Concurrent Testing with RF

Jeff Brenner
Verigy US

EK Tan
Verigy Singapore

Introduction

Integration of multiple functional cores can be accomplished through the development of either System on Chip (SOC) or System in Package (SIP) devices. A SOC device integrates the cores into a single integrated circuit. The SIP integration is the combination of multiple integrated circuits into a single package. An increase in the number of cores may lead to an increase in necessary tester resources and/or the time under test. This directly affects costs associated with testing these devices. A traditional approach to reducing the cost of test is to increase the parallelism by increasing the number of devices under test in a single insertion – parallel or multi-site testing. Concurrent test (CCT) methodology is defined by the parallel test execution of multiple cores at the die level and/or multiple die at the package level. The goal is to take maximum advantage of the parallel resources of the ATE system. When these are maximized the overall throughput of the ATE and test cell will be optimized leading to the maximum available throughput and thereby optimizing cost[1]*.* As represented in simplified form in Figure 1 - CCT Optimization, the example implementation of limited concurrency yielded an increase of resource utilization from 25% to 50% through the flow execution with the benefit of a 50% reduction in test time.

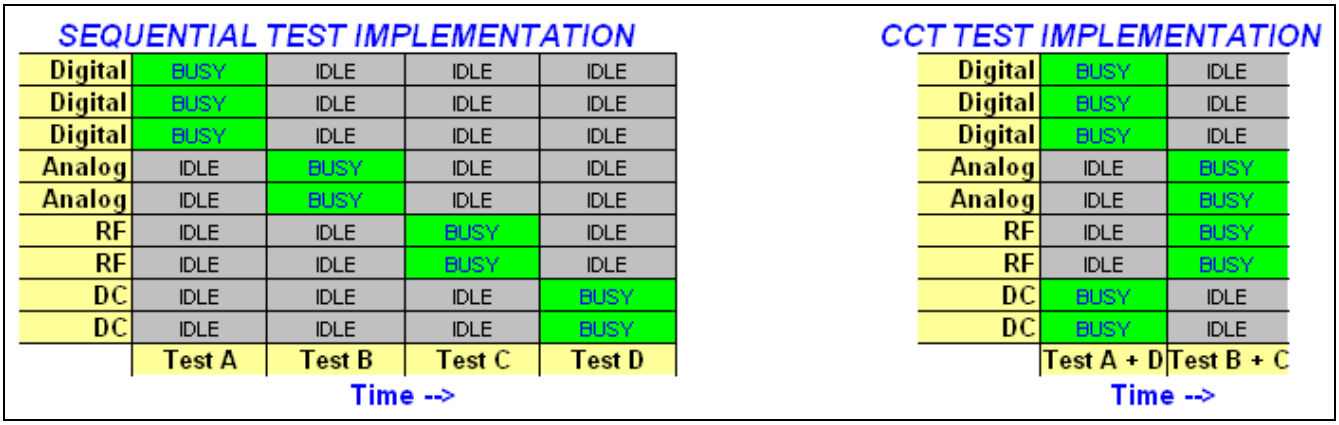

Figure 1 - CCT Optimization

Although CCT implementations are primarily motivated by cost savings through optimal parallelism of core tests there is an additional benefit to overall quality of test. As CCT motivates the implementation of the execution of multiple cores under test it simulates the operation of the device in its system execution mode. Therefore, faults which may cause test escapes during sequential core execution may be covered through parallel execution of multiple cores. For example, a wireless SOC device may incorporate multiple cores which share a common power supply such as a battery input. Testing each core sequentially may

[1] "E35: Cost of Ownership for Semiconductor Manufacturing Equipment Metrics," 1995 Book of SEMI Standards, April 1995, SEMI, Mt. View, CA.

not draw enough power to stress the power distribution of the SOC. Execution of multiple cores in parallel, as is done in CCT and in system execution mode, may cause enough stress on the power distribution to induce a voltage droop on one or more cores. This could manifest itself in many different manners such as Voltage reference offset, increased distortion or decreased F_{MAX}. Additionally care must be taken to identify fail mechanisms in concurrent execution that may not be encountered in system execution mode. For example, the implementation of digital SCAN test concurrently to analog distortion testing may induce clock artifacts that may not exist in system execution mode.[2] This could cause erroneous yield loss if not identified and addressed.

ATE Architecture for Concurrent Test

Mapping of ATE resources to the individual I/O requirements of the device under test is important to ensure optimal efficiency. The available resources are limited by the sequencer and port granularity of the tester architecture.

ATE sequencers are the hardware controls which execute the programmed sequence of events for tester resources. Traditionally perceived as a simple digital test resource, the sequencer executes microcode which outputs the digital test vectors in a given order, provides comparison to expected results and stores those results in memory. Additionally, the sequencer must provide other fundamental test capabilities including SCAN, Memory APG and Digital Capture. Often overlooked is the importance of sequencer control of Analog ATE resources. DC, Baseband Analog and RF testing requires local hardware sequencing of series of stimulus and captures to achieve optimal throughput. Offloading the sequencing to a central, shared resource is inefficient as this requires communication overhead between the central resource and instrument. The Verigy V93000 test system architecture provides per-pin sequencer control across all resource – digital and analog. The Pin Scale digital resources are based on the Verigy proprietary per-pin test processor. This test processor has been leveraged into the design of the DC VI and power supply resources, DC Scale. The DC Scale units share a common digital front end with the Pin Scale units but incorporate a custom analog front end for power supply or PMU operation. The digital front incorporates the same test processor for sequencer control as utilized in the Pin Scale. Additionally, the analog AWG/Digitizer (MCB) and RF Source/Measure (Port Scale RF) resources incorporate independent, per-unit sequencer control. This provides the ability of an individual unit to sequence through a complex series of operations through a single start event independently of other sequencers as demonstrated in Figure 2 - Sequencer Controlled RF Measurements.

[2] Fabien Perez, Markus Vogt. ‘Concurrent Test Implementation on a 2G/3G Baseband Device.’ Verigy, VOICE 2008.

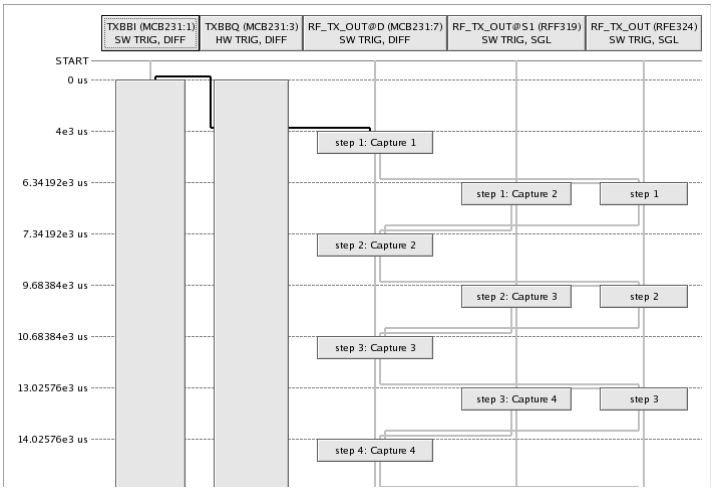

Figure 2 - Sequencer Controlled RF Measurements

A Port is a group of tester resources operating under a common sequencer control independent of other resources. Customarily this includes clocking which is segmented to an individual port. So, each port must be able to execute based on an independent clock. As complex SOC devices can incorporate multiple core clock sources it is important that the segmentation of the ATE resources provide the flexibility to match clock requirements to each port under test. As many SOC and SiP devices incorporate shared I/O for multiple cores the ATE system must be able to dynamically redefine Ports during execution of the test flow to match the core under test.

The granularity of the Port definition is to a single pin or analog unit on the Verigy V93000. Any pin can be allocated to a port and redefined to another port throughout the test flow execution. There are no limitations as to the sequencer operating modes of pins across ports. Any pin can be defined in one port to run in a specific operating mode, such as digital capture, without limiting the operating mode of any other pins. These definitions can be modified dynamically. Any pin can be defined to run at multiple of the master clock while any other pin is running at a differing multiple. As there are up to eight available master clocks in a V93000 the resources can be allocated to differing ports to provide up to eight concurrent, differing clock domains which are dynamically modifiable.

This hardware capability is critical to enable the flexibility to match ATE resources to the core under test when the device incorporates shared I/O resources. As indicated in Figure 3 - Dynamic ATE Resource Operating Mode, it is critical ensure the ATE hardware can modify it’s operating mode on a per-test basis throughout the flow execution. Devices generally have a more complex shared resource map which utilizes differing groups of I/O pins for each core test mode. Therefore, the ability to dynamically modify entire ports to device cores through the sequence of execution is critical.

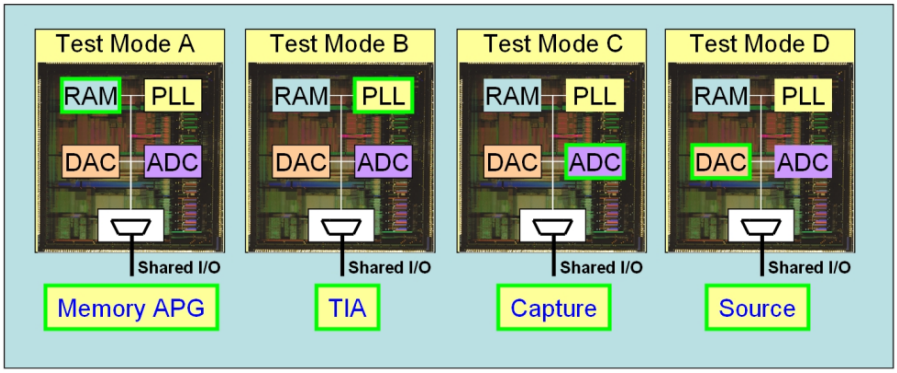

Figure 3 - Dynamic ATE Resource Operating Mode

In general, the determination of the theoretical efficiency of a concurrent test implementation can be determined by the degree to which the device cores follow the IACO (Independent, Accessible, Controllable, and Observable) guidelines.

- Independent – The ability of the cores under test to operate independent of each other. Cores may be designed to be interdependent. For example, many accelerator cores are designed as extensions of a processor cores.
- Accessible – The ability to access the cores under test independently of the others. Particularly in SOC designs external access to cores may be unavailable as the design is optimized to reduce pin count and cost. This may prevent direct access to the I/O of the core under test.
- Controllable – The ability to control the core under test independently of the others. This is demonstrated through the previous example of an accelerator core. Often programmability of accelerator cores are implemented as extensions of the processor instructions set. This would prevent control of the accelerator independent of the processor.
- Observable – The ability to observer the operations of the core under test independently of each other. Power supplies are excellent examples of design components that may affect the ability to observe the operation of the core. With shared power supplies it may not be possible to observe the voltage or noise at the core power. It may be necessary to observe the shared power supply in the hopes that it provides insight into the operation at the core under test.

Wireless devices demonstrate effectively how design decisions can affect the IACO rules. As indicated in Figure 4 - Design impact on CCT, a step in the early stages of integration may be to integrate the discrete function Integrated Circuits (IC) into a package (Multi-Chip Module/SIP) while preserving much of the I/O access to the discrete cores. This provides an excellent opportunity for a highly efficient CCT implementation.

At the highest level of integration, SOC, the core IACO rules may be compromised in order to optimize the design for lowest fabrication (die size) and package (pad/pin count) costs without consideration of the overall manufacturing costs. Consideration of the impact of CCT to test costs and coverage can help justify the implementation of Test Access Mechanisms (TAM) which can alleviate the restrictions imposed by reduced access integration. Both industry

standard, such as P1149.1[3], P1500[4], and custom design solutions[5] can readily address these issues.

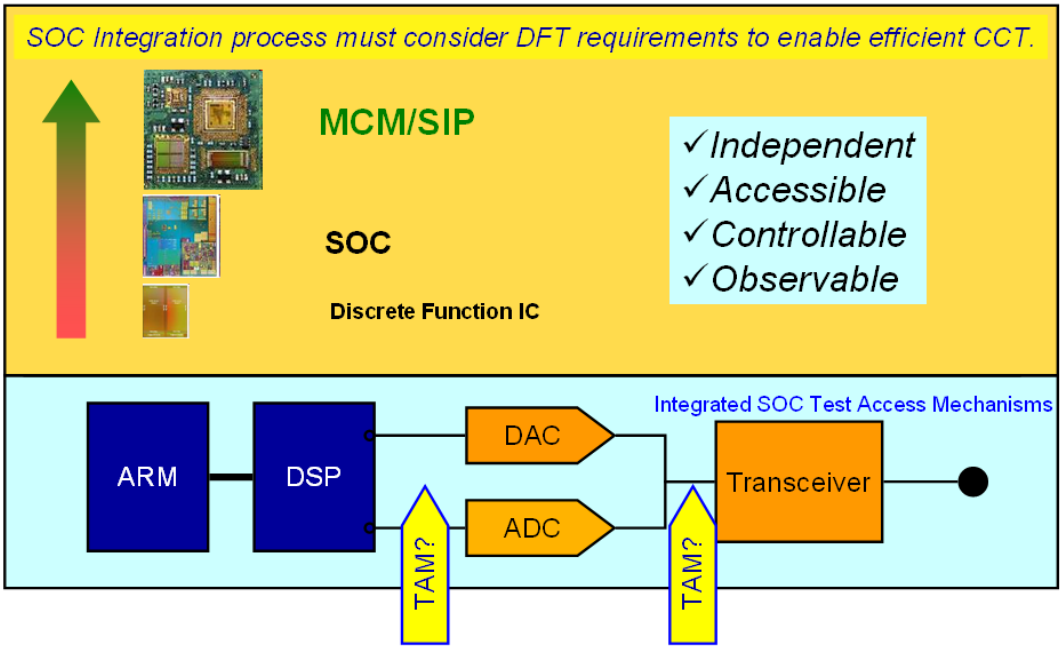

Figure 4 - Design impact on CCT

Guidelines

Although the types of I/O signals are different in across the core candidates of CCT, such as digital , mixed signal analog and RF, optimal ATE test control requires similar per-pin sequencer mechanisms. Design integration for these disparate cores follows similar IP core integration goals. Therefore, all cores are excellent candidates for concurrency. The planning and implementation of all types of cores follow similar steps throughput the process.

Implementation follows the following general steps.

1. Document IP capability – understand the concurrency capabilities of the device. This analysis should take place early in the design cycle to ensure optimization in the integration phase of the design.
2. Document tester resources – Although the ATE system has significant resource capabilities there may be restrictions of the available resources based on the manufacturing configuration. Understanding currently available resources helps define

[3] S.K. Sunter, 'Cost/benefit analysis of the P1194.1 mixed signal test bus.' IEEE Proc-Circuits Devices Syst., Vol 143, No. 6, Dec 1996.
[4] Benabdenbi, Moroufi and Marzouki. CAS –BUS: A Test Access Mechanism and a Toolbox Environment for Core-Based System Chip Testing. Journal of Electronic Testing, Volume 18, Numbers 4-5, August 2002.
[5] Brenner, "Practical design methodologies..," Semicon Europa, 2002.

an optimal concurrency schedule and enables the cost benefit analysis of modification of the configuration to take advantage of the device capabilities. Instrument upgrade and selection should be considered here as well.

3. Create a concurrency plan – based on the analysis of (1) and (2) and optimal plan can be developed to maximize utilization of the ATE resources throughout the test flow.
4. Implement – Develop the test program. The test program should follow structural guidelines which enable a simple transition from sequential flow to concurrent flow[6]. The engineers developing the individual core IP test items should adequately consider impact of their test components on other cores under test to optimize IACO compliance.

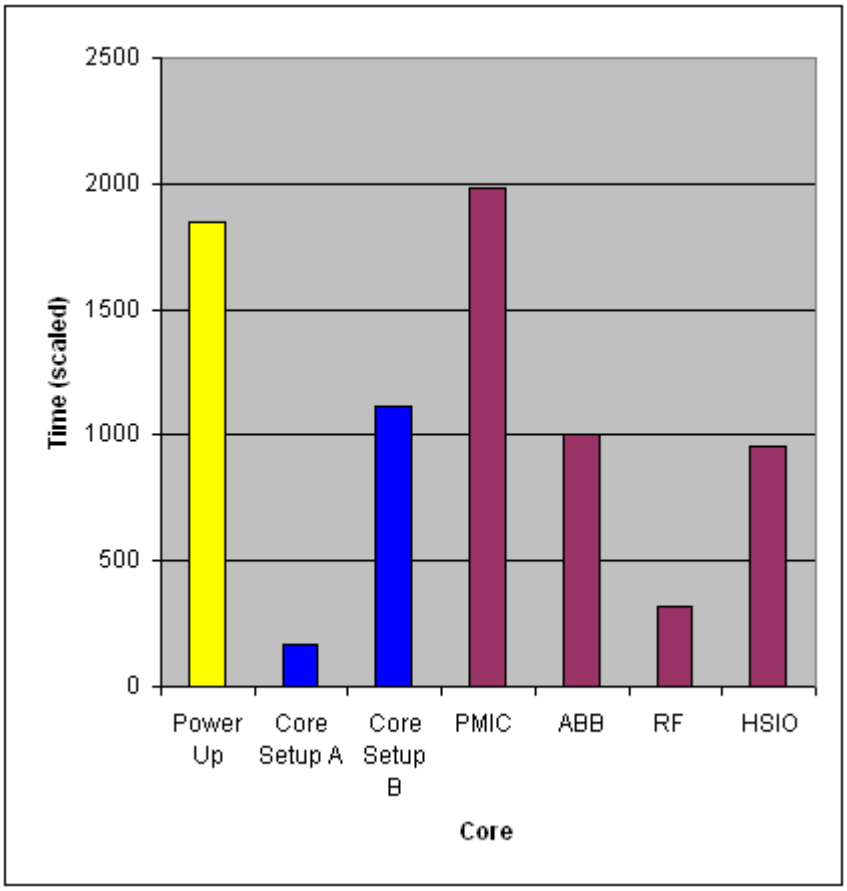

Figure 5 - Sequential Core Times

The concurrency plan should group resources based on device and resource parallelism. The plan requires the separation of core test execution from the test execution items which configure the cores into the proper operating mode for test (setup). As indicated below in Figure 5 - Sequential Core Times, the list of available core test items should be grouped to

[6] Verigy VOICE 2010. Workshop on Collaborative Development and IP Reuse. April 2010. Or, contact your local Verigy Application Engineer.

optimize available parallelism. Based on the device capability this would lead to a concurrency schedule as demonstrated in Figure 6 - Execution Flow.

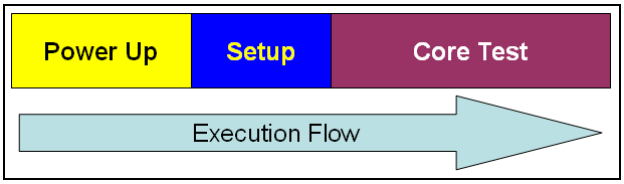

Figure 6 - Execution Flow

Case Study

The example device is an actual implementation of CCT on a 3G Cellular analog front end. It incorporates Power Management (PMIC), Analog Baseband (ABB), Transceiver (RF), and a high speed digital interface (HSIO). Multiple test engineers collaborated on the development of the test program

The implementation planning began with the concurrency planning. This planning split the test list into components based on the individual cores under test (PMIC, ABB, RF, Digital, HSIO). This core test list was further split to differentiate the core setup from the core test. The core setup consists of programming the core into its proper operating mode for test. It often requires utilization of a shared bus and may include settling times after setup. This is particularly true for setups which require the reprogramming of an internal clock to allow settling.

The efficiency of the concurrent implementation is broken into two parts. Each is primarily limited by the device design. The efficiency is calculated (Figure 7 - Concurrency Efficiency) by comparing the concurrent test time of all cores under test relative to the test time of longest single core execution block under test.

$$Efficiency \equiv \left[1 - \left(\frac{T_{CCT} - T_{SEQ\_MAX}}{T_{SEQ\_MAX}}\right)\right]$$

Figure 7 - Concurrency Efficiency

As indicated below in Figure 8 - Setup Concurrency, the device under test allows the implementation of setup of Core Group A in parallel to Core Group B. The concurrent efficiency gains of this component are 100% as the Core Group B execution is completely in parallel to the Core Group A setup execution. Note, this implementation represents a simple demonstration of the limitation of the test time reduction to the greatest single test instance execution time (GTI). As is demonstrated, concurrency provides the ability to parallelize core tests. But, the maximum gains cannot exceed the GTI, in this case the Core Group B Setup time which is 92% of the sequential execution time of the cores under test. In order to achieve additional gains other test time optimization techniques must be utilized such as multi-threading, hidden upload, etc.

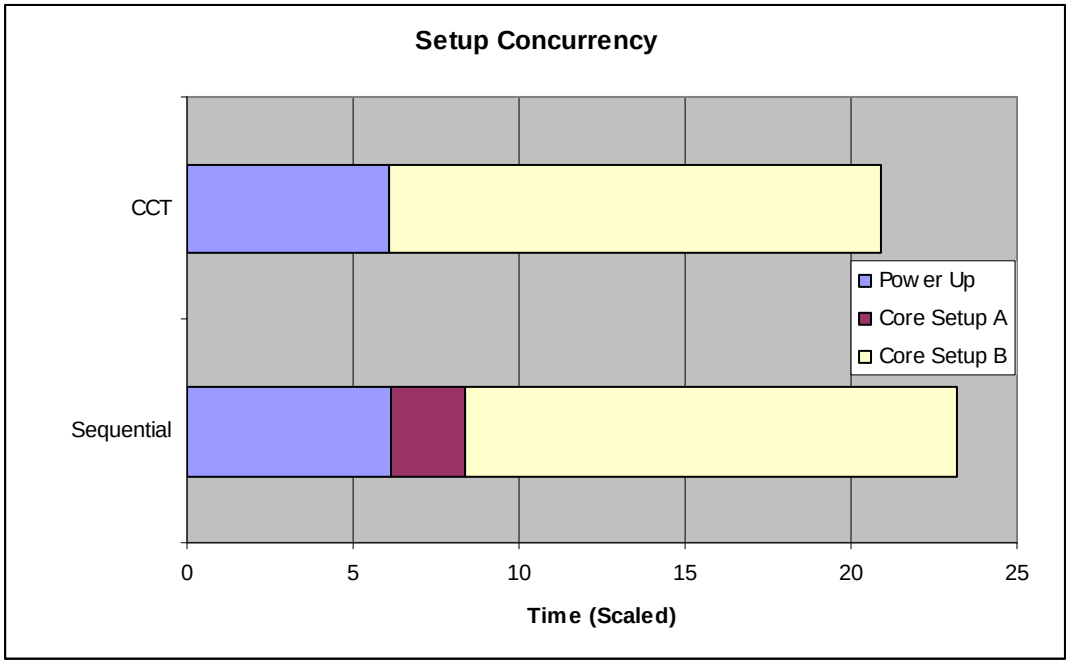

Figure 8 - Setup Concurrency

As demonstrated in Figure 9 - Core Test Concurrency, the impact of concurrency is significantly greater to overall test time when numerous tests are executed in parallel which have are closer in individual execution time. In this case, although the resulting CCT efficiency is approximately 92% the overall reduction of test time relative to the sequential implementation is 49%. As in the Setup example the limiter on the gains is the GTI execution time of the PMIC core. These gains were achieved with concurrency without compromising the test coverage of the sequential flow.

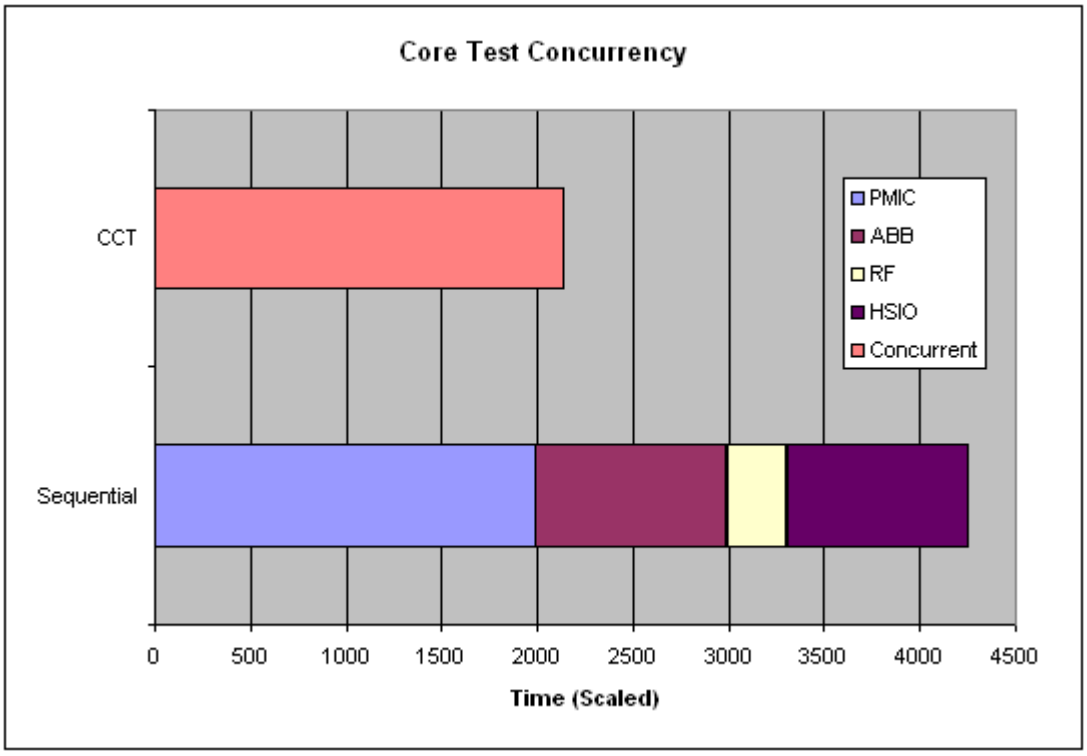

Figure 9 - Core Test Concurrency

Summary

As demonstrated in the example, significant throughput gains can be derived from the implementation of concurrent test on devices that include RF cores. This can be achieved with the benefit of additional test coverage by simulating system operating mode of the device. Effective planning starting during the design of the device can ensure optimal utilization of available tester resources.