

# Testing Embedded Memories in Telecommunication Systems

Stefano Barbagallo and Monica Lobetti Bodoni, Italtel SpA

Alfredo Benso, Silvia Chiusano, and Paolo Prinetto, Politecnico di Torino

**ABSTRACT** Extensive system testing is mandatory nowadays to achieve high product quality. Telecommunication systems are particularly sensitive to such a requirement; to maintain market competitiveness, manufacturers need to combine reduced costs, shorter life cycles, advanced technologies, and high quality. Moreover, strict reliability constraints usually impose very low fault latencies and a high degree of fault detection for both permanent and transient faults. This article analyzes major problems related to testing complex telecommunication systems, with particular emphasis on their memory modules, often so critical from the reliability point of view. In particular, advanced BIST-based solutions are analyzed, and two significant industrial case studies presented.

In the last several years, the telecommunication systems market has significantly grown to include digital communications, real-time fragmented data structure applications such as asynchronous transfer mode (ATM), and video-related areas. The continuous high market pressure to add new features and provide additional services forced a quick reduction of the average *product lifetime*. Concurrently, to maintain competitiveness, costs had to not increase; nor could dependability decrease. To fit these very strong market requirements, manufacturers have been forced to reduce the *time to market* and *time to money* by dramatically shortening the *production cycle*. This has usually been accomplished by a synergistic cooperation of different efforts.

On one hand, the most advanced integration and packaging technologies are usually widely exploited. The required high performance and the capability to process a great amount of data force heavy integration of digital, analog, and radio frequency (RF) devices. On the other hand, resorting to reusability in both hardware and software designs has significantly shrunk the design time, and most designers are nowadays forced to systematically follow *design-for-reusability* rules.

The availability of a significant number of *intellectual property* (IP) *cores* — including processors, DRAMs, digital signal processors (DSPs), and custom blocks — has geared the implementations of *systems on chip* (SOCs), where reusability should not be limited to hardware core design, but extended to all of their test-related issues, including test structures and test strategies. Meanwhile, the market registered an increase in the cost of automatic test equipment (ATE), and consequently in the cost of test per unit. These aspects stressed the introduction of new approaches aimed at embedding test capabilities onboard to the maximum extent (*Built-In Self-Test*, BIST).

Memories are key components in switching systems, where they play a crucial role in terms of availability and serviceability. Moreover, they appear in a big variety of sizes, technologies (SRAMs, DRAMs, Rambus, ...), and packaging (IP cores, chips, dedicated boards). In the last decade, several techniques have been extensively used to identify proper models for *permanent faults* affecting the functional blocks into which a memory can be conceptually partitioned. Moreover, regardless of their actual implementation, memories proved to

be among the most critical devices with respect to *transient faults*, mainly due to environmental stresses and interferences.

In the present article, problems concerning testing embedded memories in telecom systems are investigated, and two industrial case studies developed at Italtel, one of the major European manufacturers of telecom systems, are analyzed.

More precisely, the following section briefly outlines the concept used in the sequel. The article then focuses on BIST methodologies, and details their application to embedded memories. Case studies are presented, and the last section draws some conclusions.

## TELECOMMUNICATION SYSTEMS TESTING

In telecommunication systems, different and synergistic approaches have to be exploited to guarantee the required levels of performance, robustness, *availability*, and *reliability*. Systems must be kept under control during their overall mission time, and faults that are potential hazards, eventually causing failures, should be either prevented or captured as early as possible [1]. *Fault-tolerant architectures* [2] are mostly adopted to detect, locate, and recover failures. *Fault injection techniques* [3] are then used to validate the quality of the fault tolerance capabilities and the reliability of the system. Since physical defects can occur during the production phases as well as in the field, tests must be performed both at manufacturing (*end-of-production test*) and during normal operation (*in-field test*), either periodically, when the system is idle (*offline*), or continuously (*online*) [4]. Offline test is carried on during dedicated time frames, only when the system mission behavior is suspended. It aims to detect permanent faults potentially occurring in the unit under test (UUT). When transient faults have to be detected, online test strategies should be adopted, capable of continuously testing the system concurrently with its normal behavior.

Test strategies and methodologies strongly depend on the type of test. End-of-production and offline tests are usually “fault-oriented,” aimed at detecting specific *target faults*. They identify potential fault sites and suppose that one, or more, of them contain a fault; they then apply to the system a sequence of input values (*test sequence*) to activate the fault. A test sequence *detects* (or *covers*) the fault if the values it generates on the UUT outputs when the fault is present are different from those generated when it is absent. A fault is *undetectable* if no test sequence exists to detect it. *Fault coverage* is the percentage of detected faults with respect to the global fault population.

Online tests, instead, usually continuously *monitor* the sys-

tem behavior in order to minimize the time interval between the occurrence and detection of a fault (*fault latency*).

Historically, UUTs have usually been tested by resorting to ATE: machines charged with physically applying test sequences and analyzing the unit responses. Recent technology advances pose some problems in the capability of ATE to completely fulfill the test requirements and constraints in terms of costs, time, and quality both at the end of production and in the field.

Experimental results proved that *at-speed test* (i.e., applying test sequences at nominal working frequency) guarantees the highest fault coverage. Nevertheless, most advanced chips evidenced the so-called *bandwidth gap*: internally they work at frequencies usually higher than those allowed on their I/O pins. Therefore, external ATE, which obviously must be connected to I/O pins, prevents embedded IP cores from being tested at speed. Moreover, these cores often have a much higher number of pins than are available at the chip level.

Recent economical analyses showed an increasing test cost per unit, due to the diverging gap between ATE and chip costs. Eventually, it should be pointed out that external ATE is usually not available in the field, where a significant test activity is mandatory anyhow.

To overcome these limitations, the most valuable approach proved to be embedding onboard the logic required to perform the test; techniques to realize such "embedded ATE" are usually known as BIST techniques.

As an external test, BIST methodologies may be implemented to support online and/or offline tests. The two approaches can mainly be compared in terms of target faults, latency, and area overhead. In terms of area cost, online BIST solutions generally introduce higher overheads, usually being based on various types of redundancy (structural, temporal, or information). In the meanwhile, they guarantee to cover a significant amount of transient faults and to considerably reduce the latency time, since fault detection is performed concurrently with the normal system behavior. They are thus the natural candidates whenever high reliability is required.

On the other hand, offline BIST solutions are more suitable for testing permanent faults. Since the offline test is performed only when the system is idle, the status of the UUT can be temporarily modified for testing purposes, and specific test patterns applied to check for the presence of various kinds of permanent faults.

Nevertheless, in most practical cases, online and offline approaches coexist in the same system to profitably complement each other. A typical example is provided by telecommunication systems, where online test strategies are used to *monitor* the normal behavior of any single module, and offline BIST is performed to *test* each module after having functionally replaced it with a backup unit.

## MEMORY TEST

As previously pointed out, memories are very critical components, because their fault occurrence probability is higher than in other types of components. Therefore, in high-performance telecom subassemblies, it is mandatory to concurrently guarantee the highest fault coverages and lowest fault latencies. The remainder of this article focuses on problems concerning testing embedded memories, and presents two significant telecom industrial case studies.

The knowledge of the actual internal memory structure can significantly simplify the execution of the test, at the same time providing more accurate results [5, 6]. However, since

such knowledge is usually available at the manufacturing level only, to generate tests to be applied later on (e.g., at the end of production and in the field), test engineers need to resort to a generic *functional model* of the memory.

Meanwhile, they are requested to detect all the physical defects potentially occurring in the actual memory module, without any assumption concerning its actual physical implementation. Physical defects are thus mapped into a set of *functional fault models*, classified according to their possible locations. This approach is obviously costly since, for instance, the test has to look for potential interference between any couple of memory cells, even if most of these interferences cannot physically occur due to the actual cell array structure.

Fault models commonly used to represent physical defects occurring in the memory cell array are:

- *Single and multiple stuck-at*: The logic value of a cell, or line, is permanently stuck at the logic value 0 or 1.
- *Single event upset (SEU)* and *double event upset (DEU)*: The value of one or two bits of the memory cell is flipped.
- *Coupling*: A write operation that modifies the content of a cell changes the content of one (2-coupling fault) or more (*k*-coupling fault) other cells, too.
- *N-bit burst*: A sequence of *N* bits, where the first and the last bit are faulty, and the remaining *N* - 2 bits can be either faulty or correct.
- *Transition fault*: A cell or a line fails to perform a logic transition ( $0 \rightarrow 1$  or  $1 \rightarrow 0$ ).
- *Neighborhood-pattern-sensitive fault (NPSF)*: The content of a cell, and/or the ability to change its content, are influenced by the logic values of a group of neighborhood cells.

Functional faults affecting the address decoding logic can be grouped in four main categories:

- With a certain address, no cell is addressed.
- With a certain address, multiple cells are addressed simultaneously.
- Regardless of the used address, a cell is never addressed.
- A certain cell can be accessed with multiple addresses.

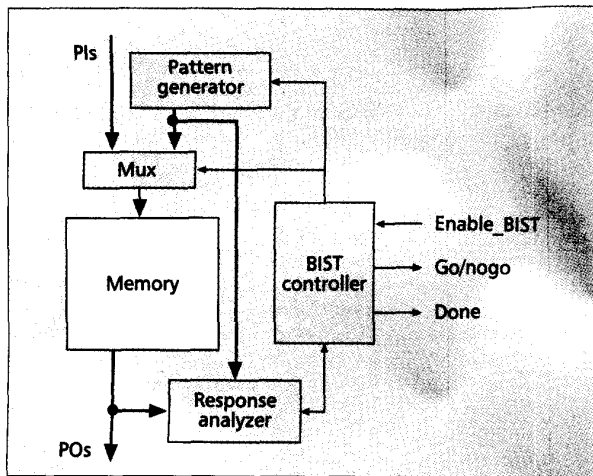
The high integration density of memory chips makes the occurrence of *multiple faults* (i.e., more than one single fault appearing at the same time) a realistic condition. In this case, the test efficiency may be significantly reduced when faults mask each other, becoming undetectable.

Memory *permanent faults* are generally induced by electrical stress due to poor design or handling, intrinsic failure mechanisms inherent to the semiconductor material (crystal and processing defects at the wafer level), or extrinsic failure mechanisms inherent to chip manufacture (metal deposition, boundary, and encapsulation). Instead, radiation of  $\alpha$  particles are typically ascribed at the occurrence of memory *transient faults*.

Memories can be tested offline and/or online. External (i.e., ATE-based) offline testing is suitable when the memory input/output data bus and address bus are easily accessible. Nevertheless, in telecommunication applications, the number of embedded memories is dramatically increasing, up to tens of various-sized memory blocks in the same chip. Most of them are neither directly controllable nor observable from I/O pins. In these cases, BIST architectures are natural candidates to play a key role, thanks to their ability to generate very effective tests in terms of quality and cost, without requiring high accessibility. Nowadays, IC designers can thus find in design cell libraries a wide variety of "BISTed" memory blocks, including, among others, RAMs, ROMs, and FIFOs.

## OFFLINE BIST MEMORY TESTING

Test algorithms cover specific sets of faults in the memory cells, address logic, and read/write logic by simply performing



■ Figure 1. The general structure of an offline BIST architecture.

proper sequences of read/write operation with proper values in proper locations. Algorithms are classified according to the number of covered fault models and to their complexity (i.e., the number of read/write operations vs. the number of memory cells).

Linear complexity tests (MATS+, MATS++, MARCHA, MARCHB, etc.) are typically exploited, thanks to their highly profitable compromise between length (i.e., application time) and coverage (i.e., set of covered faults). Instead, more complex algorithms (GALPAT, GALROW, GALCOL) can be used when it is necessary to locate (not only to detect) the fault [6].

Despite their complexity, test algorithms *destroy* the original content of the memory, since each cell is repetitively written with different values. To overcome this problem, *transparent test algorithms* [7] have been proposed to guarantee that the original memory content is left untouched at test completion. Transparent test algorithms may achieve the same fault coverage of the correspondent original ones, but require a higher test time overhead. Moreover, memory accesses for functional purposes are prevented during the whole test application.

The memory offline BIST architecture consists of adding to the memory a test collar, which includes a *pattern generator*, a *response analyzer*, a *BIST controller*, and additional custom I/O pins (Fig. 1). The pattern generator generates, in the field, the values to be written in the memory array; the response analyzer compares the read values with the expected ones, whereas the BIST controller manages the whole testing phase implementing the test algorithm. The additional pins allow the outside world to activate the BIST phase and to get the test results.

#### ONLINE BIST MEMORY TESTING

Transient faults, particularly SEUs and DEUs, can be critical from the reliability point of view. As an example, if an affected bit is read and used before the fault is detected and recovered, a system failure may occur. If detection is demanded of an offline BIST section, the detection latency may be unacceptable. In such a case, *online BIST*, based on redundancy, is the only viable solution.

Mainly, two different approaches allow information redundancy to be implemented: *hardware redundancy* and *code-based redundancy*. The former implies the replication (once or more) of the entire memory module. Data are written in all the redundant modules, and the data to be output is selected through a voting operation among the data read from each redundant module. Assuming that the voting oper-

ation is implemented in such a way as to guarantee its correctness, faults are easily detected without any service degradation. The overall cost in terms of area overhead is obviously very significant.

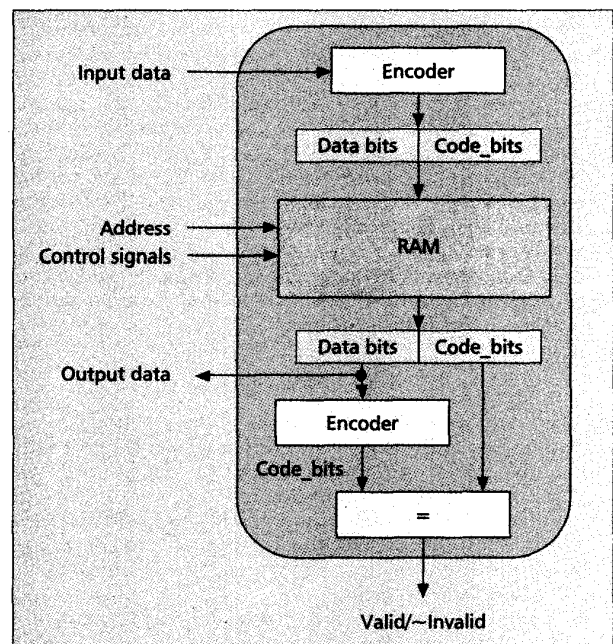
Code-based redundancy usually resorts to the introduction of error detection and/or correction codes [8]. Figure 2 presents a basic online memory BIST architecture based on separable error detection codes. The data written into the memory is not the original one, but a codeword. Therefore, before each write operation the original data is encoded, and then the resulting codeword is written in the memory array. Upon a read operation, the original data is encoded again (the codeword is separable), 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 is considered valid.

Unfortunately, this approach does not cover address faults. A faulty address line would typically result in accessing a cell different from the expected one. Since the accessed cell contains a valid codeword, the code would not detect the error. To solve the problem, more sophisticated architectures must be adopted. A practical solution, adopted in the controller of a data switching system, will be presented below.

#### CASE STUDIES

The present section focuses on two industrial case studies developed by Italtel, one of the major European manufacturers of telecom systems. Both examples concern the test of embedded memories exploited in two proprietary *data switching units* used in voice and narrowband data services, and radio-mobile interconnections. To efficiently manage a high number of channels running at different speed, and to satisfy strict reliability requirements in terms of fault coverage and detection latency, these units must be tested by resorting to BIST architectures.

The former case study, named *ECIP*, demonstrates how offline BIST can be effectively used without interrupting the system mission behavior, testing portions of the system when



■ Figure 2. On-line BIST based on separable codes.

idle. The same approach has been reused to test the entire system when normal system functionality is suspended. The example therefore shows how the same additional test circuitry can profitably be exploited for both end-of-production and in-field testing [9]. The latter example, named *PSE40K*, proves how a complex synergy of different online memory BIST approaches is mandatory to meet very tight reliability constraints.

Since Italtel is a design company relying on external silicon foundries, it is in the company strategy to develop test techniques as independent as possible from the selected foundry libraries. Therefore, all the test strategies presented below have been drawn assuming that no information about the actual physical implementation of the memories was available.

### THE ECIP TEST CASE

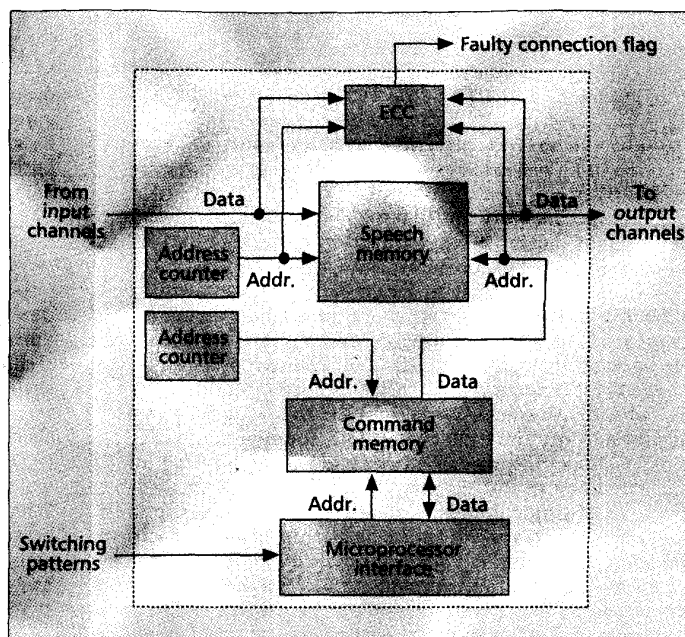
The memory core of the ECIP architecture consists in two dual-port memories, the *speech memory* and the *command memory* (Fig. 3). The former contains data coming from the input channels; it is written by the input sampling logic and read by the output flow generators, which route the data to the correct output channel. Instead, the command memory content controls the switching between input and output channels, supplying the read addresses to the speech memory. An external microprocessor is charged with writing into the command memory the *switching patterns* configuring the connections.

Performance requirements allow a maximum working frequency of 32 MHz, and the overall circuit power dissipation has to be less than 1.5 W. In the actual implementation, the command memory is a 4000 x 16 bits dual-port static RAM, and the speech memory is a dual-port static RAM too, composed of six blocks of 1000 x 8 bits each.

The reliability constraints require continuous checking of the circuitry to detect possible faults in the telephone connections with minimum latency time, without interrupting or degrading the normal circuit function. An additional block, called *embedded connection control* (ECC), verifies the correctness of the connections continuously monitoring the traffic to and from the speech memory. ECC performs a purely functional test: it periodically selects an input channel and samples its input data; when the channel is addressed by the command memory, it verifies the correspondence between the previously sampled data and that appearing at the speech memory outputs. Whenever ECC detects a mismatch, it interrupts the external microprocessor, which can identify the channels involved in the faulty connection and activate the proper recovery.

The ECC functionality relies on the correctness of the addresses produced by the command memory. Therefore, to guarantee that the monitored connections are the correct ones, it is necessary to continuously verify the functionality of the command memory itself.

Given the above requirements and relying on the circuit internal structure, an ad hoc strategy was developed based on an offline BIST approach. For this purpose, the command memory has been logically partitioned into eight blocks of 512 x 16 bits each, and an extra *spare RAM block* of the same size has been added. At any time, eight of the 8+1 available blocks are set to work in normal mode; concurrently, one block is tested. Since the content of the block under test is not functionally required, it is tested using the Nair-Thatte-Abraham algorithm [4], which is able to detect all the stuck-at and coupling faults in the memory array, as well as all the stuck-at faults in the address decoder and read-write logic.



■ Figure 3. The ECIP architecture.

Upon completion of its test, the  $i$ th block is reconfigured to functionally replace the next one: the content of the  $(i + 1)$ th block is copied into the  $i$ th block, and the address decoding logic dynamically reconfigured. The  $(i + 1)$ -th block is then tested.

This solution avoids any possible interference between the test procedure and the normal functional memory access, since they always target different blocks. From the system point of view, the execution of the test is therefore completely transparent, since through the reconfiguration procedure the global ECIP functionality is never interrupted. On the contrary, a conflict may arise if the external microprocessor accesses the  $(i + 1)$ th block when it is being copied on the  $i$ th block. To guarantee data consistency, if the processor performs a write operation on the  $(i + 1)$ th block, the new data is automatically written in the  $i$ th block, too. This operation requires the execution of a few additional memory accesses, and thus its effects on the required test time are negligible.

The adopted solution allows detecting a fault in the command memory within 182 x 512 clocks cycles, according to the reliability requirements. Using a 32 MHz clock, the test of each block is performed in 2.8 ms.

To lower costs, the logic introduced for the actual test of the single blocks is reused for the offline test of the both the command and speech memories. In this case, to reduce the whole test length, all the blocks of the memories are tested in parallel [9].

The chip has been implemented by SGS-Thomson, resorting to the ISB35000 0.5  $\mu$ , 3 V technology. The whole area overhead for the proposed test architecture is less than 11.4 percent of the functional memory area, where the largest contribution is given by the spare RAM module (about 7 percent of the whole memory area).

The added BIST logic has been tested considered as glue logic during the test of the remaining part of the chip. Adopting a partial-scan approach including 72 percent of the existing flip flops, Sunrise™ *TestGen*™ has been able to get fault coverage of 97.6 percent of single stuck-at faults in 47 hr of CPU time on a Hewlett Packard 715/70 workstation equipped with 128 Mbytes of memory.

## THE PSE40K TEST CASE

The second case study presents a complex online architecture intended to be marketed as a significantly improved version of the ECIP system.

The target memory is a 10,000 x 80 bits clocked single-port static RAM, acting as the command memory of an ATM switch unit chip (330,000 equivalent gates in a 0.25- $\mu$  technology). Even if it is possible to access the memory at every clock cycle, functional specifications require a very low average access rate (one access every 200 cycles), thus allowing long idle intervals that can be exploited for test purposes.

The fault detection constraints require targeting (detecting but not necessarily correcting) the following type of permanent and transient faults: stuck-at, coupling faults, SEUs, and DEUs in both the memory cells and addressing logic, and 7-bit burst in the memory cells. The target testable fault coverage of single stuck-at faults is 100 percent. The fault tolerance constraints impose detection of a fault within 100,000 cycles from its appearance (i.e., within 500 memory accesses at the average access rate). Moreover, even if the service should never be interrupted or degraded, fault detection must be performed during the target normal memory behavior.

Figure 4 shows the implemented online memory BIST architecture, which meets the fault tolerance and fault detection requirements by mixing different cooperating approaches (*code-based fault detection*, *architecture-based fault avoidance*, *fault-latency reduction architecture*) properly managed and controlled by a custom "BIST cell controller."

*Code-based* information redundancy implements the main fault detection feature of the architecture. To extend fault detection capabilities to address faults as well, the data and address words are considered as a single bit string, and encoded using a separable Hamming code. Based on the trade-off between the area-overhead constraints and the reliability

requirements, we chose a separable Hamming code requiring a memory overhead of a 7-bit code word for each 80-bit data word. This solution allowed covering single, double, odd bit faults, and 7-bit burst errors in both the data word and the address decoding logic.

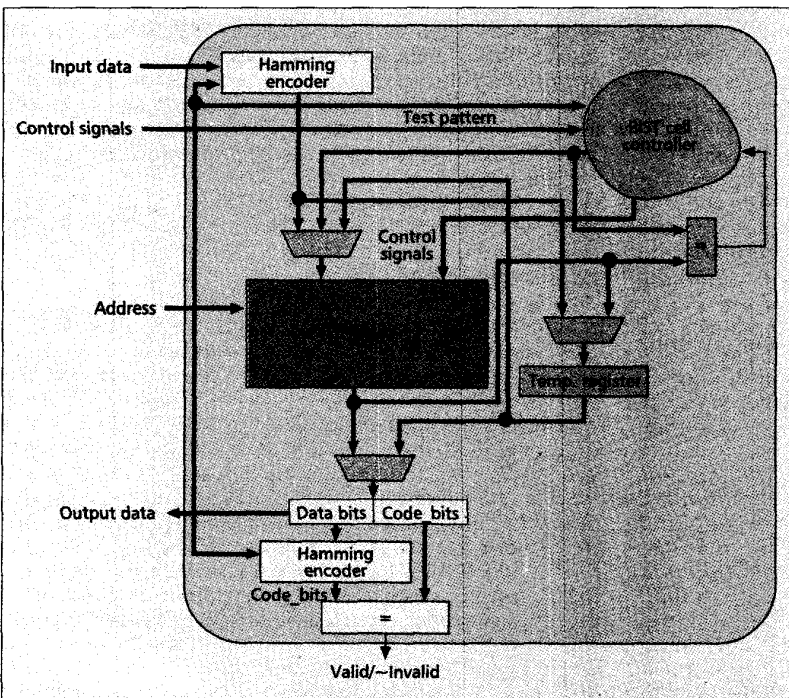
However, if a fault appears in the address logic and the wrong memory cell is addressed, or several bits of the same word are affected, the resulting errors can exceed the 7-bit burst error detectable by the code. To further increase the detection capability of the approach without adopting a more complex and area-expensive code, we resorted to an *architecture-based fault avoidance* approach. In PSE40K, the actual memory module has been split into submodules of width not larger than the length of the maximum burst error detectable by the adopted error-detecting code. The memory has been therefore implemented using 7-bit-wide submodules; the failure of a whole submodule introduces at most a 7-bit burst error. Larger burst errors can appear only in the presence of multiple submodule failures. Nevertheless, experiences on previous designs showed that this type of fault has a very low occurrence probability. Finally, to meet the strict fault latency detection specifications, an ad hoc *fault latency reduction architecture* has been implemented. The fault latency guaranteed by the code-based solution is strictly related with the memory access rate. To achieve the fault detection, a memory cell must be accessed, activating the fault contained in the memory cell itself or in the address logic (e.g., a 1 is written on a stuck-at-0 cell, or a 1 is forced on a stuck-at-0 address line). Nevertheless, especially in highly reliable environments (e.g., ATM switches) with a relatively low system access rate, it is important to detect a fault as soon as possible, without waiting for its activation by means of functional memory cell accesses. In our approach, we "artificially" increased the memory access rate, exploiting the system idle time to perform extra accesses to the memory cells. Cyclically, a *temporary register* functionally replaces

each memory cell. The replaced cell is then tested by a fast and compact algorithm able to detect stuck-at and coupling faults among all the bits of the cell itself [11]. At test completion, the content of the temporary register is copied back to the cell and another cell is set under test. The solution guarantees the detection of any target fault within 100,000 cycles from its appearance, without losing the original content of the cell under test. In addition, no performance degradations are implied: when the mission system needs to perform a read or write operation on the cell under test, the BIST cell controller transparently executes the operation using the temporary register instead of the actual memory cell.

No data about the area overhead and fault coverage of the added BIST logic are available at the moment, since the chip is still under development.

## CONCLUSIONS

In high-performance telecommunication systems, memory modules appear as one of the most critical parts. The strict reliability constraints usually impose very low fault latency and a high degree of fault detection



■ Figure 4. The PSE40K online memory BIST architecture.

of both permanent and transient faults. The adoption of offline and online BIST solutions should therefore be considered in the earliest phases of the design cycle.

Whereas offline architectures have been widely discussed in the literature, and many standard solutions proposed and evaluated, it is not possible to identify a standard online memory BIST architecture. Thus, the best approach is often a composition of a variety of customized local solutions aimed at tackling the different problems, while fulfilling the strict specification constraints in terms of reliability and overhead.

## REFERENCES

- [1] "Telecom Test Seminary," *Int'l. Test Conf.*, Washington, D.C., Oct. 1995.
- [2] D. K. Pradham, Ed., *Fault-tolerant Computer System Design*, Prentice Hall, 1996.
- [3] J. Clark and D. K. Pradham, "Fault Injection: A Method for Validating Computer-System Dependability," *IEEE Comp.*, vol. 28, no. 6, June 1995, pp. 47-56.
- [4] M. Abramovic, M. Breuer, and A. Friedman, *Digital System Testing and Testable Design*, New York: Computer Science Press, 1990.
- [5] B. F. Cockburn, "Tutorial on Semiconductor Memory Testing," *J. Elect. Testing: Theory and Apps.*, vol. 5, Boston, MA: Kluwer, 1994, pp. 321-23.
- [6] A. J. Van de Goor, *Testing Semiconductor Memories: Theory and Practice*, Chichester, U.K.: Wiley, 1991.
- [7] M. Nicolaidis, "Theory of Transparent BIST for RAMs," *IEEE Trans. Comp.*, vol. 45, no. 10, Oct. 1996, pp. 1141-55.
- [8] F. J. Macwilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes II*, North-Holland Mathematical Library, vol. 16.
- [9] S. Barbagallo et al., "Integrating Online and Offline Testing of a Switching Memory," *IEEE Design and Test*, vol. 15, no. 1, 1998, pp. 63-70.
- [10] S. Hamdioui and A. J. van de Goor, "Consequence of port restrictions on testing Two-Port memories," *Proc. ITC '98*, Oct. 1998, pp. 63-72.

## BIOGRAPHIES

STEFANO BARBAGALLO (Stefano.Barbagallo@italtel.it) received his M.S. in electronic engineering in 1988 from the Politecnico di Milano. He works in Italtel SpA, where he was formerly responsible for the design for testability research activity, and currently is the ASIC design manager in the Central R&D Department. His technical interests include ASIC design methodology, built-in self-test, and design for reuse. He is a member of the IEEE Computer Society.

MONICA LOBETTI BODONI (Monica.LobettiBodoni@italtel.it) received her M.S. in nuclear engineering in 1993 from the Politecnico di Milano, and in 1994 she got a Master in Information Technology and joined Central R&D in Italtel SpA. Since 1994 she has worked in the Design for Testability team of the Italtel SpA, supporting the ASIC design teams and defining advanced test strategies for the different designs. She is involved in ASIC DFT, PCB prototyping, and production testing. Her main research activities include embedded memories test techniques, scan insertion, ATPG, logic built-in self-test, and boundary scan.

ALFREDO BENSO (benso@polito.it) received his M.S. degree in computer engineering in 1995 and his Ph.D. in 1998 from the Politecnico di Torino, Italy. He is currently a research assistant at the same university, where his research interests include design-for-testability techniques, BIST for complex digital systems, and dependability analysis of computer-based systems.

SILVIA CHIUSANO (chiusano@polito.it) received her M.S. in computer engineering in 1996 from the Politecnico di Torino, Italy, where she is currently working toward her Ph.D. Her research interests include high-level testable synthesis, high-level testing, design-for-testability techniques, and built-in self-test.

PAOLO PRINETTO (Paolo.Prinetto@polito.it) received his M.S. in electronic engineering in 1976 from the Politecnico di Torino, Italy. Since 1990 he has been a full professor of computer engineering at the same university. His research interests cover testing, testable synthesis, test generation, and BIST. He is a Golden Core Member of the IEEE Computer Society and the elected vice-chairman of the IEEE Computer Society Test Technology Technical Council (TTC).