Abstract—A design method is presented for testing of analog iterative decoders using digital built-in self-test (BIST). Mixed-signal BIST schemes are often complex and demand larger than acceptable hardware cost. By using a digital BIST scheme, analog iterative decoders can easily be tested in the digital domain. A BIST was designed and simulated for an (8,4,4) extended Hamming decoder using 0.18µm CMOS. It is capable of detecting transistor faults in the decoder. A decoding rate of 444kbps is achieved. The digital BIST scheme is suitable for any iterative decoder using sum-product algorithm.

I. INTRODUCTION

Iterative decoders for error control codes such as Turbo and LDPC codes [1, 2] achieve outstanding performance with bit error rates close to the Shannon bound. Analog implementation of this type of decoders has often been shown to have better performance than digital ones in terms of complexity, silicon area, and power consumption [3, 4]. Some research groups have also demonstrated analog Turbo decoders for commercial standards [5, 6]. However, increasing the testability of such analog decoders has become a major challenge. The testing problem has grown as decoder designs get larger. Efficient manufacturing of larger decoder chips will depend on their testability.

Built-in self-test (BIST), a design-for-testability technique in which testing is done using built-in hardware [9] is a feasible way to improve overall testability and facilitate diagnosis and repair of these decoders. Mixed-signal BIST schemes usually require significant hardware cost and are often complex. This paper introduces a simple digital BIST approach for analog iterative decoders using the sum-product algorithm. The digital BIST scheme can make testing of analog iterative decoders easier.

A digital BIST design is implemented for an analog decoder using an (8,4,4) extended Hamming code. A brief introduction of analog iterative decoding and BIST is provided in Sections II and III. The decoder architecture along with the input/output interface is discussed in Section IV. A method for building the digital BIST is explained in Section V. Advantages of using digital BIST for analog iterative decoders are explained in Section VI. The timing and operation of the BIST and overall decoder chip by simulations are given in Section VII. Finally, Section VIII concludes this paper.

II. ANALOG ITERATIVE DECODING

The principle of iterative decoding consists of exchanging probabilistic (soft) information between constituent decoders [7]. Soft information is iterated several times to improve decoding reliability and decisions are made in the final iteration. The sum-product algorithm is a general algorithm employed to perform decoding operations on factor graphs [8]. It evaluates global conditional probabilities using only local constraints. Constraints which are described as factor graphs can be simply mapped to analog networks. This is due to the reason that certain probabilistic operations can be simply carried out by basic transistor circuits.

Fig. 1 shows the factor graph of an (8,4,4) extended Hamming code. The decoder takes 8 inputs from the channel, and produces 4 outputs, which are estimates of the systematic bits. The upper nodes are equality nodes and the bottom nodes are check nodes. These local functional nodes calculate outputs by multiplying and adding probabilities. Connections between the nodes are bi-directional. Equality and check nodes are implemented using Gilbert multipliers by using currents to represent probability distributions [7].

![Factor graph for (8, 4, 4) extended Hamming code.](image)

III. BUILT-IN SELF-TEST (BIST)

BIST is beneficial by facilitating testing in many ways:

1) BIST provides at-speed, in-system testing.
2) High fault coverage can be achieved.
3) Testing can be localized to avoid having to propagate test vectors and test results to and from hierarchical layers of the circuit under test.
4) Chips can be tested without relying on expensive automatic test equipment. Similar tests can be performed on-line and off-line. Because of these and other advantages, testing costs can potentially be decreased [9].

Fig. 2 illustrates a typical BIST architecture for analog systems. Self-test is performed in a test mode initiated by a 'start' signal and its completion is indicated by a 'complete' signal. The test controller controls the operations of all the functional blocks. The test pattern generator (TPG) creates test vectors. The input multiplexer (MUX) selects between test pattern and primary inputs depending on signals coming from the test controller. The response analyzer (ORA) checks the outputs from the circuit-under-test (CUT) and determines if the device is good or faulty. The analog BIST architecture is similar to one for digital systems except that a digital-to-analog converter (DAC) and/or analog-to-digital converter (ADC) are required.

Figure 2. BIST architecture for analog circuit.

IV. Decoder Architecture

Fig. 3 shows a system block diagram for the (8, 4, 4) extended Hamming code decoder chip. The structure is similar to the Hamming decoder built by Nguyen et al. [10]. The serial-to-parallel interface is developed by Winstead et al. [11]. Operation of the interface is synchronized by a clock signal. It converts the serial data stream received from the channel to a parallel set of input passed to the analog decoder circuit. The computation core represents the analog network, which corresponds to the factor graph shown in Fig. 1.

![Figure 3. Block diagram of an (8, 4, 4) extended Hamming code decoder chip.](image)

Initially, the input data and the reference voltage are stored into the sample-and-hold (S/H) cells. After an entire codeword has been read, the stored data are passed in parallel to the computation core for analog decoding. The core then generates soft output signals representing the probabilities that the received bits were 1s or 0s. At the end, a bank of comparators makes digital decisions by comparing the two probabilities and latches the outputs all at once [11].

V. Design of BIST for Analog Iterative Decoders

The decoder chip is divided into two sections. The first section is the computation core itself. The second section is the input/output (I/O) interface, which contains the S/H cells and the comparators. A BIST circuit is designed for each of these sections. Fig. 4 shows the structure of the decoder chip with BIST.

![Figure 4. Overall structure of decoder chip with BIST.](image)

A. BIST for Computation Core

The computation core is composed of equality and check nodes connected according to the factor graph in Fig. 1. All 4-edge nodes are constructed by connecting two 3-edge nodes in a chain. Therefore, one needs only to consider the implementation of 3-edge nodes from which larger designs can be built. Thus the computation core contains only 3-edge equality and 3-edge check nodes. Our BIST is consequently developed to test each 3-edge node separately. Fig. 5 shows the circuit for a conventional 3-edge check node.

![Figure 5. Conventional implementation of a 3-edge check node.](image)

The diode-connected transistors M1 and M2 provide current-to-voltage conversion at the circuit's outputs. BIST is designed to test all the transistors in the circuit. In order to do...
so, all 3-edge nodes are converted into digital logic circuits during the test. This can be done by adding switches to reroute the wires and to shift the bias voltage and reference voltages to ground (GND) or supply voltage (VDD). When feeding in ones and zeros to the modified circuit, it behaves like a static CMOS logic gate. Fig. 6 shows the modified 3-edge check node with added switches.

![Figure 6. Mode-switching 3-edge check node.](image)

During test mode, the ‘Dig’ signal goes to VDD and ‘Ang’ signal goes to GND. This reconfigures M1 and M2 so that they act as a CMOS inverter. The whole circuit acts like an XOR gate. Same topology is used to test the 3-edge equality nodes.

To start self-test, the BIST sends a signal to the computation core to disconnect all 3-edge nodes and to convert them into logic gates. Afterward, it sends test vectors into each node and waits for the response. The BIST then analyzes the response from each node and indicates if the core is good or faulty. When test is completed, the BIST sends out a finish signal and waits for an input signal to convert the core back to normal operation. Also, the test results for groups of three nodes are gathered into one group; called a 'check group' or 'equality group' for check nodes or equality nodes, respectively. One can access the test result of each group to see which part of the core is not working.

**B. BIST for I/O Interface**

The serial-to-parallel interface consists of some sample-and-hold cells and these cells are composed of one switch for selection, one capacitor for storing the data, and one buffer for isolating the data if there is another stage following [11]. A simple BIST is built to test the I/O interface. During test mode, test patterns are sent and stored into the S/H cells. The stored values then bypass the computation core and are sent directly to the comparators. Finally, outputs from the comparators are checked. Two signals are created to indicate test completion and the test result.

Although the I/O interface uses continuous analog voltages as inputs to the S/H cells, a 1-bit test is sufficient to detect severe faults. More subtle analog faults, such as capacitor mismatch or small comparator offsets, will not be detected.

VI. **DIGITAL BIST FOR ANALOG ITERATIVE DECODERS**

A digital BIST scheme for analog iterative decoders is desirable because of its simplicity. It avoids the use of digital-to-analog converter (DAC) to change the test pattern into an analog signal. DACs demand significant silicon area and BISTs associated with DACs are often complex due to their sampling noise [12]. When using digital BIST schemes, decoders can be divided into sub-circuits and can be tested with simple methods. Analog BIST schemes cannot do so.

When tested in the analog domain, a sophisticated statistical test is needed to confirm the chip performs within design margins. Gaussian noise must be added to the test inputs and the bit errors must be counted. It may be time-consuming to measure a suitably low bit error rate.

Large-scale analog decoders have been shown to be tolerant to mismatch under two conditions [13]. First, the mismatch statistics must not exceed a critical variance, beyond which the decoder's performance degrades exponentially. Second, all transistors must respond to their gate inputs, i.e. there are no stuck-at faults. The first condition is statistical and can be met by design. The second condition applies to all transistors, and is guaranteed by digital BIST.

**VII. SIMULATION RESULTS**

Our decoder was designed in a 0.18µm 6M1P CMOS process. The decoder can correct at most one error. We used a supply of 1.8 V and global bias current of 1 µA. The I/O circuits are clocked at 1 MHz, giving a decoding rate of 444 kbps. An IC is currently being designed. Simulation results shown in this paper were generated using Spectre. The following table describes all the signals shown in the simulation results.

<table>
<thead>
<tr>
<th>Signal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clk</td>
<td>Clock signal</td>
</tr>
<tr>
<td>Test</td>
<td>Signal for initiating self-test</td>
</tr>
<tr>
<td>Dig</td>
<td>Switches circuits between digital and analog mode</td>
</tr>
<tr>
<td>Finish_Core</td>
<td>Indicates if testing of the computation core is done</td>
</tr>
<tr>
<td>Good_Core</td>
<td>Indicates if computation core is good/faulty</td>
</tr>
<tr>
<td>Good_Node</td>
<td>Shows internal node results</td>
</tr>
<tr>
<td>Finish_I/O</td>
<td>Indicates if testing of the I/O interface is done</td>
</tr>
<tr>
<td>Good_I/O</td>
<td>Indicates if I/O interface is good/faulty</td>
</tr>
<tr>
<td>Test_Input</td>
<td>Test input for I/O interface</td>
</tr>
<tr>
<td>Outputs</td>
<td>Outputs from comparators</td>
</tr>
<tr>
<td>Input_Data</td>
<td>Information coming from channel</td>
</tr>
<tr>
<td>Decoded_Bits</td>
<td>Decoded outputs during normal operation</td>
</tr>
</tbody>
</table>

Fig. 7 presents simulation results of the BIST testing the computation core with faults. The core has errors in the first check group and equality group. The test starts at 1 µs and ends at 8 µs. The ‘Good_Core’ signal shows that the core is faulty. An address signal is pre-set to access the result of the first check group at 10 µs, last check group at 12 µs, first equality group at 14 µs, and last equality group at 16 µs. The ‘Good_Node’ signal indicates that the first check group and equality group are faulty and the last check group and
equality group are functional. The core stays in test mode until the test signal goes to GND.

Figure 7. BIST testing the computation core with faults.

Operation of the BIST for the I/O interface can be seen in Fig. 8. The test starts at 1 µs and ends at 47 µs. When testing starts, the ‘Dig’ signal goes to VDD to tell the interface to disconnect from the computation core. The ‘Good_IO’ signal starts going to VDD once the first set of results is correct. It stays high as long as the upcoming results are also correct. This signal indicates that the interface is functional. At 49 µs, the ‘Dig’ signal goes back to GND following the ‘Test’ signal and making the interface connect back to the computation core.

Figure 8. BIST testing the I/O interface.

The operation of the entire decoder chip is shown in Fig. 9. It takes 9 clock cycles to shift in serial data and another 8 clock cycles to perform decoding. 17 clock cycles are required until the first decoded word appears. Two codewords are injected into the decoder. All input probabilities are equal to 0.8. The codeword is 10101011 and 10101000 with errors shown in bold. The corrected bit is seen on the ‘Decoded_Bits’ signal. The self-test is started at 27 µs. The results show that both computation core and I/O interface are functional. When the ‘Test’ signal goes to GND, all the circuits go back to normal operation and all the test signals are reset to zero.

Figure 9. Operation of entire decoder chip.

VIII. CONCLUSION

A digital BIST scheme is introduced in this paper and the technique is applied to test an (8,4,4) extended Hamming code decoder. The decoder is modified such that it can be tested in the digital domain. Simulation results show the correct functionality of the decoder with BIST, demonstrating the feasibility of digital BIST in analog decoding. An integrated circuit is currently being designed.

REFERENCES