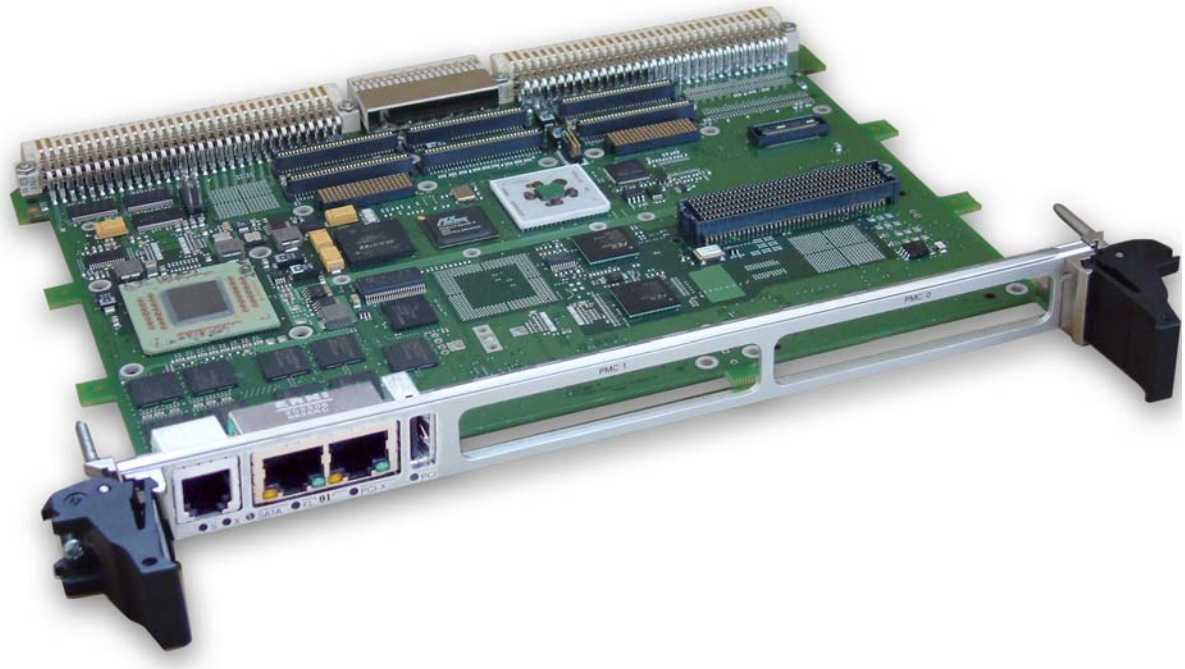
» VX6060 «



PBIT User's Guide

SD.DT.F78-1e - December 2011

Revision History

Publication Title:	VX6060 PBIT User's Guide
Doc. ID:	SD.DT.F78-1e

Rev.	Brief Description of Changes	Date of Issue
1e	PBIT version 1.7 Add new tests for SATA, USB, Ethernet devices detection Update PBIT access information under Operating System	12-2011
0e	Initial Version	06-2011

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 firmware (PBIT) for Kontron VX6060 boards. It is an optional product available under the VX6060 EFI BIOS shell environment.

The PBIT firmware is implemented as a binary executable located in the system Flash and included in the BIOS shell application. Its configuration such as tests list and tests result is stored in VX6060 system EEPROM.

The PBIT firmware includes among other features 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 EEPROM I2C and can also be accessed and reconfigured under Operating System such as Linux or VxWorks. See Chapter 3 page 22.
- Simplified test result is also available in a 8-bit PLD register (register 0x2).

VX6060 board is implemented as 2 similar CPU subsystems so the PBIT is available on each CPU subsystem and the user must configure or run it on both CPU A and CPU B independently.

1.1 Related Documents

Hardware

> VX6060 User's Guide CA.DT.A76

> VX6060 Hardware Release Notes CA.DT.A77

Firmware

> VX6060 AMI BIOS User Reference Manual SD.DT.F69

1.2 PBIT Installation and Activation

The PBIT comes pre−installed in the system Flash, along with the EFI BIOS firmware on the VX6060 boards. It is an optional product that can be activated on any VX6060. Please contact Kontron support for more information.

To install a new BIOS version including a new PBIT version, please see the VX6060 BIOS User Reference Manual - SD.DT.F69.

1.3 PBIT Configuration

The PBIT must be configured first by EFI shell command line. It is presented as a list of tests to be executed. Each test is focused on a specific device of the VX6060. The list of tests to be executed can be displayed and modified by using the EFI BIOS Shell command `kdiag` (see section 1.5 page 7).

1.3.1 PBIT Configuration by Command Line

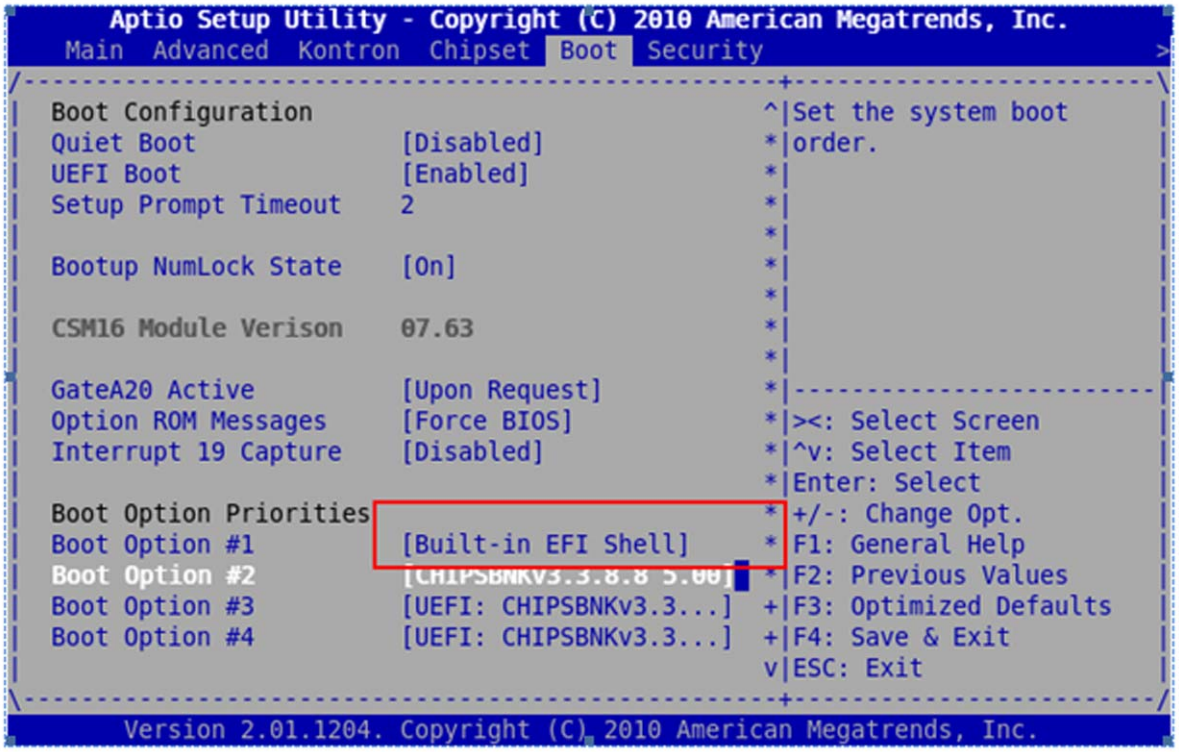
The following explains how to configure and execute the PBIT by command line first. This operation must be repeated on each CPU subsystem for a VX6060 board (CPU-A and CPU-B).

➢ Select EFI Shell as first boot device.

Enter BIOS SETUP with <F2> keyboard key and select boot menu.

Move EFI Shell like first boot device (use key <+> or <->).

Then Save & Exit



The EFI shell prompt is displayed, allowing to enter the PBIT commands. For example:

➢ Verify the PBIT version

```
VX6060-A> kdiag version

PBIT VERSION 1.4

VX6060-B> kdiag version

PBIT VERSION 1.4
```

➢ Launch the PBIT manually for verification:

```
VX6060-A> 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_6u_sw" (fast,simple)  PASSED
PBIT "pcie_vpx_sw" (fast,simple)  PASSED
PBIT "pcie_pcix_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 "eth_switch" (fast,simple)  PASSED
PBIT "voltage" (fast,simple)  PASSED
PBIT "sata0_test" (fast,simple)  PASSED
PBIT "sata1_test" (fast,simple)  PASSED
PBIT "vpd" (fast,simple)  PASSED
PBIT "eeprom" (fast,simple)  PASSED
PBIT "pmc_xmc_check" (fast,simple)  PASSED
PBIT "usb1_controller" (fast,simple)  PASSED
PBIT "usb2_controller" (fast,simple)  PASSED
PBIT "system" (fast,simple)  NOT SAVED PASSED
```

➢ Configure test “system“

By default the test named “system“ is not ready. The user should record the system configuration when the system is ready for this (all the equipments on PCIe such as PMC/XMC,VPX boards are present, see section 2.10 page 21 - PBIT System Learn Test, for more detail).

To record and then activate this test, run “kdiag system_learn”

```
VX6060-A> kdiag system_learn
   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
    . . .
Storing system configuration ...
Done
```

➢ Check the PBIT results:

To check the PBIT results under EFI shell, use “`kdiag stat`”:

```
VX6060-A> 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_6u_sw (fast,simple)
 PASSED : pcie_vpx_sw (fast,simple)
 PASSED : pcie_pcix_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 : eth_switch (fast,simple)
 PASSED : voltage (fast,simple)
 PASSED : sata0_test (fast,simple)
 PASSED : sata1_test (fast,simple)
 PASSED : vpd (fast,simple)
 PASSED : eeprom (fast,simple)
 PASSED : pmc_xmc_check (fast,simple)
 PASSED : usb1_controller (fast,simple)
 PASSED : usb2_controller (fast,simple)
 PASSED : system (fast,simple)

RUN    : 29
PASSED : 29
FAILED : 0
NOT_RUN: 0
```

1.3.2 Set PBIT to Run Automatically

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

The following explains how to have the PBIT launched at any VX6060 boot and before OS boot. This operation must be repeated on each CPU subsystem for a VX6060 board (See previous section to access and configure the PBIT first).

> Configure the PBIT to be launched at boot time

The automatic start is activated using the BIOS shell environment variable “bootcmd”

```
VX6060-A> 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.

```
VX6060-A> set bootdelay 1
```

Verify

```
VX6060-A> set
  bootcmd   : kdiag run
  bootdelay : 1
```

Then reset the system:

```
VX6060-A> reset
```

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

To stop under the EFI shell after the PBIT execution, define variable named “bootdontexit”

```
VX6060-A> set bootdontexit  1
```

To execute a shell script with a list of commands, type:

```
set startupAuto      1
set startupdelay     1
```

Then plug a USB device with file named `startup.nsh` including shell commands.

1.4 Synthetic PBIT Result

A 8-bit synthetic PBIT result can be read in the CPLD register 2.

This register is easily accessible under Operating System using CPLD OS facilities or direct memory I/O access at address 0x802. Under the BIOS EFI shell use the following command: "`kpld -r 2`":

```
VX6060-A> kpld -r 2
READ : @0x2 = 0x1
```

The 8-bit register 2 content meaning 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 Test run and PASSED

To identify a PBIT from its number then use the command `kdiag [PBITnumber]`

Example:

```
VX6060-A> kdiag 16
serial (16) - Checks the serial line 1
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. The list can be modified by the user to fulfill his specific coverage and execution time needs. The “`kdiag`” command displays both parts of the lists.

The initial default tests list can be restored with “`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 VX6060. No signal on any connector will be modified during the default test execution.

Command `kdiag` displays the tests list to run:

```
VX6060-A> kdiag
```

The default tests list to run is displayed below:

```
List of test configured to run by default:

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
pcie_6u_sw (12) - Checks central pciExpress switch
pcie_vpx_sw (13) - Checks the pciExpress backplane switch
pcie_pcix_sw (14) - Checks  PCIe/PCIX switchs (PMC/XMC)
serial (16) - Checks the serial line 1
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 82577LM front Gigabit in Loopback mode
ether_loop1 (56) - Checks 82580 Gigabit Backplane 1 in Loopback mode
ether_loop2 (57) - Checks 82580 Gigabit Backplane 2 in Loopback mode
eth_switch (58) - Checks Central Marvell Ethernet switch
voltage (60) - Checks the voltage sensors & value
sata0_controler (68) - Checks sata0 controller
sata1_controler (69) - Check sata1 controller
vpd (70) - Checks VPD data required for board operation.
eeprom (71) - Checks User EEPROM (0xA2)
pmc_xmc_check (76) - Checks PMCA 64bits/XMC path & slot
usb1_controller (86) - Checks first Ibexpeak usb controller
usb2_controller (87) - Checks second Ibexpeak usb controller
system (88) - Checks system configuration PCIe stability
```

To run the default PBIT tests list, enter on either subsystem side A or B prompt:

```
VX6060-A> kdiag run
```

1.5.2 Optional 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))
 mem_addrpat (4) - Checks Memory/ECC using address pattern (offset)
 mem_addrpat2 (5) - Checks Memory/ECC using address pattern (~offset)
 smbus0 (26) - Check SMBUS0 between PLD and backplane
 smbus1 (27) - Check SMBUS1 between PLD and backplane
 ether_link0 (50) - Checks the link status is "UP" on 82577LM Gigabit Network.
 eth-SW (51) - Checks the link status is "UP" on front or rear ETH-SW connector
 ether_lnk_sw7 (52) - Checks the link status is "UP" on P1 ETH-SW7
 ether_lnk_sw8 (53) - Checks the link status is "UP" on P1 ETH-SW8
 ether_lnk_sw9 (54) - Checks the link status is "UP" on on P1 ETH-SW9
 sata0_dev_see (63) - Checks if a SATA0 disk on rear P1 is present
 sata1_dev_see (64) - Checks if a SATA1 disk on rear P2 is present
 sata2_dev_see (65) - Checks if a SATA2 disk on rear P2 is present
 sata3_dev_see (66) - Checks if a SATA3 HDD disk on board is present
 sata4_dev_see (67) - Checks if a SATA mezzanine device is present
 pmc_xmc_present (77) - Checks if device is present on PMCA 64bits/XMC slot
 usb_A_dev_see (80) - Checks if a USB-A (Front A) device is present
 usb_AB_dev_see (81) - Checks if a USB-AB (Front A/B) device is present
 usb1_flash_see (82) - Checks if a USB1 (flash) is present
 usb2_dev_see (83) - Checks if a USB2 (Rear P1) device is present
 usb3_dev_see (84) - Checks if a USB3 (Rear P2) device is present
 faultytest (98) - A dummy test that returns FAIL
 hangtest (99) - A dummy test that will hang
```

Initially the optional tests list contains:

- Memory tests to complete the default memory tests. Long execution time.
- External equipment dependent tests (ETH, SATA, PMC, XMC, USB). Only use them when it is necessary to verify a device presence.
- `Smbus` test: to check the backplane smbus. Only significant in complex mode if specific equipment is connected. Do not use.
- Utility tests: `faultytest` helps the user 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 run list lasts 16 seconds for a 2 GB of RAM on each CPU and about 21 seconds if all existing tests are run.

Below is the execution time for each test and for a board with 2 GB/CPU. The tests that have a significant duration compared to other tests appear in red.

Test name	Test ID number	Executive time
mem_data (fast,simple)	1	680 ms
mem_addr (fast,simple)	2	3.6s
mem_bitflip (fast,simple)	3	1.4s
mem_addrpat (fast,simple)	4	1.4s
mem_addrpat2 (fast,simple)	5	1.4s
mem_pattern1 (slow,simple	6	1.4s
mem_pattern2 (slow,simple)	7	1.4s
mem_pattern3 (slow,simple)	8	1.4s
mem_pattern4 (slow,simple)	9	1.4s
pcie_6u_sw (fast,simple)	12	< 300ms
pcie_vpx_sw (fast,simple)	13	< 300 ms
pcie_pcix_sw (fast,simple)	14	< 300 ms
serial (fast,simple)	16	400 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	600 ms
ether_link0 (fast,simple)	50	< 300 ms
eth-SW (fast,simple)	51	< 300 ms
ether_lnk_sw7 (fast,simple)	52	< 300 ms
ether_lnk_sw8 (fast,simple)	53	< 300 ms
ether_lnk_sw9 (fast,simple)	54	< 300 ms
ether_loop0 (fast,simple)	55	590 ms
ether_loop1 (fast,simple)	56	1.1 s
ether_loop2 (fast,simple)	57	1.1 s
eth_switch (fast,simple)	58	3.5 s
voltage (fast,simple)	60	< 300 ms
sata0_dev_see (fast,simple)	63	< 300 ms
sata1_dev_see (fast,simple)	64	< 300 ms
sata2_dev_see (fast,simple)	65	< 300 ms
sata3_dev_see (fast,simple)	66	< 300 ms

Test name	Test ID number	Executive time
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
pmc_xmc_check (fast,simple)	76	< 300 ms
pmc_xmc_present (fast,simple)	77	< 300 ms
usb_A_dev_see (fast,simple)	80	< 300 ms
usb_AB_dev_see (fast,simple)	81	< 300 ms
usb1_flash_see (fast,simple)	82	< 300 ms
usb2_dev_see (fast,simple)	83	< 300 ms
usb3_dev_see (fast,simple)	84	< 300 ms
usb1_controller (fast,simple)	86	< 300 ms
usb2_controller (fast,simple)	87	< 300 ms
system (fast,simple)	88	< 300 ms

For a board with 4 GB/CPU, memory test 2 lasts 3.7 s and memory tests 3...9 last 2.6 s.

For a board with 8 GB/CPU, memory test 2 lasts 4.3 s and memory tests 3...9 last 5.5 s.

Chapter 2 - PBIT Command Line Reference Guide

The PBIT firmware is configured and executed using the BIOS EFI Shell command «`kdiag`». The following section describes the various `kdiag` command parameters.

2.1 On-line Help

On BIOS EFI Shell prompt enter the command `help kdiag` to display the full help.

In command format.

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

```
VX6060-A> 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

Configure PBIT(s) :

  kdiag  [oside] cfg <cfgarg> ... [<PBITname>|<PBITnum>] ...

    cfg <cfgarg> : Configure one or several PBIT(s).

                  <cfgarg> is either :
                - "delete" to delete PBIT(s) from the list of configured PBITs
                - "default" to configure PBIT(s) with a default run mode
                - a comma separated list of runflags defining a PBIT
                   run mode; for example : fast,complex.

                 valid runflags are :
                 - "SPEED" flags (can NOT be mixed)
                   slow         : run in slow mode (full testing)
                   fast         : run in fast mode (fast testing)
                 - "CONFIG" flags (can NOT be mixed)
                   simple      : run in simple mode (no external hardware)
                 complex       : run in complex mode (needs external hardware)
                 - "HALT" flags (can NOT be mixed)
                   haltonfail    : halt immediately (hang) if test fails
                   promptonfail  : halt at U-Boot prompt (no OS) if test fails

    [<PBITname>|<PBITnum>] ...
                 list of PBIT(s) to configure.
                 PBIT(s) are referenced using their name or their number.
                 All missing tests if the list is empty

    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

Display PBIT version :

  kdiag version
```

2.2 Display the List of Selected Tests

To display both the existing tests list and their configuration use the following command:

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

> `kdiag` with no argument prints list of tests that are selected to run by command "`kdiag run`"

Example:

```
VX6060-A> help kdiag

VX6060-A> 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_6u_sw (12) - Checks central pciExpress switch
   capabilities : fast,simple
   run mode 1   : fast,simple
 ...to be continued

Other PBITs available but not yet configured :

 mem_bitflip (3) - Checks Memory/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 82577LM Gigabit Network.
   capabilities : fast,simple
 eth-SW (51) - Checks the link status is "UP" on front or rear ETH-SW connector
   capabilities : fast,simple
 ether_lnk_sw7 (52) - Checks the link status is "UP" on P1 ETH-SW7
   capabilities : fast,simple
 . . .
Use ´help kdiag´ to get more info.

VX6060-A>
```

2.3 PBIT Executed from the Command Line

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

```
VX6060-A> 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_3u_sw" (fast,simple)  PASSED
PBIT "pcie_vpx_sw" (fast,simple)  PASSED
PBIT "pcie_pcix_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 "eth_switch" (fast,simple)  PASSED
PBIT "voltage" (fast,simple)  PASSED
PBIT "sata0_test" (fast,simple)  PASSED
PBIT "sata1_test" (fast,simple)  PASSED
PBIT "vpd" (fast,simple)  PASSED
PBIT "eeprom" (fast,simple)  PASSED
PBIT "pmc_xmc_check" (fast,simple)  PASSED
PBIT "usb1_controller" (fast,simple)  PASSED
PBIT "usb2_controller" (fast,simple)  PASSED
PBIT "system" (fast,simple)  PASSED
```

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

```
VX6060-A> kdiag run <PBIT number | PBIT name ...>
```

For example:

```
VX6060-A> > kdiag run usb1_controller sysflash
PBIT "usb1_controller" (fast,simple)  PASSED
PBIT "sysflash" (fast,simple)  PASSED
```

is equivalent to:

```
VX6060-A> kdiag run 84 22
PBIT "usb1_controller" (fast,simple)  PASSED
PBIT "sysflash" (fast,simple)  PASSED
```

The PBIT number ID is displayed with the `kdiag` command (with no parameter) or with the `kdiag <PBIT name>` command.

2.4 Execute PBIT in Loop Mode

To run in loop mode, enter:

```
VX6060-A > diag run loop <count>
```

where <count> is the number of loop to do.

To run a single test in loop mode, enter:

```
VX6060-A=> diag run loop <count> <PBIT number | PBIT name ...>
```

where count is the number of loop to do.

Example with 10 runs of test 55 ether_loop0:

```
VX6060-A => diag run loop 10 55
```

is equivalent to:

```
VX6060-A => diag run loop 10 ether_loop0
```

2.5 Get the PBIT Results

Use the `kdiag stat` command to get the PBIT results. Use the option “`oside`” to get the PBIT results from the other CPU of VX6060:

```
VX6060-A> 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_6u_sw (fast,simple)
 PASSED : pcie_vpx_sw (fast,simple)
 PASSED : pcie_pcix_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 : eth_switch (fast,simple)
 PASSED : voltage (fast,simple)
 PASSED : sata0_test (fast,simple)
 PASSED : sata1_test (fast,simple)
 PASSED : vpd (fast,simple)
 PASSED : eeprom (fast,simple)
 PASSED : pmc_xmc_check (fast,simple)
 PASSED : usb1_controller (fast,simple)
 PASSED : usb2_controller (fast,simple)
 PASSED : system (fast,simple)

RUN    : 29
PASSED : 29
FAILED : 0
NOT_RUN: 0

VX6060- A>
```

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:

```
VX6060-A => kdiag clrstat [oside]
```

To clear the PBIT history including "FAILED ONCE" flag, enter:

```
VX6060-A => kdiag clrallstat
```

2.7 Configure PBIT Tests List to Execute

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

Each test can be added, removed, and programmed 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 mode are defined with the following test flags:

> `HALT flags` (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 tests execution when an error is detected.

> `SPEED flags` (can NOT be mixed)

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

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

No test implements a difference between fast and slow mode in this PBIT version.

> `CONFIG flags` (can NOT be mixed)

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

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

Complex mode requires a device external to VX6060 and is reserved for factory test.

2.7.2 Adding a Test to the Current Run List

```
VX6060-A=> kdiag cfg <cfgarg> ... <PBITname>|<PBITnum> ...
```

`cfgarg` allows to choose the test run mode. Use the keyword `default` to set a default mode (typically FAST, SIMPLE)

> Examples:

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

```
VX6060-A> kdiag cfg default 16
```

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

```
VX6060-A => kdiag cfg promptonfail 16
```

To add this test with flags `complex + promptonfail` enter:

```
VX6060-A => kdiag cfg complex,promptonfail 16
```

Execute `kdiag` or `kdiag 16` to check the configuration:

```
VX6060-A> kdiag 16
serial (16) - Checks the serial line 1
    capabilities : fast,simple/complex
     run mode 1   : fast,complex,promptonfail
```

`oside` parameter can also be used to configure the other CPU side.

2.7.3 Removing a Test from the Current Run List

Example for removing the test number 16

```
VX6060-A> kdiag cfg delete 16
```

Verify

```
VX6060-A> kdiag 16
PBIT "16" is not configured to run from command line
```

`oside` parameter can also be used to configure the peer subsystem CPU side

2.7.4 Add all the Optional Tests to Current Run List

It is possible to add all the optional tests in the current run list with a single command.

To add all the optional tests with a default run mode, enter :

```
VX6060-A> kdiag cfg default
Configuration of all PBITs in default mode only
List of test to configure
--> mem_bitflip (3) - Checks Mem/ECC using bit-flip pattern ((1 << (offset % 32))
--> mem_addrpat (4) - Checks Memory/ECC using address pattern (offset)
--> mem_addrpat2 (5) - Checks Memory/ECC using address pattern (~offset)
--> smbus0 (26) - Check SMBUS0 between PLD and backplane
--> smbus1 (27) - Check SMBUS1 between PLD and backplane
--> ether_link0 (50) - Checks the link status is "UP" on 82577LM Gigabit Network.
--> eth-SW (51) - Checks the link status is "UP" on front or rear ETH-SW connector
--> ether_lnk_sw7 (52) - Checks the link status is "UP" on P1 ETH-SW7
--> ether_lnk_sw8 (53) - Checks the link status is "UP" on P1 ETH-SW8
--> ether_lnk_sw9 (54) - Checks the link status is "UP" on on P1 ETH-SW9
--> sata0_dev_see (63) - Checks if a SATA0 disk on rear P1 is present
--> sata1_dev_see (64) - Checks if a SATA1 disk on rear P2 is present
--> sata2_dev_see (65) - Checks if a SATA2 disk on rear P2 is present
--> sata3_dev_see (66) - Checks if a SATA3 HDD disk on board is present
--> sata4_dev_see (67) - Checks if a SATA mezzanine device is present
--> pmc_xmc_present (77) - Checks if device is present on PMCA 64bits/XMC slot
--> usb_A_dev_see (80) - Checks if a USB-A (Front A) device is present
--> usb_AB_dev_see (81) - Checks if a USB-AB (Front A/B) device is present
--> usb1_flash_see (82) - Checks if a USB1 (flash) is present
--> usb2_dev_see (83) - Checks if a USB2 (Rear P1) device is present
--> usb3_dev_see (84) - Checks if a USB3 (Rear P2) device is present

VX6060-A>
```

To add all optional tests in one command and with `promptonfail flag`, enter:

```
VX6060-A> kdiag cfg promptonfail
Configuration of all PBITs in default mode only
List of test to configure
--> mem_bitflip (3) - Checks Mem/ECC using bit-flip pattern ((1 << (offset % 32))
--> mem_addrpat (4) - Checks Memory/ECC using address pattern (offset)
--> mem_addrpat2 (5) - Checks Memory/ECC using address pattern (~offset)
--> smbus0 (26) - Check SMBUS0 between PLD and backplane
--> smbus1 (27) - Check SMBUS1 between PLD and backplane
--> ether_link0 (50) - Checks the link status is "UP" on 82577LM Gigabit Network.
--> eth-SW (51) - Checks the link status is "UP" on front or rear ETH-SW connector
--> ether_lnk_sw7 (52) - Checks the link status is "UP" on P1 ETH-SW7
--> ether_lnk_sw8 (53) - Checks the link status is "UP" on P1 ETH-SW8
--> ether_lnk_sw9 (54) - Checks the link status is "UP" on on P1 ETH-SW9
--> sata0_dev_see (63) - Checks if a SATA0 disk on rear P1 is present
--> sata1_dev_see (64) - Checks if a SATA1 disk on rear P2 is present
--> sata2_dev_see (65) - Checks if a SATA2 disk on rear P2 is present
--> sata3_dev_see (66) - Checks if a SATA3 HDD disk on board is present
--> sata4_dev_see (67) - Checks if a SATA mezzanine device is present
```

```
--> pmc_xmc_present (77) - Checks if device is present on PMCA 64bits/XMC slot
--> usb_A_dev_see (80) - Checks if a USB-A (Front A) device is present
--> usb_AB_dev_see (81) - Checks if a USB-AB (Front A/B) device is present
--> usb1_flash_see (82) - Checks if a USB1 (flash) is present
--> usb2_dev_see (83) - Checks if a USB2 (Rear P1) device is present
--> usb3_dev_see (84) - Checks if a USB3 (Rear P2) device is present

VX6060-A>
```

Verify for test 3

```
VX6060-A> kdiag 3
 mem_bitflip (3) - Checks Memory/ECC using bit-flip pattern((1<<(offset % 32))
   capabilities : slow/fast,simple
   run mode 1   : fast,simple,promptonfail
```

- The command will not include test number greater than 95 because these tests are considered as tooling tests for the PBIT validation.
- **oside** parameter can also be used to configure the peer CPU side

2.7.5 Restore Default Run List

To restore the default tests list, enter:

```
VX6060-A> kdiag default
```

To restore the default tests list on peer subsystem CPU side (here side B) , enter:

```
VX6060-A> kdiag oside default
```

2.8 Run the PBIT in Silent Mode

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

```
VX6060-A> kdiag silent
PBIT set in silent mode
```

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

```
VX6060-A> kdiag silent
PBIT silent mode removed
```

- In this mode error messages are displayed anyway. When set the silent mode is valid on both subsystem CPU side A and B.
- To prevent any output to the serial line use the BIOS SETUP configuration (Serial Line Console Redirection).

2.9 Display the PBIT Version

The following PBIT command displays the PBIT version:

```
VX6060-A> kdiag version
PBIT VERSION 1.7 ID 11328
```

2.10 PBIT System Learn Test

A test named "system" can be used to record the system configuration and test it later. "kdiag system_learn" is the command to record the system configuration. It must be run when the system configuration is the one to be recorded. All detected PCI device are recorded (the list is visible with BIOS shell command "pci").

Examples:

```
VX6060-VX6060-A> kdiag run system
PBIT "system" (fast,simple)  NOT SAVED PASSED

VX6060-A> kdiag system_learn
   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
    . . .

Storing system configuration ...
Done

VX6060-A> kdiag run system
PBIT "system" (fast,simple)   PASSED
```

- This command must be done on each CPU subsystem (side A and side B).
- Don't forget that moving one board on VPX backplane can result in a system test FAILED during the next boot
- There is no command to clear the system learn configuration. It must be recorded again if the configuration has changed or it must be removed from the tests list to execute if the test is not desired.

Chapter 3 - PBIT and OS Interface

The PBIT synthetic results and the PBIT detailed results are accessible under Operating System delivered with VX6060 board. The following gives you an overview of the facilities available under BSP Linux, VxWorks and Windows OS. Please refer to the appropriate OS Release Notes for more details.

3.1 Linux

3.1.1 Linux Synthetic PBIT Results

The synthetic PBIT results in CPLD register 2 are accessible with "`cpldtool`" facility. Use `cpldtool -a` option and check register 2 value.

For example:

```
# cpltool -a
Reg 0x0 - CPLD ID = 0x38
        CPLD ID=0x0
        CPLD Debug=0x0
        CPLD Version=0x38
Reg 0x1 - PCB ID = 0x30
Reg 0x2 - FIRM POST RESULT = 0x01
Reg 0x3 - FIRM POST DEBUG RESULT = 0x00
Reg 0x4 - POWER RST CONFIG = 0x03
        Power Mode State DIP SWITCH=0x0
        ALARM INHIB=0x0
        PLTRST to PERST6U_INHIB=0x0
        Reset Mode=0x0
        PERST_3U# and PERST_6U# Control=0x1


 ...
 To be continued
```

3.1.2 Linux Detailed PBIT Results

The detailed PBIT results under Linux are accessible with "`sysvartool`" facility. Use sysvartool –A pbit -l to display the PBIT results.

For example:

```
# sysvartool -A pbit -l
VX6060 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) are 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 VX6060 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 the PBIT firmware:

- EVAL (platform EVALuation)
- DEVEL (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 present the PBIT features and benefits according to targeted Use Cases.

4.1.1 EVAL

The PBIT firmware 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 serial line (COM0) 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, so as to try various coverage/execution time trade-offs.

The system test can be used at the end of the development, to capture a complex peripheral configuration (PCIe, USB, SATA) which becomes the validated final system configuration for 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 FLASH results section gives most of the information needed by a control and monitoring application to take a decision on the system following 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 controlling unit can alter Control registers to modify the behaviour of the board under control.

4.1.3 DEPLOY

The PBIT firmware 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 FLASH device. To copy these settings, boot Operating System and record the VPD EEPROM Content from Offset 0x1000 up to offset 0x3FFFF 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 the PBIT for deployments is implemented in the SYSTEM TESTS (see section 2.10 page 21 and 1.5.2 page 8). As this test scans all possible I/O ports and devices to compare with a pre-recorded system status 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 organised so as to pinpoint the faulting 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. 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.

The PBIT extension: Thanks to its modular approach, and to the EFI execution environment. the 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, 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

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.