Methodology of Virtual Testing of Trimmable Analog Circuits

Udo Sobe*, Enno Böhme* and Karl-Heinz Rooch†
*ZMD AG, Grenzstraße 28, 01109 Dresden, Germany
{Udo.Sobe, Enno.Boehme}@zmdi.com
†Fraunhofer IIS, Zeunerstraße 38, 01069 Dresden, Germany
Karl-Heinz.Rooch@eas.iis.fraunhofer.de

Abstract—This paper presents a design for test (DFT) methodology focused on high precision analog circuits using trimming methodology. A simulation technique called trimming analysis was derived from the production test process and is used for simulation-based verification of the circuit performances including trimming network and trimming algorithm. The virtual trimming by joint simulation of the test procedure and circuit allows to improve all parts and to maximize yield. The application of the analysis is shown for two different examples.

Index Terms—Virtual Testing, Precision Analog Circuits, Reference Blocks, Verification, Trimming Analysis

I. INTRODUCTION

"Time is money"—trimming is test time. Therefore the right trimming methodology is important for efficiency. Using conventional ways the trimming methodology cannot be checked completely during circuit design. This is possible late in the development cycle during test of silicon. The use of simulation-based verification methodology for virtual testing allows to consider all three parts: basic analog building blocks including trim network and trim algorithm together during circuit design. So, yield can be estimated early in the development process and allows the optimization of the entire trimming procedure.

The digital domain is the paradigm for virtual testing, as in [1], [2]. Virtual trimming as subset of virtual testing of analog circuits is poorly supported by current EDA tools. Trimming methodology is state of the art for precision analog circuits [3], [4], [5]. Precision analog circuits are e.g. on-chip references or (sensor) signal conditioning blocks. References are used to provide high absolute accuracy of the reference value like voltage, current or frequency. Signal conditioning needs adaptation of blocks which have inherent tolerances like adaption of sensor and circuit. This accuracy is given by a specification in terms of a lower (LSL) and an upper (USL) specification limit for the performance parameter (see Fig. 1a).

However this can be difficult to achieve in ICs as the absolute value of device parameters is usually poorly controlled in production.

Today the challenge for analog designers is in the ambitious goals of precision analog performance parameters and by contrast the increasing variation of device parameters for decreasing device geometries of modern technologies. By using a suitable circuit topology and optimising of its performance parameters it is possible to achieve a certain level of accuracy, e.g. for current reference typically ±20% over automotive application conditions. Beyond that the accuracy can be improved by using an electrical trimming network.

Yield is limited by a performance parameter with a large variance compared to the specification limits as shown in Fig. 1a. The yield can be improved by adding a degree of freedom allowing to shift the performance parameter so that it meets the specification limits after the manufacturing process (cf. Fig. 1b). This shifting is called trimming and can be realized by a special electrical network for trimming, whose design is not covered in this article. Additional digital circuitry is employed to trim the given parameter and to store the appropriate trim code in a non-volatile memory on chip.

Our paper addresses the systematic verification as part of DFT inside the design flow for basic analog building blocks including trim network and trim algorithm together during circuit design. The following section describes our trimming methodology, which was derived from the production test
process. Next, the simulation methods to verify all three parts together during the design process are compared.

II. VERIFICATION INCORPORATING TRIMMING

A. Trimming and verification simulation flow

In [6] we introduced a methodology to support trimming in circuit design which reflects the trimming and verification as done in manufacturing. The manufacturing process uses the flow shown in Fig. 2a. During IC production, the device parameters vary randomly within the process specification range given. Hence, certain performance parameters of every device are subject to variations as well. The following step at production test uses a trimming algorithm to reach the specification by testing. To trim the performance parameter under consideration, it is measured, an appropriate trim value determined and written to a non-volatile memory on the IC.

Afterwards, the parameter is measured again in a verification step to check whether the trimming was successful. Usually the trimming procedure is carried out at typical operating conditions. The verification may be repeated at other operating conditions to improve verification coverage.

The currently common verification flow used in circuit design was extended to adjust circuit performance parameters according to process conditions by an additional trimming algorithm step. The grey box in Fig. 2b shows the extension. So the procedure in the production process and circuit design is quite similar now (cf. Fig. 2a and b).

The outer loop sets the process conditions used for all simulations in the following trim and verification steps. Then the trim algorithm is applied to determine a suitable trim code to shift the performance parameter into the specified range. This trim code is then used in the verification step covering the entire operating condition range.

Fig. 3 shows some details of the implementation. The figure is separated into three main stages: process condition, trimming and verification. Each stage uses a specialized algorithm to control the setting of the process conditions, the trimming strategy. The simulation time $t_{TS}$ for $I$ process iterations can be estimated by the following equation:

$$t_{TS} = \sum_{i=1}^{N_{trimcodes}} t_{sim} + N_{verification} I \cdot \left(t_{sim} + \frac{I+1}{2} \cdot t_{skip}\right)$$

with $N_{trimcodes}$ average number of trim iterations required to find the trim code, $t_{sim}$ simulation time, $t_{skip}$ time to skip one Monte Carlo iteration (0 for corner simulations), $N_{verification}$ number of verification simulations, $N_{trimstep}$ number of trim steps.

The trimming stage contains two loops. The inner loop realizes the trimming of the parameter under consideration. This procedure forms a single trim step. A sequential trimming of different performance parameters is supported by the outer loop which controls the trim step sequence. The performance parameter, target value, trim register and algorithm can be defined for every trim step separately. This approach assumes that basic circuit and trim networks for successive trim steps are independent.

We added a trimming analysis feature to our in-house verification environment called zmdAnalyser because common verification environments can only manage the simulations over process conditions and operating range. This tool extends the Analog Design Environment (ADE) from Cadence. Some details are given in the next sections.

The application of the trimming analysis inside the design flow is shown in Fig. 4. The trimming analysis is an important verification step which allows to optimize basic circuit, trimming network and algorithm together during the iterative design process.
Table 1
COMPARISON OF CORNER ALGORITHMS. LEGEND: X APPLICABLE, — NOT APPLICABLE

<table>
<thead>
<tr>
<th></th>
<th>Predefined by FAB</th>
<th>Combination of device corner groups</th>
<th>Combining device corners using a correlation matrix</th>
<th>Single device corner variation</th>
</tr>
</thead>
<tbody>
<tr>
<td>goal</td>
<td>selected simulation at extreme conditions</td>
<td>comprehensive simulation at every combination of device group corners</td>
<td>technology oriented simulation at realistic device corner conditions</td>
<td>influence simulation if one single device is set at corner condition</td>
</tr>
<tr>
<td>corner model granularity</td>
<td>grouped</td>
<td>grouped</td>
<td>—</td>
<td>X</td>
</tr>
<tr>
<td>algorithm details</td>
<td>—</td>
<td>1) skip device groups which are not used inside the circuit 2) combine the corners of every used device corner group to corner sets</td>
<td>1) skip devices which are not used inside the circuit 2) combine the device corners of every used device to corner sets 3) calculate plausibility of every corner set 4) filtering of the corner sets (plausibility and count)</td>
<td>1) skip devices which are not used inside the circuit 2) set typical device corner for every device to typical corner set 3) build corner sets by variation of one single device corner basing on typical corner set</td>
</tr>
<tr>
<td>reference to used devices inside the circuit</td>
<td>—</td>
<td>—</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>considers correlation between devices</td>
<td>X</td>
<td>—</td>
<td>X</td>
<td>—</td>
</tr>
</tbody>
</table>

B. Methods to set process conditions

Monte Carlo (MC) and Corner analyses are standard methods to verify performance of circuits. Both analyses are also supported for trimming analysis. Combination of results of both analyses allows to give robust answers regarding the system to be trimmed.

Corner Analysis: The circuit performance at extreme technology conditions can be checked by corner analysis.

The capability of models to configure technology conditions (corners) of different devices to a corner set decides the usability of this method to simulate precision circuits. Many foundry PDKs (process development kits) use sections for grouping device models to grouped device corners. In general, grouped device corners do not support setting different corners for devices of one group. This simplifies the PDK and the setup for the user.

But sometimes devices are grouped although they have little or no correlation, e.g. different types of resistors. For instance, temperature compensated resistor structures can often be found inside precision analog circuits. They contain resistors with different temperature coefficients. A simulation using grouped device corners would suppress relevant scenarios (cf. Fig. 11).

The algorithm provided to define corner sets to simulate technology conditions depends on model capability (see Table 1). Combining grouped device corners is state of the art today. Grouped device corners which contain only unused devices can be skipped.

An interesting algorithm to generate corner sets with single device corners is the correlation-matrix based corner set generation. Every possible combination is evaluated by using the correlation matrix. That allows to filter out impossible combinations like device A with thin oxide paired with device B with thick oxide (see Fig. 5). This algorithm saves simulation effort and avoids overly optimistic (or overly conservative) designs.

Since local mismatch cannot be simulated, corner simulation is not sufficient to analyze mismatch related parameters. However, extreme conditions are better tested than with MC Analysis using Gaussian distributed process models.

Figure 4. Trimming analysis in the design flow of precision circuits.
Monte Carlo Analysis: Global and local variations (called Process and Mismatch in CDN) are considered, which allows yield estimation by MC Analysis. Hence mismatch between devices which may generate offset can be simulated. Results show distribution of the performance parameter around the mean value.

In general the count of samples is high around the mean value and is decreasing clearly with the distance from the mean (e.g. cf. density function of a Gaussian distribution in Fig. 1). If the number of runs is small, the number of runs with each trimming code is small as well resulting in limited statistical information for these trim values (see section III-A, Fig. 8 for an example). Hence the trim codes close to the mean value, which have high probability, are well verified, while the simulation effort is very high to check extreme technological corners due to the low probability of these process conditions (at least if the main influence parameter is Gaussian distributed). In this case it is advisable to use corner analysis.

In our implementation we changed the standard CDN MC simulation flow from a single simulation job into several jobs to simulate different trim codes and operating conditions at one specific iteration [6].

C. Trim algorithm

The trim algorithms can be developed as iterative procedure or as trim table as well as a combination of both. Our implementation supports the following default algorithms which covers most of our applications

- trimming table,
- iterative method to minimize error between target value and performance parameter,
- iterative method to reach a target range.

User defined methods are also supported.

The algorithm using a trim table is based on a look-up table (LUT). It contains intervals of nominal performance parameter values and trim codes to shift the performance parameter into the specification range. Usually a trim table is designed manually e.g. by using simple linear models of the circuit. This method is efficient for approximately linear dependency between performance parameter and trim code.

An iterative method tries to achieve a defined target. Approximation techniques like newton method are suitable for trim algorithm. So it is possible to optimize trim network with linear as well as non-linear dependency between performance parameter and trim code.

User defined methods are necessary for a few special applications. It is possible to use the combination of the above described algorithms or new methods. One example is a method with pre-selection of the trim code basing on simulated PCM data and following improvement of the trim code by using an iterative method.

D. Verification strategy

During circuit design all performance parameters of the circuit have to be checked over the so-called PVT corners:

- process conditions (P) and
- operating range (Voltage, Temperature).

The verification over operating range is carried out for a trimmed circuit at the current process condition. The implemented verification strategy bases on combination of operating parameters, which is also known as parametric sweep method. There are other strategies for verification. For example, the Worst-Case Operation algorithm inside MunEDA’s tool WiCkeD determines the worst-case operation conditions for a given performance and specification.

III. APPLICATION EXAMPLES OF THE TRIMMING ANALYSIS

A. Standard Trimming Analysis

Fig. 6 shows the principle of a current reference block with trimming network. Target of the reference is 10 μA±5% for an industrial application with temperature range of -40°C–150°C.

![Figure 5. Corner sets generated by using combination of device corners (Automatic corner generation) with reference to correlation matrix. Only 16 of 1024 possible corner sets contain no technological discrepancies. Hence 1008 (1024-16) corner sets can be skipped because they have conflicts and are not plausible.](image)

![Figure 6. Current reference block with correction by adding or subtracting the current B \( I_{corr} \). The decoder adapts the trim code B of a standard trimming algorithm to the two bus signals SP and SN of the switch array.](image)
The reference current $I_{ref}$ is derived from base current $I_{base}$. The current can be corrected by the current $I_{corr}$ via the switch array during the trimming step. The switch array is controlled by the two bus signals $SP$ and $SN$. A simplified mathematical model is given by the following equation

$$I_{ref} = I_{base} \left( 1 + \frac{1}{\xi} (S_p - S_n) \right) \text{ with } I_{corr} = \frac{I_{base}}{\xi}$$  \hspace{1cm} (2)

$I_{corr}$ is derived from $I_{base}$ by scale factor $\xi$.

To optimize the trimmed circuit and to estimate the yield, a standard trim algorithm to minimize error of $I_{ref}$ was used for trimming analysis. The trim algorithm generates a trim code at $B$ with sign-magnitude representation. Therefore a decoder to adapt trim code at $B$ to the two bus signals $SP$ and $SN$ was added to the test bench. To support simulation with the Cadence Spectre simulator [7], the decoder, which was part of the digital design, was modeled in Verilog-A.

The mathematical description of the decoder is given by the equations:

$$S_n = b_{N-1} \cdot (1 + \sum_{i=0}^{N-2} 2^i b_i) \text{ with } b_i \in \{0, 1\}.$$  \hspace{1cm} (3)

$$S_p = (1 - b_{N-1}) \cdot \sum_{i=0}^{N-2} 2^i b_i \text{ with } b_i \in \{0, 1\}.$$  \hspace{1cm} (4)

The bus signal $B$ is represented by bits $b_i$. $N$ is the width of the bus signal. The most significant bit (MSB) $b_{N-1}$ contains the sign.

The ideal mathematical model of the circuit shows a lower absolute trimming range for small basis currents $I_{corr}$. The typical characteristics of untrimmed vs. trimmed performance parameter and the mathematical model are shown in Fig. 7.

Fig. 7 shows simulation results of the MC analysis (about 800 samples) of the realized circuit. The figure also shows a good match between ideal model and implementation. A lot of samples are concentrated around the mean value of the trimming code, which is common for MC simulations (cf. Fig. 8). Some trimming steps are not used for trimming. Therefore, corner analysis was carried out to verify extreme process conditions. Those additional results are also shown in Fig. 7. There is still enough margin for trimming. The trimming steps are clearly separated. Therefore the trimming can be carried out successfully.

Fig. 9 shows the trimming table derived from MC and corner simulations for this example which can be used for the production test program.

During circuit design the test bench was further completed by additional circuit blocks like bandgap references, power supply regulator and oscillator to check performance of critical circuit blocks.

B. Trimming of Multiple Parameters using Different Trimming Algorithms

This example is a temperature measurement circuit (cf. Fig. 10 for a block diagram). This application requires high accuracy (after trimming) in a small temperature range, with an extended (less accurate) operating range of 15°C to 45°C. Due to system requirements only a moderate ADC resolution is available for the temperature measurement. Therefore, trimming of both offset and gain is necessary to adapt the
temperature signal to the ADC input range. Finally, error correction is applied to the ADC values to achieve the required accuracy.

Since offset and gain depend both on global process conditions and on local mismatch, both corner and MC simulations were necessary to verify the entire trim range.

Because in this design the offset alone can overdrive the amplifier, the offset must be trimmed first. Trimming the bus CtrlOffset is used to minimize the offset.

Then the gain is trimmed starting with an estimate based on the sensor sensitivity at $V_{Sensor}$ using CtrlGain. This is done here because simulation of $V_{Sensor}$ is significantly faster than the simulation of the entire circuit. Then fine tuning of the gain is performed.

Fig. 11 shows simulation results of MC and corner analyses of the trimming error compared with an ideal linear model. The deviations of the corner points from the model is caused by the small nonlinearity of the implementation.

The example also shows the importance of using correct pairs of corner models to ensure sufficient worst-case trim range.

Fig. 12 shows the error after gain trimming. It can be seen the trim strategy works well to reduce the gain error to acceptable limits.

IV. CONCLUSIONS

A verification method, called trimming analysis, for precision analog circuits which employ trimming was presented. The analysis is part of a systematic DFT flow to jointly design and verify basic circuit, trim network and trim algorithm. Support of both single and multiple parameter trimming was demonstrated with examples.

The virtual trimming allows verification of the trimmed circuit over the entire range of operating conditions. Thus, the yield can be estimated by MC simulation and worst-case conditions can be checked using the corner simulation.

The results can be used to choose the right trimming algorithm for the test equipment. For example the trimming table can be derived by approximating results from MC and corner simulations.

ACKNOWLEDGMENT

The project described in this report was financed by the “Bundesministerium für Bildung und Forschung” (BMBF), grant 01M3178B. The authors are responsible for the contents of this publication.

REFERENCES