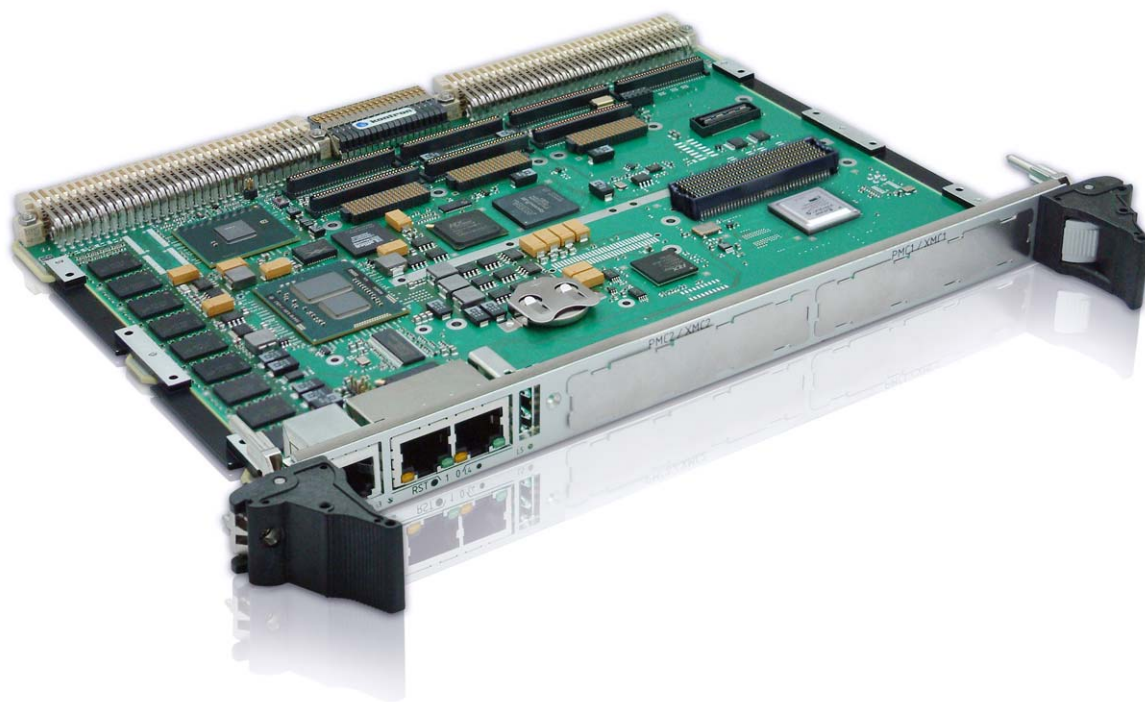


» VM6050 «



PBIT User's Guide

SD.DT.F88-4e - August 2015

Revision History

Publication Title:		VM6050 PBIT User's Guide
Doc. ID:		SD.DT.F88-4e
Rev.	Brief Description of Changes	Date of Issue
4e	New PBIT version 2.6 ID15190 New sections: - 2.11 Bypass the PBIT Tests - 3.1.3 Linux Kdiag Utility	08-2015
3e	PBIT version 2.4 ID13280 evolution Section 2.10 -PBIT System Test- updated	10-2013
2e	PBIT version 2.3 ID13077 evolution System test evolution (chap. 2.10) Add kdiag system_edit menu (chap. 2.10) Add Reset Flag parameter (chap. 2.7.1) Add kdiag deleteall command (chap 2.7.3) Modify kdiag cfg + flag behaviour (chap. 2.7.4) See BIOS User Manual SD.DT.F89 for more details	04-2013
1e	PBIT Version 2.0 ID 12180	07-2012
0e	Initial Version	11-2011

Copyright © 2015 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.

Table Of Contents

Chapter 1 - PBIT Overview	1
1.1 Related Documents	1
1.2 PBIT Installation and Activation	1
1.3 PBIT Configuration	2
1.3.1 Configure PBIT by command line	2
1.3.2 Configure PBIT to run automatically	7
1.4 Synthetic PBIT Result	8
1.5 PBIT Tests List	9
1.5.1 Selected Tests List	9
1.5.2 Not Selected Tests List	11
1.6 PBIT Execution Time	13
Chapter 2 - PBIT Command Line Reference Guide	15
2.1 On-line Help	15
2.2 Display the List of Selected Tests	18
2.3 Execute the PBIT from the Command Line	19
2.4 Execute PBIT in Loop Mode	20
2.5 Get the PBIT Results	21
2.6 Clear the PBIT Results	21
2.7 Configure PBIT Tests List to Execute	22
2.7.1 Run mode parameters	22
2.7.2 Adding a Test to the Current Run List	22
2.7.3 Removing a Test from the Current Run List	23
2.7.4 Set a RUN mode parameter to All the Tests of the Current Run List	24
2.7.5 Restore the Default Run List	29
2.8 Run Pbit in Silent Mode	29
2.9 Display the PBIT Version	29
2.10 PBIT System Test	30
2.10.1 Recording the system configuration:	30
2.10.2 Testing the current system configuration:	32
2.10.3 Editing the system configuration:	33
2.10.3.1 <i>Edit system items (to ignore specific test)</i>	33
2.10.3.2 <i>Print debug settings</i>	39
2.10.4 Clearing system stats	42
2.10.5 Clearing the system configuration	43
2.11 Bypass the PBIT Tests	43

Chapter 3 - PBIT and OS Interfaces	44
3.1 Linux	44
3.1.1 Linux Synthetic PBIT Result	44
3.1.2 Linux Detailed PBIT Result	45
3.1.3 Linux Kdiag Utility	46
3.2 VxWorks	51
3.2.1 VxWorks Synthetic PBIT Result	51
3.2.2 VxWorks Detailed PBIT Result	51
3.3 Windows XP, XPe and Seven	52
Chapter 4 - Use Cases	53
4.1 PBIT Features and Benefits	53
4.1.1 EVAL	53
4.1.2 DEVEL	53
4.1.3 DEPLOY	53
4.1.4 MAINTAIN	54
4.1.5 MANUFACTURING	54
Appendix A - List of Abbreviations	55

Chapter 1 - PBIT Overview

This document describes the PowerOn Built In Test (PBIT) for Kontron VM6050 boards.

The PBIT can be activated under the VM6050 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 VM6050 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 44.
- > Simplified test result is also available in a 8-bit PLD register (register 0x2).

1.1 Related Documents

Hardware:

- > VM6050 6U VME SBC User's Guide CA.DT.A93
- > VM6050 Hardware Release Notes CA.DT.A94

BIOS:

- > VM6050 AMI BIOS User Reference Manual SD.DT.F89

1.2 PBIT Installation and Activation

The PBIT software comes pre-installed in the system Flash, along with the EFI BIOS firmware, on the VM6050 boards.

The PBIT can be activated on any VM6050. Please contact Kontron support team for more information.

To install a new BIOS version including a new PBIT version please refer to the VM6050 BIOS User's Reference Manual - SD.DT.F89.

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 VM6050.

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 PBIT by command line

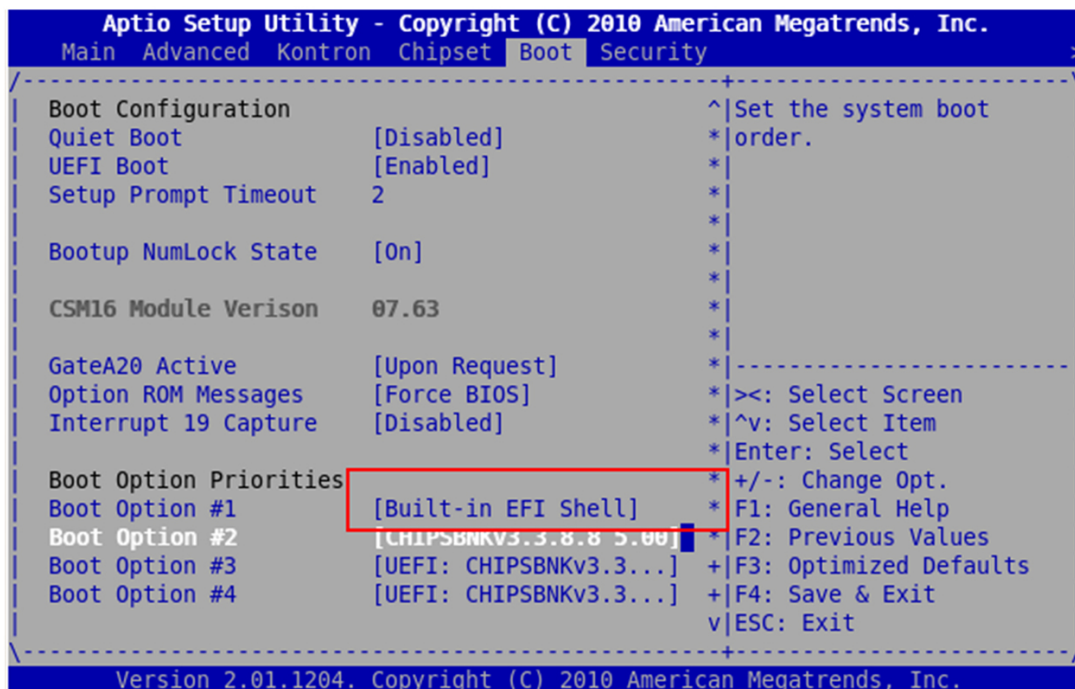
The following explains how to configure and execute the PBIT by a command line.

> 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:

```
VM6050> kdiag version
PBIT VERSION 2.6 ID15190
```

> Launch the PBIT manually for verification:

```
VM6050> 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 "serial" (fast,simple) PASSED
PBIT "rtc" (fast,simple) PASSED
PBIT "sysflash" (fast,simple) PASSED
PBIT "cpld" (fast,simple) PASSED
PBIT "temp_sensors" (fast,simple) PASSED
PBIT "temperature" (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 "ether_loop3" (fast,simple) PASSED
PBIT "voltage" (fast,simple) PASSED
PBIT "sata_controller" (fast,simple) PASSED
PBIT "vpd" (fast,simple) PASSED
PBIT "eeprom" (fast,simple) PASSED
PBIT "vme_check" (fast,simple) PASSED
PBIT "pmcA_xmc_check" (fast,simple) XMC No XMC nor PMC detected
PASSED
PBIT "pmcB_xmc_check" (fast,simple) XMC No XMC nor PMC detected
PASSED
PBIT "usb1_controller" (fast,simple) PASSED
PBIT "usb2_controller" (fast,simple) PASSED
PBIT "system" (fast,simple)
SYSTEM INFOS NOT SAVED
PASSED
```

> Optionally 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 equipment must be connected such as PMC and XMC cards, Ethernet connections to external switches, USB devices, SATA devices. See section 2.10 page 30 - “PBIT System Learn Test” for more details.

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

```
VM6050> kdiag learn system
Seg  Bus  Dev  Func
---  ---  ---  ----
00   00   00   00 ==> Bridge Device - Host/PCI bridge
      Vendor 8086 Device 006A Prog Interface 0
00   00   01   00 ==> Bridge Device - PCI/PCI bridge
      Vendor 8086 Device 0045 Prog Interface 0
00   00   02   00 ==> Display Controller - VGA/8514 controller
      Vendor 8086 Device 0046 Prog Interface 0
00   00   06   00 ==> Bridge Device - PCI/PCI bridge
      Vendor 8086 Device 0047 Prog Interface 0
00   00   16   00 ==> Simple Communications Controllers - Other communicati
      Vendor 8086 Device 3B64 Prog Interface 0
00   00   16   01 ==> Simple Communications Controllers - Other communicati
      Vendor 8086 Device 3B65 Prog Interface 0
00   00   16   02 ==> Mass Storage Controller - IDE controller
      Vendor 8086 Device 3B66 Prog Interface 85
00   00   16   03 ==> Simple Communications Controllers - Serial controller
      Vendor 8086 Device 3B67 Prog Interface 2
00   00   1A   00 ==> Serial Bus Controllers - USB
      Vendor 8086 Device 3B3C Prog Interface 20
00   00   1C   00 ==> Bridge Device - PCI/PCI bridge
      Vendor 8086 Device 3B42 Prog Interface 0
00   00   1C   04 ==> Bridge Device - PCI/PCI bridge
      Vendor 8086 Device 3B4A Prog Interface 0
00   00   1C   06 ==> Bridge Device - PCI/PCI bridge
      Vendor 8086 Device 3B4E Prog Interface 0
00   00   1C   07 ==> Bridge Device - PCI/PCI bridge
      Vendor 8086 Device 3B50 Prog Interface 0
00   00   1D   00 ==> Serial Bus Controllers - USB
      Vendor 8086 Device 3B34 Prog Interface 20
00   00   1E   00 ==> Bridge Device - PCI/PCI bridge
      Vendor 8086 Device 2448 Prog Interface 1
00   00   1F   00 ==> Bridge Device - PCI/ISA bridge
      Vendor 8086 Device 3B07 Prog Interface 0
00   00   1F   02 ==> Mass Storage Controller - IDE controller
      Vendor 8086 Device 3B2E Prog Interface 8F
00   00   1F   03 ==> Serial Bus Controllers - System Management Bus
      Vendor 8086 Device 3B30 Prog Interface 0
00   00   1F   05 ==> Mass Storage Controller - IDE controller
      Vendor 8086 Device 3B2D Prog Interface 85
```

```
00 04 00 00 ==> Network Controller - Ethernet controller
Vendor 8086 Device 150E Prog Interface 0
00 04 00 01 ==> Network Controller - Ethernet controller
Vendor 8086 Device 150E Prog Interface 0
00 04 00 02 ==> Network Controller - Ethernet controller
Vendor 8086 Device 150E Prog Interface 0
00 04 00 03 ==> Network Controller - Ethernet controller
Vendor 8086 Device 150E Prog Interface 0
00 05 00 00 ==> Bridge Device - PCI/PCI bridge
Vendor 10B5 Device 8112 Prog Interface 0
00 06 09 00 ==> Bridge Device - Other bridge type
Vendor 1059 Device 9035 Prog Interface 0
CPLD Version 0xD
DRAM size 4 GB
System Controller
Geographical Address 2
PMC1(PCI32) not present
SATA PORT 0 WDC WD800JD-75 (80.0GB)
Device detected on SMBUS0, address = 0x32
BIOS Setup Checksum : 45330
BIOS Version : ID13280
ETH0 connected device speed 1000Mb/s
ETH1 connected device speed 1000Mb/s
ETH2 connected device speed 1000Mb/s
ETH3 connected device speed 1000Mb/s
2 Drives, 2 Hubs
MassStorage Device name (0) : 0.00
MassStorage Device name (1) : Freecom FM-10 Pro 2
USB Port 1 connected
USB Port 3 connected

Number of System Test Elements detected : 90
DRAM area [ 0x2A180A0 0x2A18D00 ] will be stored in EEPROM
Storing system infos...

Storing system configuration...
```

> Check the PBIT results:

```
VM6050> 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 : serial (fast,simple)
PASSED : rtc (fast,simple)
PASSED : sysflash (fast,simple)
PASSED : cpld (fast,simple)
PASSED : temp_sensors (fast,simple)
PASSED : temperature (fast,simple)
PASSED : fnvram (fast,simple)
PASSED : ether_loop0 (fast,simple)
PASSED : ether_loop1 (fast,simple)
PASSED : ether_loop2 (fast,simple)
PASSED : ether_loop3 (fast,simple)
PASSED : voltage (fast,simple)
PASSED : sata_controller (fast,simple)
PASSED : vpd (fast,simple)
PASSED : eeprom (fast,simple)
PASSED : vme_check (fast,simple)
PASSED : pmcA_xmc_check (fast,simple)
PASSED : pmcB_xmc_check (fast,simple)
PASSED : usb1_controller (fast,simple)
PASSED : usb2_controller (fast,simple)
PASSED : system (fast,simple)

RUN      : 27
PASSED   : 27
FAILED   : 0
NOT_RUN  : 0
```

1.3.2 Configure 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".

```
VM6050> set bootcmd "kdiag run"
```

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

Default value is "1". Value "0" is possible.

```
VM6050> set bootdelay 1
```

> Verify:

```
VM6050> set
bootcmd   : kdiag run
bootdelay : 1
```

> Then reset the system:

```
VM6050> reset
```

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



1: To stop under the EFI shell after the PBIT execution, define the variable named "bootdontexit":

```
VM6050> set bootdontexit 1
```

2: To execute a shell script including a list of commands, type:

```
VM6050> set startupAuto 1
VM6050> set startupdelay 1
```

Then plug an USB device including in the top directory the script file named "startup.nsh" including the wanted shell commands.

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 VM6050. No signal on any connector will be modified during the default test execution.

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

```
VM6050> 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
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
cp1d (24) - Checks PLD, GeoAddress, watchdog
  capabilities : fast,simple/complex,allresets
  run mode 1   : fast,simple,allresets
temp_sensors (31) - Checks if all temperature sensors are detected.
  capabilities : fast,simple,allresets
  run mode 1   : fast,simple,allresets
temperature (32) - Checks if temperatures are OK.
  capabilities : fast,simple,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 82580 Gigabit FrontPanel Interface 0 in Loopback mode
  capabilities : fast,simple,allresets
  run mode 1  : fast,simple,allresets
ether_loop1 (56) - Checks 82580 Gigabit FrontPanel Interface 1 in Loopback mode
  capabilities : fast,simple,allresets
  run mode 1  : fast,simple,allresets
ether_loop2 (57) - Checks 82580 Gigabit Rear Interface 2 in Loopback mode
  capabilities : fast,simple,allresets
  run mode 1  : fast,simple,allresets
ether_loop3 (58) - Checks 82580 Gigabit Rear Interface 3 in Loopback mode
  capabilities : fast,simple,allresets
  run mode 1  : fast,simple,allresets
voltage (60) - Checks the voltage sensors & value
  capabilities : fast,simple,allresets
  run mode 1  : fast,simple,allresets
sata_controller (67) - Checks sata 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
vme_check (73) - Checks ALMA VME device access
  capabilities : fast,simple/complex,allresets
  run mode 1  : fast,complex,allresets
pmcA_xmc_check (76) - Checks PMCA 64bits/XMC path & slot
  capabilities : fast,simple/complex,allresets
  run mode 1  : fast,simple,allresets
pmcB_xmc_check (78) - Checks PMCB 32bits/XMC path & slot
  capabilities : fast,simple/complex,allresets
  run mode 1  : fast,simple,allresets
usb1_controller (87) - Checks first Ibexpeak usb controller
  capabilities : fast,simple,allresets
  run mode 1  : fast,simple,allresets
usb2_controller (88) - Checks second Ibexpeak usb controller
  capabilities : fast,simple,allresets
  run mode 1  : fast,simple,allresets
system (86) - 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:

```
VM6050> 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
fpga (25) - Check FPGA presence
  capabilities : fast,simple/complex,allresets
ether_link0 (50) - Checks the link status is "UP" on 82580 Gigabit FrontPanel Interface 0
  capabilities : fast,simple/complex,allresets
ether_link1 (51) - Checks the link status is "UP" on 82580 Gigabit FrontPanel Interface 1
  capabilities : fast,simple/complex,allresets
ether_link2 (52) - Checks the link status is "UP" on 82580 Gigabit Rear Interface 2
  capabilities : fast,simple/complex,allresets
ether_link3 (53) - Checks the link status is "UP" on 82580 Gigabit Rear Interface 3
  capabilities : fast,simple/complex,allresets
sata0_dev_see (63) - Checks if a SATA0 disk on rear is present
  capabilities : fast,simple/complex,allresets
sata1_dev_see (64) - Checks if a SATA1 disk on rear is present
  capabilities : fast,simple/complex,allresets
sata2_dev_see (65) - Checks if a SATA2 disk onboard is present
  capabilities : fast,simple/complex,allresets
sata3_dev_see (66) - Checks if a SATA3 disk on mezzanine is present
  capabilities : fast,simple/complex,allresets
pmcA_xmc_see (77) - Checks if device is present on PMCA 64bits/XMC slot
  capabilities : fast,simple/complex,allresets
pmcB_xmc_see (79) - Checks if device is present on PMCB 32bits/XMC slot
  capabilities : fast,simple/complex,allresets
usb0_dev_see (80) - Check if a USB0 device on mezzanine is present
  capabilities : fast,simple/complex,allresets
usb1_dev_see (81) - Check if a USB1 device on front panel is present
  capabilities : fast,simple/complex,allresets
usb2_dev_see (82) - Check if a USB2 device on rear is present
  capabilities : fast,simple/complex,allresets
usb3_dev_see (83) - Check if a USB3 device on rear is present
  capabilities : fast,simple/complex,allresets
usb4_dev_see (84) - Check if a USB4 device on front io module is present
  capabilities : fast,simple/complex,allresets
```

```
usb5_dev_see (85) - Check if a USB5 device on front io module is present
  capabilities : fast,simple/complex,allresets
v2pmc2_check (90) - Checks V2PMC2 paths & slots
  capabilities : fast,simple/complex,allresets
v2pmc2_pmc_see (91) - Checks a PMC is present on V2PMC
  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.

VM6050>

The unselected tests list contains:

- > Memory tests to complete the default memory tests. Long execution time.
- > External equipment dependent tests for Ethernet interfaces, Sata Devices, PMC and XMC cards, 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.

1.6 PBIT Execution Time

The default PBIT (see section 1.5.1 page 9) runs in about 15 seconds (for a 4 GB DRAM).

Below is the execution time for each test and for a board with 4 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	0.6s
mem_addr (fast,simple)	2	3.7s
mem_bitflip (fast,simple)	3	3.2s
mem_addrpat (fast,simple)	4	2.7s
mem_addrpat2 (fast,simple)	5	3.3s
mem_pattern1 (slow,simple)	6	2.7s
mem_pattern2 (slow,simple)	7	2.7s
mem_pattern3 (slow,simple)	8	2.6s
mem_pattern4 (slow,simple)	9	2.7s
serial (fast,simple)	16	0.35s
rtc (fast,simple)	20	0.2s
sysflash (fast,simple)	22	0.2s
cpld (fast,simple)	24	0.2s
fpga (fast,simple)	25	0.2s
temp_sensors (fast,simple)	31	0.2s
temperature (fast,simple)	32	0.2s
fnvram (fast,simple)	40	0.45s
ether_link0 (fast,simple)	50	0.2s
ether_link1 (fast,simple)	51	0.2s
ether_link2 (fast,simple)	52	0.2s
ether_link3 (fast,simple)	53	0.2s
ether_loop0 (fast,simple)	55	1s
ether_loop1 (fast,simple)	56	1s
ether_loop2 (fast,simple)	57	1s
eth_loop3 (fast,simple)	58	1s
voltage (fast,simple)	60	0.2s
sata0_dev_see (fast,simple)	63	0.25s
Sata1_dev_see (fast,simple)	64	0.25s
Sata2_dev_see (fast,simple)	65	0.25s
Sata3_dev_see (fast,simple)	66	0.25s
sata_controler (fast,simple)	67	0.25s
vpd (fast,simple)	70	0.2s
EEPROM (fast,simple)	71	0.3s

Test name	Test ID number	Execution Time
vme_check(fast,simple)	73	0.2s
pmcA_xmc_check (fast,simple)	76	0.3s
pmcA_xmc_see (fast,simple)	77	0.3s
pmcB_xmc_check (fast,simple)	78	0.3s
pmcB_xmc_see (fast,simple)	79	0.3s
usb0_dev_see (fast,simple)	80	0.25s
usb1_dev_see (fast,simple)	81	0.25s
usb2_dev_see (fast,simple)	82	0.25s
usb3_dev_see (fast,simple)	83	0.25s
usb4_dev_see (fast,simple)	84	0.25s
usb5_dev_see (fast,simple)	85	0.25s
usb1_controller (fast,simple)	87	0.25s
usb2_controller (fast,simple)	88	0.25s
system (fast,simple) (*)	86	1s to 10s Configurable. About 1s if all tested Ethernet links are connected. 3s must be added for each Ethernet link unconnected.
v2pmc2_check (fast,simple)	90	0.35s
v2pmc2_pmc-see (fast,simple)	91	0.35s

(*) 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 12s (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

```
VM6050> 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): 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

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
```

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.

```
VM6050> 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
ether_link0 (50) - Checks the link status is "UP" on 82580 Gigabit FrontPanel Interface 0
  capabilities : fast,simple/complex
( . . . )
Use 'help kdiag' to get more info.
VM6050>
```

2.3 Execute the PBIT from the Command Line

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

```
VM6050> 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 "serial" (fast,simple) PASSED
PBIT "rtc" (fast,simple) PASSED
PBIT "sysflash" (fast,simple) PASSED
PBIT "cpld" (fast,simple) PASSED
PBIT "temp_sensors" (fast,simple) PASSED
PBIT "temperature" (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 "ether_loop3" (fast,simple) PASSED
PBIT "voltage" (fast,simple) PASSED
PBIT "sata_controller" (fast,simple) PASSED
PBIT "vpd" (fast,simple) PASSED
PBIT "eeprom" (fast,simple) PASSED
PBIT "vme_check" (fast,simple) PASSED
PBIT "pmcA_xmc_check" (fast,simple) XMC No XMC nor PMC detected
PASSED
PBIT "pmcB_xmc_check" (fast,simple) XMC No XMC nor PMC detected
PASSED
PBIT "usb1_controller" (fast,simple) PASSED
PBIT "usb2_controller" (fast,simple) PASSED
PBIT "system" (fast,simple)
SYSTEM INFOS NOT SAVED
PASSED

VM6050>
```

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

```
VM6050> kdiag run <PBIT number | PBIT name ...>
```

> For example:

```
VM6050> kdiag run usb1_controller sysflash
PBIT "usb1_controller" (fast,simple) PASSED
PBIT "sysflash" (fast,simple) PASSED
```

That is equivalent to:

```
VM6050> kdiag run 87 22
PBIT "usb1_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 PBIT in Loop Mode

To run PBIT in loop mode, enter:

```
VM6050 > kdiag run loop <count>
```

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

To run a single test in loop mode, enter:

```
VM6050> kdiag run loop <count> <PBIT number | PBIT name ...>
```

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

```
VM6050> kdiag run loop 10 55
```

It is equivalent to enter:

```
VM6050> kdiag run loop 10 ether_loop0
```

2.5 Get the PBIT Results

To get the PBIT results, use the “kdiag stat” command:

```
VM6050> 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 : serial (fast,simple)
PASSED : rtc (fast,simple)
PASSED : sysflash (fast,simple)
PASSED : cpld (fast,simple)
PASSED : temp_sensors (fast,simple)
PASSED : temperature (fast,simple)
PASSED : fnvram (fast,simple)
PASSED : ether_loop0 (fast,simple)
PASSED : ether_loop1 (fast,simple)
PASSED : ether_loop2 (fast,simple)
PASSED : ether_loop3 (fast,simple)
PASSED : voltage (fast,simple)
PASSED : sata_controller (fast,simple)
PASSED : vpd (fast,simple)
PASSED : eeprom (fast,simple)
PASSED : vme_check (fast,simple)
PASSED : pmcA_xmc_check (fast,simple)
PASSED: pmcB_xmc_check (fast,simple)
PASSED : usb1_controller (fast,simple)
PASSED : usb2_controller (fast,simple)
PASSED : system (fast,simple)

RUN      : 27
PASSED   : 27
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:

```
VM6050> kdiag clrstat
```

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

```
VM6050> kdiag clral1stat
```

2.7 Configure 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):

haltontfail: 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 VM6050 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:

```
VM6050> 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:

```
VM6050> kdiag cfg default 16
```

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

```
VM6050> kdiag cfg promptonfail 16
```

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

```
VM6050> kdiag cfg complex,promptonfail 16
```

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

```
VM6050> 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:

```
VM6050> kdiag cfg delete 16
```

Verify:

```
VM6050> kdiag 16
PBIT "16" is not configured to run from command line
```

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

```
VM6050> kdiag deleteall
```

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

```
VM6050> 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

Using command: `kdiag cfg "runflag"` allows to set a "runflag" (HALT, CONFIG, RESET described in 2.7.1 page 22) to the all current run list test.

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

```
VM6050> 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
--> serial (16) - Checks the serial line COM2
--> rtc (20) - Checks the RTC time
--> sysflash (22) - Checks the BIOS rescue in system flash
--> cp1d (24) - Checks PLD, GeoAddress, watchdog
--> temp_sensors (31) - Checks if all temperature sensors are detected.
--> temperature (32) - Checks if temperatures are OK.
--> fnvram (40) - Checks F-NVRAM device.
--> ether_loop0 (55) - Checks 82580 Gigabit FrontPanel Interface 0 in Loopback mode
--> ether_loop1 (56) - Checks 82580 Gigabit FrontPanel Interface 1 in Loopback mode
--> ether_loop2 (57) - Checks 82580 Gigabit Rear Interface 2 in Loopback mode
--> ether_loop3 (58) - Checks 82580 Gigabit Rear Interface 3 in Loopback mode
--> voltage (60) - Checks the voltage sensors & value
--> sata_controller (67) - Checks sata controller
--> vpd (70) - Checks VPD data required for board operation.
--> eeprom (71) - Checks User EEPROM (0xA2)
--> vme_check (73) - Checks ALMA VME device access
--> pmcA_xmc_check (76) - Checks PMCA 64bits/XMC path & slot
--> pmcB_xmc_check (78) - Checks PMCB 32bits/XMC path & slot
--> usb1_controller (87) - Checks first Ibexpeak usb controller
--> usb2_controller (88) - Checks second Ibexpeak usb controller
--> usb2_controller (88) - Checks second Ibexpeak usb controller

VM6050>
```

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

```
VM6050> kdiag cfg promptonfail
Configuration of current tests in asked mode:
command,fast,simple,promptonfail,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
--> serial (16) - Checks the serial line COM2
--> rtc (20) - Checks the RTC time
--> sysflash (22) - Checks the BIOS rescue in system flash
--> cp1d (24) - Checks PLD, GeoAddress, watchdog
--> temp_sensors (31) - Checks if all temperature sensors are detected.
--> temperature (32) - Checks if temperatures are OK.
--> fnvram (40) - Checks F-NVRAM device.
--> ether_loop0 (55) - Checks 82580 Gigabit FrontPanel Interface 0 in Loopback mode
--> ether_loop1 (56) - Checks 82580 Gigabit FrontPanel Interface 1 in Loopback mode
--> ether_loop2 (57) - Checks 82580 Gigabit Rear Interface 2 in Loopback mode
--> ether_loop3 (58) - Checks 82580 Gigabit Rear Interface 3 in Loopback mode
--> voltage (60) - Checks the voltage sensors & value
--> sata_controller (67) - Checks sata controller
--> vpd (70) - Checks VPD data required for board operation.
--> eeprom (71) - Checks User EEPROM (0xA2)
--> vme_check (73) - Checks ALMA VME device access
--> pmcA_xmc_check (76) - Checks PMCA 64bits/XMC path & slot
--> pmcB_xmc_check (78) - Checks PMCB 32bits/XMC path & slot
--> usb1_controller (87) - Checks first Ibexpeak usb controller
--> usb1_controller (87) - Checks first Ibexpeak usb controller
--> usb2_controller (88) - Checks second Ibexpeak usb controller

VM6050>
```

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

```
VM6050> 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
--> 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
--> temp_sensors (31) - Checks if all temperature sensors are detected.
--> temperature (32) - Checks if temperatures are OK.
--> fnvram (40) - Checks F-NVRAM device.
--> ether_loop0 (55) - Checks 82580 Gigabit FrontPanel Interface 0 in Loopback mode
--> ether_loop1 (56) - Checks 82580 Gigabit FrontPanel Interface 1 in Loopback mode
--> ether_loop2 (57) - Checks 82580 Gigabit Rear Interface 2 in Loopback mode
--> ether_loop3 (58) - Checks 82580 Gigabit Rear Interface 3 in Loopback mode
--> voltage (60) - Checks the voltage sensors & value
--> sata_controller (67) - Checks sata controller
--> vpd (70) - Checks VPD data required for board operation.
--> eeprom (71) - Checks User EEPROM (0xA2)
--> vme_check (73) - Checks ALMA VME device access
--> pmcA_xmc_check (76) - Checks PMCA 64bits/XMC path & slot
--> pmcB_xmc_check (78) - Checks PMCB 32bits/XMC path & slot
--> system (86) - Checks system configuration SETUP,PCIe,SATA,USB,ETH stability
--> usb1_controller (87) - Checks first Ibexpeak usb controller
--> usb2_controller (88) - Checks second Ibexpeak usb controller
```

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

```
VM6050> 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
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
temp_sensors (31) - Checks if all temperature sensors are detected.
  capabilities : fast,simple,allresets
  run mode 1   : fast,simple,promptonfail,poweronreset
temperature (32) - Checks if temperatures are OK.
  capabilities : fast,simple,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 82580 Gigabit FrontPanel Interface 0 in Loopback mode
  capabilities : fast,simple,allresets
```

```

run mode 1 : fast,simple,promptonfail,poweronreset
ether_loop1 (56) - Checks 82580 Gigabit FrontPanel Interface 1 in Loopback mode
capabilities : fast,simple,allresets
run mode 1 : fast,simple,promptonfail,poweronreset
ether_loop2 (57) - Checks 82580 Gigabit Rear Interface 2 in Loopback mode
capabilities : fast,simple,allresets
run mode 1 : fast,simple,promptonfail,poweronreset
ether_loop3 (58) - Checks 82580 Gigabit Rear Interface 3 in Loopback mode
capabilities : fast,simple,allresets
run mode 1 : fast,simple,promptonfail,poweronreset
voltage (60) - Checks the voltage sensors & value
capabilities : fast,simple,allresets
run mode 1 : fast,simple,promptonfail,poweronreset
sata_controller (67) - Checks sata 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
vme_check (73) - Checks ALMA VME device access
capabilities : fast,simple/complex,allresets
run mode 1 : fast,simple,promptonfail,poweronreset
pmcA_xmc_check (76) - Checks PMCA 64bits/XMC path & slot
capabilities : fast,simple/complex,allresets
run mode 1 : fast,simple,promptonfail,poweronreset
pmcB_xmc_check (78) - Checks PMCB 32bits/XMC path & slot
capabilities : fast,simple/complex,allresets
run mode 1 : fast,simple,promptonfail,poweronreset
system (86) - Checks system configuration SETUP,PCIE,SATA,USB,ETH stability
capabilities : fast,simple/complex,allresets
run mode 1 : fast,simple,promptonfail,poweronreset
usb1_controller (87) - Checks first Ibexpeak usb controller
capabilities : fast,simple,allresets
run mode 1 : fast,simple,promptonfail,poweronreset
usb2_controller (88) - Checks second Ibexpeak usb controller
capabilities : fast,simple,allresets
run mode 1 : fast,simple,promptonfail,poweronreset

```

Verify for test 3:

```

VM6050> kdiag 3
mem_bitflip (3) - Checks Mem/ECC using bit-flip pattern ((1 << (offset % 32))
capabilities : slow/fast,simple
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:

```
VM6050> kdiag default
```

2.8 Run Pbit in Silent Mode

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

```
VM6050> kdiag silent  
PBIT set in silent mode
```

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

```
VM6050> 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:

```
VM6050> kdiag version  
PBIT VERSION 2.6 ID15190
```

2.10 PBIT System Test

2.10.1 Recording the system configuration:

The command “kdiag learn system” (equivalent to “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; Keyboards, mice, drives and hubs counts; 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

Example :

```

VM6050> kdiag learn system
  Seg  Bus  Dev  Func
  ---  ---  ---  ---
    00  00   00   00 ==> Bridge Device - Host/PCI bridge
          Vendor 8086 Device 006A Prog Interface 0
    00  00   01   00 ==> Bridge Device - PCI/PCI bridge
          Vendor 8086 Device 0045 Prog Interface 0
    00  00   02   00 ==> Display Controller - VGA/8514 controller
          Vendor 8086 Device 0046 Prog Interface 0
    00  00   06   00 ==> Bridge Device - PCI/PCI bridge
          Vendor 8086 Device 0047 Prog Interface 0
    00  00   16   00 ==> Simple Communications Controllers - Other communicati
          Vendor 8086 Device 3B64 Prog Interface 0
    00  00   16   01 ==> Simple Communications Controllers - Other communicati
          Vendor 8086 Device 3B65 Prog Interface 0
    00  00   16   02 ==> Mass Storage Controller - IDE controller
          Vendor 8086 Device 3B66 Prog Interface 85
    00  00   16   03 ==> Simple Communications Controllers - Serial controller
          Vendor 8086 Device 3B67 Prog Interface 2
    00  00   1A   00 ==> Serial Bus Controllers - USB
          Vendor 8086 Device 3B3C Prog Interface 20
    00  00   1C   00 ==> Bridge Device - PCI/PCI bridge
          Vendor 8086 Device 3B42 Prog Interface 0
    00  00   1C   04 ==> Bridge Device - PCI/PCI bridge
          Vendor 8086 Device 3B4A Prog Interface 0
    00  00   1C   06 ==> Bridge Device - PCI/PCI bridge
          Vendor 8086 Device 3B4E Prog Interface 0

```

```
00 00 1C 07 ==> Bridge Device - PCI/PCI bridge
Vendor 8086 Device 3B50 Prog Interface 0

00 00 1D 00 ==> Serial Bus Controllers - USB
Vendor 8086 Device 3B34 Prog Interface 20

00 00 1E 00 ==> Bridge Device - PCI/PCI bridge
Vendor 8086 Device 2448 Prog Interface 1

00 00 1F 00 ==> Bridge Device - PCI/ISA bridge
Vendor 8086 Device 3B07 Prog Interface 0

00 00 1F 02 ==> Mass Storage Controller - IDE controller
Vendor 8086 Device 3B2E Prog Interface 8F

00 00 1F 03 ==> Serial Bus Controllers - System Management Bus
Vendor 8086 Device 3B30 Prog Interface 0

00 00 1F 05 ==> Mass Storage Controller - IDE controller
Vendor 8086 Device 3B2D Prog Interface 85

00 04 00 00 ==> Network Controller - Ethernet controller
Vendor 8086 Device 150E Prog Interface 0

00 04 00 01 ==> Network Controller - Ethernet controller
Vendor 8086 Device 150E Prog Interface 0

00 04 00 02 ==> Network Controller - Ethernet controller
Vendor 8086 Device 150E Prog Interface 0

00 04 00 03 ==> Network Controller - Ethernet controller
Vendor 8086 Device 150E Prog Interface 0

00 05 00 00 ==> Bridge Device - PCI/PCI bridge
Vendor 10B5 Device 8112 Prog Interface 0

00 06 09 00 ==> Bridge Device - Other bridge type
Vendor 1059 Device 9035 Prog Interface 0
```

CPLD Version 0xB

DRAM size 4 GB

System Controller

Geographical Address 1

PMC1(PCI32) not present

Device detected on SMBUS0, address = 0x30

BIOS Setup Checksum : 42889

BIOS Version : ID13134

ETH2 connected device speed 1000Mb/s

2 Drives, 1 Keyboard, 1 Mouse, 3 Hubs

MassStorage Device name (0) : Generic Flash Disk

MassStorage Device name (1) : SMART eUSB 880C

USB Port 0 connected

USB Port 3 connected

Number of System Test Elements detected : 90

DRAM area [0x2DE2EA0 0x2DE3B00] 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:

```

VM6050> kdiag run system
PBIT "system" (fast,simple)
PCI... CPLD... SATA... SMBUS... BIOS... ETH... USB...
ERR)PCI 00:1F:02 DeviceID (R)0x3B2E (D)0x3B2F
ERR)PCI 00:1F:02 ClassCode (R)01018F (D)01018F
ERR)PCI Bus:Dev:Func (R)00:1F:05 (D)04:00:00
ERR)PCI 00:1F:05 DeviceID (R)0x3B2D (D)0x150E
ERR)PCI 00:1F:05 ClassCode (R)010185 (D)010185
ERR)PCI Bus:Dev:Func (R)04:00:00 (D)04:00:01
ERR)PCI Bus:Dev:Func (R)04:00:01 (D)04:00:02
ERR)PCI Bus:Dev:Func (R)04:00:02 (D)04:00:03
ERR)PCI Bus:Dev:Func (R)04:00:03 (D)05:00:00
ERR)PCI 04:00:03 VendorID (R)0x8086 (D)0x10B5
ERR)PCI 04:00:03 DeviceID (R)0x150E (D)0x8112
ERR)PCI 04:00:03 ClassCode (R)020000 (D)020000
ERR)PCI 04:00:03 LinkStat (R)LinkStat not available (D)Link is UP
ERR)PCI 04:00:03 Wdth/Spd (R)Bandwidth not available (D)x1 / 2.5GT/s
ERR)PCI Bus:Dev:Func (R)05:00:00 (D)06:09:00
ERR)PCI 05:00:00 VendorID (R)0x10B5 (D)0x1059
ERR)PCI 05:00:00 DeviceID (R)0x8112 (D)0x9035
ERR)PCI 05:00:00 ClassCode (R)060400 (D)060400
ERR)PCI 05:00:00 LinkStat (R)Link is UP (D)LinkStat not available
ERR)PCI 05:00:00 Wdth/Spd (R)x1 / 2.5GT/s (D)Bandwidth not available
ERR)PCI Bus:Dev:Func (R)06:09:00 (D)No Device
ERR)PCI 06:09:00 VendorID (R)0x1059 (D)No Device
ERR)PCI 06:09:00 DeviceID (R)0x9035 (D)No Device
ERR)PCI 06:09:00 ClassCode (R)068000 (D)068000
ERR)BIOS Checksum (R)42889 (D)43913
ERR)ETH Port 2 (R)Link UP speed 1000 Mb/s (D)Link UP speed 100 Mb/s
ERR)ETH Port 3 (R)Not Connected (D)Link UP speed 1000 Mb/s
ERR)USB Drive's Count (R)2 Drives (D)3 Drives
ERR)USB Mouse's Count (R)1 Mouse (D)0 Mouse
ERR)USB MassStorage(1) (R)SMART eUSB 880C (D)SMI USB DISK 1100
ERR)USB MassStorage(2) (R)No Device (D)SMART eUSB 880C
ERR)USB Port 1 (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:

```
VM6050> kdiag edit system
```

Equivalent to:

```
VM6050> 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 structure. 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:

```
VM6050> kdiag
      << 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          0xB
1. CPLD Dram Size       4GB
2. CPLD SysCon          System Controller
3. CPLD Geo Address     1
4. CPLD PMC Presence    PMC1(PCI32) 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          0xB
1. CPLD Dram Size       4GB
2. CPLD SysCon          System Controller    (i)
3. CPLD Geo Address     1
4. CPLD PMC Presence    PMC1(PCI32) 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          Not Connected (0.0GB)
4. SATA Port 4          Not Connected (0.0GB)
*-SystemEdit-SATA>>4
*-SystemEdit-SATA-Port 4>>i
Ignore flag has been set
*-SystemEdit-SATA-Port 4>>
```

```
*-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>>ignoreall
SMBUS entirely ignored
*-SystemEdit-SMBUS>>p
0. SMBUS0 (ignored)
SMBUS0 Device 0      0x30                (i)
SMBUS0 Device 1      No Device            (i)
SMBUS0 Device 2      No Device            (i)
SMBUS0 Device 3      No Device            (i)
SMBUS0 Device 4      No Device            (i)
SMBUS0 Device 5      No Device            (i)
SMBUS0 Device 6      No Device            (i)
SMBUS0 Device 7      No Device            (i)
SMBUS0 Device 8      No Device            (i)
SMBUS0 Device 9      No Device            (i)
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,...,24 : 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 Display Controller/VGA/8514 controller
3. PCI 00:06:00 Bridge Device/PCI/PCI bridge
4. PCI 00:16:00 Simple Communications Controllers/Other communication device
5. PCI 00:16:01 Simple Communications Controllers/Other communication device
6. PCI 00:16:02 Mass Storage Controller/IDE controller
7. PCI 00:16:03 Simple Communications Controllers/Serial controller
8. PCI 00:1A:00 Serial Bus Controllers/USB
9. PCI 00:1C:00 Bridge Device/PCI/PCI bridge
10. PCI 00:1C:04 Bridge Device/PCI/PCI bridge
11. PCI 00:1C:06 Bridge Device/PCI/PCI bridge
12. PCI 00:1C:07 Bridge Device/PCI/PCI bridge
13. PCI 00:1D:00 Serial Bus Controllers/USB
14. PCI 00:1E:00 Bridge Device/PCI/PCI bridge
15. PCI 00:1F:00 Bridge Device/PCI/ISA bridge
16. PCI 00:1F:02 Mass Storage Controller/IDE controller [FAILED ONCE] [LAST FAILED]
17. PCI 00:1F:03 Serial Bus Controllers/System Management Bus
18. PCI 00:1F:05 Mass Storage Controller/IDE controller [FAILED ONCE] [LAST FAILED]
19. PCI 04:00:00 Network Controller/Ethernet controller [FAILED ONCE] [LAST FAILED]
20. PCI 04:00:01 Network Controller/Ethernet controller [FAILED ONCE] [LAST FAILED]
21. PCI 04:00:02 Network Controller/Ethernet controller [FAILED ONCE] [LAST FAILED]
22. PCI 04:00:03 Network Controller/Ethernet controller [FAILED ONCE] [LAST FAILED]
23. PCI 05:00:00 Bridge Device/PCI/PCI bridge [FAILED ONCE] [LAST FAILED]
24. PCI 06:09:00 Bridge Device/Other bridge type [FAILED ONCE] [LAST FAILED]
25. PCI No Device No Device/No Device
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
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
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>>1
*-SystemEdit-PCI-00:01:00>>p
0. VendorId          0x8086
1. DeviceId          0x0045
2. ClassCode         060400
   Prog Interface    0
   BaseClass         Bridge Device
   SubClass          PCI/PCI bridge
3. LinkStatus        Link not UP
4. Link Speed/Width  Undefined
*-SystemEdit-PCI-00:01:00>>ignoreall
PCI 00:01:00 entirely ignored
*-SystemEdit-PCI-00:01:00>>p
0. VendorId          0x8086 (i)
1. DeviceId          0x0045 (i)
2. ClassCode         060400 (i)
   Prog Interface    0
   BaseClass         Bridge Device
   SubClass          PCI/PCI bridge
3. LinkStatus        Link not UP (i)
4. Link Speed/Width  Undefined (i)
*-SystemEdit-PCI-00:01:00>>
*-SystemEdit-PCI>>16
*-SystemEdit-PCI-00:1F:02>>p
0. VendorId          0x8086
1. DeviceId          0x3B2E
2. ClassCode         01018F
   Prog Interface    8F
   BaseClass         Mass Storage Controller
   SubClass          IDE controller
*-SystemEdit-PCI-00:1F:02>>2
*-SystemEdit-PCI-00:1F:02-ClassCode>>i
Ignore flag has been set
*-SystemEdit-PCI-00:1F:02-ClassCode>>
*-SystemEdit-PCI-00:1F:02>>p
0. VendorId          0x8086
1. DeviceId          0x3B2E
2. ClassCode         01018F (i)
   Prog Interface    8F
   BaseClass         Mass Storage Controller
   SubClass          IDE controller
*-SystemEdit-PCI-00:1F:02>>
*-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>>ignoreall
BIOS entirely ignored
*-SystemEdit-BIOS>>
```

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

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

```
*-SystemEdit>>eth
*-SystemEdit-ETH>>2
*-SystemEdit-ETH-Port 2>>i
Ignore flag has been set
*-SystemEdit-ETH-Port 2>>
*-SystemEdit-ETH>>3
*-SystemEdit-ETH-Port 3>>i
Ignore flag has been set
*-SystemEdit-ETH-Port 3>>
*-SystemEdit-ETH>>p
0. ETH Port 0           Not Connected
1. ETH Port 1           Not Connected
2. ETH Port 2           Link UP speed 1000 Mb/s (i) [FAILED ONCE] [LAST FAILED]
3. ETH Port 3           Not Connected           (i) [FAILED ONCE] [LAST FAILED]
*-SystemEdit-ETH>>
```

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

```
Edit by feature, choose:PCI (*) CPLD (*) SATA (*) ETH (*) USB SMBUS (i) BIOS (i)
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
Storing system configuration...
```

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 (don't stop scrolling by default), max for scroll limit is 30 lines. **WARNING: WATCHDOG 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 SYNTHETIC MODE with previous test:

```
VM6050>kdiag run system
PBIT "system" (fast,simple)
PCI... CPLD... SATA... SMBUS... BIOS... ETH... USB...
ERR)PCI Results : 24 ERRORS
ERR)BIOS Results : 1 ERRORS
ERR)ETH Results : 2 ERRORS
ERR)USB Results : 5 ERRORS
ERR)TOTAL : 32 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:

```
VM6050> 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>>?
e : ERRORS ONLY mode
i : INFOS & ERRORS ONLY mode
d : DEBUG LVL mode (print all)
q : return to general settings menu
--SystemEdit-generalSettings-printLvl>>d
Print Lvl DEBUG PRINT ALL set
Set Scroll limit (0= no limit , max= 30)
*-SystemEdit-generalSettings-printLvl-Scroll>>10
Scroll Limit set to 10 lines
Set Scroll limit (0= no limit , max= 30)
*-SystemEdit-generalSettings-printLvl-Scroll>>q
*-SystemEdit-generalSettings-printLvl>>q
p : Set Print Level
Current Print Lvl is: DEBUG PRINT ALL
d : Set Detail Level
Current Details Lvl is: NORMAL
*-SystemEdit-generalSettings>>d
*-SystemEdit-generalSettings-detailsLvl>>?
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
*-SystemEdit-generalSettings>>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  
Modifications have been made.  
Do you want to save your configuration before leaving? (y/n)  
*-SystemEdit-Save>>y  
Storing system configuration...
```

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 from previous system test run 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:

```
VM6050> 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>>c
System stats have been cleared
p : Set Print Level
Current Print Lvl is: ERRORS ONLY
d : Set Detail Level
Current Details Lvl is: NORMAL
c : Clear System Stats
```



Clear stats: you must exit the menu so that these changes are taken into account.

2.10.5 Clearing the system configuration

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

```
VM6050>kdiag clear system
```

Equivalent to:

```
VM6050>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 release, it is recommended to clear and learn the system configuration again.

2.11 Bypass the PBIT Tests

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

```
VM6050> kdiag bypass  
PBIT BYPASS set
```

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

```
VM6050> kdiag bypass  
PBIT BYPASS removed
```

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

Chapter 3 - PBIT and OS Interfaces

The PBIT synthetic result and PBIT detailed results are accessible under the Operating System delivered with the VM6050 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 “cp1dtool” facility.

Use the “-a” option and check the register 0x2 value.

> Example:

```
# cp1dtool -a
Reg 0x0 - CPLD_ID = 0x0B
      CPLD_ID=0x0
      CPLD_Debug=0x0
      CPLD_Version=0xB
Reg 0x1 - PCB_ID = 0x50
Reg 0x2 - FIRM_PWON = 0x00
Reg 0x3 - PWON_STATUS = 0x00
Reg 0x4 - PWR_RST_CONFIG = 0x21
      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

VM6050 detected
area = 2, arch = 4
[ 888.493276] cp1d-i2c: Delay 182 us
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
    serial: PASSED
    rtc: PASSED
    sysflash: PASSED
    cp1d: PASSED
    temp_sensors: PASSED
    temperature: PASSED
    fnvram: PASSED
    ether_loop0: PASSED
    ether_loop1: PASSED
    ether_loop2: PASSED
    ether_loop3: PASSED
    voltage: PASSED
    sata_controller: PASSED
    vpd: PASSED
    eeprom: PASSED
    vme_check: PASSED
    pmcA_xmc_check: PASSED
    pmcB_xmc_check: FAILED
    usb1_controller: PASSED
    usb2_controller: PASSED
    system: PASSED

PASSED   : 26
FAILED   : 1
NOT RUN  : 0
TOTAL    : 27
```

```
POSTs configured to run automatically from RAM:
```

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

```
POSTs configured to run automatically from ROM:
```

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

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-VM6050 product including the PBIT Firmware and the Linux BIT for VM6050.

BIT-6-VM6050 includes the Linux BIT 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

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

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

```
Clear System Configuration for system Test:
    kdiag system_clear
```

```
Display PBIT version :
    kdiag version
```

For example:

```
[root@boardVM6050 ~]# kdiag stat
Display status
Status of PBITs configured to run from command line :
PASSED : mem_data (fast,simple,promptonfail)
PASSED : mem_addr (fast,simple,promptonfail)
PASSED : mem_bitflip (fast,simple,promptonfail)
PASSED : mem_addrpat (fast,simple,promptonfail)
PASSED : mem_addrpat2 (fast,simple,promptonfail)
PASSED : mem_pattern1 (fast,simple,promptonfail)
PASSED : mem_pattern2 (fast,simple,promptonfail)
PASSED : mem_pattern3 (fast,simple,promptonfail)
PASSED : mem_pattern4 (fast,simple,promptonfail)
PASSED : sysflash (fast,simple,promptonfail)
PASSED : ether_link0 (fast,simple,promptonfail)
FAILED : ether_link1 (fast,simple,promptonfail)
NOTRUN : ether_link2 (fast,simple,promptonfail)
NOTRUN : ether_link3 (fast,simple,promptonfail)
NOTRUN : ether_loop0 (fast,simple,promptonfail)
NOTRUN : ether_loop1 (fast,simple,promptonfail)
NOTRUN : pmcB_xmc_check (fast,simple,promptonfail)

RUN      : 12
PASSED   : 11
FAILED   : 1
NOT_RUN  : 5
[root@boardVM6050 ~]#

[root@boardVM6050 ~]# kdiag
PBITs configured to run from command line :
mem_data (1) - capabilities : fast,simple,allresets
run mode 1   : fast,simple,promptonfail,poweronreset
mem_addr (2) - capabilities : fast,simple,allresets
run mode 1   : fast,simple,promptonfail,poweronreset
mem_bitflip (3) - capabilities : slow/fast,simple,allresets
run mode 1   : fast,simple,promptonfail,poweronreset
mem_addrpat (4) - capabilities : slow/fast,simple,allresets
run mode 1   : fast,simple,promptonfail,poweronreset
mem_addrpat2 (5) - capabilities : slow/fast,simple,allresets
run mode 1   : fast,simple,promptonfail,poweronreset
```

```

mem_pattern1 (6) - capabilities : slow/fast,simple,allresets
run mode 1 : fast,simple,promptonfail,poweronreset
mem_pattern2 (7) - capabilities : slow/fast,simple,allresets
run mode 1 : fast,simple,promptonfail,poweronreset
mem_pattern3 (8) - capabilities : slow/fast,simple,allresets
run mode 1 : fast,simple,promptonfail,poweronreset
mem_pattern4 (9) - capabilities : slow/fast,simple,allresets
run mode 1 : fast,simple,promptonfail,poweronreset
sysflash (22) - capabilities : fast,simple,allresets
run mode 1 : fast,simple,promptonfail,poweronreset
ether_link0 (50) - capabilities : fast,simple/complex,allresets
run mode 1 : fast,simple,promptonfail,poweronreset
ether_link1 (51) - capabilities : fast,simple/complex,allresets
run mode 1 : fast,simple,promptonfail,poweronreset
ether_link2 (52) - capabilities : fast,simple/complex,allresets
run mode 1 : fast,simple,promptonfail,poweronreset
ether_link3 (53) - capabilities : fast,simple/complex,allresets
run mode 1 : fast,simple,promptonfail,poweronreset
ether_loop0 (55) - capabilities : fast,simple,allresets
run mode 1 : fast,simple,promptonfail,poweronreset
ether_loop1 (56) - capabilities : fast,simple,allresets
run mode 1 : fast,simple,promptonfail,poweronreset
pmcB_xmc_check (78) - capabilities : fast,simple/complex,allresets
run mode 1 : fast,simple,promptonfail,poweronreset

```

Other PBITs available but not yet configured :

```

serial (16) - capabilities : fast,simple/complex,allresets
rtc (20) - capabilities : fast,simple,allresets
cp1d (24) - capabilities : fast,simple/complex,allresets
fpga (25) - capabilities : fast,simple/complex,allresets
temp_sensors (31) - capabilities : fast,simple,allresets
temperature (32) - capabilities : fast,simple,allresets
fnvram (40) - capabilities : fast,simple/complex,allresets
ether_loop2 (57) - capabilities : fast,simple,allresets
ether_loop3 (58) - capabilities : fast,simple,allresets
voltage (60) - capabilities : fast,simple,allresets
sata0_dev_see (63) - capabilities : fast,simple/complex,allresets
sata1_dev_see (64) - capabilities : fast,simple/complex,allresets
sata2_dev_see (65) - capabilities : fast,simple/complex,allresets
sata3_dev_see (66) - capabilities : fast,simple/complex,allresets
sata_controller (67) - capabilities : fast,simple/complex,allresets
vpd (70) - capabilities : fast,simple,allresets
eeprom (71) - capabilities : fast,simple/complex,allresets
vme_check (73) - capabilities : fast,simple/complex,allresets
pmcA_xmc_check (76) - capabilities : fast,simple/complex,allresets
pmcA_xmc_see (77) - capabilities : fast,simple/complex,allresets
pmcB_xmc_see (79) - capabilities : fast,simple/complex,allresets
usb0_dev_see (80) - capabilities : fast,simple/complex,allresets
usb1_dev_see (81) - capabilities : fast,simple/complex,allresets
usb2_dev_see (82) - capabilities : fast,simple/complex,allresets
usb3_dev_see (83) - capabilities : fast,simple/complex,allresets
usb4_dev_see (84) - capabilities : fast,simple/complex,allresets
usb5_dev_see (85) - capabilities : fast,simple/complex,allresets

```

```
usb1_controller (87) - capabilities : fast,simple,allresets  
usb2_controller (88) - capabilities : fast,simple,allresets  
system (86) - capabilities : fast,simple/complex,allresets  
v2pmc2_check (90) - capabilities : fast,simple/complex,allresets  
v2pmc2_pmc_see (91) - capabilities : fast,simple/complex,allresets  
faultytest (98) - capabilities : fast,simple,allresets  
hangtest (99) - capabilities : fast,simple,allresets
```

Use 'kdiag help' to get more info.

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.

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
  serial: PASSED
  rtc: PASSED
  sysflash: PASSED
  cpld: PASSED
  temp_sensors: PASSED
  temperature: PASSED
  fnvram: PASSED
  ether_loop0: PASSED
  ether_loop1: PASSED
  ether_loop2: PASSED
  ether_loop3: PASSED
  voltage: PASSED
  sata_controller: PASSED
  vpd: PASSED
  eeprom: PASSED
  vme_check: PASSED
  pmcA_xmc_check: PASSED
  pmcB_xmc_check: PASSED
  usb1_controller: PASSED
  usb2_controller: PASSED
  system: PASSED

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

value = 0 = 0x0
->
```

3.3 Windows XP, XPe and Seven

The Windows XPe BSP for VM6050 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\CPLDUtils>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
    serial: PASSED
    rtc: PASSED
    sysflash: PASSED
    cp1d: PASSED
    temp_sensors: PASSED
    temperature: PASSED
    fnvram: PASSED
    ether_loop0: PASSED
    ether_loop1: PASSED
    ether_loop2: PASSED
    ether_loop3: PASSED
    voltage: PASSED
    sata_controller: PASSED
    vpd: PASSED
    eeprom: PASSED
    vme_check: PASSED
    pmcA_xmc_check: PASSED
    pmcB_xmc_check: PASSED
    usb1_controller: PASSED
    usb2_controller: PASSED
    system: FAILED
```

```
PASSED : 26
FAILED : 1
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, PCI, 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

The 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 30). 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. In addition to them, each test implements a dedicated "FAILED ONCE" flag which has to be reset separately. 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, the 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
V2PMC2	Dual slot PCI-X/PMC VME carrier from Kontron

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-ATD-Toulon@Kontron.com
Support: GSS-ATD-Toulon@Kontron.com

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