Infrastructure IP for Configuration and Test of Boards and Systems

CJ Clark and Mike Ricchetti
Intellitech Corporation

Abstract

Advances in semiconductor technology and design automation, together with increased market competition, have driven engineers to achieve higher levels of integration with shortened development cycles. Expectations are that new products will have improved performance, more advanced features, and lower cost, when compared to the generation of products they replace. As a result, systems are becoming increasingly more complex and corporations are striving for higher levels of productivity from engineering teams. This creates challenges for engineers to design electronic systems with higher performance and increased functionality, while reducing manufacturing costs and improving time to market. Structured embedded configuration and test for PCBs and systems offers the best methodology to achieve these high product expectations if a parallel is drawn to the history of IC design and test. However, the opportunity costs and reduced engineering staff are making it increasingly difficult for engineers to dedicate time and resources to design and validate a robust system configuration and test solution on their own. Pre-engineered intellectual property (IP) infrastructure for embedded configuration and test of complex PCBs and multi-board systems can help engineers meet these growing challenges.

Introduction

The use of on-chip infrastructure IP has been increasing as a viable solution to many of the issues mentioned above amongst integrated circuit design engineers [1]. For ICs, yield and test concerns with very deep-submicron semiconductor technologies have forced chip designers to incorporate infrastructure IP into their designs to assist with silicon debug, to improve test quality and manufacturing yields. Examples of such infrastructure IP include Built-in-Self-Test (BIST) for logic and memories, Built-in-Self-Repair (BISR) for embedded memories, embedded core test logic for SoCs, and embedded timing analysis circuitry. The IP infrastructure is, in general, dedicated infrastructure, which is separate from the functional circuitry of the IC design, enabling re-use from one design to the next. The IC designer spends little time manually designing strategies for scan-based test. Design automation has facilitated the use of such IP infrastructure, and so the IC design-for-test effort has been greatly reduced.

As PCBs and systems mimic more characteristics of an IC – physical access only at the periphery - board and system designers are now confronting many of the same issues as IC designers have faced at the chip level. For example, advanced device packaging such as ball grid arrays (BGAs), along with the demand for product miniaturization, has made physical access to nets on PCBs and systems either extremely difficult, or simply impossible. As a result, traditional methods of physically probing interconnect for the purposes of prototype debug and manufacturing test are no longer possible. To further compound the problem, the higher clock speeds and data rates, that come with increased performance and high speed serial interconnects, often prevent physical probing due to signal integrity concerns. Testing products purely through functional means yields the same results as already learned by IC designers in the early 90s – low fault coverage and long test development times [2]. Infrastructure IP to support cost efficient debug, configuration and test of boards and systems is clearly desirable. While there are standards today at the device level for test and configuration [3], [4], there is a need to have a standard approach to accessing these devices on complex boards or in multi-board systems. Previous attempts to design infrastructure in-house have used ad-hoc methods and do not fully address the latest issues facing today's design engineers. The ad-hoc approaches are not easily accommodated by commercial ATE or PC based boundary-scan test equipment, further increasing engineering time for configuration and test development, validation and debug.

What is required is off-the-shelf, plug-and-play IP, that provides for a scalable in-system configuration, debug, and test infrastructure that is automatically understood by external ATE. This will enable system designers to build in

the field re-configurable and high quality self-testable products with a minimum of engineering time and effort. Furthermore, a unified approach to test and configuration will enable designers of these field adaptable products to lower their manufacturing test costs, field support costs, and extend the products' useful life. Infrastructure IP for the board and system level will save engineering time and will reduce design risk - since the PCB and system configuration and test details are pre-engineered. The solutions can be made to be scalable and re-usable, during all phases of the product life cycle, and from one product design to the next.

Problem Overview

Systems must be designed to meet the requirements of their configuration and test needs. Sound methods must be developed to support test and configuration throughout the life cycle of a product. Without the proper up front consideration, only ad-hoc methods will be possible. This is one of the major factors contributing to the high costs of designing, validating and testing a reconfigure-able system. The methods and strategies used for configuration and test play an important role in determining if a particular solution will be cost effective. Many of the current approaches to configuration and test are being challenged to keep up with the increasing complexity and cost constraints of today' s boards and systems. Some background on typical configuration and test methods for PCBs and systems, and the key issues to consider, are discussed below.

FPGA Configuration Issues

A compounding issue for system designers is the push for configurable products with a quick time to market. This has motivated designers to increase the adoption of programmable architectures. As such, a growing number of products now utilize programmable logic devices, such as FPGAs and CPLDs, and programmable non-volatile memories, such as EEPROM and FLASH. With continuously changing industry standards, field upgrade-able fixes and enhancements are becoming a common product requirement. The ability to remotely upgrade programmable system logic and memories in the field is a typical approach to addressing the requirement. However, obtaining the access to all of the programmable devices is increasingly more difficult, especially for PCBs with mezzanine cards or multi-PCB backplane based systems. For the system designer, these capabilities can add to the costs and design effort required to develop, and later to manufacture, such configurable products when the designer must create his own ad-hoc functional based methods for debug, configuration and test.

The method that is predominately used to configure FPGA logic employs special electrically alterable configuration PROMs. These PROMs are programmed with a design and loaded into the FPGA at power-up. A large FPGA, on average, can require from three to seven erasable PROMs for a single design configuration. The number of PROMs impacts PCB parts costs, area and layout time. Another limitation of PROMs is that they only hold one FPGA design. Today, in order to satisfy customer requirements it may be necessary to load a product with different FPGA designs based on the target use of the product by the customer. For example, it is sometimes desirable to load different protocols or algorithms into an FPGA based on the target communication medium or geographical location where the product will be used. In addition, configuration PROMs employ a proprietary method for programming FPGAs, so they are not interchangeable between different programmable device vendors. Finally, PROMs themselves require an on-board mechanism that enables them to be re-configured in-system.

These issues surrounding PROM based configuration are driving design engineers to explore new methods for in-system configuration. Unfortunately, many design teams have had to develop and create custom configuration capabilities. While on-the surface they may appear to be adequate, the downstream costs should be evaluated. A custom configuration method often ends up to be a 'one-off' design and is not cost effective due to the added part cost, increased engineering verification and debug time, lack of support by commercial design and test tools, added software integration costs and lack of re-use in the next version of a design. The integration of FPGA configuration with test cannot be overstressed. For instance, FPGAs now support various I/O logic families such as GTL, SSTL, HSTL, PECL and LVDS. To maximize fault coverage in a complex system, the FPGA based PCBs need to be tested in at least two scenarios, with the FPGA I/O configured and with the I/O un-configured. It is also desirable to program the FPGAs during boundary-scan test with small 'test' helper circuits to maximize test coverage and add at-speed tests. When FPGA configuration is done through a proprietary method, the loss of coordination of configuration and test through 1149.1 based test development and validation tools adds to

engineering time, increases test cost and complexity, especially embedded self-test complexity, and results in loss of fault coverage.

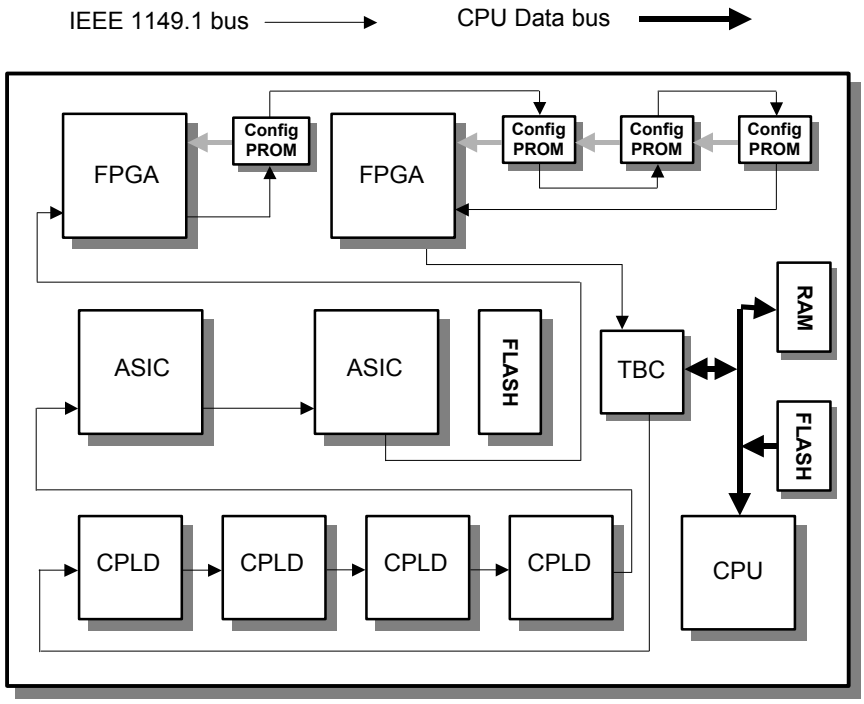

Figure 1. Traditional CPU and Test Bus Controller

Another common approach is to interface the system processor to an 1149.1 test bus controller IC [11] to reconfigure the PROMs, as shown in Figure 1. The approach has serious drawbacks in that it doesn't reduce configuration PROM costs or PCB area, since the PROMs are still needed for FPGA configuration. Another major drawback is that there is poor correlation between the PROM vendor's external programming tools and the TBC method. Successful programming of the PROMS with the external tools does not guarantee that the TBC and CPU firmware can program the PROMs without additional validation and debug. Configuration of FLASH based PROMS and FPGAs can be timing dependent; it is difficult to predict the timing, throughput and delivery of the configuration data with this approach. Interfacing to the CPU is not always straightforward, requires glue logic, and perhaps even a CPLD to provide peripheral addressing on the CPU address/data bus to the TBC. Glue logic is also needed around the TBC to allow access by external PROM and FPGA programming tools during bring-up and validation.

FPGA Configuration with System Processor

Designers are tempted to use the mission mode processor to program FPGAs directly, however, this creates hidden engineering and product costs downstream. The slow configuration speed prohibits practical application of multi-design FPGA loading "on-the-fly" based on a customer's application for some types of products. Systems that use a general purpose CPU take multiple minutes (4 minutes in one example known to the authors) to configure all of the FPGAs in the system. This affects throughput during manufacturing adding to the cost of the product since the FPGAs have to be configured and un-configured for the test process. In some cases, an entire system has to be integrated and 'booted' just to perform the FPGA configuration and IEEE 1149.1 tests, complicating prototype bring-up, system debug and manufacturing fault diagnostics. The Achilles' heal with this approach is system wide 'resets' that are controlled by devices, such as CPLDs, that also have IEEE 1149.1 test capability. IEEE 1149.1 based debug and test development, which should be straightforward, inherits undue complexity since basic boundary-scan instructions can toggle system and FPGA 'reset' pins in ways that the designer did not envision. The goal of IEEE 1149.1 itself, pin level diagnostics, is defeated, as a simple stuck at fault in a critical intersection between FPGA and CPU will negatively affect all of the serial data of the boundary-scan test. The perceptive PCB designer will take a cue from IC design-for-test and avoid integrating mission mode logic with serial configuration and test mechanisms.

The final major drawback is that custom firmware must be developed for the target PCB and each individual FPGA configuration must be validated and debugged for the entire life of the product. Software such as "JAM/STAPL" for CPLD/PROM configuration has been offered by FPGA vendors [12], however, there is little support and much consideration must be done in order for it to be used in a complex multi-board system. For instance, when this software is executed by the mission mode CPU and FLASH, additional care must be taken into account so when new FPGA designs are distributed in the field and programmed in the FLASH, the FLASH has no possibility of becoming corrupt and hang the mission mode CPU. Now, the IEEE 1532 standard has made the JAM/STAPL approach obsolete. These types of risks should be assessed before in-house development is considered. The on-going software development and software maintenance costs when target CPUs change, architectures change or software development teams change, also cannot be overlooked.

Test Issues

Current test strategies often employ In Circuit Test (ICT) equipment to test product subassemblies, such as PCBs. A key challenge with PCB test on ICT is that there is limited physical access on boards with surface-mount parts and BGA packaging. In addition, there are signal-integrity issues to consider when high-speed interconnects (e.g., SERDES) exist. In this case, the pogo pins of the ICT fixture will act as antennas that introduce unwanted capacitance and noise, and will interfere with high-speed testing. Further, there is a lack of scalability and an inability to re-use ICT manufacturing tests during other phases of the product's testing, for example in the field. These issues coupled with long test and configuration times suggest boundary-scan test on ICT is not a long-term approach or the cost effective place to execute IEEE 1149.1 based test and configuration.

A common approach to embedded test is one that utilizes traditional functional diagnostic code, as developed by test engineers and systems designers, stored on-board the product (e.g., stored in the CPU's FLASH). These "embedded tests" are used as a means to test the integrated systems, both in manufacturing and in the field. These functional test programs are ad-hoc, custom, embedded software applications. They require specialized resources to develop, validate and maintain, which results in high costs due to long development times and related engineering resources. Quality managers cannot deterministically measure the fault coverage of functional tests. The risks of software based functional tests are high. While significant resources can be invested, any software based functional tests that can't identify faults in the field or isolate faults enough to repair a failing PCB offer little value over running the system in mission mode and identifying that it doesn't function. Functional tests also require a (mostly) working system in order to execute, and so offer limited value in system-bring up and debug. As systems grow more complex, it is becoming impractical to continue with this approach, just as it became impractical to continue testing digital ICs purely with functional test.

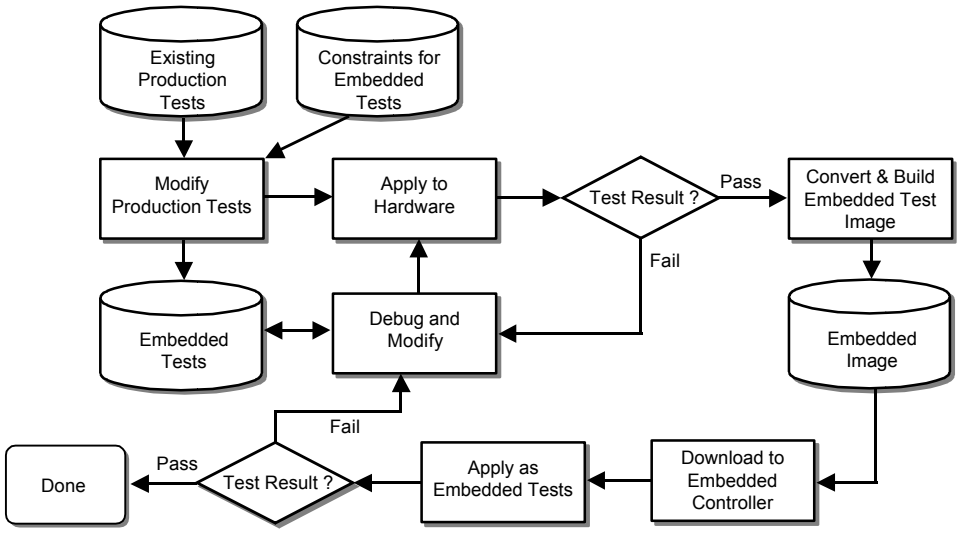

Figure 2. Traditional CPU and TBC Development and validation Flow

Embedded test using the mission CPU and IEEE 1149.1 test bus controllers has also been used; the authors have been involved in using this approach for over a decade. The approach enables a foundation of embedded

structural test for embedded functional tests but has its own shortcomings. As recently as late 2002, papers continue to be written promoting the method [10]. Figure 2 shows a typical flow used in this type of approach. Notice that it is necessary to modify existing tests and re-validate them after each modification. The modifications of the production tests are needed since the CPU, FLASH and connected devices cannot be tested while the CPU is applying the tests. Once they are modified, they must be converted for the embedded environment and validated a second time as embedded tests. The engineering resource cost of this approach when used for every test and every FPGA design on every system configuration should not be overlooked.

Non-Volatile Memory Updating

FLASH memory is commonly used in processor and micro controller based designs to store the system's firmware. The re-programmability of FLASH offers the flexibility for upgrades and bug fixes when product designers adopt a method for in-system programming. In-system programming also enables the FLASH devices to be programmed during the manufacturing process, using the latest, up-to-date, firmware.

One approach that is typically used for in-system programming of FLASH is to make use of the standard IEEE 1149.1 test infrastructure, using the EXTEST instruction to access the FLASH devices. This technique has limited impact on the design and only requires that the FLASH memory's address, data and control signals be directly connected to a IEEE 1149.1 compliant device, so that the device's boundary register can be scanned to perform FLASH write sequences in EXTEST. Although there is little design impact, in-system programming of FLASH devices using this method has serious performance issues. Today's designs employ numerous high pin-count boundary scan devices, which impact the length of the scan path. These long boundary-scan chain registers, together with low TCK frequencies, will have a major impact on in-system FLASH programming times. While much has been touted by adding physical access to FLASH Ready/Busy pins, when the time to scan a long boundary-scan chain is longer than the minimum FLASH 'ready' time, no benefit is gained by adding access. Incorporating physical access to the FLASH write enable (WE) input *can* cut the programming time almost in half, but, programming times in the tens of minutes are still unacceptable. The added cost of carrying the direct access through pins and routes in a passive backplane or multi-PCB design cannot be overlooked.

Configuration times of programmable devices during production manufacturing must also be considered. Given the increased need for in-system FLASH programming, CPLD configuration and FPGA PROM programming, configuration and test times on in-circuit test equipment range from 3 to 6 minutes in length for a "C sized" PCB and may reach as long as 10-12 minutes in the near future. Programming devices through IEEE 1149.1/1532 on in-circuit testers is not economically viable for certain product classes due to these lengthy programming times and higher dollar per hour cost of the ICT. Separating ICT from boundary-scan allows the continued use of ICT for what it tests best - accessible analog components and nets, non-boundary-scan 'glue' logic. The remaining IEEE 1149.1 based tests and configuration to be done afterwards on lower cost platforms or at the next level of test and integration.

Scan-Chain Design Issues

Partitioning of scan chains at both the board and system level is also an important issue to consider. To reduce test and in-systems device programming times it is often desirable to keep board level scan paths as short as possible. This reduces the number of TCK clock cycles required to program a FLASH or other device in-system. In addition, chaining devices that have different TCK rates together into one chain means that all devices on the chain must be accessed at the rate of the slowest device. Therefore, designers often use multiple scan chains at the board level to manage and reduce long scan chains. There are many challenges in designing PCBs and system that use multiple scan chains. For example, accommodating the various voltage levels of all the ICs on the board requires the designer to add voltage translators. Signal Integrity must be also addressed in the distribution of global TAP signals (i.e., TCK, TMS and TRST) to assure the best performance of the configuration and test bus can be maintained.

The complexity of a common configuration and test bus increases significantly at the systems level. With multiple board systems and passive backplanes, the significance of configuration and test time is even more of a challenge, as the number of assemblies can grow considerably. There is also the issue of fault coverage and diagnosis. For example, board-to-board interconnect testing must be considered to test for opens on devices connected to the backplane connector. As we move from a single PCB to assembled systems, design,

configuration and test complexity increases significantly as well as the expertise required to integrate and test the system.

We have developed novel approaches to these problems, which we describe in the remaining sections. We have created a complete development environment for structured embedded configuration and test. This will provide design and test engineers with comprehensive infrastructure IP that they can use to create, validate, and apply IEEE 1149.1-based system configuration and test suites. Our methodology can be used throughout the entire product life cycle, to help design engineers bring-up and debug prototype designs in the lab, to help manufacturers test products during volume production, to re-configure products in the field, and to test and diagnose systems.

A new approach - Structured Embedded Configuration and Test

Our approach provides a structured methodology for embedding configuration and test at the board and system level to take advantage of the cost efficiencies over the entire product life cycle. This reduces the design time for engineers, reduces test-engineering development and reduces test execution time. A structured embedded configuration and test approach is the ideal solution to address the problems associated with current configuration and test methods previously discussed. By including dedicated infrastructure IP in products, board and system designers can simplify in-system device configuration while enabling comprehensive structural test throughout the system. The embedded approach we describe provides the design and test engineer with a scalable and reusable methodology, which augments existing test and configuration standards. This embedded configuration and test architecture was designed to off-load ICT equipment, such that structural digital test and device configuration can be now be done in-system - which is more cost effective - while expensive ICT equipment can be better leveraged for analog testing. The architecture of our infrastructure IP was developed with the following objectives:

* The infrastructure should be completely scalable and reusable at any level of integration
* Support anytime/anywhere testing and in-the-field re-configuration for an entire multi-PCB system.
* Provide for scalable, highly parallel test and configuration to reduce reconfiguration times in the field and eliminate throughput bottlenecks in production test.
* Provide a foundation of high-fault coverage structural test from which to build functional test and diagnostics firmware on.
* Support in-system at-speed interconnects testing without the signal-integrity problems associated with ICT fixtures.
* Automate creation and validation of embedded tests and configurations. Cost-effective and provides for reuse at anytime during a product's life cycle.
* Greatly reduce or eliminate the need for physical access.

To implement our methodology we provide a family of patent-pending infrastructure IP. The IP may be delivered as embeddable cores, or as integrated designs that engineers can simply design into the board and system levels of their products. The basic building blocks include a board level scan ring linking device, an embedded configuration and test processor, a fast access controller, and a parallel test bus controller. The infrastructure IP blocks and the methodology are described in further detail in the following sections.

Board Level Scan Ring Partitioning

Commercial devices for 'linking' IEEE 1149.1 compliant scan-chains have existed since the early 1990s [5], [6]. The devices such as the TI Scan Path Linker and National Scan Bridge link four and three scan-chains respectively into a single path for PCB level interconnect testing, but enable the secondary scan chains to be enabled one at a time to reduce scan chain lengths for in-system configuration. However, few scan-rings, fixed 3.3V voltage levels and the lack of adequate distribution of critical signals for high-speed IEEE 1149.1 operation has limited their use in today's designs.

A scan ring-linking device was designed to provide PCB designers with infrastructure IP for easily partitioning scan paths at the board level. The architecture goals of the scan ring linking IP include:

* The linking IP can be embedded into a CPLD, FPGA or ASIC and can be configured to accommodate multiple voltage levels on N number of scan chains. This reduces parts costs, saving board area, and provides for a flexible implementation.
* The linking device handles distribution of global TAP signals: TCK, TMS and TRST. Providing better signal integrity and maximum performance.
* Provides a single entry point for IEEE 1149.1 access, allowing boundary scan access without the need for expensive, lower performance, ICT fixtures or equipment. The scan ring linker is fully testable and configurable with the associated external IEEE 1149.1 Test Development Environment tools.
* Provide for an easy interface to CPU scan-chains so emulation equipment can work hand-in-hand with IEEE 1149.1 based hardware debuggers.
* Provide direct access to all I/O of the IP so the IP itself can be in-system tested easily
* The linking device is fully compatible with, and can be integrated with, other infrastructure IP components.

The linking IP connects any number of scan rings (secondary scan chains) into a single path, which permits devices on secondary scan chains to be independently tested and configured through a single 1149.1 external interface. This dramatically shortens scan operations and thus enables maximum data throughput. The linking device also provides proper buffering of test signals, eliminating the need for additional external components on the board with untestable pins, such as buffers and voltage level translators as with the commercial devices described in [13], [14]. The PCB level scan ring linking IP was designed to provide the design engineer with an infrastructure IP module that can be easily embedded into a CPLD, FPGA or ASIC. The linking device is used at the PCB level to reduce the complexities and costs of designing IEEE 1149.1 test infrastructure for designs that require multiple scan chains. The linking device infrastructure IP can be configured for the required number of scan paths, voltages, etc. and is fully compliant with the IEEE 1149.1 standard. Availability of the linking device means that designers do not need to develop their own solutions that typically are not supported by commercial test tools. Development time is reduced and they can design in scan path partition at a higher level of integration, by simply adding the device to their board or ASIC design. The scan ring linker provides a plug-and-play solution for PCB designers who use multiple scan chains.

An example scan chain configuration at the board level using the linking device is shown in Figure 3. This shows how the linking IP can handle multiple scan chain partitions, with different voltage requirements. Since FPGAs, CPLD, and ASICs can be placed on separate chains, this helps speed the configuration times. It also makes it easier to keep ill-behaved devices, on other chains, in their safe states, while other devices are being programmed. Additionally, some vendor's devices will not program when devices from another vendors are in the same chain, thus they must be partitioned on separate board level scan chains. Processors can also be placed on a separate chain, allowing the CPU to be controlled for emulation (i.e., through a dedicated, stand alone, emulation connector) while the rest of the chains and on-board logic are accessed for hardware debug via the linking device.

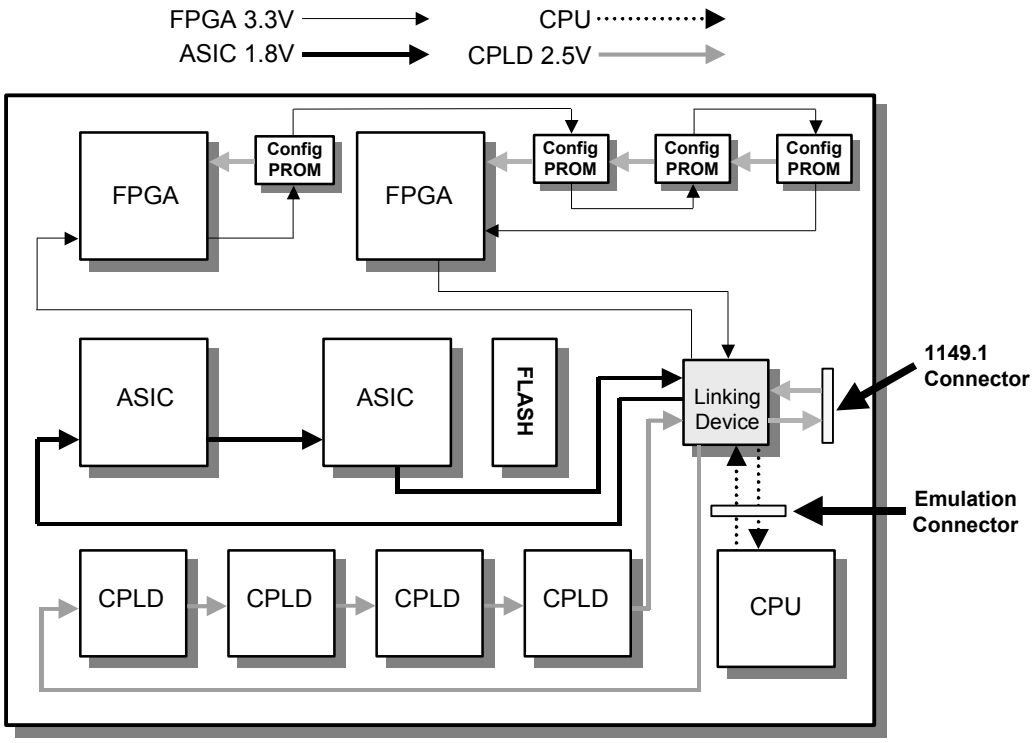

Figure 3. Scan chain configuration for a single board

Centralized Embedded Configuration and Test Management

It has been estimated that a PCB will be tested up to seven times during its product life [8]. This coupled with the desire to perform these tests in geographically disperse areas provides a compelling reason to embed test into the PCB itself. In our approach, a dedicated embedded configuration and test processor was developed to be a centralized manager for configuring and testing PCBs and systems [14]. We have many dedicated special purpose processors in systems today, network processors, audio processors, digital signal processors and video processors. There are many advantages to using them over using the mission mode general purpose CPU for such tasks. Dedicated architectures for embedded test enable testing of the general purpose CPU, testing the related CPU logic and logging of all failures without the need for the system to function. The configuration and test processor is designed around a novel architecture and is interoperable with all of the other IP infrastructure blocks. It enables manufacturing tests and device configuration suites to be developed and validated with automated PC based tools and subsequently automatically embedded into the system.

The goals of our architecture were:

* A code-less processor that reduces engineering design effort and is cost effective.
* A scalable and reusable methodology that supports centralized anytime/anywhere re-configuration and test of an entire system.
* Reduce board area and parts costs for FPGA/CPLD configuration
* Leverage 1149.1 and 1532 standards to provide vendor independent device configuration
* Manage different pre-programmed system designs enabling designers to create in-the-field configurable products.
* Provide detailed diagnostic data to enable PCB and System repair
* Enable integration with other infrastructure IP blocks.

In production manufacturing, board level device configuration and testing would typically be run using ICT. However, if we embed a dedicated configuration and test processor, we can eliminate the need to run configuration and digital tests on in-circuit test equipment. This lowers manufacturing costs by greatly reducing the time a board spends sitting on expensive capital equipment. Embedded test also allows the same set of high

quality tests to be used in many different environments and through all phases of the product's life cycle. This includes lab prototyping, volume PCB manufacturing, system integration, vibration test, HALT/HASS test, power-up self-test, field service and for depot repair. A dedicated configuration and test processor is able to manage multiple system configurations, so system re-configuration can take place anywhere and engineering changes can be easily made at any time during a product's life cycle.

The configuration and test processor was designed to be a vendor independent solution, leveraging the new IEEE 1532 standard and eliminating the need for proprietary PROM or FLASH based solutions. With this infrastructure IP, designers no longer need to develop customized solutions for embedded in-system configuration. Development time is reduced, and in addition, configuration and test are built-into the product. A single configuration and test processor at power-up, or under CPU control, can automatically run the entire manufacturing test stream, including scan tests, logic and memory BIST, and board/system interconnect tests as well as configure all the programmable logic in the system.

Figure 4 shows an example of how the configuration and test processor can be used at the board level. The configuration PROMs, as shown in Figure 1, are replaced with the dedicated processor. The processor interfaces to a FLASH device, which is used to store test and configuration suites, and drives the IEEE 1149.1 bus to the linking device, which links the multiple scan chains on the board. The processor also has an interface to an external 1149.1 connector, which is used to communicate to and from external validation tools. This 1149.1 interface is used develop and validate configuration and test vectors with traditional external PC based tool sets. This is a major advantage in that the external boundary-scan tools can communicate through the processor to the on-board 1149.1 scan-chains, thus guaranteeing equivalent drive and signal integrity for the on-board configuration and test mechanism as was achieved with the external PC-Based tools. The result is that just a single configuration and test validation step is required.

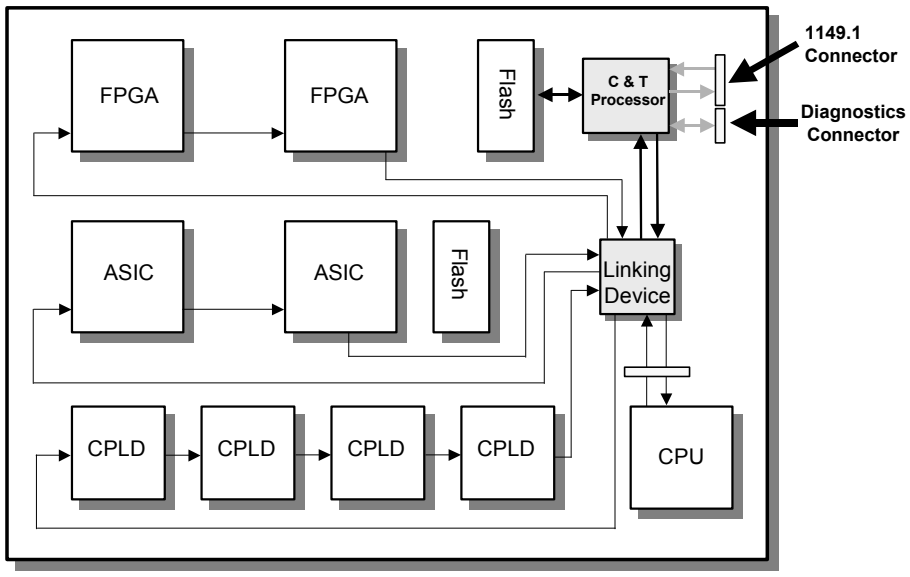

Figure 4. Embedded Configuration and Test Processor for a single PCB system

Once the test and configuration suites are finalized, they are downloaded into the FLASH and are ready for embedded execution. Subsequently, the external 1149.1 equipment is unplugged and the processor has control of running the embedded test suites and programming the FPGAs. The architecture provides flexible done and failure indications, including go/no-go LED, 7 segment LED displays for diagnostic failure codes, or a serial port for text based diagnostics. The diagnostics information, suitable for FRU (Field-Replaceable Unit) replacement or PCB repair, is embedded along with configuration and test data. This novel approach to embedding test and configuration architecture is "code-less", reducing engineering time; the exact same test and configuration vectors that are developed by the external tools for the board are embedded in the FLASH.

Figure 5 illustrates how the configuration and test processor connects externally to automated development tools for developing and validating configuration data and test programs. This figure also shows the simplified flow in our architecture over the more complex embedded test flow shown in Figure 2 as used in [9], [10], and [13]. We

use traditional PC based IEEE 1149.1 software ATPG and debug tools as the development environment and interface to an IEEE 1149.1 external controller, which connects to the PCB. The PCB, is shown with the configuration and test processor separate from the linking device, however, where desirable, the scan ring linking device can be integrated with the configuration and test processor to create a single packaged chip solution.

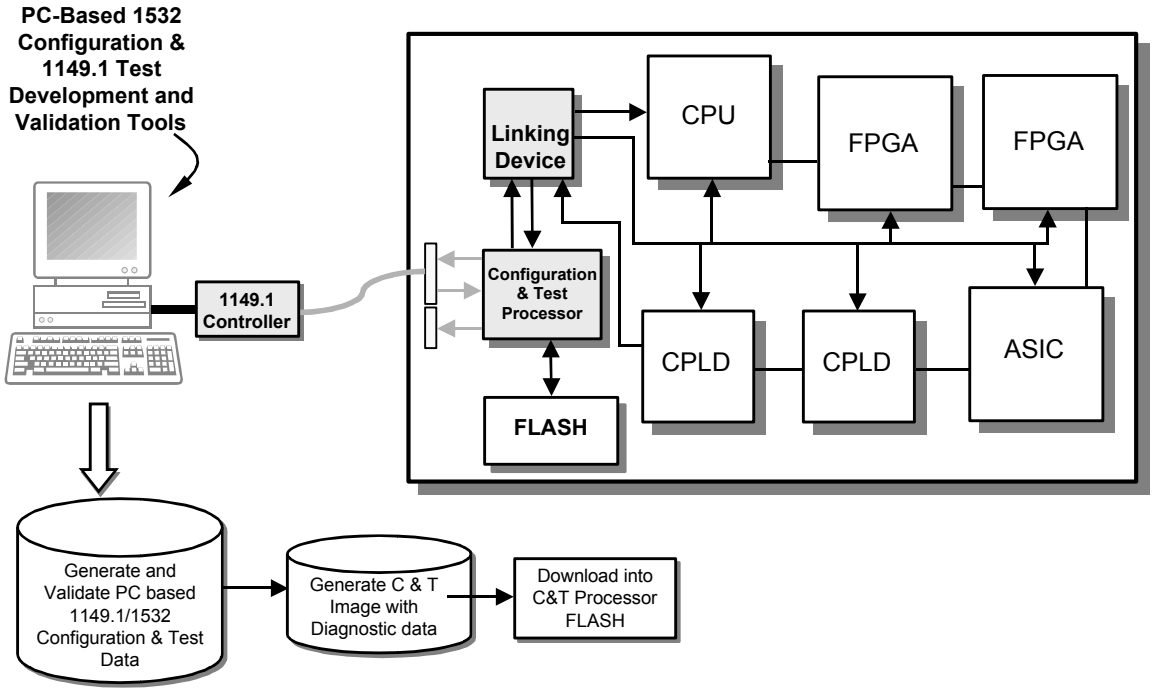

Figure 5. Optimized Embedded Configuration and Test Flow

Fast FLASH Programming

A complementary infrastructure IP module was designed to provide for high-speed, high-throughput, in-system configuration and test of external FLASH and memories [7]. To simplify discussion it will be referred to as the Fast Access Controller or (FAC). The FAC achieves its fast on-board programming times by using advanced data de-serialization and a programmable control protocol to minimize the number of scan operations and serial scan data required during FLASH programming and memory testing. This enables the FAC to program FLASH devices in-system over an 1149.1 bus at speeds equivalent to off-board or direct-access programming techniques. The memory protocols of the FAC are fully configurable in-system, through PC based boundary-scan tools. The FAC is also customizable, depending on the application, and this allows it to support access to a wide variety of FLASH and memory devices. With its' programmable capability, a single FAC can even be used to access multiple memories.

A block diagram of the FAC is shown in Figure 6. The FAC is embedded in an IC and interfaces to the 1149.1 TAP controller of the chip, and in this example to an external FLASH. The interface to the FLASH is through a multiplexer that selects between the system logic and the FAC. The FAC leverages the serial 1149.1 test infrastructure in a novel way to enable access to the memory in a parallel fashion. The patent-pending FAC method is scalable, and can achieve optimal programming throughput of FLASH devices, even with lower test clock rates (<3Mhz). In addition, the scan length or number of 1149.1 devices in the boundary-scan chain of the PCB does not affect the FAC architecture. The FAC enables the IEEE 1149.1 bus to be used as a central high-speed serial bus for in-system configuration of all on-board FLASH devices. Furthermore, it eliminates the performance issues associated with 1149.1-based in-system FLASH programming when using an EXTEST approach.

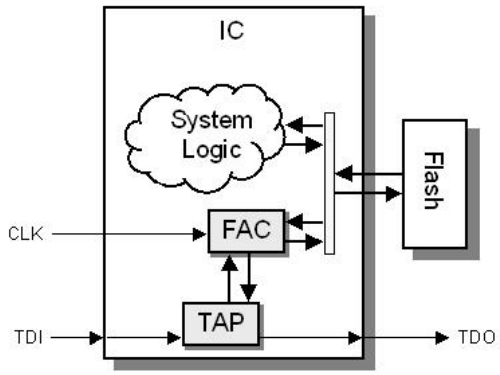

Figure 6. Fast Access Controller (FAC)

The FAC provides a plug-and-play IP solution for processor, ASIC/SoC and FPGA designers. For FPGA designs, it is also possible for the FAC to be temporarily downloaded. This provides a "zero" overhead approach that allows fast programming access to a FLASH device, and then afterwards, the normal system design can be re-programmed back into the FPGA. The method is compatible with other infrastructure IP. For example, the FAC can be used by the embedded configuration and test processor to provide designers with a complete embedded solution for in-system FLASH programming and configuration of CPLD/FPGA devices.

There is a growing use of memory storage and devices in today's electronic products, including embedded SRAM/DRAM memories, and non-volatile memories such as FLASH. Consequently, for debug and in production manufacturing, fast test access to large memories and programming of FLASH is becoming essential. Compared to the FAC, traditional methods are more costly and programming times for FLASH are much slower. In addition, the FAC provides for reuse during all phases of product's life cycle, where other methods do not. With the FAC, FLASH devices can be re-programmed during prototype bring-up, in production manufacturing or in the field using a unified methodology. Designers can lower overall product costs by using the FAC infrastructure IP and manufacturing can realize higher throughput through shorter programming times.

Parallel Configuration and Test

In developing our infrastructure IP, we have also incorporated novel capabilities to improve configuration and test of multiple board systems, and for greatly improving throughput in production manufacturing. The following sections describe these capabilities and how they improve upon current methods.

Multiple Board Systems

For multiple board systems with backplanes and removable PCBs, we need to consider configuration and test access not just for each individual board, but also for the entire system. This includes testing board-to-board interconnects and optimizing configuration and test at the system level.

While each board in a multi-board system has partitioned scan chains and a board-level access port (i.e., via the scan ring linking device), in order to access these board level scan chains at the system level we must include a way to "chain" the boards together in the system. Here, we must consider the configuration of the system, in that the system may have optional boards that may or may not be present in the backplane. The problem is that when a board is not present it leaves an open slot in the backplane, and this would break the scan chain if we simply chained the boards in series. The solution is to use what is commonly referred to as a multi-drop 1149.1 bus.

The signal connections for a multi-drop 1149.1 bus are shown in Figure 7. In addition to the global TAP signals (i.e. TCK, TMS and TRST) being bussed, the TDI and TDO are also bussed. Consequently, an open slot in backplane will not break the TDI or TDO connections to other boards in the system. Because TDI and TDO are bussed, it means that each of the N PCBs in Figure 7 will receive the exact same test data input stream on TDI, and that only one PCB must be driving TDO if we want to receive test data back on the bus. For this reason,

boards must be addressed or 'selected' to enable which board is receiving configuration and test data and which can drive responses back on TDO. Typically, this is done with special 'addressable' devices such as the TI 74ACT8996 [14] or the NS Scan Bridge PSC110F [6]. The detailed aspects of our particular multi-drop architecture are beyond the scope of this paper; the reader is encouraged to review them in the supplied references [15], [17]. Our architecture is especially novel and provides for innovative capabilities that are not available with current techniques that address only a portion of the issues in multi-PCB configuration and test. These features are discussed in the following section on parallel test.

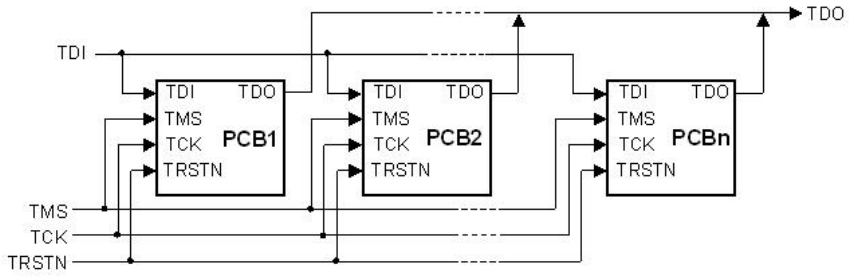

Figure 7. Multi-drop scan bus for a multiple board system

Parallel Test Bus Controller

To address configuration and test for multiple board systems, we have developed an architecture called the Parallel Test Architecture (PTA) [15], [17].

The goals of the architecture is as follows:

* Support for parallel test and configuration of PCBs over the IEEE 1149.1 multi-drop bus
* Support for system level multi-vendor PCB identification and addressing over IEEE 1149.1 multi-drop bus.
* Support for re-useable PCB-to-PCB interconnect tests in dynamic systems
* Simplify signal integrity issues on multi-drop backplanes
* Provide direct access to all I/O of the IP so the IP itself can be in-system tested easily
* Enable integration and use with other infrastructure IP blocks.

The main component of the PTA is infrastructure IP we will refer to as the Parallel Test Bus Controller (PTBC) for simplicity. Of the many features of the Parallel Test Bus Architecture, the most innovative is that it allows PCBs of the same type in the system to be configured or tested in parallel over a standard IEEE 1149.1 multi-drop bus. So for example, a system with ten PCBs of a certain type can be configured and tested simultaneously. The obvious benefit of this is that it reduces configuration and test times dramatically, which lowers the overall cost of the product. If it takes 2 minutes to configure and test a single PCB, it takes only 2 minutes to configure and test ten PCBs in a chassis, not 20 minutes. This landmark achievement in IEEE 1149.1 test can be accomplished with just a few hundred flip-flops and gates added to an ASIC or FPGA design. The utility of unlimited parallel test over IEEE 1149.1 in other areas of digital test, such as high volume consumer products and even integrated circuits, has not escaped the authors, however, further discussion is beyond the scope of this article.

The PTBC also supports position independent board-to-board interconnect testing with specialized instructions used to optimize the access between boards. Those skilled in the art of PCB-to-PCB interconnect testing with multi-drop ICs from TI and National Semiconductor and others can appreciate that system level interconnect tests cannot be re-used since the physical address of each PCB must be incorporated in the test patterns. If three PCB types or versions of the same type are targeted for a 4 slot system, and the PCBs can be plugged into the system in any combination, there are some 3^4 or 81 possible system level interconnect tests needed. Using the patent-pending address 'aliasing', the PTBC based system would only need six interconnect tests for any system built or updated in the field. Another important aspect not addressed by commercial multi-drop devices is integrated PCB type identification over IEEE 1149.1. In multi-vendor systems, such as cPCI or VME it is key that

vendors can provide unique identification for their PCB over the multi-drop bus. Designers have developed ad-hoc methods of PCB identification and PCB addressing through I^2C busses, Serial PROMS and other functional based methods. When these approaches cannot easily be accessed and controlled by embedded test mechanisms or understood by standard test development tools, the complexity of configuration and test development increases dramatically. Architectures that require the system to be 'booted' to perform this identification slow manufacturing throughput, and are of little use for system bring-up or logging of failures in the field. The Parallel Test Architecture and the parallel test bus controller provides for vendor independent PCB identification. Multi-vendor PCB identification is critical for embedded configuration and test processors to automatically identify what PCBs are present in a dynamic system and what tests and configuration data needs to be applied.

Figure 8 shows how the PTBC is implemented on a single PCB that is designed to be a 'blade' in a backplane based system. The PTBC interfaces to the bussed IEEE 1149.1 signals through the system backplane. The slot addressing inputs and type identification inputs are not shown. In this example, the PTBC also links the scan chain partitions on the PCB. This is achieved by integrating the board-level scan ring linking device, as was previously described, with the PTBC IP. The slave PCB in

Figure 8 also contains two embedded fast-access-controllers used for programming the on-board FLASH in-system over the parallel test bus.

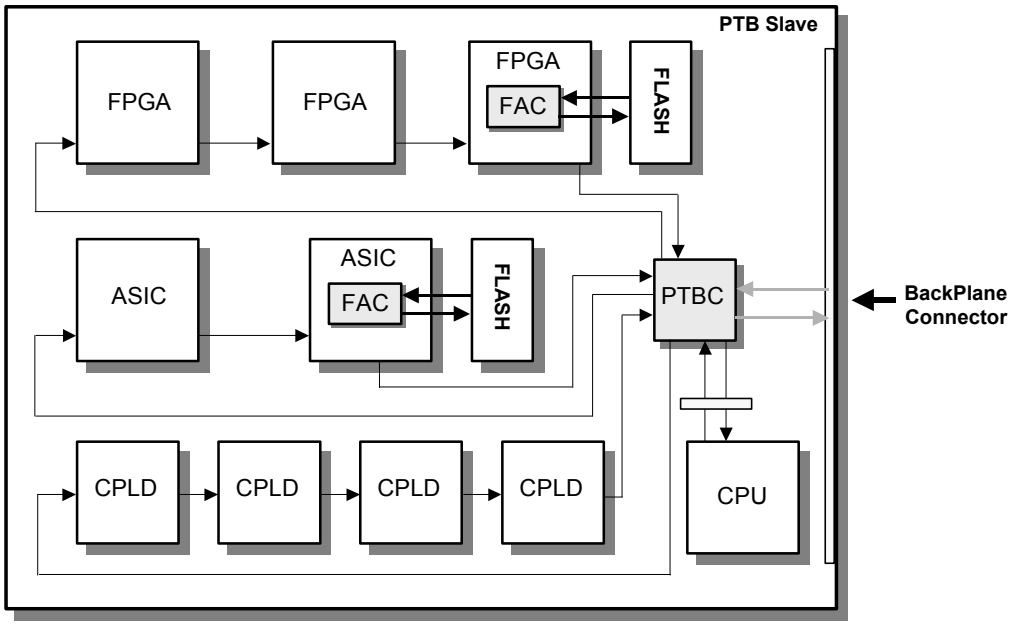

Figure 8. Scan chain configuration for a multiple board system using the PTBC

Figure 9 shows a multi-board system based on the Parallel Test Bus Architecture. Each system board has a PTBC on-board and a multi-drop connection to the PTB. The external 1149.1 controller and embedded processor can be used together for validation prior to embedding configuration and test data in much the same way as was described previously for a single PCB. In this architecture, the embedded configuration and test processor enables stand-alone PCB self-test on master/slave PCBs, embedded system level parallel configuration, embedded parallel test and system level interconnect. When slave PCBs optionally have a local configuration and test processor, that architecture enables parallel test of like and unlike PCBs in a system. Master/Slave PCBs also enable a system to be built with all PCBs of the same type, where the master is determined on the fly. The system shown provides optimal test and configuration times with a minimum of FPGA configuration

validation, easy access for prototype debug, straight forward interconnect test development with a minimum number of tests, configuration and test of the entire system separate from the mission mode logic enabling in-the-field fault logging. The centralized control over logic/FLASH configuration enables upgrades in the field and simple pre-embedding validation by external 1149.1 tools.

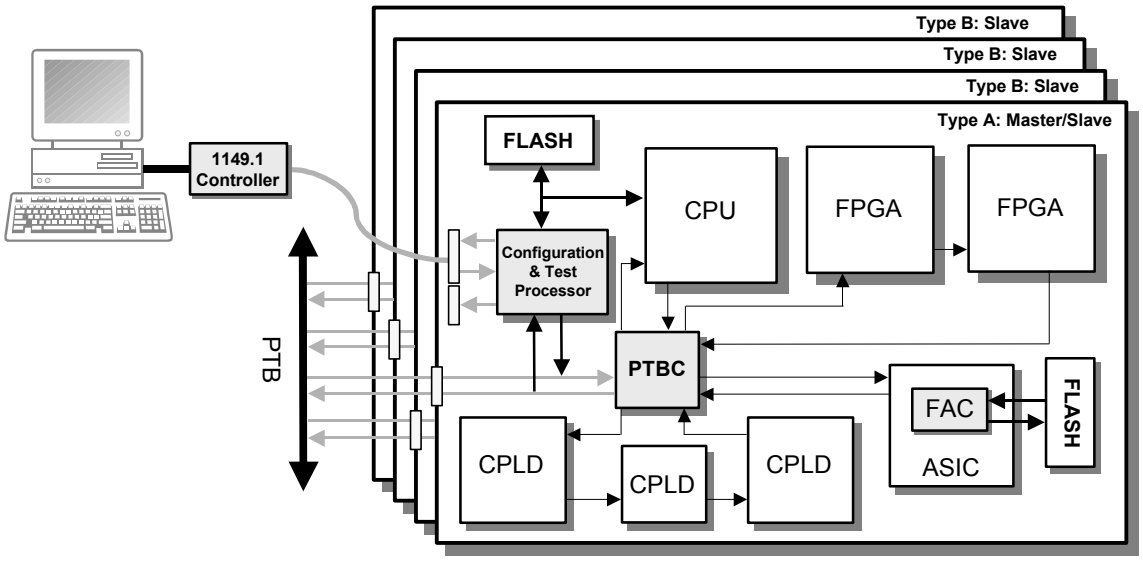

Figure 9. Optimized Architecture for embedded configuration and test of a multi-board system

Conclusions

We have described a family of infrastructure IP that can be used by board and systems designers to enable novel new approaches to configuration and test of systems. The problems associated with combining mission mode CPUs and mission mode logic for embedded configuration and test was described. The leveraging of new standards such as IEEE 1532 for configuration of programmable devices and its importance for the purpose of embedded test was demonstrated. Merging FPGA configuration and embedded test functions onto a common high throughput configuration and test bus offers the best long-term strategy for building re-configurable and self-testable systems. The infrastructure IP shown here is interoperable, in that each member of the family was architected and designed to work with the others and independent of system operation. Any of them can be used alone, or combined into an integrated solution. The IP will lower the overall product costs, reduce configuration and test time, and provide new methodologies that improve upon current approaches. Key to the infrastructure IP is its re-use, both in terms of the IP itself and the reuse of configuration and test suites in all phases of the product life cycle, from prototype bring-up to the field.

References

[1] Y. Zorian, “Guest Editor Introduction: What is Infrastructure IP?”, *IEEE Design and Test of Computers*, vol. 19, no. 3, May-June 2002, pp. 5-7.

[2] C.J. Clark, “Can System Test and IC Test learn from each other?”, *International Test Conference*, October 2002.

[3] IEEE Std 1149.1-2001, “IEEE Standard Test Access Port and Boundary-Scan Architecture”, Institute of Electrical and Electronic Engineers, Inc., New York, NY, USA.

[4] IEEE Std 1532-2000, “IEEE Standard for In-System Configuration of Programmable Devices”, Institute of Electrical and Electronic Engineers, Inc., New York, NY, USA

[5] Anon., "Scan Path Linkers With 4-Bit Identification Buses Scan-Controlled IEEE Std", *Texas Instruments Data Sheets*, SCAS157D, December 1996

[6] Anon., "Scan Bridge Hierarchical and Multidrop Addressable JTAG Port", *National Semiconductor Corp.*, Data Sheet SCANPSC110F, October 1999.

[7] Ricchetti, M. Clark, CJ, Dervisoglu, B., "Method and Apparatus for Providing Optimized Access to Circuits for Debug, Programming, and Test", *PCT Patent Application WO0171876*, World Intellectual Property Organization, Geneva, Switzerland, March 23, 2000. http://ep.espacenet.com/

[8] Parker, Kenneth, et al.,” Boundary Scan Signals Future Age of Test”, *EP&P*, 7/1/2002.

[9] Noeninckx, Greg, “CPLD Reprogramming – A Practical Application of an 1149.1 Backplane Bus”, *Digest, Board Test Workshop* (BTW02*)*, Baltimore, MD, October 2002.

[10] Van Treuren, Bradford G., Miranda, Jose M., “Embedded Boundary Scan Testing”, *Digest, Board Test Workshop* (BTW02*)*, Baltimore, MD, October 2002.

[11] Forstner, P. "Test-Bus Controller SN74ACT8990", *Texas Instruments Application Report*, SCAA044, August 2000.

[12] Anon., "Using JAM STAPL for ISP & ICR via an Embedded Processor", *Altera Corp.*, Application Note 122, March 2000

[13] Harrison, Stephen, et.al., “Hierarchical Boundary Scan a Scan Chip-Set Solution”, *Proc. Int’l Test Conf.* (ITC 2001), IEEE Press, Piscataway, NJ, 2001, pp. 480-486.

[14] Ricchetti, M. Clark, CJ, "Method and Apparatus for Embedded Built-In Self-Test (BIST) of Electronic Circuits and Systems", *US Patent Application 10/142,556*, US Patent and Trademark Office, Washington, D.C., December 4, 2001. http://www.uspto.gov

[15] Anon., "10-Bit Addressable Scan Ports Multidrop-Addressable IEEE Std", *Texas Instruments Data Sheets*, SCBS489C, April 1999.

[16] Ricchetti, M. Clark, CJ, "Method and Apparatus for Optimized Parallel Testing and Access of Electronic Circuits", *US Patent Application 2003009715*, US Patent and Trademark Office, Washington, D.C., July 5, 2001. http://www.uspto.gov.

[17] Clark, CJ, Ricchetti, M., "Method and Apparatus for Optimized Parallel Testing and Access of Electronic Circuits", *PCT Patent Application WO03005050*, World Intellectual Property Organization, Geneva, Switzerland, July 5, 2001. http://ep.espacenet.com

CJ Clark is the President and CEO of Intellitech Corporation. He has been the elected chairperson of the IEEE 1149.1 working group and served from 1996 to 2002. Under his direction, the IEEE 1149.1 Standard was re-released with improvements in 2001. CJ is also member of the IEEE 1149.4, IEEE 1532 and P1500 working groups. He has given many papers and presentations at industry conferences and workshops such as the International Test Conference, VLSI Test Symposium, TECS (Testing Embedded Cores-Based Systems) Workshop, the Board Test Workshop, Ottawa Test Workshop and VLSI Test Symposium. CJ also serves on the University of New Hampshire College of Engineering and Physical Science (CEPS) Advisory Board. He also serves on the UNH Department of Electrical Engineering Advisory Board.

Mike Ricchetti is Chief Technology Officer at Intellitech Corporation. He is well known in the Design-for-Test industry for his expertise in DFT methodologies such as core test and BIST. He is a Member of the IEEE and the Computer Society's Test Technology Technical Council (TTTC). Mike has been active as a Working Group member in several IEEE Test Standards, including 1149.1, the 1149.6 Advanced I/O Test Standard and the P1500 Embedded Core Test Standard. He also chairs the TTTC Technical Activities Committee on Silicon Debug and Diagnosis. Mike has over 20 years of experience in Design for Testability and he has authored several papers, and participated in many technical panels, in the area of Design for Test.

You may also direct questions or comments about this article to CJ Clark, Intellitech Corporation, 70 Main Street, Durham NH 03824; (603) 868-7116; cjclark@intellitech.com.

