A 5 Gb/s 9-Port Application Specific SRAM with Built-In Self Test

Steven W. Wood†, G.F. Randall Gibson†, Saman M. I. Adham†, Benoit Nadeau-Dostie‡

†Northern Telecom, P.O. Box 3511, Station C, Ottawa, Ontario, Canada K1Y 4H7
‡LV Software (Canada) Inc., 1525 Carling Avenue, Suite 404, Ottawa, Ontario, Canada K1Z 8R9

Abstract

This paper describes the architecture of a Time-Slot Interchange (TSI) SRAM for a SONET switching application and its associated BIST architecture. To reduce the number of data RAMs required for full switching, the memory throughput is boosted by providing multiplexed access to the core at twice the system clock rate. The nature of the memory requires a novel BIST architecture to ensure full test coverage and ensure easy access of the BIST function at different levels of system integration.

I. Introduction

This paper describes the architecture of a Time-Slot Interchange (TSI) SRAM for a SONET switching application and its associated BIST architecture. The memory port architecture of eight byte-wide, read-only ports and a single, 128-bit write-only port was designed to provide maximum data throughput with the minimum number of memories and to fit easily into the data flow at the ASIC level. To reduce the number of data RAMs required for full switching, the memory throughput is boosted by providing multiplexed access to the core at twice the system clock rate. Other mission-mode features such as self-contained, sequential write-address generation, application-specific address mapping, no-power mode, and page-synchronization controls provide system-level flexibility and simplified ASIC design. A synchronous memory interface with scan path collar and special test modes is provided to simplify scan application and design while enhancing multiport-BIST coverage. The nature of the memory requires a novel BIST architecture to ensure full test coverage.

II. 9-Port Memory Architecture

The 9-port SRAM, shown in Figure 1, is configured as 384 bytes of memory storage with a single 24×128-bit (16 byte) write port and eight 384×8-bit read ports. It is implemented as a 48 row by 64 column core cell array. The 128-bit write is row interleaved with 64-bits written to each of two selected rows per write cycle to give a more optimum array aspect ratio.

A 5-port core cell with four fully-differential read ports and a single-ended pseudo-differential write port has been designed. Differential read ports were selected for enhanced performance relative to a more compact single-ended architecture. An indirect read-access architecture is used to eliminate multiport cell stability problems associated with multiple simultaneous accesses to a cell. The single-ended write scheme combined with row interleaving saves two write bit lines per column. A local bit line inversion scheme is used to provide a pseudo-differential write capability at the core-cell for equivalent-to-differential write performance.

The 9-port operation is obtained by time multiplexing the four physical read ports in the core array. Two full read accesses to the core are performed on these ports in each clock cycle. The read data is then re-timed and latched into eight output ports for presentation to the user on the rising edge of the system clock. All read port inputs are provided to the 9-port interface at the same rising clock edge and are internally pipelined to perform the read

0-8186-7102-5/95 $04.00 © 1995 IEEE

access. Read operation is controlled by a self-timed clock generator that shuts down each read cycle as fast as possible following data access to minimize power. The self-timed operation provides nearly constant power dissipation across all process cases [1]. The entire multiplexed read operation is initiated by a single rising edge of the system clock, simplifying the top-level interface and minimizing clock duty-cycle requirements.

The read memory map is customized to provide two independent data pages which can be switched with a single-bit toggle. The write port data is mapped physically, via data register placement, to match the read data map. No column decoding is required in the write-port. Since the TSI application requires sequential write addressing, a built-in write address counter is provided along with page synchronization controls to detect page switch events.

In addition, the memories are provided with an ultra-low power "no-power" mode by internally gating all interface signals, including the clock, with a power-down signal, eliminating any signal transitions within the memory while power-down is asserted.

In order to facilitate scan and BIST design methodologies, the memory interface includes built-in scan chains for address and data signals plus a special multiport test mode called shadow write. The scan chains are independently controllable so that multi-segment chains can be supported.

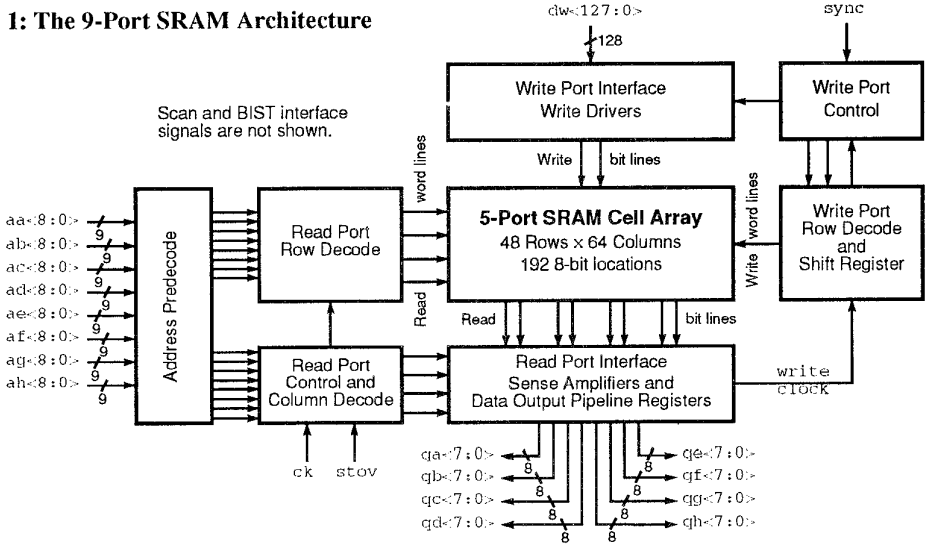
The shadow write mode is provided for inter-port short detection [2] by assertion of a combination of test mode control signals. As shown in Figure 2, two read-port bit line pairs can be driven to ground potential while test patterns are run on the other two bit line to detect read errors. This mode is implemented with no critical path overhead by specific control of the bit line clamp devices.

Performance of the 9-port TSI memory in a 0.8 µm BiCMOS technology is summarized in Table 1. The memory requires only the CMOS devices of the analog enhanced BiCMOS technology described in [1]; no bipolar, resistor or capacitor devices are required.

Table 1: 9-Port TSI Memory Performance

Cycle Time (tCHCH)	4.5 ns
Pipelined Access Time (tCHQV)	3.2 ns
Setup Time (tDVCH)	1.7 ns
Hold Time (tCHDX)	0 ns
Active Power	5.0 mW/MHz
Test conditions: 3.3 V, 27 °C, typical process	

Figure 1: The 9-Port SRAM Architecture



III. Built-In Self Test Scheme

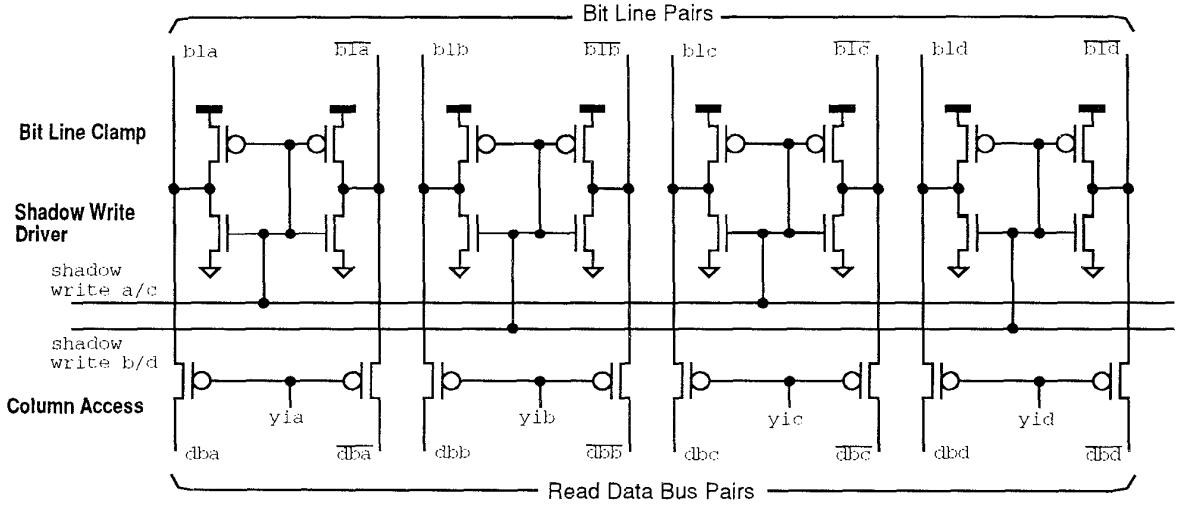
The objective of the 9-port memory BIST is to provide a high fault coverage test for the memory that can be utilized at different levels of system integration. The architecture of the memory mandates a custom self test scheme. The write port does not have random addressing capability because of the sequential write operation. This makes the use of the serial interfacing technique and SMARCH algorithm [2] ineffective for this architecture.

3.1 Fault Model

The objective of the TSI memory BIST is to comply with the system level test strategy by providing high-speed and complete self test capability of the memory and related interconnect, while minimizing the impact of the BIST interface on mission mode performance. The design of the BIST was based on the following considerations:

1. The BIST must detect stuck-open faults in the address decoder of the read only ports. Address transitions are required to sensitize these faults which can only be detected through read contention. The uncertainty of detection through read contention forces the use of at least two pairs of foreground / background patterns.
2. The memory has asymmetrical address space for unidirectional ports. All words are accessible from each port for read or write operations, but words are written 16 at a time to the write port and read one at a time from the eight read ports. The write port access is limited to the implemented sequential addressing scheme; a complete row pair of 16 bytes is selected by a shift register. This selection mechanism, although simpler to use than a normal decoder, can only be tested indirectly by reading from the other ports. Bidirectional read / write ports were not considered feasible for performance and complexity reasons.
3. The fault mechanisms for bit line shorts and word line shorts are described in [2]. The shadow write mode was implemented to allow testing of two read ports while the bit lines of the other two read ports are driven to ground potential. Both bit line shorts and word line shorts are detected in this manner.
4. Shorts between bits of the same word are not possible due to the bit sliced architecture of the memory. Alternate foreground / background data patterns are used to cover potential shorts in the data paths.

Figure 2: Memory Column Access and Bit Line Clamp with Shadow Write



3.2 The BIST Algorithm

The BIST algorithm, shown in Figure 3, is similar to Galpat [3], at least on the read ports. The algorithm is repeated with four different foreground / background (FG/ BG) patterns; i.e. 00,FF,AA,55 (HEX). Where W represents the distinct number of locations for each read port (384). The foreground / background patterns are expressed in terms of bytes as read from the read ports.

The BIST write operation is similar to the normal (functional) write operation. All locations in the RAM are written in bursts with the write operation disabled once both pages of data have been written.

The approximate execution time (T) in clock cycles of the BIST algorithm is given in (EQ 1).

$$T = (2W^2 + 24W) \times 2 \times 4 = 2.4 \text{ M cycles} \qquad \text{(EQ 1)}$$

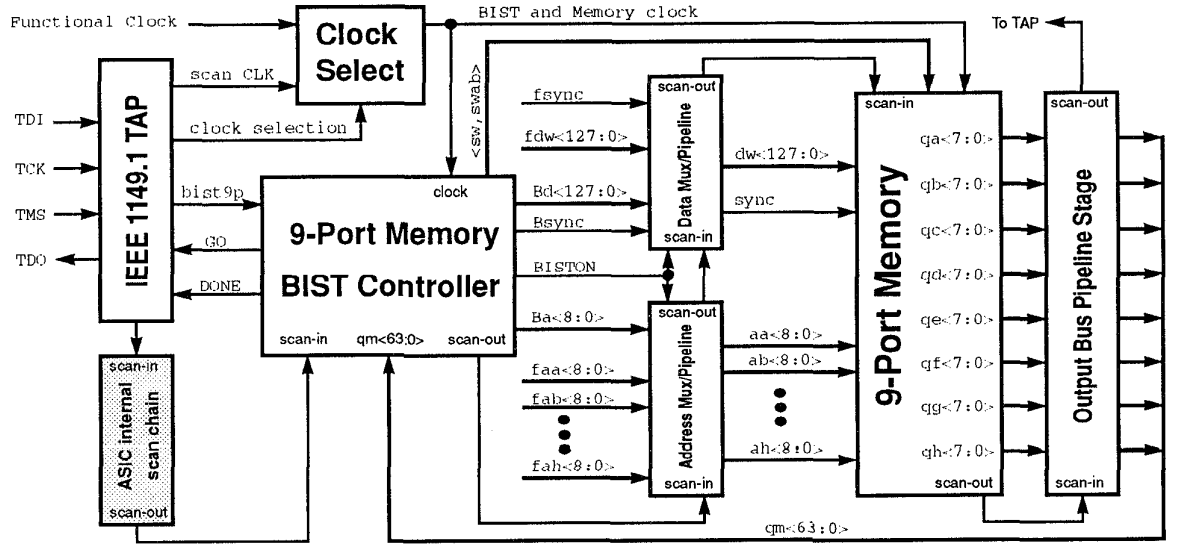
where, the multiplication factor of 4 is used to account for the four different foreground / background patterns to be applied. The multiplication factor of 2 is introduced to account for the shadow write mode in which only half of the read ports are tested at one time.

Note that a variation of the MARCH algorithm [2] plus the DECSO algorithm [3] to detect decoder stuck-open faults is almost as complex as this variation of the GALPAT algorithm.

Figure 3: The BIST Algorithm

```
/* initialize array */
for i = 0 to W-1
    Write 0
endfor

for i = 0 to W-1

    /* write FG / BG */
    for j = 0 to W-1
        if i ≠ j
            Write[j] = BG
        else
            Write[i] = FG
        endif
    endfor

    /* verify FG / BG */
    for j = 0 to W-1
        if i ≠ j
            Read[j] = BG
            Read[i] = FG
        else
            Read[j] = FG
            Read[i] = FG
        endif
    endfor

endfor
```

Figure 4: The BIST Architecture



3.3 The BIST Architecture

The BIST circuit was designed to capture the above algorithm and provide the necessary access to the test through the IEEE 1149.1 test access port (TAP) available on chip. Figure 4 shows the BIST architecture. The number of wires between the BIST controller and the datapath were kept low by limiting the number of data patterns used for the foreground / background. The functional input data bus of the memory is multiplexed with the BIST data under the control of the BIST controller. The read addresses for all eight ports are also multiplexed with the BIST address. The output read data ports are monitored by the BIST controller and compared with the reference data. In both cases the functional pipeline stages are used for the BIST data and address paths. The functional pipeline stage of the output ports were also employed for the BIST data bus.

Performance testing of the memory is achieved by using the functional clock for the BIST controller as well as the memory. However, for diagnostics purposes, the BIST can also be run using the test clock or a slower functional clock.

The BIST is initiated by shifting a private instruction into the TAP instruction register. This instruction forces the `bist9p` signal high. Upon receiving the start BIST instruction the BIST controller goes through an initialization phase and asserts the GO and de-asserts the DONE signals. The BIST controller proceeds to apply the patterns described in the algorithm above and monitors the responses of the memory accordingly. Once an error is detected, the BIST controller de-asserts the GO signal indicating a test failure. If no failure was detected after applying all patterns, the BIST controller asserts the DONE signal. Both the GO and DONE signals are captured in the TAP and are shifted out for off-chip comparison.

The BIST controller implementation is optimized to reduce area overhead and minimize BIST execution time. To reduce execution time, half of the ports are tested for the same address at any time. Furthermore, to decrease the routing area, all read ports are compared to the expected data in parallel with the compare signature shifted out serially.

Figure 5 shows the block diagram of the BIST controller. Four major blocks were designed; the write data generator (WRGEN), the reference data generator (RFDGEN), the Comparator and the BIST Core.

The WRGEN block generates the background data needed to write to the memory. The 128-bit output bus is divided into 16 bytes. The BG data is injected onto the bus depending on the write cycle counter (from the BIST

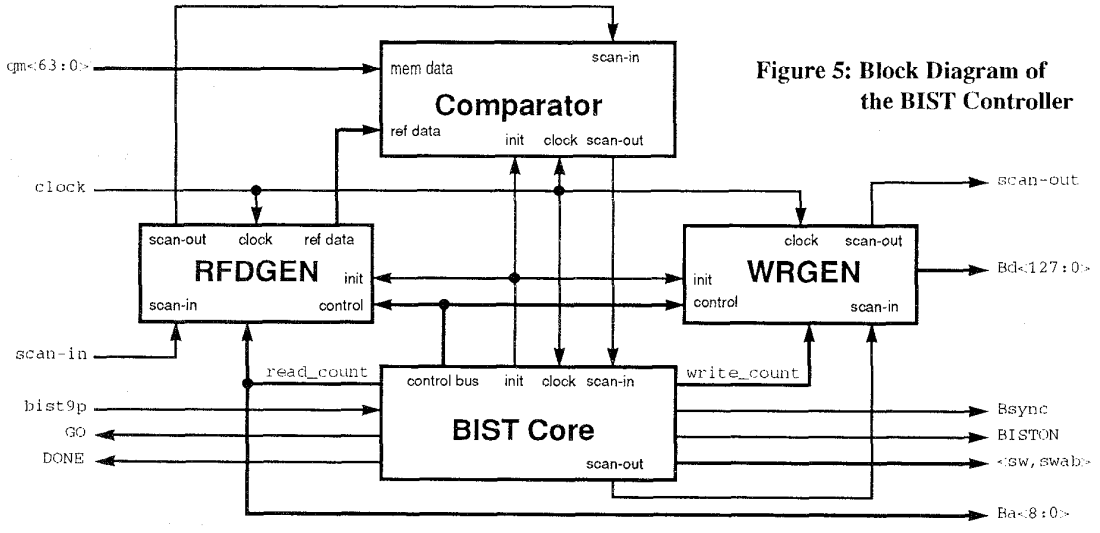


Figure 5: Block Diagram of the BIST Controller

Core). This port is to be connected to the write data port of the memory.

The RFDGEN module generates the expected data (one byte) for all ports. Depending on the read address (generated by the BIST Core), foreground data is calculated and applied to the Comparator to be compared to the data read form the memory.

A 64-bit parallel Comparator was designed to compare the data read form the memory to the reference data generated by RFDGEN. The Comparator is invoked only during the read cycle and only read ports under test are compared. The compare registers of the other ports hold their previous state. At the end of the read cycle, the contents of the comparator registers are shifted out (through the scan path chain) to the BIST Core module and monitored for errors. All zero value data in the compare registers are considered to be the correct signature; otherwise the previous write / read cycle has detected a fault in the memory. This procedure provides diagnostic information for the BIST that simplifies the process of fault location.

The BIST Core block controls the operation of the BIST. It consists of several counters, a signature monitor and a finite state machine (FSM). It is also responsible for communicating to the chip and providing the necessary signals to control the memory interface.

To accommodate the high speed test the BIST controller is designed and optimized for a 100 MHz clock rate. A minimum number of pipeline stages were added at selected places to minimize the area overhead.

A scan chain is inserted in the BIST controller and is connected to the internal scan chain of the chip and 9-port memory scan interface. All pipeline stages are scan testable guaranteeing full testability of the BIST controller and the memory interface.

IV. Conclusions

A high-speed, multiport, application-specific datapath SRAM architecture with its associated built-in self test scheme has been presented.

A 5-port core cell was designed with four fully-differential read ports and a single-ended write port. The 9-port operation is obtained by time multiplexing the four physical read ports in the core array. The memory is designed with built-in features to support structural testing.

To facilitate product testing, a novel BIST architecture based on the Galpat algorithm was designed and implemented. The number of wires between the BIST controller and the datapath was kept low by limiting the number of data patterns used for the foreground / background. The BIST circuit is designed to provide access to the self test function at all levels of system integration. Performance testing is achieved by running the BIST at the functional clock rate, however running the BIST at slower clock speed is also available as an option that can be used for diagnostic purposes.

REFERENCES

[1] A. Silburt *et al.*, "A 180-MHz 0.8-μm BiCMOS Modular Memory Family of DRAM and Multiport SRAM, IEEE Journal of Solid-State Circuits, Vol 28, No. 3, pp. 222-232, March 1993.

[2] B. Nadeau-Dostie *et al.*, "A Serial Interfacing Technique for External and Build-In Self Testing of Embedded Memories", IEEE Design and Test of Computers, Vol. 7 No. 2, pp. 56-64, April 1990.

[3] A.J. van de Goor, "Testing Semiconductor Memories Theory and Practice", John Wiley & Sons Ltd., Baffens Lane, Chichester, West Sussex, PO19 1UD, England, 1991.