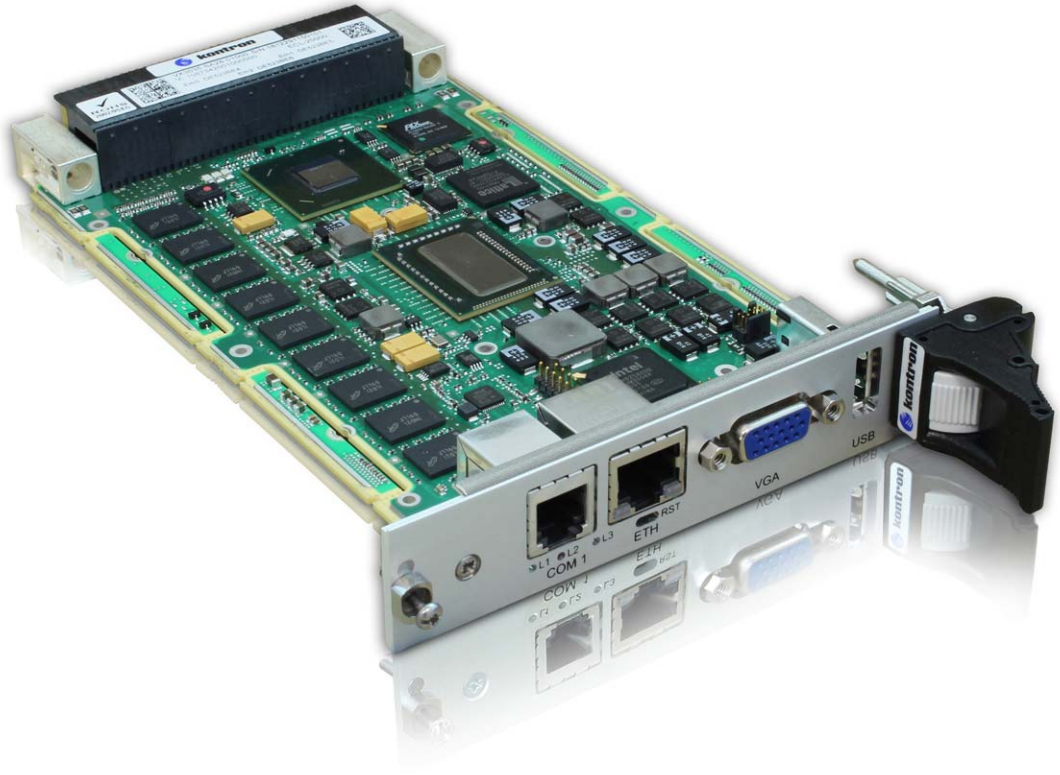
kontron

» VX3035 «



PBIT User's Guide

SD.DT.G08-2e - January 2013

Revision History

Publication Title:	VX3035 PBIT User's Guide	
Doc. ID:	SD.DT.G08-2e	
Rev.	Brief Description of Changes	Date of Issue
2e	“kdiag” command changed	01-2013
1e	System test enhanced, Hwmon test added	10-2012
0e	Initial Version	07-2012

Proprietary Note

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

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

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

Trademarks

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

Environmental Protection Statement

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

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

Environmental protection is a high priority with Kontron.

Kontron follows the DEEE/WEEE directive.

You are encouraged to return our products for proper disposal.

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

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

Conventions

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

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

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

ESD: This banner indicates an Electrostatic Sensitive Device.

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

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

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

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

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

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

For Your Safety

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

High Voltage Safety Instructions

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

Caution, Electric Shock!
Before installing a not hot-swappable Kontron product into a system always ensure that your mains power is switched off. This applies also to the installation of piggybacks. Serious electrical shock hazards can exist during all installation, repair and maintenance operations with this product. Therefore, always unplug the power cable and any other cables which provide external voltages before performing work.

Special Handling and Unpacking Instructions

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

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

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

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

General Instructions on Usage

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

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

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

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

Special care is necessary when handling or unpacking the product. Please consult the special handling and unpacking instruction on the previous page of this manual.

Table Of Contents

Chapter 1 - PBIT Overview

This document describes the PowerOn Built In Test (PBIT) for Kontron VX3035 boards.

The PBIT is an optional product available under the VX3035 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 VX3035 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 compromise 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 35.
- Simplified test result is also available in a 8-bit PLD register (register 0x2).

1.1 Related Documents

Hardware:

- VX3035 3U VPX User's Guide CA.DT.A95
- VX3035 Hardware Release Notes CA.DT.A96

BIOS:

- VX3035 AMI BIOS User Reference Manual SD.DT.F97

1.2 PBIT Installation and Activation

The PBIT software comes pre−installed in the system Flash, along with the EFI BIOS firmware, on the VX3035 boards.

The PBIT is an optional product that can be activated on any VX3035. Please contact Kontron support team for more information.

To install a new BIOS version including a new PBIT version please refer to the VX3035 BIOS User's Reference Manual - SD.DT.F97.

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 VX3035.

The list of tests to be executed can be displayed and modified by using the EFI Shell command "`kdiag`" (see section 1.5 page 8).

1.3.1 Configure the PBIT by command line

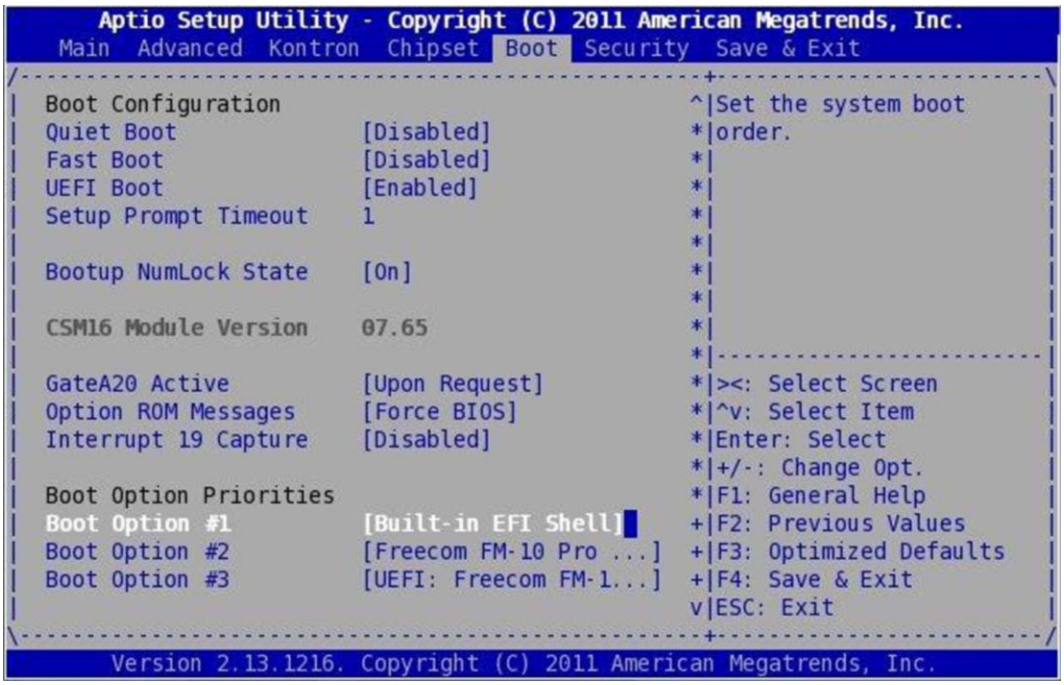
The following explains how to configure and execute the PBIT by a command line.

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

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

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

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



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

➢ Verify the PBIT version:

```
VX3035> kdiag version
PBIT VERSION 2.0 ID12170
```

> Launch the PBIT manually for verification:

```
VX3035> 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 "pcie_vpx_sw" (fast,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)   NOT EQUIPPED PASSED
PBIT "ether_loop4" (fast,simple)   NOT EQUIPPED PASSED
PBIT "voltage" (fast,simple)  PASSED
PBIT "hwmon" (fast,simple)  PASSED
PBIT "sata0_controler" (fast,simple)  PASSED
PBIT "sata1_controler" (fast,simple)   DISABLED PASSED
PBIT "vpd" (fast,simple)  PASSED
PBIT "eeprom" (fast,simple)  PASSED
PBIT "usb1_controller" (fast,simple)  PASSED
PBIT "usb2_controller" (fast,simple)  PASSED
PBIT "system" (fast,simple)  NOT SAVED CPLD INFOS
NOT SAVED SATA
NOT SAVED SMBUS
NOT SAVED PCIe
NOT SAVED USB
NOT SAVED BIOS SETUP INFOS
NOT SAVED ETH
PASSED
```

> Configure the “system” test:

By default the test named “**system**” is not “ready”. The end user should record the system configuration when the system is ready for this. All the equipments must be connected such as other CPU boards, VPX switch boards, Ethernet connections to external switches, USB devices, SATA devices. See section 2.10 page 25 “PBIT System Learn Test” for more details).

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

```
VX3035> kdiag system_learn

CPLD Version 0x6
DRAM size 8 GB
System Controller
Geographical Address 1
Storing CPLD infos system configuration ... Done
```

```
 Storing SATA system configuration ... Done

Device detected on SMBUS0, address = 0x30
Device detected on SMBUS0, address = 0x3C
Storing SMBUS device(s) system configuration ... Done

   Seg  Bus  Dev  Func
   ---  ---  ---  ----
    00   00   00    00 ==> Bridge Device - Host/PCI bridge
              Vendor 8086 Device 0104 Prog Interface 0
    00   00   01    00 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 0101 Prog Interface 0
    00   00   01    01 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 0105 Prog Interface 0
    00   00   02    00 ==> Display Controller - VGA/8514 controller
              Vendor 8086 Device 0116 Prog Interface 0
    00   00   16    00 ==> Simple Communications Controllers - Other communicati
              Vendor 8086 Device 1C3A Prog Interface 0
    00   00   19    00 ==> Network Controller - Ethernet controller
              Vendor 8086 Device 1502 Prog Interface 0
    00   00   1A    00 ==> Serial Bus Controllers - USB
              Vendor 8086 Device 1C2D Prog Interface 20
    00   00   1C    00 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 1C10 Prog Interface 0
    00   00   1C    04 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 1C18 Prog Interface 0
    00   00   1C    07 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 1C1E Prog Interface 0
    00   00   1D    00 ==> Serial Bus Controllers - USB
              Vendor 8086 Device 1C26 Prog Interface 20
    00   00   1F    00 ==> Bridge Device - PCI/ISA bridge
              Vendor 8086 Device 1C4F Prog Interface 0
    00   00   1F    02 ==> Mass Storage Controller - UNDEFINED
              Vendor 8086 Device 1C03 Prog Interface 1
    00   00   1F    03 ==> Serial Bus Controllers - System Management Bus
              Vendor 8086 Device 1C22 Prog Interface 0
    00   00   1F    06 ==> Data Acquisition & Signal Processing Controllers - Ot
              Vendor 8086 Device 1C24 Prog Interface 0
    00   02   00    00 ==> Bridge Device - PCI/PCI bridge
              Vendor 10B5 Device 8609 Prog Interface 0
    00   02   00    01 ==> Base System Peripherals - Other system peripheral
              Vendor 10B5 Device 8609 Prog Interface 0
    00   03   01    00 ==> Bridge Device - PCI/PCI bridge
              Vendor 10B5 Device 8609 Prog Interface 0
    00   04   00    00 ==> Bridge Device - Other bridge type
              Vendor 10B5 Device 8609 Prog Interface 0
    00   06   00    00 ==> Network Controller - Ethernet controller
              Vendor 8086 Device 1510 Prog Interface 0
    00   06   00    01 ==> Network Controller - Ethernet controller
              Vendor 8086 Device 1510 Prog Interface 0

Storing system configuration ... Done

USB connected device(s) :
1 Drive, 2 Hubs
```

```
MassStorage Device name (0) : Freecom FM-10 Pro 2
Storing USB system configuration ... Done

BIOS Setup Checksum : 71307
BIOS Version : ID12275
Storing BIOS Setup system configuration ... Done

ETH2 connected device speed 1000Mb/s
Storing ETH system configuration ... Done
```

➢ Check the PBIT results:

```
VX3035> 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 : pcie_vpx_sw (fast,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 : ether_loop4 (fast,simple)
PASSED : voltage (fast,simple)
PASSED : hwmon (fast,simple)
PASSED : sata0_controler (fast,simple)
PASSED : sata1_controler (fast,simple)
PASSED : vpd (fast,simple)
PASSED : eeprom (fast,simple)
PASSED : usb1_controller (fast,simple)
PASSED : usb2_controller (fast,simple)
PASSED : system (fast,simple)

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

1.3.2 Configure the PBIT to run automatically

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

> Configure the PBIT to be launched at boot time:

The automatic start is activated using the environment variable “**bootcmd**”.

```
VX3035> set bootcmd "kdiag run"
```

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

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

```
VX3035> set bootdelay 1
```

> Verify:

```
VX3035> set
    bootcmd   : kdiag run
    bootdelay : 1
```

> Then reset the system:

```
VX3035> 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**”:

```
VX3035> set bootdontexit            1
```

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

```
VX3035> set startupAuto      1
VX3035> set startupdelay     1
```

Then plug a USB device including in the top directory the script file named “startup.nsh” including the wanted shell commands.

1.4 Synthetic PBIT Result

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

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

Under the BIOS EFI shell use the following command:

```
VX3035> kpld -r 2
READ : @0x2 = 0x1
```

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

Test Fail Number 1.. 128							run
7	6	5	4	3	2	1	0

Bit 0: 0 = NOT RUN 1 = ALL RUN

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

➢ Examples:

0x00 => PBIT not run

0x03 => ALL Tests run and Test 1 FAILED

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

0x01 => ALL Tests run and PASSED

To identify a PBIT test by its number , use the command “`kdiag [PBITnumber]`”
Example:

```
VX3035> kdiag 16
 serial (16) - Checks the serial line COM2
   capabilities : fast,simple/complex
   run mode 1   : fast,simple
```

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

1.5 PBIT Tests List

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

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

1.5.1 Selected Tests List

The default selected tests list contains all the diagnostics that can be run without any specific equipment. All the tests have been designed to be safe for the system containing a VX3035. No signal on any connector will be modified during the default test execution.

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

```
VX3035> 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
 mem_pattern3 (8) - Checks Memory/ECC using pattern 0xAAAAAAAA
   capabilities : slow/fast,simple
   run mode 1   : slow,simple
 mem_pattern4 (9) - Checks Memory/ECC using pattern 0x00000000
   capabilities : slow/fast,simple
   run mode 1   : slow,simple
 pcie_vpx_sw (13) - Checks the pciExpress backplane switch
   capabilities : fast,simple/complex
   run mode 1   : fast,simple
 serial (16) - Checks the serial line COM2
   capabilities : fast,simple/complex
   run mode 1   : fast,simple
 rtc (20) - Checks the RTC time
   capabilities : fast,simple
   run mode 1   : fast,simple
 sysflash (22) - Checks the BIOS rescue in system flash
   capabilities : fast,simple
   run mode 1   : fast,simple
 cpld (24) - Checks PLD, GeoAddress, watchdog
   capabilities : fast,simple/complex
   run mode 1   : fast,simple
temp_sensors (31) - Checks if all temperature sensors are detected.
   capabilities : fast,simple
   run mode 1   : fast,simple
```

```
temperature (32) - Checks if temperatures are OK.
   capabilities : fast,simple
   run mode 1   : fast,simple
 fnvram (40) - Checks F-NVRAM device.
   capabilities : fast,simple/complex
   run mode 1   : fast,simple
 ether_loop0 (55) - Checks 82579 (Lewisville) Gigabit Ethernet in Loopback mode
   capabilities : fast,simple/complex
   run mode 1   : fast,simple
 ether_loop1 (56) - Checks 82580 Gigabit Rear Interface 1 in Loopback mode
   capabilities : fast,simple/complex
   run mode 1   : fast,simple
 ether_loop2 (57) - Checks 82580 Gigabit Rear Interface 2 in Loopback mode
   capabilities : fast,simple/complex
   run mode 1   : fast,simple
 ether_loop3 (58) - Checks 82580 Gigabit Rear Interface 3 in Loopback mode
   capabilities : fast,simple/complex
   run mode 1   : fast,simple
 ether_loop4 (59) - Checks 82580 Gigabit Rear Interface 4 in Loopback mode
   capabilities : fast,simple/complex
   run mode 1   : fast,simple
 voltage (60) - Checks the voltage sensors & value
   capabilities : fast,simple
   run mode 1   : fast,simple
 hwmon (61) - Checks hardware monitoring, temperature and voltage sensors
   capabilities : fast,simple,allresets
   run mode 1   : fast,simple,allresets
 sata0_controler (68) - Checks sata0 controller
   capabilities : fast,simple/complex
   run mode 1   : fast,simple
 sata1_controler (69) - Check sata1 controller
   capabilities : fast,simple/complex
   run mode 1   : fast,simple
 vpd (70) - Checks VPD data required for board operation.
   capabilities : fast,simple
   run mode 1   : fast,simple
 eeprom (71) - Checks User EEPROM (0xA2)
   capabilities : fast,simple/complex
   run mode 1   : fast,simple
 usb1_controller (86) - Check first PCH usb controller
   capabilities : fast,simple
   run mode 1   : fast,simple
 usb2_controller (87) - Check second PCH usb controller
   capabilities : fast,simple
   run mode 1   : fast,simple
 system (88) - Checks system configuration PCIe,SATA,USB,ETH stability
   capabilities : fast,simple/complex
   run mode 1   : fast,simple

VX3035>
```

To run the default PBIT, enter the command:

```
VX3035> 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
 mem_addrpat (4) - Checks Memory/ECC using address pattern (offset)
   capabilities : slow/fast,simple
 mem_addrpat2 (5) - Checks Memory/ECC using address pattern (~offset)
   capabilities : slow/fast,simple
 smbus0 (26) - Check SMBUS0 between PLD and backplane
   capabilities : fast,simple/complex
 smbus1 (27) - Check SMBUS1 between PLD and backplane
   capabilities : fast,simple/complex
 ether_link0 (50) - Checks the link status is "UP" on 82579 (Lewisville) Gigabit Network.
   capabilities : fast,simple/complex
 ether_link1 (51) - Checks the link status is "UP" on 82580 GigaEth Rear 1
   capabilities : fast,simple/complex
 ether_link2 (52) - Checks the link status is "UP" on 82580 GigaEth Rear 2
   capabilities : fast,simple/complex
 ether_link3 (53) - Checks the link status is "UP" on 82580 GigaEth Rear 3
   capabilities : fast,simple/complex
 ether_link4 (54) - Checks the link status is "UP" on 82580 GigaEth Rear 4
   capabilities : fast,simple/complex
 sata0_dev_see (63) - Checks if a SATA0 disk on rear P1 is present
   capabilities : fast,simple/complex
 sata1_dev_see (64) - Checks if a SATA1 disk on rear P2 is present
   capabilities : fast,simple/complex
 sata2_dev_see (65) - Checks if a SATA2 disk on rear P2 is present
   capabilities : fast,simple/complex
 sata3_dev_see (66) - Checks if a SATA3 disk on rear P1 is present
   capabilities : fast,simple/complex
 sata4_dev_see (67) - Checks if a SATA mezzanine device is present
   capabilities : fast,simple/complex
 usb0_dev_see (80) - Check if a USB0 device on front panel is present
   capabilities : fast,simple/complex
 usb1_dev_see (81) - Check if a USB1 device on mezzanine is present
   capabilities : fast,simple/complex
 usb2_dev_see (82) - Check if a USB2 device on rear P1 is present
   capabilities : fast,simple/complex
 usb3_dev_see (83) - Check if a USB3 device on rear P1 is present
   capabilities : fast,simple/complex
```

```
usb4_dev_see (84) - Check if a USB4 device on rear P2 is present
  capabilities : fast,simple/complex
usb5_dev_see (85) - Check if a USB5 device on rear P2 is present
  capabilities : fast,simple/complex
faultytest (98) - A dummy test that returns FAIL
  capabilities : fast,simple
hangtest (99) - A dummy test that will hang
  capabilities : fast,simple
```

The unselected tests list contains:

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

All the USB device detection tests are limited to USB Mass storage devices. They will not detect USB keyboard or mouse.

1.6 PBIT Execution Time

The default PBIT (see section 1.5.1 page 8) runs in about 14 seconds for a 4 GB DRAM board and 24 seconds for a 8 GB DRAM board.

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

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

Test name	Test ID number	Execution Time
mem_data (fast,simple)	1	300 ms
mem_addr (fast,simple)	2	3.2s
mem_bitflip (slow,simple)	3	5.6s
mem_addrpat (slow,simple)	4	4.2s
mem_addrpat2 (slow,simple)	5	4.4s
mem_pattern1 (slow,simple)	6	4.2s
mem_pattern2 (slow,simple)	7	4.1s
mem_pattern3 (slow,simple)	8	4.2s
mem_pattern4 (slow,simple)	9	4.1s
pcie_vpx_sw (fast,simple)	13	<300 ms
serial (fast,simple)	16	<300 ms
rtc (fast,simple)	20	<300 ms
sysflash (fast,simple)	22	<300 ms
cpld (fast,simple)	24	<300 ms
temp_sensors (fast,simple)	31	<300 ms
temperature (fast,simple)	32	<300 ms
fnvram (fast,simple)	40	400 ms
ether_link0 (fast,simple)	50	<300 ms
ether_link1 (fast,simple)	51	<300 ms
ether_link2 (fast,simple)	52	<300 ms
ether_link3 (fast,simple)	53	<300 ms
ether_loop0 (fast,simple)*	55	300 ms
ether_loop0 (fast,complex)*	55	1.2s
ether_loop1 (fast,simple)*	56	300 ms
ether_loop1 (fast,complex)*	56	1.2s
ether_loop2 (fast,simple)*	57	300 ms
ether_loop2 (fast,complex)*	57	1.2s
ether_loop3 (fast,simple)*	58	300 ms
ether_loop3 (fast,complex)*	58	1.2s
ether_loop4 (fast,simple)*	59	300 ms
ether_loop4 (fast,complex)*	59	1.2s
voltage (fast,simple)	60	<300 ms
hwmon (fast,simple)	61	700 ms

Test name	Test ID number	Execution Time
sata0_dev_see (fast,simple)	63	<300 ms
Sata1_dev_see (fast,simple)	64	<300 ms
Sata2_dev_see (fast,simple)	65	<300 ms
Sata3_dev_see (fast,simple)	66	<300 ms
Sata4_dev_see (fast,simple)	67	<300 ms
sata0_controler (fast,simple)	68	<300 ms
sata1_controler (fast,simple)	69	<300 ms
vpd (fast,simple)	70	<300 ms
eeprom (fast,simple)	71	400 ms
usb0_dev_see (fast,simple)	80	<300 ms
usb1_dev_see (fast,simple)	81	<300 ms
usb2_dev_see (fast,simple)	82	<300 ms
usb3_dev_see (fast,simple)	83	<300 ms
usb4_dev_see (fast,simple)	84	<300 ms
usb5_dev_see (fast,simple)	85	<300 ms
usb1_controller (fast,simple)	86	<300 ms
usb2_controller (fast,simple)	87	<300 ms
system (fast,simple)**	88	~2.5s

* In simple mode (default) ether_loop*x* tests only internal registers. Loopback test is done in complex mode only because execution time is not deterministic for eth0 and it needs a link for the 1000BASE-BX eth1, 2, 3, 4 interfaces.

** The execution time for the system test depends on the configuration and if the test is successful or not. In case of failure on Ethernet links the execution time may reach 5s.

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

```
VX3035> 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

Record System Configuration for system Test:
  kdiag system_learn

Edit System Configuration for system Test:
(cf documentation for details)
  kdiag system_edit

Clear System Configuration for system Test:
  kdiag system_clear
  kdiag system_clear_pcie
  kdiag system_clear_usb
  kdiag system_clear_sata
  kdiag system_clear_eth
  kdiag system_clear_cpld
  kdiag system_clear_bios
  kdiag system_clear_smbus

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.

```
VX3035> kdiag
PBITs configured to run from command line :
 mem_data (1) - Checks Memory/ECC data lines
   capabilities : fast,simple
   run mode 1   : fast,simple
 mem_addr (2) - Checks Memory/ECC address lines
   capabilities : fast,simple
   run mode 1   : fast,simple
 mem_pattern1 (6) - Checks Memory/ECC using pattern 0xFFFFFFFF
   capabilities : slow/fast,simple
   run mode 1   : slow,simple
 mem_pattern2 (7) - Checks Memory/ECC using pattern 0x55555555
   capabilities : slow/fast,simple
   run mode 1   : slow,simple
(...)
Other PBITs available but not yet configured :
 mem_bitflip (3) - Checks Mem/ECC using bit-flip pattern ((1 << (offset % 32))
   capabilities : slow/fast,simple
 mem_addrpat (4) - Checks Memory/ECC using address pattern (offset)
   capabilities : slow/fast,simple
 mem_addrpat2 (5) - Checks Memory/ECC using address pattern (~offset)
   capabilities : slow/fast,simple
 smbus0 (26) - Check SMBUS0 between PLD and backplane
   capabilities : fast,simple/complex
 smbus1 (27) - Check SMBUS1 between PLD and backplane
   capabilities : fast,simple/complex
 ether_link0 (50) - Checks the link status is "UP" on 82579 (Lewisville) Gigabit Network.
   capabilities : fast,simple/complex
 ether_link1 (51) - Checks the link status is "UP" on 82580 GigaEth Rear 1
   capabilities : fast,simple/complex
(...)

Use ´help kdiag´ to get more info.
VX3035>
```

2.3 Execute the PBIT from the Command Line

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

```
VX3035> 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 "pcie_vpx_sw" (fast,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)   NOT EQUIPPED PASSED
PBIT "ether_loop4" (fast,simple)   NOT EQUIPPED PASSED
PBIT "voltage" (fast,simple)  PASSED
PBIT "hwmon" (fast,simple)  PASSED
PBIT "sata0_controler" (fast,simple)  PASSED
PBIT "sata1_controler" (fast,simple)   DISABLED PASSED
PBIT "vpd" (fast,simple)  PASSED
PBIT "eeprom" (fast,simple)  PASSED
PBIT "usb1_controller" (fast,simple)  PASSED
PBIT "usb2_controller" (fast,simple)  PASSED
PBIT "system" (fast,simple)   NOT SAVED PCIe  USB  SATA  ETH PASSED
VX3035>
```

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

```
VX3035> kdiag run <PBIT number | PBIT name ...>
```

➢ For example:

```
VX3035> kdiag run usb1_controller sysflash
PBIT "usb1_controller" (fast,simple)  PASSED
PBIT "sysflash" (fast,simple)  PASSED
```

That is equivalent to:

```
VX3035> 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 the PBIT in Loop Mode

To run the PBIT in loop mode, enter:

```
VX3035 > kdiag run loop <count>
```

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

To run a single test in loop mode, enter:

```
VX3035> kdiag run loop <count> <PBIT number | PBIT name ...>
```

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

```
VX3035> kdiag run loop 10 55
```

It is equivalent to enter:

```
VX3035> kdiag run loop 10 ether_loop0
```

2.5 Get the PBIT Results

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

```
VX3035> 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 : pcie_vpx_sw (fast,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 : ether_loop4 (fast,simple)
 PASSED : voltage (fast,simple)
 PASSED : hwmon (fast,simple)
 PASSED : sata0_controler (fast,simple)
 PASSED : sata1_controler (fast,simple)
 PASSED : vpd (fast,simple)
 PASSED : eeprom (fast,simple)
 PASSED : usb1_controller (fast,simple)
 PASSED : usb2_controller (fast,simple)
 PASSED : system (fast,simple)

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

2.6 Clear the PBIT Results

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

> To clear the PBIT results enter:

```
VX3035> kdiag clrstat
```

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

```
VX3035> kdiag clrallstat
```

2.7 Configure the PBIT Tests List to Execute

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

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

2.7.1 Run mode parameters

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

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

  `haltonfail`: halt immediately (hang) if test fails

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

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

- `SPEED` flag (can NOT be mixed)

  `slow`: run in slow mode (full testing)

  `fast`: run in fast mode (fast testing)

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

- `CONFIG` flag (can NOT be mixed)

  `simple`: run in simple mode (no external hardware)

  `complex`: run in complex mode (needs external hardware)

  Complex mode requires external devices to the VX3035 and is reserved for the factory tests.

2.7.2 Adding a Test to the Current Run List

To add a test to the run list, enter:

```
VX3035> 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:

```
VX3035> kdiag cfg default 16
```

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

```
VX3035> kdiag cfg promptonfail 16
```

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

```
VX3035> kdiag cfg  complex,promptonfail 16
```

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

```
VX3035> 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:

```
VX3035> kdiag cfg delete 16
```

Verify:

```
VX3035> kdiag 16
PBIT "16" is not configured to run from command line
```

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

```
VX3035> kdiag deleteall
```

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

```
VX3035> kdiag run
WARNING: No PBIT will be run because run list is empty
```

2.7.4 Changing Flags for all the Current Tests

To change all the “current tests” with the promptonfail flag, enter:

```
VX3035> kdiag cfg promptonfail
Configuration of current tests in asked mode:
command,fast,simple,promptonfail,allresets
List of test to configure
--> ether_link0 (50) - Checks the link status is "UP" on 82579 (Lewisville) Gigabit Network.
--> ether_link1 (51) - Checks the link status is "UP" on 82580 GigaEth Rear 1
--> ether_link2 (52) - Checks the link status is "UP" on 82580 GigaEth Rear 2
--> ether_link3 (53) - Checks the link status is "UP" on 82580 GigaEth Rear 3
--> ether_link4 (54) - Checks the link status is "UP" on 82580 GigaEth Rear 4

VX3035>
```

Verify for test 50:

```
VX3035> kdiag 50
ether_link0 (50) - Checks the link status is "UP" on 82579 (Lewisville) Gigabit Network.
   capabilities : fast,simple/complex,allresets
   run mode 1   : fast,simple,promptonfail,allresets
```

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:

```
VX3035> kdiag default
```

2.8 Run the PBIT in Silent Mode

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

```
VX3035> kdiag silent
PBIT set in silent mode
```

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

```
VX3035> 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:

```
VX3035> kdiag version
PBIT VERSION 2.0 ID12170
```

2.10 PBIT System Test

Recording the system configuration:

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

It records:

- All the detected PCI devices (the list is visible with the BIOS shell command "`pci`").
- PCIe devices infos (vendorID, deviceID, ClassCode) from detected PCI devices.
- PCIe link width and speed for PCI/PCI bridge devices.
- SATA information (Port Connected, disk name, disk size).
- USB information (Port connected, Keyboard, mice, Mass Storage device name(s)).
- Ethernet information (Port link up, speed).
- BIOS information: BIOS ID and Checksum (checksum of BIOS setup).
- Information from CPLD registers: CPLD Version, DRAM size, system controller status, geographical address.
- SMBUS connected devices addresses on backplanes SMBUS0 and SMBUS1

Example :

```
VX3035> kdiag system_learn
CPLD Version 0x6
DRAM size 8 GB
System Controller
Geographical Address 1
Storing CPLD infos system configuration ... Done

Storing SATA system configuration ... Done

Device detected on SMBUS0, address = 0x30
Device detected on SMBUS0, address = 0x3C
Storing SMBUS device(s) system configuration ... Done

   Seg  Bus  Dev  Func
   ---  ---  ---  ----
    00   00   00    00 ==> Bridge Device - Host/PCI bridge
              Vendor 8086 Device 0104 Prog Interface 0
    00   00   01    00 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 0101 Prog Interface 0
    00   00   01    01 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 0105 Prog Interface 0
    00   00   02    00 ==> Display Controller - VGA/8514 controller
              Vendor 8086 Device 0116 Prog Interface 0
    00   00   16    00 ==> Simple Communications Controllers - Other communicati
              Vendor 8086 Device 1C3A Prog Interface 0
    00   00   19    00 ==> Network Controller - Ethernet controller
              Vendor 8086 Device 1502 Prog Interface 0
```

```
    00   00   1A    00 ==> Serial Bus Controllers - USB
              Vendor 8086 Device 1C2D Prog Interface 20
    00   00   1C    00 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 1C10 Prog Interface 0
    00   00   1C    04 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 1C18 Prog Interface 0
    00   00   1C    07 ==> Bridge Device - PCI/PCI bridge
              Vendor 8086 Device 1C1E Prog Interface 0
    00   00   1D    00 ==> Serial Bus Controllers - USB
              Vendor 8086 Device 1C26 Prog Interface 20
    00   00   1F    00 ==> Bridge Device - PCI/ISA bridge
              Vendor 8086 Device 1C4F Prog Interface 0
    00   00   1F    02 ==> Mass Storage Controller - UNDEFINED
              Vendor 8086 Device 1C03 Prog Interface 1
    00   00   1F    03 ==> Serial Bus Controllers - System Management Bus
              Vendor 8086 Device 1C22 Prog Interface 0
    00   00   1F    06 ==> Data Acquisition & Signal Processing Controllers - Ot
              Vendor 8086 Device 1C24 Prog Interface 0
    00   02   00    00 ==> Bridge Device - PCI/PCI bridge
              Vendor 10B5 Device 8609 Prog Interface 0
    00   02   00    01 ==> Base System Peripherals - Other system peripheral
              Vendor 10B5 Device 8609 Prog Interface 0
    00   03   01    00 ==> Bridge Device - PCI/PCI bridge
              Vendor 10B5 Device 8609 Prog Interface 0
    00   04   00    00 ==> Bridge Device - Other bridge type
              Vendor 10B5 Device 8609 Prog Interface 0
    00   06   00    00 ==> Network Controller - Ethernet controller
              Vendor 8086 Device 1510 Prog Interface 0
    00   06   00    01 ==> Network Controller - Ethernet controller
              Vendor 8086 Device 1510 Prog Interface 0

Storing system configuration ... Done

USB connected device(s) :
1 Drive, 2 Hubs

MassStorage Device name (0) : Freecom FM-10 Pro 2
Storing USB system configuration ... Done

BIOS Setup Checksum : 7130?
BIOS Version : ID12275
Storing BIOS Setup system configuration ... Done

ETH2 connected device speed 1000Mb/s
Storing ETH system configuration ... Done
```

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 following is an example showing several detected failures:

```
VX3035> kdiag run system
PBIT "system" (fast,simple)
 SMBUS Devices Infos State Failed =>
Recorded addresse(s) on SMBUS0      Detected addresse(s) on SMBUS0
0x30                                 0x30
0x3C

ERROR => Device(s)Missing
Expected (bus,dev,func)    Found (bus,dev,func)
-----------------------------------------------
           00 00 00                00 00 00
           00 01 00                00 01 00
           00 01 01                00 01 01
           00 02 00                00 02 00
           00 16 00                00 16 00
           00 19 00                00 19 00
           00 1A 00                00 1A 00
           00 1C 00                00 1C 00
           00 1C 04                00 1C 04
           00 1C 07                00 1C 07
           00 1D 00                00 1D 00
           00 1F 00                00 1F 00
           00 1F 02                00 1F 02
           00 1F 03                00 1F 03
           00 1F 06                00 1F 06
           02 00 00                02 00 00
           02 00 01                02 00 01
           03 01 00                03 01 00
           04 00 00                06 00 00
           06 00 00                06 00 01
           06 00 01

USB Config FAILED => Device(s) list changed =>
Recorded device(s)                        Detected device(s)
1 Drive, 2 Hubs                                 2 Hubs

FAILED USB mass Storage name list changed =>
Recorded mass storage name:        Detected mass storage name:
Freecom FM-10 Pro 2
```

```
 ERROR USB port connected Changed =>
Recorded connected port list:      Detected connected port list:
PORT(s):                           PORT(s):
0

Ethernet Connected State FAILED =>
     Recorded Link State                        Detected Link State
ETH0 Not Connected                           Not Connected
ETH1 Not Connected                           Not Connected
ETH2 Link UP speed 1000 Mb/s       Not Connected
FAILED
```

» Editing the system configuration:

A menu is available to edit the system configuration and choose the items to be checked. To enter the menu, type:

```
VX3035>kdiag system_edit
```

The menu is displayed only if a system configuration has been previously recorded with the "`kdiag system_learn`" command. Else, messages inform that no configuration has been recorded yet:

```
VX3035>kdiag system_edit
NOT SAVED CPLD INFOS
NOT SAVED SATA
NOT SAVED SMBUS
NOT SAVED PCIe
NOT SAVED USB
NOT SAVED BIOS INFOS
NOT SAVED ETH
```

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

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

The following shows a `system_edit` session ignoring the System Controller status information from CPLD, the SATA port 4, all the SMBUS0 and SMBUS1 devices, all information for PCI device 1, the ClassCode information for PCI device 16, the ethernet interfaces 3 and 4:

```
VX3035> kdiag system_edit
CPLD infos
Recorded CPLD infos :
Syscon (s): 1
Cpld Version (c): 0x5
Geographical Address (g): 1
DRAM size (d): 8GB
```

```
Use character between brackets to ignore element
´n´ to go to next edit menu
´q´ to quit edit menu
s
Syscon ignored

Use character between brackets to ignore element
´n´ to go to next edit menu
´q´ to quit edit menu
n

SATA
Recorded SATA infos :
SATA PORT 0      EMPTY     (0.0GB)
SATA PORT 1      EMPTY     (0.0GB)
SATA PORT 2      EMPTY     (0.0GB)
SATA PORT 3      EMPTY     (0.0GB)
SATA PORT 4      EMPTY     (0.0GB)

Type port number to be ignored
´n´ to go to next edit menu
´q´ to quit edit menu
4
SATA port 4 ignored

Type port number to be ignored
´n´ to go to next edit menu
´q´ to quit edit menu
n

SMBUS0
Device(s) address(es) recorded on SMBUS0 :
0x30 (0)
Use number between () to ignore device at corresponding address
´i´ to ignore all smbus0 devices
´n´ to go to next edit menu
´q´ to quit edit menu
i
SMBUS0 IGNORED

SMBUS1
Device(s) address(es) recorded on SMBUS1 :
Use number between brackets to ignore device at corresponding address
´i´ to ignore all smbus1 devices
´n´ to go to next edit menu
´q´ to quit edit menu
i
SMBUS1 IGNORED
```

```
PCIe
Recorded PCIe devices infos :
Bus Dev Func
                Device Number 0
Bus Dev Func  Vendor ID:  Device ID:  Classcode:
00  00   00   8086        0104        060000
                Device Number 1
Bus Dev Func  Vendor ID:  Device ID:  Classcode:  LinkStatus:
00  01   00   8086        0101        060400      4097
                Device Number 2
Bus Dev Func  Vendor ID:  Device ID:  Classcode:  LinkStatus:
00  01   01   8086        0105        060400      4161
                Device Number 3
Bus Dev Func  Vendor ID:  Device ID:  Classcode:
00  02   00   8086        0116        030000
                Device Number 4
Bus Dev Func  Vendor ID:  Device ID:  Classcode:
00  16   00   8086        1C3A        078000
                Device Number 5
Bus Dev Func  Vendor ID:  Device ID:  Classcode:
00  19   00   8086        1502        020000
                Device Number 6
Bus Dev Func  Vendor ID:  Device ID:  Classcode:
00  1A   00   8086        1C2D        0C0320
                Device Number 7
Bus Dev Func  Vendor ID:  Device ID:  Classcode:  LinkStatus:
00  1C   00   8086        1C10        060400      4097
                Device Number 8
Bus Dev Func  Vendor ID:  Device ID:  Classcode:  LinkStatus:
00  1C   04   8086        1C18        060400      61474
                Device Number 9
Bus Dev Func  Vendor ID:  Device ID:  Classcode:  LinkStatus:
00  1C   07   8086        1C1E        060400      4097
                Device Number 10
Bus Dev Func  Vendor ID:  Device ID:  Classcode:
00  1D   00   8086        1C26        0C0320
                Device Number 11
Bus Dev Func  Vendor ID:  Device ID:  Classcode:
00  1F   00   8086        1C4F        060100
                Device Number 12
Bus Dev Func  Vendor ID:  Device ID:  Classcode:
00  1F   02   8086        1C03        010601
                Device Number 13
Bus Dev Func  Vendor ID:  Device ID:  Classcode:
00  1F   03   8086        1C22        0C0500
                Device Number 14
Bus Dev Func  Vendor ID:  Device ID:  Classcode:
00  1F   06   8086        1C24        118000
                Device Number 15
Bus Dev Func  Vendor ID:  Device ID:  Classcode:  LinkStatus:
02  00   00   10B5        8609        060400      65
                Device Number 16
Bus Dev Func  Vendor ID:  Device ID:  Classcode:
02  00   01   10B5        8609        088000
```

```
                Device Number 17
Bus Dev Func  Vendor ID:  Device ID:  Classcode:  LinkStatus:
03  01   00   10B5        8609        060400      1
                Device Number 18
Bus Dev Func  Vendor ID:  Device ID:  Classcode:
06  00   00   8086        1510        020000
                Device Number 19
Bus Dev Func  Vendor ID:  Device ID:  Classcode:
06  00   01   8086        1510        020000

Use device number to be edited
´n´ to go to next edit menu
´q´ to quit edit menu
1
´a´ to ignore all infos (Vendor ID, Device ID, Classcode, Link status) of PCIe device
number 1
´v´ to ignore Vendor ID
´d´ to ignore Device ID
´c´ to ignore Classcode
´l´ to ignore Link status
´b´ to go back to previous menu
a
PCIe device number 1 entirely ignored

´a´ to ignore all infos (Vendor ID, Device ID, Classcode, Link status) of PCIe device
number 1
´v´ to ignore Vendor ID
´d´ to ignore Device ID
´c´ to ignore Classcode
´l´ to ignore Link status
´b´ to go back to previous menu
b
Use device number to be edited
´n´ to go to next edit menu
´q´ to quit edit menu
16
´a´ to ignore all infos (Vendor ID, Device ID, Classcode, Link status) of PCIe device
number 16
´v´ to ignore Vendor ID
´d´ to ignore Device ID
´c´ to ignore Classcode
´l´ to ignore Link status
´b´ to go back to previous menu
c
PCIe device number 16 : Classcode ignored

´a´ to ignore all infos (Vendor ID, Device ID, Classcode, Link status) of PCIe device
number 16
´v´ to ignore Vendor ID
´d´ to ignore Device ID
´c´ to ignore Classcode
´l´ to ignore Link status
´b´ to go back to previous menu
b
Use device number to be edited
´n´ to go to next edit menu
```

```
´q´ to quit edit menu
n

USB
Recorded USB infos :

Connected port(s) number(s) :
0

Recorded mass storage name(s)
Freecom FM-10 Pro 2

USB connected device(s) :
1 Drive, 2 Hubs

Type USB port number to be ignored
´m´ to ignore mass storage names
´c´ to ignore connected devices list
´n´ to go to next edit menu
´q´ to quit edit menu
n

BIOS
Recorded BIOS infos :
BIOS Setup Checksum (c): 71307
BIOS Version (v): ID12270

Use character between brackets to ignore element
´n´ to go to next edit menu
´q´ to quit edit menu
n

ETH
Recorded ETH infos :
ETH0 Not Connected
ETH1 Not Connected
ETH2 Not Connected
ETH3 Not Equipped
ETH4 Not Equipped

Type Eth Interface number to be ignored
´q´ to quit edit menu
3
ETH port 3 ignored

Type Eth Interface number to be ignored
´q´ to quit edit menu
4
ETH port 4 ignored

Type Eth Interface number to be ignored
´q´ to quit edit menu
Q
```

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.

After system edition, there is no possibility to remove IGNORED flags one by one in order to test items again. The only way is to clear the system configuration and/or to learn the system configuration again ("kdiag system_learn" command). All the IGNORED flags will be removed.

Clearing the system configuration

Commands are available to clear the system configuration entirely or partially.

To clear the entire configuration, enter:

```
VX3035>kdiag  system_clear
```

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

```
"NOT SAVED CPLD INFOS
 NOT SAVED SATA
 NOT SAVED SMBUS
 NOT SAVED PCIe
 NOT SAVED USB
 NOT SAVED BIOS INFOS
 NOT SAVED ETH"
```

The system test result will be PASSED.

To clear the recorded PCIe configuration, enter :

```
VX3035>kdiag  system_clear_pcie
```

Then, if a system test if performed, the PCIe devices will not be checked and the message "NOT SAVED PCIE" will be printed. The PCIe system test result will be PASSED.

To clear the recorded SATA configuration, enter :

```
VX3035>kdiag  system_clear_sata
```

Then, if a system test if performed, the SATA ports and devices will not be checked and the message "NOT SAVED SATA" will be printed. The SATA system test result will be PASSED.

To clear the recorded USB configuration, enter :

```
VX3035>kdiag  system_clear_usb
```

Then, if a system test if performed, the USB ports and devices will not be checked and the message "NOT SAVED USB" will be printed. The USB system test result will be PASSED.

To clear the recorded Ethernet configuration, enter :

```
VX3035>kdiag  system_clear_eth
```

Then, if a system test if performed, the Ethernet interfaces will not be checked and the message "NOT SAVED ETH" will be printed. The Ethernet test result will be PASSED.

To clear the recorded SMBUS configuration, enter :

```
VX3035>kdiag  system_clear_smbus
```

Then, if a system test if performed, the SMBUS devices will not be checked and the message “NOT SAVED SMBUS” will be printed. The SMBUS test result will be PASSED.

To clear the recorded BIOS information, enter :

```
VX3035>kdiag  system_clear_bios
```

Then, if a system test if performed, the BIOS information will not be checked and the message “NOT SAVED BIOS INFOS” will be printed. BIOS test result will be PASSED.

To clear the recorded CPLD information, enter :

```
VX3035>kdiag system_clear_cpld
```

Then, if a system test if performed, the CPLD information will not be checked and the message “NOT SAVED CPLD INFOS” will be printed. CPLD test result will be PASSED.

1. To summarize, before running the system test, the specific commands must be performed in this order: system_learn then system_clear_xxx or system_edit depending the chosen test granularity. 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.

Chapter 3 - PBIT and OS Interfaces

The PBIT synthetic result and PBIT detailed results are accessible under the Operating System delivered with the VX3035 board.

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

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

3.1 Linux

3.1.1 Linux Synthetic PBIT Result

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

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

➢ Example:

```
# cpldtool -a
Reg 0x0 - CPLD_ID = 0x01
            CPLD_ID=0x0
            CPLD_Debug=0x0
            CPLD_Version=0x1
Reg 0x1 - PCB_ID = 0x35
Reg 0x2 - FIRM_PWON = 0x00
Reg 0x3 - PWON_STATUS = 0x00
Reg 0x4 - PWR_RST_CONFIG = 0x23
            PWRON_MODE=0x0
            PLTRST_to_PERST3U_INHIB=0x0
            Alarm_inhib=0x1
            PLTRST_to_PERST6U_INHIB=0x0
(....)
```

3.1.2 Linux Detailed PBIT Result

The detailed PBIT results are accessible under Linux with the “`sysvartool`” facility.

Use the “`-A pbit -l`” options to display the PBIT results.

➢ Example:

```
# sysvartool -A pbit -l
VX3035  detected
area = 2, arch = 2
POSTs configured to run from command line:
mem_data: PASSED
mem_addr: PASSED
mem_pattern1: PASSED
mem_pattern2: PASSED
mem_pattern3: PASSED
mem_pattern4: PASSED
ether_loop0: PASSED
ether_loop1: PASSED
ether_loop2: PASSED
system: PASSED (FAILED ONCE)
PASSED : 10
FAILED : 0
NOT RUN : 0
TOTAL : 10
```

3.2 VxWorks

3.2.1 VxWorks Synthetic PBIT Result

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

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

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

3.2.2 VxWorks Detailed PBIT Result

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

```
-> pbitdisplay
POSTs configured to run from command line:
mem_data: PASSED
mem_addr: PASSED
mem_pattern1: PASSED
mem_pattern2: PASSED
mem_pattern3: PASSED
mem_pattern4: PASSED
pcie_vpx_sw: PASSED
serial: PASSED
rtc: PASSED
sysflash: PASSED
cpld: PASSED
temp_sensors: PASSED
temperature: PASSED
fnvram: PASSED
ether_link0: PASSED
ether_loop0: PASSED
ether_loop1: PASSED
ether_loop2: PASSED
ether_loop3: PASSED
ether_loop4: PASSED
voltage: PASSED
sata4_dev_see: PASSED
sata0_controler: PASSED
sata1_controler: PASSED
vpd: PASSED
eeprom: PASSED
usb1_controller: PASSED
usb2_controller: PASSED
system: PASSED

PASSED  : 29
FAILED  : 0
NOT RUN : 0
TOTAL   : 29

value = 0 = 0x0
->
```

3.3 Windows XP, XPe and Seven

The Windows XPe BSP for VX3035 provides a Kontron embedded API to access the devices.

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

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) which becomes the validated final system configuration for the deployment.

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

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

4.1.3 DEPLOY

PBIT software implements unrivalled features which make Kontron product deployment easier and cheaper. The PBIT code is located in the same device as the BIOS code. It is deployed along with the BIOS (see Appendix A of the BIOS User’s Guide). The PBIT settings (and results) are located in their dedicated non volatile EEPROM device. To copy these settings, boot the Operating System and record the VPD EEPROM content from the offset address 0x1000 up to 0x3FFF and store it in a file. Then use this file to copy back data to the new board VPD EEPROM to configure.

One of the most useful features of PBIT for deployments is implemented in the “system” test (see section 2.10 page 25). As this test scans all the possible I/O ports and devices to compare with a pre-recorded system reference, it can signal any change in the expected I/O devices availability. This can cover I/O device failures, or more subtle system alterations (for example a storage device left unconnected). The “system” test results are organized to pinpoint the faulty I/O port easily, before attempting a complete system boot and having to manage the corresponding cascading failures due to this incorrect peripheral configuration.

4.1.4 MAINTAIN

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

The PBIT results record the behaviour of the last PBIT run (see section 2.5 page 20). In addition to these results, each test implements a dedicated “FAILED ONCE” flag which has to be reset separately (see section 2.6 page 21). 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 support-kom-sa@kontron.com to open a feature request discussion with Kontron.

4.1.5 MANUFACTURING

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

Specific features have been implemented for that purpose.

Appendix A - List of Abbreviations

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

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

TELEPHONE AND E-MAIL
+33 (0) 4 98 16 34 00
sales@kontron.com
support-kom-sa@kontron.com

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