Implementation of a Semi-Automatic Design Procedure of Preamplifiers in a Multistage CMOS Comparator

João Lucas J. Brum, Beatriz E. Hatschbach Rezende, Cristian Müller, Lucas C. Severo,
Alessandro G. Girardi and Paulo César C. de Aguirre
Computer Architecture and Microelectronics Group - GAMA
Federal University of Pampa - UNIPAMPA, Alegrete, Brazil
{joaobrum.aluno, alessandro.girardi, pauloaguirre}@unipampa.edu.br

***Abstract*—This paper presents a systematic design procedure for the preamplifiers of a CMOS multi-stage comparator in 130-nm process technology. The procedure is based on the gm/I_D methodology, which relates the transconductance-to-current ratio gm/I_D and the normalized current I_D1/(W/L) as a technology characteristic curve. The comparator consists of three preamplifiers and, in addition to signal amplification, performs voltage scaling from a 5-V input to a 1.5-V output. The comparator has as application a high-resolution SAR ADC. The script for analytical sizing is implemented in Python, saving design time. The results show that the analytical systematic design procedure presents design solutions with performance estimation that is very close to the obtained with electrical simulation.**

***Index Terms*—Multistage comparator, CMOS, gm/ID, Python**

I. Introduction

Analog-to-digital converters (ADCs) are indispensable tools for bridging the gap between the real and digital worlds. These circuits are widely used for acquiring biomedical signals, audio data, and sensor readings.

In high-resolution SAR -based (Successive Approximation Register) ADCs, one of the most important circuits is the comparator. It requires high precision, low offset voltage and high gain-bandwidth product (GBW).

This paper presents a three-stage comparator applied to high-resolution ADCs [1], [2], [3], where the voltage is stepped down from 5 V to 1.5 V in each stage. This is necessary to allow voltage scaling between the analog and digital parts of the circuit. The schematic diagram of the multistage comparator is shown in Fig.1. In addition to the three preamplifiers, the comparator also has a regenerative latch before the output. This work is limited to the design of the preamplifiers and does not address the design of the latch.

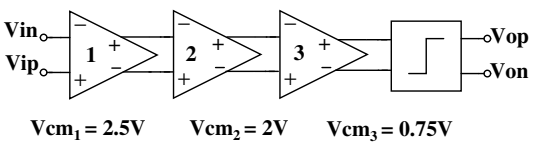


Fig. 1: Multi-stage comparator simplified block diagram with three preamplifiers and a regenerative latch.

To avoid complex calculations and optimize time, the gm/I_D design methodology presented in [4] is used for sizing the preamplifiers. The method exploits the transconductance over drain current ratio (gm/I_D) versus the normalized current I_D/(W/L) for modeling the transistors, widely applied to operational amplifier circuits [5], [6], [7]. This method provides an accurate estimate of input transistor bias and resistance values. This approach meets the necessary specifications for proper operation of the ADC.

The remainder of this article is organized as follows: Section II describes the gm/ID methodology; in Section III the code in Python language and its description are presented; in Section IV, the simulation results are presented; finally, Section V presents the conclusions of this work.

II. gm/I_D methodology

As a starting point, it is necessary to define the performance specifications required for each amplification stage. The specifications include low-frequency voltage gain, gain-bandwidth product, load capacitance, and DC common voltage levels at input and output. Table I presents this target specifications used in this work.

To meet the performance specifications, a specific amplifier topology is chosen - in this case, a common-source differential amplifier with resistive load based on PMOS transistors, as shown in Fig.2.

The classic topology amplifies the difference between two input signals while rejecting any common mode signals that may be present. A pair of P-type CMOS transistors are connected in a common source configuration with an tail current bias block. Input signals are applied to the gate of the

TABLE I: Design specifications for the preamplifiers

Parameter	Preamp_1	Preamp_2	Preamp_3
Transistor type	PMOS 5 V	PMOS 5 V	PMOS 1.5 V
Av_0 (V/V)	-6.00	-6.00	-6.00
GBW (MHz)	300.00	300.00	300.00
C_L (fF)	71.54	125.00	77
VDD (V)	5.00	5.00	1.50
Vin_{CM} (V)	2.50	2.00	0.75
$Vout_{CM}$ (V)	2.00	0.75	0.75

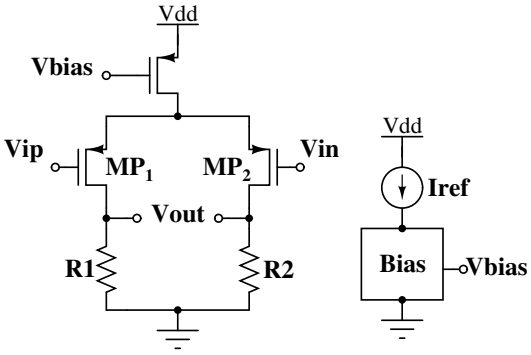


Fig. 2: Schematic of the preamplifiers.

PMOS transistors. The drain of each transistor is connected to a load resistor and the differential output is obtained between the the two load resistors.

For this topology, the voltage gain equation of the amplifier is given by $Av_0 = -gm_1 \cdot R_1$. It is possible to manipulate this equation to put Av_0 in function of gm/ID:

$$Av_0 = -gm_1 \cdot R_1 = -\frac{gm_1}{I_{D1}} \cdot R_1 I_{D1} = -\frac{gm_1}{I_{D1}} \cdot Vout_{CM} \quad (1)$$

After defining the circuit performance specifications, the gm/ID value is calculated for each transistor. Since the value of the common-mode output voltage is a constraint in this design, we can use Equation 2 to calculate the required gm/ID for the transistors.

$$gm/I_{D1} = \frac{-Av_0}{Vout_{CM}} \quad (2)$$

The transconductance of each transistor (gm) can be calculated using the equation 3.

$$gm_1 = GBW \cdot C_L \cdot 2\pi \quad (3)$$

Here, GBW is the target gain-bandwidth product and C_L is the load capacitance. With the values of gm_1 and gm/I_D, we can isolate I_{D1} to obtain the drain current of the transistors.

$$I_{D1} = \frac{gm_1}{(gm/I_D)} \quad (4)$$

The value of the resistors are defined as equation 5.

$$R = \frac{-Av_0}{(gm/I_D) \cdot I_{D1}} \quad (5)$$

The resistor value is a critical factor that needs fine-tuning, as the physical implementation of the resistor in the integrated circuit introduces parasitic capacitance at the output node. The equation 6 represents the total output capacitance of the circuit, where C_L is the load capacitance and C_{R1} is the parasitic capacitance that the resistor adds to the circuit.

$$C_{L(tot)} = C_L + C_{R1} \quad (6)$$

Using equation 3, we can recalculate the GBW value, now considering the total output capacitance.

$$GBW = \frac{gm_1}{C_{L(tot)} \cdot 2\pi} \quad (7)$$

An iterative procedure is necessary to adjust the resistance value that produces an output load capacitance compatible to the desired GBW.

Based on the previous calculations, two technology characteristic curves are used for estimating the transistor bias point and the W/L. The gm/I_D x V_{GS} curve (Fig. 3) and gm/I_D x I_n curve (Fig. 4) can be obtained through electrical simulation of a single transistor for a fixed channel length.

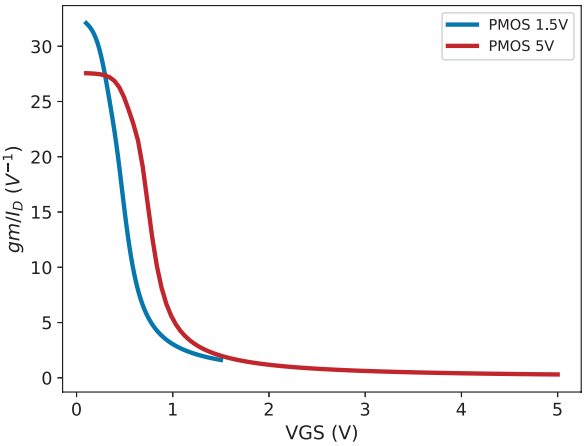


Fig. 3: Simulated gm/I_D x VGS curves.

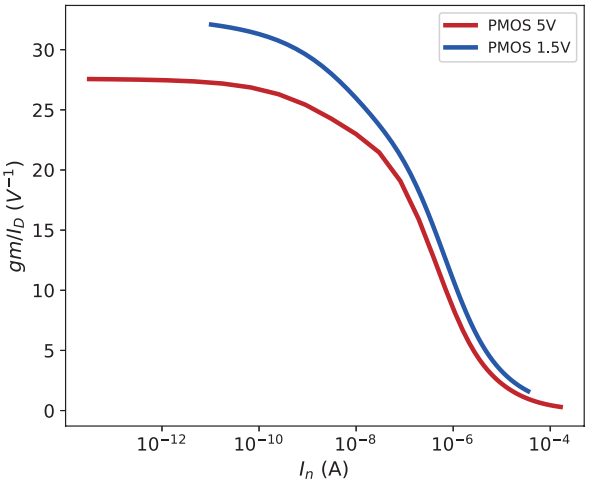


Fig. 4: Simulated gm/I_D x I_n caracteristic curves.

By means of the value of gm/I_{D1} we estimate the size of transistor through the curve gm/I_D x I_n, where I_n is the normalized current, represented by:

$$I_n = \frac{I_D}{(W/L)} \quad (8)$$

Knowing the W/L ratio, and considering a fixed L, it is possible to calculate the W of the transistor.

The gate-source voltage values (V_{GS1}) are taken from the curve gm/I_D x V_{GS} using the same procedure.

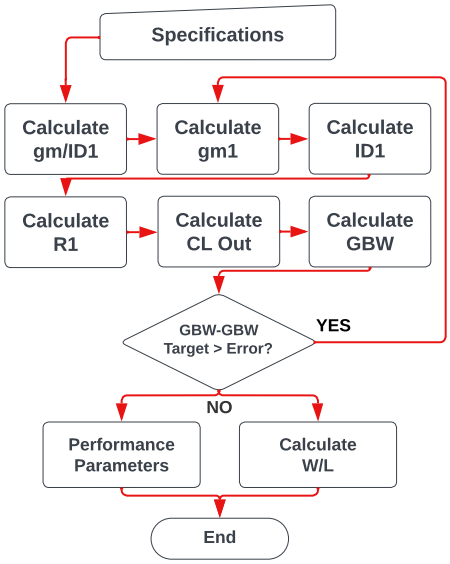


Fig. 5: Design flow for sizing the preamplifiers.

III. Implementation of the Design Procedure in Python Language

To automate the design of the preamplifiers, the design procedure was implemented in Python programming language according to the flowchart shown in Fig. 5.

The Python code is shown in Fig. 6. It uses the numpy module for mathematical operations.

The code starts with the declaration of desired specifications for the preamplifier (lines 2 to 7). In this case, the values of specifications in the code of Fig. 6 are related to the preamplifier 3. In lines 8 and 9 the calculation of gm/I_{D1} and the estimation of the total output capacitance are performed.

Lines 10 and 11 define two auxiliary variables for controlling the loop (go) and counting the number of iterations (i), respectively. The conditions for stopping the loop are specified in lines 12 and 13, namely the maximum error for the desired and the calculated GBW (error) and the maximum number of iterations (i_max). The iteractive loop to calculate GBW starts in line 14, after which the design procedure described in equations 3 to 7 is executed. The capacitance of resistor R1 is estimated by calculating the number of segments required to implement the required resistance (line 20). Since the capacitance per segment (variable CR1_seg) is known, the capacitance C_{R1} is calculated in line 24. With the new value of $C_{L(tot)}$ calculated in line 25 it is possible to estimate the new value of GBW in line 26.

The verification if the loop condition is satisfied is done in lilnes 28 to 31. At the end of the loop, the following information is obtained: the iteration number, the value of gm1, I_{D1}, $C_{L(tot)}$, Rsegments, R_1, C_{R1} and GBW.

The performance parameters of the circuit are estimated in lines 33 to 36 and the W/L of the PMOS input transistors is calculated in lines 37 and 38 through the gm/I_D x I_n curve.

```
import numpy as np
Av_target = -6  # V/V (target low-frequency voltage gain)
GBW_target = 300e6 # Hz (target gain-bandwidth product)
CL = 77e-15 # F (load capacitance)
VDD = 1.5   # V (supply voltage)
Vin_CM = 0.75 # V (common-mode input voltage)
Vout_CM_target = 0.75  # V (target common-mode output voltage)
gmid1 = -Av_target / (VDD - Vout_CM_target)
CLtotal = CL # Initial estimative of total load capacitance
go = True
i = 0
error = 5e6  # Maximum absolute error of calculated GBW relative to the desired GBW
i_max = 100  # Maximum number of iterations
while go:
    i = i + 1
    gm1 = GBW_target * CLtotal * (2 * np.pi)
    ID1 = gm1 / gmid1
    R1 = -Av_target / (gmid1 * ID1)
    # Number of R1 segments
    Rsegments = int(R1 / Seg_Res)
    # Estimated value of R1
    R1 = Rsegments * Seg_Res
    # Recalculate R1
    CR1 = Rsegments * CR1_seg
    CLtotal = CL + CR1
    GBW = gm1 / (CLtotal * 2 * np.pi)
    # New GBW considering CR1
    if abs(GBW - GBW_target) < error:
        go = False
    if i >= i_max:
        go = False
print('\n*** End of while ***\n')
IBIAS = ID1 * 2
Pdiss = IBIAS * VDD
Vout_CM = R1 * ID1
Av = -gmid1 * R1 * ID1
In = gmid_versus_In(gmid1)
WoverL = ID1/In
print('\n*** End of design procedure ***\n')
```

Fig. 6: Implementation of the design procedure in Python language.

IV. Simulation Results

After estimating the sizes and specifications of each preamplifier through the Python code, the sized circuits were simulated in Cadence Virtuoso tool. The estimated and simulated results are shown in Table III. It is possible to verify that the simulated results are very close to the estimated using the analytical procedure.

In order to verify the performance variability of the designed amplifiers, Monte Carlo simulations using 1000 runs were executed. Figures 7 shows the obtained histograms for GBW. The Monte Carlo simulations show a large variation of GBW. This is due to the resistor implemented in the circuit. Since R1 varies with the manufacturing processes, the GBW also vary. The mean value for preamplifier 1 is 308.33 MHz, for preamplifier 2 it is 306.00 MHz, and for preamplifier 3 it is 313.53 MHz. Furthermore, the corresponding standard deviations in 3σ are measured as 19.12 MHz, 24.87 MHz, and 21.61 MHz, respectively. The DC gain is also evaluated

TABLE III: Calculated and simulated performance parameters and devices sizes.

Parameter	Preamp 1		Preamp 2		Preamp 3	
	Analytic	Sim.	Analytic	Sim.	Analytic	Sim.
gm/I_{D1} ($V^{-}1$)	3.83	4.13	9.87	9.26	10.87	10.62
gm_1 (μS)	449.65	467.61	495.89	464.743	480.58	475.22
I_{D1} (μA)	117.51	113.18	50.25	50.15	44.24	44.73
GBW (MHz)	305.00	308.32	300.00	306.18	300.00	313.53
I_{BIAS} (μA)	235.02	226.37	100.50	100.30	88.38	
VGS_1 (V)	1.43	1.38	1.20	1.20	0.60	0.60
I_{n1} (μA)	4.13	3.97	0.73	0.72	1.00	0.98
W (μm)	17.07		41.25		26.54	
L (μm)	0.60		0.60		0.60	
R (kΩ)	16.97		14.85		16.97	

for each stage, preamplifier 1 we have an average gain of 17.03 dB (≈7.1 V/V) and a standard deviation in 3σ of 0.513 dB; for preamplifier 2 the average gain is 16.17 dB (≈6.4 V/V) with standard deviation of 0.735 dB; and finally the average gain of preamplifier 3 is 16.89 dB (≈7.0 V/V) and standard deviation of 0.567 dB.

V. Conclusion

This paper addressed the automation of a classic gm/ID design methodology implemented in the Python language for three common-source amplifier applied to a multistage comparator. The presented code aimed to reduce the design time and facilitate the use of the design method. It proved to be efficient, and despite the process variations caused by the resistor, the code worked perfectly. It was possible to have an accurate estimate of the circuit performance for each preamplifier with respect to the obtained simulated performance. The main contribution of this paper is the demonstration of an automatic design procedure implemented in Python language for the design of CMOS common-source amplifiers.

VI. Acknowledgments

The authors would like to thank Chipus Microelectronics (UNIPAMPA P&D number 31/2022) for supporting this work.

References

[1] Q. Zhang, N. Ning, J. Li, Q. Yu, Z. Zhang, and K. Wu, "A High Area-Efficiency 14-bit SAR ADC With Hybrid Capacitor DAC for Array Sensors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 12, pp. 4396–4408, 2020.

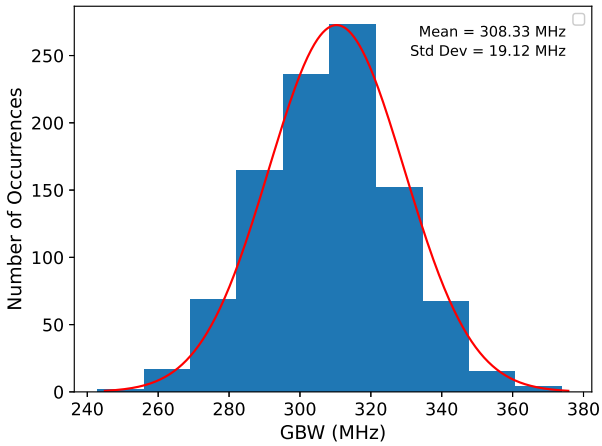
[2] Z. Zhang, L. Qiu, Y. Shan, and Y. Dong, "A 16-bit 8-MS/s SAR ADC with a foreground calibration and hybrid-charge-supply power structure," *IEICE Electronics Express*, vol. 17, no. 10, pp. 20 200 097–20 200 097, 2020.

[3] Q. Zhang, J. Li, Z. Zhang, K. Wu, N. Ning, and Q. Yu, "A Low-Power and Area-Efficient 14-bit SAR ADC with Hybrid CDAC for Array Sensors," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2019, pp. 1–5.

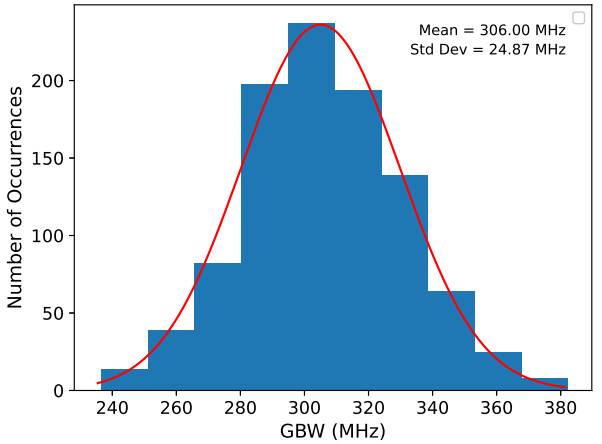
[4] F. Silveira, D. Flandre, and P. Jespers, "A g/sub m//I/sub D/ based methodology for the design of CMOS analog circuits and its application to the synthesis of a silicon-on-insulator micropower OTA," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 9, pp. 1314–1319, 1996.

[5] I. Abel and H. Graeb, "Automatic initial sizing of operational amplifiers with support of weak, moderate and strong inversion," in *ANALOG 2020; 17th ITG/GMM-Symposium*, 2020, pp. 1–4.

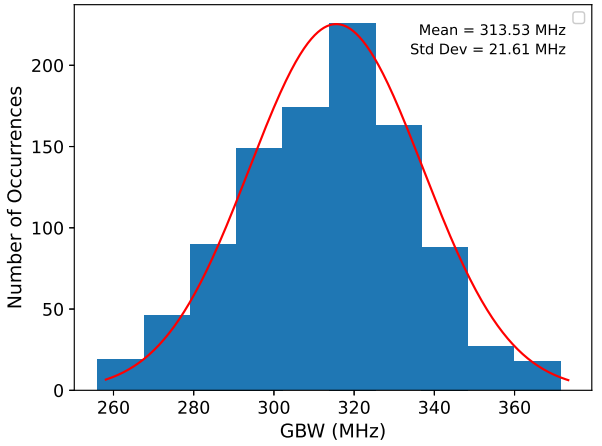
[6] A. A. Youssef, B. Murmann, and H. Omran, "Analog ic design using precomputed lookup tables: Challenges and solutions," *IEEE Access*, vol. 8, pp. 134 640–134 652, 2020.



(a) Preamplifier 1.



(b) Preamplifier 2.



(c) Preamplilfier 3.

Fig. 7: Monte Carlo simulation of the GBW for the three preamplifiers that compose the CMOS comparator.

[7] A. Ayed, H. Ghariani, and M. Samet, "Design and optimization of cmos ota with gm/id methodology using ekv model for rf frequency synthesizer application." in *2005 12th IEEE International Conference on Electronics, Circuits and Systems*, 2005, pp. 1–5.