

DFT and BIST of a multichip module for high-energy physics experiments

*Original*

DFT and BIST of a multichip module for high-energy physics experiments / Benso, Alfredo; Chiusano, SILVIA ANNA; Prinetto, Paolo Ernesto. - In: IEEE DESIGN & TEST OF COMPUTERS. - ISSN 0740-7475. - STAMPA. - 19/3:(2002), pp. 92-103. [10.1109/MDT.2002.1003804]

*Availability:*

This version is available at: 11583/1404462 since:

*Publisher:*

IEEE

*Published*

DOI:10.1109/MDT.2002.1003804

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

# DFT and BIST of a Multichip Module for High-Energy Physics Experiments

Alfredo Benso, Silvia Chiusano, and Paolo Prinetto

Politecnico di Torino

Engineers at Politecnico di Torino designed a multichip module for high-energy physics experiments conducted on the Large Hadron Collider. An array of these MCMs handles multichannel data acquisition and signal processing. Testing the MCM from board to die level required a combination of DFT strategies.

■ **AT CERN** (the European Center for Nuclear Research in Geneva), a new generation of experiments in high-energy physics on the Large Hadron Collider<sup>1</sup> accelerator use a calorimetric readout system that detects collisions between high-energy particles.<sup>2</sup> To select and store a few interesting events out of the 800 million generated every second, the system uses an array of silicon-on-silicon multichip modules (MCMs) for multichannel data acquisition and signal processing. This article presents the test structures and strategies we adopted while designing this MCM and, in particular, one of its ASICs, which required online testing capabilities.

To reduce time to market and test costs, we made reusability a chief design consideration. In particular, we defined the test architectures and strategies for both horizontal and vertical reuse: The *horizontally reusable* strategies work

during different phases of the MCM life cycle, such as at bare die, assembly, end of production, in-field offline, and in-field online. The *vertically reusable* test strategies work at different levels of integration—at the die, MCM, and board levels, and so on.

Following the example of researchers who tested an MCM for space applications,<sup>3</sup> we adopted several solutions to obtain the best results in terms of coverage, time, and area overhead. We maximized the flexibility and reusability of the same test access mechanisms during different phases of the test and production cycle. We accomplished this by carefully planning the test control strategy from board to die level, using boundary scan logic extensively, and adopting an FPGA-based test processor. At-speed and online built-in self-test (BIST) solutions increased system reliability and serviceability during mission time.

## System description

We developed the MCM for the Large Hadron Collider in collaboration with Aurelia Microelettronica, an Italian design center, within the framework of the European Strategic Programme for Research and Development (Esprit) “Low-Cost Large Area Panel Processing of MCM-D Substrates and Packages” project. The foreseen substrate volume is about 35,000 square inches per year, beginning in 2000, with a potential volume three times larger.

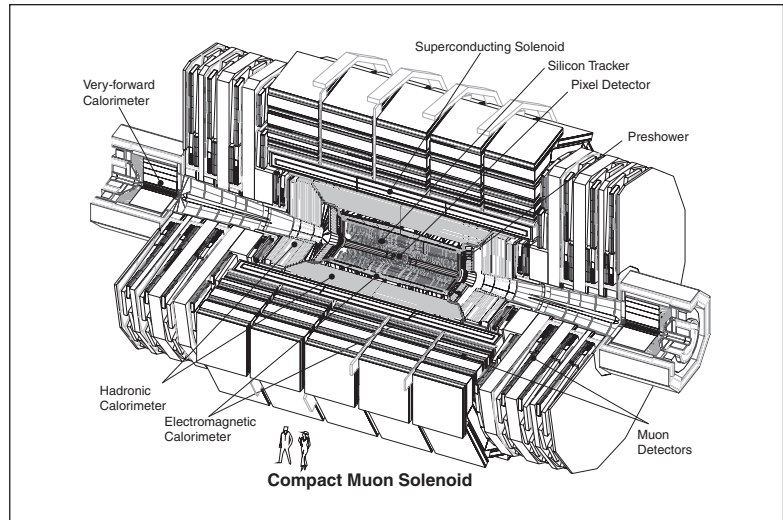
Figure 1 shows some of the detectors employed in the Large Hadron Collider for the Compact Muon Solenoid experiment (CMS).

The collision chamber, or tracker, is wrapped with different readout systems, which detect the energy produced by collisions between particles. Thousands of detectors connected through optical fibers to the readout system measure the energy.

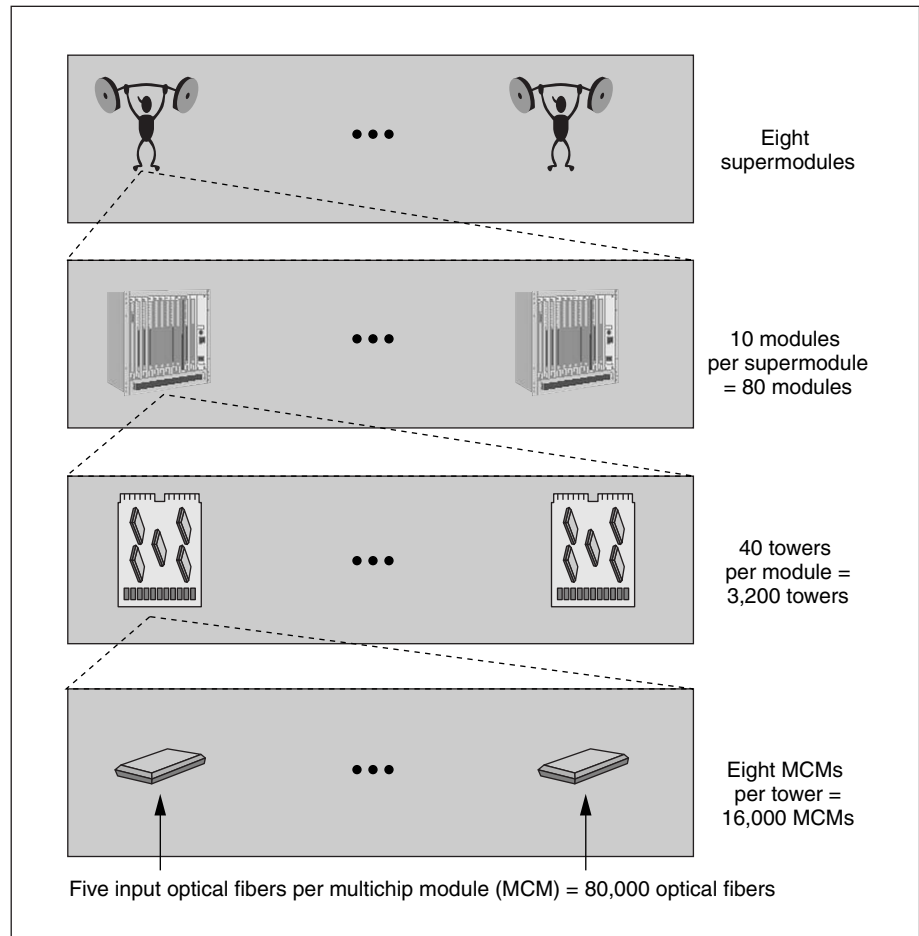
In the Electromagnetic Calorimeter (ECAL), the readout system must then select and store, for offline analysis, a few interesting events out of the 800 million that the detectors generate every second. To achieve a reduction rate of  $10^7$  to  $10^8$ , the system uses several *trigger systems*, organized in a hierarchy of trigger levels that analyze the read data with different precision and granularity to eliminate insignificant readouts. During the trigger phase, the readout system must temporarily store the detected data while it awaits a trigger decision. If the data is accepted, the system then transfers it to the next step in the processing chain.

The entire processing chain of a complex readout system consists of a hierarchy of computational units. Each top-level unit, called a supermodule, contains about ten modules. A module contains about a thousand channels, organized into towers of 25 channels each. The readout system implements a single tower as a printed circuit board containing four MCMs, each elaborating five data acquisition channels. Figure 2 shows a scheme for this readout hierarchy.

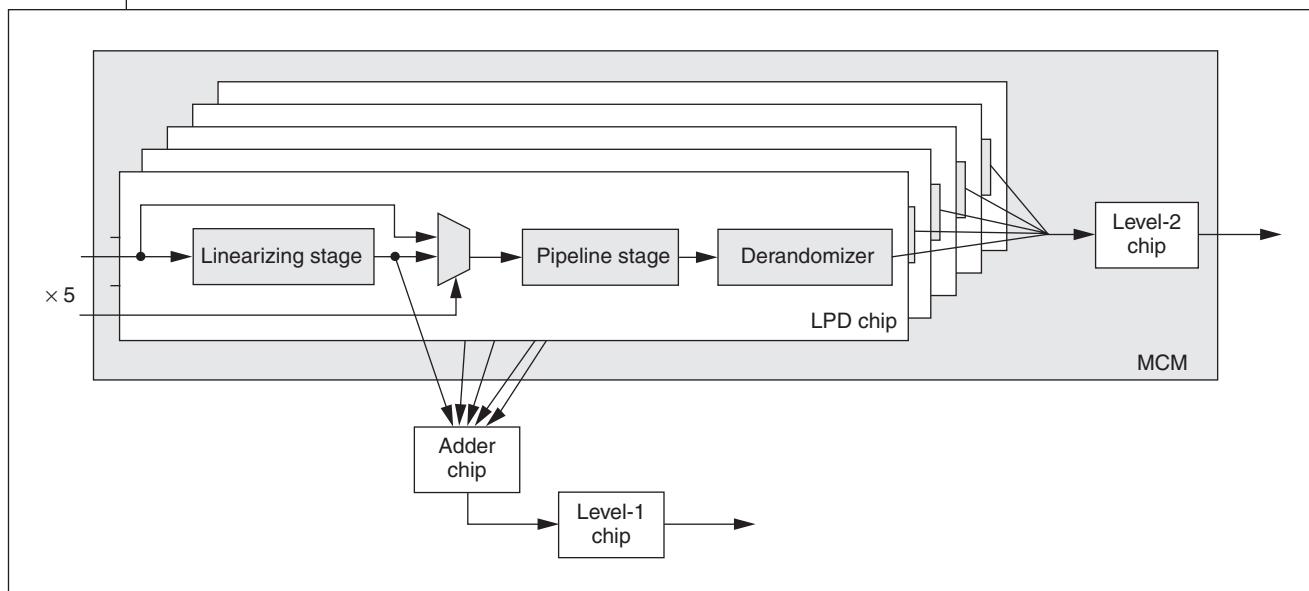
Each MCM is the building block of the data acquisition path's first hierarchical level. The input data to each MCM is converted into a digital format and compressed at a rate of 40 MHz. As Figure 3 (next page) shows, in the data acquisition path's first step, nonlinear data enters a linearizer built around an adder for offset correction and a multiplier for gain adjustment. The linearized data is then stored in a pipeline and concurrently sent to the trigger path, which analyzes it and classifies it as either interesting or insignificant. In the trigger path, data coming from



**Figure 1. Detectors in the Large Hadron Collider. The Compact Muon Solenoid tracker comprises about 250 square meters of silicon detectors—about the area of a 25-meter-long swimming pool. For each second that CMS runs, the system must record a volume of data equivalent to 10,000 *Encyclopædia Britannicas*.**



**Figure 2. The readout system processing chain.**



**Figure 3. MCM basic architecture. Each MCM contains five identical LPD chips, one for each data acquisition channel.**

the five acquisition channels goes to an adder circuit and then to a dual finite-impulse-response (FIR) filter—the level-1 chip in Figure 3—which extracts the energy information and formats the data according to the trigger system requirements. If the data qualifies as interesting, the MCM generates a first-level trigger event. This transfers the data stored in the pipeline to the derandomizer event buffer and then to another filter (the level-2 chip), containing three parallel FIR filters and a nonlinear-order statistic operator. These functional blocks occupy four different chips:

- the LPD chip (which includes the linearizer, pipeline, and derandomizer blocks),
- the adder chip,
- the level-1 chip, and
- the level-2 chip.

Our MCM contains five LPD chips and one level-2 chip. For flexibility and reusability, we implemented the level-1 and adder chips outside the MCM.

### Testing the MCM

The entire design of the MCM followed strict DFT rules.<sup>4,5</sup> Here, we present the test strategies we adopted, from both the board- and MCM-level perspectives.

As recommended in other research,<sup>6</sup> we organized the global in-field MCM test strategy into the following phases:

- *Identity check.* Exploiting the boundary-scan architecture, this test determines whether an incorrect die or MCM has been mounted on the substrate or on the board.
- *Interconnect test.* Using boundary-scan logic, this test checks the interconnections between the dies and the substrate and among the MCMs within the host board.
- *Functional and structural chip test.* Using proper test vectors generated by the BIST logic, this phase tests each die and MCM functionally, structurally, or both.
- *Performance test.* During the structural test, specific solutions determine whether the finished module meets its performance requirements.

To support all the phases of this planned test strategy, we inserted the following test architectures into the design of the MCM's chips:

- *Boundary scan* allows efficient interconnect tests among the MCM's chips. Along with the standard IEEE 1149.1 functionalities, the implemented test access port (TAP) controller provides full control of all the test architectures.

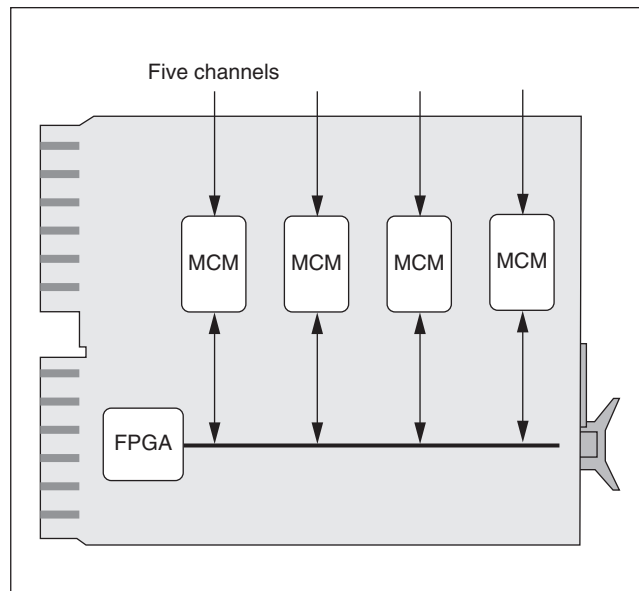
- *Memory BIST* detects address faults, stuck-at faults, transition faults, and linked idempotent coupling faults (CFids)<sup>5</sup> in the chip's memories.
- *Circular BIST* (CBIST) allows at-speed testing of the random logic.
- *Online data integrity checks* provide early detection of transient and permanent faults using Hamming and residue codes.

#### Board-level test strategy

We planned the overall board-level test strategy to minimize the number of test I/O pins and to reduce both the assembly costs and the bonding complexity. These goals led us to choose a *test access mechanism* based on boundary scan. At the MCM level, user-defined boundary-scan instructions implemented in the MCMs' TAP controller enable activation and collection of the different tests' results. At the board level, we enhanced the functionalities of an on-board FPGA to manage the board test by directly controlling each MCM's TAP, as Figure 4 shows.

The complete board-level test strategy consists of the following phases:

- *Identity check phase.* The FPGA uses the ID-code boundary-scan standard mode to check the correct placement of the components at both the board and MCM levels.
- *MCM-level interconnect test.* The FPGA controls the execution of the boundary-scan-based test of the interconnections among the MCMs' different chips.
- *Board-level interconnect test.* The FPGA exploits the dual-boundary-scan architecture<sup>7</sup> implemented in the MCM to test the interconnections among the different components on the board.
- *Die-level test.* The FPGA uses the TAP port to activate the BIST procedures of the different chips of the five MCMs and, after the required time, to collect the BIST results.
- *MCM-level structural test.* The FPGA activates the circular BIST structures to run an at-speed structural test of the MCMs.
- *Online test.* The FPGA continuously checks the status of the online BIST architectures. If it detects a failure, it enables the execution of proper diagnostic procedures.



**Figure 4. Basic tower architecture. Each board integrates five identical MCMs, for a total of 25 input data acquisition channels. The on-board FPGA is used to control the data acquisition process of each MCM.**

With four instances of the same MCM on each board and five identical chips in each MCM, the system is intrinsically redundant. We exploited this redundancy to optimize the test results collection phase, in which the FPGA can compare the results of the tests without storing reference signatures.

#### MCM-level test strategy

An advanced TAP controller handles activation and control of the MCMs' different test phases. In particular, the boundary-scan logic inserted in the MCM implements the *dual boundary scan* solution proposed by Jarawala.<sup>7</sup> In *normal mode*, the boundary-scan chain connects all the I/O pins of the chips within the MCM, allowing an MCM-level interconnect test. In *special mode*, the boundary-scan chain connects only the chip I/O pins that are also I/O pins for the MCM. In this way, during the board-level interconnect test, the MCM operates as a standard boundary-scan device. This solution lets us use the boundary-scan logic for both interconnect tests, thus reducing area overhead and test complexity. Moreover, when we set the TAP controller to special mode, an additional set of user-defined boundary-scan instructions

Table 1. TAP boundary-scan instructions available in special mode.

Instruction	Opcode	Selected boundary-scan data register
RUNBIST	010	BIST result register
DO_CBIST	100	Circular BIST chain*
DO_MUXSCAN	101	CBIST chain*
CHECKCBIST	110	CBIST result register

\* The CBIST chain is not connected between TDI and TDO (test data in and test data out) but has custom scan-in and scan-out connections.

become available for controlling the test of the MCM (see Table 1).

The special-mode boundary-scan instructions operate as follows:

- **RUNBIST** activates the memory BIST procedures and reads the test results from the MCM's BIST result register.
- **DO\_CBIST** initializes the CBIST chain when in the shift data register state (shift-DR), and the CBIST test mode when in the run-test/idle state. An efficient comparison of the chips' final CBIST signatures occurs automatically in the shift-DR state at the end of the test.
- **DO\_MUXSCAN** tests the circuit in full-scan mode when the TAP controller is in the run-test/idle state.
- **CHECK\_CBIST** reads the result of the signature-matching phase of the CBIST test. It also resets the LPD circuit before normal operations resume, because the CBIST test can leave the circuit in an unknown state.

To add flexibility to the MCM's test access mechanism, all the functionalities implemented by the user-defined boundary-scan instructions can also be controlled via a serial interface already present in the MCM for configuration purposes.

#### LPD test architecture

The LPD chip is one of the most important units in the readout system's first hierarchical level; we implemented several test architectures in this chip to perform functional, structural, and performance tests.

**Functional chip test.** From the perspective of testability, memories are key components because they play a crucial role in terms of availability and serviceability. In an environment exposed to high radiation, they are also very sensitive to *transient faults* such as single-event upsets (SEUs), which can cause a bit flip in one of the system's memory elements.

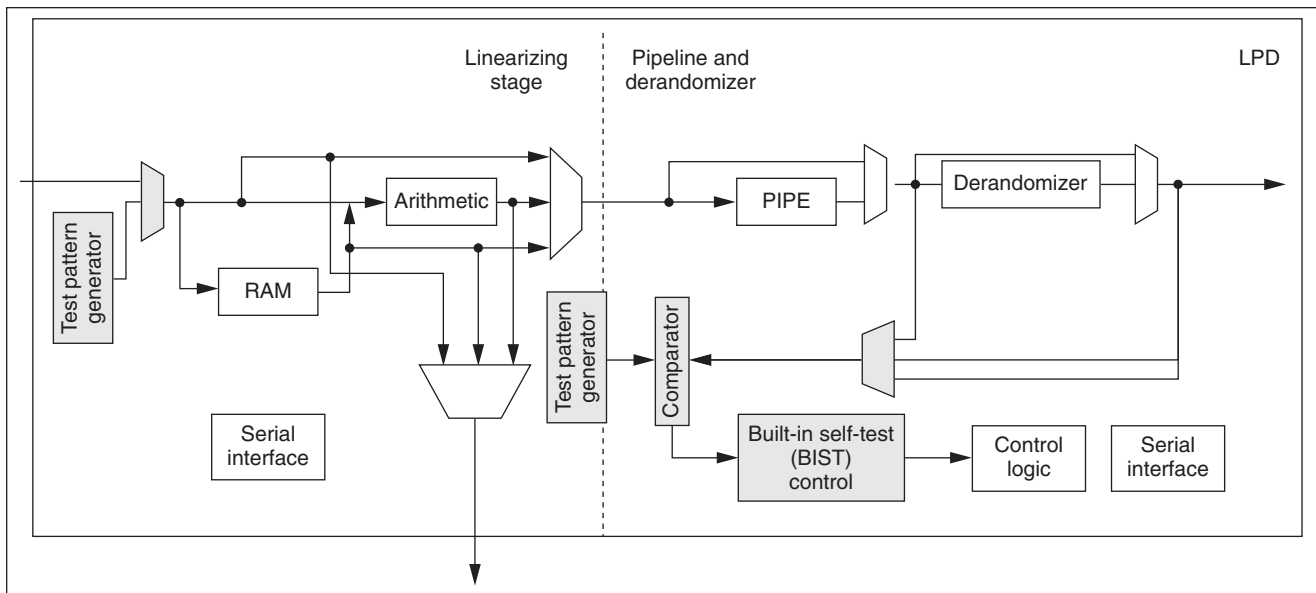
Therefore, in the LPD chip, we implemented two different, but complementary, BIST approaches to cover both permanent and transient faults. The blocks that we added to the original architecture fall into two groups:

- *Offline memory BIST blocks* execute the offline test of the memories to detect the most common permanent faults, such as stuck-at faults.
- *Online BIST blocks* operate online to check the correct behavior of the memories and some other functional units, mainly to detect transient data faults.

The memories used in the LPD chip design—in the linearizer, the pipeline, and the derandomizer—are implemented as eight identical dual-port 256×32-bit memory modules, realized in 0.6-micron AMS technology, and occupying an area of about 1.8 mm<sup>2</sup> each.

For the offline memory BIST, the architecture implemented in the LPD chip exploits the configurable path architecture—the linearizer, pipeline, and derandomizer blocks can be bypassed—to minimize the number of necessary BIST blocks. As shown in Figure 5, where the shaded boxes represent the test logic, the chip needs two test pattern generators (TPGs) to test the eight memory modules. One TPG always generates the patterns written in the memories; the other generates the patterns to be compared with the memory content during the test. Using only one TPG for both pattern generation and comparison would not cover faults occurring in the TPG itself. Two serial interfaces serve to configure the LPD behavior and functional paths.

The test algorithms implemented by the BIST controller usually depend on the different memories' type and functionality. In our case, the memories consist of identical dual-port memory

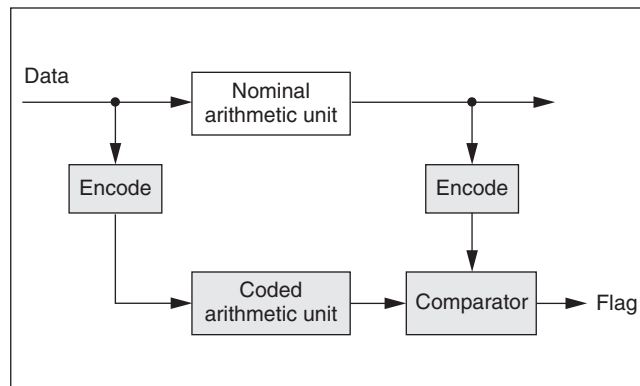


**Figure 5. LPD offline memory BIST blocks. (Shaded boxes represent the test logic.)**

modules with two unused ports. An additional simplification is that these memories are *single-ordered addressed* (SOA). We could therefore implement a single BIST controller to execute the same algorithm for all the memories. The algorithm we chose was MARCH B,<sup>8</sup> which covers all SOA address faults, stuck-at faults, transition faults, coupling faults, and linked CFids (except  $\langle \uparrow; 0/1 \rangle$  linked to  $\langle \downarrow; 1/0 \rangle$ ).

Using a specific register of the serial interface, we can select the subset of memories to be tested. Testing a subset of memories lets us test only the parts of the chip that are actually used. For example, the MCM could be integrated into a readout system requiring only the circuit's linearizer; this would make it possible to test only four RAMs, reducing test time and increasing production yield. The memories under test are always tested in parallel.

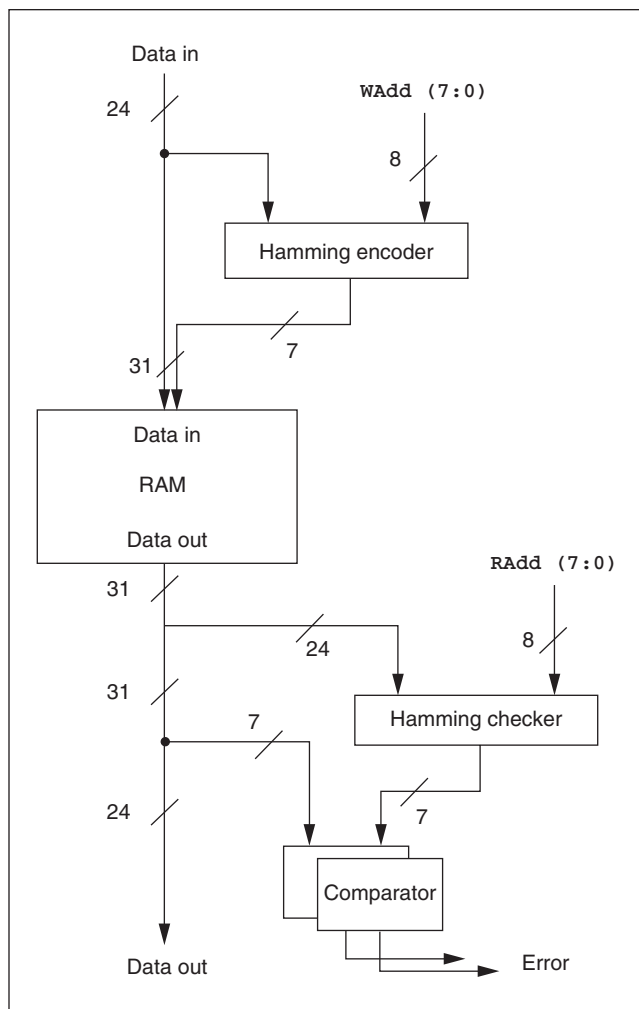
The online BIST blocks can use spatial, temporal, or information redundancy to effectively detect and correct transient faults. Information redundancy—adding redundant information to the original data through a code-based approach—is usually the best solution for memory and arithmetic components. Test engineers must strike a balance between information overhead (for example, a single-parity-bit, Hamming, or cyclic code) and the approach's detection and correction capabilities.



**Figure 6. LPD three-code online checking. Arithmetic operations are always executed on coded data to avoid computations on faulty data. (Shaded boxes represent the test mechanism.)**

In the LPD chip, we use an arithmetic residue code to check the operation results online, as Figure 6 shows. An encoder encodes data before each arithmetic operation. The computation is then executed on both the original and the encoded data. Finally, a comparison block checks the two results to verify the operation's correctness. Using an arithmetic code results in less area overhead than a simple duplication of the arithmetic units.<sup>9</sup>

To increase the coverage and the three memories' robustness, we implemented an online memory BIST architecture based on informa-



**Figure 7. Online memory BIST logic is mandatory in physics experiments where high-radiation environments can cause transient faults in the memory cells of the digital components.**

tion redundancy. As Figure 7 shows, rather than the original data, a code word is written into the memory. Therefore, before each write operation, the Hamming encoder encodes the data and the write address  $wAdd$  and writes the resulting code word into the memory. For each read operation, the Hamming checker encodes the original data and the read address  $rAdd$ , and the resulting code bits are compared with those stored during the write operation. If the two sets of code bits match, the original data qualifies as valid. The validation operation can fail only if either the number of faults appearing in the addressed word is higher than the number of faults that the code can detect or the com-

parator is faulty. For this reason, we duplicated the comparator. The error bit provided by the Hamming checker is then stored in a status register and also sent to an OR gate with the data word's *integrity bit*, which identifies, at every instant, the validity of a data word traveling within the MCM. Including the write or read address in the encoding and decoding operations lets us detect address decoder faults as well.

Our online memory BIST solution lets us detect

- single and double transient bit flips (SEUs) on every data cell,
- seven-bit burst errors,
- single and multiple stuck-at faults,
- coupling faults between cells or bits of the same cell (covered when modification in the coupled cell produces an error detectable by the code), and
- address decoder faults.

**Structural and performance tests.** The solutions we've presented so far do not cover all the mission logic of the original design—the serial interface, the control logic, and the arithmetic block. Moreover, we need an at-speed test to cover performance faults. To solve the problem, we implemented a solution based on the circular BIST technique.<sup>10,11</sup> We use two additional control signals added to the LPD chip to control each CBIST flip-flop's four possible modes of operation; the TAP controller set in special mode can also control them directly.

The main advantage of CBIST over many other BIST techniques is that it is a real at-speed test that requires a short test time and detects delay faults. This feature is also very useful for detecting timing violations during prelayout and postlayout simulations. In fact, timing violations during CBIST simulation generate unknown values in the scan chain, which propagate throughout the circuit in a few clock cycles and are therefore easily recognizable.

After a preliminary initialization phase for all the flip-flops in the chain, we start the BIST phase by configuring the flip-flops in CBIST mode. This configuration transforms the whole circuit into a large BIST structure, in which the flip-flop chains act both as pattern generators



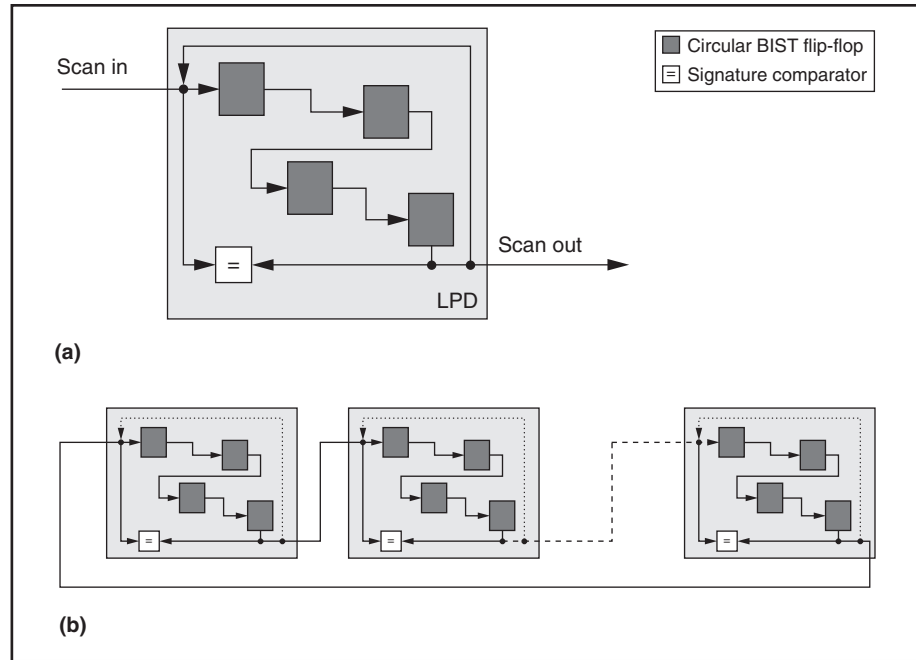
and output compactors. This structure performs the actual test by applying a given number of clock cycles, scanning out the content of the CBIST chain, and comparing it with a reference signature. For very large circuits, this simple technique provides very high fault coverage.<sup>11</sup>

For the MCM level, when running the test on all five LPD chips, the scanning phase following the actual test is optimized to exploit the MCM design's redundancy, as Figure 8 shows. In particular, rather than comparing the scanned data with a reference signature, we adopted a daisy-chain solution, in which each LPD chip compares its signature with that of the preceding one. Therefore, to execute the scanning phase of the CBIST signature concurrently on the five LPD chips, we do not need to store a reference signature. Moreover, if we run the CBIST-based test concurrently on all five of the board's MCMs, we can apply the same strategy at the board level, executing a daisy-chain comparison of each MCM's scan-out signal with the same signal of the previous MCM. This solution garnered considerable savings in terms of test time, area, and routing overhead.

The CBIST chain, automatically inserted using a commercial tool, contains about 3,500 flip-flops. The chain does not include flip-flops belonging to the following parts:

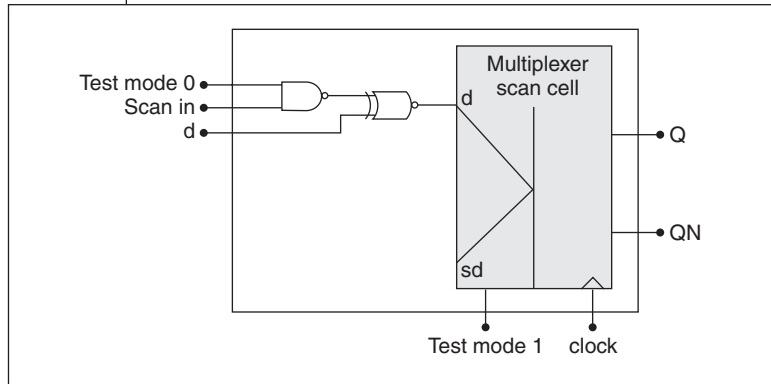
- the TAP controller, because it must control the CBIST test;
- the boundaries of the memory modules, to avoid unpredictable behavior of the memory modules arising from pseudorandom patterns generated during the test; and
- the tristate outputs, to avoid conflicts on the buses caused by the pseudorandom patterns.

Inserting the CBIST circuitry in such a complex circuit required careful attention to many details. The following were the most challenging problems that arose during this operation:



**Figure 8. CBIST signature checking within the LPD chips. The intrinsic redundancy of the MCM allows comparison between the CBIST scan chains of each LPD without the need of storing a reference signature on chip. The five LPDs are therefore connected in a daisy chain, so that each LPD can compare its scan chain with that of the LPD preceding it in the daisy chain.**

- *Area overhead.* The CBIST cell that we used in the LPD circuit (see Figure 9 and Table 2, next page) does not implement parallel loading of the initial state. Instead, it uses a serial shift of the initial state to reduce the area overhead and to increase the flexibility in choosing the initial state. We implemented the CBIST cells using standard cells because the effort required to design and test custom cells would be justified only by a wide use of the CBIST method with the same technology. The standard-cell approach requires placing the CBIST cells belonging to the same flip-flop close to one another to reduce the delay introduced by the scan chain insertion. Finally, the CBIST cells can operate as standard scan flip-flops, letting us test and detect random pattern-resistant faults by applying deterministic patterns.
- *Asynchronous resets.* Using flip-flops with asynchronous reset requires special care when performing scan chain shifting or CBIST test to avoid accidental resets. Therefore, to test the



**Figure 9. CBIST cell. A CBIST cell is similar to a multiplexer-scan cell, but with one additional operating mode that lets it generate pseudorandom test patterns. In the figure, *d* is the input data line, *Q* the output data line, and *QN* the opposite value of *Q*.**

Test mode 1	Test mode 0	Behavior
0	0	Normal
0	1	Run CBIST
1	0	Scan in/out
1	1	Scan in/out

reset signals' correct behavior (which is not testable during the CBIST phase), we apply an asynchronous reset to all the flip-flops after loading the scan chain with '1'. In this way, if the reset signal were stuck at an inactive state, the CBIST would start with a different initial state, and the test would detect the error when it compared the final signatures.

- **Embedded cores.** Inserting the RAM cores in the LPD chip required special care in the design phase, because pseudorandom vectors can bring these blocks to an undefined state. For this reason, during the CBIST test we keep the memories completely isolated from the rest of the circuit.
- **I/O isolation.** Because the CBIST pseudorandom sequence depends on the initial state and on the input signals' values, we use the boundary-scan logic during the CBIST test to force the input signals to known values. Moreover, we used the same solution on the outputs to prevent tristate conflicts during the pseudorandom sequence.
- **Asynchronous logic.** During the CBIST test,

the asynchronous logic is isolated from the digital part of the circuit.

- **Timing constraints.** Inserting CBIST logic on flip-flop inputs can increase a path's delays. In our case, the resulting delay was quite small—1 ns in the worst case, using AMS CMOS 0.6-micron TLM technology—and did not require special attention except on the critical paths.
- **Multiple clock domains.** Using multiple clocks (and multiple edges) requires special rules to create the correct shift operation in the scan chain. This problem can be particularly evident in a CBIST architecture implementation. Possible solutions include joining different clocks or exploiting multiple scan chains. The approach we took in the LPD circuit was to join, in test mode, all the chip's clocks and to run the test using the highest allowed clock speed.
- **Synthesis.** Because of all the timing issues we've mentioned, we must perform synthesis using constraints that match the at-speed test behavior. Therefore, we applied the tightest performance constraints even to the slower clock domains.

CBIST insertion required extra design time for synthesis and layout to meet both the original and the additional test timing constraints. However, this ensured early and full debugging of timing violations.

## Experimental results

Now we present some experimental results that we obtained from the LPD synthesis, simulation, and test generation.

### Test area overheads

We implemented the LPD chip in AMS technology (CMOS 0.6-micron); it has 144 I/O signals, and its area is 35 mm<sup>2</sup>. The total number of gates is approximately 104,000 (33% for the linearizer and 34% each for the pipeline and the derandomizer), of which 40,000 are for random logic and the remaining 64,000 are for RAMs. The test hardware area overhead, less than 17% of the global chip area, comes mainly from using standard cells to realize the CBIST flip-flop. Of the random logic, we inserted 95% in the CBIST chain (as described earlier, we

excluded only the TAP controller), for a total of 3,500 flip-flops. Obviously, a custom realization of the CBIST cells would dramatically reduce the overhead. Table 3 reports the area overhead of the test hardware as a percentage of the total area (including I/O pads).

We have observed no relevant performance degradation, and the system meets the target maximum working frequency of 50 MHz.

#### Test time and fault coverage

To provide comprehensive fault coverage and test time results, we ran a fault simulation of the chip, including, in sequence, the following steps:

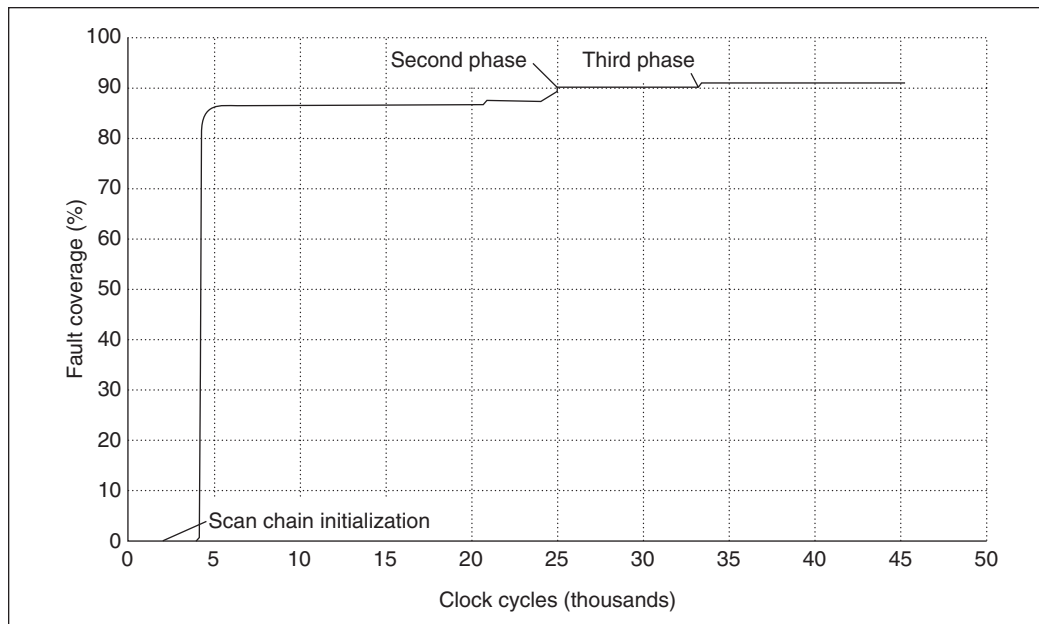
- *CBIST test.* This test included three phases: the CBIST chain initialized with all 0s, the CBIST chain initialized with all 1s, and the CBIST test executed with synchronous resets enabled to detect faults on the reset signal.
- *RAM memory BIST.* We used the TAP controller to activate the memory test.
- *Functional test.* We applied additional specific functional tests to improve the coverage on the paths that include RAMs and were not fully tested by the memory BIST.
- *Multiplexer-scan test.* This test applied standard multiplexer-scan vectors to detect random pattern-resistant faults.

Table 4 reports the test times in terms of clock cycles required to execute all these tests at 50 MHz.

Figure 10, a graphical representation of the CBIST fault coverage as a function of the number of applied clock cycles, shows how all three phases increase the global fault coverage. After 45,000 patterns, the increase in fault cov-

Test structure	Area overhead (% of total area)
CBIST cells	12.42
TAP controller	0.40
BIST controller	0.97
Test pattern generators	1.68
Hamming encoders and checkers	0.96
Arithmetic coding blocks	0.55
Total	16.98

Test	Main clock cycles	TAP clock cycles
CBIST	45,000	500
BIST	164,000	100
Check MCM connections	0	10,000
Total	209,000	10,600



**Figure 10. CBIST test fault coverage for increasing test lengths. This function shows how the reinitializations of the CBIST chains increased the final fault coverage result.**

Table 5. CBIST fault coverage, including FC (the cumulative fault coverage) and UDF (the undetected faults in the specified module as a fraction of the total number of undetected faults).

Module	CBIST (45,000 clock cycles)		RAM BIST (164,000 clock cycles)		Functional test (15,000 clock cycles)		Multiplexer- scan (1,000,000 clock cycles)	
	FC (%)	UDF (%)	FC (%)	UDF (%)	FC (%)	UDF (%)	FC (%)	UDF (%)
LPD core	92.06	100	94.68	100	95.57	100	99.39	100
Pipeline/derandomizer	94.30	11.50	95.43	13.77	96.22	13.66	99.77	6.00
Serial interface	98.24	9.40	98.34	13.22	98.60	13.43	99.93	4.97
Linearizer	88.08	38.38	88.76	54.01	91.23	50.62	98.81	49.96
BIST controller	47.15	21.41	86.85	7.95	86.85	9.54	99.42	3.08
Others	87.97	19.29	95.37	11.04	95.60	12.72	95.71	35.97

erage proved marginal, so we considered the test concluded at that point.

Table 5 reports the fault coverage of the LPD chip's different blocks. Fault coverage is cumulative from left to right; the coverage reported in each FC column includes faults detected in the previous test. The UDF column represents the number of undetected faults in the specified module as a fraction of the total number of undetected faults. A higher UDF indicates that module's greater criticality in lowering the overall coverage; this parameter is very useful in determining which modules require closer attention.

A detailed hierarchical analysis of the fault simulation results showed that the low coverage of some modules was due to the embedded RAMs, which decrease the observability of the surrounding logic. Placing observation flip-flops near embedded RAMs could increase coverage but would also introduce critical paths that don't add to the circuit's functionality. We therefore preferred a functional test approach to increase the coverage. As reported, BIST and functional testing significantly increase the fault coverage, leaving only spare faults that require too much design effort to be targeted with functional tests. Automatic test-pattern generation with multiplexer-scan vectors provided an easy solution to this problem, yielding a very high final coverage. Nevertheless, because there are so many flip-flops, multiplexer-scan requires the application and storage of very long vector sequences. Therefore, it is useful only for end-of-production testing by automatic test equipment.

The high undetected fault coverage per-

centage that Table 5 reports for "others" (other modules) depends on some asynchronous units that cannot be fully fault simulated. Real fault coverage would be higher.

**IN DEVELOPING THIS** multichannel data-acquisition and signal-processing MCM, we adopted different test strategies for different hierarchical levels. In particular, we maximized the flexibility and reuse of the test logic during different test phases of the production cycle. We accomplished this by carefully planning the test control strategy from the board level to the die level, using boundary-scan logic extensively, and by adopting an FPGA-based test processor. At-speed BIST solutions gave the best results in terms of structural and performance fault coverage. Eventually, we implemented widespread concurrent online BIST to increase system reliability and serviceability during mission time. This work shows how to achieve excellent results in terms of fault coverage and dependability. In complex digital systems, it is mandatory to plan the overall test strategy taking vertical and horizontal reuse into account. The best solution is usually a coordinated mix of DFT approaches, each addressing a particular testability issue. ■

## Acknowledgments

Partial support for this work came from Esprit 26261 "Low-Cost Large Area Panel Processing of MCM-D Substrates and Packages" project, and from Politecnico di Torino under the Giovani Ricercatori 1999 grant.

## ■ References

1. A. Dell'Acqua et al., "FERMI—A New Generation of Electronic Modules for Large Data Acquisition Arrays Required by High Energy Physics," *IEEE Trans. Components, Packaging, and Manufacturing Tech., Part B*, vol. 17, no. 3, Aug. 1994, pp. 302-309.
2. FERMI Collaboration, "A Digital Readout System for High Resolution Calorimetry," *Proc. Third Workshop Electronics for LHC Experiments*, CERN/LHCC/97-60, CERN, Geneva, 1997, pp. 388-392.
3. K. Sasidhar, L. Alkalai, and A. Chatterjee, "Testing NASA's 3D-Stack MCM Space Flight Computer," *IEEE Design & Test of Computers*, vol. 15, no. 3, July-Sept. 1998, pp. 44-55.
4. A. Benso et al., "Testing an MCM for High-Energy Physics Experiments: A Case Study," *Proc. IEEE Int'l Test Conf. (ITC 99)*, IEEE Press, Piscataway, N.J., 1999, pp. 38-46.
5. R. Mariani, S. Motto, and S. Giovannetti, "MCM Production: Testing and Related Aspects," *Proc. Fourth Workshop Electronics for LHC Experiments*, CERN/LHCC/98-36, CERN, Geneva, 1998, pp. 584-588.
6. Y. Zorian, "Fundamentals of MCM Testing and Design-for-Testability," *J. Electronic Testing: Theory and Applications (JETTA)*, vol. 10, nos. 1-2, 1997, pp. 7-14.
7. N. Jarawala, "Designing 'Dual Personality' IEEE 1149.1 Compliant Multi-Chip Modules," *J. Electronic Testing: Theory and Applications (JETTA)*, vol. 10, nos. 1-2, 1997, pp. 77-86.
8. A.J. van de Goor, *Testing Semiconductor Memories*, John Wiley & Sons, New York, 1991.
9. L. Breveglieri, L. Dadda, and V. Piuri, "Fast Arithmetic and Fault Tolerance in the FERMI System," *Proc. IEEE Int'l Conf. Application-Specific Systems, Architectures, and Processing*, IEEE CS Press, Los Alamitos, Calif., 1997, pp. 374-383.
10. M. Abramovici, M.A. Breuer, and A.D. Friedman, *Digital Systems Testing and Testable Design*, Computer Science Press, New York, 1990.
11. F. Corno, P. Prinetto, and M. Sonza Reorda, "Circular Self-Test Path for FSMs," *IEEE Design & Test of Computers*, vol. 13, no. 4, Winter 1996, pp. 50-60.



**Alfredo Benso** is a research assistant at Politecnico di Torino, Italy. His research interests include DFT techniques, BIST for complex digital systems, dependability analysis of computer-based systems, and software-implemented hardware fault tolerance. Benso has an MS and PhD in computer engineering from Politecnico di Torino. He chairs the IEEE Computer Society Test Technology Technical Council Electronic Media Group.



**Silvia Chiusano** is a research assistant at Politecnico di Torino. Her research interests include high-level testing, DFT techniques, BIST, and dependability. Chiusano has an MS and PhD in computer engineering from Politecnico di Torino.



**Paolo Prinetto** is a full professor of computer engineering at Politecnico di Torino, Italy, and a joint professor at the University of Illinois, Chicago. His research interests include testing, test generation, BIST, and dependability. Prinetto has an MS in electronic engineering from Politecnico di Torino. Prinetto is a Golden Core Member of the IEEE Computer Society and the elected chair of the society's Test Technology Technical Council.

■ Direct questions and comments about this article to Paolo Prinetto, Politecnico di Torino, Dip. Automatica e Informatica, Corso Duca degli Abruzzi 24, I-10129 Torino TO, Italy; prinetto@polito.it.