

ScanWorks® Embedded Diagnostics

Introduction

High-availability, mission-critical systems in such industries as telecom, military/aerospace, storage and high-end computing are expected to operate 24 x 7 with close to zero outage downtime. To achieve this level of performance, such systems must combine high reliability with rapid recovery from the inevitable software and/or hardware failure.

“Crashed” or “hung” systems are often the result of intermittent or catastrophic hardware failure from which the software cannot recover – for example, kernel drivers accessing faulty hardware which returns a bad result, or double-bit failures in RAM where the code was executing. These failure modes can also be caused by application module writes to protected store, out-of-range errors, infinite loops and other bugs. The root cause for such crashes must be determined to raise the overall quality, reliability and availability of embedded systems.

In some of these situations the system’s main processor will hang, and at that point a watchdog timeout will occur, and the system is restarted. It is crucial to gather forensic data on the system state prior to re-initialization. Such forensic data might include the system’s instruction pointer, stack pointer, particular areas of memory, and the state of the devices on the system’s printed circuit board.

ASSET provides embedded middleware solutions for running out-of-band (i.e. independent of the Operating System) diagnostics on Intel®-based and other CPU-based designs. Use of third-party middleware Intellectual Property (IP) reduces development costs and accelerates time-to-market.

Technical Description

ASSET delivers emulation test functionality to its customers via its benchtop processor-controlled test (PCT) solution. This application uses an external PC and hardware controller to access the debug port of printed circuit boards which use Intel®, Freescale™, AMD®, ARM®, etc. processors. PCT uses the on-board CPU to read and write registers, memory and I/O for the purposes of prototype bring-up, manufacturing testing and board debug & repair. This CPU emulation-based technology provides high test coverage, fast test times and discrete diagnostics, and addresses the limitations of legacy hardware-based solutions for today’s manufacturing floor.

This same technology can be embedded into a system to provide diagnostic capability under the control of the main element management system. The main emulation functions, such as setting breakpoints, single-stepping through code, and reading/writing registers, memory and I/O buses can therefore be provided remotely. This provides a very powerful capability of debugging the most intermittent problems which cannot be duplicated in lab conditions. It also enables the execution of user-written diagnostics, flash programming, remote testing, and other maintenance/management functions under user control or under autonomous control of the system. Within Intel embedded platforms, this is sometimes referred to as Remote In-Target Probe (Remote ITP).

The implementation topology involves connecting the debug port of the target board's processor to either a dedicated FPGA or service processor. Within the FPGA-based design, the FPGA runs the debug scan control via the target processor's JTAG port and any additional sideband signals which are required. An on-board or off-board service processor connects to the FPGA via a low-latency bus and hosts the run-control functionality. This is as shown in Figure 1.

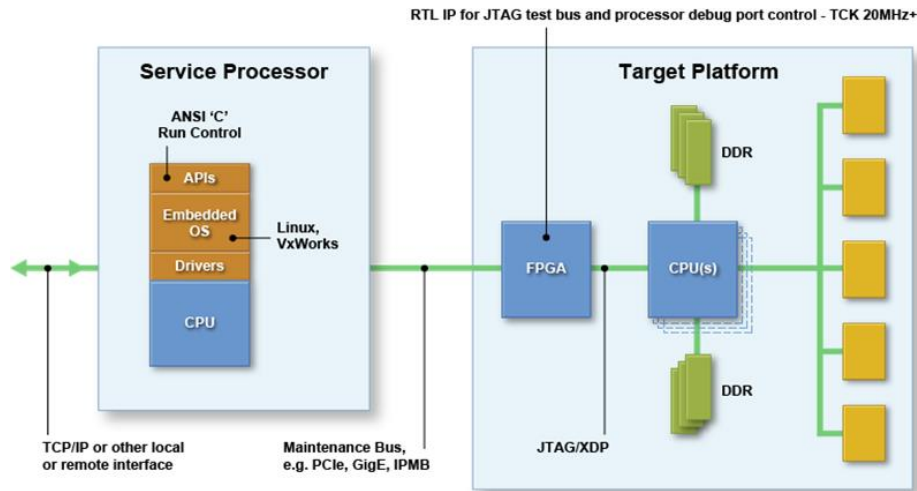


Figure 1

Within boards that only have a service processor with no FPGA, debug port control and API run-control all reside on the service processor. The service processor needs to provide discrete control of the JTAG and sideband signals, as displayed in Figure 2.

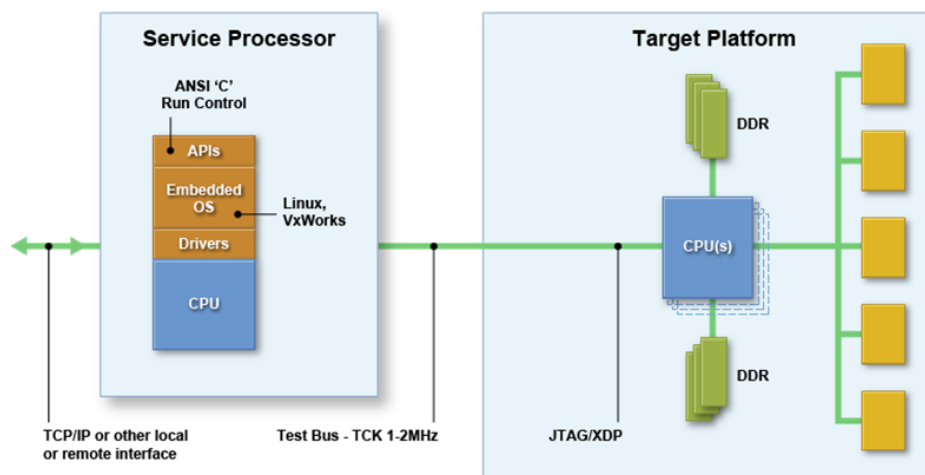


Figure 2

Use Cases

Embedded diagnostics solutions are typically used:

- In the lab, to troubleshoot system software during prototype development.
- At a system's initial deployment site during Installation/Commissioning, to diagnose problems during equipment set-up and turn-up.
- In the field, where hung systems are encountered and must be debugged.
- Back in the lab, where NTF problems need resolution.

All functions are provided via APIs which the OEM incorporates into its system software, callable via its element management system.

Since ScanWorks® embedded diagnostics are embedded in-system, they can be used throughout a product's lifecycle, from prototype, through production, to field deployment and repair. In many cases costly external test equipment can be avoided.

Capabilities Supported

For the above, some or all of the following low-level capabilities are provided:

- In a multi-processor or multi-board system, access to all CPUs and architecturally visible registers
- Dumping internal information from a processor hang
- Setting internal CPU breakpoints as supported by CPU debug interface
- Single-stepping of processor commands
- Dumping processor configuration, registers, and memories
- Running processor, memory, and I/O channel diagnostics

TAP Controller IP

The IP that implements the IEEE processor debug port control is a proven high-performance design that is utilized in several of ASSET's hardware offerings. The FPGA code is written in VHDL and will be provided (in obfuscated form with exception of the top level entity) to be included in the customer design. ASSET will need to know the target FPGA type to validate compatibility and ensure seamless integration with the customer design. Current implementations include several devices within the Altera® and Xilinx® families.

For cases whereby the debug port control is provided via a service processor, obfuscated 'C' code will be provided.

FPGA Resources

The resources required by the debug port control IP are small when compared to today's FPGAs. The IP will reside in fewer than 4,000 logic cells (or equivalent) and requires only 74k-bits of memory. The memory requirement is based on a 2k by 18-bit FIFO for both send and receive buffers.

It is assumed that the debug port control IP can share an existing information bus for data exchange between the controlling processor and FPGA.

The address space required by the TAP Controller IP is 128 32-bit locations of contiguous addresses within the FPGA.

Two global clock circuits will be required for routing the system and JTAG clock sources within the FPGA.

DFT Requirements

The FPGA or service processor must have access to a select number of the debug port control signals of the target processor(s).

For Intel x86 platforms for example, the applicable debug port control signals for a UP design are:

- TDI
- TDO
- TMS
- TCK
- TRST*
- XDP_DBR_R_N
- XDP_CONN_RESET_N
- CPU_PREQ_N
- CPU_PRDY_N
- PWRGOOD (optional)

The FPGA-based design will run at a continuous TCK of 20MHz. It is assumed that a low-latency bus exists between the FPGA and the processor executing the API code and drivers, such as PCIe, Ethernet, IPMB, etc.

In the case of no FPGA and with only service processor access to the debug port of the target processor, throughput will be 1 - 2MHz depending upon the service processor.

Software API

The embedded software is written in ANSI 'C' and is intended to be portable across multiple platforms. It requires a test control processor and embedded OS – specifically Linux, VxWorks or Windows Embedded. Resource requirements will vary across platforms and depending upon the application, but in general the core library and APIs use up 2MB. Among the interfaces provided are:

EnterDebugMode	Enters CPU debug mode
SetActiveCPU	Select the active CPU
SetActiveCore	Select the active core
MSRRead	Reads the contents of the specified Machine Status Register
MSRWrite	Writes contents to specified Machine Status Register
ReadGPR	Reads the contents of a CPU General Purpose Register

ReadCSR	Read Configuration Status Register
ReadMemory	Read contents of memory address
SetBreakPoints	Sets processor breakpoints
SingleStep	Execute a single CPU instruction (assembly)
ReadIO	Reads a UUT I/O port address in 8/16/32 bit format
IOBusTest	Performs a specified I/O data bus integrity check, checking for shorts, opens and stuck-at faults
DownloadHexFileToMemory	Loads UUT memory with a hex file
ProgramFlash	Programs a specified flash from a specified program Intel hex file
RAMTest	Performs a specified RAM test

More interface information is available upon request.

Conclusion

The ScanWorks embedded diagnostics solution from ASSET:

- Provides embedded diagnostics capability via a printed circuit board's CPU debug port.
- Is IP embedded within a board's service processor and/or FPGA.
- Provides breakpoint-setting and kernel trace/dump capabilities on Intel and other platforms.

Summary

ASSET provides a comprehensive portfolio of embedded diagnostics capabilities for the system designer. With these capabilities, designers and field support technicians can debug intermittent, irreproducible software and hardware defects in the lab and in the field which are often the source of costly NTFs and customer dissatisfaction. This allows companies to focus on continuous uptime, product differentiation, and high levels of customer service.

ScanWorks Platform for Embedded Instruments

The ScanWorks platform for embedded instruments is a seamless software environment that validates, tests and debugs circuit boards, chips and systems. It accesses and runs tests or other operations from any instrument embedded in your chips, circuit boards or systems, and then it collects, reports and analyzes the results.

ASSET Contacts:

Please contact your ScanWorks sales representative for more information.

ASSET InterTech, Inc.
2201 N. Central Expy., Ste 105
Richardson, TX 75080
+1 888 694-6250 or +1 972 437-2800
<http://www.asset-intertech.com>