



**HETEROGENEOUS  
INTEGRATION ROADMAP  
2019 Edition**

**Chapter 17: Test Technology**

**Section 10: DFT, Concurrent and SOC Testing**

<http://eps.ieee.org/hir>

The HIR is devised and intended for technology assessment only and is without regard to any commercial considerations pertaining to individual products or equipment.

We acknowledge with gratitude the use of material and figures in this Roadmap that are excerpted from original sources.  
Figures & tables should be re-used only with the permission of the original source.



### Section 10: DFT, Concurrent, and SOC Testing

A SOC design consists of multiple IP cores, each of which is an individual design block, and its design, its embedded test solution, and its interface to other IP cores are encapsulated in a design database. There are various types of IP cores (logic, memory, analog, high speed IO interfaces, RF, etc.) using different technologies. This assortment requires a diversity of solutions to test dies of the specific technologies corresponding to these embedded cores. Thus, SOC test implies a highly structured DFT infrastructure to observe and control individual core test solutions. SOC test must include the appropriate combination of these solutions associated with individual cores, core test access, and full-chip testing that includes targeting the interfaces between the cores and the top-level glue logic (i.e. a logic not placed within a core) in addition to what is within each core instance. Effective hierarchical or parallel approaches and scan pattern compression techniques will be required to evaluate and adjust the overall quality and cost of the SOC to an acceptable level for customers.

On the other hand, it is indispensable to improve the SOC test technology to handle a progression of design technologies accelerated by the evolving applications. The well-organized roadmap and the potential solutions that reflect these design intents should be reviewed by the readers. For example, low power design methodologies, which improve the chip performance, are widely adopted in various current SOCs. However, it is not easy to test the SOC without deeply understanding its functional behaviors and physical structures. As a result, the conventional DFT that focuses only on the static logic structure is not enough anymore, and the evolution to tackle this issue is strongly required. In product area requiring high reliability, in particular automotive devices, recently in-system self-test of digital circuits are required and logic BIST has come into use for this purpose. Furthermore, wide adoption of FinFET transistor technology possibly brings new and elusive defects on the silicon. Additional tests on logic and memory must be captured by introducing the corresponding test structure as extensions of existing DFT.

The quantitative trends and requirements of a consumer logic chip are shown in Logic section, compared with a MPU chip. Table 1 introduces the guideline for DFT design and the requirements for EDA tools.

Table 1: DFT Requirements

Manufacturable solutions exist, and are being optimized	
Manufacturable solutions are known	
Interim solutions are known	
Manufacturable solutions are NOT known	

Definitions for DFT Requirements Table 1 (next page):

- [1] STIL (Test Interface Language, IEEE1450.x) is an example. I/F should include not only test vectors, but also parametric factors.
- [2] A method to obtain overall test quality measure of SoC considering all cores; logic, memory and analog.
- [3] Growing number of row & column spares, and both divided and shared spares for segments in the future.
- [4] The current BISR for two dimensional repair is limited to a few row and column spares.
- [5] IEEE1500 and IEEE P1687 (IJTAG) are examples.
- [6] ATE software analyzes power and noise at testing and schedules concurrent test from IP/chip information.

Year of Production	2018	2019	2020	2021	2026	2031
<b>DFT Methodology for SOC</b>						
Hierarchical DFT Techniques This includes use of test compression within blocks/cores and top level tie-in of compression structures to chip scan interface resources. (FAL: Full Automation Limited use, GA: Generally Applied to any SOC)	GA	GA	GA	GA	GA	GA
3D Stacked Die DFT Techniques for chips with TSVs This includes DFT for wafer test for middle or top die in stack which may have only TSVs and micro-bumps to contact neighboring die in stack; it also includes DFT for stack testing. (PA: Partially automated, FAL: Full Automation with limited industry use, GA: Generally Applicable to most SoCs)	PA	PA	FAL	FAL	GA	GA
<b>Logic Core Integration</b>						
Supported Fault Models by ATPG for Overall Test (DBFM: Defect-based Fault, Model SDX: Extended Small Delay)	+DBFM	+DBFM	+SDX	+SDX	+SDX	+SDX
Standardization of DFT-ATE I/F [1] (LF: Limited Use of Full Information, F: General Use of Full Information)	LF	F	F	F	F	F
SoC Level Fault Coverage [2] (AH: Adhoc, L: Logic, M: Memory, IO: I/O, A: Analog)	L+M	+IO	+IO	+A	+A	+A
Inter-Core/Core-Interface Test (PA: Partially Automated ; FA: Fully Automated)	FA	FA	FA	FA	FA	FA
<b>Embedded cores: Memory</b>						
Repairing Mechanism of Memory Cells to improve Yield [3] (RC: BISR/BISD for a few Row & Col R/D, RCM: for more Row & Col R/D, M: for More Sophisticated R/D)	RCM	RCM	M	M	M	M
Area Investment of BIST/BISR/BISD [4] (Kgates/Mbits)	35	35	35	35	35	37
<b>AMS Core Integration</b>						
AMS BIST with digital interface; covers PLLs, High Speed SERDES, DA & AD, other AMS cores with BIST. Should include coverage estimate for Core. (PA: Partial Automation, FAL: Full Automation Limited use, GA: Generally Application to any SOC)	PA	PA	FAL	FAL	GA	GA
AMS non-BIST; covers any AMS cores that require functional tests using analog stimulus and/or response. Should include coverage estimate for Core. (PA: Partial Automation, FAL: Full Automation Limited use, GA: Generally Application to any SOC)	PA	PA	FAL	FAL	GA	GA
<b>DFT in Manufacturing</b>						
Systematic Hierarchical Diagnosis (L: Logic, M: Memory, I: Interface, A:Analog)	+I	+I	+A	+A	+A	+A
Supported Defect Type for Fault Diagnosis (C: Conventional (SAF, TF, BF), D: Delay Fault Model Considering Defective Delay Size, CT: Cross-talk, TRF: Transient Response Fault)	+CT	+CT	+CT	+TRF	+TRF	+TRF
Standardized Diagnosis Interface/Data in the diagnosis flow (ATE: Tester Log, DFT: DFT Method, PFA: Physical Failure Analysis)	+PFA	+PFA	+PFA	+PFA	+PFA	+PFA
Volume Diagnosis Database (SI: Collection and Storing Defect Information (B: Bad sample, G: Good sample), AD: Automated SoC Diagnosis)	+AD	+AD	+AD	+AD	+AD	+AD
<b>Concurrent Testing</b>						
Automated DFT environment for Concurrent Testing; integrates efficient interfaces for test of core itself and core test access [5]. (D: Digital , A: Analog)	D	D	D+A	D+A	D+A	D+A
ATE for Concurrent Testing There are some items to be carefully considered when Test scheduling [6] (R: pin Resource T: Test time P: Power consumption N: Noise)	R+T+P	+N	+N	+N	+N	+N
Standardized IP core access interface for Concurrent Testing (L: Logic , M : Memory , HV: high-voltage I : high-speed Interface A : Analog)	+HV	+HV	+A	+A	+I	+I
Test time reduction ratio by concurrent test (%) (L: Logic , M : Memory , HV: high-voltage I : high-speed Interface A : Analog)	95 (L+M)	90 (+HV)	75 (+A)	60 (+I)	60	60

### **Requirements for Logic Cores**

Sophisticated DFT methods such as random pattern logic BIST or compressed deterministic pattern test are required to reduce large amount of test data for logic cores. The adopted method should consider the pros and cons regarding DFT area investment, design rule restrictions, and associated ATE cost. DFT area mainly consists of the test controllers, compression logic, core wrappers and test points, which can be kept constant over time by using a hierarchical design approach.

Both SOC and MPU devices have an increasing amount of digital logic on the devices. Table 1 shows a common view of the DFT techniques which are expected to be used moving forward in an effort to cover the most likely faults (as modeled by the EDA systems) while attempting to keep the test costs low by effectively managing the test data volume.

There are four basic approaches in use for scan test generation:

- **Flat:** The EDA tools can consider the circuit in its entirety and generate what's called a "flat" test without leveraging the hierarchal design elements nor including pattern compression techniques. Virtually no one does this anymore, but it is useful for comparison purposes with the more appropriate approaches briefly described below.
- **Hierarchical:** The EDA tools can consider the hierarchal design elements to achieve an on-die parallel test setup. Parallel test would be applied to instances of wrapped cores to enable multiple instances to be tested in parallel.
- **Compression:** The EDA tools can imbed compression and decompression circuitry around the scan chains, allowing for many times more chains internally without increase of ATE scan pin resources, resulting in less data being required to be stored on the ATE for stimulus or output comparison purposes.
- **Hierarchical and Compression:** The EDA tools can implement a combination of 2 and 3 for a hierarchal compressed approach. This would involve cores being wrapped for isolation and including compression within the cores. Further compression can be obtained by testing multiple instances with the same set of scan-in pins, considered scan pin sharing, to allow testing of multiple instances of cores in parallel. The test data/scan outputs from each core instance can be observed independently or further compressed together and sent to a common set of chip scan-out pins, possibly resulting in more chip scan pin sharing.

The approach used to apply tests to embedded cores will have a large impact on test time and probably also test data volume. One traditional approach is to test a core in isolation and route its stimulus and expected responses up to the SOC pins to avoid running ATPG for the core at the SOC level. This saves CPU time for running ATPG, but fails to help reduce test time for the SOC. A more effective test compression approach is to test multiple cores in parallel and not put them into complete isolation from other cores. If compression can be used inside cores, it can also be used in the upper hierarchy of the cores to send the scan stimulus to multiple cores in parallel and to compact the output from several cores before sending it off-chip.

A tradeoff between test quality and test cost is a great concern. ATPG should support not only stuck-at and transition faults but also small delay, cell-aware and other defect-based faults to achieve a higher-level of test quality. Scan test pattern count will increase over the roadmap as logic transistor count increases. To avoid rising test cost, the test application time per gate should be reduced over the roadmap time period. Therefore, various approaches, such as test pattern reduction, scan chain length reduction and scalable speed-up of scan shift frequency, should be investigated. The acceleration of scan shift speed increases the power consumption during scan shift cycles and it might possibly make the test power problem more serious. Therefore, some DFT and ATPG approaches to solve the problem are required. Power consumption during the scan capture cycle is also an important issue and several approaches to relax this issue have been proposed. However, most of them cause an increase of test pattern counts and consequently make its impact on test application time intolerable. Some low capture power test approaches to minimize the increase of test pattern counts are also required. The impact on test data volume is shown with a 20% test data volume premium in the low-power rows. This will be too optimistic for cases where very low (e.g. less than 15%) switching is required since that could easily result in a doubling of the pattern count for the same coverage.

Another problem caused by the increase of test patterns is the test data volume. Even assuming tester memory size will be doubled every three years, high test data compression ratios will be required in the near future; therefore, test data reduction will remain a serious issue that must be tackled. One possible solution to reduce test application time and test data volume at a time is simultaneous test of repeatedly used IP cores in a design that can share a

common set of chip scan pins. By broadcasting the same scan-in stimulus to all such core instances, we reduce the bandwidth of data being sent from the ATE onto the chip and need less storage for that data on the ATE. Observing the outputs from each instance independently can aid in diagnosing failures, but by compressing the core instance outputs together and observing them at a common set of chip pins further increases the effectiveness of compression inside each core.

The increase of power domains may require some additional test patterns. Since the increase of test patterns will be linear to the number of power domains, it won't have severe impact on overall test pattern counts. Nevertheless, the increase of power domains or restrictions on test power possibly prevents maximum simultaneous test of identical IP cores. The impact of this effect should be investigated for future editions of the roadmap.

The issue of power consumption during test mentioned above is one cause for the increase of test patterns which will increase test data volume. Therefore, requirements on test data reduction also take account of this issue.

Year of Production	2016	2017	2018	2019	2020	2025	2030
<b>Worst Case (Flat) Data Volume (Gb)</b>							
MPU-HP - High Performance MPU (Server)	3673	4998	6802	9256	11366	31737	88623
MPU-CP - Consumer MPU (Laptop/Desktop)	2230	3035	4130	5620	6901	19272	53811
SOC-CP - Consumer SOC (Consumer SOC, APU, Mobile Processor)	1150	1565	2133	2907	3964	18662	85923
<b>Best-Case Test Data Volume (Hierarchal &amp; Compression) (Gb)</b>							
MPU-HP - High Performance MPU (Server)	7.5	8.9	10.3	12.0	12.6	15.1	16.7
MPU-CP - Consumer MPU (Laptop/Desktop)	6.2	7.2	8.5	9.8	10.2	12.1	13.1
SOC-CP - Consumer SOC (Consumer SOC, APU, Mobile Processor)	4.9	5.8	7.0	8.4	9.6	20.6	41.1
<b>Best-Case Compression Factor (Hierarchal &amp; Compression) (Gb)</b>							
MPU-HP - High Performance MPU (Server)	487	564	658	770	905	2106	5319
MPU-CP - Consumer MPU (Laptop/Desktop)	362	420	488	571	675	1595	4118
SOC-CP - Consumer SOC (Consumer SOC, APU, Mobile Processor)	237	269	303	347	414	905	2089

Figure 2: Scan Test Data Compression Factors (Flat with No Compression = 1)

Figure 2 shows the impact of hierarchical and compression scan test techniques on the test data increase. The current compression technologies utilize the fact that each test vector has many 'X-values' (don't care bits that don't contribute to the increase of test coverage), and factors of more than 100X compression are often achieved. However, even a 500X compression won't be enough for SoC (as shown in Table 3 in the Logic Device Testing section); therefore, more sophisticated technologies will be required in the future. Figure 2 shows the level of compression anticipated. The similarity of test vectors applied on scan chains will allow a chance of achieving higher compression ratios. The similarity of test vectors applied in time-space possibly also allows further compression. Thus, utilizing multi-dimensional similarity will be a potential solution.

As shown earlier (Table 2 in the Logic Device Testing section: Logic Test Data Volume), the external scan pins count for SOC-CP is comparatively small and it implies the necessity for scan input/output pins sharing among multiple embedded IP cores. More percentage of scan pins sharing increases the number of cores that can be tested in parallel, and it provides lower data volume and better compression factors.

In order to map this anticipated test data volume to tester and test time requirements, one must take into account the number of externally available scan chains and the data rate used to clock the test data into and out of the device. Estimations of these important parameters are shown in the SOC and MPU sections of the previous table (Table 2: Logic Test Data Volume). Since these parameters may vary on a part by part basis, the resulting data will need to be adjusted based on the approach taken on one part versus another:

- Designing more scan chains into a device results in more parallel test efficiency and a proportionally shorter test time and less memory per pin in the test system. This assumes the scan chain lengths are proportionately reduced.
- Clocking the scan chains at a faster speed also results in a shorter test time but doesn't reduce the pattern memory requirements of the ATE.

The other question when looking at the ATE memory requirements is which pattern compression technique is chosen for a given device. This question is impacted by many parameters including device size, personal preference and time to market constraints. As such, the analysis (in Table 2: Logic Test Data Volume) shows the minimum patterns per pin necessary to test the most complex devices. Thanks to the usage of more elaborate pattern generation techniques the data suggests that the minimum pattern requirement will only grow by 2X to 3X over the roadmap period.

The scan data shifting frequency impacts the test time necessary to drive and receive this large volume of test data. As cost effective testers with higher performance are deployed, apparently it seems that scan data shifting can be accelerated and then it can reduce the test time per device. The analysis calculates this impact and suggests that test times will be dropping over time due to this faster scan shifting. It should be noted that keeping the test application time per gate constant does not immediately mean a stable test application cost. Therefore, some approaches to reduce ATE cost, such as the increase of parallel sites, the use of low-cost ATE, or a speed up of test, are also required to establish the scalable reduction of test cost per transistor.

Concurrent parallel test in the core hierarchy is a potential solution of the test time reduction. ATPG/DFT level reduction technologies should be developed in the future. “Test per clock” means a test methodology that is quite different from the scan test (i.e. non-scan test). The test is done at each clock pulse and the scan shift operation is not required. There is some research regarding this methodology; however, more research will be required for industrial use.

High-level design languages are being used to improve design efficiency, and it is preferable for DFT to be applied at the high-level design phase. DFT design rule checking, testability analysis and fault coverage estimation are already available to some extent. Those features, including non-scan-design approaches and DFT synthesis in high-level design, are required in the next stage. Furthermore yield-loss is a great concern. As test patterns excite all possible faults on the DUT, it will lead to excessive transistor switching activity, which does not occur in normal functional operations. This will cause excessive power consumption which makes the functional operation unstable, and eventually makes the test fail, which will cause over-kill. In addition, signal integrity issues due to resistive drop or crosstalk can also occur which would make the functional operation unstable or marginal, and eventually cause failures. Therefore, predictability and control of power consumption and noise during DFT design is required. The leak current of test circuit itself should also be considered as a part of power consumption.

The discussion so far in this section has focused on the automatically generated scan-based testing requirements. Functional test techniques continue to be broadly deployed in order to enhance the scan-based testing techniques in an attempt to confirm the device’s suitability for the desired end-use application. Additionally, more and more memory arrays are getting embedded inside of both MPU and SOC devices.

### ***Requirements for Embedded Memory Cores***

As process technology advances, and due to some special application needs, both the number of memory instances and the total capacity of memory bits increase and will cause an increase in area investment for BIST, repair and diagnostic circuitry for memories. As the density and operating frequency of memory cores grow, memory DFT technologies as follows are implemented on SOCs and are factors of area investment increase:

- To cover new types of defects that appear in the advanced process technologies, dedicated optimal algorithms must be applied for a given memory design and defect set. In some cases, a highly programmable BIST that enables flexible composition of the testing algorithms is adopted.
- Practical embedded repair technologies, such as built-in redundancy allocation (BIRA) which analyzes the BIST results and allocates redundancy elements, and built-in self-repair (BISR) which performs the actual reconfiguration (hard-repairing) on-chip, are implemented for manufacturing yield improvement.
- On-line acquisition of failure information is essential for yield learning. A built-in self-diagnostic (BISD) technology distinguishes failure types such as bit, row, and column failures or combinations of them on-chip without dumping a large quantity of test results to ATE to utilize them for the yield learning. The testing algorithm programmability mentioned above has to be more sophisticated to contribute the diagnostics resolution enhancement. It must have a flexible capability to combine algorithm and test data/condition, and a memory diagnostic-only test pattern generation capability which is not used in the volume production testing.
- All the above features need to be implemented in a compact size, and operate at the system frequency.

The embedded memory test, repair and diagnostic logic size was estimated to be up to 35k gates per million bits in 2013. This contains BIST, BIRA, BISR, and BISD logic, but does not include the repair programming devices such as optical or electrical fuses. The ratio of area investment to the number of memory bits should not increase over the next decade. This requirement is not easily achievable. In particular, when the memory redundancy architecture becomes more complex, it will be difficult to implement the repair analysis with a small amount of logic. Therefore, a breakthrough in BIST, repair and diagnostic architecture is required. Dividing BIST, repair and

diagnostic logic of memory cores into a high-speed and a low-speed portion might reduce the area investment and turn-around-time for timing closure work. A high-speed portion that consists of counters and data comparators can be embedded in the memory cores, which will relax the restrictions for system speed operation in testing mode. A low-speed portion that consists of the logic for scheduling, pattern programming, etc. can be either designed to operate at low-speed or shared by multiple memory cores, which will reduce area investment and ease logical and physical design work. A lot of small-size memory cores are very often seen in modern SOCs; however, they require a larger amount of DFT gates than for a single memory core of the same total bit count. Therefore, consolidating memory cores into a smaller number of memory blocks can reduce memory DFT area investment dramatically. Testability-aware high-level synthesis should realize this feature in the memory cell allocation process and consider the parallelism of memory access on system operation.

### ***Requirements for Integration of SOC***

Reuse of IP cores is the key issue for design efficiency. When an IP core is obtained from a third-party provider, its predefined test solution must be adopted. Many EDA tools already leverage a standard format for logic cores (for example, IEEE Std 1500); and this format must be preserved and extended to other core types, such as analog cores. The DFT-ATE interface is going to be standardized (for example, IEEE Std 1450), and it should include not only test vectors but also parametric factors. An automatic design and test development environment is required to construct an SOC-level test logic structure and generate tester patterns from the test design information and test data of each IP core. This environment should realize concurrent testing described below.

Test quality of each core is now evaluated using various types of fault coverage such as stuck-at fault, transition-delay fault, or small-delay fault coverage. A unified method to obtain overall test quality that integrates the test coverage of each core should be developed. Conventionally, functional test has been used to compensate structural test's quality. However, automated test for inter-core or core interface should be developed in the near future. SOC-level diagnosis requires a systematic hierarchical diagnosis platform that is available for learning the limiting factors in a design or process (such as systemic defects). It should hierarchically locate the defective core, defective part in the core, and the defective X-Y coordinate in the part. A menu of supported defect types must be enhanced to meet with the growing population of physical defects in the latest process technology. Smooth standardized interfaces of design tools with ATE or failure analysis equipment are also required. Volume diagnosis is required to collect consistent data across multiple products containing the same design cores, which is stored in a data base and is analyzed statistically using data mining methods. The menu of data items is crucial for efficient yield learning, but it is a know-how issue now.

### ***Concurrent Testing***

For SOC test time reduction, concurrent testing, which performs tests of a number of IP (non-identical) cores concurrently, is a promising technology. For instance, the long test time of high-speed IO can be mitigated if other tests could be performed at the same time, which would decrease the total test time dramatically. To realize the concurrent testing concept, there are items that must be carefully considered in the product design process. These items include the number of test pins, power consumption during test, and restrictions of the test process. These items are classified as either DFT or ATE required features in Figures 3 and 4 below. IP cores should have a concurrent test capability that reduces the number of test pins (Reduced Pin Count Test: RPCT) without a test time increase, and a DFT methodology which enables concurrent testing for various types of cores. As these requirements differ corresponding to the core types on a chip, a standardized integration method of RF, MEMS and optical devices into a single SOC with conventional CMOS devices can be developed. It includes unification and standardization of test specifications which are used as interfaces by IP vendors, designers, DFT engineers and ATE engineers that can be combined with breakthroughs on analog-mixed signal/RF DFT methodologies (e.g. integrated efficient interfaces for test of the core itself and core test access, or wide adoption of IEEE Std 1500, and its extension to analog etc.)

DFT and ATE must cooperatively consider concurrent testing requirements and restrictions. This may not be an easy task as there are multiple challenges to enable concurrent testing. For instance, ATE software needs to be able to perform concurrent test scheduling after analyzing the power and noise expected during testing based upon design and test information specified for each IP core and chip architecture by the designer.

Figure 3: Required Concurrent Testing DFT Features

Features	Contents
External test pin sharing	Each JTAG-enabled IP core must use the 5 JTAG interfaces (TRST, TMS, TCK, TDI, TDO). Cores that have non-JTAG interfaces must be able to share external test pins with other cores.
Design for concurrent testing	The test structure of an IP core must be operationally independent from that of all other IP cores.
Identification of concurrent test restrictions	The presence of any test restrictions for each IP core must be identified to the scheduler (e.g. some IP cores are not testable at the same time due to noise, measurement precision, etc.).
Dynamic test configuration	Test structures/engines that can change the order of test and the combination of the simultaneous test to each IP core.
Test Data Volume	The test data volume of all IP cores must be able to be stored in the tester memory.
Test scheduling	Critical information on each IP must be available to the test scheduler: a) Test time of each IP core. b) Peak current and average power consumption for each IP core. c) Test Frequency for each IP core.
Common core interface	The test access interface of IP cores must be common among all IP cores (e.g. IJTAG)
Defective IP identification	There must be a mechanism to identify defective IP cores prior to and during test.

Figure 4: Required Concurrent Testing ATE Features

Features	Contents
Numerous Tester Channels with Frequency Flexibility	A large Number of Test channels that cover a wide range of frequencies will enable efficient concurrent testing. Test channels must provide test data such as clocks, resets, data, or control signals to many corresponding IP blocks. Testing can be more flexible if channel assignments are dynamically changeable.
Mixed Data type support	Capability of loading/unloading test data that is a mixed combination of digital, analog, and high-speed I/O data is required.
IP block measurement accuracy	Measuring accuracy of testing (e.g. high-speed I/O test) should be preserved in concurrent testing to match the specifications.
Test Data Handling Efficiency	Test data loadable to each divided test channel should closely match memory usage efficiency as that of non-concurrent test.
Power supply capability	A large number of capable power supply pins will enable large number of IP blocks to be simultaneously tested.
Multi-Site Testing capability	Capability to perform both multi-site testing and IP-level concurrent testing at a time will enable efficient testing.
Capable Software	Automated test scheduling software that can decide test scheduling configurations while considering many constraints is required.

Figure 5: Comparison between Multisite and Concurrent

Pin Count	Production Volume	Efficiency	
		Multisite Testing	Concurrent Testing
Many	Large	Medium	High
	Small	Low	High
Few	Large	High	Medium
	Small	Low	Low
Consideration		Cost of Jig (Initial Cost) - Probe Card, Test Board, etc. Cost of Tester - Pin Count, Power Supply, etc.	Reduction of Test Pins (RPCT) Cost of Chip - Impact on area, etc. Cost of Design

Multi-site testing is another approach to reduce effective test time per die or chip. The effect of cost reduction for each approach depends mainly on the number of test pins and the production volume, as shown in Figure 5 – Comparison between Multisite and Concurrent. Larger number of test pins will make the number of multi-sites

smaller, and higher production volume will get larger profit by the cost reduction. To estimate an accurate profit, cost of jigs and expense on engineering and designing should be also considered.

### ***DFT for Low-Power Design Components***

Low power consumption design is indispensable for battery-powered devices to enhance system performance and reliability. The design includes multiple power domains which are independently controlled by PMU (Power Management Unit), and some special cells used for controlling power at the physical level, such as level shifters, isolators, power switches, state retention registers, and low-power SRAMs.

However, the design raises new requirements in testing, which are some dedicated functions in test. For example, an isolator should be tested in both active and inactive mode to fully test its functionality, and a state retention cell requires a specific sequence of control signals to check if it satisfies a specification on power shut-off and turn-on. Please refer to the Figure 6 – Low-power Cell Test – for more low-power cells. Some of the defects on the special low-power cells can possibly be detected in an ordinary test flow but it is usually not enough to assure entire low power features of a design. These functions have not been treated in the historical scan test that only focuses on the structure of circuits. Therefore, a full support of these dedicated test functions for special low power cells is strongly required.

*Figure 6: Low-Power Cell Test*

#	Component	Test Contents
1	Isolator	Generate patterns controlling power-on/off of the power domain
2	Level Shifter	Include the cell faults in the ATPG fault list
3	Retention F/F	Generate patterns to confirm saved data after RESTORE operation
4	LP SRAM	Generate patterns which activate peripheral circuit inside the macro during the sleep mode and confirm cell data retention
5	Power Switch	Generate patterns to measure IDDQ with domains power on/off

### ***Summary***

SOC test difficulty will rise depending on the complexity and size of the chip. Adoption of new process technology or new devices such as MEMS, and high-quality test requirements for automotive or other application areas, also will introduces new test challenges which must be considered comprehensively.

For logic cores, since test pattern size will significantly increase for new types of defects and high-quality test requirements, hierarchical test logic structures and higher scan compression ratios will be essential. For memory cores, more sophisticated built-in features of test, repair and diagnostics are required. Introducing new types of embedded memory devices requires studies on necessary test sets. While test and DFT methodologies for logic and memory cores have been basically established, more studies are required for other types of cores such as analog, RF, etc.

Test cost reduction using concurrent testing necessitates standardized test structures and tool supports for them. Low power design trends introduce various design methodologies which make testing more complicated. Consistent automated design flow based on a standardized power format is required.