



BAG2-assisted analog layout synthesis for TSMC 65 nm and GPDK 45 nm

Matthew Bio · Wolfgang Scherr · A. S. Agbemenu · Santiago Martin Sondón · Vinayak Hande · Johannes Sturm

Received: 29 September 2023 / Accepted: 6 December 2023 / Published online: 31 January 2024
 © The Author(s) 2024

Abstract Analog circuit design automation remains an intense area of attention and has seen both new and existing tools continuing to be developed and targeting different phases of the analog design flow to reduce development time and cost. One of the promising tools is the Berkeley Analog Generator (BAG2) framework which is an open-source analog layout generator for automating and verifying circuit layouts. It promises a process-independent flow and it encourages design reuse due to using parameterized generators which can be scaled as required. This reduces the layout development time compared to manual handmade layouts. This work describes the effort and results of evaluating the BAG2 framework for the TSMC 65 nm and Cadence GPDK 45 nm processes. A case study is made with a number of circuits to discuss the problems in setting up and using BAG2 for the above technologies as well as the limitations and solutions required to utilize the framework effectively.

Keywords Berkeley Analog Generator · BAG2 · TSMC · 65 nm · GPDK · 45 nm · Differential amplifier · Common-mode feedback · CMFB · Python · Virtuoso

BAG2-unterstützte analoge Layout-Synthese für TSMC 65 nm und GPDK 45 nm

Zusammenfassung Die Automatisierung des analogen Schaltungsdesigns ist und bleibt ein wichtiger Forschungsbereich. Es wurden sowohl neue als auch bestehende Tools weiterentwickelt, die auf verschiedene Phasen des Designablaufs abzielen, um Entwicklungszeit und -kosten zu reduzieren. Eines der vielversprechenden Tools ist das Berkeley Analog Generator (BAG2)-Framework, ein Open Source Analog Layout-Generator zur Automatisierung und Verifikation von Schaltungslayouts. Es verspricht einen prozessunabhängigen Ablauf und fördert die Wiederverwendung von Designs durch die Verwendung parametrisierter Generatoren, die je nach Bedarf skaliert werden können. Dies reduziert die Layout-Entwicklungszeit im Vergleich zu manuellen, handgefertigten Layouts. Diese Arbeit beschreibt die Anwendung und die Ergebnisse der Evaluierung des BAG2-Frameworks für die Prozesse TSMC 65 nm und Cadence GPDK 45 nm. Es wird eine Fallstudie mit einer Reihe von Schaltkreisen erstellt, um die Probleme bei der Einrichtung und Verwendung von BAG2 für die oben genannten Technologien sowie die Einschränkungen und Lösungen zu diskutieren, die für eine effektive Nutzung des Frameworks erforderlich sind.

Schlüsselwörter Berkeley Analog Generator · BAG2 · TSMC · 65 nm · GPDK · 45 nm · Differenzverstärker · Gleichtaktrückkopplung · CMFB · Python · Virtuoso

The financial support by the Austrian Federal Ministry for Digital and Economic Affairs and the National Foundation for Research, Technology and Development and the Christian Doppler Research Association is gratefully acknowledged.

M. Bio (✉) · W. Scherr · S. M. Sondón · V. Hande
 Josef Ressel Centre for System-on-Chip Design Automation,
 Carinthia Institute for Microelectronics, Carinthia University
 of Applied Sciences, Europastraße 4, 9500 Villach, Austria
 mbio.coe@gmail.com

A. S. Agbemenu
 Department of Computer Engineering, Kwame Nkrumah
 University of Science and Technology, Kumasi, Ghana

J. Sturm
 Department of Integrated Systems and Circuit Design,
 Carinthia University of Applied Sciences, Villach, Austria

1 Introduction

As the design of integrated circuits (ICs) increase in complexity and scope, the subject of design automation continues to be crucial in not only speeding up the development but also to optimize and verify the functionality of the designs to ensure they meet the required specifications. For the most part, digital circuit designs can be completely automated by describing the behaviour and constraints of the circuit in a highly abstracted hardware description language such as VHDL or System verilog. Analog and mixed-signal (AMS) circuit design automation in contrast still involves a lot of intermediate steps with manual inputs that are necessary to fully develop a complete and functionally verified circuit. This man-in-the-loop approach to analog design is a well-known bottleneck and over the past years a lot of effort has been expended to reduce the level of hands-on by introducing automation concepts in different phases of the analog circuit development flow.

The initial phase of analog circuit design focuses on concept development, where tools like System-C AMS and MATLAB can be used to model and explore the functional space of the circuit. What follows after that is the mapping of component models to specific architectures which are built from fundamental devices like transistors, resistors and capacitors. Instead of using abstract component models as a first step to verify the model, tools like the hardware description python library (HDL21) [1] present a hardware description language for designing custom digital circuits similar to the programmable approach of digital design. This allows models to encapsulate the constraints (electrical) of specific components based on the process technology required for the design. This also creates an alternative to the traditional GUI schematic approach that is currently standard in the industry with tools like Cadence Virtuoso.

Several design exploration techniques can be used to select the architecture and define the specifications and constraints. In older long-channel process technologies, the square-law equation proves sufficient but is less accurate for short-channel devices and hence the gm/I_D method is usually preferred. These techniques lend themselves well to automation as has been shown in [2].

The layout phase of analog design presents the biggest challenge in terms of automation and is usually left to experienced layout engineers to implement manually. This has meant making last minute changes to the design is usually undesirable for time-sensitive projects. Efforts in tool development to help mitigate this problem include generator-based [3–5] and template-based [6] approaches to name a few.

In this paper, we investigate the claim of BAG2 being able to create process-independent analog layouts using parameterized generators, and consider the ef-

fort as well as the problems in getting the tool working as required.

The next section describes the details of the BAG2 framework and references some related works of analog circuits implemented with the framework. The benefits are discussed as well as the problems of the current implementation. Section III discusses the setup and configuration of BAG2 for planar process technologies, specifically TSMC 65 nm and GPDK 45 nm. Section IV provides the details of a case study of some analog circuit implementations in the above processes and the efforts as well as the challenges required to get the framework running. The final section discusses the findings from the case study and draws some conclusions on the BAG2 framework.

2 The Berkeley analog generator (BAG2) framework

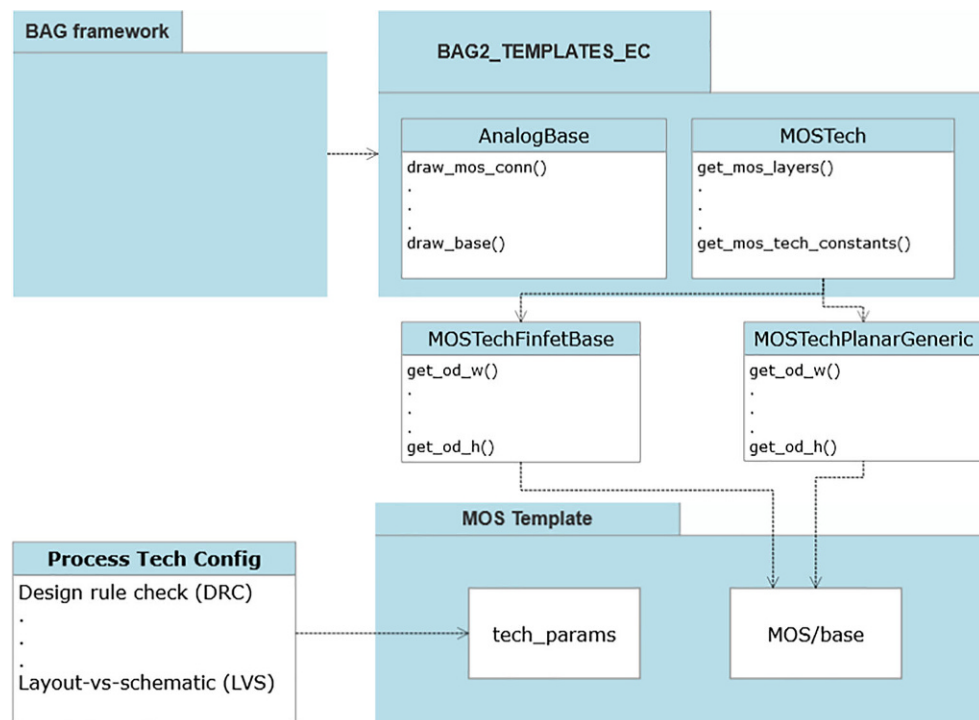
The BAG2 framework presents a non-traditional approach to analog circuit layout development, in that, it captures the specifications of the circuit into a parameterized generator which can then be executed to automatically generate specific instances of analog circuit layouts. One of the main usefulness of BAG2 is in its process-agnostic approach which allows generators to be used across different process technologies with minimal effort. In this regard, BAG2 has been demonstrated in a lot of published works to show how designs can be ported to different process technologies. [3] shows a differential amplifier for 3 different technologies (GF 45 nm RF SOI, TSMC 28 nm and TSMC 16 nm) using the same generator. In [7], the authors show how circuit layout can be easily iterated depending on specification changes and floor-plan constraints which allows for last minute design changes, which is not usually possible for manually-drawn layouts without considerable effort and time. In [8], the performance of a BAG2-generated layout for a single-tail dynamic comparator is compared to a manually drawn layout and shows similar results which demonstrate the benefits of BAG. Finally in [9], the vanilla BAG2 framework is setup and evaluated to determine the effort required to get it running, for which this work is an extension.

In spite of these benefits, there are still a number of notable limitations with the BAG2 generated circuit layout. Common layout optimization techniques used in analog circuits to reduce device-mismatches such as common-centroid or inter-digitized, as well as guard rings which are used to prevent latch-up currents require considerable effort to implement.

3 BAG2 setup for planar technologies

The promise of the BAG framework is that it enables circuit designers to develop parameterized generators that can be executed to generate layouts for different process technologies. In setting up BAG2 to work

Fig. 1 BAG2 Code dependency structure



with a new process, two problems can be immediately identified. The first is the lack of detailed setup documentation and information channel which makes the setup considerably difficult. The second problem is the lack of sufficient documentation of the application programming interface (API) which is standard and crucial for any software framework. This in addition to the limited source code documentation unnecessarily increases debug time when working with BAG2 and developing generators.

The process-specific parameters are captured in a YAML configuration file. BAG2 comes with a template for planar, fin-FET and SOI processes but only the fin-FET template for the dummy Cadence CDS_FF_MPT 18nm process is complete.

BAG2 introduces two layout engines that can be used when designing circuit generators. The default XBase layout engine provides some base classes which can be extended to implement the transistor structures for different process technologies. A sample of code structure is shown in Fig. 1. The **MOSTechFinfetBase** and **MOSTechPlanarGeneric** classes are extended to implement fin-FET and planar transistors respectively.

Given that TSMC 65nm and GPDK 45nm are planar processes, our initial setup extended the transistor base from the **MOSTechPlanarGeneric** class which provides already defined functions for drawing and routing transistors. Unfortunately, the **MOSTechPlanarGeneric** class appears incomplete and hence would not run. Upon debugging and trying to fix the code, other problems like missing functions and configurations parameters meant the generated layout had mismatches in device widths,

via metal widths, and TAP spacing and hence did not pass DRCs. Fig. 2 shows the layout generated.

Since there was no readily available working planar MOS template released with BAG2, the other option was to adapt the available dummy Cadence 18nm finFET CDS_FF_MPT process template for the planar technologies.

The other layout engine released for BAG2 is LAYGO [10] which offers a different approach to

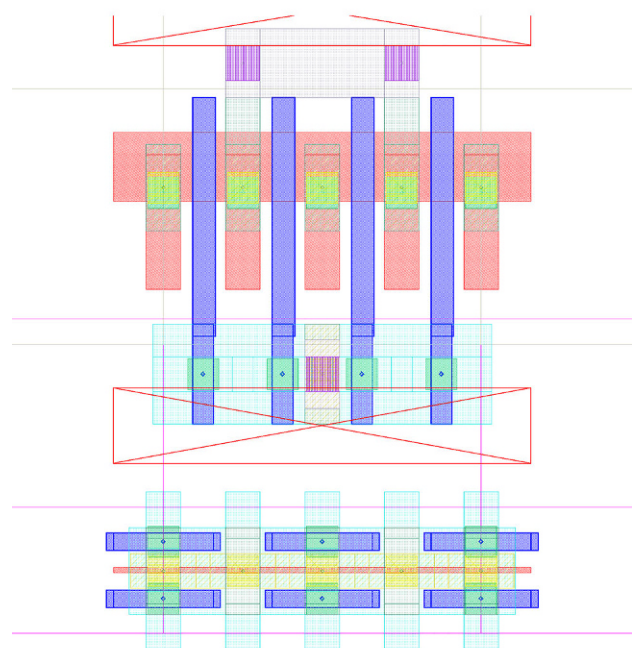


Fig. 2 BAG2 Planar MOS layout for TSMC 65nm

Table 1 Differential amplifier specs

Parameter	Value	Units
Gain, A_v	35	dB
Unity Gain Bandwidth, UGB	1	MHz
Supply Voltage, V_{DD}	1.2	V
Input Common-mode, $V_{in,CM}$	0.6	V
Load Capacitance, C_L	10	pF

generating circuit layouts. The details and limitations of LAYGO are not discussed in this paper as the focus is on the default XBase engine.

4 Case study

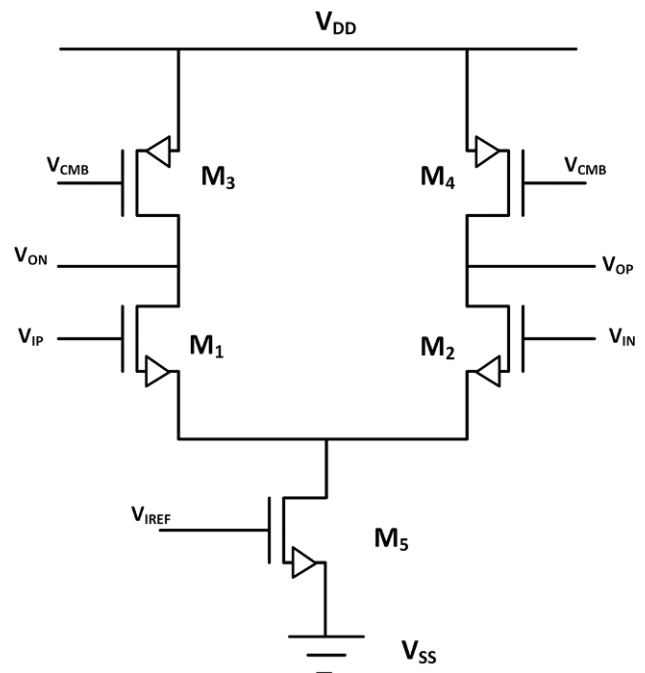
To demonstrate the viability of BAG2 and evaluate its process-independent layout generation claims, two generators were developed and executed for two different process technologies being TSMC 65 nm and Cadence dummy GPDK 45 nm. The generators were also executed on the dummy Cadence fin-FET process (CDS_FF_MPT) for completeness.

4.1 Differential amplifier

Differential amplifiers are used extensively in analog and mixed-signal circuit designs. The specific architecture used is usually determined by the requirements and constraints of the design. For this demonstration, a minimal-effort architecture was chosen which is a differential input and output with NMOS input and tail transistors and PMOS active loads. The supply voltage is 1.2 V as required for the model of transistor used. The dimensioning of the circuit was achieved with the gm/Id methodology.

The specifications for the differential amplifier are summarized in the table below.

With the specifications defined, a schematic template was implemented with arbitrary device dimensions. The real values are passed from the layout generator using the layout-first approach of BAG2. The schematic for the differential amplifier as generated by BAG2 is shown in Fig. 3 below.

**Fig. 3** Differential Amplifier Schematic

The layout generator was designed with TSMC 65 nm as the reference process. This meant the routing grid was tailored to the minimum metal spacing and areas of this process. The tracks for routing the transistors were also based on this reference. Fig. 4 shows the generated layout for the TSMC 65 nm process.

In the case of GPDK 45 nm which is a smaller node, the same routing grid, the number of tracks as well as transistor dimensions were used to maintain consistency. In order to take advantage on the smaller process, the design will have to be resized and the generator updated to use a different routing grid and number of tracks. Fig. 5 shows the generated layout for the GPDK 45 nm process. The difference in width compared to the TSMC 65 nm process is due to the difference in technology configuration for both processes. Once again, by optimizing the configuration,

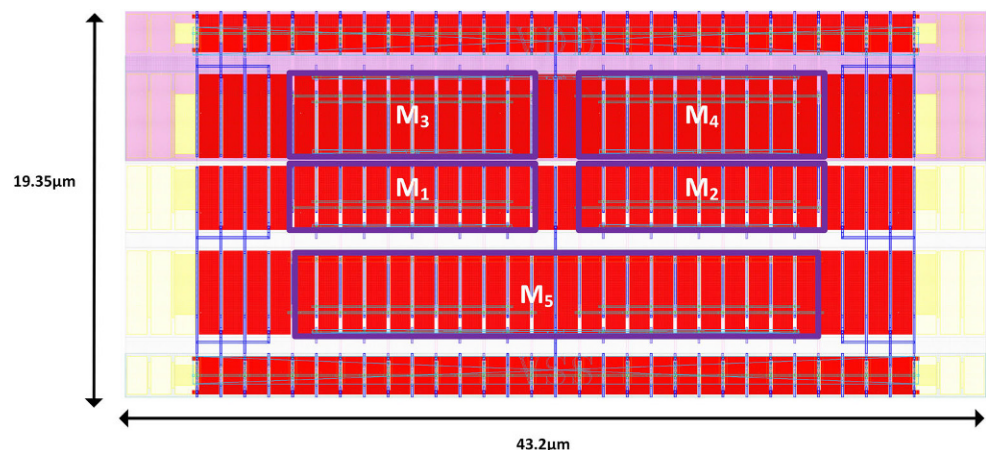
Fig. 4 Differential Amplifier layout for TSMC 65 nm

Fig. 5 Differential Amplifier layout for GPDK 45 nm

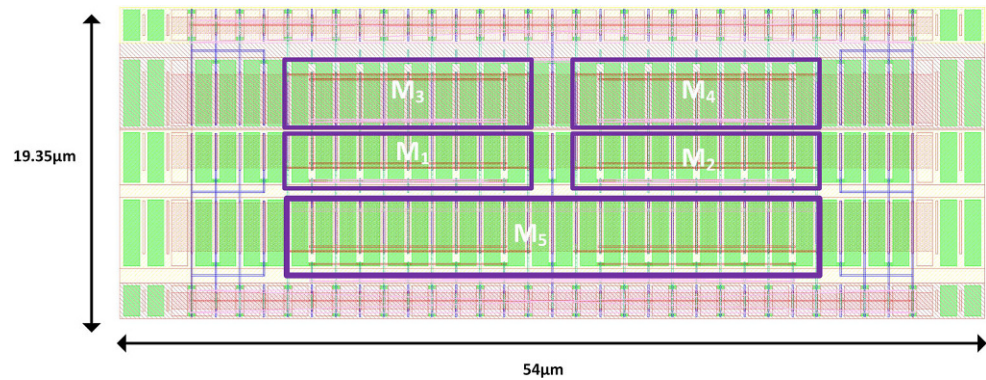


Fig. 6 Common-mode Feedback Circuit Schematic

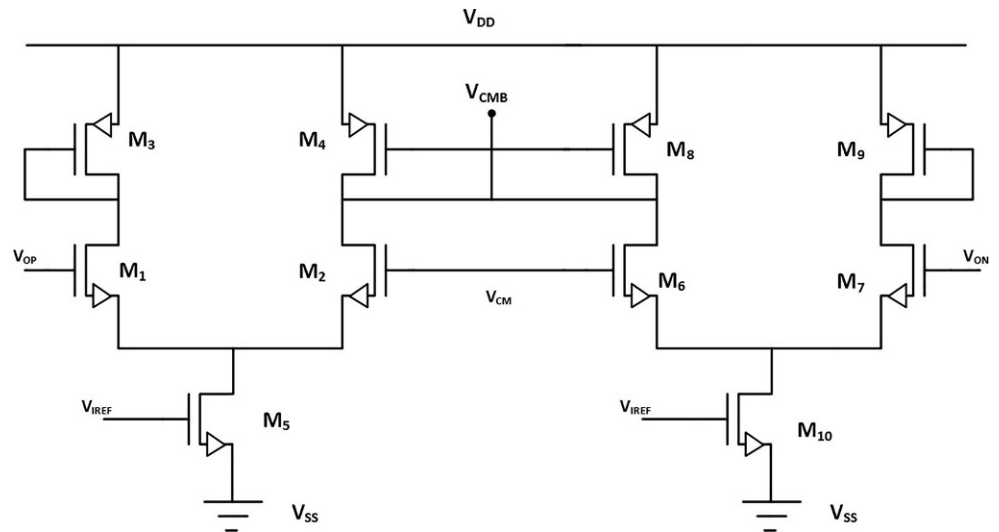
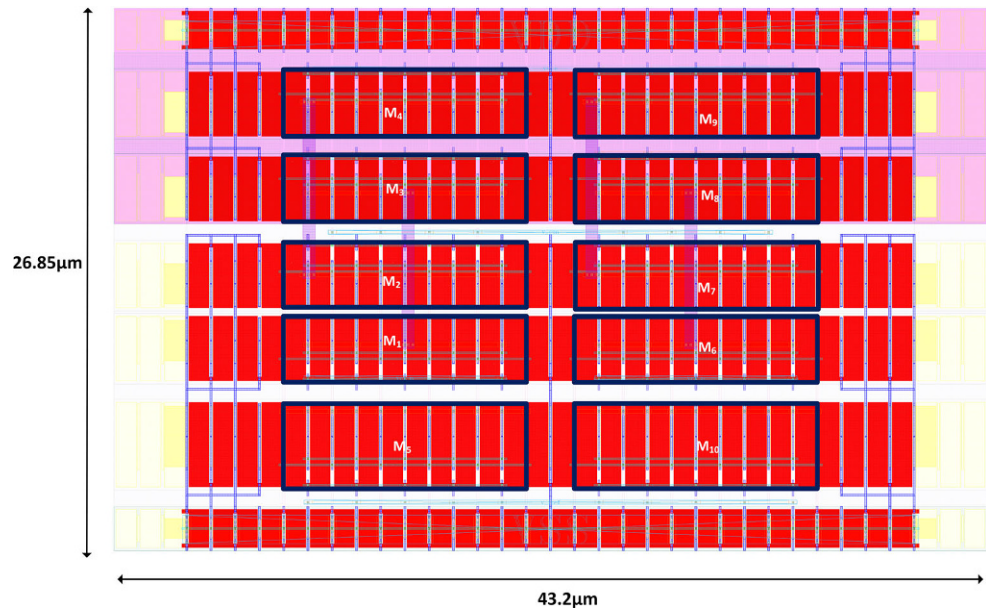


Fig. 7 Common-mode Feedback Circuit layout for TSMC 65 nm

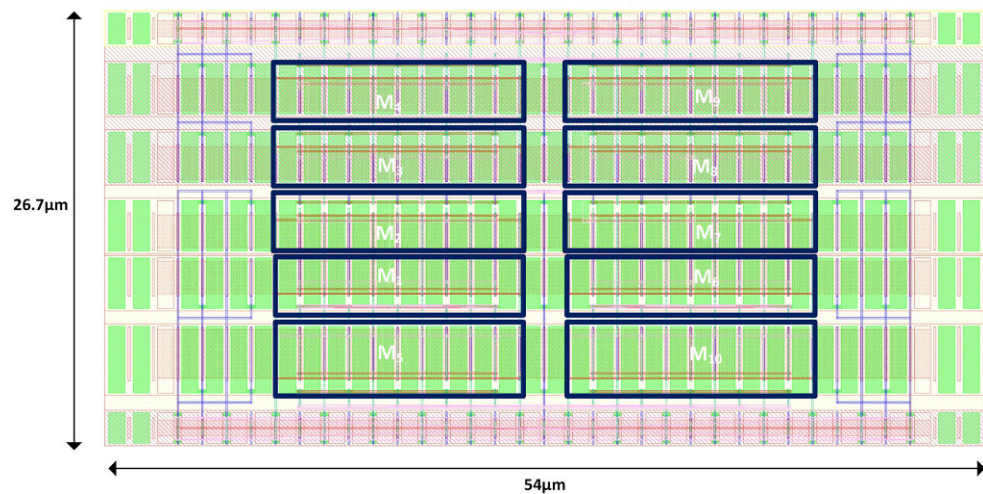


the GPDK 45 nm should be able to achieve smaller width and hence area.

4.2 Common-mode feedback circuit

Another circuit which is used to stabilize the output common-mode voltage of amplifiers is the common-mode feedback circuit (CMFB). The common-mode

Fig. 8 Common-mode Feedback Circuit layout for GPDK 45 nm



voltage in this case is set to half the supply voltage which is 0.6V

The schematic template for the CMFB circuit was first developed with arbitrary transistor dimensions and then the real dimensions were calculated from the generated layout. The schematic as generated by BAG2 is shown in Fig. 6 below.

The layout generator for the CMFB circuit was designed for the TSMC 65 nm process with the same routing grid as for the differential amplifier with a suitable floor-plan taking into account the number of devices in the circuit and their interconnections. Fig. 7 shows the generated layout.

The final step was to also run the generator on the GPDK 45 nm process with the same routing grid, number of routing tracks and the layout as generated is shown in Fig. 8 below.

5 Generator design and evaluation

5.1 Generator design

Traditionally, layout engineers entirely depend on a visual environment to develop analog circuit layouts. Tools like the Cadence Virtuoso Layout Editor or open-source alternatives like KLayout [12] and MAGICAL [11] allow designers to create and modify device instances to create a layout floor-plan that is consistent to their implementation requirements while being considerate of how the devices are routed by signal and power wires to minimise the impact of parasitic capacitance and device mis-matches.

In the case of BAG2 which is entirely a code-based approach to layout development, designers don't have direct feedback path to visualising the final layout of the circuit but instead have to rely on an initial draft floor-plan that is implemented by either paper and pencil or some other digital drawing application. This drawback can be significant in the case of complex circuits where the parameters such as number and width of transistor fingers can dramatically change the re-

sulting layout that is generated by the code. Also, the de-coupling of floor-plan draft makes it a bit more difficult for layouters without programming experience to run the generator without additional and sufficient documentation of the code.

From the common-mode feedback circuit shown in Fig. 6, the parameters of the transistors used to achieve the required performance is shown in Table 2. The pair of values in the finger width and number of fingers columns are used to generate 2 different

Table 2 CMFB circuit transistor parameters

Transistor	Finger Width (μm)	Number of fingers	Total Width (μm)
M_1, M_6	[5.84, 3.65]	[10, 16]	58.4
M_2, M_7	[5.84, 3.65]	[10, 16]	58.4
M_3, M_8	[11.68, 7.3]	[10, 16]	116.8
M_4, M_9	[11.68, 7.3]	[10, 16]	116.8
M_5, M_{10}	[11.68, 7.3]	[10, 16]	116.8

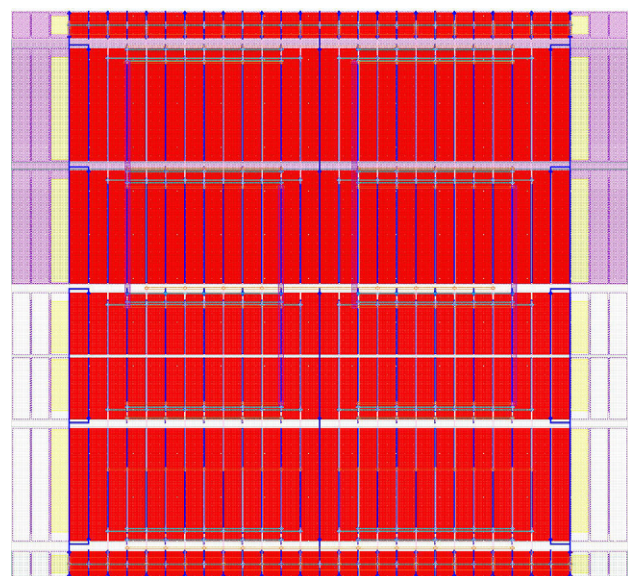


Fig. 9 CMFB Layout with number of fingers = 10

Fig. 10 CMFB Layout with number of fingers = 16

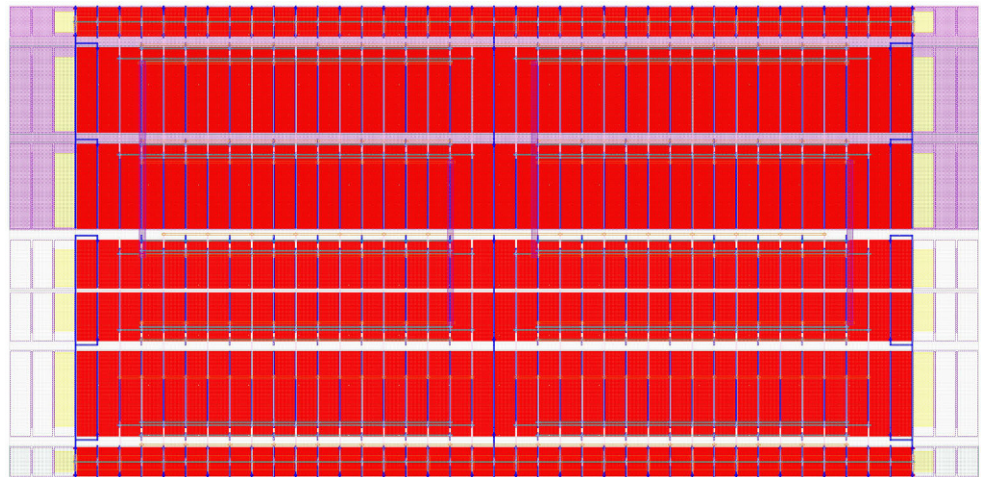


Fig. 11 Hierarchical Circuit Floor-plan 1

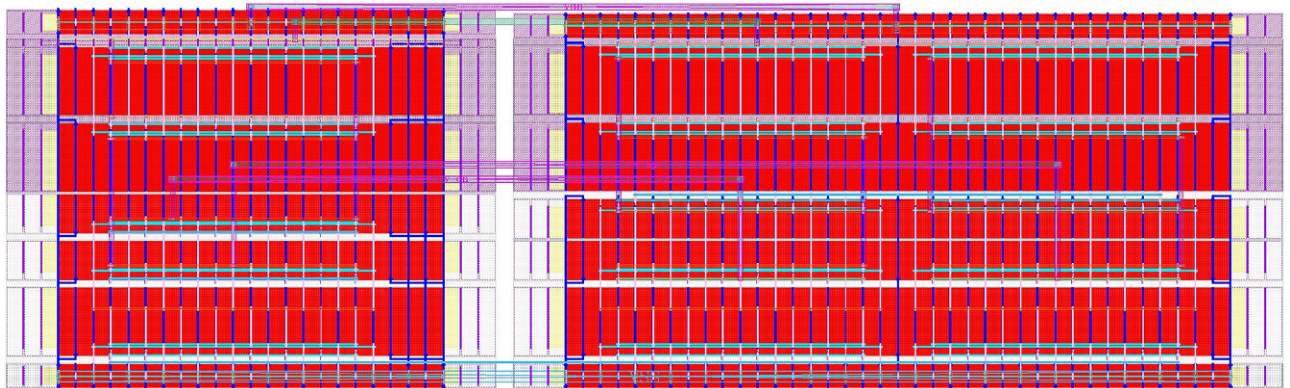
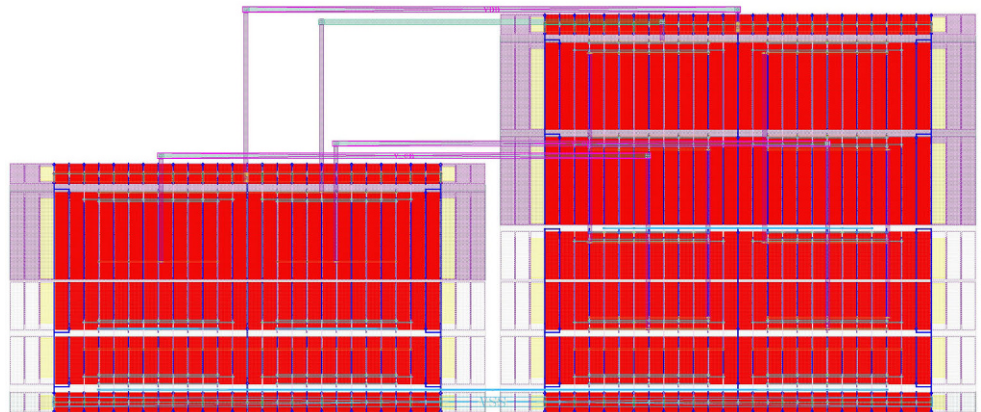


Fig. 12 Hierarchical Circuit Floor-plan 2

layouts for the CMFB circuit, the details of which are discussed next.

Stacking transistors M3, M4, M1, M2 and M5 into a single column and 5 rows with each transistor having 10 number of fingers and a corresponding finger width for the desired total width as specified in Table 2 results in a balanced layout as shown in Fig. 9.

On the other hand, opting for more number of fingers, in this case 16 and a corresponding finger width produces a more wider layout that might be necessary if the physical chip structure requires this floor-plan.

In either case, the number of dummy transistors that are used to separate the transistors as well as to pad the edges can also change how the final floor-plan of the layout turns out as shown in Fig. 10.

A third option will be to either arrange the devices into a single row of PMOS transistors stacked on top of a row of NMOS transistors to get an extremely horizontal layout or stack each device on top of each other to get vertical layout but this is considered impractical and is therefore not implemented in this research.

Fig. 13 Monte-Carlo Gain using Schematic of Floor-plan 1

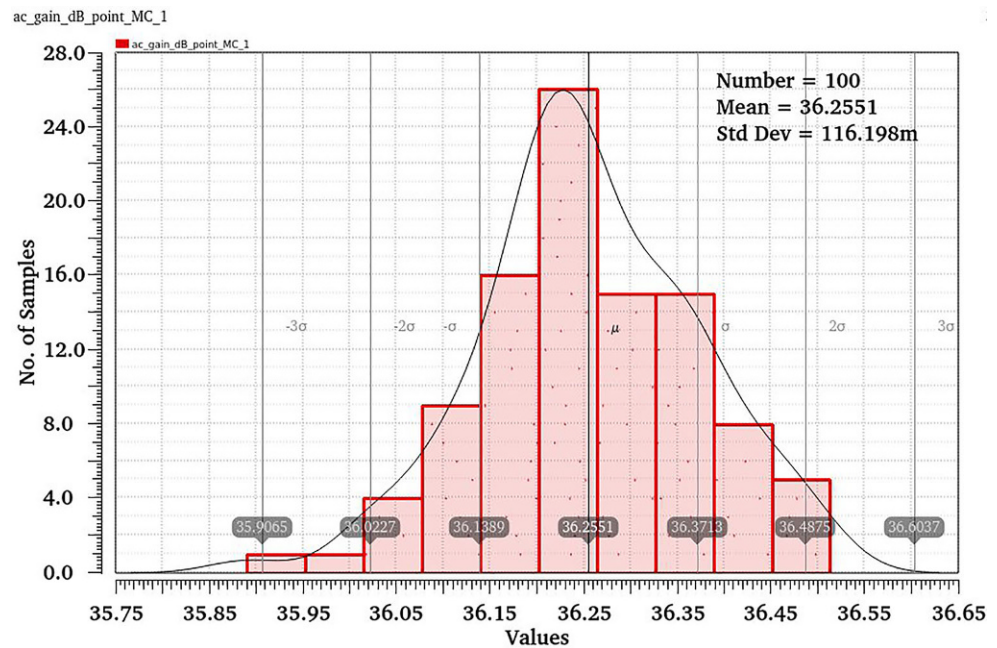
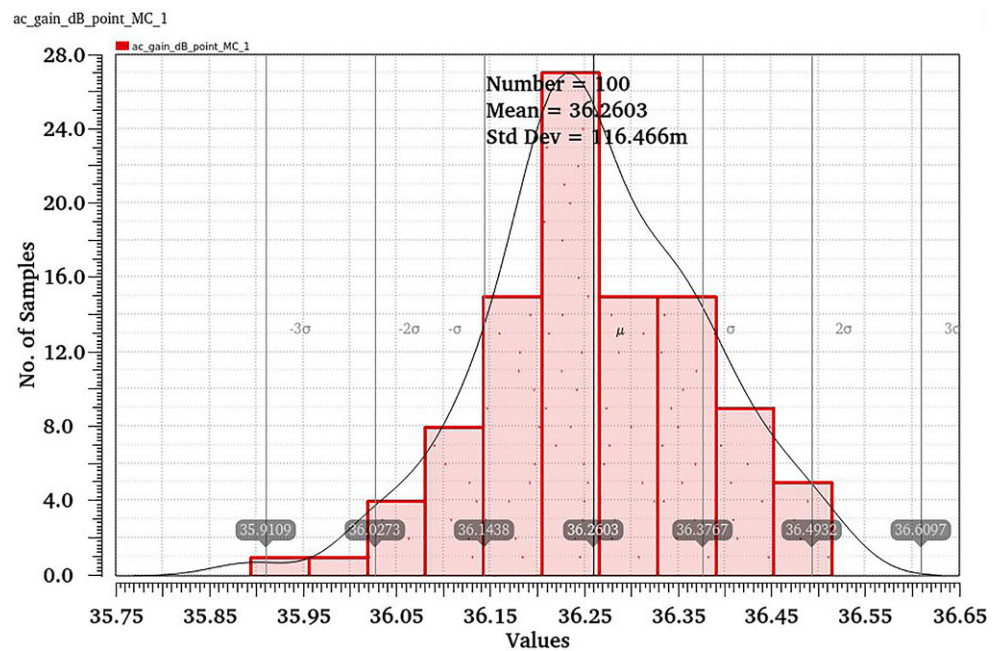


Fig. 14 Monte-Carlo Gain using PEX of Floor-plan 1



Given the massive design space that has to be explored to achieve the optimum layout, having to write a different generator for every instance of a particular layout floor-plan is not trivial and even though the code can be generalised to some extent to account for a set of layout types, the effort needed to do that might be too significant.

5.2 Hierarchical circuits: differential amplifier with CMFB

BAG2 generators can be instantiated in top-level generators to create a system hierarchy. The each sub-module in a block can be connected to every other

sub-module at the same level to achieve the integration that is typically done by hand except in this case the module are connected using code.

Using the 2 different CMFB layout floor-plans, the top-level differential amplifier with CMFB circuit are show in Figs. 11 and 12 which are implemented using two different generators for the CMFB circuit.

It should be noted that the floorplan optimization is also applied to the differential amplifier as well but this has a less significant effect in this case since the CMFB circuit is much bigger and has more devices. For any generator circuit, all the individual blocks will have to be optimized to achieve the best top-level layout.

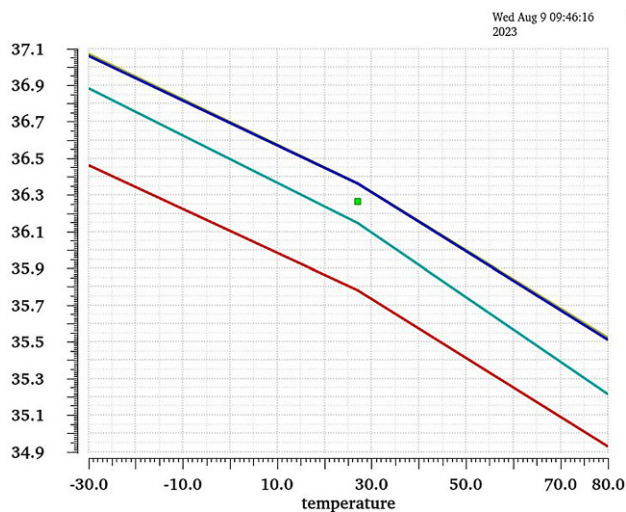


Fig. 15 Parametric Simulation of Gain (dB) for Floor-plan 1 Schematic. Legend: red FF Corner, cyan FS Corner, violet SF Corner, yellow SS Corner, green TT

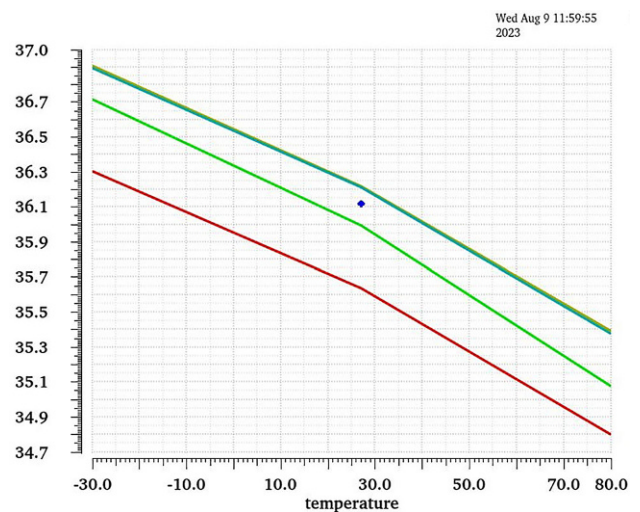


Fig. 16 Parametric Simulation of Gain (dB) for Floor-plan 1 PEX Layout. Legend: red FF Corner, green FS Corner, cyan SF Corner, yellow SS Corner, violet TT

6 Evaluation and discussion

6.1 Evaluation

The verification of analog circuit layout requires the use of tools like Cadence Assura or Siemens Calibre to perform design-rule checks (DRC) and layout-vs-schematic (LVS) to ensure the designs layout matches the schematic netlist as well as meets the requirements as specified in the process development kit (PDK) by the foundry. In addition, the performance of the layout can be evaluated by parasitic extraction (PEX) and compared with the schematic which serves as an ideal target.

In this case, the DC and AC performance of the circuit was evaluated to ensure they met the specification for gain, output common-mode voltage and gain-bandwidth using both parametric simulation of the device over all corners for process-temperature-voltage variation as well as Monte-Carlo.

6.1.1 Hierarchical circuit floor-plan 1 (FP 1)

As can be seen from the results of Monte-Carlo simulation in Figs. 13 and 14, the variation between the schematic and PEX layout using the floor-plan in Fig. 11 is minimal which is primarily a result of using less number of fingers and hence wider finger widths which makes it more tolerant to process variations.

Fig. 17 Monte-Carlo Gain using Schematic of Floor-plan 2

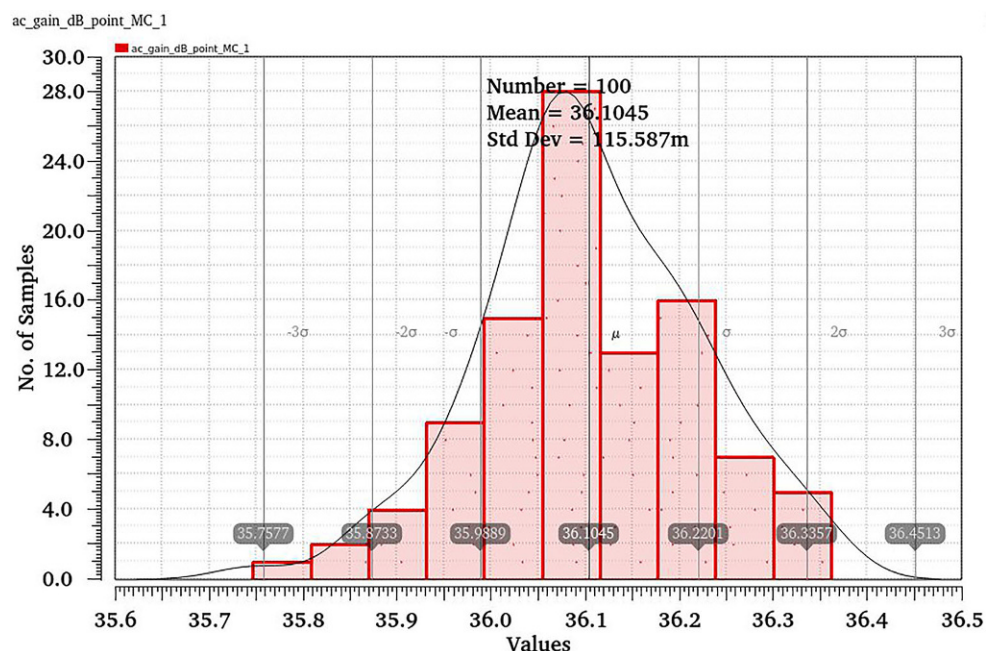
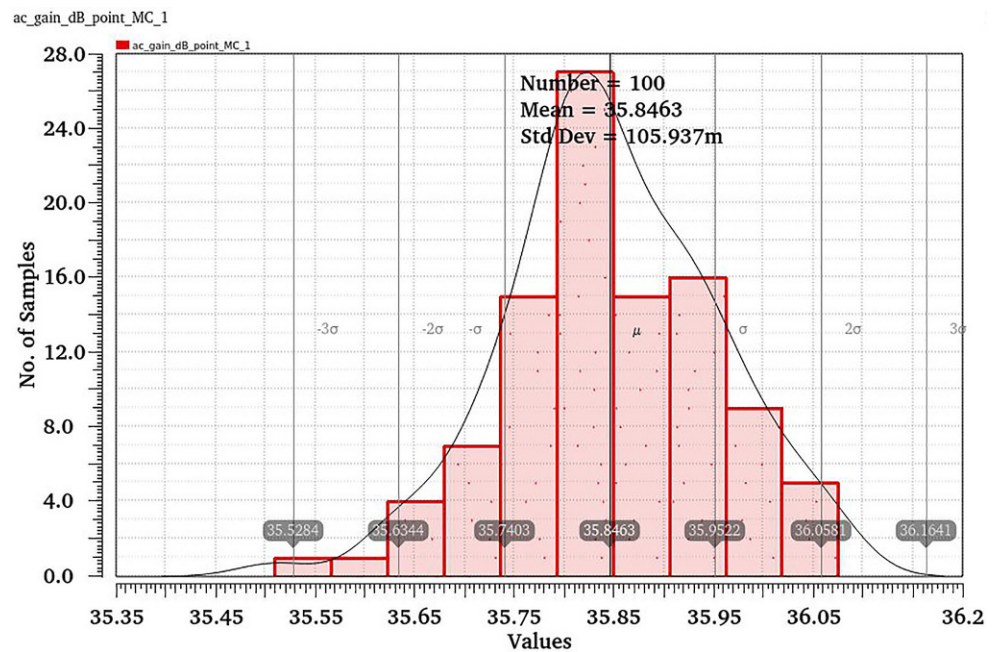


Fig. 18 Monte-Carlo Gain using PEX of Floor-plan 2



The same is true for the parametric simulation for the device corners over temperature which can be seen in Figs. 15 and 16. Here the temperature is swept between -30°C and 80°C .

6.1.2 Hierarchical circuit floor-plan 2 (FP 2)

When using more number of fingers and hence a thinner finger width as is the case for the floor-plan in Fig. 12, the variation in the Monte-Carlo simulation of the gain between the schematic (mean: 36.1045) and PEX layout (mean: 35.8463) is more pronounced which is primarily attributable to effects caused by process variations.

The parametric simulation once again shows a noticeable variation in the gain performance of the circuit

between the schematic (36.91 for best case and 36.31 for worst case) and the PEX layout (36.64 for best case and 35.85 for worst case) over the given temperature range which demonstrates how the choice of floor-plan variables can affect the performance of the circuit as shown in Figs. 19 and 20.

The results of the parametric simulation for the device corners over temperature is shown in full in the appendix. For much more complex circuits with stringent performance requirements, this variation will be significant and will require more consideration when developing a generator to get the best outcome.

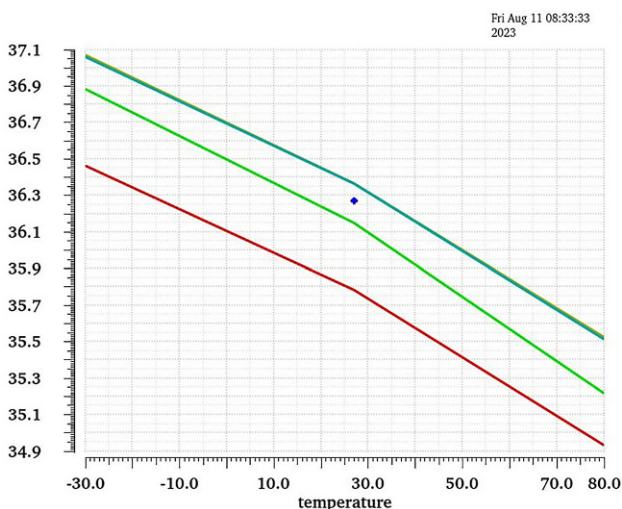


Fig. 19 Parametric Simulation of Gain (dB) for Floor-plan 2 Schematic. Legend: red FF Corner, green FS Corner, blue SF Corner, yellow SS Corner, violet TT

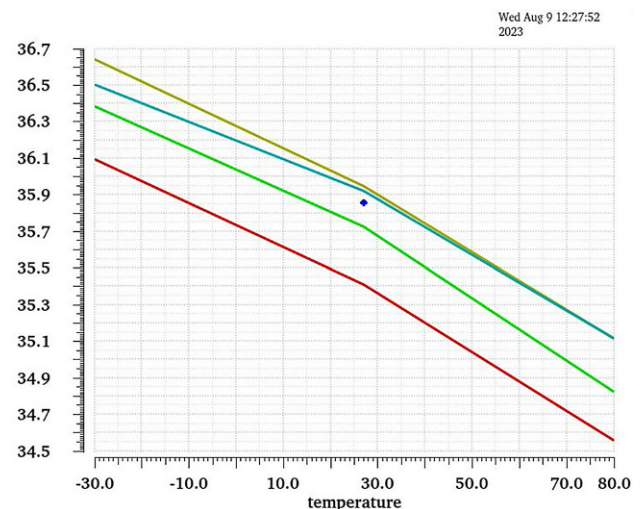


Fig. 20 Parametric Simulation of Gain (dB) for Floor-plan 2 PEX Layout. Legend: red FF Corner, green FS Corner, blue SF Corner, yellow SS Corner, violet TT

6.2 Discussion

The promise of BAG2 as a process-independent framework for automating analog layouts for the most part works as intended but with significant hurdles that have to be overcome to setup the framework for a new process technology. The most notable problem is the lack of appropriate documentation of the API and source code. This makes it close to impossible to determine and even utilize all the features of the framework. BAG2 releases with a number of tutorial generators and some basic work-flow with minimal documentation and this means generator development has to be tailored to these sample generators which increases the effort for writing more complex generators. Also the effort of debugging and tracing code execution paths is significant, especially when setting up BAG2 for a new process due to the lack of documentation. In this instance, it took about 3 months to fully adapt and setup BAG2 for the TSMC 65 nm process and another 2 weeks to get it to work for the GPDK 45 nm process. The generator code on the other hand took about a month to develop, with a few more days needed to iterate the layout floorplan.

Once these hurdles are overcome though, generators largely work across multiple processes as shown in this case study.

In order for BAG2 generators to take advantage of any specific process, the generator code requires modification due to differences in the routing grid and number of tracks as a result of different wire lengths/spacing for each process. This inconvenience, though minor is necessary to prevent shorting wires in the circuit and generating DRC-clean layouts.

Lastly, lack of important layout structures like device interleaving to reduce mismatch and guard rings prevents the generated layouts from being optimal.

7 Conclusion and outlook

In effect, this paper evaluates the use of BAG2 as a reliable tool for automating analog layouts for different processes namely the TSMC 65 nm and GPDK 45 nm and shows that it offers a lot of benefits. A case study of a number of analog circuits is used to evaluate the framework.

As noted, the lack of flexibility in using the BAG2 frameworks coupled with the strictly code-based development environment poses a significant hurdle that needs to be overcome by layout engineers to use the framework. The need to develop multiple generators for similar layouts is also inconvenient and increases the initial time and effort which has to be invested to produce a viable layout.

As a next step, the BAG2 generated layout could be compared to a hand-made layout and their performance examined to determine how much of an advantage or disadvantage each solution offers. A more

objective analysis would require both the automatic and hand-made layouts implemented on a physical chip which will provide a much more useful insight and would capture the real effect of the process variations and its impact on the chip.

Acknowledgements The financial support by the Austrian Federal Ministry for Digital and Economic Affairs and the National Foundation for Research, Technology and Development and the Christian Doppler Research Association is gratefully acknowledged.

Funding Open access funding provided by Carinthia University of Applied Sciences (CUAS).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

8 Appendix

8.1 Parametric corners

- Nominal PMOS – Nominal NMOS (TT)
- Fast PMOS – Fast NMOS (FF)

Table 3 Gain Corner Simulation Results

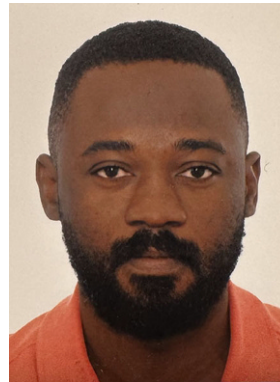
Corners	Schematic		PEX Layout	
–30 °C				
	FP1	FP2	FP1	FP2
TT				
FF	36.27	36.31	36.46	36.10
FS	37.07	36.91	37.07	36.64
SF	36.88	36.72	36.88	36.39
SS	37.06	36.90	37.06	36.50
27 °C				
	FP1	FP2	FP1	FP2
TT	36.27	36.18	36.27	35.86
FF	36.27	36.31	36.46	36.10
FS	37.07	36.91	37.07	36.64
SF	36.88	36.72	36.88	36.39
SS	37.06	36.90	37.06	36.50
80 °C				
	FP1	FP2	FP1	FP2
TT				
FF	36.27	36.31	36.46	36.10
FS	37.07	36.91	37.07	36.64
SF	36.88	36.72	36.88	36.39
SS	37.06	36.90	37.06	36.50

- Fast PMOS – Slow NMOS (FS)
- Slow PMOS – Fast NMOS (SF)
- Slow PMOS – Slow NMOS (SS)

References

1. Fritchman D (2023) HDL21: Hardware Description Python Library. GitHub June 2023 (<https://github.com/dan-fritchman/Hdl21>)
2. Pinjare SL, Nithya G, Nagaraja VS, Sthuthi A (2018) A Gm/id based methodology for designing common source amplifier. In: 2nd International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE)
3. Chang E, Han J, Bae W, Wang Z, Narevsky N, Nikolic B, Alon E (2018) BAG2: a process-portable framework for generator-based AMS circuit design. IEEE Custom Integrated Circuits Conference, Apr 2018
4. Chen H, Liu M, Xu B, Zhu K, Tang X, Li S, Lin Y, Sun N, Pan DZ (2021) MAGICAL: an open- source fully automated analog IC layout system from Netlist to GDSII. IEEE Design and Test, Apr. 2021
5. Dhar T, Kunal K, Li Y, Lin Y, Madhusudan M, Poojary J, Sharma AK, Burns SM, Harjani R, Hu J, Mukherjee P, Yaldiz S, Sapatneka SS (2020) The ALIGN open-source analog layout generator: v1.0 and beyond (invited talk). IEEE/ACM International Conference On Computer Aided Design (ICCAD), Nov. 2020
6. Castro-Lopez R, Guerra O, Roca E, Fernandez FV (2008) An integrated layout-synthesis approach for analog ICs. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Jun. 2008
7. Crossley J, Puggelli A, Le H-P, Yang B, Nancollas R, Jung K, Kong L, Narevsky N, Lu Y, Sutardja N, An EJ, Sangiovanni-Vincentelli AL, Alon E (2013) BAG: a designer-oriented integrated framework for the development of AMS circuit generators. IEEE/ACM International Conference on Computer-Aided Design, Nov. 2013
8. Bashir M, Abbassi F, Mistic MV, Sturm J, Getnot H (2020) Performance comparison of BAG and custom generated analog layout for single-tail dynamic comparator. IEEE Austrochip Workshop on Microelectronics, Sep. 2020
9. Bio M, Scherr W, Agbemeny AS, Sondón SM, Sturm J, Hande V (2023) BAG2 assisted hierarchical analog layout synthesis for planar technologies. In: 2023 Austrochip Workshop on Microelectronics (Austrochip), Graz, Austria, pp 44–47 <https://doi.org/10.1109/Austrochip61217.2023.10285163>
10. Laygo (2023) LAYGO – LAYout with gridded objects. GitHub June 2023 (<https://ucb-art.github.io/laygo/index.html>)
11. Magical-Eda magical-eda/MAGICAL: machine generated analog IC layout. <https://github.com/magical-eda/MAGICAL>. Accessed 1 Sept 2023
12. Koefferlein M (2019) KLayout layout viewer and editor. <http://www.klayout.de/>. Accessed 12 Aug 2023

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



(2022). Research interests include IC Design Workflows with focus on systems modelling and verification, framework development for design automation.



Wolfgang Scherr, DI(FH), MSc, worked for 25 years in the semiconductor industry in the field of automotive sensors, automotive and industrial power as Principal Systems Engineer, ASIP expert and Functional Safety Professional. Afterwards, he joined in 2020 the University of Applied Sciences in Villach as Key Researcher for digital design, modelling and simulation and lead of the RESPECT research group.



A. Agbemeny, born in Accra, 1981 is a lecturer in the Department of Computer Engineering, KNUST, Kumasi, Ghana. He is also the lead of the TI Microelectronics Lab, KNUST and a researcher with the Responsible AI Lab (RAIL, KNUST). He is a member of IEEE-CASS and IEEE-SSCS societies. His current research interests are in data converters and using AI analog design. He was a visiting scholar to the Analog and Mixed Signal Center of the TAMU, Texas.



Santiago Sondón, Holds degrees in Electronic Engineering and a Doctorate in Engineering from Universidad Nacional del Sur in Argentina (UNS), which he earned in 2006 and 2014, respectively. Currently, he serves as the Head of the Josef Ressel Center for System-on-Chip Design Automation and is a professor in the master's program for Integrated Systems and Circuit Design at Fachhochschule Kärnten (CUAS) in Austria. Santiago is also a key researcher and deputy

director for the RFFE-Lab, a collaborative laboratory funded by Silicon Austria Labs and CUAS. Throughout his career, Santiago has made significant contributions in both research and industry transfer, particularly in the areas of analog/mixed-signal and RF design, radiation effects in circuits and systems, and recently in the field of analog automation and high-level system modeling. His professional journey includes roles as an assistant professor at UNS, research assistant at CONICET, leading the RF research group at CMNB-INTI in Argentina, and a notable appointment as a research engineer at the Max Planck Institute for Solar System Research in Germany during 2014/15 for the ESAs mission JUICE.



Vinayak Hande, PhD, received his PhD degree in Electrical Engineering from Indian Institute of Technology Bombay (IIT-Bombay), India, in 2015. Before joining the PhD program, he worked in GDA Technologies Limited, India, as an analog IC designer. After completing PhD, he joined the Department of Electrical Engineering as an Assistant Professor at Indian Institute of Technology Ropar (IIT-Ropar) till 2021. Since then, he is working in Carinthia University

of Applied Sciences Villach (CUAS-Villach), Austria, as a Senior Researcher. His research areas include analog and mixed-signal circuit design for sensor front-end systems, PVT tolerant references along with developing routines for the analog design automation.



Johannes Sturm, Received the Dipl.-Ing. degree in physics from Graz University of Technology in 1995 and his Dr. techn. degree from Vienna University of Technology in 2006 respectively. He joined the Siemens-Semiconductor-Group in Villach, Austria, in 1996, working in semiconductor process development. In 1999 he joined Infineon Technologies development center in Villach as analog designer, concept engineer and project manager for analog,

mixed-signal ASIC projects in the field of optical sensors and high-speed CMOS circuits. Since 2006 he has been a Professor for analog circuit design at Carinthia University of Applied Sciences (CUAS), Villach, Austria, where he is chairing the department of Integrated Systems and Circuits Design since 2017. From 2019 to 2021 he established the research unit for Frontend Integrated Circuits at Silicon Austria Labs (SAL) in Villach. He is responsible for the Carinthia Institute for Microelectronics research center at CUAS with his current research interests in the field of RF integrated circuits, high-speed serial data communication, photonic integrated circuits and integrated sensor frontends.