FAST FLASH PARALLEL IN-SYSTEM PROGRAMMING

BY KENT ZETTERBERG

By Kent Zetterberg – Product Manager

Kent Zetterberg started his career in the automation industry, working with systems from ABB, Siemens and others. Following graduation from the University of Gävle with a Bachelor's of Science Degree in Computer Engineering, he worked 15 years in the telecom industry where he held various positions involving hardware test and debug. He joined Ericsson AB in Sweden in 1997 where he developed functional test programs for processor boards, and designed interface boards and test fixtures. At Ericsson he became an expert in boundary scan and eventually led the boundary scan team. With ASSET Kent has held several positions in support, serving as a customer trainer and European support team leader. Currently he is the technical product manager for ScanWorks boundary-scan test products.

Table of Contents

Table of Figures

Executive Summary

The value of programming flash memories in-system – after the chips have been soldered to a printed circuit board (PCB) – is well established. Unfortunately, the industry seems to have taken a step backward to the 1990s as the increasing size of flash memory devices has driven manufacturers to program flash outside of the regular assembly flow with stand-alone programming stations before populating a PCB with programmed devices. Instead of capturing the greater efficiencies of in-system programming (ISP) integrated into the assembly line, manufacturing engineers are relegating memory programming to off-line processes because the capacity of flash memories and the amount of data being programmed into them has increased to the point where legacy ISP methods have become too slow to be tolerated in the manufacturing flow. The speed of every process in an assembly line is critical to achieving cost-effective production rates or maintaining manufacturing's 'beat' rate. Any particularly slow in-line process might bring a production line to a crawl.

Formerly, methods based on boundary-scan (JTAG) access to PCBs and the devices on them had been a mainstay for ISP as well as structural and electrical testing in manufacturing. Designers have also employed boundary-scan systems from certain suppliers to validate high-speed I/O buses before the design moved into manufacturing. Boundary-scan technology, which was developed in the 1990s, has demonstrated over the years that it can be quite adaptable. In fact, several innovations have been developed in recent years that enable high-speed in-line ISP quite effectively with boundary-scan tools. Increasing the speed of the boundary-scan test clock (TCK) is certainly one way to accelerate in-line ISP, but another more advanced method is based on temporarily configuring an on-board FPGA with intellectual property (IP) that defines an embedded instrument for ISP. This method not only accelerates the rate at which a flash device is programmed, but it also supports parallel or concurrent programming whereby multiple PCBs each with their own flash devices can be programmed simultaneously by one boundary-scan station in manufacturing, speeding up the ISP process even further.

This eBook discusses several methods that can accelerate ISP and bring it back into the production line. In addition, high-speed ISP is described in several different manufacturing environments, including high-volume, and low-volume/high-mix production scenarios.

Introduction

The benefits manufacturers have enjoyed from programming flash memory in-system and as part of the board assembly process is being threatened by the larger capacity of memory devices and the increasing amount of data that must be programmed into these devices. Ideally, manufacturers would prefer to assemble PCBs with empty flash devices, such as NOR/NAND memories, SPI, I2C and others various types of components with on-chip memory. With empty memories assembled onto a PCB, the manufacturer can then load data or software into the devices as a step on the assembly line. By adopting ISP, manufacturers were able to program these devices at practically any time, even after PCBs have been fully assembled and placed in inventory. In this way, user-specific and application data or code can be programmed into flash memories just-in-time for delivery.

In addition, the firmware or application software for a new PCB design is often not complete when prototype PCBs are first available for design validation and board bring-up. Or, the firmware might be changed because of a newly found bug after a previous version of the firmware had already been loaded into PCBs. All of these situations point up the value of ISP over de-populating assembled PCBs to program memory devices individually. The boundary-scan or JTAG port on circuit boards and many components continues to be an excellent entry point for ISP methods. In fact, recent tool and technology enhancements have made boundary-scan ISP methods even more effective over legacy boundary-scan methods.

Distributed programming with boundary scan/JTAG

Boundary-scan/JTAG test systems typically connect to the PCB that is being programmed or the unit-under-test (UUT) through an interface module or pod (Figure 1).

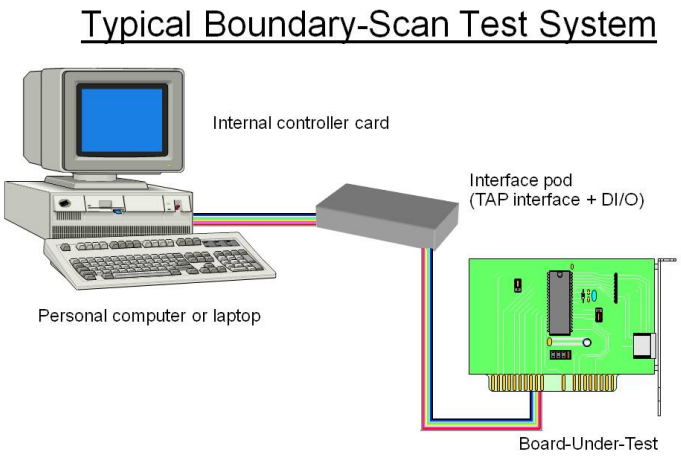


Figure 1: A typical boundary-scan programming and test station

Some suppliers of boundary-scan tools have integrated intelligent hardware with processing power and memories into their interface pods. Because of this, the boundary-scan system's programming application can be distributed from the PC to the interface pod. This means that multiple boundary-scan controllers can be configured in the same boundary-scan system. Since the interface pods can communicate over a common interface like Ethernet, applications, such as ISP, can be truly distributed, enabling all of the benefits of a distributed system architecture.

For example, consider a design which must be programmed in-system, except legacy boundary-scan-based/JTAG programming methods would require 120 seconds per PCB and the manufacturing circumstances are such that the entire process must be accomplished in 12 seconds. In other words, the legacy JTAG ISP method would have to be accelerated by a factor of 10, not a simple assignment for any programming tool.

A distributed and parallel programming application, which can apply additional resources to the problem in a coherent and concerted fashion, is capable of meeting these requirements. With a distributed programming application, the programming vectors may be applied to multiple devices on multiple boards in parallel (at the same time) or each board can be programmed asynchronously, depending on the particular requirements of the assembly line and how the manufacturing flow prior to programming is configured. With such a distributed programming

application, the test executive that manages the entire process must be very flexible so that it can handle practically any scenario. (Figure 2)

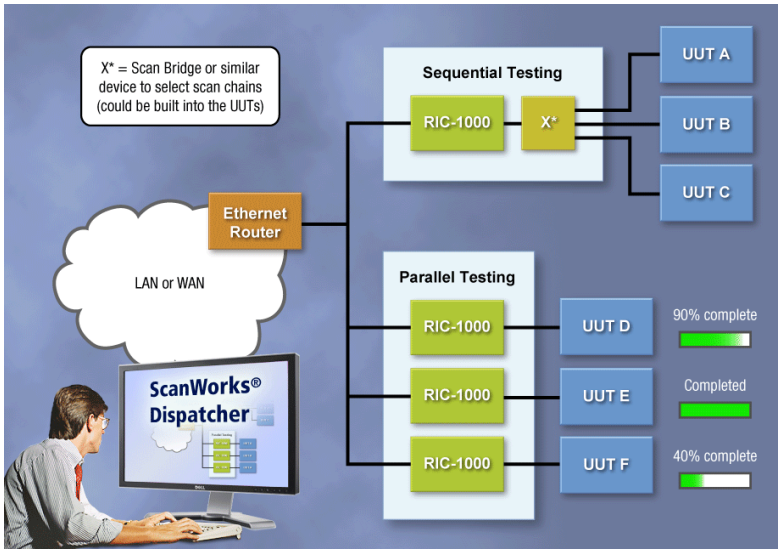


Figure 2: Parallel programming and test with distributed vector application

In certain cases, a distributed boundary-scan programming and test system can handle hundreds of PCBs, each based on a different design, at the same time. The following sections of this eBook describe several production scenarios where this distributed boundary-scan programming methodology can be effectively deployed.

It should be pointed out that the same boundary-scan tool employed in an in-line, in-system parallel programming application can also perform a number of structural and electrical tests on a PCB. Explaining these testing procedures is beyond the scope of this eBook.

High-volume programming and test

Many high-volume manufacturing operations typically assemble PCBs on parallel assembly lines where programming and test are performed on each line simultaneously. The manufacturing engineers who design and implement the overall manufacturing flow are usually intent on eliminating or alleviating any process that could possibly disrupt the manufacturing flow and cause bottlenecks in assembly operations. If device programming and test might slow down the manufacturing 'beat' rate, these processes might be moved off-line outside of the main manufacturing flow. From the standpoint of manufacturing efficiency, this would be a worst case

scenario, because it would add additional steps into the manufacturing flow, slowing it down even further and requiring expensive capital equipment expenditures for off-line programming stations. In some cases, the programmable devices may have already been assembled onto the PCBs. Depopulating each PCB individually to reprogram the on-board devices and then repopulating each PCB is extremely time consuming, resource intense and laborious. The following several graphs illustrate how distributed in-system parallel programming and test can be integrated effectively into an efficient high-volume production flow. (Figure 3 and 4)

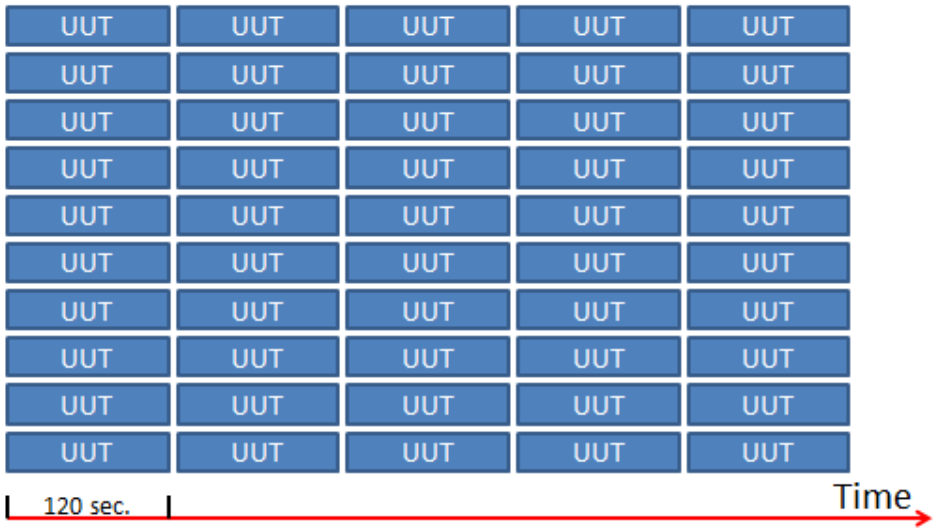


Figure 3: True parallel programming and test. Each programming or test step is started and stopped at the same time.

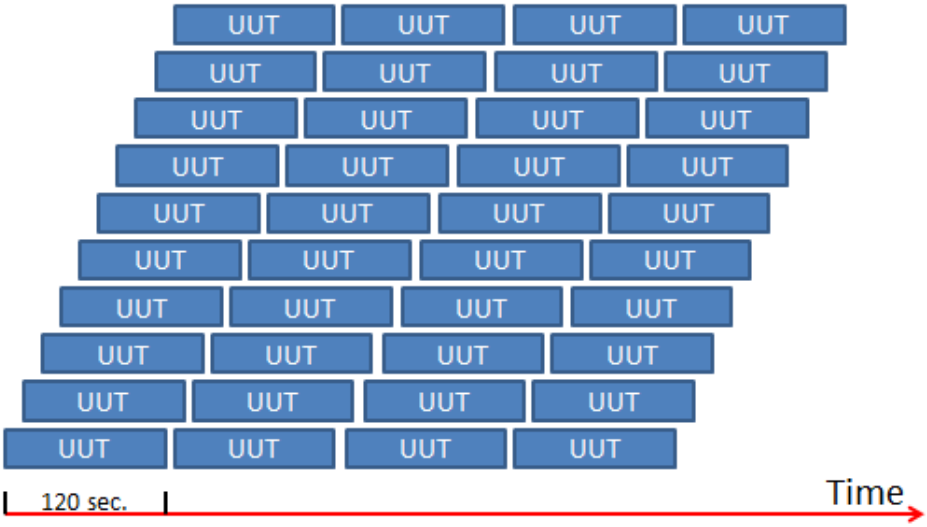


Figure 4: Asynchronously starting programming or test. Each step starts independently of the others.

High-volume manufacturing will most likely require a rack-mounted implementation of the programming and test applications. (Figure 5)

Figure 5: Rack-mounted backplane for programming and test applications.

Low-volume / high-mix test and programming

Some manufacturing plants regularly assemble a great variety of different designs in relatively small numbers. Electronic manufacturers serving the defense, avionics, automotive and several other industries typically run their plants with this sort of low-volume/high-mix of products. The challenge this presents is how to apply the same production systems and assembly machines on many different designs simultaneously and asynchronously. In these types of scenarios, the programming and test steps in the manufacturing lines still face the same kind of speed requirements. (Figure 6)

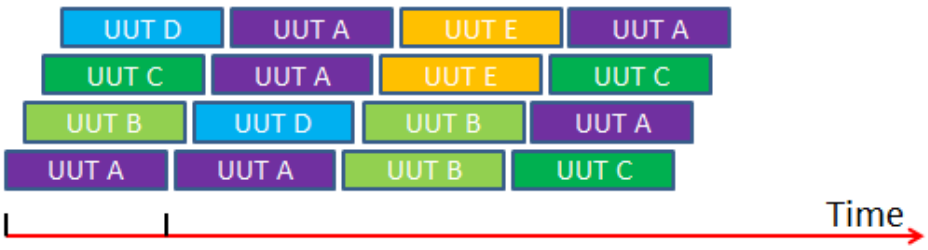


Figure 6: Programming/test in a low-volume/high-mix manufacturing setting..

The case illustrated in Figure 6 is most likely an environment where the boundary-scan programming/test systems are mounted in racks on the manufacturing line. The PCBs or UUTs might arrive at the programming station in batches of four, for example. Each batch of four PCBs might be based on several different board designs. The programming/test system may be equipped with bar code scanners that automatically scan the board's identity so that the correct programming files are loaded and appropriate test vectors are applied. Although Figure 6 is easier to understand, it does not illustrate a real world scenario insofar as programming and test processes will likely not require the same amount of time on each design.

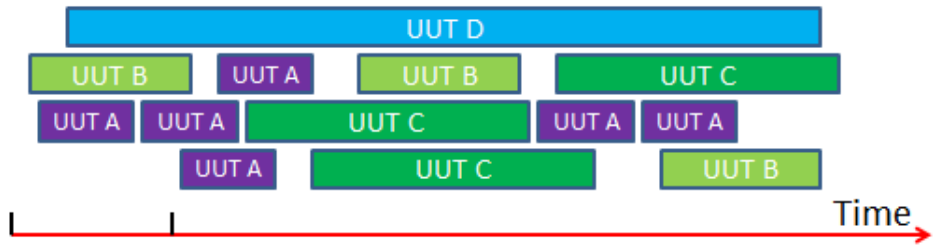


Figure 7: A realistic illustration of a low-volume/high-mix manufacturing setting.

The second low-volume/high-mix case illustrated in Figure 7 is probably closer to reality than Figure 6. Figure 7 shows varying amounts of programming time for the different designs. In this type of scenario, a more flexible backplane solution for the boundary-scan programming and test station is more likely. This would involve an operator manually populating the PCBs as they flow through the production line. This is a very effective way to maximize the utilization of the programming and test equipment when a numerous designs are being assembled and each has its own programming time.

Although rack-mounted boundary-scan programming and test systems could be employed in both of the manufacturing scenarios described above, a more flexible fixture-based test system (Figure 8) could also be incorporated into a low-volume/high-mix production line.

Figure 8: A fixture–based programming and test system.

Mid-volume / mid-mix programming and test

While every production line is unique, boundary-scan programming and test tools are flexible to meet every need, including a mid-volume/mid-mix production environment (Figure 9).

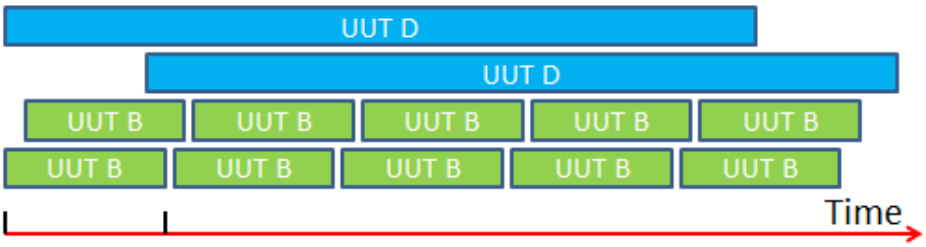


Figure 9: Mid-volume/mid-mix production environment.

Real world scenarios

All of the various programming and test processes described in this eBook have been implemented by manufacturers who serve many different types of markets. The following briefly

describes several real world scenarios and the benefits the manufacturer has derived from parallel in-system and in-line programming.

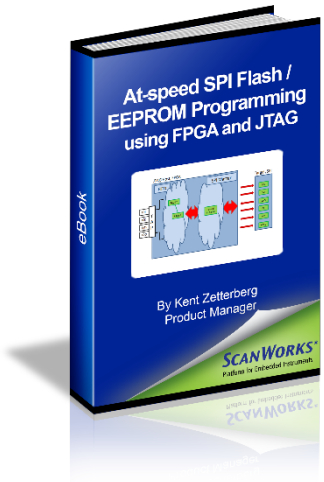
- Scenario: one rack–mounted station programs and tests in parallel more than 20 PCBs that are all based on the same design.
    - Benefits: production throughput can increase up to 20 times, depending on the number of PCBs that are programmed and tested in parallel.
- Scenario: a flexible backplane programming and test system with anywhere from three to six boundary-scan/JTAG controllers (one per shelf in the test rack) can program and test 10-20 different designs simultaneously and asynchronously.
    - Benefits: allows for a very flexible system in a low-volume/high-mix manufacturing environment. The same station and operator can handle multiple different designs, reducing capital equipment expenditures as well as labor costs.
- Two or four PCBs are programmed and tested in parallel.
    - Benefits: It is fairly typical for the programming time to increase from one generation of a design to the next. This solution remains the same no matter the generation of the design and, even though programming times per PCB may increase, this type of boundary-scan station can actually increase production throughput by programming multiple PCBs in parallel instead of one at a time.

Conclusions

The benefits to the manufacturer of in-system programming are too great to ignore. As just-in-time production practices revolutionized manufacturing in general, so too just-in-time programming, made possible by in-system programming, has delivered tremendous returns to those who have adopted it. Now, new high-speed methods of programming flash memories after they have been soldered to a circuit board give manufacturers the ability to reduce the time it takes to program much larger memories with the larger quantities of data and code required by new-generation systems today. By accelerating flash programming tremendously, these advanced methods return ISP to its rightful place on the assembly line where manufacturers can reap the greatest benefits. In addition, several of these new programming and test methods are enabled in tools through a distributed processing architecture which makes parallel programming and test possible. As a consequence, multiple PCBs or UUTs based on different designs can be programmed and tested simultaneously, significantly enhancing the throughput of an assembly line even further and delivering product to the market much sooner.

Learn More

Learn more about how to perform at-speed in-system programming of SPI flash/EEPROM.



Register Today!

You can learn more about boundary-scan test/JTAG for your programming and test-related requirements by visiting our eResources.