» Whitepaper «

Power-On Built-In Test Solution for Kontron VPX and VME Single Board Computers

For Power Up test coverage and Problem Solving

François Boulagnon, Vincent Chuffart

Power-On Built-In Test on VPX and VME Single Board Computers

PBIT is an acronym for Power on Built In Test. Its main purpose is to test a computer platform prior to launching the main software to assess its proper behaviour and health status. According to the results, a system can decide to cancel the normal operation and enter a planned alternate mode, or stop all activity and report a fault.

Kontron PBIT is featured on CPU boards such as **VX3230**, **VM6250**, **VX6060**, **VX3030**, **VM6050**, **VX3035**, (and soon **VX3040**) operating in VPX or VME enclosures deploying embedded and ruggedized Military, Aerospace or Transportation system market applications.

CONTENTS

1. Overview

Kontron PBIT software is an extensible framework for Power On Built In Test of SBCs. Its execution occurs early in the boot process. PBIT is a powerful tool to help fast and accurate diagnostic, avoiding to waste precious time in deep investigations of the application code when a failure is showing up in a system.
PBIT has been designed to be executed **as fast as possible** and be an integral part of the operational system start sequence (automatic mode).
PBIT also has a user interface that can be called up to access an interactive test mode. A configurable tests list allows adding and removing specific feature test. This list is tuned by the user to reduce the test time; only the features used for a given application need to be tested at power up.
PBIT also include an innovative complex test approach allowing maximum coverage with minimal learning curve for Kontron customers deploying SBC based systems: PBIT has a **learning mode** to record the status of all ports and buses connected to the processor, allowing a quick and extensive **system configuration diagnostic**.

PBIT includes, among others, the following services:

- PBIT offers multiple tests that can be **added or removed** from a run list by according to the desired compromise between time to boot, test coverage and system configuration.
- **Cold Start/Hot-reboot tests**: thanks to BIT modularity and scalability, the cold start BIT session is usually set to focus on the maximum test coverage, whereas the ultra-fast hot reboot mission-critical BIT session will target to check the availability of the computer system essentials in the shortest time, meaning a few seconds, not minutes.
- PBIT also offer a unit **SYSTEM TEST** that **can quickly spot** any unwanted **configuration change**.
- PBIT can **run automatically** (when booting firmware) or in an **interactive mode** (at firmware or EFI BIOS Shell firmware prompt)
- PBIT configuration and detailed results are stored in a dedicated part of the board non volatile storage. This content can be accessed and reconfigured under Operating System such as Linux, VxWorks or Windows.
- A Synthetic test result value is also stored into a board register, accessible by a simple mapped I/O. This register is also available to the chassis management board (such as Kontron CMB) via the VME or VPX backplane SMB feature.

2. Easy evaluation of Kontron PBIT

PBIT is an optional product to be ordered with each SBC. To allow quick evaluation, PBIT is always part of all Kontron Easy Evaluation Systems.
PBIT can be launched from the BIOS (Intel-based SBC) or the Uboot Firmware (PowerPC-based SBC), and quickly evaluate using its interactive mode.
The main PBIT operator interface for command and results is the one provided by the firmware. According to the firmware setting the serial line (COM0) or VGA (or Display Port) Screen plus keyboard are supported.

Extensive and synthetic BITS results are made available in multiple forms to implement any system boot policiy.

PBIT incorporates many firmware sub-commands for on-line help, execute run mode, dump/add/remove unitary test, configure test run mode and so on.

```
VX3030> 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_link0" (fast,simple)  WARNING: Link speed 100Mb/s 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 "sata4_dev_see" (fast,simple)  PASSED
PBIT "sata0_controler" (fast,simple)  PASSED
PBIT "sata1_controler" (fast,simple)  PASSED
PBIT "vpd" (fast,simple)  PASSED
PBIT "eeprom" (fast,simple)  PASSED
PBIT "usb0_dev_see" (fast,simple)  FAILED
PBIT "usb1_dev_see" (fast,simple)  PASSED
PBIT "usb1_controller" (fast,simple)  PASSED
PBIT "usb2_controller" (fast,simple)  PASSED
PBIT "system" (fast,simple)   NOT SAVED PASSED


VX3030>
```

Figure 1: Example of PBIT execution under BIOS EFI Shell of a VX3030 with one "failed" test.

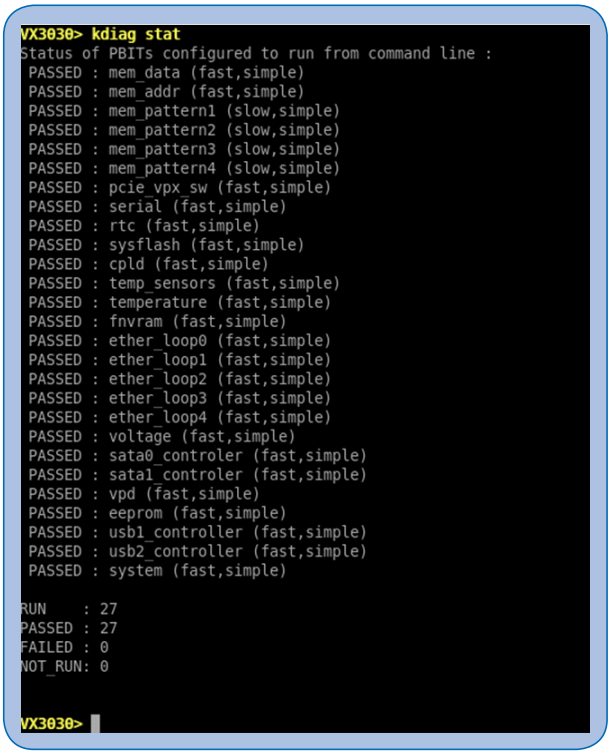

Figure 2: Reading results from a previous PBIT session from VX3030 BIOS Shell. In this case every test PASSED.

3. Kontron PBIT Features & Benefits

- PBIT is **modular** and comes with a set of **unitary tests** that can be launched separately to cover each board feature one by one.

This is very useful when troubleshooting transient errors such as faulty cables or connectors.

- PBIT is **configurable and customizable**, thus any unitary test can be added or removed to a given run.

Multiple versions of this list are supported, allowing various cold and warm start test policies.

- PBIT must be **extensible**. As such PBIT is able to **detect and process any peripheral device** such as SATA, USB, network link, PMC/XMC, other VPX boards...

Implementing extensive coverage of anything attached to the SBC is the most innovative feature of Kontron PBIT.

In its operational form, PBIT must not modify any external device attached to the SBC. It must not interfere with the system around the board.

We make sure no unit test can create unsafe situations in the system under test.

- PBIT **Execution time** is configurable and documented for all unitary tests. This puts the Coverage/Execution time compromise in the hands of the SBC user.

Putting the Boot Time/Test Coverage compromise in the hands of the end user is PBIT main advantage.

- PBIT provides **visual progress indicator** when needed.
- PBIT session **always recovers** from hardware defects. If a test would hang the processor, a **watchdog** protection ensures the system recovery.
- PBIT can detect a **reference configuration change** (link status, disk presence, other board change in the VPX backplane ...). See System Configuration Test

PBIT offers unmatched system coverage. Imagine how nice it will be to detect a USB stick left by accident in a random port of a complex multi board, multi rack system, or track a loose network cable even before the OS is started.

- PBIT results are **accessible** from the **Operating System**, thanks to a dedicated PBIT API.

PBIT synthetic results are also available from System Management Bus (dedicated I²C connection on each slot in VME or VPX backplanes). SMB management is best achieved with Kontron CMB (Chassis Management Board) to synthesize the system health and make it available (LEDs, Operator on the serial line, or Network: Http or SNMP).

4. System Test and Real Life Deployment Issues

One of the most useful features of PBIT for deployments is the **SYSTEM TEST**.
For a long time, creating an accurate PBIT configuration which would cover not only the SBC but also anything attached to it (storage, network) was somewhat impossible or required a tedious development phase specific to each program.
On the other hand, the SBC seemed to be ideally located in a system to be able to become THE system configuration guardian. This is why we have developed our SYSTEM TEST in a radically innovative way.

The **SYSTEM TEST** can be used **to capture a complex peripheral configuration** (PCIe agents across a VPX backplane, PCIe mezzanines on PMC/XMC slot, but also USB and SATA peripherals, along with network interface status).
How to use it? Once the final system integration phase is over, a simple PBIT command: "**kdiag system learn**" is run once to discover and record the current system configuration.
After the initial capture, PBIT considers this record as the only valid system configuration.
This recorded information is used to detect any change in the system being deployed.

With this approach PBIT SYSTEM TEST is able to catch real life issues such as:

- Bad or faulty contact (SATA, USB, Network, undetected mezzanine PMC, XMC)
- Wrong configuration (missing board in VPX, extraneous USB Stick left after system maintenance, board in wrong slot)

All this will result in a system test FAILED status.

5. PBIT result access under Operating System

PBIT results are accessible from the Operating System running on the SBC. A PBIT API is proposed for use under Linux, VxWorks or Microsoft Windows BSP.
An alternative for **RAPID PROTOTYPING** is to intially use the synthetic result register, directly accessible at a given PCI I/O address. This will work on any OS, or in bare metal (OS less) applications.

```
C:\CPLDUtils>Post
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: FAILED
       temp_sensors: PASSED
       temperature: PASSED
       fnvram: PASSED
       ether_loop0: PASSED
       ether_loop1: PASSED
       ether_loop2: PASSED
       voltage: PASSED
       sata0_test: PASSED
       sata1_test: PASSED
       vpd: PASSED
       eeprom: PASSED
       usb1_controller: PASSED
       usb2_controller: PASSED
       system: PASSED

PASSED  : 24
FAILED  : 1
NOT RUN : 0
TOTAL   : 25
```

Figure 3: Example of a detailed PBIT result accessed from Windows XP embedded running on VX3030.

6. PBIT and system boot policy

PBIT itself can be configured to **HALT** the executing board in case of a test failure, to **STOP** and return to firmware prompt or even to **execute an alternate boot sequence**.

From the **operating system** level, various PBIT results collection methods can be selected at the application design time to fulfil the complete system test policy.
The Operating System API gives access to the complete PBIT FLASH results section. This gives the most detailed PBIT information required by a control and monitoring application to take a decision on the system following boot steps.
The malfunctioning device can be identified in the test report list stored by PBIT. The **PBIT synthetic result register** featured by the board control unit (cPLD) can also be used from a management unit such as Kontron **CMB** or from other SBCs linked **to the same SMB bus on the VME or VPX backplane**. Using this result, the controlling unit can use the same SMB channel to alter a given board Control registers and modify its behaviour.

7. PBIT efficiency in system production and deployments

Kontron PBIT software implements **unrivalled features** which make Kontron product deployment easier and cheaper.

When delivered with the board, the **PBIT code** is located in the **same device as the BIOS code**. It is deployed along with the BIOS (please refer to the PBIT user's guide). The PBIT settings (and results) are located in their dedicated non-volatile FLASH device. Duplicating these settings **on a clone system can easily be done** from the board firmware or using dedicated software delivered within Kontron Operating System BSPs for Linux, VxWorks or Microsoft Windows. This allows our customers to augment the floor plant productivity by allowing software tools to replace tedious step by step hardware configuration.

The **SYSTEM TEST** can also help for deployment. As this test scans all possible I/O ports and devices to compare with a pre-recorded system status reference, it can pinpoint and signal any change in the **expected I/O devices availability**. This can cover device failures, or more subtle system alterations (for example a faulty storage device or a network link left unconnected or an undetected VPX board on the backplane). The system test results are organized 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.
Note: for this mode, the delayed start timer implemented in the BIOS (from 0 to 900ms) can help guarantee that all the devices have a chance to become ready before the BIT session checks their status.

8. Maintenance and diagnostic of deployed systems: PBIT Failed Once flag

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. This long term specific flag can be used by maintenance teams to investigate the root cause of previously aborted boots, **even in situations where a system is regularly re-started**, which often occurs when the system operates under the control of a higher level control authority.

Updating the BIOS and the PBIT code is done at the same time, using binary images containing both software due 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 cases. Please contact **support-kom-sa@kontron.com** to open a feature request discussion with Kontron.

9. PBIT Execution Time

The PBIT test list can be modified in order to try various coverage/execution time trade-offs. PBIT execution time is measured between BIOS setup establishment and OS boot. Typical PBIT execution time is **around 10-20 seconds** for a complete board check, depending on the board and will be executed automatically before Operating System booting. This execution time **can be heavily reduced when necessary** if arbitration is needed between coverage rate and execution time.
Since PBIT tests each feature independently, all combinations of test list or execution mode (fast/slow, complex/simple) is permitted.

10. PBIT Coverage Rate

PBIT is designed to cover the maximum board feature set and thus can reach coverage of **more than 95%** of what is testable. This coverage rate is calculated from SBC hardware design information and PBIT software designer. It is based on the ratio of SBC signals really exercised during Firmware/ BIOS/PBIT execution.

11. Continuous Improvement

Since we also value the kind of service PBIT can deliver, we heavily use PBIT during our manufacturing process. Some special features have been implemented for this purpose. With this approach, our PBIT quality increases towards improved failure diagnostic and hidden system fault detection at each new board design.

12. Conclusion

For all systems that must guarantee their operational integrity and shall detect any faulty configuration or defect as early as possible, Kontron PBIT is the **complete, unique and ideal solution**.
Kontron Boards and Systems equipped with PBIT will avoid or reduce any added development at customer level to detect and react to a system fault.
Kontron PBIT integrates a very flexible interface with its various possible configuration modes and unitary tests.
The **SYSTEM configuration TEST** "learn" concept offers an innovative method to system-wide PBIT, so far unmatched in the Embedded SBC world.
Kontron PBIT is uniquely positioned to reach the expectations of **Military**, **Aerospace** and **Transportation** Embedded System market where the efficiency of system maintenances at lower cost becomes preponderant. With one of the best intuitive ease of use in the industry PBIT allows leading edge system BIT implementation to all Kontron customers, whatever their previous expertise in the test domain.

About Kontron

Kontron is a global leader in embedded computing technology. With more than 30% of its employees in Research and Development, Kontron creates many of the standards that drive the world's embedded computing platforms. Kontron's product longevity, local engineering and support, and value-added services, helps create a sustainable and viable embedded solution for OEMs and system integrators. Kontron works closely with its customers on their embedded application-ready platforms and custom solutions, enabling them to focus on their core competencies. The result is an accelerated time-to-market, reduced total-cost-of-ownership and an improved overall application with leading-edge, highly-reliable embedded technology.

Kontron is listed on the German TecDAX stock exchange under the symbol "KBC".
For more information, please visit: **www.kontron.com**

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

CORPORATE OFFICES

Europe, Middle East & Africa
Oskar-von-Miller-Str. 1
85386 Eching/Munich
Germany
Tel.: +49 (0)8165/ 77 777
Fax: +49 (0)8165/ 77 279
info@kontron.com

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

Asia Pacific
17 Building,Block #1,ABP.
188 Southern West 4th Ring Road
Beijing 100070, P.R.China
Tel.: + 86 10 63751188
Fax: + 86 10 83682438
info@kontron.cn