Top-Down, Constraint-Driven Design Methodology Based Generation of a Second Order Σ-Δ A/D Converter

Henry Chang Eric Felt Alberto Sangiovanni-Vincentelli

Electronics Research Laboratory
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA 94720, U.S.A.

Abstract

To accelerate the design cycle for analog circuits and mixed-signal systems, we have proposed a top-down, constraint-driven design methodology [1]. In this paper we present a complete design flow to illustrate this design methodology as it applies to the design of a second order sigma-delta (Σ-Δ) analog-to-digital (A/D) converter. We start from its performance and functional specifications and ending with the testing of the *fabricated parts*. Experimental results are presented.

1 Introduction

As the level of system integration increases, one apparent bottleneck is the design of the analog circuit components. Though these components are often small in area, their design times can be high. To address this problem we have proposed a top-down, constraint-driven design methodology [1], and, with it, have provided a variety of design tools such as behavioral level simulators, optimization tools, and physical layout tools. To show the effectiveness of this methodology, we have undertaken design examples beginning with industrial-strength specifications, and not ending until the designs have been fabricated and tested. Previously presented has been the complete design flow for a class of interpolative switched current source D/A converters [2]. This paper presents a complete design flow of a second order (Σ-Δ) A/D converter. The two examples differ for the following reasons: (1) the subcomponents are different, (2) the performances and the reasons for their degradation are different, and (3) the sub-blocks are not in regular arrays. The key point, however, is that the methodology is applied in exactly the *same* way.

Presented in this paper are: (1) the application of the methodology to the design synthesis phase (performance specifications to sized schematics), (2) the physical synthesis phase, and (3) the exerimental results. Conclusions and a breakdown of the design times are also be presented.

2 Design Specifications

Table 1 lists the specifications for the designed and fabricated chip. The architecture of the converter is shown in Figure 1. It is a second order system with a 1-bit quantizer similar to the one found in [3]. The input anti-aliasing filter and the decimation filter to convert the bit-stream output to data words is done off chip.

3 Synthesis Path

The two-level hierarchy used for synthesis is shown in Figure 2. During high-level synthesis, specifications for the Σ-Δ A/D are mapped into **intermediate-level parameters** which become the specifications for the integrator, comparator, D/A, digital control logic, and routing. The low-level synthesis phase uses these specifications as **constraints** to design the sub-blocks.

Type	Specifications	Value
Performance	Min. Signal-to-Noise (SNR) ratio	74 dB
	Nyquist Frequency (f_x)	250 kHz
Operation	Supply voltage	5 V
Technology	Design rules	scmos
	SPICE parameters	HP CMOS26B
Layout	Input/output terminal locations, etc.	
Optimization	Minimize area	

Table 1: Σ-Δ A/D Design Specifications

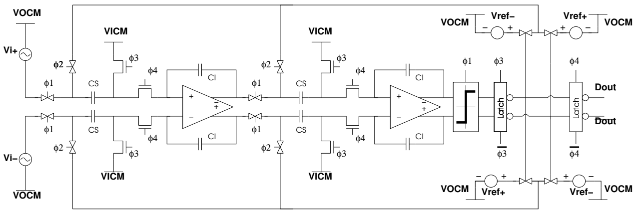

Figure 1: Σ-Δ A/D Architecture

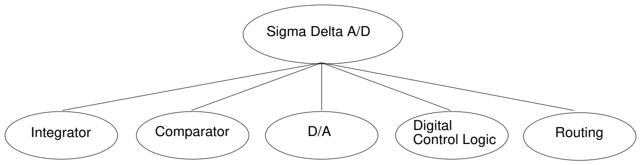

Figure 2: Σ-Δ Hierarchy

3.1 High-Level Synthesis

The intermediate-level parameters (the variables) for the high-level synthesis phase are shown in Table 2. Given values for each of these parameters, along with f_x from Table 1, the SNR for the Σ-Δ A/D can be computed using a behavioral model developed for this particular system and the behavioral simulator MIDAS [3].

Since the behavioral simulator is low-level-implementation *independent*, area as a function of the parameters cannot be determined. Since we cannot minimize area directly, we choose to **maximize design flexibility** during this optimization step. We build flexibility functions of the following form for each of the parameters:

$$flex_P(x_p) = -ax_p^2 - \frac{b}{x_p} + c$$

Parameter	Description
M	Oversampling ratio
V_{FS}	Full scale voltage of Σ-Δ system
O_{off}	Offset due to the OTA
O_{range}	Output range of the OTA relative to V_{FS}
O_{th}	Thermal noise due to the OTA
$O_{1/f}$	1/f noise due to the OTA
O_{gain}	Open loop gain of the OTA
I_{τ}	Time constant for the integrator
I_{SR}	Slew rate of the integrator
$I_{kT/C}$	kT/C noise of the integrator
I_{Amis}	Integrator gain mismatch
I_{gain}	Integrator gain
C_{off}	Offset due to the comparator
$C_{hysteresis}$	Hysteresis due to the comparator
C_{noise}	RMS noise due to the comparator
D_{off}	Offset in the D/A
D_{noise}	Noise contribution for the D/A
ΔSNR	Non-characterized degradations to SNR

Table 2: Parameters for the Σ-Δ A/D Block

where x_p is the value for the parameter. If low values for the parameters are more difficult to obtain than high values, $a = 0$, e.g. thermal noise. If the reverse is true, $b = 0$, e.g. OTA gain. [2] contains a general description. Two points are required to find the remaining two constants. One corresponds to a "moderate" difficulty design point, and the second to a "hard" design point. These are heuristically chosen by a designer.

Because of this heuristic nature, overall optimality is lost for the design. However, if a solution is found, *feasibility* is insured. If the design is found to be too suboptimal when completed, the flexibility functions can be rebuilt to reflect data from previous design passes for future iterations. [4] attempts to *a priori* build extremely accurate contours (flexibility functions) to also guarantee an optimal result. Unfortunately, this is extremely expensive, since not only does the entire design space need to be simulated, but also all possible combinations of architectures must be explored *before* any designs can be obtained. Our method has the advantage that a design solution is *immediately* found. A feasible solution exists at the end of all iterations.

Let x_i be the i^{th} parameter ($x_1 \equiv M$, $x_2 \equiv V_{FS}$, $\ldots$ $x_n \equiv \Delta SNR$), n be the number of parameters (n=18), and c, be a vector containing the additional parameters from Table 1. The optimization problem[1] is

$$\begin{array}{lll} \max & \sum_{i=1}^{n} flex_i(x_i) & (1) \\ \text{s.t.} & \text{SNR}(x_1, x_2, \ldots x_n, c) \geq 74dB & \\ & l_i \leq x_i \leq u_i & \forall i = 1 \ldots m \end{array}$$

u_i and l_i represent the upper and lower bounds on the parameters, x_i.

Normally, an "off-the-shelf" **optimization program** such as MINOS [5] is used to solve this problem. However, when functions are computed using iterative techniques (simulators), because of numerical noise, general purpose optimizers tend to fail. This was true for this problem. The principal difficulties were (1) when computing SNR, MIDAS has a large standard error, $\epsilon \approx 2$ dB, and (2) SNR is extremely nonlinear with respect to its input variables. A typical plot of SNR versus an input variable, in this case O_{range}, is shown in Figure 3. Because of the unsmooth surface, MINOS would either get trapped in false local minima or iterate excessively

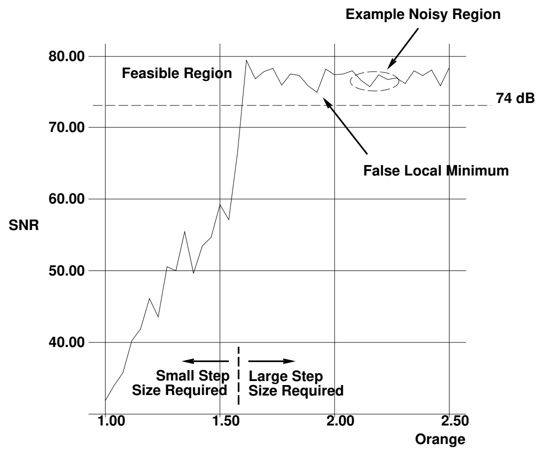

Figure 3: SNR versus O_{range}

while it reduced the step size, believing a noisy region to be an extremely nonlinear region. The step size can be adjusted in MINOS, but because of the nonlinear nature, the appropriate step size was not chosen correctly when it switched regions.

Averaging multiple simulations with a varying sine wave input, we reduced ϵ to approximately 0.5 dB. The following is the optimization algorithm that we used:

1. Start with feasible solution, x^*. Let $x_o = x^*$, $j = 1$.
2. Replace the SNR constraint function with a linearized SNR constraint around x^*.

$$\text{SNR}(x) = \text{SNR}(x^*) + \left[\nabla \text{SNR}(x)\right]^T_{x=x^*} (x - x^*)$$

3. Impose bounds

$$(1 - d_i)x_i^* \leq x_i \leq (1 + d_i)x_i^* \quad \forall\, i = 1...m$$

   where d_i is a preset value set by the user (typically 0.1) depending on how nonlinear SNR is.
4. Solve the problem using MINOS. Let the solution be x'_j.
5. For each i, if $x'_{i,j}$ is at the upper bound and $x'_{i,j-1}$ was at the lower bound, or if $x'_{i,j}$ is at the lower bound and $x'_{i,j-1}$ was at the upper bound, then reduce d_i by a factor of two.
6. If $\|x_j - x_{j-1}\| \leq \epsilon$, exit with x'_j as the solution.
7. Set $j = j + 1$. Goto 2.

The derivatives in Step 2 are computed using finite differences. However, because of ϵ, a derivative was only considered valid when the finite difference caused a 1 dB change (with a certain tolerance) in the SNR. Binary search was used to find this point. 1 dB was chosen, because Δx_i should be small, but the change in SNR has to be greater than ϵ. The idea behind this approach is to dynamically control the step size, using the performance change as the measure of whether or not the step size is valid. This allows the optimizer to see the overall shape of the curve rather than the shape of the noise. The solution can be found as constraints in Equation 2 and Section 3.2.2.

3.2 Low-Level Synthesis

The solutions from the high-level optimization problem are now **constraints** during the low-level synthesis phase. The design for the integrator and comparator are described as examples. The D/A and digital circuitry were designed in a similar manner.

[1] Less critical parameters were chosen *a priori*.

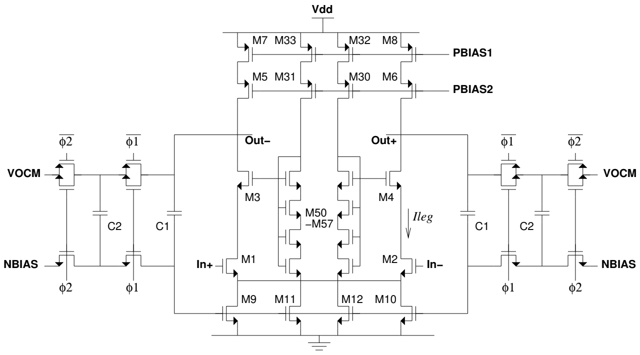

Figure 4: Schematic of the OTA

3.2.1 Integrator/OTA

The integrators are shown in Figure 1. Bottom plate sampling is employed to reduce the charge injection from the switches into the signal. A telescopic or unfolded cascode operational transconductance amplifier (OTA) [6][7] (shown in Figure 4) was selected. A replica bias circuit provides the reference and bias voltages for the system components.

As in the high-level optimization problem, values for the non-critical parameters were chosen *a priori* to simplify the problem, e.g. current ratio between M_{11} to M_9, ratio between C_1 and C_2 of CMFB circuit, peak leakage current in bias circuit, safety margin for V_{ds}. Seven variables remain to describe the integrator. They are listed is Table 3. This set is sufficient to generate all component

Parameter	Description
L_{nmos}	Length of nmos transistors in gain stage of OTA
L_{pmos}	Length of pmos transistors in gain stage of OTA
L_{bias}	Length of bias transistors in OTA and bias circuit
L_{M44}	Length of transistor M_{44} of the bias circuit (not shown)
$V_{gs}-V_T$	$V_{gs}-V_T$ voltage for the transistors in the gain stage
I_{leg}	Current flowing through the transistors in the gain stage
C_S	Size of the sampling capacitor

Table 3: Design Parameters for the Integrator

transistor and capacitor values, and is sufficient to characterize each of the design constraints. In keeping with the Σ-Δ A/D design specifications, the **objective** during this phase is to minimize the active area of the integrator.

The problem was solved in two steps, because of the integer nature of the L's. These were determined first. L_{nmos}, L_{pmos}, and L_{bias} were minimized subject to the constraint on gain and on common-mode rejection ratio (CMRR) issues. L_{nmos} needs to be as small as possible to maximize the speed of the OTA. Enumerating the L's for L_{nmos} and L_{pmos} for $L = 0.8\mu$m, 1.3μm, 1.8μm, 2.3μm, it was determined, in order to satisfy the gain requirements with allowance for sufficient V_{ds}, that $L_{nmos} = 1.3\mu$m, and $L_{pmos} = 1.8\mu$m. L_{bias} was chosen to have a transistor length of 1.8 μm based on CMRR issues, and L_{M44} was determined to be 2.3 μm.

Equation 2 shows the **optimization problem** for the integrator. All performances are evaluated analytically resulting in a problem easily solved by standard optimization techniques. MINOS solved the problem in less than 1 CPU second. $V_{gs}-V_T$= 330 mV; I_{leg}= 1.22 mA; C_S= 207 fF.

Using the values determined for ($V_{gs}-V_T$), I_{leg}, C_S, gate lengths, and some of the design constants, the component values for

$$
\begin{array}{ll}
\text{min} & area(V_{gs}-V_T, I_{leg}, C_S) \\
\text{s.t.} & O_{off}(V_{gs}-V_T, I_{leg}, C_S) \leq 25\ mV \\
& O_{range}(V_{gs}-V_T, I_{leg}, C_S) \geq 2.0 \\
& O_{th}(V_{gs}-V_T, I_{leg}, C_S) \leq 20.0\ \mu V_{rms} \\
& O_{1/f}(V_{gs}-V_T, I_{leg}, C_S) \leq 1.05\ mV_{rms} \\
& O_{gain}(V_{gs}-V_T, I_{leg}, C_S) \geq 250\ (V/V) \\
& I_{\tau}(V_{gs}-V_T, I_{leg}, C_S) \leq 1.5\ ns \\
& I_{SR}(V_{gs}-V_T, I_{leg}, C_S) \geq 1045\ V/\mu s \\
& I_{kT/C}(V_{gs}-V_T, I_{leg}, C_S) \leq 20\ \mu V_{rms} \\
& O_{\tau,CMFB}(V_{gs}-V_T, I_{leg}, C_S) \leq 2.0\ ns \\
& V_{gs}-V_T \geq 100\ mV \\
& I_{leg} \geq 100\ \mu A \\
& C_S \geq 100\ fF
\end{array}
\quad (2)
$$

the OTA, integrator, and bias circuitry are computed using analytic equations.

3.2.2 Comparator

The comparator used in [3] was used in this design. The set of **constraints** required for the design of the comparator were $C_{off}\leq$ 100 mV, $C_{hysteresis}\leq$ 100 mV, and $C_{noise}\leq$ 9.35 mV_{rms}. Since the design constraints are fairly lax for this comparator, device sizes similar to the switch sizes were chosen for the design. These sizes were verified with SPICE and by analytic equations to meet the design constraints.

3.3 Physical Design

Physical assembly is also performed hierarchically. The leaf (transistor-level) cells, e.g., OTA, comparator, are generated first. Most of the layout was automatically synthesized with OCT-TOOLS [8][1] using the following steps for each leaf cell:

1. Enter the initial schematic in the form of a SPICE deck.
2. Add analog constraints to the SPICE deck either automatically, using PARCAR, or manually. The extra lines added: (i) indicate transistor matching, (ii) indicate transistor symmetry, and (iii) split transistors into smaller parallel transistors for decreasing drain capacitance and for canceling linear gradient effects.
3. Generate the **transistor stacks** using MKSTACK [9]. If transistors have the same gate width and share diffusion contacts, they are combined to form stacks of parallel transistors.
4. Generate the capacitors using an automatic layout tool.
5. Implement the nets of the parallel transistors which are in stacks using another automatic layout tool. This routing is highly organized.
6. Instantiate objects in the layout and connect the nets symbolically using BDNET.
7. Generate the layout placement with the simulated annealing tool, PUPPY-A.
8. Put down I/O vias for the leaf cell using PADPLACE.
9. Route the leaf cell using the area router ROAD.
10. Compact the leaf cell using SPARCS-A.

Manual layout was required for two phase clock generator, because of its highly organized nature. The chip-level routing and compacting was done automatically.

4 Extraction and Verification

We used hierarchical simulation to verify circuit *performance*, and we used non-hierarchical simulations to verify circuit *functionality* (nominal behavior). In the hierarchical method, each sub-block was extracted. Using SPICE simulations, each sub-block's performances were determined. These performances were fed into MIDAS which returned the SNR for the Σ-Δ A/D. This verified the performance.

In the non-hierarchical method, we extracted the entire circuit, and performed pad-to-pad SPICE transient simulations for a number of clock periods, given a fixed amplitude sine wave at the input. The bit stream result was fed into MIDAS which decimated the output and returned an SNR value, which was greater than the one returned from the hierarchical simulations. This result was expected, since there were no noise sources in the flat SPICE simulation. The higher SNR value verified the circuit's functionality.

5 Experimental Results

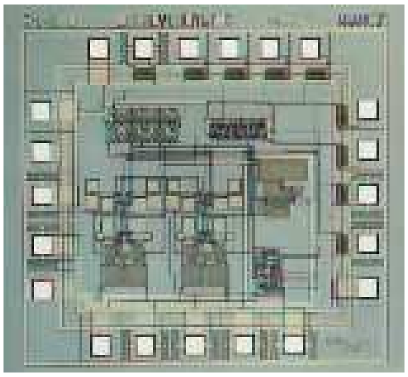

Figure 5: Σ-Δ A/D Converter Die Photo

This chip was fabricated on a MOSIS 0.8 μm process (HP CMOS26B). A die photo is shown in Figure 5. The active area (chip without pads) is 1.1 mm^2. The total area is 3.1 mm^2. Using a filtered sine wave as input, a logic analyzer was used to capture the bit stream output from the Σ-Δ A/D converter. This data was then transferred into MIDAS for decimation filtering and SNR calculation. The minimum clock period at which the chips function without error is 30 ns (33.33 MHz), which corresponds to a Nyquist rate (f_x) of 167 kHz for the input signal. The distortion analysis routine in MIDAS computes a peak SNR of 70.0 dB (slightly over 11 bits). An FFT of the output corresponding to a 100 kHz input sinusoid is shown in Figure 6; integration of the noise floor in the FFT yields an SNR of 71.4 dB. The design specifications called for a minimum f_x of 250 kHz and a minimum SNR of 74 dB, so the measured performance is close to but does not quite meet the specifications.

6 Conclusion

Following the top-down, constraint-driven paradigm, a complete design cycle for a Σ-Δ A/D has been performed. One of the critical metrics for this methodology is how long designs take. The total elapsed design time for this project, including the design synthesis and physical synthesis phases, was approximately two months. Table 4 shows a breakdown. These times are intended to give an accurate account of the length of this project from its inception to "taping-out" to illustrate the *real time required* for design when using this methodology.

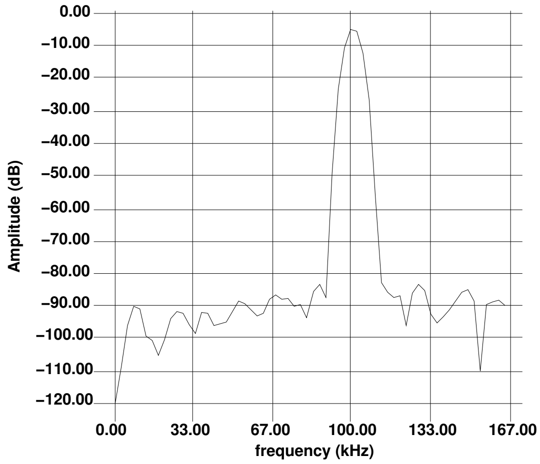

Figure 6: 128-Point FFT of Decimated Σ-Δ Output

Future work for the Σ-Δ A/D consists of further tests and the development of a testing methodology which will attempt to speed up the testing process. One possible approach is to find an optimal set of sine wave input amplitudes and frequencies in order to quickly characterize SNR.

Design Phase	Time Required
Behavioral model development	4 days
High-level optimization	15 days
Low-level synthesis	5 days
Layout generation	10 days
Extraction/Verification	5 days

Table 4: Design Times

7 Acknowledgements

This research has been partially supported by The National Science Foundation, The Intel Foundation, SRC (93-DC-008), ARPA, and the Armed Forces Communications and Electronics Association.

References

[1] H. Chang, A. Sangiovanni-Vincentelli, F. Balarin, E. Charbon, U. Choudhury, G. Jusuf, E. Liu, E. Malavasi, R. Neff and P. Gray, "A Top-down, Constraint-Driven Design Methodology for Analog Integrated Circuits," in *Proc. IEEE CICC*, pp. 841–846, May 1992.

[2] H. Chang, E. Liu, R. Neff, E. Felt, E. Malavasi, E. Charbon, A. Sangiovanni-Vincentelli and P. R. Gray, "Top-Down, Constraint-Driven Methodology Based Generation of n-bit Interpolative Current Source D/A Converters," in *Proc. IEEE CICC*, pp. 369–372, May 1994.

[3] B. E. Boser, *Design and Implementation of Oversampled Analog-to-Digital Converters*, Ph.D. Thesis, Integrated Circuits Laboratory, Stanford Univeristy, Stanford, Oct 1988.

[4] J. Shao and R. Harjani, "Macromodeling of Analog Circuits for Hierarchical Circuit Design," in *Proc. IEEE ICCAD*, pp. 656–663, November 1994.

[5] "Modular In-core Nonlinear Optimization System (MINOS 5.3)", Systems Optimization Laboratory, Department of Operations Research, Stanford University.

[6] K. A. Nishimura, *Optimum Partitioning of Analog and Digital Circuitry in Mixed-Signal Circuits for Signal Processing*, Technical report, Ph.D. Thesis, University of California at Berkeley, July 1993.

[7] R. Castello and P. Gray, "A High-Performance Micropower Switched-Capacitor Filer," *IEEE Journal of Solid State Circuits*, vol. SC-20, pp. 1122–1132, December 1985.

[8] D. S. Harrison, A. R. Newton, R. L. Spickelmier and T. J. Barnes, "Electronic CAD Frameworks," *Proc. of the IEEE*, vol. 78, n. 2, pp. 393–417, February 1990.

[9] E. Charbon, E. Malavasi, D. Pandini and A. Sangiovanni-Vincentelli, "Imposing Tight Specifications on Analog IC's through Simultaneous Placement and Module Optimization," in *Proc. IEEE CICC*, pp. 525–528, May 1994.