0x87D0
ERR)PCI 03:08:00 ClassCode (R)060400                (D)060400
ERR)PCI 03:08:00 Wdth/Spd  (R)Undefined             (D)Bandwidth not available
ERR)PCI Bus:Dev:Func       (R)03:09:00              (D)02:00:03
ERR)PCI 03:09:00 DeviceID  (R)0x8725                (D)0x87D0
ERR)PCI 03:09:00 ClassCode (R)060400                (D)060400
ERR)PCI 03:09:00 LinkStat  (R)Link is UP            (D)LinkStat not available
ERR)PCI 03:09:00 Wdth/Spd  (R)x4 / 5GT/s            (D)Bandwidth not available
ERR)PCI Bus:Dev:Func       (R)06:00:00              (D)02:00:04
ERR)PCI 06:00:00 VendorID  (R)0x111D                (D)0x10B5
ERR)PCI 06:00:00 DeviceID  (R)0x808C                (D)0x87D0
ERR)PCI 06:00:00 ClassCode (R)060400                (D)060400
ERR)PCI 06:00:00 LinkStat  (R)Link is UP            (D)LinkStat not available
ERR)PCI 06:00:00 Wdth/Spd  (R)x4 / 5GT/s            (D)Bandwidth not available
ERR)PCI Bus:Dev:Func       (R)07:02:00              (D)03:00:00
ERR)PCI 07:02:00 VendorID  (R)0x111D                (D)0x10B5
ERR)PCI 07:02:00 DeviceID  (R)0x808C                (D)0x8725
```

```
ERR)PCI Bus:Dev:Func       (R)07:04:00              (D)03:08:00
ERR)PCI 07:04:00 VendorID  (R)0x111D                (D)0x10B5
ERR)PCI 07:04:00 DeviceID  (R)0x808C                (D)0x8725
ERR)PCI Bus:Dev:Func       (R)07:06:00              (D)03:09:00
ERR)PCI 07:06:00 VendorID  (R)0x111D                (D)0x10B5
ERR)PCI 07:06:00 DeviceID  (R)0x808C                (D)0x8725
ERR)PCI 07:06:00 LinkStat  (R)Link not UP           (D)Link is UP
ERR)PCI 07:06:00 Wdth/Spd  (R)Undefined             (D)x4 / 5GT/s
ERR)PCI Bus:Dev:Func       (R)07:08:00              (D)06:00:00
ERR)PCI 07:08:00 LinkStat  (R)Link not UP           (D)Link is UP
ERR)PCI 07:08:00 Wdth/Spd  (R)Undefined             (D)x4 / 5GT/s
ERR)PCI Bus:Dev:Func       (R)07:0C:00              (D)07:02:00
ERR)PCI Bus:Dev:Func       (R)07:10:00              (D)07:04:00
ERR)PCI Bus:Dev:Func       (R)07:14:00              (D)07:06:00
ERR)PCI Bus:Dev:Func       (R)No Device             (D)07:08:00
ERR)PCI 00:00:00 VendorID  (R)No Device             (D)0x111D
ERR)PCI 00:00:00 DeviceID  (R)No Device             (D)0x808C
ERR)PCI 00:00:00 ClassCode (R)No Device             (D)No Device
ERR)PCI 00:00:00 Wdth/Spd  (R)Bandwidth not available(D)Undefined
ERR)PCI Bus:Dev:Func       (R)No Device             (D)07:0C:00
ERR)PCI 00:00:00 VendorID  (R)No Device             (D)0x111D
ERR)PCI 00:00:00 DeviceID  (R)No Device             (D)0x808C
ERR)PCI 00:00:00 ClassCode (R)No Device             (D)No Device
ERR)PCI 00:00:00 Wdth/Spd  (R)Bandwidth not available(D)Undefined
ERR)PCI Bus:Dev:Func       (R)No Device             (D)07:10:00
ERR)PCI 00:00:00 VendorID  (R)No Device             (D)0x111D
ERR)PCI 00:00:00 DeviceID  (R)No Device             (D)0x808C
ERR)PCI 00:00:00 ClassCode (R)No Device             (D)No Device
ERR)PCI 00:00:00 Wdth/Spd  (R)Bandwidth not available(D)Undefined
ERR)PCI Bus:Dev:Func       (R)No Device             (D)07:14:00
ERR)PCI 00:00:00 VendorID  (R)No Device             (D)0x111D
ERR)PCI 00:00:00 DeviceID  (R)No Device             (D)0x808C
ERR)PCI 00:00:00 ClassCode (R)No Device             (D)No Device
ERR)PCI 00:00:00 Wdth/Spd  (R)Bandwidth not available(D)Undefined
ERR)SATA Port 3             (R)WDC WD800JD-75 (80.0GB)(D)Not Connected (0.0GB)
ERR)BIOS Checksum          (R)117663                (D)118175
ERR)ETH Port 0              (R)Not Connected         (D)Link UP speed 1000 Mb/s
ERR)USB MassStorage(0)     (R)UFD 3.0 Silicon-Pow   (D) 0.00
ERR)USB Port 1             (R)Connected             (D)Not Connected
ERR)USB Port 5             (R)Not Connected         (D)Connected
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 configuring a specific level of debug when the system test is executed. To enter system edit menu enter:

```
VX304x>kdiag  edit system
```

Equivalent to:

```
VX304x>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. To 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 `system_edit` session ignoring the System Controller status information from CPLD, the SATA port 4, all the SMBUS0 and SMBUS1 devices, all information for PCI device 1, the ClassCode information for PCI device 16, all bios information and the Ethernet interfaces 2 and 3:

```
VX304x>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>>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           0x5
1. CPLD Dram Size         4GB
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
0. CPLD Version           0x5
1. CPLD Dram Size         4GB
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            Not Connected (0.0GB)
1. SATA Port 1            Not Connected (0.0GB)
2. SATA Port 2            Not Connected (0.0GB)
3. SATA Port 3            WDC WD800JD-75 (80.0GB) [FAILED ONCE] [LAST FAILED]
4. SATA Port 4            GLS85LS1032A C (32.0GB)
*-SystemEdit-SATA>>3
*-SystemEdit-SATA-Port 3>>i
Ignore flag has been set
*-SystemEdit-SATA-Port 3>>
*-SystemEdit-SATA>>p
0. SATA Port 0            Not Connected (0.0GB)
1. SATA Port 1            Not Connected (0.0GB)
2. SATA Port 2            Not Connected (0.0GB)
3. SATA Port 3            WDC WD800JD-75 (80.0GB)(i) [FAILED ONCE] [LAST FAILED]
4. SATA Port 4            GLS85LS1032A C (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        0xEE
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 (i) 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
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,..,39 : 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 00:01:01 Bridge Device/PCI/PCI bridge
3.  PCI 00:02:00 Display Controller/VGA/8514 controller
4.  PCI 00:14:00 Serial Bus Controllers/USB
5.  PCI 00:19:00 Network Controller/Ethernet controller
6.  PCI 00:1A:00 Serial Bus Controllers/USB
7.  PCI 00:1D:00 Serial Bus Controllers/USB
8.  PCI 00:1F:00 Bridge Device/PCI/ISA bridge
9.  PCI 00:1F:02 Mass Storage Controller/UNDEFINED
10. PCI 00:1F:03 Serial Bus Controllers/System Management Bus
11. PCI 00:1F:06 Data Acquisition & Signal Processing Controllers/Other DAQ & SP
controllers
12. PCI 01:00:00 Network Controller/Ethernet controller
13. PCI 01:00:01 Network Controller/Ethernet controller
```

```
14. PCI 02:00:00 Bridge Device/PCI/PCI bridge
15. PCI 03:00:00 Bridge Device/PCI/PCI bridge
16. PCI 03:08:00 Bridge Device/PCI/PCI bridge
17. PCI 03:09:00 Bridge Device/PCI/PCI bridge
18. PCI 06:00:00 Bridge Device/PCI/PCI bridge
19. PCI 07:02:00 Bridge Device/PCI/PCI bridge
20. PCI 07:04:00 Bridge Device/PCI/PCI bridge [FAILED ONCE] [LAST FAILED]
21. PCI 07:06:00 Bridge Device/PCI/PCI bridge
22. PCI 07:08:00 Bridge Device/PCI/PCI bridge [FAILED ONCE] [LAST FAILED]
23. PCI 07:0C:00 Bridge Device/PCI/PCI bridge
24. PCI 07:10:00 Bridge Device/PCI/PCI bridge
25. PCI 07:14:00 Bridge Device/PCI/PCI bridge
26. PCI No Device No Device/No Device
27. PCI No Device No Device/No Device
28. PCI No Device No Device/No Device
29. PCI No Device No Device/No Device
30. PCI No Device No Device/No Device
31. PCI No Device No Device/No Device [FAILED ONCE] [LAST FAILED]
32. PCI No Device No Device/No Device
33. PCI No Device No Device/No Device
34. PCI No Device No Device/No Device
35. PCI No Device No Device/No Device
36. PCI No Device No Device/No Device [FAILED ONCE] [LAST FAILED]
37. PCI No Device No Device/No Device
38. PCI No Device No Device/No Device
39. PCI No Device No Device/No Device
*-SystemEdit-PCI>>           <<  KONTRON SYSTEM PBIT : EDIT MODE  >>

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

*-SystemEdit>>pci
*-SystemEdit-PCI>>17
*-SystemEdit-PCI-03:09:00>>ignoreall
PCI 03:09:00 entirely ignored
*-SystemEdit-PCI-03:09:00>>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-03:09:00>>
*-SystemEdit-PCI>>           <<  KONTRON SYSTEM PBIT : EDIT MODE  >>
```

```
Edit by feature, choose:PCI (*) CPLD (*) SATA (*) ETH USB SMBUS (i) 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          117663                  [FAILED ONCE] [LAST FAILED])
1. BIOS Version           ID13274
*-SystemEdit-BIOS>>0
*-SystemEdit-BIOS-Checksum>>i
Ignore flag has been set
*-SystemEdit-BIOS-Checksum>>
*-SystemEdit-BIOS>>p
0. BIOS Checksum          117663                 (i) [FAILED ONCE] [LAST FAILED])
1. BIOS Version           ID13274
*-SystemEdit-BIOS>>           <<  KONTRON SYSTEM PBIT : EDIT MODE  >>

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

*-SystemEdit>>eth
*-SystemEdit-ETH>>p
0. ETH Port 0             Not Connected           [FAILED ONCE] [LAST FAILED]
1. ETH Port 1             Not Connected
2. ETH Port 2             Not Connected
*-SystemEdit-ETH>>0
*-SystemEdit-ETH-Port 0>>i
Ignore flag has been set
*-SystemEdit-ETH-Port 0>>
*-SystemEdit-ETH>>p
0. ETH Port 0             Not Connected          (i) [FAILED ONCE] [LAST FAILED]
1. ETH Port 1             Not Connected
2. ETH Port 2             Not Connected
*-SystemEdit-ETH>>           <<  KONTRON SYSTEM PBIT : EDIT MODE  >>

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

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

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.

> 3 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.

> 3 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:

```
VX304x> 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.

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

```
VX304x> kdiag system_edit
          <<  KONTRON SYSTEM PBIT : EDIT MODE  >>

Edit by feature, choose:PCI (*) CPLD (*) SATA (*) ETH (*) USB SMBUS (i) 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: SYNTHETHIC
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: SYNTHETHIC
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>>d
Details Lvl DETAILED set
*-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>>q           <<  KONTRON SYSTEM PBIT : EDIT MODE  >>
```

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

*-SystemEdit>>q
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 (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:

```
VX304x>kdiag  system_edit

           <<  KONTRON SYSTEM PBIT : EDIT MODE  >>

Edit by feature, choose:PCI (*) CPLD (*) SATA (*) ETH (*) USB SMBUS (i) 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>>
```

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

2.10.5 Clearing the system configuration

A command is avavailable to clear the system configuration entirely.To clear the entire configuration, enter:

```
VX304x>kdiag  clear system
```

Equivalent to:

```
VX304x>kdiag  system_clear
```

Then, if a system test is run, no test will be performed and the following will be displayed:

```
SYSTEM INFOS NOT SAVED
```

The system test result will be PASSED.

1. To summarize, before running the system test, the specific commands must be performed in this order: “`kdiag learn system`” (or “`kdiag system_learn`”), “`kdiag edit system`” (or “`kdiag system_edit`”) depending the chosen test granularity then “`kdiag run system`” (or “`kdiag system_run`”).
   Clearing the entire system configuration will completely bypass the system test.
   Learning a system configuration will overwrite a previous recorded configuration.
2. After updating the PBIT or BIOS release, it is recommended to clear and learn the system configuration again.

Chapter 3 - PBIT and OS Interfaces

The PBIT synthetic result and the PBIT detailed results are accessible under the Operating System delivered with the VX304x board.

The following gives you an overview of the facilities available under Linux, VxWorks and Windows OS Board Support Packages.

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
Reg 0x0 - CPLD_ID = 0x01
            CPLD_ID=0x0
            CPLD_Debug=0x0
            CPLD_Version=0x1
Reg 0x1 - PCB_ID = 0x35
Reg 0x2 - FIRM_PWON = 0x00
Reg 0x3 - PWON_STATUS = 0x00
Reg 0x4 - PWR_RST_CONFIG = 0x23
            PWRON_MODE=0x0
            PLTRST_to_PERST3U_INHIB=0x0
            Alarm_inhib=0x1
            PLTRST_to_PERST6U_INHIB=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
VX304x  detected
area = 2, arch = 2
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
ether_loop0: PASSED
ether_loop1: PASSED
ether_loop2: PASSED
system: PASSED (FAILED ONCE)
PASSED : 10
FAILED : 0
NOT RUN : 0
TOTAL : 10
```

3.2 VxWorks

3.2.1 VxWorks Synthetic PBIT Result

The synthetic PBIT result stored in the CPLD register 0x2 can be accessed with a memory I/O access at address I/O 0x802.

The command "`sysInByte(0x802)`" will return the synthetic PBIT result.

```
-> sysInByte(0x802)
value = 1 = 0x1
```

3.2.2 VxWorks Detailed PBIT Result

The detailed PBIT results (if implemented) is accessible using the "`pbitdisplay()`" call.

```
-> pbitdisplay
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_dmi: PASSED
            tpm: PASSED
    pcie_vpx_sw: PASSED
         serial: FAILED
            rtc: PASSED
       sysflash: PASSED
           cpld: PASSED
         fnvram: PASSED
    ether_loop0: PASSED
    ether_loop1: PASSED
    ether_loop2: PASSED
          hwmon: PASSED
sata0_controler: PASSED
sata1_controler: PASSED
            vpd: PASSED
         eeprom: PASSED
      xmc_check: PASSED
usb1_controller: PASSED
usb2_controller: PASSED
xHCI_controller: PASSED
         system: FAILED

PASSED  : 25
FAILED  : 2
NOT RUN : 0
TOTAL   : 27
```

3.3 Windows XP, XPe and Seven

The Windows XPe BSP for VX304x provides a Kontron embedded API to access the devices.

The detailed PBIT results are accessible by running the executable “post.exe”.

```
C:\Users\user\Desktop\WIN7_x64_BSP_VX304x_R01.00\Drivers\CPLD\Test applications\
Test applications\Post>Post.exe
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_dmi: PASSED
            tpm: PASSED
    pcie_vpx_sw: PASSED
         serial: PASSED
            rtc: PASSED
       sysflash: PASSED
           cpld: PASSED
         fnvram: PASSED
    ether_loop0: PASSED
    ether_loop1: PASSED
    ether_loop2: PASSED
          hwmon: PASSED
sata0_controler: PASSED
sata1_controler: PASSED
            vpd: PASSED
         eeprom: PASSED
      xmc_check: PASSED
usb1_controller: PASSED
usb2_controller: PASSED
xHCI_controller: PASSED
         system: PASSED (FAILED ONCE)

PASSED  : 27
FAILED  : 0
NOT RUN : 0
TOTAL   : 27
```

Chapter 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 VGA or DP (Display Port) 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 VPD EEPROM content from the offset address 0x1000 up to 0x3FFF and store it in a file. Then use this file to copy back data to the new board VPD EEPROM to configure.

One of the most useful features of PBIT for deployments is implemented in the “system” test (see section 2.10 page 32). 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 21). In addition to these results, each test implements a dedicated “FAILED ONCE” flag which has to be reset separately (see section 2.6 page 22). 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

MAILING ADDRESS
Kontron Modular Computers S.A.S.
150 rue Marcelin Berthelot - BP 244
ZI TOULON EST
83078 TOULON CEDEX - France

TELEPHONE AND E-MAIL
+33 (0) 4 98 16 34 00
Sales: Order-MAR-Toulon@kontron.com
Support: GSS-MAR-Toulon@kontron.com

For further information about other Kontron products, please visit our Internet web site: www.kontron.com.