An Area Efficient Mixed-Signal Test Architecture for Systems-on-a-Chip

Hari V. Venkatanarayanan Michael L. Bushnell
hariven@caip.rutgers.edu bushnell@caip.rutgers.edu
ECE Dept. and CAIP Research Center, Rutgers University
96 Frelinghuysen Rd., Piscataway, NJ 08854-8088

Abstract

Mixed-signal systems-on-a-chip *(SoCs) are tested using the IEEE 1149.4 analog test bus, but the area overhead and test time are high. We present a new mixed-signal SoC test architecture, which uses the circuit components along with* design-for-testability *(DFT) hardware. Rather than building a custom analog test waveform generator on chip exclusively for testing, we instead stimulate a digital core to generate a digitized version of the analog test waveform at its outputs. We use a functional* digital-to-analog converter *(DAC) connected to the core's outputs to convert the samples into an analog multi-tone test waveform, for the cascaded analog core. We add a small amount of DFT hardware into the digital circuit to make it generate a more accurate sample set. In this new test architecture, first, the DFT hardware is tested, and then, the DAC is tested for its static and dynamic parameters using the test tones generated from the digital core. Next, the digital core is tested using vectors applied at the primary inputs and, finally, in the* analog test mode*, the analog tones, generated using the digital core and the DAC, test the cascaded analog core. On five mixed-signal SoCs, the new architecture reduced average test area overhead by 78.6% and test times by 74%, compared to the 1149.4 standard.*

1 Introduction

At present, mixed-signal *systems-on-a-chip* (SoCs) are widely used, due to cost advantages. They often include voice band analog circuits, but this introduces a serious testing problem due to the uncontrollability of the analog core inputs and the unobservability of the digital core outputs. This problem is solved by the IEEE 1149.4 standard [10], but with very high *design-for-testability* (DFT) hardware overhead on the analog core, long test times, and limited test signal bandwidth.

We propose a new test architecture for monolithic SoCs with voice-band analog hardware cores and on-board *digital-to-analog converters* (DACs), as well as microprocessors and/or *digital signal processors* (DSPs). With test patterns, we stimulate the digital core that drives the analog core through the DAC, so that in test mode, the digital core generates a sequence of high-quality samples that, when converted to an analog signal by the DAC, provide the multi-tone analog test waveform for the analog core. This eliminates an on-chip analog waveform generator solely for analog test, and the need to shift in control signals to configure the 1149.4 analog pin DFT hardware to receive an analog test waveform from an *automatic test equipment* (ATE) through external analog test signals.

We first analyze the analog core and determine the multi-tone analog test waveform needed for static linearity and distortion tests. We digitize the desired multi-tone waveform, accounting for the DAC resolution. Bisaria and Bushnell's [2] method characterizes an arbitrary sequential circuit. They stimulate the digital circuit with random inputs, and record the autocorrelation of the digital outputs and the cross-correlation between each digital output and each digital input, in a matrix C [2]. They record the flip-flop correlations, by treating each one as a *pseudo primary input*, *pseudo primary output* pair. Their method usually predicts the next input sequence necessary for the sequential circuit to generate a desired output sample, by multiplying the output sample by C. This input sequence will generate the desired output sample set for the digital core, with the exception of a few *primary outputs* (POs). They provide DFT hardware to control these POs directly. This uses far less DFT hardware than the 1149.4 standard.

We also test the digital core that is being used as a *tone generator* (TG). We analyze the digital core cascaded with the DAC using a digital core input sequence that provides the conventional ramp (at the digital core output) for testing the DAC for *differential non-linearity* (DNL), *integral non-linearity* (INL), and *signal-to-noise ratio* (SNR). Excessive DNL indicates either a defective DAC or digital stuck-at faults in the digital core. A simulation indicates which stuck-at faults are tested in this fashion. The undetected stuck-at faults in the digital core are targeted using sequential *automatic test-pattern generation* (ATPG). We apply the vectors generated for these faults, called *ATPG test vectors*, to the digital core. We measure the SNR for the DAC while these vectors are applied, and compare that to the *noise floor*. But, the SNR measured at this output will be the combined SNR of the DAC and the analog core. We overcome this by tapping the value at the output of the DAC with an *analog MUX* (AMUX) so that the raw DAC output is routed to an output analog pin. Stuck-at faults in the digital core cause the noise of the DAC to rise above the *noise floor* and are detected in this fashion. However, some stuck-at faults alias with the good machine noise in the DAC, so between the digital core outputs and the DAC, we magnify the effect of aliasing faults by activating hardware that flips the *most significant bit* (MSB) going from the digital core into the DAC, thus grossly magnifying the noise effect of the fault.

We created five SoCs and analyzed their testing and DFT

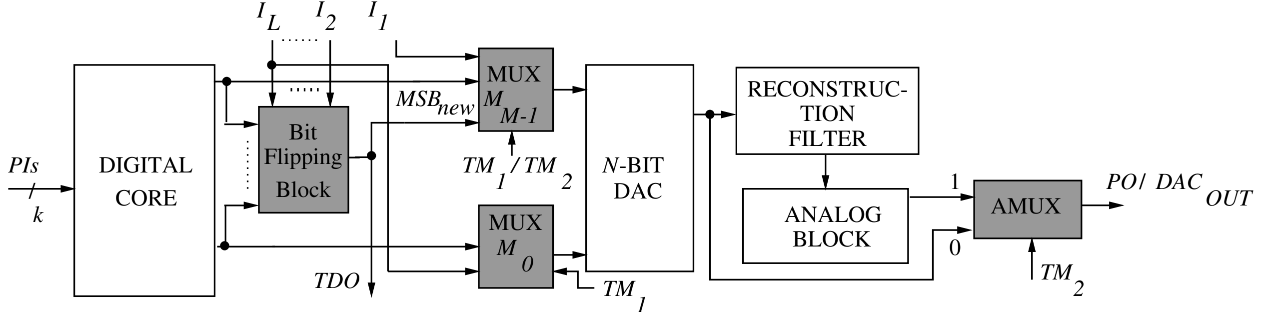

Figure 1: Mixed-signal SoC test framework with added test hardware shaded

hardware, compared to using the 1149.4 analog test bus. On average, we reduced DFT hardware by 78.6%, and test time by 74%. We increased the bandwidth of the analog test signal that could be created on-chip, compared to using the 1149.4 bus, by eliminating shifting sequences to scan in set up signals for the analog test pins. We can generate much higher frequency test tones than the 1149.4 standard, for communications circuit test.

Section 2 discusses prior work, Section 3 presents our architecture, Section 4 describes our SoC test procedure, Section 5 proves that our testing is complete, Section 6 presents results, and Section 7 concludes.

2 Prior Work

The test architecture for testable SoCs [3] uses a *test source* to provide test vectors, a *test sink* for an on-chip signature analyzer, and a *test access mechanism* (TAM), which is a user-defined test data communication structure. The *test controller* is the 1149.1 boundary scan *test access port* (TAP) and controller. The SoC has boundary scan hardware added to all digital and analog pins, with the analog portion tested using the IEEE 1149.4 standard [10].

Dufort and Roberts [4] generated multi-tone sinusoidal signals by periodically applying digital *pulse code modulation* (PCM) bit streams to a sigma-delta modulator. Ozev *et al.* [11, 12] proposed a system-level test synthesis methodology for mixed-signal designs using a basic block level test translation approach. This finds signal paths through which the test inputs and responses can propagate. Fang and Kerkhoff [5] proposed a core-based algorithm for finding the tolerance-box for each individual analog core in a given mixed-signal SoC. They determine the tolerance box for each analog core, and then for the entire test path. Sehgal *et al.* [13] optimized TAMs and test scheduling for mixed-signal SoCs. Tofte *et al.* [14] and Ong *et al.* [9] proposed *built-in-self-test* (BIST) for mixed-signal SoC's using a second-order delta sigma modulator, and an on-chip DSP for generating the digital test stimulus.

Linearity testing of DACs involves applying a ramp signal as input to the DAC. The actual response is compared with the expected response by an ATE to determine static parameters such as DNL and INL [7, 8]. Hassan *et al.* [6] tested DACs by inserting them into an oscillating loop to form a sigma-delta modulator.

3 Mixed-Signal SoC DFT Architecture

The new SoC test architecture is in Fig. 1. The digital core, the N-bit DAC, and the analog block are the CUTs. These are the testing components of this architecture:

- The *tone generator block* (TG) is the *digital core* and the N-bit *DAC-under-test*.
- The *analog block* or core is the various analog *components-under-test* (CUTs).
- The *primary inputs* (PIs) input the test patterns and signals $(I_1, \ldots, I_L)$ input the good machine responses to test the TG and operate the tone generator.
- The *analog PO* observes the analog block output. The DAC_{OUT} pin, which is the raw DAC output before reconstruction filtering, is MUXed to the analog PO for observing the TG block output.
- The *bit flipping* block (optional) magnifies stuck-fault effects in the *least significant bits* (LSBs) by flipping the MSB as well, when the LSB flips, to introduce a huge distortion in the output analog spectrum due to the fault. Stuck-at faults affecting the LSBs of the digital core outputs are unlikely to significantly distort the *Fast Fourier Transform* (FFT) of the DAC output, but flipping makes the stuck-at faults observable through the DAC.
- The *digital multiplexers* $(M_0, \ldots, M_{M-1})$ control inputs to the DAC to integrate the digital core into an analog test tone generator, where M is the number of DAC input bits needing direct control from PIs.
- The *analog multiplexer* (AMUX) switches the analog PO between the analog core output and the DAC_{OUT} pin.
- The *test pins* (TM_1, TM_2, and $I_1 \cdots I_L$) control the DFT hardware. The TDO pin observes the bit flipping block during test.

$L \neq M$ because not every DAC input line needs DFT hardware.

Digital and Analog MUXes. The MUXes (M_0 to M_{M-1}) are controlled using the TM_1 pin, to apply direct digital signals to selected bits of the DAC in order to make it synthesize a desired analog test waveform. The digital core synthesizes a sample set for a nearly correct realization of a desired analog multi-tone test waveform. However, a few bit positions usually need to be provided directly, for high waveform quality, by these MUXes. In the M_{M-1} MUX, the $\{TM_1, TM_2\}$ pins select signals from

either the digital core, the bit flipping block, or the external pin. We observe the DAC output signal DAC_{OUT} directly without reconstruction filtering through an AMUX, controlled by signal TM_2, on the analog output of the system.

Modes of Operation. Table 1 shows the four operation modes, depending on $\{TM_1, TM_2\}$. Multiplexers M_0 to M_{M-1}

Table 1: Selection of modes

$\{TM_1, TM_2\}$	Mode	$\{TM_1, TM_2\}$	Mode
01	Normal	10	DAC-test
00	Digital test	11	Analog test

(if M_{M-1} is controlled by the TM_1 pin) receive inputs from the digital core when $TM_1 = 0$ (*normal* and *digital test modes*) and from the external pins I_1 to I_L when $TM_1 = 1$ (*DAC-test* and *analog test modes*). The AMUX selects the DAC output when $TM_2 = 0$ (*DAC-test* and *digital test modes*) and selects the analog component output when $TM_2 = 1$ (*normal* and *analog test modes*).

4 Testing the SoC

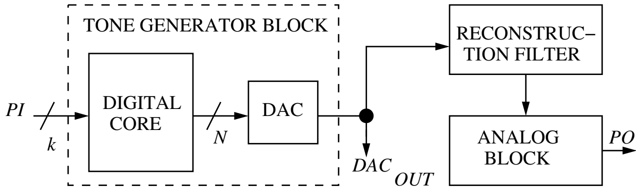

Figure 2: Test configuration

4.1 Tone Generator

We test the TG by: (1) testing the digital core by propagating the digital faults through the DAC and observing their effects in the analog output, and (2) testing the DAC with input test patterns generated by the given digital core. Digital test input patterns are applied at the PIs and the analog output is observed at the DAC_{OUT} pin (see Fig. 2). The *tone generator* has errors due to digital block *stuck-at* faults, and DAC nonlinearity errors and noise. We test DAC errors with SNR and DNL tests.

The TG is tested by capturing the digital core and DAC fault effects in the DAC output signal.

1. The DAC errors are detected by applying the DAC test patterns and from the worst case DNL error, the offset error, and the gain error (called the *DNL test*). The N-bit DAC ramp test input sequence ranges from minimum (0) to maximum code (2^{N-1}) and back again, for 50 cycles.
2. The digital core *stuck-at* faults can be detected in the DAC output, by using the ATPG and DAC test vectors to excite and propagate fault effects to the DAC output, where they perturb the analog spectrum in phase or magnitude.

The DAC test patterns can also propagate the faulty responses from the digital core through the DAC to its analog output, but not all digital *stuck-at* faults are detected this way.

The TG is tested as one unit, by applying the ramp-type DAC test patterns during DNL test. If either the DAC or the digital core fails, their faults produce non-linearities in the raw DAC analog output. A DAC is faulty if the computed parameters exceed $DNL_{\mathbf{cut-off}}$, $offset_{\mathbf{cut-off}}$, or $gain_{\mathbf{cut-off}}$ values given in the core vendors' data sheets. Digital core *stuck-at* faults propagated by DAC test patterns (ramp type) are detected if:

$$Worst\ case\ DNL\ error_{\mathbf{fault}} > DNL_{\mathbf{cut-off}} \tag{1}$$

Next, during SNR test, the ATPG test patterns detect the digital core *stuck-at* faults that were previously undetected. We find the *noise floor* of the DAC in terms of the SNR using the DAC test patterns and then find the SNR using the ATPG test patterns (SNR_{ATPG}). A fault in the digital core makes $SNR_{ATPG} < noise\ floor$.

Sometimes, the digital faults are detected by neither the DNL nor the SNR test, because their fault effect aliases with DAC errors. If a *stuck-at* fault flips only an LSB, then due to DAC errors, its effect on the analog output may be less than an LSB error, and the DNL test declares the TG good. For the SNR test, the SNR_{ATPG} will be much closer to the *noise floor*. Detecting the digital core faults using the SNR test depends on the aliasing probability of the digital core stuck-at faults with the good noise in the DAC, which is determined by pre-testing analysis (see Section 4.3).

The digital core faults with 1 or 2 LSB errors will produce maximum error using the *bit flipper* DFT hardware, where the MSB of the DAC is flipped if there is a flipping due to a fault in any of the low-order bits. This hardware is described in Subsection 4.4. Thus, all faults produce a DNL error greater than 1 LSB and are detected by the DNL test, or the SNR_{ATPG} is less than the *noise floor* and faults are detected by the SNR test.

4.2 Testability of the Tone Generator

We analyze the cases where the faulty digital core and DAC can be detected using the DNL and SNR tests. A TG can have any one of these cases:

- *Case 1*: The DAC is faulty.
- *Case 2*: The digital core is faulty.
- *Case 3*: Both the DAC and the digital core are faulty.

For Cases 1 and 3, the DNL test will find the DAC faulty. For Case 2, the DAC is good, and the worst DNL, offset, and gain errors will be below the $DNL_{cut-off}$, $offset_{cut-off}$, and $gain_{cut-off}$ values and the DNL test will find the DAC to be good. Some digital core *stuck-at* faults may alias with the good DAC noise and be undetected by either DNL or SNR tests.

In Case 2, faults that were not detected by the DNL test are now detected by the SNR test and if the SNR_{ATPG} value of these detected faults is near the *noise floor*, then the test will declare the digital core good, even though the fault effects are detectable

at the digital core PO. Since the DAC is good, the DNL test will declare the DAC and the TG good. To avoid this, we determine the aliasing probability of the *stuck-at* faults of a circuit with the good DAC noise using pre-testing analysis. If there is aliasing, then *bit flipping* is used to eliminate it completely. The DAC errors will be within their tolerance limits and these errors will be averaged over a number of cycles. For *stuck-at* faults with high aliasing probability, *bit flipping* removes aliasing.

4.3 Pre-Testing Analysis

Pre-testing analysis determines whether the *stuck-at* faults of a given circuit alias with the good noise of the DAC. We prove that digital core faults will get detected in the presence of good noise in the DAC, as follows:

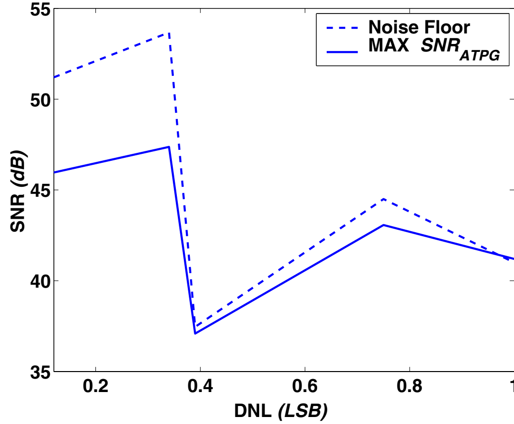

Figure 3: Aliasing probability of c432

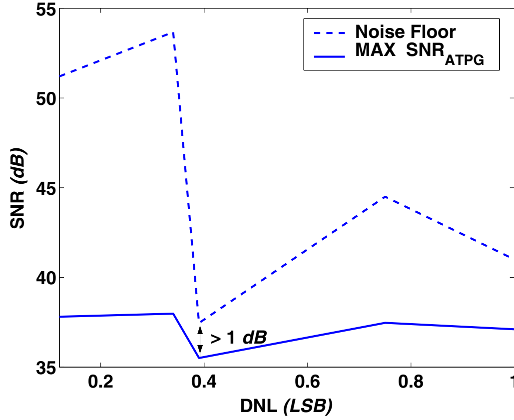

Figure 4: Aliasing probability of c432 after bit flipping

1. Simulate the given TG with various errors in the DAC that are within tolerance limits. Consider only the errors with *worst case DNL error* $< DNL_{\mathbf{cut-off}}$.
2. Find the *noise floor* of the DAC in each case.
3. Find the SNR_{ATPG} value for all the faults.
4. Find the maximum faulty SNR_{ATPG} value.
5. Find whether the *noise floor* and the maximum faulty SNR_{ATPG} value alias at any point.
6. If they alias then use *bit flipping*.

For example, Figs. 3 and 4 show the aliasing probabilities of the c432 circuit before, and after, *bit flipping*. Circuit c432 is a 27-channel interrupt controller. The plots show the values of the *noise floor* and the maximum SNR_{ATPG} for the DAC DNL less than 1 LSB only.

4.4 Bit Flipping to Increase Digital Fault Effects

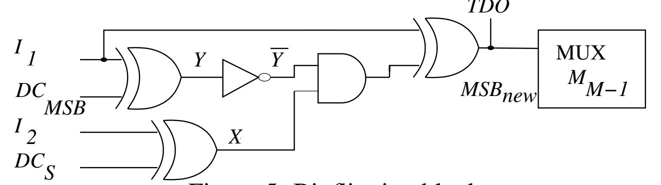

Figure 5: Bit flipping block

The optional bit flipping block (see Fig. 5) is used only if *stuck-at* faults in the digital core alias with DAC noise. The inputs DC_{MSB} (the MSB of the digital core) and DC_S are from the digital core and the inputs $I_1 \ldots I_L$ are from the external pins and provide the expected good machine response. DC_S is the bit that flips. Usually, none or one bit is selected for *bit flipping*.

Bit flipping increases the digital fault effect in the DAC analog output, so that the DNL exceeds 1 LSB ($DNL_{cut-off}$) and the SNR will be less than the *noise floor* by 1 dB in the presence of digital core faults. A 1 dB SNR difference between the *noise floor* and the SNR_{ATPG} equals 2 or 3 LSB errors.

We select the digital core output bit(s) for which the MSB of the digital core (DC_{MSB}) is flipped. A is the number of faults that are aliasing and V is the number of test vectors. Let $max_count = A \times V$ denote the upper bound on the number of times bit k can flip. If bit_count_k denotes the number of times a bit k has flipped for the faults (from simulation) and $fault_count_k$ denotes the number of faults for which bit k flipped, then bit k's weight is:

$$bit_weight_k = \frac{bit_count_k}{max_count} + \frac{fault_count_k}{A} \tag{2}$$

Bits are sorted according to their weights in decreasing order to give a *preference list*, and are selected as follows:

- Flip the DC_{MSB}, when there is a flipping in the bit from the preference list.
- Determine the DNL and SNR change caused by the faults in the DAC analog output.
- Select the bit(s) for which, on flipping DC_{MSB}, the maximum DNL error for each of the faults *exceeds* 1 LSB or the SNR for each of the faults is maximal.

If the selected bit flips, then DC_{MSB} is flipped. The occurrence of flipping in the selected bit is determined by comparing the actual response with the expected (good machine) response. We flip DC_{MSB} when due to a fault, the selected bit flips, but DC_{MSB} does not flip. Otherwise, DC_{MSB} is not flipped.

Fig. 5 is the *bit flipping* hardware to flip DC_{MSB} when only one digital core output bit flips (DC_S is one of the LSBs affected by the fault). The new DC_{MSB} value is:

$$MSB_{new} = I_1 \oplus (\overline{Y} \cdot X) \tag{3}$$

Note that MSB_{new} and TDO are the same signal.

For example, let the bit selected by the *bit selector* for flipping the MSB be bit 0. Table 2 shows the result of flipping the MSB for the given expected and actual responses. The MSB value of the actual response is changed in responses 3 and 4, since there is a flipping in bit 0 of the actual response. The MSB value of the actual response is unchanged in response 5, since flipping occurs in bit 1 and not in bit 0. When multiple bits are selected for flipping, we OR the flip command signals $(\overline{Y} \cdot X)$ from all such blocks, and use the ORed signal as the command to the XOR gate to flip the MSB.

Table 2: Flipping of MSB based on bit 0

	Response						MSB after Flipping		
	Expected			Actual					
Bit	2	1	0	2	1	0	2	1	0
	1	0	0	1	1	0	1	1	0
	1	0	1	1	1	1	1	1	1
	1	0	0	1	1	1	0	1	1
	1	0	0	1	0	1	0	0	1
	1	1	1	1	0	1	1	0	1

Testing the Bit Flipping Hardware. The *bit flipping block* is tested in the *digital test mode*, with test patterns applied at pins I_1, I_2, DC_S, and DC_{MSB}, and faults are observed at TDO. For cases where we select more than one LSBs of the digital core for bit flipping (we select at most 2), so that their fault effects can be magnified by flipping the MSB of the digital core, we will use multiple bit-flipping blocks. These blocks are tested by wiring their outputs to multiple TDO pins. We MUX any of the available SoC output pins in test mode to route TDO_1 and TDO_2 to the pads.

4.5 Reconstruction Filter and Analog Block Testing

The reconstruction filter in Fig. 2 is tested separately from the DAC using TG tones, to avoid aliasing of the DAC and reconstruction filter noise. The analog components in the analog block are tested using the TG [1] (see Fig. 2). The inputs to the TG block are the digital test patterns, from the PIs. The fault-free response of the digital core is converted into analog test tones by the DAC.

5 Proof of Digital Core Fault Coverage

Theorem 1: *A given digital core's F_T non-redundant stuck-at faults are detected using the DNL and the SNR test. The DNL test detects (x) and the SNR test detects the remaining $(F_T - x)$ stuck-at faults, where $x > F_T - x$.*

Proof: The x faults detected by the DNL test have worst case *differential linearity error* (DLE), $DLE > \pm$ 1 LSB, and are detected by ramp test patterns. The remaining $F_T - x$ faults are detected using the SNR test, if and only if the SNR_{ATPG} is greater than the *noise floor* (SNR_{DAC}) of the DAC, during pre-testing analysis.

$$|SNR_{ATPG} - SNR_{DAC}| > 1\ dB\ \ (limiting\ condition) \tag{4}$$

If this is not true, then *bit flipping* hardware is added to achieve it. The ramp type test patterns are exhaustive, so they can detect the maximum number of faults and hence $x > F_T - x$. ■

6 Results and Analysis

The test scheme has two stages.

Stage I – Tone Generator Test

1. Apply DAC test patterns and calculate DNL error, offset error, gain error, and SNR *noise floor* in dB.
2. Apply the ATPG test patterns to calculate SNR_{ATPG}.

There are two possible outcomes:

- Case 1: If the DAC parameters calculated are within the tolerance limits and the SNR calculated using the ATPG test patterns is the same as the *noise floor*, then neither the digital core nor the DAC is faulty (the TG is good).
- Case 2: If either the DAC parameters calculated are outside the tolerance limits or the SNR calculated using the ATPG test patterns is less than the *noise floor* by 1 dB, then the TG is faulty.

Stage II – Analog Component Test

The different analog components in the analog block are tested by generating the required analog test tones using the TG block, which is found to be good from Stage I. For both Stages I and II, test patterns were applied for *50 cycles* to average out noise.

Table 3: SoC components

SoC #	Digital Core	R-2R DAC	# DAC Bits Controlled	Analog Component
1	8-bit Adder	9-bit	3	1st order RC LPF
2	c432	7-bit	3	1st order RC LPF
3	pcont2	8-bit	3	1st order RC LPF
4	pcont2	8-bit	3	2nd order Biquadratic LPF
5	pcont2	8-bit	3	1st order $\Sigma-\Delta$ Modulator

The test scheme was implemented on five SoC designs shown in Table 3, four of which used a *low-pass filter* (LPF) as the analog core. Note that pcont2 is an 8-bit controller for DSP applications. The results shown in Table 4 are for SoC5 and were obtained by simulations on a *SunBlade 100 SPARC* system. Three external pins are required for controlling bits 5, 6, and 7 of the DAC directly for generating test tones, and other bits are controlled from the digital core. All five SoCs had a 100% *test case coverage* (TCC). The TCC is:

$$\%TCC = \frac{\#\ of\ bad\ cases\ declared\ as\ bad}{Total\ \#\ of\ bad\ cases} \times 100$$

Table 4: Test scheme results for SoC5

Test Case	Error Type	Stage I	Stage II	Test Result
1	7016 stuck-at faults	DNL Test = 5668 faults, SNR Test = 1348 faults, Op-Amp Offset Error = 0.76 *LSB*	–	Bad
2	7016 stuck-at faults	DNL Test = 4959 faults, SNR Test = 2057 faults, Op-Amp Offset Error = 0.03 *LSB*	–	Bad
3	DAC Op-Amp Offset Error	Offset Error = 1.54 *LSB*	–	Bad
4	DAC Op-Amp Gain Error	Gain Error = -2.03 *LSB*	–	Bad
5	Resistor Error (R-2R ladder)	DNL = 2.63 *LSB*	–	Bad
6	Both Digital Core & DAC Faulty	DNL = 1.19 *LSB*	–	Bad
7	Noisy Modulator	DNL = 0.24 *LSB*	SNR $\simeq$ 68 *dB*	Bad
8	No Fault Components	DNL = 0.35 *LSB*	SNR $\simeq$ 90 *dB*	Good

IEEE 1149.4 Area Overhead and Test Time. The chip area overhead for the 1149.4 standard is calculated in terms of the digital and analog test hardware and the number of test pins. The *TAP controller* has 8 flip-flops, and 38 logic gates. The *digital boundary module* (DBM) has 2 MUXes and 2 flip-flops. The *analog boundary module* (ABM) has decoder logic, 6 switches, 4 MUXes, 8 flip-flops and 1 digitizer. The *test bus interface circuit* (TBIC) has 10 switches, 2 digitizers, and a decoder. We analyzed the layout areas of analog components, and found their equivalent number of NAND gates, in terms of chip area in a 0.25 μm process: analog switch (12), digitizer (81), and test pin (490). We calculated test hardware area overhead in terms of these equivalents.

Let V_{ATPG} and V_{ramp} be the numbers of ATPG and ramp type test vectors, respectively. PI_FF, PO_FF, and DAC_FF are the numbers of *scan-in*, *scan-out*, and DAC flip-flops, respectively. T is the number of test cycles for which the DAC test input and the analog component test input are repeated. Total test time is:

$$\begin{aligned} Total\ Time_{1149.4} &= Digital\ Core\ Time + DAC \\ &\quad Time + Analog\ Time \qquad (5) \\ Digital\ Core\ Time &= (PI_FF + PO_FF) \times V_{ATPG} \\ &\quad \times T_{CLK} + (V_{DC} \times T_{CLK}) \\ DAC\ Time &= DAC_FF \times V_{DAC} \times T_{CLK} \\ &\quad + (V_{DAC} \times T_{CLK}) \\ Analog\ Time &= T_{CLK} \times T \end{aligned}$$

Area Overhead and Test Time Using the Test Scheme. The area overhead is calculated in terms of the digital and analog test hardware and the number of test pins. The test scheme has 2 test pins $\{TM_1, TM_2\}$ and *bit flipping* hardware. Let V_{ATPG}, V_{ramp}, and T be as before. V_{ANALOG} is the number of test vectors for testing the analog component. The test vectors are repeated for T cycles in order to average out the noise. The total test time is the same as Equation 5, but the individual times change:

$$\begin{aligned} Digital\ Core\ Time &= V_{ATPG} \times T_{CLK} \times T \\ DAC\ Time &= V_{ramp} \times T_{CLK} \times T \\ Analog\ Time &= V_{ANALOG} \times T_{CLK} \times T \end{aligned}$$

Comparison. We compare the area overhead of the new scheme with the IEEE 1149.4 mixed-signal test standard in Table 5 for five SoCs. Columns 2 and 3 give the test hardware in logic gates for the 1149.4 standard and for the new scheme. T_{CLK} has a 2 *ns* period (1/500 *MHz*). Table 6 shows the test times for both the 1149.4 standard and the new scheme for T = 100 test cycles. The new scheme test time is 74% less than for the 1149.4 standard.

Table 5: Test hardware saved using the new scheme

SoC No.	Test Hardware (# of logic gates)		% Saved
	IEEE 1149.4	New Scheme	
1	4698	1004	79
2	5024	1013	80
3	4496	1004	78
4	4496	1004	78
5	4496	1004	78

Table 6: Test time comparison

SoC No.	Test Time (*ms*)		Reduction %
	IEEE 1149.4	New Scheme	
1	5.1	1.1	79
2	1.0	0.34	66
3	2.3	0.62	73
4	2.3	0.62	73
5	2.3	0.47	79

The IEEE 1149.4 standard *analog test bus* is slow, and usually has less than 1 *MHz* bandwidth [3]. Components are usually tested with 10 $KHz <= F <= 100\ KHz$ [10]. For the new scheme the frequency of the test tones depends on the digital test clock period (T_{CLK}) and the number of test tone patterns P. If $T_{CLK} = 2\ ns$, then the sampling frequency is $F_S = 500\ MHz$. Let $P = 100$. The generated tone frequency is:

$$F = \frac{F_S}{P} = \frac{500 \times 10^6}{100} = 5\ MHz$$

Fig. 6 shows the generated test tone frequency as a function of T_{CLK} and P. The test tone frequency decreases as P increases, but we still have much greater bandwidth than the 1149.4 bus.

Table 7: Comparison of new method with method of Tofte *et al.*

# Test Sessions n	Low-pass Filter				Band-pass Filter			
	Noise (%)		THD (%)		Noise (%)		THD (%)	
	New Method	Tofte *et al.* [14]	New Method	Tofte *et al.* [14]	New Method	Tofte *et al.* [14]	New Method	Tofte *et al.* [14]
1	0.02695	0.1368	0.09076	0.1939	0.0464	0.0695	0.4508	0.4882
4	0.02170	0.0477	0.09075	0.1898	0.0185	0.0259	0.4435	0.4882
9	0.02042	0.0293	0.09073	0.1916	0.0167	0.0172	0.4787	0.4882
16	0.02023	0.0220	0.09018	0.1899	0.0114	0.0150	0.4103	0.4881
25	0.01996	0.0186	0.09067	0.1907	0.0098	0.0129	0.4118	0.4885
Best Result	0.01996	0.0186	0.09067	0.1907	0.0098	0.0129	0.4118	0.4885

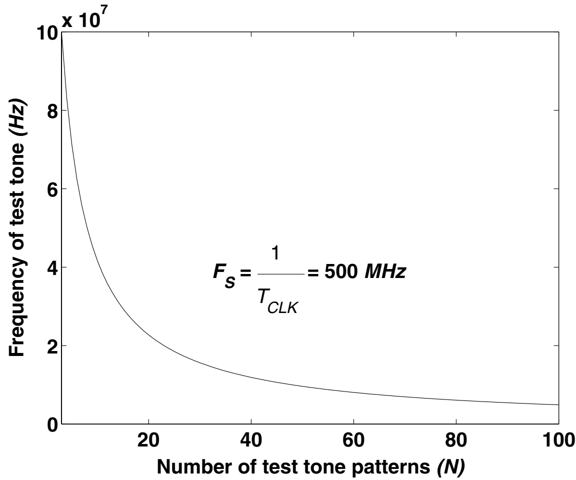

Figure 6: Test tone frequency generated using the new scheme

Comparison with BIST Distortion Testing. Table 7 compares the performance of our new method for noise testing and distortion testing with that of Tofte *et al.* [14]. In their method, they generate a test waveform using a *linear feedback shift register* (LFSR), convert it to an analog signal with a DAC, stimulate the analog core, loop back the analog output of the core, and digitize the looped-back signal with an *analog-to-digital converter* (ADC). In our method, instead we engage the DFT hardware to configure the tone generator, apply digital test vectors to the digital *pcont2* core, convert the *pcont2* outputs to an analog signal with an 8-bit DAC, and apply the DAC output to the analog circuit. They calculate cross-correlation between the LFSR signal and the digitized response of the analog core as a signature, whereas we use an ATE to examine the spectrum of the analog core directly. In both methods, the stimulus for LPF test was 33 cycles of a 20.625 KHz tone, and the stimulus for the *band-pass filter* (BPF) test was 206 cycles of a 128.75 KHz tone. For the experiments using our method, the tone amplitude was 1 V_{pp} (*Volts* peak-to-peak). Both results were produced by analog simulation. In their work, they inject white noise with the LFSR, whereas we inject good machine white noise of 2 $mVRMS$ into the DAC. We see that for the LPF, our method generated slightly higher noise than theirs, but our *total harmonic distortion* (THD) was significantly lower. THD is the ratio of the signal energy in the harmonics of the fundamental to the energy of the fundamental. For the BPF, we not only had lower noise, but also significantly lower THD. This is because their LFSR pattern generator inserts huge amounts of high amplitude white noise into the circuit, due to its random nature, whereas our tone generator is tailored to create the desired multi-tone waveform at the desired frequencies, with noise coming only from the DAC good machine noise. Thus, our method is superior.

Failure Effect Analysis. The proposed test scheme can diagnose down to the individual components of the SoC. Stage I of the test scheme determines whether the digital core, the DAC, the DFT hardware, or some combination of these is failing. The bit flipping hardware detects whether a digital core stuck-fault propagates to a DAC LSB. Stage II detects whether the analog core fails using the tones generated from the fault-free tone generator. So, we can diagnose down to each block in Figure 1, except that we cannot differentiate between faults in the reconstruction filter and faults in the analog block.

7 Conclusion

We proposed a new test architecture for mixed-signal SoCs where we use existing digital cores and DACs as an analog tone generator for either full or partial analog built-in self-test. The advantages are that it generates very precise multi-tone analog test waveforms, without the overhead of a custom, on-chip analog multi-tone waveform generator or a DSP. Also, analog multi-tone waveform generators are notoriously difficult to manufacture accurately in volume production, due to analog component variations. We showed that this architecture can test 100% of the testable digital core stuck-at faults by using an external ATE to examine the spectrum of the analog output signal, where the fault effects show up as magnitude or phase disturbances. The results for this new SoC test method show that the digital faults can be detected without using the 1149.4 *boundary scan hardware* by propagating the faults through the DAC and analog core, and then observing them in the analog domain.

On five SoCs, where the analog cores were either 1st-order low-pass filters or 1st-order $\Sigma - \Delta$ modulators, we covered 100% of the analog core faults, complete static linearity tests of

the DAC, and 100% of the testable digital core stuck-at faults. This method requires 78.6% less test hardware than the 1149.4 standard, while providing better fault detection. Only rarely do we require the *bit flipping* block. Analog components can now be tested using the tones generated from the tone generator block (consisting of a modified digital core and DAC) instead of using the *analog test bus* to provide the tones. The test time was reduced by 74% compared to using the 1149.4 analog test bus, due to the elimination of scan shifting sequences. This testing method has much higher bandwidth than the 1149.4 bus, and therefore can generate test tones of 15 MHz for a digitized waveform sample set of size 50.

Future Work. The bit flipping block in this method could be extended to detect whether any one of the output bits of the digital core flips due to a fault, not just the one or two LSBs. This use will increase the hardware overhead of the bit flipping block, since good machine outputs must now be provided to the DFT hardware for all bits during testing. We propose to use this information in the diagnosis of the digital core. Furthermore, we will use the concept of the on-chip pattern generation using the digital core and DAC in testing high-speed serial input/output interfaces (SERDES).

Acknowledgment. This material is based on work supported by the National Science Foundation under Grant Nos. 0098304 and 0429743. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] A. Bisaria, “Digital Spectral Test Generation for Mixed-Signal Circuit Testing,” Master’s thesis, Rutgers U., ECE Dept. May 2002.

[2] A. Bisaria and M. L. Bushnell, “Digital Spectral Test Generation for Mixed-Signal Circuit Testing,” in *Proc. of the North Atlantic Test Workshop*, May 2002, pp. 92–100.

[3] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits*. Norwell, Mass.: Kluwer Academic Publishers, 2000.

[4] B. Dufort and G. W. Roberts, “On-Chip Analog Signal Generation for Mixed-Signal Built-In Self-Test,” *IEEE J. of Solid-State Circuits*, vol. 34, no. 3, pp. 318–330, Mar. 1999.

[5] L. Fang and H. G. Kerkhoff, “Tolerance-Box Generation in Mixed-Signal SoC Testing,” in *Proc. of the ProRISC Conf.*, Nov. 2001, pp. 342 – 347.

[6] I. H. S. Hassan, K. Arabi, and B. Kaminska, “Testing Digital to Analog Converters Based on Oscillation-Test Strategy Using Sigma-Delta Modulation,” in *Proc. of the Int’l. Conf. on Computer Design*, 1998, pp. 40–47.

[7] J.-L. Huang, C.-K. Ong, and K.-T. Cheng, “A BIST Scheme for On-Chip ADC and DAC Testing,” in *Proc. of the Design, Automation and Test in Europe Conf.*, Mar. 2000, pp. 216–220.

[8] M. Mahoney, *DSP-Based Testing of Analog and Mixed-Signal Circuits*. Los Alamitos, CA: IEEE Computer Society Press, 1987.

[9] C.-K. Ong, K.-T. Cheng, and L.-C. Wang, “Delta-Sigma Modulator Based Mixed-Signal BIST Architecture for SoC,” in *Proc. of the Design Automation Conf.*, Jan. 2003, pp. 669 – 674.

[10] A. Osseiran, editor, *Analog and Mixed-Signal Boundary-Scan A Guide to the IEEE 1149.4 Test Standard*. Boston: Kluwer Academic Publishers, 1999.

[11] S. Ozev, I. Bayraktaroglu, and A. Orailoglu, “Test Synthesis for Mixed-Signal SoC Paths,” in *Proc. of the Design Automation and Test in Europe Conf.*, Mar. 2000, pp. 128–133.

[12] S. Ozev and A. Orailoglu, “Path-Based Test Composition for Mixed-Signal SoCs,” in *Proc. of the Int’l. Test Conf.*, Feb. 2000, pp. 153–158.

[13] A. Sehgal, S. Ozev, and K. Chakrabarty, “TAM Optimization for Mixed-Signal SoCs using Analog Test Wrappers,” in *Proc. of the Int’l. Conf. on Computer-Aided Design*, Nov. 2003, pp. 95–99.

[14] J. A. Tofte, C. K. Ong, J. L. Huang, and K.-T. Cheng, “Characterization of a Pseudo-Random Testing Technique for Analog and Mixed-Signal Built-in Self-Test,” in *Proc. of the VLSI Test Symp.*, Apr. 2000, pp. 237–246.