POLITECNICO DI TORINO

MASTER'S DEGREE IN ELECTRONICS ENGINEERING

Master Degree Thesis

Design and silicon implementation of virtual voltage reference circuits

Supervisors
Prof. Paolo Crovetti
Roberto Rubino

Candidate
Gabriel Romero Rondon
Student ID: 288251

October, 2022

Acknowledgements

I would like to start highlighting my family as the main supporters of my career. To my brothers who always show me the path, both academically and spiritually. To the unconditional love that I received from my mother, and to my father who taught me to not be afraid of pursuing the goals I set. They all helped me to keep my spirit up and were present regardless of the physical distance that separates us.

I also want to express my deep gratitude to my girlfriend, for being present in the appropriate moment with the appropriate words. Sharing this experience at the Politecnico with her helped me to grow in many aspects of my life.

Special thanks to Angelica, being part of her family was a unique experience for me. Many thanks also to my friends in Turin who have been present and will continue being present in my life.

Finally, I would like to thank my academic advisors, Paolo Crovetti, Roberto Rubino and Luis Rueda. Their mentoring and confidence in my abilities was fulfilling.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Requirements in nowadays circuit applications

The requirements for the circuits and systems that are embedded within a System-On-Chip (SoC) have become more demanding with time. The following subsections focus in two particular applications that serve as an example of the challenges that these applications have to face. In this case, the focus is on biomedical engineering applications (specifically body dust) and the Internet of Things (IoT).

1.1.1 Biomedical engineering applications

Inside the field of biomedical sciences, there is a specific branch that focuses on developing microscopic devices that are able to measure and monitor variables in their environment; these devices are commonly known as Smart Dust. The term Smart Dust was first used in 1999 [3], and since then, several developments and products have been created to cover a wide range of applications. The applications range from measurement of weather conditions such as moisture and temperature [4] to their usage inside the human body as biosensors [5, 6].

Focusing on the biosensors field, literature has highlighted that, besides the requirement for microscopic sensors using microelectromechanical systems (MEMS), it is necessary to limit the area of the CMOS chips and their power consumption drastically [7]. The motivation behind the minimum area requirement is straightforward, considering that the expectation from a Smart Dust element is to minimize the possible discomfort that could create in the host; therefore, its dimensions must be maintained very low.

Fortunately, the development of smaller technology nodes is moving in the same direction as the low area requirement, allowing chips to have areas well below $1\,\mathrm{mm}^2$. Some examples of this miniaturization are analog-to-digital converters (ADC) with

areas around $0.05\,mm^2$ [8, 9], memory cells that occupy only $0.08\,mm^2$ [10] and amplifiers with an area of only $0.02\,mm^2$ [11]. These examples show that it is possible to have circuits with such low areas by dropping some of their performance in terms of number of bits (for the ADC) and memory size (for the SRAM).

Nevertheless, the constraint is not only on the area but also on the power consumption minimization. Considering that these elements are expected to be within the human body, it is recommended to avoid adding elements such as batteries that could represent a mechanical and chemical risk. Then, the batteries are usually replaced by energy harvesters which could be using some energy from mechanical vibrations (either from piezoelectric materials or MEMS) [12]. Thus, this shows how relevant it is for the system to minimize its power consumption because the power budget is minimal and intricate to harvest.

In conclusion, this particular example shows how power and area are critical in the present and the future of microelectronics applications.

1.1.2 IoT applications

Another application where the power and area are crucial for the success of the technology corresponds to IoT. Coincidentally, the first appearance of the term "Internet of Things" corresponds also to 1999 [13]. Then, with the new century, new technical advances arrived, making it possible to have many nodes connected through the cloud. This possibility also increased the research towards such promising technology, and such research allowed to identify which are the main requirements (and also bottlenecks) of IoT nodes [14]:

- User constraints: Cost, security
- Capabilities: Wireless, sensing, and processing
- Interaction: **Battery lifetime**, always connected
- Physical constraints: **Form factor (size)**, access to power

This classification of the requirements allows recognizing that both the area and power are a priority for improving the current IoT nodes that can be found on the market. Notice that, in this case, the classification refers to the battery lifetime of a node, considering that the reduction of the power consumption is crucial for having nodes whose batteries do not need to be often replaced. However, the constrained power budget applies not only to the battery-powered elements but also to the energy-scavenging nodes, where the power is also limited. Consider also that both the static and the dynamic power consumption of the IoT nodes have to be minimized. Given that many circuits nowadays contain a power-saving mode,

static power consumption will play a determinant role in the battery life for such circuits.

The list of the most relevant IoT requirements performed by Alioto [14] also highlights the physical shape of the IoT node as a relevant aspect. The reason behind the relevance of the physical shape lies in the fact that, in some cases, the nodes have to be located in environments that are sensitive to the presence of an external agent. Hence, the volume that occupies the node becomes relevant and such constraint pushes towards even smaller devices.

Therefore, a new example of an application where an SoC's area and power consumption are crucial has been discussed. This example justifies again the motivation behind the development of circuits and systems that are smaller and that consume less power.

In conclusion, the previous examples have shown that the market is always requiring devices that are faster and smaller while also keeping (or even increasing) the battery lifetime.

1.2 From analog towards mixed-signal circuits

Now that the reasons for prioritizing the area and power in the circuits have been discussed, it is relevant to analyse how it is possible to accomplish such improvements. As highlighted in section 1.1, semiconductor devices will continue shrinking. Last year's version of the International Roadmap for Devices and Systems (IRDS) supports this statement, where it is predicted that this trend will continue at least for one decade more [15]. At first sight, it might be considered that this automatically covers the lower area requirement discussed in the previous section. However, when the design involves analog components, the assumption of shrinking due to smaller technologies is not straightforward and cannot be taken for granted.

The impact of the shrinking on the performance of analog circuits has a mixed outcome since it brings both advantages and disadvantages. From the side of the advantages, the smaller channel length improves the speed of the devices. Thus the intrinsic transit frequency (f_T) increases. On the other hand, the transistors' intrinsic gain decreases, deeply affecting some specific circuits whose operating principle is based on a high gain amplifier. To give some idea of what is gained versus what is lost, when moving from the 0.5 µm technology node towards the 22 nm node, f_T increases by a factor of 25 (16 GHz to 400 GHz) whereas the intrinsic gain drops by 30x (from 180 to 6 V/V) [16].

Also, the miniaturization of the devices is closely related to the scaling of the supply voltages of the circuits. Using the same transition as before, from 0.5 µm to 22 nm channel length, the voltage supply went from 5 V to 1 V [16]. This means that there is a reduction of the available voltage swing used for representing analog signals. Besides, there is a critical degradation of the signal-to-noise ratio of the analog signals since the voltage ranges that are available for carrying information have shrunk.

In order to overcome these issues, analog designers have to rely on bold techniques that allow having circuits that may operate under extremely low voltage supplies [17] and also using very small technology nodes, e.g. 14 nm [18]. Nevertheless, such techniques are becoming more scarce as technology moves towards even smaller devices.

On the other hand, in parallel to the usual approach of redesigning the analog circuits as a new technology node arrives, there is another way to solve the scaling problem: by using more digital circuits for functions usually performed by analog circuits [19, 20]. This trend has been well adopted, and it is possible to find several examples of such circuits, for instance, synthesizable digital to analog converters (DACs) [21, 22] and analog to digital converters (ADCs) [23, 24] as well as voltage regulators that are composed by digital elements [25, 26]. The same concept can be extrapolated to the system level, and nowadays, it is already possible to find radio-frequency (RF) front-ends that allow having a mostly-digital system [27, 28].

The usage of digital circuits for solving analog problems presents a new advantage related to the fact that the design flow of digital circuits is well automated. Even if some efforts have been concentrated on creating an automatic layout generator for analog circuits [29, 30, 31], the large number of degrees of freedom involved in the creation of an analog layout presents an obstacle to the automation of such practice. In addition, the automated design flow for digital circuits has been on the market for decades, which means that it could also take many years for an analog layout generator tool to arrive at a level of performance that is comparable to the level of the current digital design flow tools.

Figure 1.1 shows a parallel between the steps that are involved in the flow of the design of digital circuits (on the left) and the steps for designing analog circuits (on the right). Even though the analog flow appears to be shorter, it has a bottleneck related to the physical implementation since circuit designers must perform this process manually. In contrast, when referring to the physical implementation of a digital circuit, there is the possibility of relying on an algorithm that performs the placement and routing of the circuit. This advantage drastically decreases the time required before having a physical circuit version, which may require several

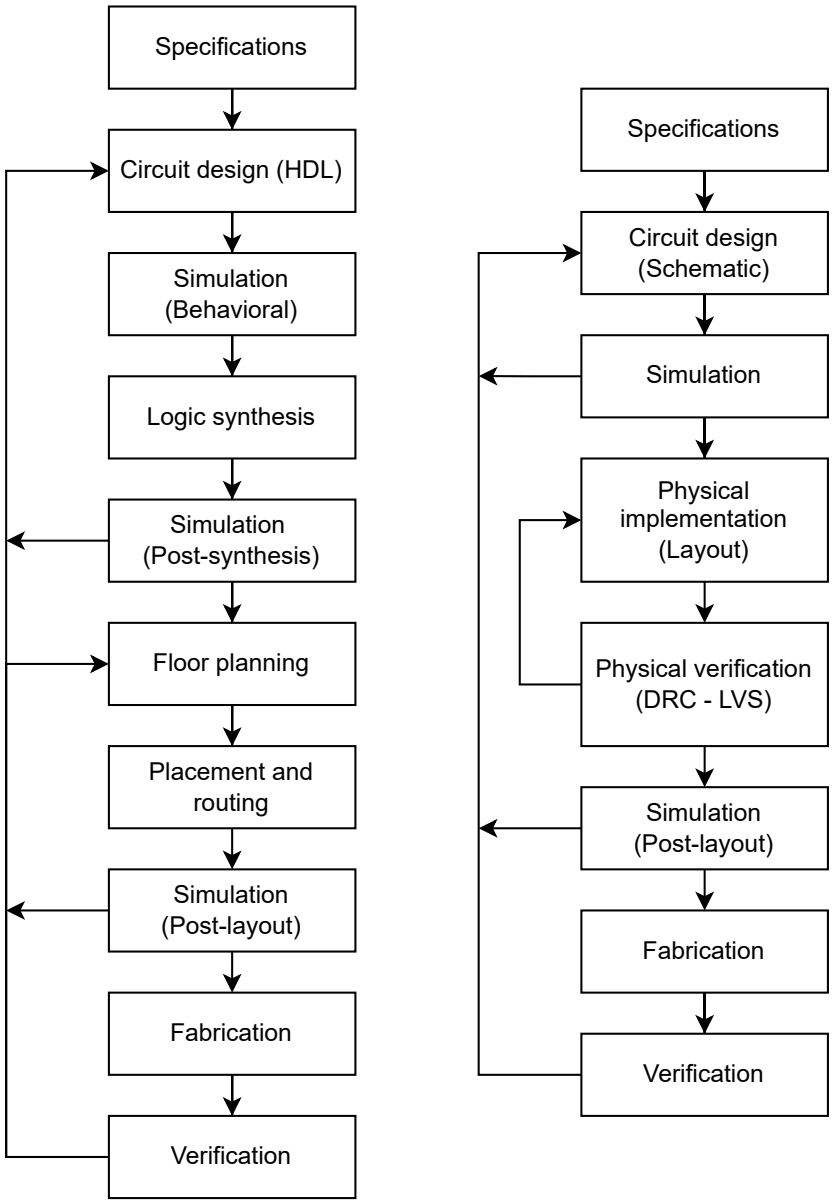


Figure 1.1. Digital (left) and analog (right) design flow. Adapted from [1]

iterations as depicted in Figure 1.1.

Furthermore, digital circuits do not usually have the problem of portability towards new technology nodes. Easy portability is another advantage that drastically reduces the time required for a product to be launched using a new technology node. Since the HDL design adapts much better to the changes in the transistors than

the analog circuit schematic, it is more convenient to have a design contained in an HDL format that can be synthesized to different technology nodes.

In conclusion, moving from the analog towards the digital (or mixed-signal) domain poses as the solution for overcoming the bottlenecks that the analog circuit design is experiencing.

1.3 Voltage references in a SoC

Both analog and digital circuits require voltage references in order to work correctly. Several circuit examples mentioned in the previous sections correspond to circuits that use a voltage reference as one of the fundamental components. Both ADCs and DACs use voltage references to compare and then convert signals between the digital and the analog domain; also, operational amplifiers and voltage regulators rely on voltages (and also currents) that are relatively constant.

The expectation from a voltage reference is that it provides an exact voltage that does not vary (in theory) with respect to process variations, supply voltage, and temperature (PVT). This means that the variation with respect to these variables should be bounded within a specific range. Voltage references have to be distinguished from the voltage regulators from the characteristic that the regulators are expected to provide a much larger current than the voltage references. Then, the specification for the variation of a voltage regulator is not as strict as it is for a voltage reference. As a matter of fact, a voltage regulator usually contains a voltage reference embedded since the regulators require a voltage that is constant to be used as a physical standard for the creation of the regulated voltage.

For the applications mentioned in sections 1.1.1 and 1.1.2 it is possible to notice that circuits such as ADCs, DACs, operational amplifiers, and voltage regulators are usually part of those types of systems. This plurality of applications demonstrates that voltage references are a fundamental part of the systems being designed for both IoT and biomedical applications. Moreover, the consequence is that the voltage references must then be compliant with the constraints regarding ultra-low power consumption and a very small circuit area.

This document discusses the design of a voltage reference circuit based on the concept of the virtual voltage reference. chapter 2 discusses the operating principle of a voltage reference, the usual analog implementation, and the figures of merit used for comparing different designs of voltage references. Also, this chapter introduces the concept of the virtual voltage reference and the algorithm involved in the realization of this type of voltage reference. Then, chapter 3 shows what is new in this implementation of the virtual voltage reference. This chapter also discusses

the main challenges involved in the design of a virtual voltage reference and how such challenges have been tackled. The following chapter (chapter 4) contains the details regarding the design of this particular voltage reference, along with the assumptions performed and the final physical implementation of the circuit, i.e., the layout view. Finally, chapter 5 represents the results obtained using this voltage reference and compares them with respect to the state of the art.

Chapter 2

Voltage reference circuits

After discussing the relevance of voltage references in section 1.3, this chapter shows how it is possible to design voltage references. Also, the characteristics that are used for measuring the performance of the voltage references are enumerated here. These characteristics allow performing some comparison between different solutions, and such solutions will be discussed in the section that describes the state of the art of low power voltage references. Finally, the concept of virtual voltage reference is also presented as an alternative that could be used to have a mixed-signal solution for a voltage reference.

2.1 Figures of merit for comparing voltage references

Several parameters are measured to assess the behaviour and performance of this type of circuit based on what is expected from a voltage reference. Considering that the main requirement of a voltage reference is to provide a voltage that does not depend on factors such as voltage and temperature, the two main parameters are the line regulation and the temperature coefficient. Furthermore, the impact of dynamic variations of the supply voltage on the voltage reference is measured through the power supply rejection ratio (PSRR). Finally, considering that, nowadays, the reduction of power consumption is a priority, the quiescent current of the circuit is another figure of merit that allows comparing different designs in terms of power.

Another relevant figure of merit to be compared in the different designs is the area, but this parameter does not need to be extended in this section. In addition, although the robustness with respect to process and also the time drift of a voltage reference are both relevant, they are more challenging to measure and compare.

Therefore, they will not be considered reference parameters in this analysis of voltage references.

Figure 2.1 summarizes the main parameters used for assessing the performance of a voltage reference. Two parameters are related to the supply voltage (V_{DD}): the line regulation and the PSR. The quiescent current (I_Q) is also represented in the figure as the current sourced from the supply and does not flow through the load. Finally, the temperature coefficient of the voltage reference is also shown as the parameter that characterizes the variation of V_{REF} for different temperatures.

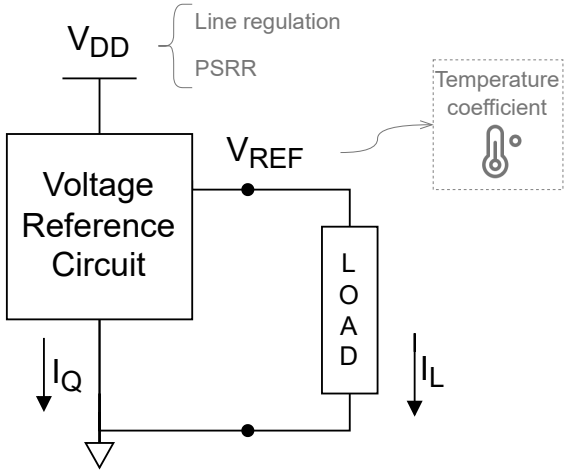


Figure 2.1. Voltage reference and its main parameters.

The following sections discuss the figures of merit of voltage references and also show how they are usually measured in the literature.

2.1.1 Line regulation

The line regulation represents the output voltage variation as a function of a variation in the input voltage (V_{DD}). Then, this parameter represents the sensitivity of the reference voltage with respect to the supply. In some cases, the units of measurement correspond to V/V; in others, the line regulation is measured in ppm/V or %/V. In this document the second approach will be used; this means that the nominal value of the reference (V_{REF}) will be in the denominator. Then, the formula for the line regulation is:

$$LR\ [\%/\mathrm{V}] = \frac{\Delta V_{REF}}{\Delta V_{DD} \cdot V_{REF}} \cdot 100\% = \frac{V_{REF\_max} - V_{REF\_min}}{\Delta V_{DD} \cdot V_{REF}} \cdot 100\% \qquad (2.1)$$

Where V_{REF} corresponds to the nominal voltage that is expected at the output of the voltage reference, ΔV_{REF} represents the variation of the reference voltage,

and ΔV_{DD} corresponds to the range that is swept for the supply voltage. In case the line sensitivity is measured in ppm/V, 100% is replaced by 10^6

In some cases, the range of V_{DD} that is considered for the line regulation is another parameter since it is expected for a voltage reference to cover a wide range of voltage (more than hundreds of millivolts). Besides, the minimum V_{DD} is also relevant for some applications where the available supply voltage is a relevant constraint. Then, the minimum supply voltage under which the voltage reference can still provide a regulated output is also a relevant figure of merit.

2.1.2 Temperature coefficient

As previously highlighted in Figure 2.1, the temperature conditions under which the voltage reference operates drastically affect its behaviour. Then, it is usually expected that the voltage at the output of the reference varies when the temperature deviates from the nominal temperature. This behaviour of the circuit is measured under the temperature coefficient.

$$TC\ [\text{ppm/°C}] = \frac{\Delta V_{REF}}{\Delta T \cdot V_{REF}} \cdot 10^6 = \frac{V_{REF\_max} - V_{REF\_min}}{\Delta T \cdot V_{REF}} \cdot 10^6 \qquad (2.2)$$

As it can be noticed, the equation is very similar to the equation of the line regulation. However, it is far more usual to represent the temperature coefficient in $ppm/°C$. The reason behind such units is that usually, the expectation is that the temperature drift of a voltage reference is very well compensated.

It is relevant to highlight that this is an essential parameter because it usually also defines the accuracy of the circuits that use a voltage reference. For instance, when assessing the bevaviour of an ADC for a particular temperature range, the expectation is that the temperature drift of the voltage reference is minimal to avoid propagating such error through the analog-to-digital conversion chain.

2.1.3 Power supply rejection (PSR)

The supply voltage provided to almost any type of circuit contains some spurious high-frequency components. The reason behind such a problem could be the digital circuitry that is constantly toggling and is affecting the power rail, or it could also be due to the intrinsic behaviour of switched power converters that could be supplying this voltage reference.

The purpose of a voltage regulator is to be robust against such high-frequency noise and avoid propagating such errors to the regulated voltage. This dynamic figure of merit is called power supply rejection ratio (PSRR), and it can be seen as

the gain at the circuit's output where the input is the supply voltage. Since this is a dynamic parameter, it is defined for a given range of frequencies, and it is also given by the following expression:

$$PSR(f)\ [\text{dB}] = 20 \cdot \log \frac{V_{REF}(f)}{V_{DD}(f)} \tag{2.3}$$

Given that this parameter can be seen as a gain, it is usually represented in decibels.

Digital circuits are known to be more robust to noise and power supply variations than their analog counterparts. However, since this corresponds to a continuous circuit built using digital components, such robustness is not the same as a circuit that performs digital processing tasks.

2.1.4 Quiescent current

Power consumption is a crucial parameter in circuits nowadays explains why highlighting the quiescent current (I_Q) of voltage references as a figure of merit. Figure 2.1 shows the path through which the current I_Q flows in the circuit. As it can be noticed, this current is drawn from the supply but is not used for supplying the load.

A possible way of measuring I_Q corresponds to setting a load with infinite resistance ($R_{LOAD} \to \infty$), since this will make $I_L \approx 0$, and then measure the current that is being provided to the whole circuit.

Then, if it is necessary to calculate the power consumption of the voltage reference, this can be obtained by doing the following:

$$P_{steady\ state} = V_{DD} \cdot I_Q \tag{2.4}$$

Moreover, this corresponds to another figure of merit that has to be minimized when it is necessary to consider the constrained power budget.

2.2 Bandgap voltage references

Among the different figures of merit discussed in the previous section, one of the main bottlenecks of voltage references in the previous century was achieving temperature independence. By that time, it was clear that the way of accomplishing such independence relied on combining elements (or primitives) that provided temperature coefficients that were complementary to each other. Such complementary

behaviour could compensate for the temperature drift of the circuits. It was also clear that the voltage through a diode biased by a constant current decreases with temperature. However, finding the opposite behaviour of a primitive (i.e., a voltage or current) that increases with temperature was not trivial.

Then, Widlar demonstrated in Fairchild Semiconductor that it was possible to achieve a positive temperature coefficient by obtaining the difference between the junctions of two diodes that are biased at two different current densities [32]. This finding encouraged the creation of many different topologies of bandgap voltage references.

The circuit is known as bandgap voltage reference because, in theory, the output voltage corresponds to the bandgap voltage of the semiconductor used for building the circuit (around 1.2 V for Silicon).

Figure 2.2 summarizes the principle of the operation of bandgap voltage references. The negative temperature coefficient is also known as the complementary to absolute temperature (CTAT) coefficient. In this figure, the CTAT behaviour is obtained from the base-emitter voltage of a diode-connected BJT transistor. On the other hand, the proportional to absolute temperature (PTAT) response is acquired from the difference between the base-emitter voltages of two diode-connected BJT transistors. This quantity will be proportional to the thermal voltage (V_T). Then, by combining the PTAT and the CTAT behaviour, temperature independence can be achieved:

$$\frac{\partial V_{REF}}{\partial T}(T) = m_1 \cdot \frac{\partial V_{PTAT}(T)}{\partial T} + m_2 \cdot \frac{\partial V_{CTAT}(T)}{\partial T} \approx 0 \tag{2.5}$$

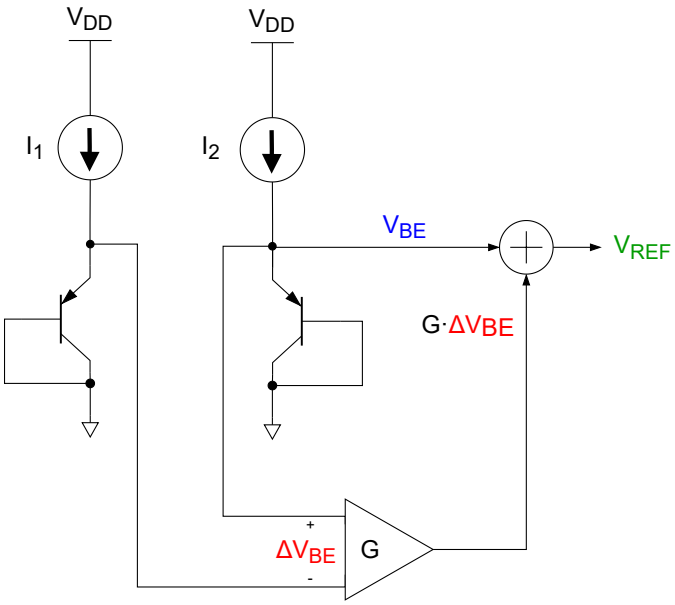
Since, by definition:

$$\frac{\partial V_{PTAT}(T)}{\partial T} > 0 \quad ; \quad \frac{\partial V_{CTAT}(T)}{\partial T} < 0 \tag{2.6}$$

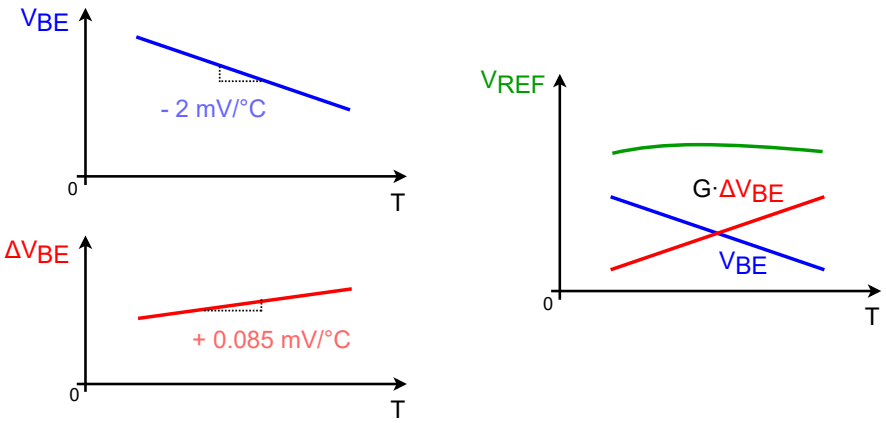
Then, the main task for designing the bandgap voltage reference is to choose the values of m_1 and m_2 correctly. This allows to achieve the expected temperature compensation, i.e., $\frac{\partial V_{REF}}{\partial T} \approx 0$. It is important to highlight that this temperature compensation usually can only be achieved at a nominal temperature value, then, this compensation is often referred as first-order temperature compensated.

It is relevant to highlight also that due to the nature of the primitives, there is a significant mismatch in the absolute values of the temperature coefficients.

In the case of the base-emitter voltage (V_{BE}), it can be evaluated as:



(a)



(b)

Figure 2.2. Bandgap voltage reference operating principle.

$$V_{BE} = V_T \cdot \ln\left(\frac{I_C}{I_S}\right) \tag{2.7}$$

Where V_T is the thermal voltage ($= KT/q$), I_C is the collector current, whereas I_S is the saturation current of the BJT. Then, the temperature coefficient of V_{BE} corresponds to:

$$\frac{\partial V_{BE}\,(T)}{\partial T} = \frac{V_{BE} - (4+m)\cdot V_T - E_g/q}{T} \approx -1.5\,\mathrm{mV/°C} \tag{2.8}$$

The factor m corresponds to approximately -1.5, and E_g is the bandgap energy. Equation (2.8) clearly shows that the behavior of the voltage V_{BE} is complementary to the temperature increment. Then, considering a silicon BJT, we obtain a very typical value of the temperature coefficient of $-1.5\,\mathrm{mV/°C}$.

On the other hand, for the PTAT voltage, it is necessary to consider the difference between the base-emitter voltages of two different BJT transistors (ΔV_{BE}). Also, the current that flows through these devices is n times larger in one device than in the other. Using Equation (2.7) we can define ΔV_{BE} as follows:

$$\Delta V_{BE} = V_T \cdot \ln\left(\frac{n\cdot I_C}{I_S}\right) - V_T \cdot \ln\left(\frac{I_C}{I_S}\right) \tag{2.9}$$

$$\Delta V_{BE} = V_T \cdot \ln(n) = \frac{KT}{q}\cdot \ln(n) \tag{2.10}$$

Then, if we consider the temperature coefficient of this primitive:

$$\frac{\partial \Delta V_{BE}\,(T)}{\partial T} = \frac{K}{q}\cdot \ln(n) \approx 0.087 \cdot \ln(n)\ \mathrm{mV/°C} \tag{2.11}$$

By comparing equations (2.8) and (2.11) it is possible to observe not only that the two temperature coefficients have opposite signs but also that the difference between them is considerable (almost two orders of magnitude). Then, this drives towards the idea of using a value of n that is sufficiently large to cover such differences. However, for the values mentioned above, it would be necessary to use $n = 2.9\times 10^7$, which is impractical considering that this parameter is related to the area of one BJT device with respect to the other.

Then, as an alternative, some authors added some features to the circuit principle proposed by Widlar to add a degree of freedom that allows increasing the gain of the PTAT component. Figure 2.3 shows the architecture of Brokaw's proposal (left) [33] and Kuijk's bandgap [34] (right). As shown in the figure, the addition of an operational amplifier (OPAMP) allows having the desired gain to the ΔV_{BE} voltage. Such OPAMP was not present in Widlar's design [32].

Focusing, for instance, on the case of the Kuijk bandgap, from circuit analysis, we know that:

$$V_{REF} = V_{BE_1} + I_2' \cdot R_2 \tag{2.12}$$

Nonetheless, if the current I_3 that enters the operational amplifier is neglected, i.e., assuming an ideal OPAMP, the current I_2' is:

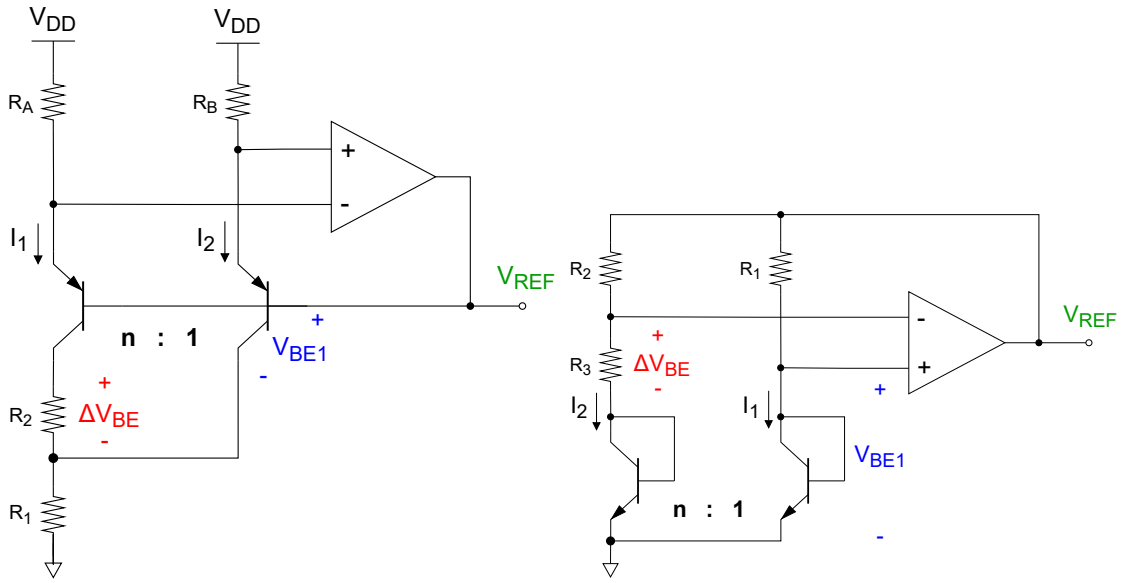


Figure 2.3. Brokaw and Kuijk bandgap voltage reference architectures

$$I_2' = \frac{\Delta V_{BE}}{R_3} \tag{2.13}$$

And combining the last two equations:

$$V_{REF} = V_{BE_1} + \frac{R_2}{R_3} \cdot \Delta V_{BE} = V_{BE_1} + \frac{R_2}{R_3} \ln\left(\frac{R_2 I_{S_2}}{R_1 I_{S_1}}\right) \cdot V_T \tag{2.14}$$

Notice that the previous equations contains both the CTAT and PTAT terms of equations (2.7) and (2.9) respectively. Also, the PTAT component is multiplied by a factor: R_2/R_3, which reduces the need for such a large difference between the area of the BJT devices.

The final equation of the voltage reference proposed by Brokaw is similar to Equation (2.14). The reference voltage (V_{REF}) in Brokaw's case corresponds to:

$$V_{REF} = V_{BE_1} + \frac{2R_1 \ln(n)}{R_2} \cdot V_T \tag{2.15}$$

Then, it is evident that even though the circuit elements and their connections are not the same, these two circuits share some fundamental similarities; for instance, their primitives are ultimately the same (V_{REF} and ΔV_{REF}). The primitives refer to the physical quantities provided by the circuit and used for obtaining the desired behaviour of the voltage reference. In this case, the objective of the

primitives corresponds to providing complementary behaviours with respect to temperature. Such behaviour allows to achieve temperature independence.

After observing these examples of bandgap voltage references, it is possible to summarize the concept by separating between three primary units that compose voltage reference circuits.

- A **physical element**, that corresponds to a semiconductor device that creates the voltage primitives that the processing element will use. In the examples of Figure 2.3, the physical elements correspond to BJT devices (Q_1 and Q_2).
- A **signal processing element** that is in charge of using the primitives in order to generate the output voltage of the reference. For the examples shown before, the signal processing element is the OPAMP that performs the addition among the PTAT and the CTAT primitives.
- A **bias network**, that is in charge of biasing the physical elements. The expectation is that this biasing should be supply-independent, as it occurs for current of Equation (2.13). The bias network of both bandgap references shown in Figure 2.3 corresponds to the currents that are set through the branches. Notice that, in both cases, the output stage of the OPAMP is in charge of setting such currents in each of the BJT branches.

This generalization of a voltage reference is summarized in Figure 2.4. This figure shows how the fundamental blocks of a voltage reference interact to provide a voltage at the output that does not depend on process, voltage, and temperature. This abstraction of a voltage reference is a powerful concept that will be used later in this chapter (section 2.3) to introduce the principle of the virtual voltage references.

2.3 Virtual voltage references

Voltage reference circuits are not excluded from the trend of moving some building blocks of traditionally analog circuits into the digital domain, as discussed in the previous chapter. According to the abstraction of a voltage reference presented in Figure 2.4, it is possible to move the bias network and the signal processing to the digital domain. Moving the physical standard to the digital domain is not a trivial task since the expectation from the primitive is to provide a continuous voltage (or current) that is then processed. Then, this section discusses the implementation of a voltage reference that is composed of a digitally-implemented bias network and signal processing unit; this is known as a virtual voltage reference.

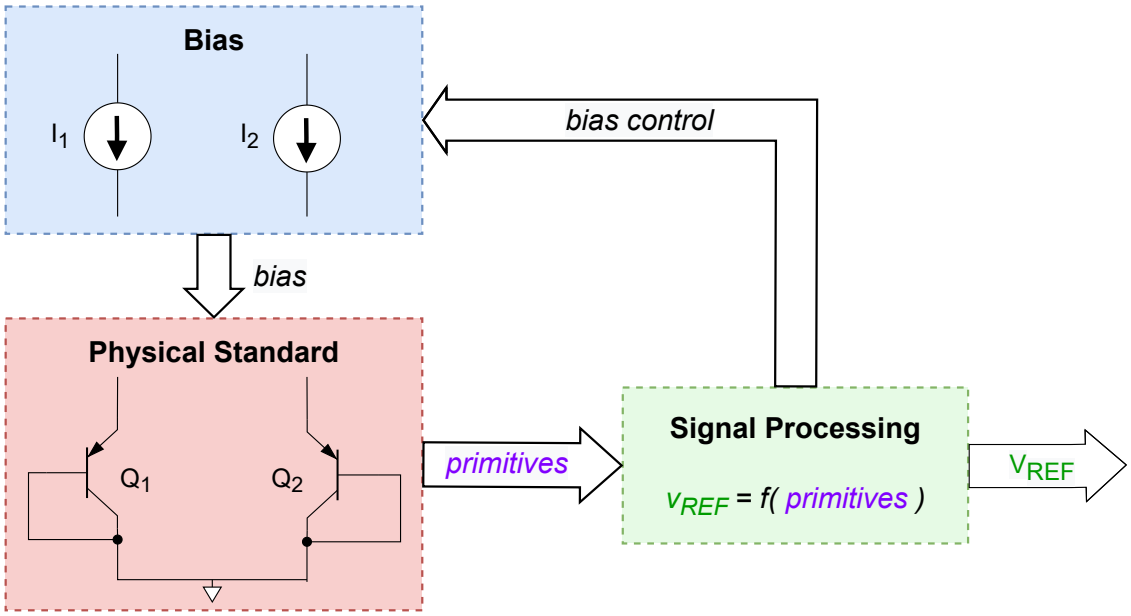


Figure 2.4. Block diagram of a general voltage reference

Figure 2.5 depicts the block diagram of a virtual voltage reference. In this case, the three main components can be identified as follows:

- The **physical element** corresponds to a **diode** that is being biased by a certain current i_D.
- The **signal processing element** undoubtedly corresponds to the **digital processor**. In order to obtain the primitives from the diode, an ADC is required for converting the voltage v_D into a digital word that the digital processor then uses.
- Finally, the **bias network** is represented by a **DAC** that biases the diode according to the value provided at the digital processor's output.

Similar to the case of the bandgap voltage references discussed in section 2.2, the purpose of this virtual voltage reference is to create two different current densities through p-n junctions. Then, once these current densities are created, the voltage present in the p-n junction is measured for both cases. Finally, these voltages are used for building the primitives: v_D and Δv_D. Notice that the obtained primitives are CTAT and PTAT, respectively. This means that the digital processor can use the values to build a temperature-independent digital value.

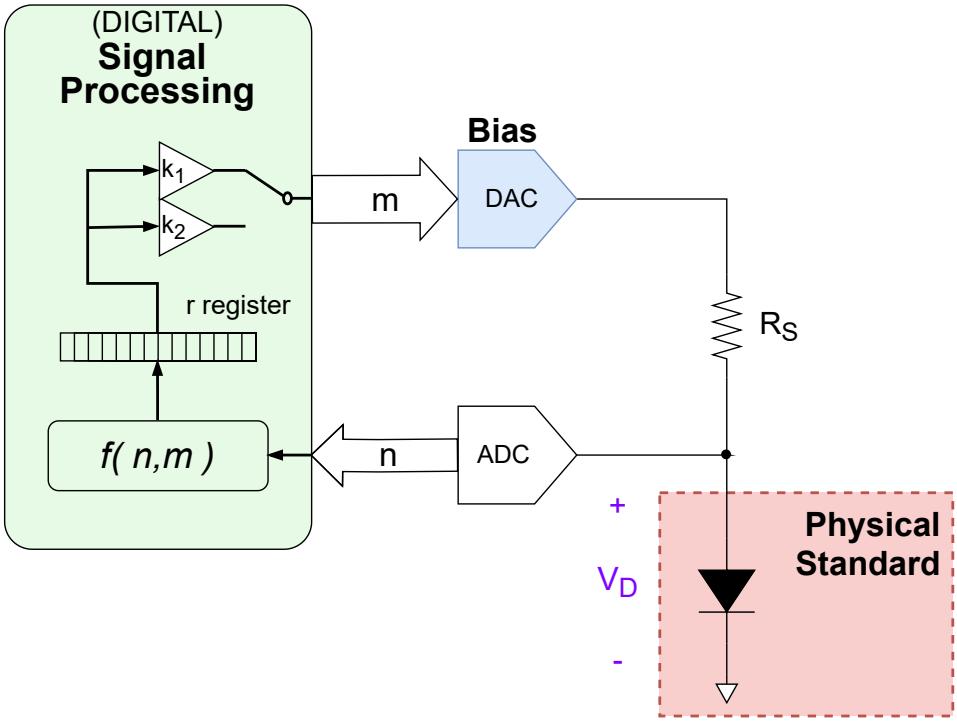


Figure 2.5. Virtual voltage reference block diagram

It is worth highlighting that this architecture does not impose any specific constraints regarding the supply voltage (V_{DD}). Then, this could be a temperature-dependent voltage source. The only constraint for the supply voltage is that its value needs to be constant for a certain period while the virtual reference algorithm is being executed.

The following sections describe in detail the virtual voltage reference's operating principle and the algorithm used for implementing this type of reference.

2.3.1 Theoretical analysis of the virtual reference

The DAC of the circuit is able to create a voltage at its output that depends on the N-bit digital word m:

$$v_{DAC} = \frac{m}{2^N} \cdot V_{DD} \tag{2.16}$$

In parallel, the voltage at the diode (v_D) can be obtained from the digital value at the N-bit ADC output:

$$v_D = \frac{n}{2^N} \cdot V_{DD} \tag{2.17}$$

Notice that the previous equations neglect the quantization error of both the DAC and the ADC. Then, using these values, it is possible to calculate the current that flows through the diode:

$$i_D = \frac{v_{DAC} - v_D}{R} = \frac{m - n}{2^N \cdot R} \cdot V_{DD} \tag{2.18}$$

Then, using this expression for both measurements allows calculating the ratio of the current densities of the diode (h). This value is used for correction with respect to an intrinsically fixed current density ratio of the diode (h^*).

$$h = \frac{{i_D}^{(1)}}{{i_D}^{(2)}} = \frac{m^{(1)} - n^{(1)}}{m^{(2)} - n^{(2)}} \tag{2.19}$$

Where ${i_D}^{(1)}$ represents the current that flows through the diode when the input of the DAC is $m^{(1)}$ and the output of the ADC is $n^{(1)}$. And ${i_D}^{(2)}$ is analogous to ${i_D}^{(1)}$, but when the second value ($m^{(2)}$) is applied.
Then, it is possible to define the PTAT primitive as:

$$\Delta v_D = \frac{\log h^*}{\log h} \cdot \left({v_D}^{(1)} - {v_D}^{(2)}\right) \tag{2.20}$$

Finally, if the primitives are combined, the reference voltage (v_{REF}) can be defined as:

$$\begin{aligned} v_{REF} &= {v_D}^{(1)} + \alpha \Delta v_D \\ v_{REF} &= {v_D}^{(1)} + \alpha \frac{\log h^*}{\log h} \cdot \left({v_D}^{(1)} - {v_D}^{(2)}\right) \\ v_{REF} &= \left[n^{(1)} + \frac{\alpha \log h^*}{\log h}\left(n^{(1)} - n^{(2)}\right)\right] \cdot \frac{V_{DD}}{2^N} \end{aligned} \tag{2.21}$$

Where α represents the factor used for achieving the temperature cancellation, i.e., for matching the temperature coefficients of opposite sign. The quantity in the square brackets of Equation (2.21) can be calculated by using the variables that are available to the digital processor: $m^{(1)}$, $m^{(2)}$, $n^{(1)}$, $n^{(2)}$ (see also Equation (2.19)). This means that it is straightforward to calculate this value since the digital processor can perform such arithmetic operations.

The quantity in the square brackets of equation Equation (2.21) is a crucial variable in this type of voltage reference since this factor is expected to be a digitally-calculated reference value. Thus, this quantity is known as the **virtual reference** (r):

$$r = f(m^{(1)}, m^{(2)}, n^{(1)}, n^{(2)}) = n^{(1)} + \frac{\alpha \log h^*}{\log h}\left(n^{(1)} - n^{(2)}\right) \tag{2.22}$$

The virtual reference has the characteristic of counteracting the variations present in V_{DD} (the unregulated and PVT-dependent source). Then, by replacing Equation (2.22) into Equation (2.21), it is clear that a PVT-independent voltage reference is achieved:

$$v_{REF} = \frac{r}{2^N} \cdot V_{DD} \tag{2.23}$$

An essential point is the discussion regarding where the reference voltage v_{REF} can be observed in the circuit. To show the observability of v_{REF}, the result of Equation (2.23) is replaced into Equation (2.16), obtaining:

$$v_{DAC} = \frac{m}{r} \cdot v_{REF} \tag{2.24}$$

This result shows that by choosing $m = r$, the reference voltage v_{REF} is observed at the output of the DAC.

Another feature that provides the virtual voltage reference is the possibility of scaling the output voltage v_{REF} by a constant k set inside the digital controller. Then, if the parameter m gets assigned a multiple of r ($m = k \cdot r$), the output of the DAC corresponds to a PVT-independent voltage scaled by a specific factor k:

$$v_{DAC} = \frac{k \cdot r}{r} \cdot v_{REF} = k \cdot v_{REF} \tag{2.25}$$

This is a significant advantage that is provided by virtual voltage references with respect to bandgap voltage references that are limited to values that are very close to the bandgap of the semiconductor (as discussed in section 2.2).

2.3.2 The virtual reference algorithm

The procedure to calculate the value of the virtual reference (r) as in Equation (2.22) is not trivial. The approach discussed by Crovetti [35] corresponds to an iterative algorithm that performs several calculations of r and expects the convergence of r after a few iterations (typically two or three).

Figure 2.6 describes the iterative algorithm used in [35] for calculating r. Notice that the value of r is also used for setting the values of $m^{(1)}$ and $m^{(2)}$ for the next iteration. Whereas the first value of r consists of an initial guess used as a starting point for the algorithm. In the figure, the symbol $\lfloor \cdot \rceil$ refers to rounding to the

nearest integer.

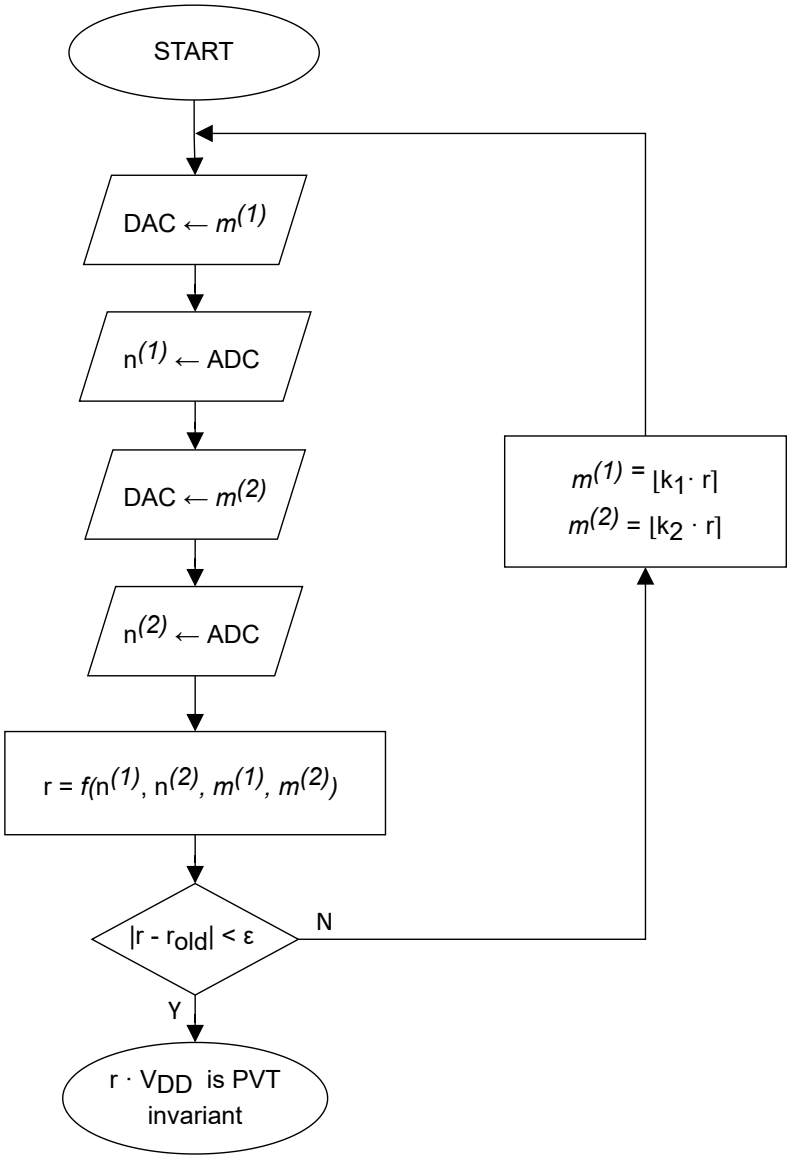


Figure 2.6. Algorithm for calculating the virtual reference

The reason behind setting m as a multiple of r ($m = \lfloor k \cdot r \rceil$) lies in the fact that in this way, the bias current of the diode becomes independent of V_{DD}. This can be noted when combining Equation (2.25) into Equation (2.18), leading to:

$$i_D = \frac{k \cdot v_{REF} - v_D}{R} \tag{2.26}$$

Furthermore, this behaviour is an analogy to the behaviour observed in the bandgap voltage references discussed in section 2.2. Equation (2.13) shows the case of the V_{DD} independent current in the bandgap of Kuijk [34], and a similar principle allows to have a supply-independent bias in the case of Brokaw's bandgap [33].

The algorithm then consists of setting two values of m for the DAC to convert them ($m^{(1)}$ and $m^{(2)}$) and then use the ADC to sample the voltage of the physical element, obtaining $n^{(1)}$ and $n^{(2)}$. Then, Equation (2.22) is used for calculating the value of r. The new value of r sets to two new values of m and then the loop continues until convergence is obtained, i.e. $|r - r_{old}| < \epsilon$. And the final value of r corresponds to the virtual voltage reference.

The advantage of using a digital processor as the signal processing element is that the current algorithm could be easily modified. This particular implementation of the virtual reference concept contains several degrees of freedom that make the design flexible and adaptable to possible changes, either by changing the constants of the algorithm (such as k_1, k_2, ϵ) or by modifying the algorithm itself. The intrinsic flexibility of the digital circuits cannot be achieved by their analog counterparts.

Chapter 3

Virtual voltage reference using the discharge of a capacitor

After discussing the concept of virtual voltage reference, it is possible to go further into the details of the design of a voltage reference that relies on this principle. This chapter shows the main design challenges in developing this type of circuits, while also introducing the advantages and disadvantages with respect to their analog counterparts.

The purpose of this specific implementation of the virtual reference is to show that adopting the virtual reference concept brings a significant advantage in terms of generality. In contrast to what occurred in the last century in the early stages of the design of bandgap voltage references (see section 2.2), virtual voltage references pose as an alternative that allows incorporating a new set of primitives that could be obtained from the circuits. This advantage is attributable to the possibility of converting those new primitives to the digital domain and then performing the analysis of such primitives using a digital processor.

In addition, the following example of the virtual voltage reference shows another precedent that proves that it is possible to model a circuit in order to obtain a voltage reference. Notice that this model should consider all the necessary parameters to create an output independent of voltage and temperature variations. This new example reinforces the ideas discussed by Crovetti [35]. In that work, an equation was proposed for using the voltage drop of a diode as a primitive

3.1 Selecting the physical reference for the virtual voltage reference

For this specific implementation, the primitive obtained for the virtual voltage reference corresponds to the discharge time of a capacitor through a network composed of MOSFET devices. The specific reasons for choosing such a circuit for obtaining the primitive will be presented in chapter 4.

Figure 3.1 depicts the idea for the primitive that is used in this design of the virtual voltage reference. Figure 3.1a shows the circuit that must be implemented to control the capacitor's charging and discharge. An initial voltage ($V_{initial}$) is set, and the capacitor is charged up to this value through the source resistance R_S. Then, both switches toggle, allowing the capacitor (C) to discharge through a certain discharge network. The current through the discharge network is represented as i_{dis}, whereas its voltage is v_{dis}.

Similarly, Figure 3.1b illustrates the transient behaviour that is expected for the voltage $v_{dis}(t)$. This figure also highlights the threshold voltage (V_{thres}) that is used for measuring the discharge time ($t_{discharge}$). Then, this is the only parameter that is obtained from the circuit and that allows building a voltage reference.

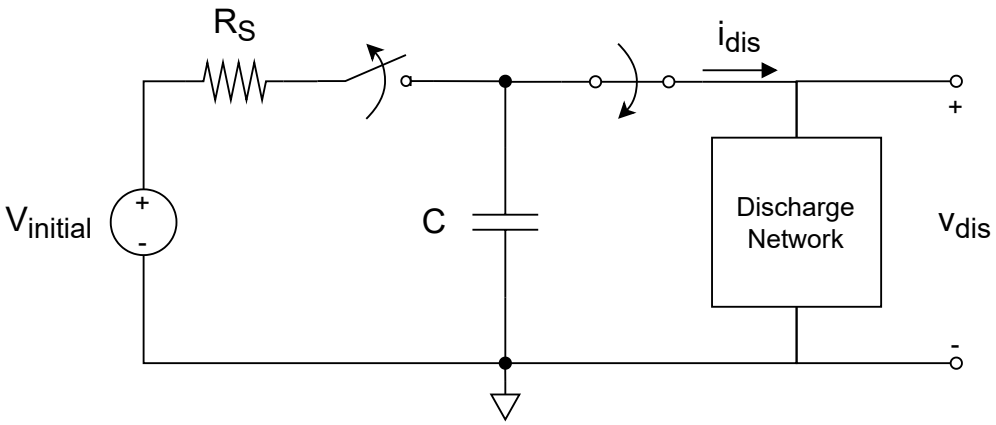
However, for building a voltage reference, it is not enough to choose an arbitrary circuit parameter. There are specific characteristics that the primitive must have in order to be regarded as a potential parameter for building a voltage reference. The following section focuses on those aspects and discusses the alternatives that have been considered for this design of a virtual voltage reference to choose the circuit's physical reference.

3.1.1 Requirements of a physical reference

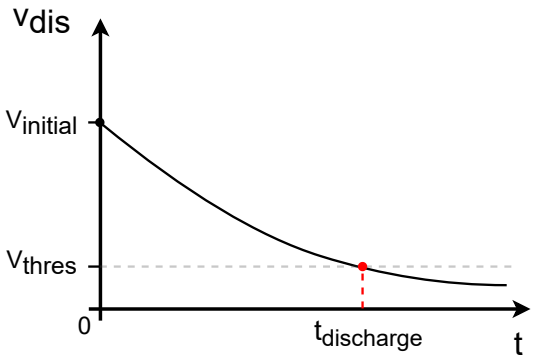
From the analysis of section 2.3 (see Figure 2.5), it is possible to distinguish the physical reference as one of the main parts of the virtual voltage references. Then, for choosing the discharge network, it is essential to review the relevant characteristics of such a circuit.

The first aspect to contemplate is related to area minimization. Considering that the area is one of the constraints of this circuit, the discharge network should be as small as possible. Then, it is expected that only a few devices compose this block.

Besides, the area constraint pushes towards using devices as small as the technology allows them. Thus, it is desired to have MOSFET devices with their minimum dimensions in order to save area. Conveniently, one of the advantages of the virtual



(a) Circuit representation of the idea for obtaining the primitive



(b) Transient behaviour of V_D and $t_{discharge}$ measurement

Figure 3.1. Description of the primitive for the virtual reference.

voltage references with respect to the analog bandgap reference is the dropping of the matching requirement for their devices since the primitives are obtained from a single device. Then, given that the matching is not a priority in the design, it is possible to use the minimum length provided by the technology.

The second relevant aspect to consider for the discharge network is that it provides a relation between the discharge time ($t_{discharge}$) and the initial voltage value ($V_{initial}$) that is not constant in the range used for $V_{initial}$. This means that it is required to avoid a flat plot of $t_{discharge}$ vs $V_{initial}$. It may be possible to have some flatness, but the ideal behaviour of this curve is always to have a derivative different from zero. This aspect may seem trivial, but it is used for discarding structures that provide a very small variation of $t_{discharge}$ vs $V_{initial}$.

The third requirement for the primitives of a virtual voltage reference consists in that the combination of such primitives must allow obtaining both a PTAT and CTAT behaviour. This requirement means that after acquiring a certain number of samples of discharge time (typically two) and combining such samples in different ways, it must be possible to obtain two variables that vary complementary with respect to the temperature.

Before describing the last requirement for the primitive, it is necessary to briefly introduce how the voltage $V_{initial}$ and the discharge time ($t_{discharge}$) are created and measured, respectively.

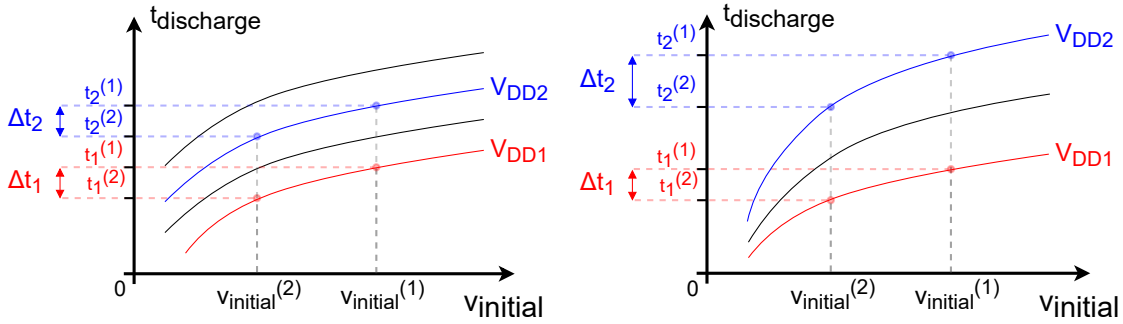
For the generation of $V_{initial}$, it is necessary to have a DAC that creates an analog voltage according to the digital word that the processor sets, as shown in Figure 2.5. More details regarding the specific implementation of the DAC are given in subsection 4.2.1. It is crucial to highlight that, due to the intrinsic behaviour of the DAC, the actual value of $V_{initial}$ will depend on the value of the supply (V_{DD}).

On the other hand, for measuring the discharge time, it is necessary to use a circuit that compares the voltage v_{dis} with respect to a certain threshold (V_{thres}). Then, considering that no other voltage references are available for setting this threshold, it is natural to assume that also V_{thres} depends on the supply V_{DD}.

This shows that both variables, the initial point of the curve and the point where the discharge time is measured, are strongly dependent on the supply.

The impact of the unregulated supply V_{DD} on the measured discharge time allows the circuit to compensate for the variations of the supply, since it is possible to include V_{DD} in the model and observe how it affects the measurements. However, it is not enough to have a variation with V_{DD} since it is required that such variation causes more than a simple vertical shift to the plot of $t_{discharge}$ versus $V_{initial}$. The reason behind that is avoiding the cancellation of the influence of V_{DD} on the discharge time when using a very common primitive: the difference of two discharge times ($\Delta t_{discharge} = t^{(1)}_{discharge} - t^{(2)}_{discharge}$) under two different starting points ($V^{(1)}_{initial}$ and $V^{(2)}_{initial}$).

Figure 3.2 explains graphically why the influence of the voltage in the discharge time must suppose more than a vertical shift of the curves. Figure 3.2a shows the case where the relation between $t_{discharge}$ and V_{DD} is just a vertical shift. This behaviour can be summarized by an equation such as: $t_{discharge} = f(V_{initial}) + k \cdot V_{DD}$. Then, when obtaining the difference between two samples ($\Delta t_{discharge}$), it is clear to conclude that the information regarding the actual value of V_{DD} has been lost. On the other hand, Figure 3.2b depicts the case where the effect of V_{DD} cannot be isolated in the equation, and therefore it represents more than a vertical shift. This relevant aspect allows discarding physical references that do not provide this behaviour in their discharge time.



(a) Linear relation of $t_{discharge}$ with V_{DD} (b) Non-linear relation of $t_{discharge}$ with V_{DD}

Figure 3.2. Discharge time versus initial voltage for different V_{DD}

In summary, there are four requirements for the selected physical reference to be suitable for the virtual voltage reference:

- Minimum possible area.
- Variation with respect to the input parameter. In this case, variation of the discharge time with the initial voltage value.
- Provide primitives that allow obtaining a PTAT and a CTAT behaviour when combined.
- Dependence of the input voltage that is non-linear.

3.1.2 Assessment of architectures for the discharge structure

Several circuits have been proposed as discharge networks for the capacitor of the circuit shown in Figure 3.1. Figure 3.3 summarizes some of the possibilities that have been considered in the search for a suitable physical reference. The capacitor is not shown in these figures, and the two nodes floating on the left correspond to the ports of the discharge network block in Figure 3.1.

Based on the minimization of both the area and the static power consumption, several of these architectures are based on transistors working in subthreshold, i.e., at a very low gate-source voltage (V_{GS}), even $V_{GS} = 0$ is used for several potential architectures. Notice that, in all the mentioned architectures, the length of the

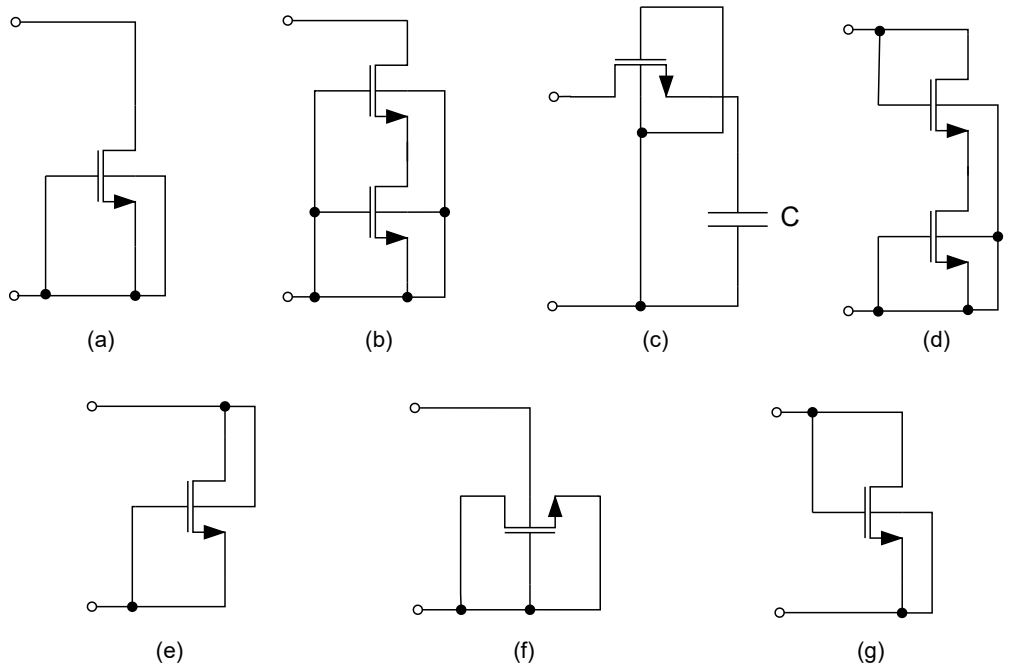


Figure 3.3. Studied architectures for the discharge network

transistors is always the minimum possible, in this case, 180 nm.

Figure 3.3a shows an NMOS transistor that is connected in a configuration that allows to have $V_{GS} = 0$. This condition entails that the transistor operates in the (deep) subthreshold region, thus adding a non-linear behaviour that is expected from the discharge network. Similarly, Figure 3.3b stacks two transistors with their gate voltages connected to ground. This configuration reduces the drain-source voltage V_{DS} available to each device. The purpose of this connection was to assess the impact of this V_{DS} reduction on the discharge time.

Figure 3.3c uses a transistor in subthreshold and an additional capacitor to create charge distribution and discharge the node through the transistor's parasitic elements. Again, this idea was considered searching for some non-linearity in the discharge time. Then, Figure 3.3d stacks a transistor in subthreshold along with a diode-connected transistor. This idea is related again to reducing the V_{DS} and achieving more non-linearity.

The last three architectures are conceptually similar since they all try to obtain a behaviour similar to a diode. Figure 3.3e still uses a transistor with zero V_{GS}, but with the bulk connected to the drain instead of the ground. This architecture

aims to exploit the parasitic diode between the bulk and the source of the NMOS device. Next, Figure 3.3f follows a similar idea but uses the two parasitic diodes between the P-type substrate and the N-type drain/source. Finally, Figure 3.3g uses the usual diode configuration of an NMOS transistor.

Two different analyses have been performed to assess the behaviour of these architectures. Considering that it is relevant for the physical reference to show variation with respect to the supply voltage V_{DD} and temperature, the transient analysis has been performed when varying V_{DD} and the temperature as parameters. These simulation results are obtained using the software Virtuoso IC 6.1.7 with a simulation setup similar to the one shown in Figure 3.1a.

The first simulation results discussed are the influence of the supply voltage V_{DD} on the discharge time. For this reason, a testbench is created that involves setting the initial condition of the voltage of a capacitor and then extracting from the results the time elapsed until the voltage crosses the threshold (V_{thres}). In this case, $V_{initial}$ is swept from $0.3 \cdot V_{DD}$ up to V_{DD}. Similarly, the threshold is set as a fourth of V_{DD}: $V_{thres} = 0.25 \cdot V_{DD}$. Finally, the simulation is performed for a sweep of V_{DD} from 400 mV up to 800 mV since this is the expected range of operation of the voltage reference.

Figure 3.4 shows the simulation results of the first four architectures (from (a) to (d) in Figure 3.3). Notice that all these curves show a clear influence of the initial value on the discharge time. Also, the discharge times go from a few milliseconds up to tens or hundreds of milliseconds. This magnitude is relevant since it will set requirement regarding the speed of the circuit used for measuring the discharge time.

Considering that the different V_{DD} curves seem to be equispaced, this suggests that there is a linear relation between V_{DD} and $t_{discharge}$. For instance, Figure 3.4c shows that for a wide range of $V_{initial}$, the different curves have very similar slopes. And, as highlighted in subsection 3.1.1 this characteristic is undesired from the $t_{discharge}$ vs $V_{initial}$ curves. This characteristic discards the usage of the architecture that consists of a transistor and a capacitor (Figure 3.3c).

The behavior with respect to V_{DD} of the single transistor in subthreshold (Figure 3.4a) shows that the slopes of the curves are very similar from one to another. In the search for a more non-linear behaviour of the discharge time with respect to the supply voltage, the architectures (b) and (d) reduce the V_{DS} by placing either another transistor in subthreshold or a diode-connected transistor. However, as shown in figures 3.4b and 3.4d, the resulting curves are very similar to the single-transistor case. For the case of the stacked subthreshold transistors, the magnitude

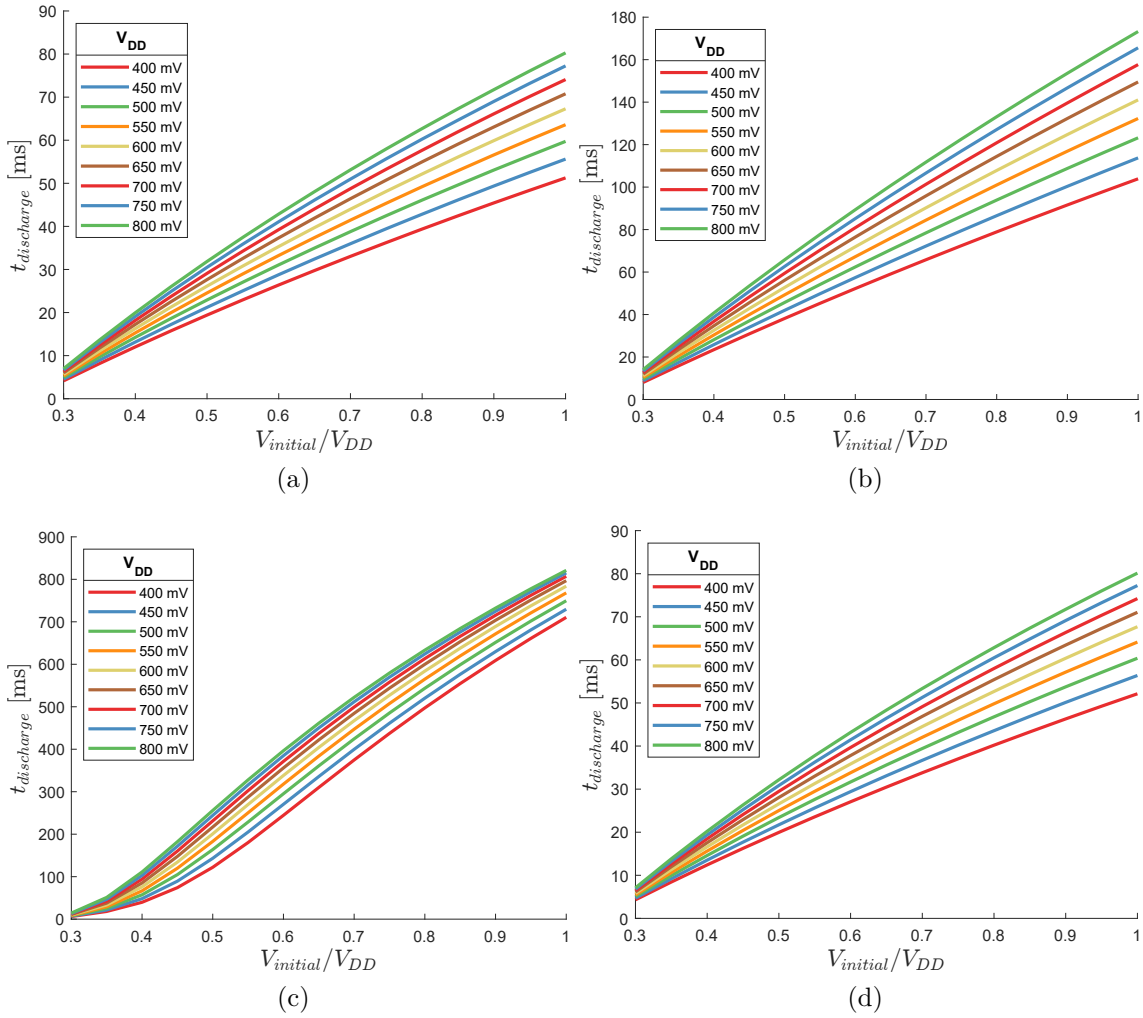


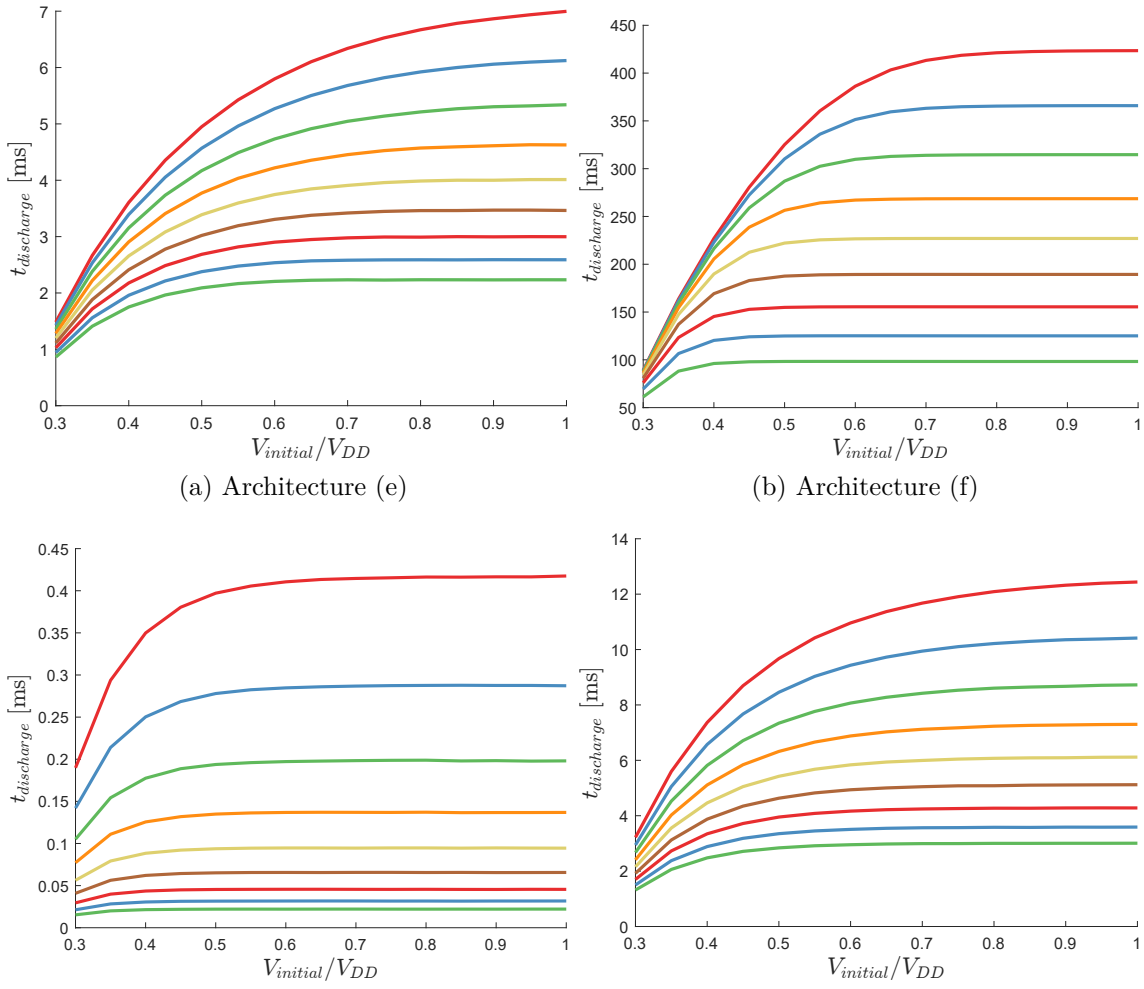
Figure 3.4. Discharge time versus initial voltage for different V_{DD} (architectures (a) to (d))

of the discharge time is around twice that of the single transistor case. In contrast, the case of the diode in series with the subthreshold transistor is almost identical to the single transistor case.

Continuing the evaluation for the rest of the architectures, Figure 3.5 represents the plots for the remaining architectures. The color of the curves for the different V_{DD} correspond to the same colors of the curves in Figure 3.4.

The first characteristic to highlight from these curves corresponds to the reduced variation of $t_{discharge}$ with $V_{initial}$. It is clear that these curves are flatter than

the curves of Figure 3.4. On the other hand, these curves present a considerable advantage compared to the ones in the previous figure: the non-linear relationship with V_{DD}. Notice that the variation with the supply voltage modifies the discharge time value and influences the slope of the $t_{discharge}$ vs. $V_{initial}$ curve. Besides, the effect on the value of $t_{discharge}$ is more evident when the value of V_{DD} is smaller. This characteristic proves again that these curves are less linear with respect to V_{DD} as it is desired by the primitive of the virtual reference.



(a) Architecture (e)

(b) Architecture (f)

(c) Architecture (g)

(d) Two diodes in series

Figure 3.5. Discharge time versus initial voltage for different V_{DD} (architectures (e) to (g))

The similarities in the behaviour of these curves were expected since they all correspond to the model of a capacitor discharging through a diode. The differences

among them consist of the approach used to exploit a diode using a transistor. According to the expectations for the variation of a primitive with respect to the supply voltage, the behaviour shown in Figure 3.5c is the one that satisfies the expected behaviour since it offers a wide and non-linear variation both in absolute value and also in the slope of the curves.

However, the architecture of the diode-connected transistor has two potential issues. First, the variation of $t_{discharge}$ with $V_{initial}$ is not granted for the whole $V_{initial}$ range. In the region $V_{initial}/V_{DD} > 0.6$ it is evident that the discharge time is almost constant with respect to $V_{initial}$. Second, the absolute values of the discharge times are very small (the smallest of all the architectures). Then, considering that this will constrain the circuit that measures the discharge time, it is better to have larger discharge times. Also, considering that it is expected that the discharge time will drop for larger temperatures, it is important to have relatively large discharge times in nominal conditions (27 °C).

For these reasons, the approach of connecting several devices in series that were also used for other architectures in the past is also used here. Fortunately, the results in this case give a behaviour corresponding to the circuit's requirements. As shown in Figure 3.5d, adding two diode-connected transistors in series brings satisfactory results since it solves both problems mentioned above. The discharge times increased by more than an order of magnitude, and the curves are less flat than the curves of Figure 3.5c.

The results shown in Figure 3.5d correspond to the connection of two diode-connected PMOS transistors since it was observed that these devices provided a better variation of $t_{discharge}$ with $V_{initial}$ than the case that used two NMOS transistors in series. Notice that the behaviour of architecture (e) (Figure 3.5a) shows similar results to the one of the two diodes in series, however, this architecture showed poor performance when connecting several devices in series for increasing the discharge times.

Then, the circuit that fits the most as a physical reference according to the analysis performed on the supply voltage variation corresponds to a stack of diode-connected PMOS transistors. Figure 3.6 depicts the circuit schematic of this discharge network.

The second simulation analysis to be performed corresponds to the temperature behaviour of the circuit architectures. This analysis expects to characterize whether it is possible to obtain a PTAT and a CTAT variable from the measurements of the discharge time of the circuit. In order to do so, this test consists of performing the measurements of the discharge time for different temperatures and initial points.

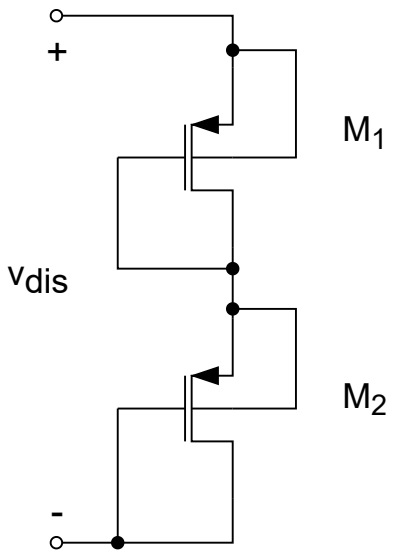


Figure 3.6. Discharge network composed of two PMOS transistors.

Considering that the discharge time usually provides a CTAT behaviour, then the other variable to be considered in this case corresponds to the ratio between two discharge times ($t^{(2)}_{discharge}/t^{(1)}_{discharge}$) at two different initial points ($m^{(1)}$ and $m^{(2)}$).

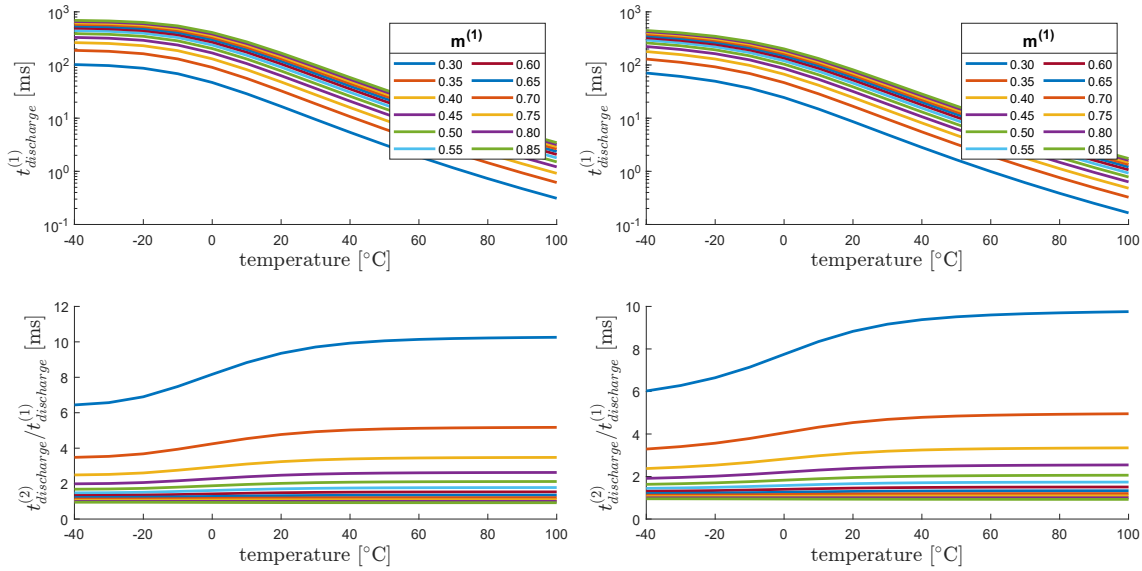
The reason for choosing the ratio is related to the fact that this is a typical primitive, also considering that the ratio between two samples of the discharge time allows for creating some independence of the clock used for measuring the discharge time. Besides, the difference between two samples ($t^{(2)}_{discharge} - t^{(1)}_{discharge}$) showed a CTAT behavior in most of the cases, this motivated to consider other function in order to assess the possibility of obtaining a PTAT component by combining the discharge time measurements.

For this experiment, the initial point ($m^{(2)}$) was set to its maximum possible value, then $V^{(2)}_{initial} \approx V_{DD}$. On the other hand, the initial point $m^{(1)}$ was swept from near-threshold ($m^{(1)}_{min}/2^N = 0.3$) up to a value close to the maximum: $m^{(1)}_{max}/2^N = 0.8$. The temperature sweep covers the range from $-40\,°C$ up to $100\,°C$. This simulation was performed for all the architectures shown in Figure 3.3. However, not all the results will be discussed here since some of these architectures were already discarded because they are unsuitable for obtaining the V_{DD} independence.

Figure 3.7 shows the results of the temperature characterization of architectures (a) (Figure 3.7a) and (d) (Figure 3.7b). The single discharge time measurements (top of the figure) are shown in a logarithmic scale for the y-axis. These results

show that the discharge time has a CTAT behaviour, whereas the ratio between the two discharge times shows a PTAT behaviour. However, notice that both the CTAT and the PTAT behaviour have a non-constant slope through the range; this corresponds to a disadvantage with respect to another architecture that might show a constant slope through the range.

The results regarding architecture (b) are not shown, but they are very similar to the results shown in Figure 3.7.

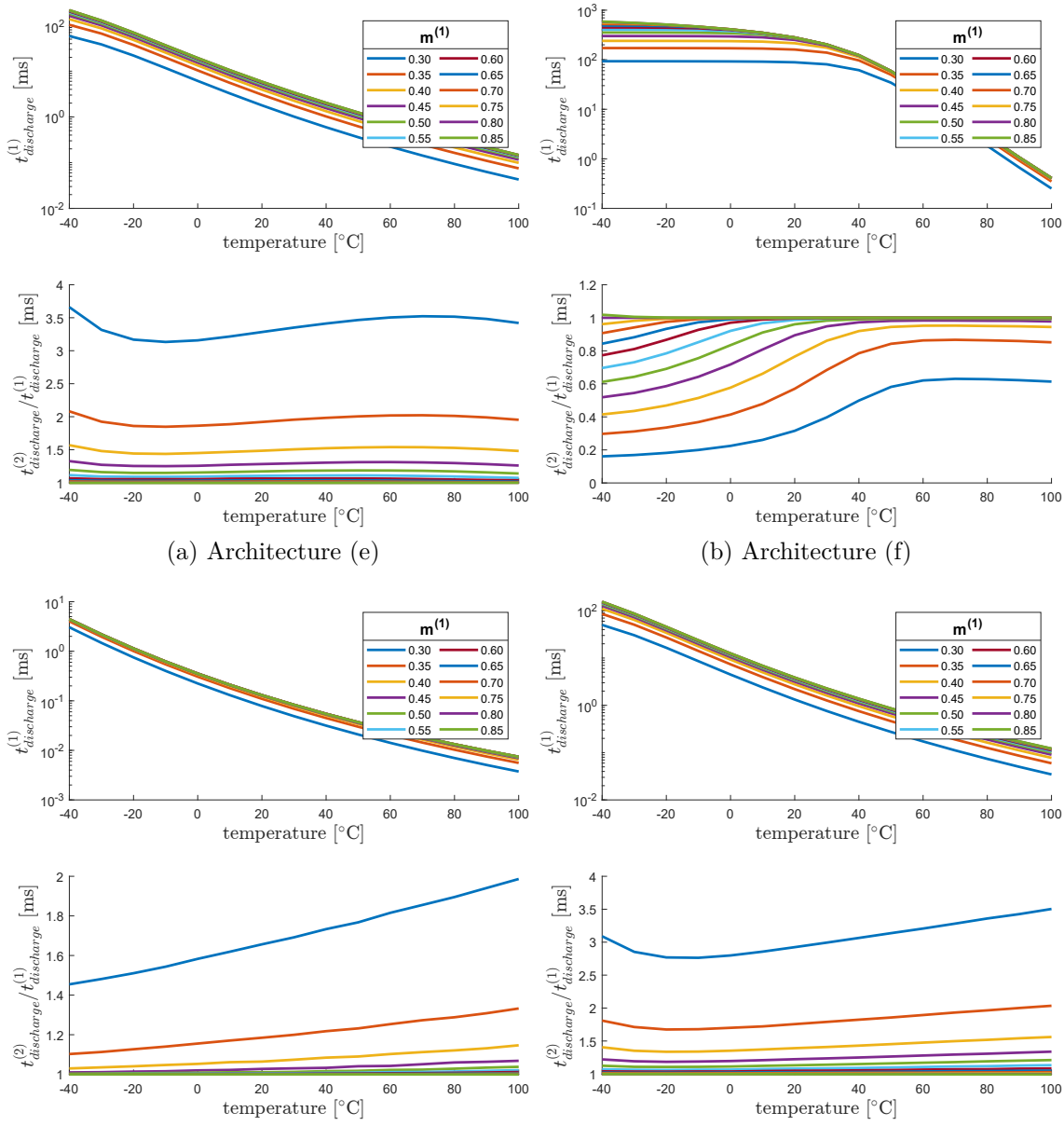


(a) Architecture (a) (b) Architecture (d)

Figure 3.7. Discharge time versus temperature for different initial voltages. ($m^{(2)} = 1$)

Figure 3.8 shows the results of the same temperature characterization of the four remaining architectures. As in the previous cases, the discharge time describes a CTAT behaviour. However, in the case of the architecture that exploits the two parasitic diodes (architecture (f) in Figure 3.8b), the discharge time is almost constant when the temperature is below 40 °C. This architecture is discarded since it cannot provide a PTAT component. Similarly, architecture (e) shows a PTAT behaviour in a short range of temperature, from 0 °C to 60 °C.

Notice that both for the voltage supply and for the temperature analysis, the best behaviour corresponds to the diode-connected transistors. In this case, Figure 3.8c and Figure 3.8d show that the discharge time describes a CTAT behaviour and the ratio between two discharge times has a PTAT behaviour. In contrast with the



(a) Architecture (e)

(b) Architecture (f)

(c) Architecture (g)

(d) Two diodes in series

Figure 3.8. Discharge time versus temperature for different initial voltages. ($m^{(2)} = 1$)

conclusion obtained for the V_{DD} analysis, these results show an advantage of the single diode architecture with respect to the architecture that contains two diodes in series since the latter shows a PTAT behaviour only for temperatures above $-20\,°C$. For this application, the temperature range from $-20\,°C$ to $100\,°C$ could be enough for achieving the temperature compensation. Besides, the values of the

discharge times for the case of a single diode are very small, and this supposes a further constraint on the clock that will be used for measuring the discharge time. Then, in line with the expectations of low power consumption of the circuit, it is preferable to choose the implementation that requires a lower clock frequency.

Notice also that the slope of the CTAT curve is not constant with respect to temperature. This characteristic is common to all the architectures; all of them are plotted in the semi-logarithmic plot, thus, showing that the slope is not constant.

In conclusion, the architecture that shows the most suitable characteristics when considering the overall temperature and power supply behaviour corresponds to the two diode-connected transistors in series (as in Figure 3.6). Then, this circuit is used as a discharge network for the implementation of the virtual voltage reference.

3.2 Virtual voltage reference with capacitor and diode stack

After choosing the appropriate circuit for the discharge network, it is possible to represent the proposed version of the virtual voltage reference by employing a block diagram. Figure 3.9 depicts the blocks and circuits that compose the voltage reference. Notice the similarities between this circuit and the block diagram presented in Figure 2.5.

In this circuit, the digital processor does not only generate the digital words for the DAC (`m`) and receives the digital words from the ADC (`n`) but also creates a control signal (`ctrl`) that is used for the synchronization. This control signal is connected to the input of a transistor that works as a switch for the discharge network, the same switch represented in Figure 3.1. Then, once the DAC charges the node v_{dis} according to the received digital signal, the control signal enables the discharge network while the DAC output is set to a high-impedance state. This moment is also used for setting the start of the discharge time measurement.

Once the discharge of the node v_{dis} starts, an additional block is used for converting the information of the discharge time into a digital signal. Such a block is known as a time-to-digital converter (TDC). This particular TDC uses the time that passes from the enable of the control signal until the moment when v_{dis} crosses the threshold and then converts it into a digital word that is sent to the digital processor.

This work focuses on the design of the discharge network and the algorithm that compensates the voltage and temperature variations. The details regarding the

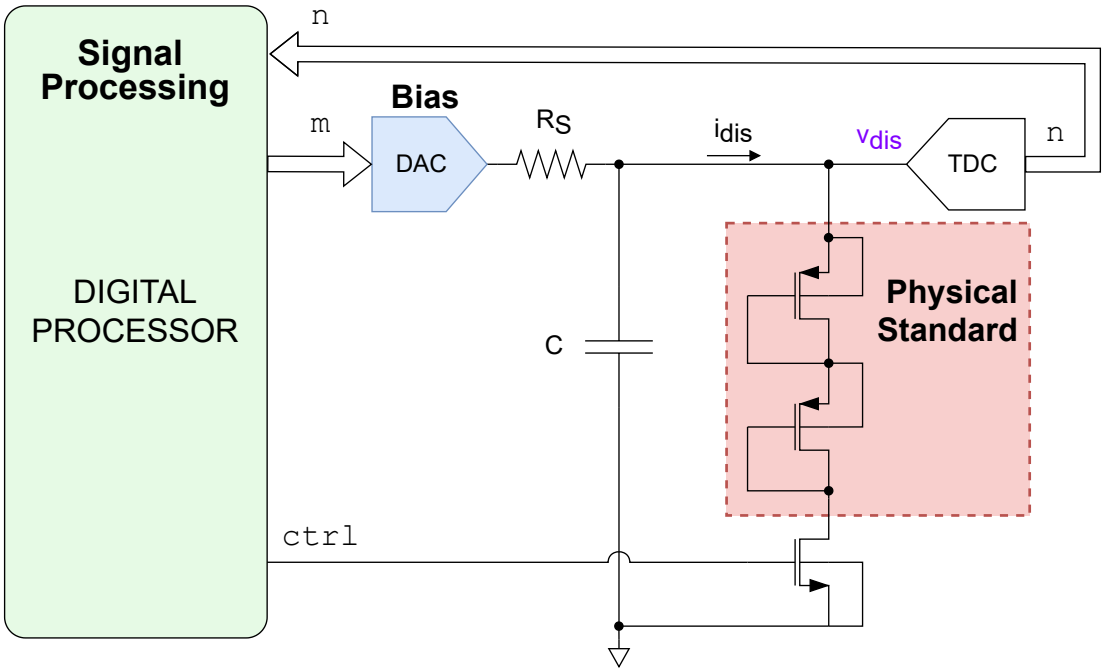


Figure 3.9. Diagram of the virtual voltage reference with discharge network

circuits to be used as DAC and TDC are given in the next chapter, particularly in sections 4.2.1 and 4.2.2, respectively. Therefore, the following sections focus on the theoretical modelling of the physical reference and the development of the algorithm to obtain the supply and temperature compensation of the output reference voltage.

3.3 Theoretical analysis of the diode-based discharge structure

Before implementing an algorithm related to the physical reference selected for the virtual voltage reference, it is necessary to gain some knowledge regarding the circuit's behavior since this could be useful for creating such model. Figure 3.10 shows the circuit that has to be analyzed in this case. In order to simplify the model, the DAC is assumed to have an infinite output impedance during the discharge of the capacitor. Then it does not provide an additional discharge path. Similarly, the series resistance of the transistor that works as a switch is neglected. Besides, to maintain the analysis's generality, the discharge network is supposed to be composed of a number P of PMOS transistors connected in series.

The following analysis will be performed assuming that all transistors in series

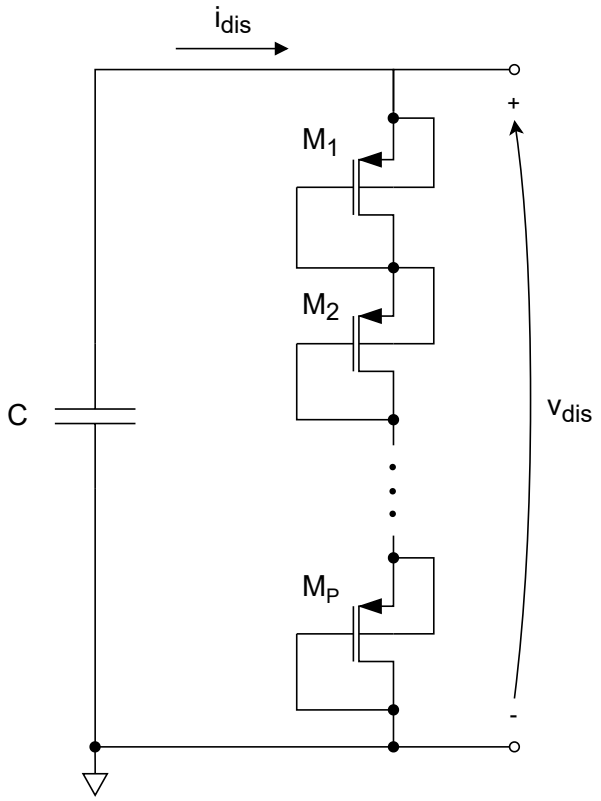


Figure 3.10. Discharge network of the virtual voltage reference

are operating in the subthreshold region. This is a reasonable assumption considering that the expected supply voltages are very small (from 400 mV to 800 mV). Also, even in the case of the 800 mV supply voltage, since several devices are connected in series, this voltage should be distributed among the several devices. This voltage distribution leads to a V_{GS} that is smaller than the threshold voltage of the transistors (V_{th}), ensuring that the devices work in the subthreshold region.

Before moving on, it is necessary to recall the equation that describes the subthreshold current of a transistor (for long-channel devices) [36]:

$$I_D = \mu C_d \frac{W}{L} V_T^2 \cdot \exp\left(\frac{V_{GS} - V_{th}}{\eta \cdot V_T}\right) \cdot \left(1 - \exp\left(-\frac{V_{DS}}{V_T}\right)\right) \tag{3.1}$$

Where $C_d = \sqrt{\epsilon_{Si} q N_{sub}/(4\Phi_B)}$ corresponds to the capacitance of the depletion region that is formed under the gate and $\eta = 1 + C_d/C_{ox}$, being C_{ox} the oxide capacitance. Henceforth, the quantity I_0 will be used in replacement of $\mu C_d \frac{W}{L} V_T^2$.

Then, using the previous equation for the specific discharge network allows for defining the current I_{dis}. Given that all the devices are theoretically identical, and the same current flows through them, the available voltage is expected to be distributed through the stack of transistors. The following equation can summarize this behaviour:

$$V_{GS} = V_{DS} = \frac{V_{dis}}{P} \tag{3.2}$$

Where P is the number of diode-connected transistors in series.

Once the transistor's current is defined, it is possible to obtain a differential equation considering that this corresponds to the discharge current of the capacitor:

$$I_{dis} = I_0 \cdot \exp\left(\frac{\frac{V_{dis}}{P} - V_{th}}{\eta \cdot V_T}\right) \cdot \left(1 - \exp\left(-\frac{V_{dis}}{P \cdot V_T}\right)\right) = -C \cdot \frac{dV_{dis}}{dt} \tag{3.3}$$

The previous equation can be rewritten as:

$$I_0 \cdot \exp\left(\frac{-V_{th}}{\eta \cdot V_T}\right) \cdot \left[\exp\left(\frac{V_{dis}}{P \cdot \eta \cdot V_T}\right) - \exp\left(\frac{V_{dis}(1-\eta)}{P \cdot \eta \cdot V_T}\right)\right] = -C \cdot \frac{dV_{dis}}{dt} \tag{3.4}$$

Separating the variables of the differential equation will allow searching for a solution:

$$-\frac{I_0 \cdot \exp\left(\frac{-V_{th}}{\eta \cdot V_T}\right)}{C} dt = \frac{dV_{dis}}{\exp\left(\frac{V_{dis}}{P \cdot \eta \cdot V_T}\right) - \exp\left(\frac{V_{dis}(1-\eta)}{P \cdot \eta \cdot V_T}\right)} \tag{3.5}$$

Then, the two integrals have to be solved:

$$-\int \frac{I_0 \cdot \exp\frac{-V_{th}}{\eta \cdot V_T}}{C}\, dt = \int \frac{1}{\exp\left(\frac{V_{dis}}{P \cdot \eta \cdot V_T}\right) - \exp\left(\frac{V_{dis}(1-\eta)}{P \cdot \eta \cdot V_T}\right)}\, dV_{dis} \tag{3.6}$$

In order to solve the integral of the right-hand side (RHS), it is possible to rely on the Taylor expansion of the expression $\exp(x)$:

$$\exp(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \dots \tag{3.7}$$

Using this expansion truncated to the fourth term allows representing the RHS as:

$$\frac{1}{\exp\left(\frac{V_{dis}}{P \cdot \eta \cdot V_T}\right) - \exp\left(\frac{V_{dis}(1-\eta)}{P \cdot \eta \cdot V_T}\right)} \approx \frac{1}{\frac{V_{dis}}{P \cdot V_T} + \left(\frac{V_{dis}}{\eta \cdot P \cdot V_T}\right)^2 \cdot \frac{1-(1-\eta)^2}{2!} + \left(\frac{V_{dis}}{\eta \cdot P \cdot V_T}\right)^3 \cdot \frac{1-(1-\eta)^3}{3!}} \tag{3.8}$$

Then, using the approximation of Equation (3.8) and replacing into the RHS of Equation (3.6) corresponds to the new integral to be solved. The following equation represents the result of such integral.

$$\int \frac{dV_{dis}}{\exp\left(\frac{V_{dis}}{P\cdot\eta\cdot V_T}\right)-\exp\left(\frac{V_{dis}(1-\eta)}{P\cdot\eta\cdot V_T}\right)} = \frac{PV_T}{2}\cdot\ln\left(\frac{V_{dis}^2}{\frac{V_{dis}^2}{(\eta\cdot P\cdot V_T)^3}\cdot\frac{1-(1-\eta)^3}{3!}+\frac{V_{dis}}{(\eta\cdot P\cdot V_T)^2}\cdot\frac{1-(1-\eta)^2}{2!}+\frac{1}{P\cdot V_T}}\right)+ -\frac{P\cdot V_T\cdot(1-(1-\eta)^2)}{8(\eta\cdot P\cdot V_T)^2}\int\frac{dV_{dis}}{\frac{V_{dis}^2}{(\eta\cdot P\cdot V_T)^3}\cdot\frac{1-(1-\eta)^3}{3!}+\frac{V_{dis}}{(\eta\cdot P\cdot V_T)^2}\cdot\frac{1-(1-\eta)^2}{2!}+\frac{1}{P\cdot V_T}} \tag{3.9}$$

Also the integral on the RHS of Equation (3.9) corresponds to:

$$\int\frac{dV_{dis}}{\frac{V_{dis}^2}{(\eta\cdot P\cdot V_T)^3}\cdot\frac{1-(1-\eta)^3}{3!}+\frac{V_{dis}}{(\eta\cdot P\cdot V_T)^2}\cdot\frac{1-(1-\eta)^2}{2!}+\frac{1}{P\cdot V_T}} = \frac{1}{\sqrt{\left[\frac{1-(1-\eta)^2}{(\eta PV_T)^2 2!}\right]^2-4\left[\frac{1-(1-\eta)^3}{(\eta PV_T)^3 3!}\right]\cdot\frac{1}{PV_T}}}\cdot \cdot\ln\left(\frac{\frac{2[1-(1-\eta)^3]}{(\eta PV_T)^3 3!}\cdot V_{dis}+\frac{1-(1-\eta)^2}{(\eta PV_T)^2\cdot 2!}+\sqrt{\frac{(2-\eta)^2}{4}-\frac{2(\eta^2-3\eta+3)}{3}}}{\frac{2[1-(1-\eta)^3]}{(\eta PV_T)^3 3!}\cdot V_{dis}+\frac{1-(1-\eta)^2}{(\eta PV_T)^2\cdot 2!}-\sqrt{\frac{(2-\eta)^2}{4}-\frac{2(\eta^2-3\eta+3)}{3}}}\right) \tag{3.10}$$

In parallel to this integral, the LHS of Equation (3.6) can be solved as:

$$-\int\frac{I_0\cdot\exp\left(\frac{-V_{th}}{\eta\cdot V_T}\right)}{C}\,dt = -\frac{I_0\cdot\exp\left(\frac{-V_{th}}{\eta V_t}\right)}{C}\cdot t+K \tag{3.11}$$

Where K corresponds to the constant that is given by the initial condition. For this circuit, the initial corresponds to $V_{dis}=V_{initial}$ when $t=0$.

Then, the value of K can be obtained by combining the results of equations 3.9 3.10 and 3.11:

$$K=\frac{PV_T}{2}\cdot\ln\left(\frac{R\cdot V_T\cdot V_{initial}^2}{\alpha V_{initial}^2+\beta V_{initial}+1}\right)-\frac{P\cdot V_T(2-\eta)}{8\cdot\lambda}\cdot\ln\left(\frac{\alpha_1 V_{initial}+\beta_1-\lambda}{\alpha_1 V_{initial}+\beta_1+\lambda}\right) \tag{3.12}$$

Where the constants correspond to:

$$\begin{aligned}
\alpha =& \frac{\eta^2 - 3\eta + 3}{6(\eta P V_T)^2}; \\
\beta =& \frac{2-\eta}{2\eta P V_T}; \\
\alpha_1 =& \frac{\eta^2 - 3\eta + 3}{3\eta P V_T}; \\
\beta_1 =& \frac{2-\eta}{2}; \\
\lambda =& \sqrt{\frac{(1-\eta)^2}{4} - \frac{2(\eta^2 - 3\eta + 3)}{3}};
\end{aligned} \tag{3.13}$$

Finally, gathering all the results allows defining the discharge time of the circuit as a function of the other parameters. Assuming that at a time $t = t_{discharge}$ the voltage at node V_{dis} is equal to V_{thres} (see Figure 3.1), then the discharge time corresponds to:

$$\begin{aligned}
t_{discharge} = \frac{C \cdot R \cdot V_T}{2I_0} \cdot \exp\left(\frac{V_{th}}{\eta V_T}\right) \cdot \Bigg[& \ln\left(\frac{V_{initial}^2 \cdot (\alpha V_{thres}^2 + \beta V_{thres} + 1)}{V_{thres}^2 \cdot (\alpha V_{initial}^2 + \beta V_{initial} + 1)}\right) + \dots \\
& - \frac{2-\eta}{8\lambda} \ln\left(\frac{\alpha_1 V_{initial} + \beta_1 - \lambda}{\alpha_1 V_{initial} + \beta_1 + \lambda} \cdot \frac{\alpha_1 V_{thres} + \beta_1 + \lambda}{\alpha_1 V_{thres} + \beta_1 - \lambda}\right) \Bigg]
\end{aligned} \tag{3.14}$$

For different values of η, the absolute value of the second term inside the square brackets is much smaller than the first term. Thus, this second term will be neglected in the following steps of the analysis. Leading to the final equation of the discharge time as a function of the initial voltage and the threshold:

$$t_{discharge} \approx \frac{CP \cdot \exp\left(\frac{V_{th}}{\eta V_T}\right)}{2 \cdot \mu C_d \cdot \frac{W}{L} \cdot V_T} \cdot \ln\left(\frac{V_{initial}^2 \cdot (\alpha V_{thres}^2 + \beta V_{thres} + 1)}{V_{thres}^2 \cdot (\alpha V_{initial}^2 + \beta V_{initial} + 1)}\right) \tag{3.15}$$

Now, it is relevant to highlight that both voltage variables ($V_{initial}$ and V_{thres}) in Equation (3.15) depend on the supply voltage V_{DD}. Then, they can be written as follows:

$$\begin{aligned}
V_{initial} =& \frac{\mathtt{m}}{2^{\mathtt{N}}} \cdot V_{DD} \\
V_{thres} =& K_{thres} \cdot V_{DD}
\end{aligned} \tag{3.16}$$

Where `m` corresponds to the digital word set from the output of the processor and `N` is the number of bits of the DAC used. Also, K_{thres} corresponds to the percentage of V_{DD} used to set the threshold when the discharge time is measured.

Replacing the Equation (3.16) into Equation (3.15) leads to the following equation. From now onwards, the quatity t_d represents also the discharge time ($t_d = t_{discharge}$). Also τ_d is used to represent the quantity $(CP \cdot \exp \frac{V_{th}}{\eta V_T})/(2 \cdot \mu C_d \cdot \frac{W}{L} \cdot V_T)$

$$t_d = \tau_d \cdot \ln \left(\frac{(\frac{m}{2^N} \cdot V_{DD})^2 \cdot (\alpha (K_{thres} \cdot V_{DD})^2 + \beta K_{thres} \cdot V_{DD} + 1)}{(K_{thres} \cdot V_{DD})^2 \cdot \left(\alpha (\frac{m}{2^N} \cdot V_{DD})^2 + \beta \frac{m}{2^N} \cdot V_{DD} + 1 \right)} \right) \tag{3.17}$$

That could be simplified to:

$$t_d = \tau_d \cdot \ln \left(\frac{(\frac{m}{2^N})^2 \cdot (\alpha (K_{thres} \cdot V_{DD})^2 + \beta K_{thres} \cdot V_{DD} + 1)}{K_{thres}^2 \cdot \left(\alpha (\frac{m}{2^N} \cdot V_{DD})^2 + \beta \frac{m}{2^N} \cdot V_{DD} + 1 \right)} \right) \tag{3.18}$$

Hence, this result shows that, as expected, the relation between V_{DD} and the discharge time is not linear and cannot be easily simplified when performing the difference between two $t_{discharge}$ samples. There is an additional advantage when performing the difference between two discharge times measured with two initial points. The following equation corresponds to the difference between two samples of the discharge time ($t_{discharge}^{(1)}$ and $t_{discharge}^{(2)}$) when the initial points are set from $\mathtt{m}^{(1)}$ and $\mathtt{m}^{(2)}$.

$$\begin{aligned} t_d^{(1)} - t_d^{(2)} = \tau_d \cdot \ln \Bigg(& \frac{(\frac{\mathtt{m}^{(1)}}{2^N})^2 \cdot (\alpha (K_{thres} \cdot V_{DD})^2 + \beta K_{thres} \cdot V_{DD} + 1)}{K_{thres}^2 \cdot \left(\alpha (\frac{\mathtt{m}^{(1)}}{2^N} \cdot V_{DD})^2 + \beta \frac{\mathtt{m}^{(1)}}{2^N} \cdot V_{DD} + 1 \right)} \cdot \\ & \cdot \frac{K_{thres}^2 \cdot \left(\alpha (\frac{\mathtt{m}^{(2)}}{2^N} \cdot V_{DD})^2 + \beta \frac{\mathtt{m}^{(2)}}{2^N} \cdot V_{DD} + 1 \right)}{(\frac{\mathtt{m}^{(2)}}{2^N})^2 \cdot (\alpha (K_{thres} \cdot V_{DD})^2 + \beta K_{thres} \cdot V_{DD} + 1)} \Bigg) \end{aligned} \tag{3.19}$$

Furthermore, this equation could be simplified to:

$$\Delta t_d = t_d^{(1)} - t_d^{(2)} = \tau_d \cdot \ln \left(\frac{(\mathtt{m}^{(1)})^2 \cdot \left(\alpha (\frac{\mathtt{m}^{(2)}}{2^N} \cdot V_{DD})^2 + \beta \frac{\mathtt{m}^{(2)}}{2^N} \cdot V_{DD} + 1 \right)}{(\mathtt{m}^{(2)})^2 \cdot \left(\alpha (\frac{\mathtt{m}^{(1)}}{2^N} \cdot V_{DD})^2 + \beta \frac{\mathtt{m}^{(1)}}{2^N} \cdot V_{DD} + 1 \right)} \right) \tag{3.20}$$

Notice that Equation (3.20) does not contain any dependence on the threshold constant (K_{thres}) that has been set for this circuit. This result demonstrates that it is reasonable to include the difference between two samples in the primitives used by the virtual voltage reference algorithm. This is the most relevant outcome of the theoretical analysis that has been performed.

3.4 Implementation of the virtual reference algorithm

The current virtual voltage reference implementation was divided into two steps. The first step focused on creating a voltage reference that could counteract the variations with the supply voltage. At the same time, the second part of the design of the virtual reference algorithm concentrates on the compensation for temperature variations.

3.4.1 Supply voltage compensation of the circuit

Several approaches were used to solve the problem of the output voltage variation with respect to the voltage supply. The objective of this compensation consists of finding a variable that counteracts the variation of V_{DD}. Such variable must be a function of variables that are available to the digital processor, i.e. $\mathtt{m}^{(i)}, \mathtt{n}^{(i)}$. Then, the outcome of this first step in the design of the processor of the virtual reference corresponds to an equation that represents the variable r. This variable r must be complementary to the variation of V_{DD}, allowing to have a voltage independent of V_{DD} at the output of the DAC. This process is analogous to the process discussed in [35], whose outcome is shown in Equation (2.22).

This first approach used for obtaining the V_{DD} compensation for the voltage reference consisted of using the equations obtained from the theoretical analysis of the circuit (section 3.3). If it is possible to obtain an expression for V_{DD} as a function of the variables $\mathtt{m}$ and $\mathtt{n}$, then the inverse of such function allows finding a variable that compensates for the variations of V_{DD}.

By observing both Equation (3.18) and Equation (3.20), it is possible to observe that these equations could be solved in order to find an expression for V_{DD}. Notice that it is preferable to use the equation that uses Δt_d instead of the equation that uses a single measurement because the Δt_d primitive is less dependent on the threshold (K_{thres}). Besides, using two measurements is better than relying on a single-point measurement.

Focusing on the Δt_d, Equation (3.20) can be represented as:

$$\exp \frac{\Delta t_d}{\tau_d} = \frac{(\mathtt{m}^{(1)})^2 \cdot \left(\alpha(\frac{\mathtt{m}^{(2)}}{2^N} \cdot V_{DD})^2 + \beta \frac{\mathtt{m}^{(2)}}{2^N} \cdot V_{DD} + 1\right)}{(\mathtt{m}^{(2)})^2 \cdot \left(\alpha(\frac{\mathtt{m}^{(1)}}{2^N} \cdot V_{DD})^2 + \beta \frac{\mathtt{m}^{(1)}}{2^N} \cdot V_{DD} + 1\right)} \tag{3.21}$$

Then, using MATLAB, the equation can be solved for V_{DD}, leading to:

$$V_{DD} = \frac{3(\eta-2)\eta P V_T}{2(\eta^2-3\eta+3)} \cdot \left[\frac{\frac{\mathtt{m}^{(1)}}{2^N} - \frac{\mathtt{m}^{(2)}}{2^N} \cdot \exp\frac{\Delta n}{\tau_d}}{\frac{\mathtt{m}^{(1)}}{2^N} \cdot \frac{\mathtt{m}^{(2)}}{2^N} \cdot (\exp\frac{\Delta n}{\tau_d} - 1)} \right] \tag{3.22}$$

And this can be expressed as:

$$V_{DD} = \frac{3(\eta-2)\eta P V_T}{2(\eta^2-3\eta+3)} \cdot f_1(\mathtt{m}^{(1)}, \mathtt{m}^{(2)}, \Delta n) \tag{3.23}$$

The function f_1 is expected to follow V_{DD} if the theoretical model represents the circuit's behavior well. In order to test that this is correct, it is necessary to simulate the circuit and then juxtapose the results of the simulations and the theoretical model.

A capacitor of 500 fF is used for the circuit simulation. This capacitor value corresponds to the characteristics of the DAC used in this architecture. More details regarding the DAC are given in subsection 4.2.1. Then, a stack of six diode-connected transistors for such a small capacitor is used for the discharge network. The high number of diodes, in this case, is because with more transistors in series, a more significant variation of the discharge times is observed with respect to $V_{initial}$ (as discussed in subsection 3.1.2).

The length of the diode-connected transistor is the minimum due to the area minimization constraint. For the technology node used for this version of the voltage reference, the minimum possible length is 180 nm. For this simulation, the width of the transistors was set to 1 µm; this value will be reduced in the final design due to the area constraints. The threshold voltage of the transistor can be obtained from the simulator, and for this technology, it corresponds to 573 mV. Finally, the parameters η and μC_d are left as a degree of freedom that allows the fitting of the theoretical and the simulation results.

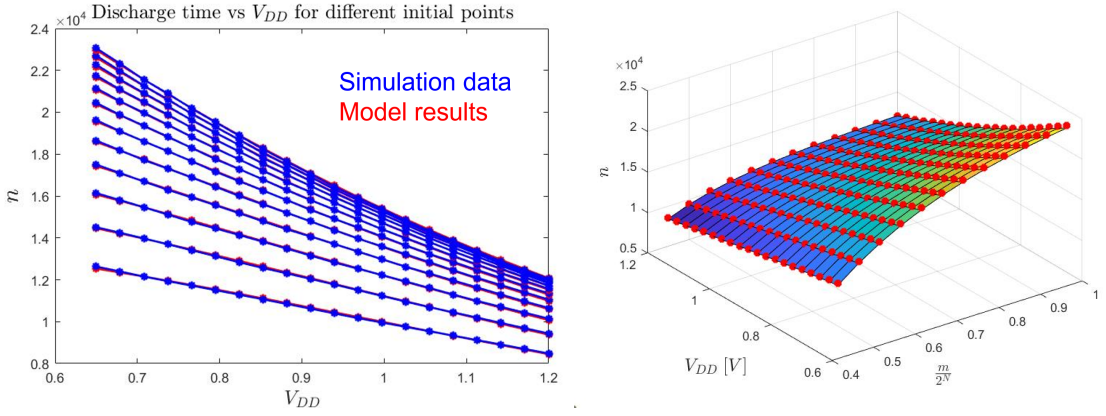
A transient simulation is then performed on the circuit and the discharge time is measured and converted into a digital word. The LSB of a such digital word corresponds to 1 µs. This resolution means that the measured values for `n` are of the order of 1×10^4, which corresponds to tens of milliseconds.

Then, using the simulation results and relying on the two degrees of freedom of the theoretical equation (η and μC_d), the model results are fitted into the simulation results. The following table summarizes the values of the parameters obtained that provide the best fitting between the model and the simulation results. These parameter values were found through an algorithm that would minimize the error between the extracted simulation results and the values of the model.

Parameter	Value
C	500 pF
P	6 diode-connected transistors
(W/L)	(1 µm / 180 nm)
V_{th}	573 mV
K_{thres}	0.25
V_T	26 mV
η	3.86885
μC_d	8.986×10^{-8} F V s^{-1}

Table 3.1. Summary of the resulting parameters of the model fitting

The juxtaposition of the simulations results and the model of Equation (3.18) using the parameters of Table 3.1 is represented graphically in Figure 3.11. For this simulation V_{DD} has been swept from 0.65 V up to 1.2 V whereas the initial point $\mathtt{m}/2^N$ (or $V_{initial}/V_{DD}$) is swept from 0.3 up to 1.0. The resulting Normalized Root Mean Square (NRMS) error of this fitting is 0.87 %. Notice that the figure also shows a reasonably good fitting between the two procedures.



(a) (b)

Figure 3.11. Comparison between simulation and theoretical model results.

Now, the objective of modeling the circuit consists of obtaining an equation representing the behaviour of V_{DD}. This behaviour is expected to be obtained from f_1 of Equation (3.23). This equation can be transformed to:

$$\frac{V_{DD}}{f_1(\mathtt{m}^{(1)}, \mathtt{m}^{(2)}, \Delta n)} = V_0 = \frac{3(\eta - 2)\eta P V_T}{2(\eta^2 - 3\eta + 3)} \tag{3.24}$$

Since all the dependency of V_{DD} is contained inside the function f_1. It is expected that when V_{DD} varies, f_1 follows it, leading to a constant ratio between these two quantities. The ratio between V_{DD} and f_1 will be named V_0, and it represents a voltage that is independent of the V_{DD} variations.

The value of V_0 can be calculated using the parameters listed in Table 3.1. Then, in this case:

$$V_0 = \frac{3(\eta - 2)\eta P V_T}{2(\eta^2 - 3\eta + 3)} = 0.1418\,\mathrm{V} \tag{3.25}$$

Figure 3.12 presents the results of the plot of the parameter V_0 vs V_{DD}. First, notice that the values are very close to the value of V_0 obtained in Equation (3.25). Besides, it is also relevant to highlight that the best behaviour is obtained from the yellow and orange curves. This means that it is favourable to have a large difference between the two samples of the initial value. This result coincides with the behaviour observed in Figure 3.5, where it is clear that using a wide separation between the two initial points allows having a larger difference between two measurements of $\Delta t_{discharge}$ for different V_{DD}. Such a larger difference makes it easier for the model to distinguish what is the actual V_{DD} value.

Figure 3.12 allows defining also the line regulation of the parameter V_0. This measurement is relevant since it tells how accurate the assumption is that V_0 is constant with V_{DD}. Thus, using Equation (2.1) with V_{DD} that ranges from 0.65 V to 1.2 V and using the nominal value as the one given by Equation (3.25) it is possible to calculate the line regulation. The best case for the line regulation corresponds to the case where $m^{(1)} = 0.85$ and $m^{(2)} = 0.45$:

$$LR_{theoretical\ \ model} = 2.26\ \%/\mathrm{V} \tag{3.26}$$

This result is very poor considering that a voltage reference's usual line regulation values are usually one order of magnitude lower.

Then, it is clear that the theoretical model does not offer a fitting that is adequate for the requirements of a virtual voltage reference. This outcome implies using different alternatives for modelling the circuit's behaviour. One of those alternatives consists in relying on MATLAB for elaborating the model between the variables and then allowing to obtain a function of $\mathtt{m}^{(i)}, \mathtt{n}^{(i)}$ that follows V_{DD}.

An important consideration regarding the combination of the values $\mathtt{n}^{(i)}$ to be used is that it is not only important to achieve some independence on the threshold

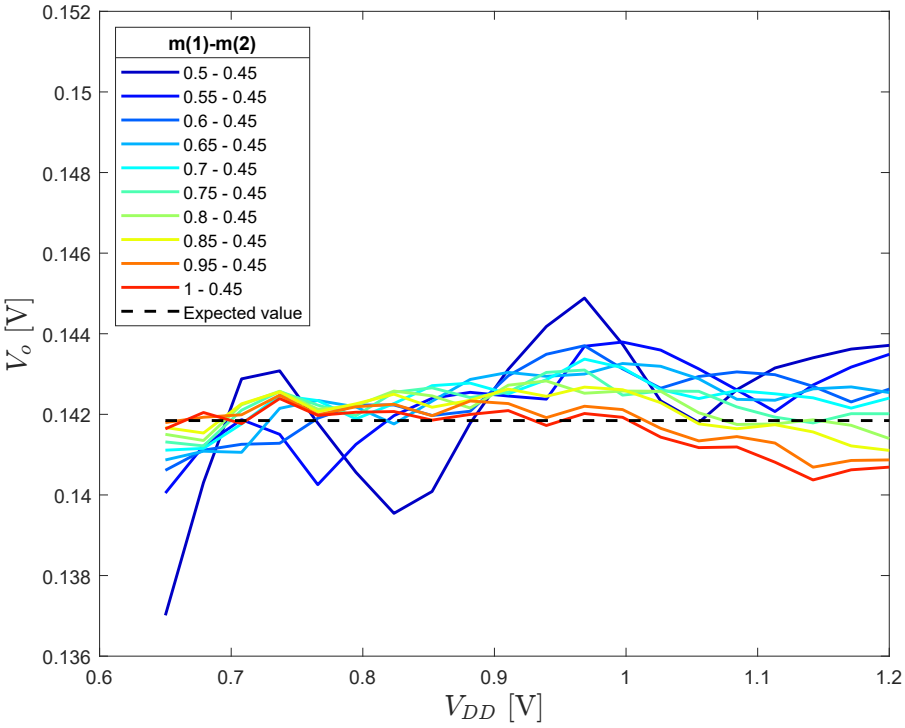


Figure 3.12. V_0 vs V_{DD} for different $\mathtt{m}^{(1)}, \mathtt{m}^{(2)}$

(K_{thres}), but it is also relevant to have some independence regarding the clock frequency to be used in the TDC. Notice that the values of $\mathtt{n}^{(i)}$ are found by dividing the discharge time by the clock period:

$$\mathtt{n}^{(i)} = \frac{{t_{discharge}}^{(i)}}{T_{CLK}} = {t_{discharge}}^{(i)} \cdot f_{CLK} \tag{3.27}$$

Then, in the potential case of having clock uncertainties due to jitter or similar effects, this could drastically affect the behaviour of the virtual reference. Then, it would be very convenient to perform the ratio between two measurements of the discharge time since this could lead to the independence of the clock frequency (f_{CLK}).

Considering that it is relevant to keep the independence with respect to both K_{thres} and f_{CLK}, a possible combination of the $\mathtt{n}^{(i)}$ values that allows keeping such independence could be:

$$ratio\_deltas\_n = \frac{\mathtt{n}^{(1)} - \mathtt{n}^{(2)}}{\mathtt{n}^{(1)} - \mathtt{n}^{(3)}} \tag{3.28}$$

Notice that an additional measurement would be required. This means that instead of two points, it is necessary to have three events of the discharge time for

using such an equation. Then, in this case, the modelling task consists of finding the function r that follows V_{DD}, and that is a function of four variables:

$$r = f_2(\mathtt{m}^{(1)}, \mathtt{m}^{(2)}, \mathtt{m}^{(3)}, ratio\_deltas\_n) = \frac{1}{V_{DD}} \tag{3.29}$$

The fact that three variables are required constrains the possibilities regarding the achievable models since MATLAB polynomial fitting allows a maximum of two independent variables. Then, instead of using the polynomial fitting for the function of Equation (3.29), the Regression Learner from the Statistics and Machine Learning Toolbox from MATLAB is used for achieving the fitting.

Using the Regression Learner allowed for testing different modelling approaches. Table 3.2 summarizes the results obtained with the different machine learning regressions that are available in the MATLAB toolbox. Notice that the best fitting is provided by the Neural Network (NN). In this case, the NN is composed of two layers, each of a size of ten elements, and the activation function corresponds to Rectified Linear Unit (RLU).

Model		R^2
Linear Regression		0.09
Regression Tree	Fine	0.64
	Medium	0.58
	Coarse	0.51
Quadratic SVM		0.64
Neural Network		0.98

Table 3.2. Machine learning regression models results

However, even with the best case of the fitting, the result provided for the line regulation is very poor:

$$LR_{Neural\ \ Network} = 0.8\ \%/\mathrm{V} \tag{3.30}$$

The previous results show that relying on a more robust strategy is necessary to compensate for the voltage variations. Besides, the large number of variables limits the possibilities regarding the alternatives that can be used for modelling the circuit.

A possible solution for limiting the exploration space and the degree of complexity of the model is to use the virtual voltage reference algorithm concept to model the circuit. Recall from subsection 2.3.2 that the initial values, when the algorithm is used, are proportional to the virtual reference value r. Then, using the principle

given by the algorithm allows to set the initial values even in the modelling stage of the circuit:

$$\begin{aligned} \mathtt{m}^{(1)} &= \lfloor k_1 \cdot \mathtt{r} \rceil \\ \mathtt{m}^{(2)} &= \lfloor k_2 \cdot \mathtt{r} \rceil \\ \mathtt{m}^{(3)} &= \lfloor k_3 \cdot \mathtt{r} \rceil \end{aligned} \tag{3.31}$$

Then, instead of modelling the circuit in order to obtain $\mathtt{r}$ as a function of the initial values and discharge time measurements, i.e. $\mathtt{r} = f_2(\mathtt{m}^{(1)}, \mathtt{m}^{(2)}, \mathtt{m}^{(3)}, \mathtt{n}^{(1)}, \mathtt{n}^{(2)}, \mathtt{n}^{(3)})$, it is possible to abstract all the information on only two variables. As a result of the iterative nature of the virtual voltage reference algorithm, $\mathtt{r}$ can be calculated as:

$$\mathtt{r} = f(\hat{\mathtt{r}}, y) \tag{3.32}$$

Where $\hat{\mathtt{r}}$ corresponds to the value of $\mathtt{r}$ in the previous algorithm iteration, and y corresponds to a function of the measured discharge times: $\mathtt{n}^{(1)}, \mathtt{n}^{(2)}, \mathtt{n}^{(3)}$. This abstraction enables the reduction of the space of independent variables and allows to use MATLAB to perform the modelling of the function f based on polynomials of high order (up to degree 5).

Recall that the supply voltage V_{DD} is expected to vary on a range from 400 mV up to 800 mV and the outcome of the circuit model is to provide a reference voltage $V_0 = \mathtt{r} \cdot V_{DD} = 1\,\mathrm{V}$, theoretically, . Then, noticing that $\mathtt{r}$ is expected to vary as $1/V_{DD}$, the expected range of $\mathtt{r}$, as a real number, goes from 1.25 up to 2.5. This information is useful for building the simulation for the characterization of the circuit and also for setting the constraint of the values k_1, k_2 and k_3 to be used in Equation (3.31).

Considering that k_i is a number between 0 and 1, that also represents a portion of the voltage V_0, there are two constraints that the values k_i must accomplish:

1. From one side, the value k_i must be within the reasonable voltage values that the DAC can provide. Hence, the product $k_i \cdot V_0$ must lead to a reasonable voltage value. Then, the minimum V_{DD} sets a constraint for the maximum value of k_i: $V_{DDmin} > k_i \cdot V_0$.

2. On the other hand, the minimum value of k_i is limited by the threshold that is used for measuring the discharge time of the circuit. Considering that V_{thres} is expected to follow the value of V_{DD} with a constant rate (K_{thres}) of 0.25. Then, it is necessary to ensure that the DAC output is larger than V_{thres} even when the threshold has its maximum value: $k_i \cdot V_0 > V_{thres\_max}$

Combining both conditions for the values of k_i results in the following:

$$V_{DDmin} > k_i \cdot V_0 > V_{thresmax} \quad \text{for } i = 1,2,3 \tag{3.33}$$

Given that $V_0 = 1\,\text{V}$, $V_{DDmin} = 400\,\text{mV}$ and $V_{thresmax} = V_{DDmax} \cdot K_{thres} = 200\,\text{mV}$. It is possible to obtain the ranges that the three values of k must meet:

$$0.4 > k_i > 0.2 \quad \text{for } i = 1,2,3 \tag{3.34}$$

Previous simulation results have allowed showing that it is better to use values of $\mathtt{m}^{(i)}$ that are well separated from one another. Additionally, since the subtraction is usually performed between $n^{(1)}$ and $n^{(2)}$; and $n^{(1)}$ and $n^{(3)}$, it is reasonable to chose the value of k_1 well separated from the other two (k_2, k_3). This reasoning leads to the following values chosen for k_i:

$$\begin{aligned} k_1 &= 0.4 \\ k_2 &= 0.3 \\ k_3 &= 0.25 \end{aligned} \tag{3.35}$$

Notice that some margin is necessary from the minimum possible value since measuring the discharge time when $V_{initial}$ is equal to V_{thres} is not possible.

These variables are then inserted into the Verilog-A model of the circuit that contains the algorithm. More details of the Verilog-A implementation are given in section 4.1. Then, parametric simulations are performed in the aforementioned ranges of V_{DD} and also $\hat{\mathtt{r}}$ (that corresponds to the expected range of $\mathtt{r}$). The simulation results are then ported to MATLAB in order to perform the appropriate modeling of the system.

Following the previously discussed ideas regarding which could be a good function for fitting the circuit's behavior, the ratio of exponential functions is still chosen as the function y. This function allows a lower error when fitting the simulation results into an equation. Then, the y function used in this case is defined by:

$$y = \exp\left(\frac{\mathtt{n}^{(1)} - \mathtt{n}^{(2)}}{\mathtt{n}^{(3)} - 1} - 2\right) \tag{3.36}$$

This function has been chosen since it best fits the r function. Notice that this function is not entirely independent of the threshold nor the clock frequency since it is not the ratio between the two samples' differences. In any case, it still uses the ratio and the difference between samples; thus, it is expected to gain some independence with respect to those two parameters.

Since the number of variables is now reduced, it is possible to use the function `fit` of MATLAB that could be used for fitting surfaces. A total of 20.000 points

result from the parametric sweep of the circuit. Then, using MATLAB for fitting these points into a surface, for this specific case, the type of fitting corresponds to a polynomial of degrees three and five. Figure 3.13 depicts the 20.000 simulation points obtained from the analog simulator and the surface that fits those points.

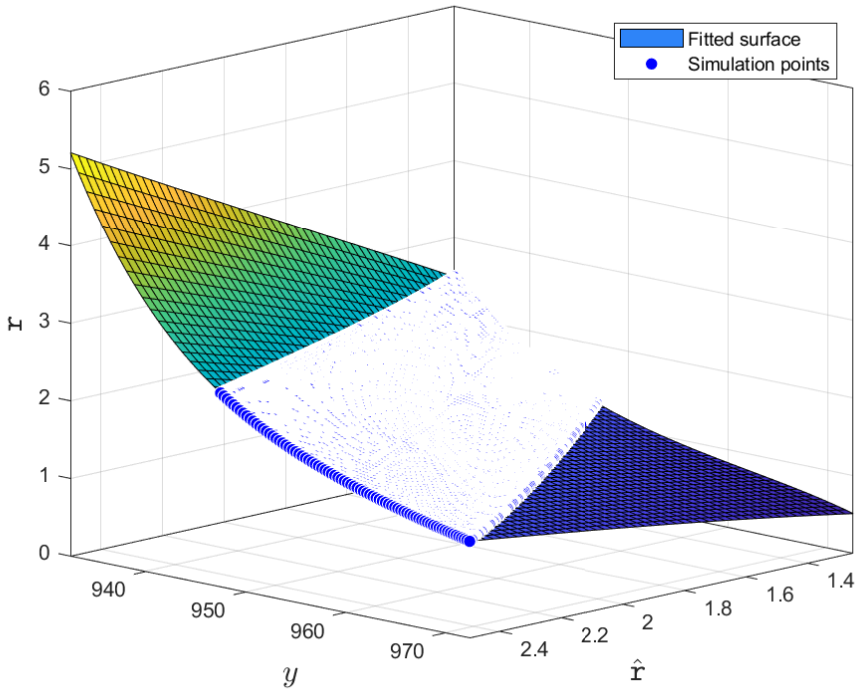


Figure 3.13. Fitted surface of the model that uses the virtual reference algorithm

For this surface, the polynomial uses 18 coefficients, and this polynomial is given by the formula:

$$\begin{aligned} r = {} & p00 + p10 \cdot \hat{\mathtt{r}} + p01 \cdot y + p20 \cdot \hat{\mathtt{r}}^2 + p11 \cdot \hat{\mathtt{r}} \cdot y + p02 \cdot y^2 + p30 \cdot \hat{\mathtt{r}}^3 + p21 \cdot \hat{\mathtt{r}}^2 \cdot y + \\ & + p12 \cdot \hat{\mathtt{r}} \cdot y^2 + p03 \cdot y^3 + p31 \cdot \hat{\mathtt{r}}^3 \cdot y + p22 \cdot \hat{\mathtt{r}}^2 \cdot y^2 + p13 \cdot \hat{\mathtt{r}} \cdot y^3 + p04 \cdot y^4 + \\ & + p32 \cdot \hat{\mathtt{r}}^3 \cdot y^2 + p23 \cdot \hat{\mathtt{r}}^2 \cdot y^3 + p14 \cdot \hat{\mathtt{r}} \cdot y^4 + p05 \cdot y^5 \end{aligned} \tag{3.37}$$

The fitting has been performed for normalized values of the variables $\hat{\mathtt{r}}$ and y. For $\hat{\mathtt{r}}$, the mean value corresponds to 1.875 and the standard deviation to 0.3623, whereas for y, the mean is 950.5 and the standard deviation is 8.31. Besides, the parameter values are summarised in table Table 3.3.

Parameter	Value
p00	1.699
p10	0.299
p01	-0.420
p20	-3.413×10^{-3}
p11	-59.568×10^{-3}
p02	69.305×10^{-3}
p30	0.217×10^{-3}
p21	1.786×10^{-3}
p12	4.525×10^{-3}
p03	-20.910×10^{-3}
p31	-0.129×10^{-3}
p22	-0.386×10^{-3}
p13	-0.660×10^{-3}
p04	5.599×10^{-3}
p32	0.161×10^{-3}
p23	39.861×10^{-6}
p14	-0.332×10^{-3}
p05	-1.274×10^{-3}

Table 3.3. Parameters of the polynomial used for fitting the surface

Then, an algorithm very similar to the algorithm presented in Figure 2.6 is used in the analog simulator for obtaining the compensation with respect to the supply voltage variation. In this case, the function used for setting the value of r at each iteration corresponds to Equation (3.37), where y is obtained as in Equation (3.36). Also, $\hat{\mathtt{r}}$ corresponds to the value of r in the previous iteration.

3.4.2 Temperature compensation of the circuit

After achieving the power supply independence of the voltage reference, it is also relevant to study the possible ways of obtaining independence with respect to temperature. Recall from Figure 2.2 that the usual approach for achieving temperature independence is by combining two variables that have different behaviours with temperature: a PTAT variable and a CTAT variable.

As a starting point, it is relevant to consider what is the temperature behaviour of the model of the circuit that is proposed from the supply independence, i.e., the result of the previous subsection. This analysis implies performing a circuit simulation in some relevant temperature range. In this case the temperature is swept

from 20 °C up to 100 °C. Lower temperatures were not considered since the model does not adapt well to such temperatures and responds with a value of r outside the expected range.

Figure 3.14 shows the results of assessing the temperature behaviour of the virtual voltage reference. Notice that the voltage reference describes a CTAT behaviour.

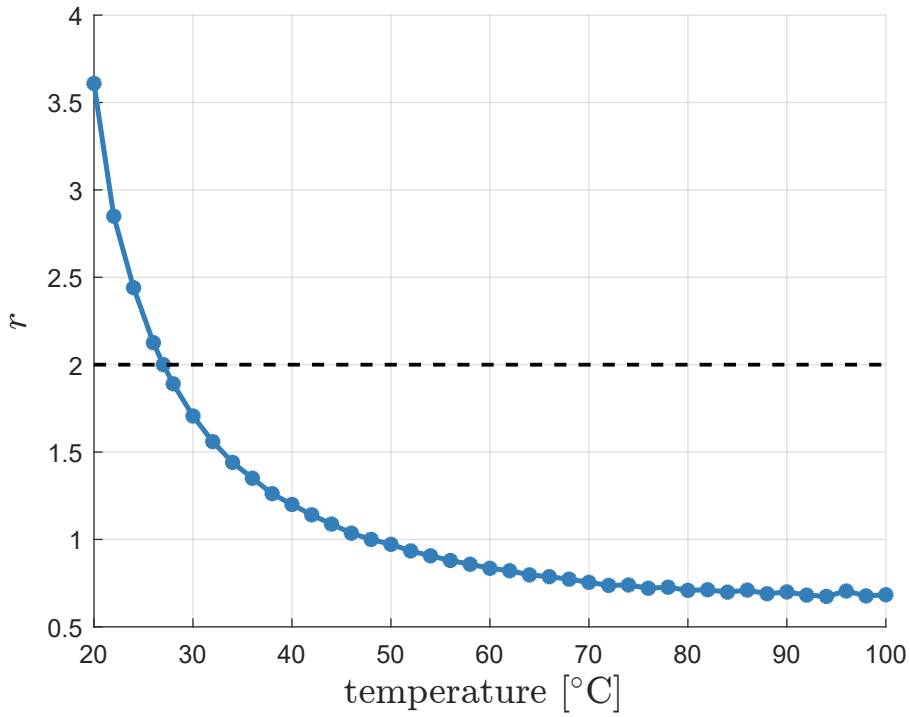


Figure 3.14. Temperature behaviour of supply independent voltage reference

In order to achieve the temperature compensation, finding another function r that has a PTAT behaviour is necessary. The purpose here is to combine them and provide the weighting of such functions in order to obtain a temperature-independent voltage reference.

In this case, the approach consisted of redefining the function y, which would mean that a new fitting process has to be performed for the r as a function of $\hat{\mathtt{r}}$ and the new y. In order to avoid confusion with the function y of Equation (3.36), the new function will be called y_{PTAT}. The experience with the circuit and also the temperature simulations discussed in subsection 3.1.2 (see Figure 3.8), suggest that this behaviour can be achieved by using the ratio of the discharge times. This lead to the following equation.

$$y_{PTAT} = \exp\left(\frac{\mathtt{n}^{(1)} - \mathtt{n}^{(2)}}{\mathtt{n}^{(1)} - \mathtt{n}^{(3)}}\right) \tag{3.38}$$

In contrast with Equation (3.36), notice that Equation (3.38) does include both the difference between samples and also the ratio among them. Thus, this means that y_{PTAT} is expected to be less dependent from both K_{thres} and f_{CLK}.

Parameter	**Value**
p00	1.643
p10	0.317
p01	0.412
p20	0.3373×10^{-3}
p11	90.312×10^{-3}
p02	0.124
p30	-0.269×10^{-3}
p21	9.894×10^{-3}
p12	0.425×10^{-3}
p03	0.819×10^{-3}
p31	-5.039×10^{-3}
p22	-0.145×10^{-3}
p13	-6.402×10^{-3}
p04	0.368×10^{-3}
p32	4.329×10^{-3}
p23	-1.641×10^{-3}
p14	-7.819×10^{-3}
p05	4.329×10^{-3}

Table 3.4. Parameters of the new polynomial used for fitting the surface

Then, the fitting of r as a function of $\hat{\mathtt{r}}$ and y_{PTAT} is performed using the MATLAB fitting function. In this case, also a polynomial of degree three-five is obtained. Hence, r is still described by Equation (3.37), but the values of the coefficients are different. Table 3.4 describes the fitting parameters that correspond to Equation (3.37).

Figure 3.15 shows the temperature behaviour of the virtual voltage reference algorithm when the function y is replaced by y_{PTAT}. Notice that the value of r increases accordingly with the temperature, as expected. Unfortunately, the fitting of r with $\hat{\mathtt{r}}$ and y_{PTAT} is not as good as the fitting of r with $\hat{\mathtt{r}}$ and y. This means that the performance with respect to the variation of V_{DD}, i.e., the line regulation,

will be more poor using this function.

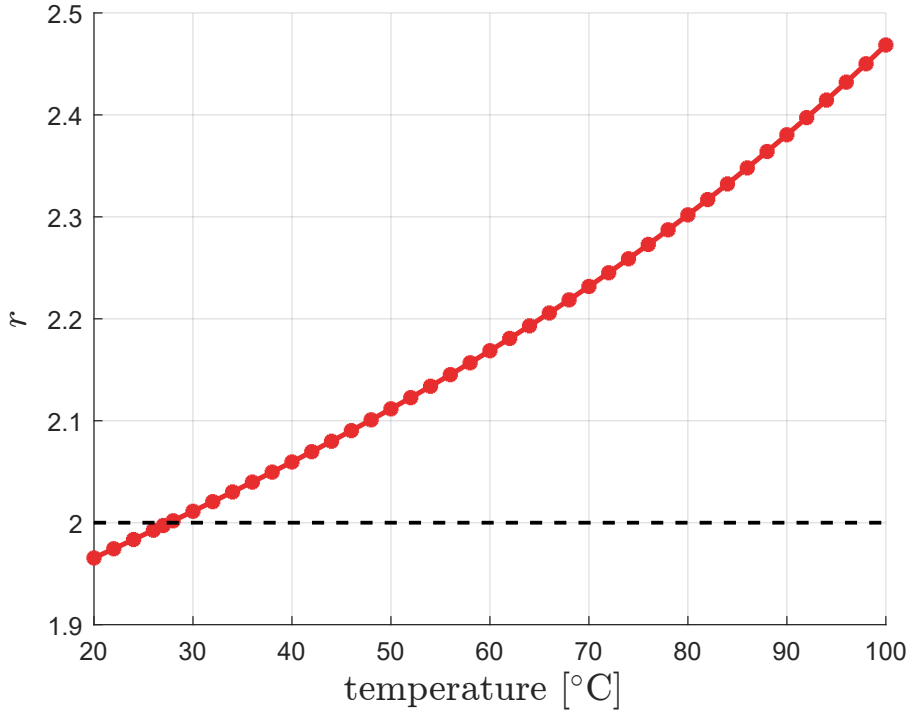


Figure 3.15. Temperature behaviour of voltage reference when using y_{PTAT}

The final step for achieving temperature independence combines both equations of r into a single value of the virtual reference that compensates with respect to supply and temperature variations. During this step, it is crucial to correctly choose the coefficients used for weighting each of the functions. The weighting process can be represented using the following equation.

$$r = \Gamma \cdot r_{PTAT} + (1 - \Gamma) \cdot r_{CTAT} \tag{3.39}$$

Where the coefficient Γ is used to resolve the mismatch between the PTAT and the CTAT curves. Also, r_{CTAT} corresponds to the solution of Figure 3.14 whereas r_{PTAT} is shown in Figure 3.15.

Finally, Figure 3.16 shows the flowchart used to implement the virtual voltage reference. It is relevant to highlight that this version now uses three different initial values instead of two.

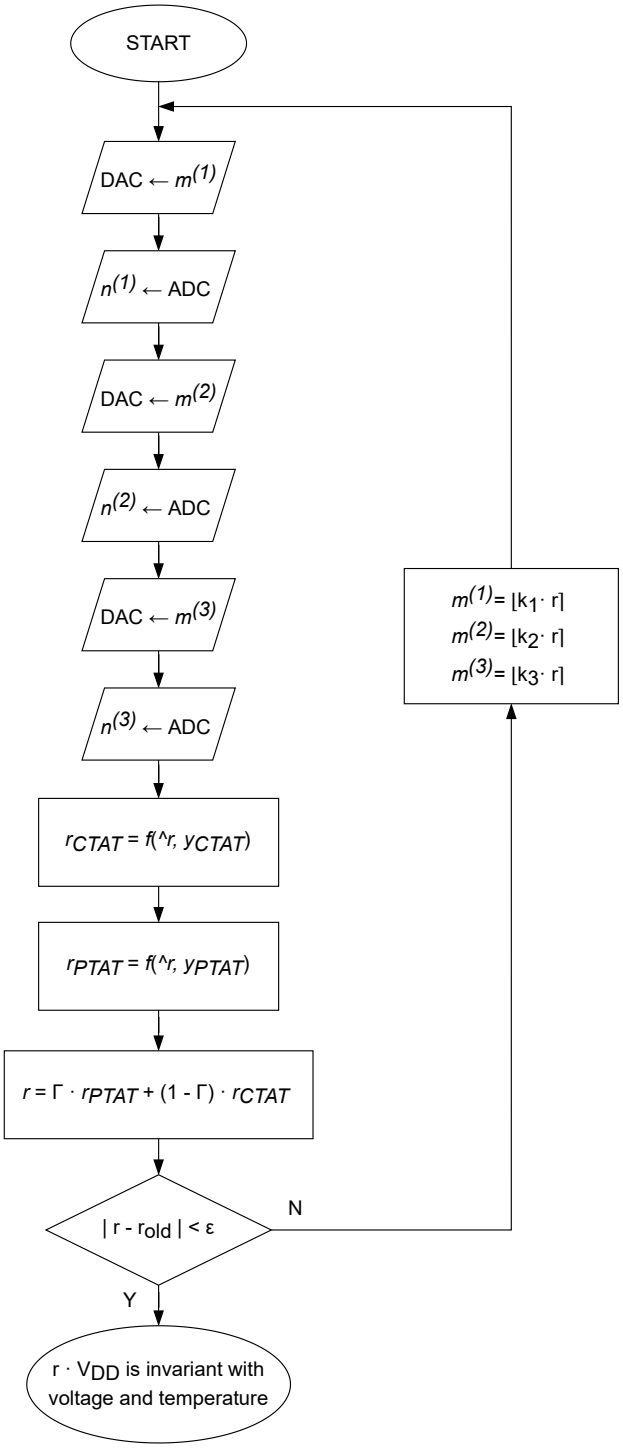


Figure 3.16. Algorithm of the proposed virtual voltage reference

The following chapter gives a detailed description of the algorithm's implementation. Furthermore, the results obtained for the current virtual voltage reference are discussed in chapter 5.

Chapter 4

Implementation details of the virtual reference algorithm

Figure 4.1 presents the top-level circuit of the virtual voltage reference. This circuit corresponds to the schematic representation in Virtuoso of the circuit depicted in Figure 3.9. The `ref_algorithm` block corresponds to the algorithm's Verilog-A implementation and the circuit's control sequence. Notice that this block is also responsible for sampling the node V_{dis}. Besides, this block provides the excitation for charging the capacitor. Then, both the DAC and the TDC are embedded within the `ref_algorithm` block.

The capacitor and the discharge network are also shown in Figure 4.1, they are both inside the `discharge_network` block. These correspond to the structures discussed in previous chapters that allow maximizing the circuit's performance with respect to voltage and temperature variations. More details regarding the transistor implementation of the discharge network, along with the resulting layout, are given in section 4.4. Finally, the voltage source V_1, on the left of Figure 4.1, represents the non-PVT-independent voltage supply of the circuit.

This chapter focuses on the details of the Verilog-A constructs used for modelling the digital processor, the DAC, and the TDC inside the `ref_algorithm` block. Besides, some possible implementations of suitable architectures for the DAC and the TDC are discussed. Finally, the layout of the discharge network is presented when using a 180-nm CMOS technology node.

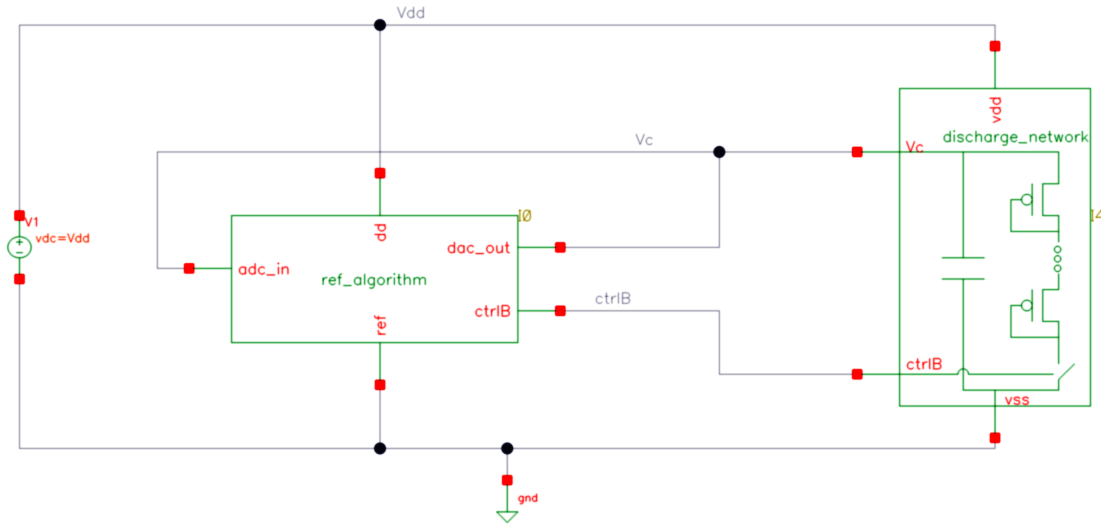


Figure 4.1. Top level circuit of the virtual voltage reference

4.1 Modelling of the architecture for simulation

Appendix A contains the Verilog-A implementation of the module `ref_algorithm`. This module receives the power supply and ground through the pins `dd` and `ref`, respectively. The input `adc_in` is in charge of sampling the capacitor node. In contrast, the node `dac_out` creates the voltages according to the digital word set internally and also proportional to the supply voltage. Finally, the control signal corresponds to the port `ctrlB`.

The following listing shows the definition of the module and the parameters that can be set when instantiating the module. The constants k_1, k_2 and k_3 are set from lines 7 to 9 using the values presented in Equation (3.35), and also the constant K_{thres} is set to 0.25 as discussed in the previous chapter. Another relevant parameter is the relative error that sets the condition when the algorithm finishes its execution. Also, the number of bits of the ADC and DAC are defined as parameters. These numbers are used for truncating the results when converted from digital into analog signals.

A critical parameter is `k_final`, which is the value set at the DAC output when the algorithm converges. Then, once the virtual voltage reference algorithm has finished processing, the expected outcome is to have a voltage of 350 mV at the DAC output.

```
module ref_algorithm(dd, ref, adc_in, dac_out, close_sw);

//...

parameter real r_hat = 1.5; //Initial r (first iteration)
parameter real thres = 0.25; //Discharge time threshold (
    K_thres)
parameter real k1 = 0.4;
parameter real k2 = 0.3;
parameter real k3 = 0.25;
parameter real k_final = 0.35; // Final constant k that sets
    the regulated voltage
parameter real relativeError_r = 0.01; // Relative error for
    assessing the algorithm convergence
parameter integer Nbits_DAC = 14; //Number of bits of the DAC
     (used for truncation)
parameter integer Nbits_ADC = 14; //Number of bits of the ADC
     (used for truncation)
```

Before moving on with the details of the algorithm, it is relevant to highlight that the same Verilog-A block was used for both the model extraction and the assessment of the virtual voltage reference. For this reason, the parameter `collect_mode` is used as a switch through the algorithm in order to disregard the finite DAC and TDC resolution. Besides, the `collect_mode` is used as a flag to stop the algorithm execution after the first iteration. This circuit description shows that the virtual reference concept has been used both for the modelling and achieving a voltage and temperature independent circuit, as discussed in subsection 3.4.1.

```
integer collect_mode = 0;      // Collect_mode = 1 when
    collecting data for modelling
```

Once all the relevant parameters and variables are defined, the `analog` section of the circuit starts. The first part to be executed at the start of the simulation is inside the `initial` construct, as shown in the following listing.

```
@(initial_step)
  begin
    s_Vdd = V(dd, ref);
    r = r_hat;          // Set the first iteration of r to
       r_hat
    m1 = (collect_mode == 1) ? (k1*r) : ($floor(k1*r*(2**
       Nbits_DAC))/(2**Nbits_DAC));
```

```
6       m2 = (collect_mode == 1) ? (k2*r) : ($floor(k2*r*(2**
            Nbits_DAC))/(2**Nbits_DAC));
7       m3 = (collect_mode == 1) ? (k3*r) : ($floor(k3*r*(2**
            Nbits_DAC))/(2**Nbits_DAC));
8       s_Vdac = s_Vdd*m1;  // Assign the first DAC
9       s_ctrlB = 1;        // Close switch at DAC output
10    end
```

The supply voltage is measured during this initial step, as shown in line 3. Considering that the testbenches used for measuring the line regulation and the temperature drift set a constant value of V_{DD} through the transient simulation, it is enough to measure the voltage once at the beginning of the simulation. Lines 4 to 6 show how the DAC's limited resolution is modeled using the `$floor` function. And this also shows how the initial value of `r`, which is $\hat{r}$, is used for setting the values of $\mathtt{m}^{(1)}, \mathtt{m}^{(2)}$ and $\mathtt{m}^{(3)}$. Notice also that the `collect_mode` variable is used to disable the truncation that models the DAC behaviour.

The variable `s_Vdac` represents the DAC voltage inside the Verilog-A model. Then, line 8 of the previous listing shows that this variable is a portion of V_{DD}. Notice that in this case, the variable $\mathtt{m}^{(1)}$ represents a value between 0 and 1, which is the equivalent of stating that the digital word has been normalized with respect to 2^N. Furthermore, this listing also shows how the value of the control signal (`s_ctrlB`) is set to a high value, which means that the capacitor at the output of the DAC will charge up to the voltage `s_Vdac`.

Apart from the initial routine, the rest of the Verilog-A model consists of events that get executed either at the zero crossing of a signal (through the `cross` method) or at a specific time (using the `timer` construct). Following the order of the events, the first event that gets executed corresponds to setting the DAC output to high impedance and closing the switch of the discharge network. This action is performed using the following statement:

```
1   /* First step */
2   @(timer(timeFirstOpenSwitch)) s_ctrlB = 0;
```

Some similar statements are used for setting the DAC output for the second and third steps, as well as for setting the control signal for each step (see lines 132 to 162 in Appendix A). These statements show how the DAC is modelled inside the `ref_algorithm` module.

Another important part of this circuit corresponds to the model of the TDC. For modelling such behaviour, an `if-else` statement is executed when the following `cross` event occurs:

```
1 @(cross(V(adc_in,ref)-thres*Vdd, -1))
```

This statement measures at each simulation step the voltage at the input `adc_in` and then subtracts the quantity `thres*Vdd` from it. This command internally creates a new signal that triggers the event when there is a zero crossing in the negative direction. Once the event is triggered, the value of the variable `step` is compared with its three possible values:

```
1  if (step == 0) begin
2    n1_pre = $abstime - timeFirstOpenSwitch;
3    n1 = (collect_mode == 1) ? n1_pre : ($floor(n1_pre*(2**
        Nbits_ADC))/(2**Nbits_ADC));
4    step = 1;
5    timeSecondCloseSwitch = ($abstime > timeFirstOpenSwitch+
        timeSetDacOutput) ? ($abstime + timeAfterThresholdCross)
         : (timeFirstOpenSwitch+timeSetDacOutput+
        timeAfterThresholdCross); // Make sure that the output
        switch is closed ONLY after the DAC output is set
6  end
7  else if (step == 1)
8    //...
9  else if (step == 2)
10   //...
11 end
```

Notice that this variable has been initialized to `0`. Hence, during the first step, the current simulation time when the ADC input crosses the threshold (`$abstime`) is subtracted by the time the discharge event starts. As shown before, the start time of the discharge is stored in the variable `timeFirstOpenSwitch`. The truncation is also used for the TDC model, using the variable `collect_mode` as a switch. Then, the variable `n1` stores the discharge time measured for the first step.

During this process, the value of the `step` variable is increased, and the time when the second closing of the switch occurs is set. The statement of line 5 ensures that the switch is closed after the DAC output has been set to the next value.

The following listing shows the code snipped inside the `step == 2` case. Since this is the last step of each iteration, the value of `r` must be calculated here.

```
1    r_old = r;
2    r_old_norm = (r_old-1.873999999999996)/0.362580047183432;
        // Set the previous 'r' to 'r_old' to use the formula.
3
4    // Combine PTAT and CTAT
```

```
5   y_CTAT = exp((n1-n2)/(n3-1) - 2);
6   y_CTAT_norm = (y_CTAT-(0.132919955329098))
       /00.0007607323084131251;
7
8   r_CTAT = f1(y_CTAT_norm, r_old_norm); // Full equation not
       shown
9
10  y_PTAT = 100*exp((n1-n2)/(1*(n1-n3)));
11  y_PTAT_norm = (y_PTAT-173.1305858952748)/4.818967756767401;
12
13  r_PTAT = f2(y_PTAT_norm, r_old_norm); // Full equation not
       shown
14  r = (5*r_PTAT + r_CTAT)/6;
```

The first part corresponds to the assignment of $\hat{r}$ (in the script `r_old`) to the current value of `r`. Then, this value is normalized in order to be used by the polynomial model.

The fifth line of the listing presents the equation for calculating the variable y_{CTAT} using `n1`, `n2` and `n3`. Notice that this formula corresponds to Equation (3.36). Then, the polynomial is applied to find the value of r_{CTAT} as in line 8. On the other hand, lines 10-13 show the analogous process for obtaining the value of r_{PTAT}, in this case, using a different polynomial.

The equation shown in line 14 corresponds to the operation described in Equation (3.39). In this case, the value of Γ is set to 5/6. This value was obtained from the comparison of the slopes of the PTAT and the CTAT behaviour (shown in Figures 3.15 and 3.14).

The next listing shows the logic used to decide if convergence has been achieved. In this case, the parameter `relativeError` is used for comparing with respect to the relative difference between the previous and the current value of `r` (as depicted in Figure 3.16). If the convergence has been achieved, the value of the DAC output is set according to `k_final` (lines 2-3); otherwise, the values of `m1`, `m2` and `m3` are set according to the current value of `r` (lines 7-11). Notice that when the convergence has not been achieved, the `step` variable is set to its initial value.

```
1   if (abs(r_old - r)/r < relativeError) begin
2     s_Vdac = s_Vdd*k_final*r; // Prepare DAC output to the
         final value
3     s_convergence = 1;        // Set the convergence flag
4   end
5   else begin
6     // Calculate the m values according to the new r
```

```
7       m1 = (collect_mode == 1) ? (k1*r) : ($floor(k1*r*(2**
           Nbits_DAC))/(2**Nbits_DAC));
8       m2 = (collect_mode == 1) ? (k2*r) : ($floor(k2*r*(2**
           Nbits_DAC))/(2**Nbits_DAC));
9       m3 = (collect_mode == 1) ? (k3*r) : ($floor(k3*r*(2**
           Nbits_DAC))/(2**Nbits_DAC));
10      s_Vdac = s_Vdd*m1;          // Prepare DAC output to the
           first value of the next iteration
11      step = 0;                   // Restart the step counter
12    end // convergence if
```

The last part of the Verilog-A model is in charge of converting the variables `s_Vdac` and `s_ctrlB` into electrical variables that the analog simulator can use. The following statements perform this process:

```
1   // Assign voltage outputs
2   V(ctrlB, ref) <+ transition(s_ctrlB*s_Vdd, tdel, trise,
       tfall);
3   V(dac_out_inner, ref) <+ transition(s_Vdac, tdel, trise,
       tfall);
4
5   // Set the DAC output to the internal voltage or to high
       impedance, depending on the control signal
6   if(s_ctrlB == 1)
7     V(dac_out, ref) <+ V(dac_out_inner, ref);
8   else
9     V(dac_out, ref) <+ (1e+15)*I(dac_out, ref);
```

Notice that the output `dac_out` could be set to the DAC output or to a high impedance ($Z = 1 \times 10^{15}\,\Omega$), depending on the control signal. Besides, the values of the delay and rising and fall time are set to very low values compared to the other timing characteristics of the circuit. In this case, the three parameters are set as 100 fs.

4.2 Implementation of the DAC and TDC

As highlighted in section 1.2, circuits nowadays rely on digital circuits to perform operations traditionally done by analog circuits. The current section focuses on showing that the principle of operation of the virtual voltage reference tailors this trend, and it is possible to obtain a virtual voltage reference by using a majority of digital circuits. subsection 4.2.1 focuses on the design of the DAC by using a digital implementation. In parallel, subsection 4.2.2 describes a possible implementation of a TDC corresponding to a circuit composed of digital components.

4.2.1 Relaxation DAC

The Relaxation DAC [2, 37] (also known as ReDAC) corresponds to a possible architecture used for converting digital words into the analog domain by relying on digital components. The operating principle of the ReDAC is based on the behaviour of charging the capacitor in an RC network. In this case, the capacitor is charged through a stream of rectangular pulses that are the output of the digital circuit.

Figure 4.2 shows the principle of operation of the ReDAC through a block diagram. This block diagram can be divided into two parts:

1. The digital blocks are located to the left: shift register, control unit, and three-state buffer.
2. The RC filter that is used for creating the output analog voltage is connected to the output of the three-state buffer.

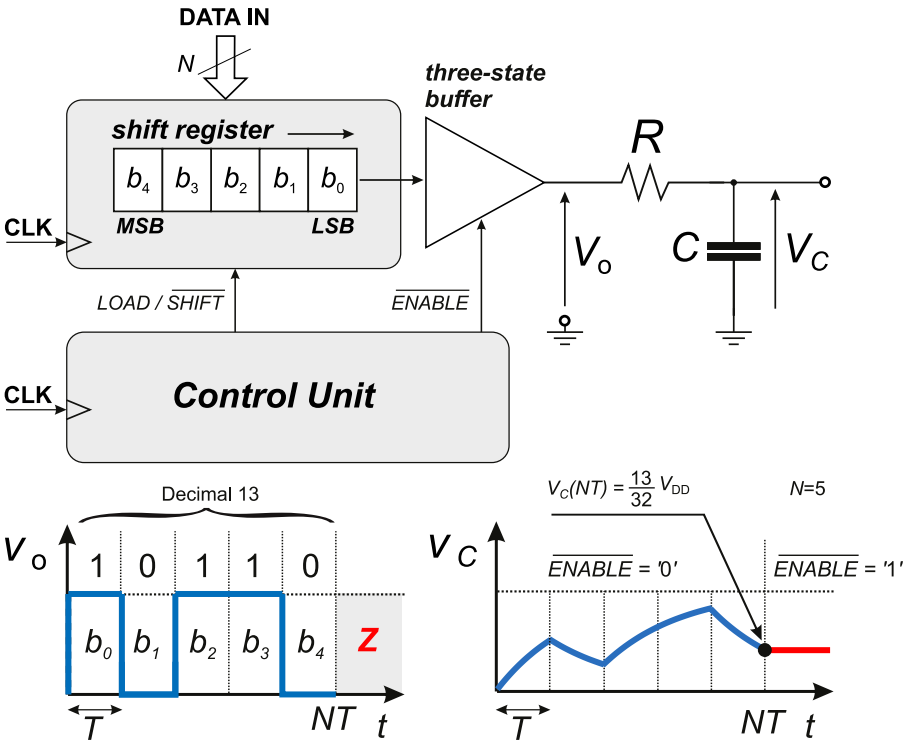


Figure 4.2. Operating principle of the Relaxation DAC. Obtained from [2]

The waveforms on the bottom of Figure 4.2 permit to demonstrate the operation of the ReDAC. The N-bit digital word `DATA IN` is stored in a shift register.

Then, at each clock cycle, the voltage V_o is set to either 0 or V_{DD} according to the current bit at the output of the shift register. These voltages are used to charge or discharge the capacitor during the bitstream stage. Thus, the voltage accumulates to a specific value according to the digital word. Finally, after N clock cycles, the output node will be charged to $\frac{\texttt{DATA\_IN}}{2^N} \cdot V_{DD}$, as it must occur in a digital-to-analog converter.

The three-state buffer is used because, when the N clock cycles have finished, it is necessary to drive the output node of the digital part towards high-impedance in order to avoid discharging the capacitor through the buffer. Notice that this behaviour and the fact that the ReDAC operation relies on the charging of a capacitor show that this implementation of a DAC is well suited as a digital-to-analog interface for the virtual voltage reference circuit.

Another relevant aspect of the ReDAC is that the typical voltage ranges comply with the ranges discussed in this document for the virtual voltage reference. For instance, for the ReDAC design discussed in [22], the circuit operates correctly using a 600 mV voltage supply; besides, it occupies an area of only 677 µm^2 (in 40 nm technology). A deeper analysis regarding the requirements of the number of bits is contained in section 4.3.

4.2.2 Time-to-Digital Converter

As in the case of the DAC, it is also possible to consider a fully-digital implementation of the time-to-digital converter (TDC). The purpose of this circuit is to measure the time elapsed before the signal V_{dis} crosses a certain threshold, as shown in Figure 3.1.

Figure 4.3 depicts a possible implementation of a TDC relying on digital components. This circuit's principle of operation consists of counting the number of clock cycles before the threshold crossing event occurs. This behaviour is accomplished by connecting the output of a comparator between V_{dis} and the threshold voltage to the enable input of a digital counter.

Then, for this implementation of the TDC, it is possible to use an asymmetric inverter (i.e., with a modified switching point) to compare the voltages V_{dis} and the threshold voltage (V_{thres}). Recall from section 3.3 that the threshold voltage is expected to follow the voltage supply V_{DD}: $V_{thres} = K_{thres} \cdot V_{DD}$. Then, it is enough to adjust the ratio of the transistors in the inverter gate to set the threshold to the correct value.

For this particular implementation of the virtual voltage reference, the value of

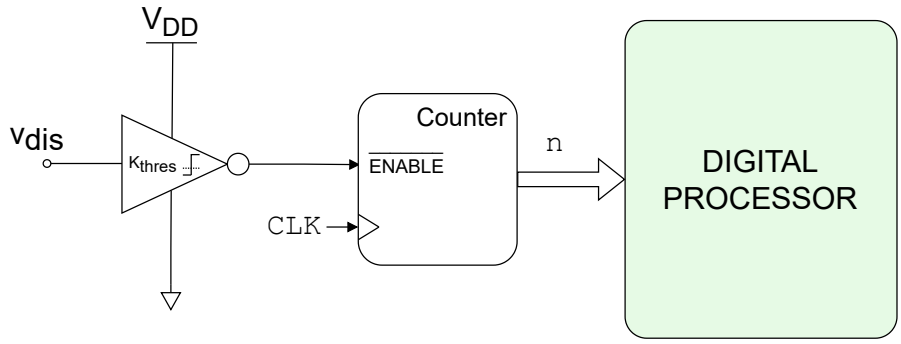


Figure 4.3. Time-to-Digital Converter using a digital circuit.

K_{thres} is 0.25. Hence, it is necessary to size the PMOS transistor to a larger (W/L) ratio than the standard inverter. The opposite occurs in the (W/L) ratio of the NMOS transistor. Recall also that, due to the functions used for obtaining y_{CTAT} and y_{PTAT}, the threshold errors should not drastically affect the output of the virtual voltage reference. The same occurs with the errors in the clock frequency of the TDC.

Another relevant characteristic of the TDC corresponds to choosing the counter's clock frequency. The criteria for choosing this parameter are related to the required time resolution of the circuit. Section 4.3 uses the Verilog-A model to analyze the required resolution of the circuit. This information then allows setting the clock frequency and the number of bits required for the time-to-digital converter.

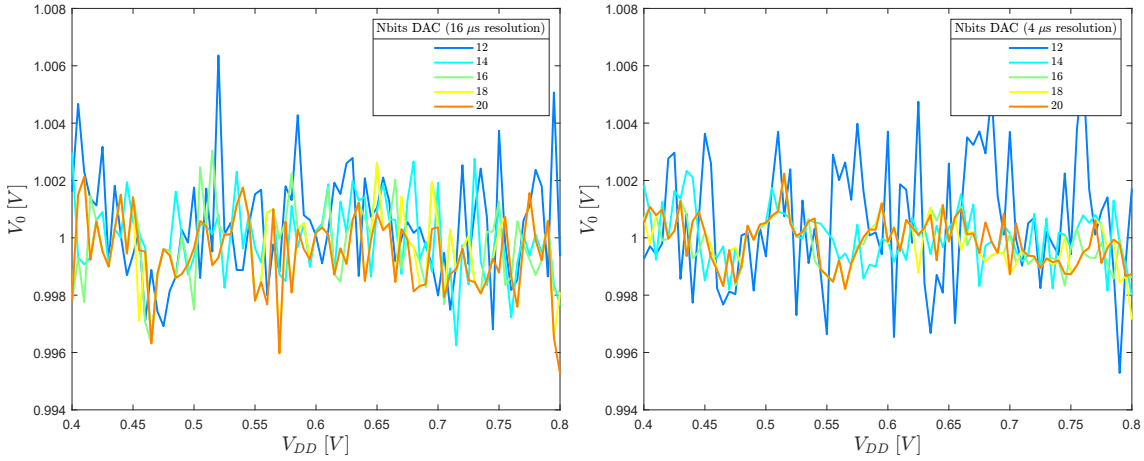
4.3 Analysis of number of bits

After discussing the possible architectures of circuits that serve as an interface between the analog and the digital domain, it is necessary to discuss the impact that the resolution of such circuits could have on the voltage reference circuit. This section discusses the circuit's performance when using different resolutions for the DAC and the TDC. Considering that one of the central figures of merit of the virtual voltage reference corresponds to the line regulation, this figure of merit is used for assessing the circuit behaviour for different resolutions.

As shown in the model of the architecture in section 4.1, the Verilog-A model allows to consider the resolution of the DAC and the TDC by truncating the values of $\mathtt{m}^{(i)}$ and $\mathtt{n}^{(i)}$, respectively. In the case of $\mathtt{m}^{(i)}$, since they correspond to values between 0 and 1, the truncation corresponds directly to the number of bits that are required to well represent this number in a binary representation without affecting

the behaviour of the circuit. On the other hand, for the values of $\mathtt{n}^{(i)}$, in the model, they correspond to time values measured in seconds; then, the resolution, in this case, is interpreted in timing units. The result of the analysis of the TDC resolution will allow recognizing the maximum clock period that should be used for the counter involved in the converter (see Figure 4.3).

Figure 4.4 shows the behavior of the curve of V_0 vs. V_{DD} when using different resolutions for both the DAC and the TDC. Recall that V_0 corresponds to the product between r and V_{DD}, and it is expected to be constant with respect to V_{DD}. Then, in the ideal case, the curves should not be affected by the variation of V_{DD}. In this case, Figure 4.4a corresponds to sweeping the DAC resolution from 12 bits up to 20 bits while setting the TDC to use a 16 µs clock. Instead, Figure 4.4b shows the same DAC resolution sweep when TDC uses a time resolution of 4 µs.



(a) DAC resolution sweep with $f_{CLK} = 16\,\mu s$ (b) DAC resolution sweep with $f_{CLK} = 4\,\mu s$

Figure 4.4. Line regulation for different DAC resolutions

Then, using these curves, the line regulation is calculated and summarised in Table 4.1. As expected, the line regulation degrades when decreasing the resolution of the DAC. This effect is also evident in Figure 4.4. Besides, notice that the difference between using a 16 bits DAC or a 20 bits DAC is not as relevant as the difference between using a 16 µs clock or a clock that is four times faster. This result suggests that the resolution of the TDC is more critical in this design.

Figure 4.5 shows the behaviour of the theoretical voltage V_0 versus V_{DD} for different resolutions of the TDC. When comparing these results with the results shown in Figure 4.4 it is clear that the line regulation is affected by the time resolution

DAC resolution	Line regulation [%/V]	
	16 µs TDC resolution	4 µs TDC resolution
12 bits	2.65	2.38
14 bits	1.70	1.09
16 bits	1.68	1.01
18 bits	1.66	0.87
20 bits	1.60	0.80

Table 4.1. Line regulation for different number of bits of the DAC

more than by the DAC resolution.

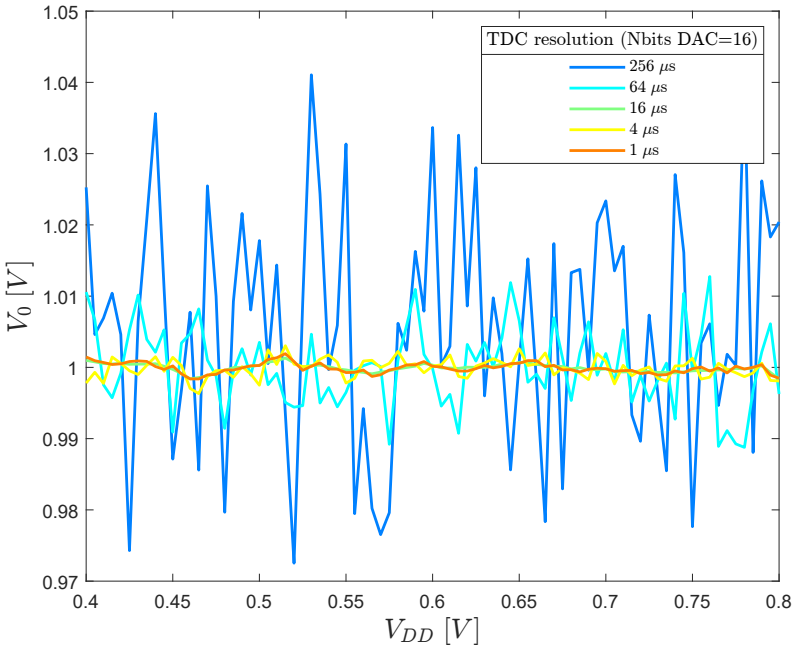


Figure 4.5. Line regulation for different TDC resolutions

Table 4.2 shows the line regulation results for the TDC resolution sweep. Notice that this sweep has the same step of the DAC resolution, i.e., at each step, the resolution is four times less. Then, these results show quantitatively that it is more relevant to choose the resolution of the TDC appropriately than it is for the DAC. For this application, the time resolution chosen corresponds to 4 µs because it corresponds to a good trade-off between the clock frequency and the line regulation.

Besides, the time resolution also sets the number of bits required for the counter

TDC resolution	Line regulation [%/V]
256 µs	17.13
64 µs	5.99
16 µs	1.7
4 µs	0.88
1 µs	0.73

Table 4.2. Line regulation for different TDC resolutions

of the TDC. The number of bits must be enough to count up to the maximum possible discharge time, this can be expressed by the formula:

$$Nbits_{TDC} = \log_2 \left(\frac{t_{discharge\_max}}{T_{CLK}} \right) \tag{4.1}$$

Replacing with the maximum discharge time corresponding to the low-temperature characterization of the circuit allows us to find the number of bits required for the TDC. In this case, the largest measured discharge time is close to 100 ms.

$$Nbits_{TDC} = \log_2 \left(\frac{100\,\text{ms}}{4\,\mu\text{s}} \right) = 15\ bits \tag{4.2}$$

In summary, the previous results allow choosing the resolution for the DAC and the TDC. In this particular implementation of the virtual voltage reference, the DAC is chosen to be a 14-bit DAC. At the same time, the TDC uses a clock with a 250 kHz frequency and requires a 15 bits counter to cover the temperature range that has been considered for this design.

4.4 Layout development

This section discusses the layout implementation of the discharge network. Recall that the discharge network corresponds to the block on the right of Figure 4.1. A detailed schematic representation of this block is shown in Figure 4.6. In this case, the inverter cell corresponds to a standard cell that the silicon vendor provides. Besides, the capacitor corresponds to a MOS capacitor that has been sized accordingly to provide the required capacitance for the ReDAC output: 500 fF. Finally, the discharge structure composed of five PMOS transistors in the 180 nm technology node is also shown in this figure.

Notice that the switch controlling the discharge network is an NMOS transistor with a (W/L) ratio large enough to allow the current to flow without creating a significant voltage drop. Then, the inverter gate that controls this switch is sized

according to the load. Thus, the standard cell that has been selected does not correspond to the smallest inverter available from the vendor. Instead, this standard cell provides a fanout that is double the smallest possible inverter cell.

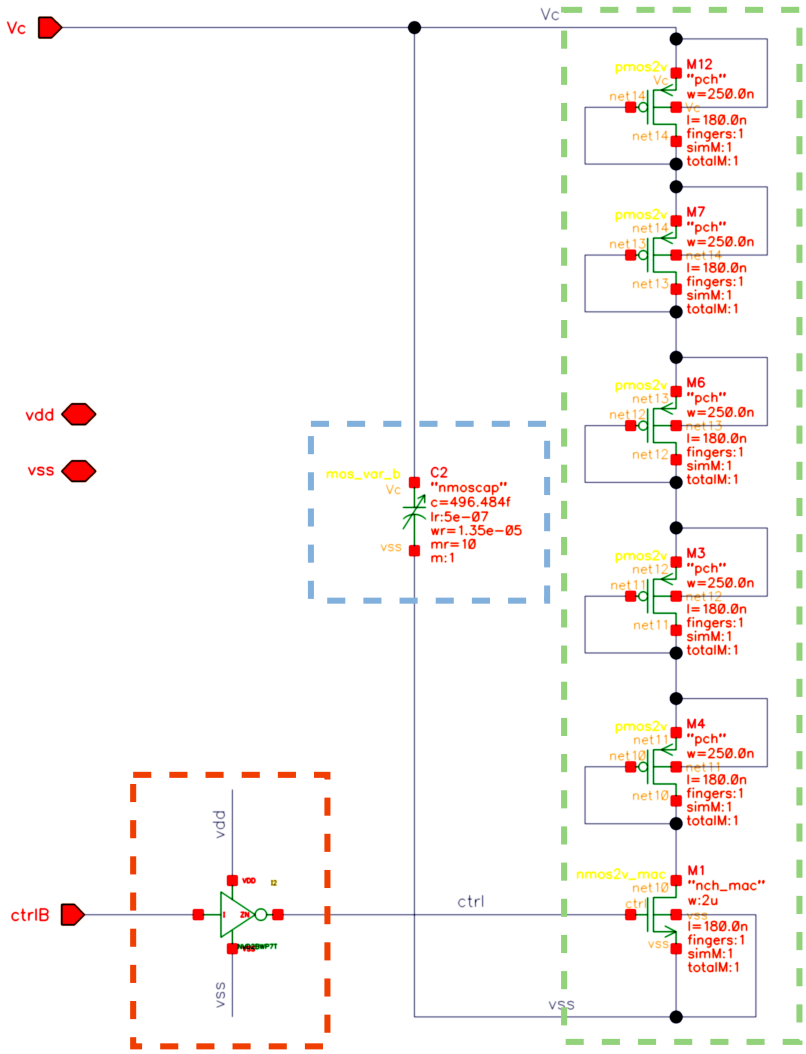


Figure 4.6. Schematic of the discharge network of the virtual voltage reference

After choosing the appropriate cell inverter and sizing the switch correctly, it is possible to perform the layout of the discharge network. Figure 4.7 shows the finished layout that corresponds to the circuit of Figure 4.6. The blue block on the left corresponds to the MOS capacitor created using ten fingers. The block highlighted in red corresponds to the inverter cell, whereas the elements highlighted in green are the transistors used in the discharge network.

Since each of the PMOS transistors has a different connection to the bulk, each device requires a different n-well and a different connection to the well. This characteristic does not allow the transistors to be abutted, as it could be the case if they shared the same n-well. Besides, the minimum distance between the n-wells connected to different nodes pushes towards a larger area of the layout. In the end, the resulting layout corresponds to a rectangle of dimensions 21.1 µm × 22.2 µm. This corresponds to an area of 468 µm^2.

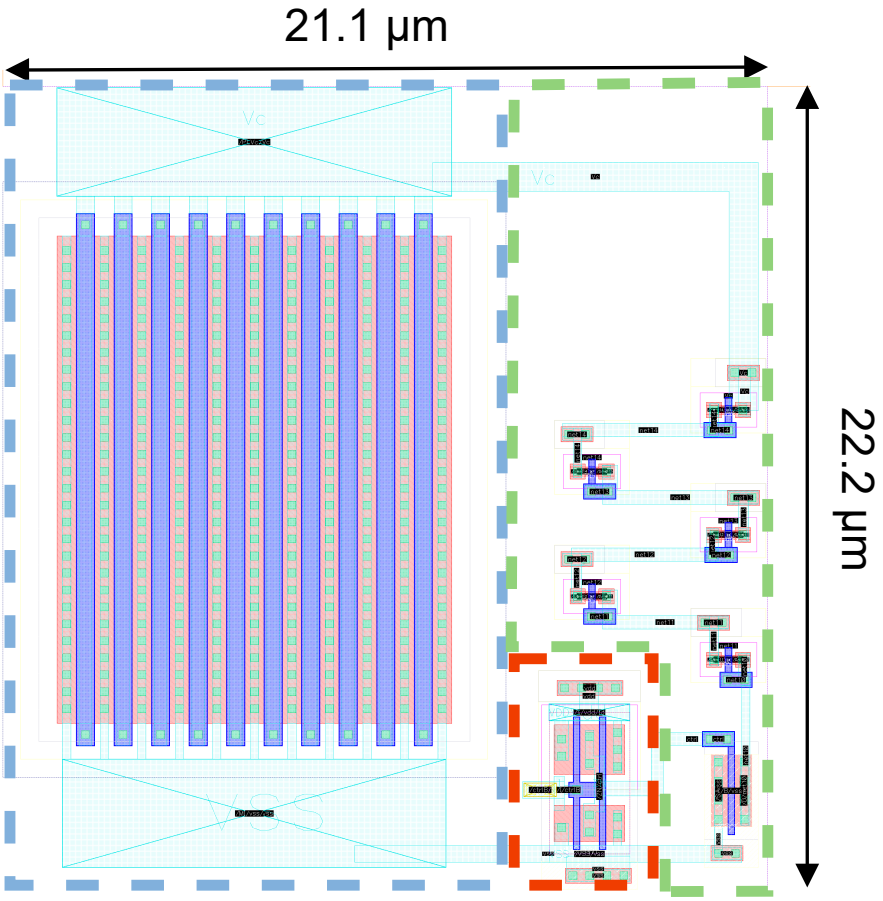


Figure 4.7. Layout of the discharge network of the virtual voltage reference

Chapter 5

Results

In order to show in detail the resulting behaviour of the algorithm, Figure 5.1 depicts the transient simulation of the virtual voltage reference when using as initial value $\hat{\mathtt{r}} = 2$. Notice that, for all the values of V_{DD} in the range from 400 mV to 800 mV, the final value of the DAC output (V_{ref}) corresponds to 350 mV.

The purpose of this figure is to show that the proposed virtual reference is independent of the supply voltage. These results have been obtained using a parametric transient simulation, with a simulation step of 500 ns. Recall that this simulation also considers the limited resolution of the DAC and the TDC as discussed in section 4.3.

Figure 5.1 also shows that it is enough to have two iterations of the algorithm before it converges to a final value within a one percent range. Recall that the threshold for setting the convergence is another parameter set by the Verilog-A script. Besides, notice that this value is always reached after a few hundred milliseconds. The final voltage value of V_{ref} of the transient simulation allows for defining the line regulation, as it is shown in the following.

Figure 5.2 shows the results of the variation of the output voltage of the algorithm with respect to different values of V_{DD}. Then, for each of the curves of Figure 5.1, and also for different values of $\hat{r}$, the final value of V_{ref} is captured and plotted into this new figure. As it can be observed, the output value is very close to the expected final value that corresponds to the equation:

$$V_{ref} = V_0 \cdot k_{final} = 1\,\mathrm{V} \cdot 0.35 = 350\,\mathrm{mV} \tag{5.1}$$

Then, the results shown in the previous two figures allow us to obtain the line regulation of the virtual voltage reference. This figure of merit is calculated using Equation (2.1). In this case the nominal value of V_{ref} corresponds to the value

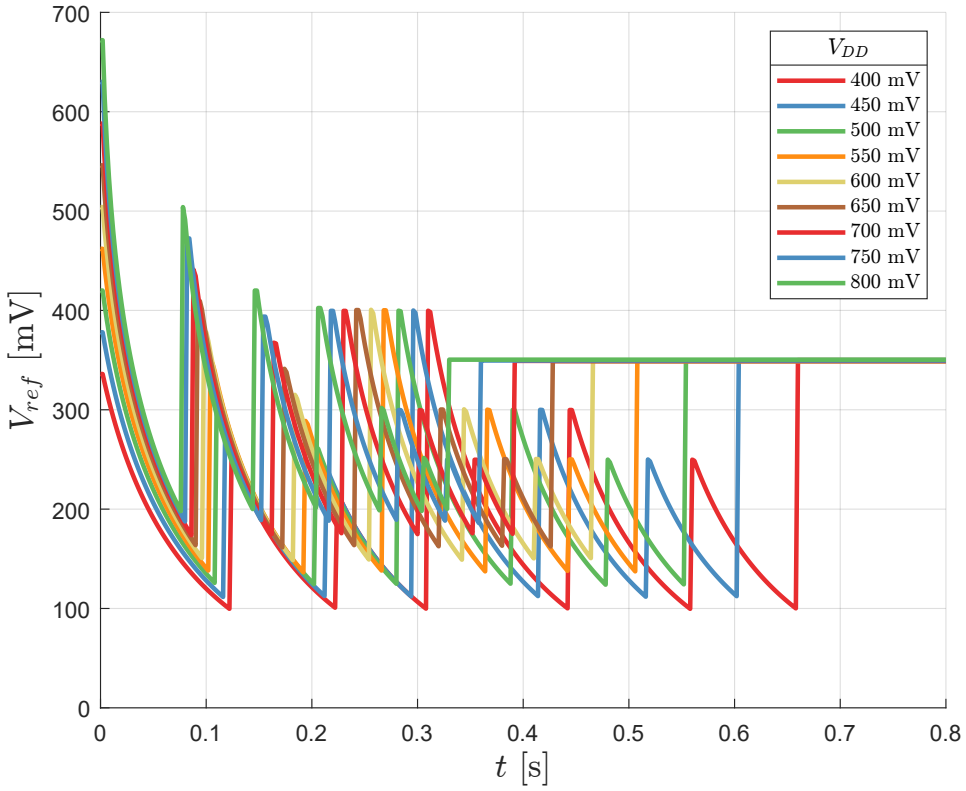


Figure 5.1. Transient simulation of the virtual voltage reference for different V_{DD} values.

calculated in Equation (5.1). Table 5.1 summarises the results of the line regulation for the four different cases of the initial values $\hat{r}$. Notice that the differences between the distinct values chosen as initial points are not significant in this implementation of the voltage reference.

$\hat{\mathbf{r}}$	**Line Regulation**
1.5	1.08 %/V
1.83	1.03 %/V
2.17	1.04 %/V
2.50	1.08 %/V

Table 5.1. Parameters of the polynomial used for fitting the surface

For comparison, the results of the CTAT voltage reference that has been presented in subsection 3.4.1 (Equation (3.37)) are also discussed here. Figure 5.3 presents the line regulation results for that implementation of the virtual voltage reference. Notice that this implementation exhibits a better line regulation for the

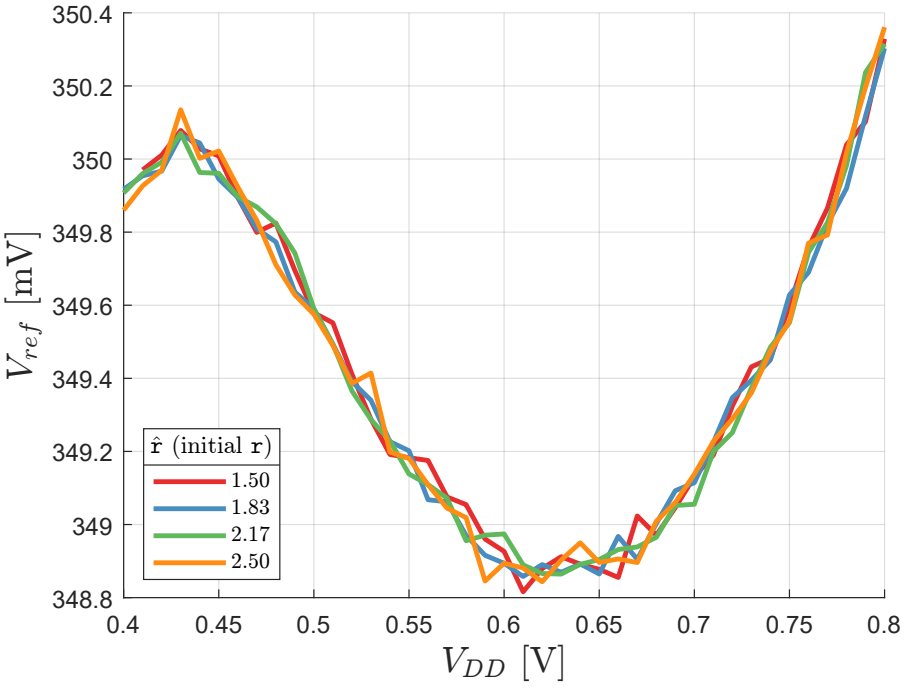


Figure 5.2. Regulated voltage versus V_{DD} for different initial points of the algorithm

same sweep of the supply voltage. In fact, the line regulation in this case is below 0.1 %/V for all the different cases of the initial $\hat{r}$.

However, the disadvantage of this version is that it does not compensate for the temperature variations as the final version of the virtual voltage reference does. Figure 5.4 depicts the temperature behaviour of the virtual voltage reference. This figure also shows the CTAT (in blue) and PTAT (in red) behaviour, as in Figure 3.14 and Figure 3.15, respectively. Then, the green curve represents the final temperature behaviour of the combined primitives. This voltage reference uses the combination of the primitives proposed in Equation (3.39), with $\Gamma = \frac{5}{6}$.

Figure 5.4 was obtained for the case when V_{DD} is set to 500 mV. Besides, this figure allows finding the temperature coefficient of the final virtual voltage reference. Using Equation (2.2), we obtain:

$$TC = \frac{391\,\text{mV} - 342\,\text{mV}}{80\,^{\circ}\text{C} \cdot 350\,\text{mV}} \cdot 10^6 = 1750\ \text{ppm}/^{\circ}\text{C} \tag{5.2}$$

This result is much better than if we had considered only the CTAT part of the virtual voltage reference. Then, these results show a clear trade-off between the line regulation and the temperature coefficient of the virtual voltage reference.

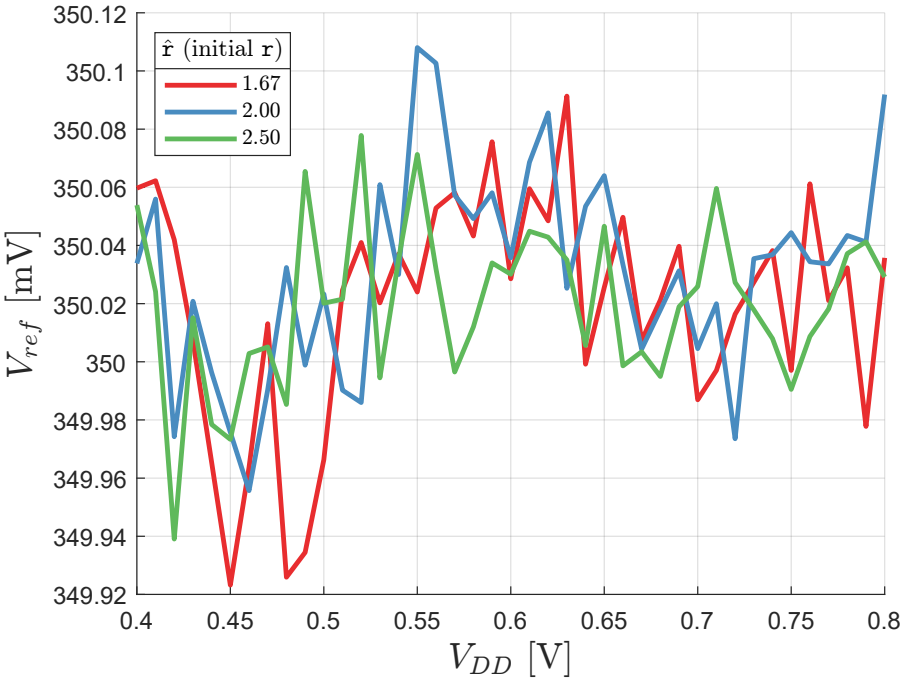


Figure 5.3. Regulated voltage versus V_{DD} for different $\hat{r}$ (CTAT voltage reference)

If the final design would consider only the CTAT reference, that provides a line regulation below 0.1 %/V, then, the temperature coefficient would be very poor: 18320 ppm/°C. For this reason, the final selected design corresponds to the design that compensates both for the temperature and the voltage supply variations.

Table 5.2 shows the comparison between different architectures of voltage references. Both analog and digital virtual references are included in this table. Notice that one of the main advantages of the virtual voltage reference that has been implemented in this work is that the requirement for the voltage supply is very low. The voltage supply could be as low as 600 mV while the voltage reference could be able to provide a 350 mV voltage at the output. The limit of the 600 mV is related to the minimum operating voltage of the ReDAC ([38]) and also of the TDC.

Furthermore, another advantage of the current implementation of the virtual voltage reference corresponds to the significantly low area that is required for the physical reference. This low area is a consequence of choosing small devices and using a load that also fits the requirements of the ReDAC since the load capacitor serves two purposes.

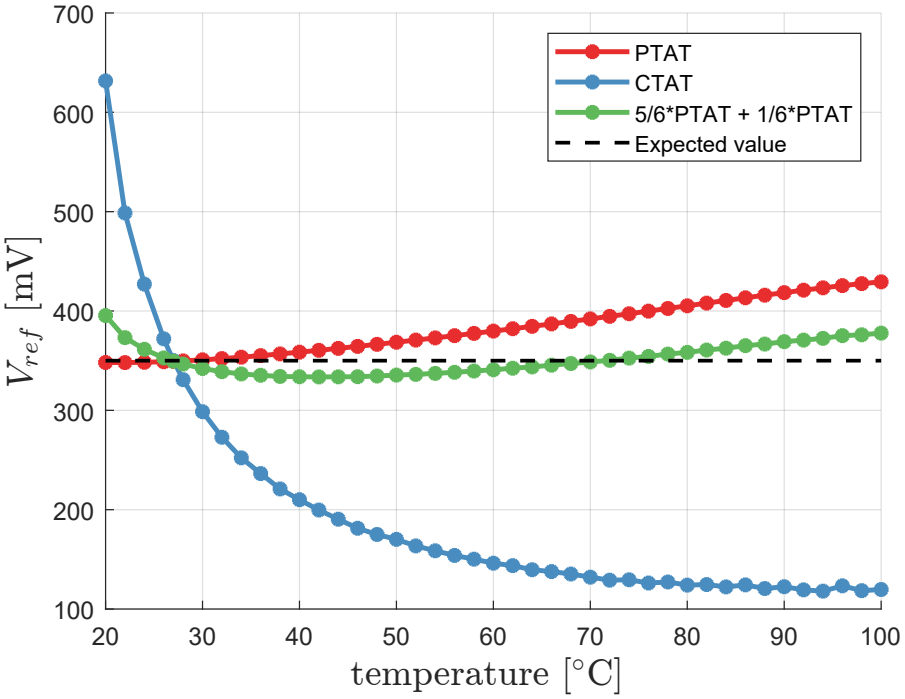


Figure 5.4. Temperature behaviour of the virtual voltage reference

However, regarding the two main figures of merit (line regulation and temperature coefficient), this virtual voltage reference's performance is poor compared with current solutions. These results show that there is room for improvement in the temperature compensation of this design of the virtual voltage reference.

Another relevant feature to verify for the voltage reference corresponds to the independence of the regulated voltage on the threshold that is set for measuring the discharge time. This parameter is known as K_{thres}, with a nominal value of 0.25. Then, in order to measure the dependence of the regulated voltage on K_{thres}, this parameter is varied with a span of 20 %.

From the analysis performed in section 3.3, it is possible to recall that when the primitive is composed of the difference between two timing samples, such primitive is expected to be independent of the value of the threshold. Then, considering that the virtual voltage reference is composed of the CTAT component given by Equation (3.36) and the PTAT component given by Equation (3.38), the expected results are that y_{PTAT} is less dependent of K_{thres}. Consequently, r_{PTAT} is expected to be independent of K_{thres}, in contrast with r_{CTAT} which is expected to vary more with K_{thres}.

	[34] Kuijk	[39] Boo	[40] Huang	[41] Zhou	[42] Crovetti	[35] Crovetti	This work
Published	JSSC	JSSC	TCAS-I	TCAS-I	IET CDS	TCAS-I	N/A
Year	1973	2021	2022	2019	2007	2015	2022
Type	Analog	Analog	Analog	Analog	Analog	Digital	Digital
Tech. [nm]	180	180	180	350	350	180	180
V_{REF} **[mV]**	1210	1142	2141	1141	47	1000	350
LR [%/V]	0.05	N/A	0.015	0.175	1	0.02	1
DAC Res. [bits]	N/A	N/A	N/A	N/A	N/A	16	14
ADC Res. [bits]	N/A	N/A	N/A	N/A	N/A	16	15
T Range [°C]	-40/125	-40/125	-25/125	-40/125	-40/110	-40/140	20/100
TC [ppm/°C]	30	4.3	1.2	1	30	7	1750
Min. V_{DD} **[mV]**	1620	1600	3200	2000	850	900	600
Area [mm^2]	0.0121	0.38	0.256	0.0396	0.0036	0.0006 1.02	0.00046 [a] 0.01405 [b]

[a] Measurement including the physical standard only
[b] Measurement including the physical standard and the ReDAC

Table 5.2. Comparison between different architectures of voltage references

Figure 5.5 shows the results of the simulation when the threshold of the discharge time is varied while using the same virtual reference model. In this case, the values of r_{PTAT}, r_{CTAT} and r are shown for a constant V_{DD} of 500 mV. Notice that these results confirm that the variation with K_{thres} is much larger when the primitives are not composed only by the difference between two discharge times, as it occurs for r_{PTAT}.

The results of the variation of the variable r when considering a 20 % variation for K_{thres} are summarised in Table 5.3. Fortunately, in this case the weighting constant (Γ) in Equation (3.39) allows to have a final value of r that is less dependent of K_{thres}. Consider that this variation could be worse considering that the variation of r_{CTAT} is large.

$\hat{\mathtt{r}}$	**Variation with** K_{thres}
r	0.38 %
r_{CTAT}	2.23 %
r_{PTAT}	0.04 %

Table 5.3. Variation of the virtual voltage reference value with K_{thres}

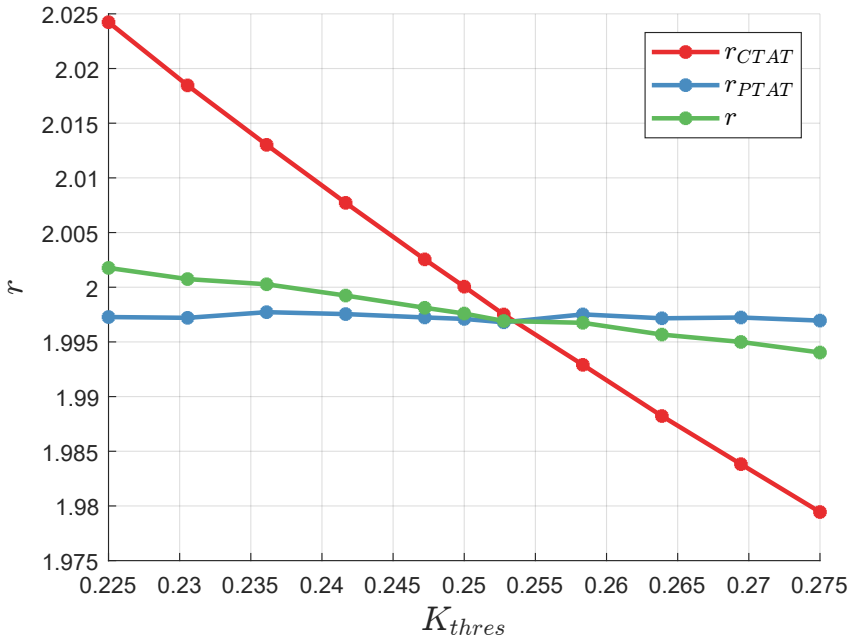


Figure 5.5. Variation of the voltage reference with the discharge time threshold

Bibliography

[1] M. M. Alam, S. Chowdhury, B. Park, D. Munzer, N. Maghari, M. Tehranipoor, and D. Forte, "Challenges and opportunities in analog and mixed signal (ams) integrated circuit (ic) security," *J. Hardware and Systems Security*, vol. 2, no. 1, pp. 15–32, 2018.

[2] P. Crovetti, R. Rubino, A. Abdullah, and F. Musolino, "Emerging relaxation and ddpm d/a converters: Overview and perspectives," in *2022 IEEE 65th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1–6, 2022.

[3] J. Kahn, R. Fellow, and K. Pister, "Next century challenges: Mobile networking for "smart dust"," 06 1999.

[4] S. Khan and M. Hamid, "On the optimal number of smart dust particles," in *7th International Multi Topic Conference, 2003. INMIC 2003.*, pp. 472–475, 2003.

[5] S. Carrara and P. Georgiou, "Body dust: Miniaturized highly-integrated low power sensing for remotely powered drinkable cmos bioelectronics," 04 2018.

[6] R. L. Weinstein, S. L. Schwartz, R. L. Brazg, J. R. Bugler, T. A. Peyser, and G. V. McGarraugh, "Accuracy of the 5-Day FreeStyle Navigator Continuous Glucose Monitoring System: Comparison with frequent laboratory reference measurements," *Diabetes Care*, vol. 30, pp. 1125–1130, 05 2007.

[7] S. Carrara, "Body dust: Well beyond wearable and implantable sensors," *IEEE Sensors Journal*, vol. 21, no. 11, pp. 12398–12406, 2021.

[8] K. Kiyoyama, K.-W. Lee, T. Fukushima, H. Naganuma, H. Kobayashi, T. Tanaka, and M. Koyanagi, "A very low area adc for 3-d stacked cmos image processing system," in *2011 IEEE International 3D Systems Integration Conference (3DIC), 2011 IEEE International*, pp. 1–4, 2012.

[9] M. Furuta, M. Nozawa, and T. Itakura, "A 0.06mm2 8.9b enob 40ms/s pipelined sar adc in 65nm cmos," in *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*, pp. 382–383, 2010.

[10] K. Cheng, A. Khakifirooz, P. Kulkarni, S. Ponoth, B. Haran, A. Kumar, T. Adam, A. Reznicek, N. Loubet, H. He, J. Kuss, M. Wang, T. M. Levin,

F. Monsieur, Q. Liu, R. Sreenivasan, J. Cai, A. Kimball, S. Mehta, S. Luning, Y. Zhu, Z. Zhu, T. Yamamoto, A. Bryant, C. H. Lin, S. Naczas, H. Jagannathan, L. F. Edge, S. Allegret-Maret, A. Dube, S. Kanakasabapathy, S. Schmitz, A. Inada, S. Seo, M. Raymond, Z. Zhang, A. Yagishita, J. Demarest, J. Li, M. Hopstaken, N. Berliner, A. Upham, R. Johnson, S. Holmes, T. Standaert, M. Smalley, N. Zamdmer, Z. Ren, T. Wu, H. Bu, V. Paruchuri, D. Sadana, V. Narayanan, W. Haensch, J. O'Neill, T. Hook, M. Khare, and B. Doris, "Etsoi cmos for system-on-chip applications featuring 22nm gate length, sub-100nm gate pitch, and 0.08um2 sram cell," in *2011 Symposium on VLSI Circuits - Digest of Technical Papers*, pp. 128–129, 2011.

[11] P. Toledo, P. S. Crovetti, H. D. Klimach, F. Musolino, and S. Bampi, "Low-voltage, low-area, nw-power cmos digital-based biosignal amplifier," *IEEE Access*, vol. 10, pp. 44106–44115, 2022.

[12] B. Cook, S. Lanzisera, and K. Pister, "Soc issues for rf smart dust," *Proceedings of the IEEE*, vol. 94, no. 6, pp. 1177–1196, 2006.

[13] K. Ashton, "That 'internet of things' thing."

[14] M. Alioto, *Enabling the Internet of Things: From Integrated Circuits to Integrated Systems*. 01 2017.

[15] "2021 - international roadmap for devices and systems." `https://irds.ieee.org/images/files/pdf/2021/2021IRDS_ES.pdf`.

[16] B. Xu, S. Li, N. Sun, and D. Z. Pan, "A scaling compatible, synthesis friendly vco-based delta-sigma adc design and synthesis methodology," in *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1–6, 2017.

[17] V. Ivanov, R. Brederlow, and J. Gerber, "An ultra low power bandgap operational at supply from 0.75 v," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 7, pp. 1515–1523, 2012.

[18] T. Oshita, J. Shor, D. E. Duarte, A. Kornfeld, and D. Zilberman, "Compact bjt-based thermal sensor for processor applications in a 14 nm tri-gate cmos process," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 3, pp. 799–807, 2015.

[19] P. Crovetti, F. Musolino, O. Aiello, P. Toledo, and R. Rubino, "Breaking the boundaries between analogue and digital," *Electron. Lett.*, vol. 55, pp. 672–673, 2019.

[20] P. Toledo, R. Rubino, F. Musolino, and P. Crovetti, "Re-thinking analog integrated circuits in digital terms: A new design concept for the iot era," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 3, pp. 816–822, 2021.

[21] E. Ansari and D. D. Wentzloff, "A 5mw 250ms/s 12-bit synthesized digital to analog converter," in *Proceedings of the IEEE 2014 Custom Integrated Circuits Conference*, pp. 1–4, 2014.

[22] P. S. Crovetti, R. Rubino, and F. Musolino, "Relaxation digital-to-analog converter with foreground digital self-calibration," in *2020 IEEE International*

Symposium on Circuits and Systems (ISCAS), pp. 1–5, 2020.

[23] M.-J. Seo, Y.-J. Roh, D.-J. Chang, W. Kim, Y.-D. Kim, and S.-T. Ryu, "A reusable code-based sar adc design with cdac compiler and synthesizable analog building blocks," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 12, pp. 1904–1908, 2018.

[24] S. Weaver, B. Hershberg, and U.-K. Moon, "Digitally synthesized stochastic flash adc using only standard digital cells," in *2011 Symposium on VLSI Circuits - Digest of Technical Papers*, pp. 266–267, 2011.

[25] X. Ma, Y. Lu, R. P. Martins, and Q. Li, "A 0.4v 430na quiescent current nmos digital ldo with nand-based analog-assisted loop in 28nm cmos," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, pp. 306–308, 2018.

[26] G. Cai, C. Zhan, and Y. Lu, "A fast-transient-response fully-integrated digital ldo with adaptive current step size control," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–4, 2019.

[27] R. Staszewski, K. Muhammad, D. Leipold, C.-M. Hung, Y.-C. Ho, J. Wallberg, C. Fernando, K. Maggio, T. Jung, J. Koh, S. John, I. Deng, V. Sarda, O. Moreira-Tamayo, V. Mayega, R. Katz, O. Friedman, O. Eliezer, E. De Obaldia, and P. Balsara, "All-digital tx frequency synthesizer and discrete-time receiver for bluetooth radio in 130-nm cmos," *Solid-State Circuits, IEEE Journal of*, vol. 39, pp. 2278 – 2291, 01 2005.

[28] F. T. Gebreyohannes, A. Frappé, P. Cathelin, A. Cathelin, and A. Kaiser, "All-digital transmitter architecture based on two-path parallel 1-bit high pass filtering dacs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 11, pp. 3956–3969, 2018.

[29] H. Chen, M. Liu, T. Xiyuan, K. Zhu, N. Sun, and D. Pan, "Challenges and opportunities toward fully automated analog layout design," *Journal of Semiconductors*, vol. 41, p. 111407, 11 2020.

[30] T. Dhar, K. Kunal, Y. Li, M. Madhusudan, J. Poojary, A. K. Sharma, W. Xu, S. M. Burns, R. Harjani, J. Hu, D. A. Kirkpatrick, P. Mukherjee, S. Yaldiz, and S. S. Sapatnekar, "Align: A system for automating analog layout," *IEEE Design Test*, vol. 38, no. 2, pp. 8–18, 2021.

[31] H. Chen, M. Liu, B. Xu, K. Zhu, X. Tang, S. Li, Y. Lin, N. Sun, and D. Z. Pan, "Magical: An open- source fully automated analog ic layout system from netlist to gdsii," *IEEE Design And Test*, vol. 38, no. 2, pp. 19–26, 2021.

[32] R. J. Widlar, "Some circuit design techniques for linear integrated circuits," *IEEE Transactions on Circuit Theory*, vol. 12, pp. 586–590, 1965.

[33] A. Brokaw, "A simple three-terminal ic bandgap reference," *IEEE Journal of Solid-State Circuits*, vol. 9, no. 6, pp. 388–393, 1974.

[34] K. Kuijk, "A precision reference voltage source," *IEEE Journal of Solid-State Circuits*, vol. 8, no. 3, pp. 222–226, 1973.

[35] P. S. Crovetti, "A digital-based virtual voltage reference," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, pp. 1315–1324, 2015.

[36] B. Razavi, *Design of Analog CMOS Integrated Circuits*. Electrical Engineering Series, McGraw-Hill, 2001.

[37] R. Rubino, P. S. Crovetti, and F. Musolino, "Fpga-based relaxation d/a converters with parasitics-induced error suppression and digital self-calibration," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 6, pp. 2494–2507, 2021.

[38] R. Rubino, P. S. Crovetti, and O. Aiello, "Design of relaxation digital-to-analog converters for internet of things applications in 40nm cmos," in *2019 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, pp. 13–16, 2019.

[39] J.-H. Boo, K.-I. Cho, H.-J. Kim, J.-G. Lim, Y.-S. Kwak, S.-H. Lee, and G.-C. Ahn, "A single-trim switched capacitor cmos bandgap reference with a 3-sigma inaccuracy of 0.02 for battery-monitoring applications," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 4, pp. 1197–1206, 2021.

[40] S. Huang, M. Li, H. Li, P. Yin, Z. Shu, A. Bermak, and F. Tang, "A sub-1 ppm/c bandgap voltage reference with high-order temperature compensation in 0.18-um cmos process," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 4, pp. 1408–1416, 2022.

[41] Z.-K. Zhou, Y. Shi, Y. Wang, N. Li, Z. Xiao, Y. Wang, X. Liu, Z. Wang, and B. Zhang, "A resistorless high-precision compensated cmos bandgap voltage reference," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 1, pp. 428–437, 2019.

[42] P. Crovetti and F. Fiori, "Compact, very low voltage, temperature-independent reference circuit," *Circuits, Devices And Systems, IET*, vol. 1, pp. 63 – 71, 03 2007.

Appendix A

Verilog-A model of the virtual reference algorithm

```
1  // Algorithm for creating initial values for the discharge of
       the capacitor and also for measuring the discharge time
     of the cell
2  // Author: Gabriel Romero
3  //
4  // The purpose of this block is to create the  analog
     voltages and then
5  // measure the discharge time of the capacitor
6
7  `include "constants.vams"
8  `include "disciplines.vams"
9
10 module ref_algorithm(dd, ref, adc_in, dac_out, ctrlB);
11 output dac_out; electrical dac_out; // DAC output of the
     algorithm
12 input  adc_in;  electrical adc_in;  // ADC (TDC) input to be
     sampled
13 electrical dd;                     // Supply voltage input
14 electrical ref;                    // Ground node
15 electrical ctrlB;                  // Control signal for the
       discharge network switch
16
17 electrical dac_out_inner;          // Internal signal that
     stores the DAC output before the output stage (high-Z
     control)
18
19 /* Parameters used for the analog construct 'transition' */
20 parameter real trise = 0.1e-9;
```

```
21 parameter real tfall = 0.1e-9;
22 parameter real tdel = 0.1e-9;
23
24 /* Parameters used for the virtual reference algorithm */
25 parameter real r_hat = 1.5;            // Initial r (first
      iteration)
26 parameter real thres = 0.25;           // Discharge time
      threshold (K_thres)
27 parameter real k1 = 0.4;               // k1 constant
28 parameter real k2 = 0.3;               // k2 constant
29 parameter real k3 = 0.25;              // k3 constant
30 parameter real k_final = 0.35;         // Final constant k
      that sets the regulated voltage
31 parameter real relativeError = 0.01;   // Relative error for
      assessing the algorithm convergence
32 parameter integer Nbits_DAC = 20;      // Number of bits of
      the DAC (used for truncation)
33 parameter integer Nbits_ADC = 20;      // Number of bits of
      the ADC (used for truncation)
34
35 /* Auxiliary signals*/
36 real s_Vdd, s_Vdac, s_ctrlB;  // Signals used internally for
      measuring Vdd and setting the outputs
37 real s_convergence = 0;        // Signal representing the
      algorithm convergence
38 real m1, m2, m3;               // Initial values of the DAC
      output (ranges btw. 0 and 1)
39 real n1, n2, n3;               // Discharge times measured (
      ADC)
40 real y, y_norm;                // 'y' variable of the
      algorithm and its normalized version
41 real r_old, r_old_norm;        // Virtual reference variable '
      r'
42 real r, r_PTAT, r_CTAT;        // Virtual reference variable '
      r'
43 integer step = 0;              // Counter for the algorithm
      steps (0 -> Collecting n1; 1 -> Collecting n2; 2 ->
      Collecting n3)
44 integer collect_mode = 0;      // Collect_mode = 1 when
      collecting data
45 integer iteration_counter = 0;// Alternative exit sequence
46
47 /* Signals for controlling the timing of the switches */
48 real timeOpenSwitch = 3e-3;
49 real timeSetDacOutput = 1e-3;
50 real timeAfterThresholdCross = 1e-3;
```

```
51 real timeFirstOpenSwitch = 2e-3;
52 real timeSecondOpenSwitch = 1e9;
53 real timeThirdOpenSwitch = 1e9;
54 real timeSecondCloseSwitch = 1e9;
55 real timeThirdCloseSwitch = 1e9;
56 real timeFourthCloseSwitch = 1e9;
57
58 real n1_pre, n2_pre, n3_pre;
59
60 analog begin
61   // Initial iteration
62   @(initial_step)
63   begin
64     s_Vdd = V(dd, ref);
65     r = r_hat;          // Set the first iteration of r to
         r_hat
66     m1 = (collect_mode == 1) ? (k1*r) : ($floor(k1*r*(2**
         Nbits_DAC))/(2**Nbits_DAC));
67     m2 = (collect_mode == 1) ? (k2*r) : ($floor(k2*r*(2**
         Nbits_DAC))/(2**Nbits_DAC));
68     m3 = (collect_mode == 1) ? (k3*r) : ($floor(k3*r*(2**
         Nbits_DAC))/(2**Nbits_DAC));
69     s_Vdac = s_Vdd*m1;  // Assign the first DAC
70     s_ctrlB = 1;        // Close switch at DAC output
71   end
72
73   // Obtain discharge time and control the iterations when
       crossings occur
74   @(cross(V(adc_in,ref)-thres*s_Vdd, -1))
75   begin
76     if (step == 0) begin
77       n1_pre = $abstime - timeFirstOpenSwitch;
78       n1 = (collect_mode == 1) ? n1_pre : ($floor(n1_pre*(2**
           Nbits_ADC))/(2**Nbits_ADC));
79       step = 1;
80       timeSecondCloseSwitch = ($abstime > timeFirstOpenSwitch
           +timeSetDacOutput) ? ($abstime +
           timeAfterThresholdCross) : (timeFirstOpenSwitch+
           timeSetDacOutput+timeAfterThresholdCross); // Make
           sure that the output switch is closed ONLY after the
            DAC output is set
81     end
82     else if (step == 1) begin
83       n2_pre = $abstime - timeSecondOpenSwitch;
84       n2 = (collect_mode == 1) ? n2_pre : ($floor(n2_pre*(2**
           Nbits_ADC))/(2**Nbits_ADC));
```

```
85          step = 2;
86          timeThirdCloseSwitch = ($abstime > timeSecondOpenSwitch
               +timeSetDacOutput) ? ($abstime +
               timeAfterThresholdCross) : (timeSecondOpenSwitch+
               timeSetDacOutput+timeAfterThresholdCross); // Make
               sure that the output switch is closed ONLY after the
                DAC output is set
87      end
88      else if (step == 2) begin
89          n3_pre = $abstime - timeThirdOpenSwitch;
90          n3 = (collect_mode == 1) ? n3_pre : ($floor(n3_pre*(2**
               Nbits_ADC))/(2**Nbits_ADC));
91
92          r_old = r;
93          r_old_norm = (r_old-1.873999999999996)
               /0.362580047183432; // Set the previous 'r' to '
               r_old' to use the formula.
94
95          y = exp((n1-n2)/(n3-1) - 2);
96          y_norm = (y-(0.132919955329098))
               /00.0007607323084131251;
97
```

```
98        r_CTAT = (673755449305469*r_old_norm)/2251799813685248
             - (7571652985603477*y_norm)/18014398509481984 -
             (4365014933572403*r_old_norm**2*y_norm**2)
             /9223372036854775808 + (1290239191088913*r_old_norm
             **2*y_norm**3)/18446744073709551616 +
             (6033762861234771*r_old_norm**3*y_norm**2)
             /36893488147419103232 - (8594838985494325*r_old_norm
             *y_norm)/144115188075855872 + (1328628026108131*
             r_old_norm*y_norm**2)/288230376151711744 +
             (994169870894269*r_old_norm**2*y_norm)
             /5764607523034234888 - (2789138658474469*r_old_norm*
             y_norm**3)/4611686018427387904 - (4734082878298817*
             r_old_norm**3*y_norm)/73786976294838206464 -
             (3445847528079745*r_old_norm*y_norm**4)
             /9223372036854775808 - (3567928347400101*r_old_norm
             **4*y_norm)/59029581035870571 2 -
             (1921756738470021*r_old_norm**2)/5764607523034234888
             + (8057430154330161*r_old_norm**3)
             /36893488147419103232 - (1230760004431959*r_old_norm
             **4)/36893488147419103232 + (6530074602252195*
             r_old_norm**5)/118059162071741130342 4 +
             (4996515669152669*y_norm**2)/72057594037927936 -
             (755215517163547*y_norm**3)/36028797018963968 +
             (3218847922929037*y_norm**4)/5764607523034234888 -
             (5790306723199905*y_norm**5)/4611686018427387904 +
             7655721418912779/4503599627370496;
99
100       y = 100*exp((n1-n2)/(1*(n1-n3)));
101       y_norm = (y-173.1305858952748)/4.818967756767401;
102
```

```
103         r_PTAT = (5715848509975757*r_old_norm)
               /18014398509481984 + (7434881826675035*y_norm)
               /18014398509481984 - (8385078747515265*r_old_norm
               **2*y_norm**2)/576460752303423488 -
               (3783891236640757*r_old_norm**2*y_norm**3)
               /2305843009213693952 + (4991778148263825*r_old_norm
               **3*y_norm**2)/1152921504606846976 +
               (6507712449075073*r_old_norm*y_norm)
               /72057594037927936 + (6125931868081057*r_old_norm*
               y_norm**2)/144115188075855872 + (2851754763770505*
               r_old_norm**2*y_norm)/288230376151711744 -
               (3690775709261481*r_old_norm*y_norm**3)
               /576460752303423488 - (90782928644667*r_old_norm**3*
               y_norm)/18014398509481984 - (4507129231277491*
               r_old_norm*y_norm**4)/576460752303423488 +
               (6223262994699357*r_old_norm**2)
               /1844674407370955161 6 - (1238737261791977*r_old_norm
               **3)/4611686018427387904 + (8739078832649*y_norm**2)
               /70368744177664 + (368991164045485*y_norm**3)
               /4503599627370496 + (2658791938901859*y_norm**4)
               /72057594037927936 + (155967684760767*y_norm**5)
               /36028797018963968 +
               7399787364939711/4503599627370496;
104
105         // Combine PTAT and CTAT
106         r = (5*r_PTAT + r_CTAT)/6;
107
108         // Check the condition for convergence
109         if ( collect_mode == 1 ) begin
110           s_convergence = 1;
111         end
112         else begin
113
114           if (abs(r_old - r)/r < relativeError) begin
115             s_Vdac = s_Vdd*k_final*r; // Prepare DAC output to
                  the final value
116             s_convergence = 1;        // Set the convergence
                  flag
117           end
118           else begin
119             // Calculate the m values according to the new r
120             m1 = (collect_mode == 1) ? (k1*r) : ($floor(k1*r
                  *(2**Nbits_DAC))/(2**Nbits_DAC));
121             m2 = (collect_mode == 1) ? (k2*r) : ($floor(k2*r
                  *(2**Nbits_DAC))/(2**Nbits_DAC));
```

```
122             m3 = (collect_mode == 1) ? (k3*r) : ($floor(k3*r
                   *(2**Nbits_DAC))/(2**Nbits_DAC));
123             s_Vdac = s_Vdd*m1;          // Prepare DAC output to
                   the first value of the next iteration
124             step = 0;                   // Restart the step
                   counter
125           end // convergence if
126         end // collect_mode if
127
128         timeFourthCloseSwitch = $abstime +
               timeAfterThresholdCross; // Set the fourth step flag
                to check for the convergence flag. Necessary to
               avoid setting the output and closing the switch in
               the same instant (convergency errors)
129       end
130     end
131
132     /* First step */
133     @(timer(timeFirstOpenSwitch)) s_ctrlB = 0;
134
135     @(timer(timeFirstOpenSwitch+timeSetDacOutput)) s_Vdac =
           s_Vdd*m2; // Prepare DAC voltage for the next iteration
136
137     /* Second step */
138     @(timer(timeSecondCloseSwitch))
139     begin
140       s_ctrlB = 1;
141       timeSecondOpenSwitch = $abstime + timeOpenSwitch;
142     end
143
144     @(timer(timeSecondOpenSwitch)) s_ctrlB = 0;
145
146     @(timer(timeSecondOpenSwitch+timeSetDacOutput)) s_Vdac =
           s_Vdd*m3; // Prepare DAC voltage for the next iteration
147
148     /* Third step */
149     @(timer(timeThirdCloseSwitch))
150     begin
151       s_ctrlB = 1;
152       timeThirdOpenSwitch = $abstime + timeOpenSwitch;
153     end
154
155     @(timer(timeThirdOpenSwitch)) s_ctrlB = 0;
156
157     /* Fourth (or first) step */
158     @(timer(timeFourthCloseSwitch))
```

```
159   begin
160     s_ctrlB = 1;
161     if (s_convergence == 0) timeFirstOpenSwitch = $abstime +
          timeOpenSwitch;
162   end
163
164   // Assign voltage outputs
165   V(ctrlB, ref) <+ transition(s_ctrlB*s_Vdd, tdel, trise,
        tfall);
166   V(dac_out_inner, ref) <+ transition(s_Vdac, tdel, trise,
        tfall);
167
168   // Set the DAC output to the internal voltage or to high
        impedance, depending on the control signal
169   if(s_ctrlB == 1)
170     V(dac_out, ref) <+ V(dac_out_inner, ref);
171   else
172     V(dac_out, ref) <+ (1e+15)*I(dac_out, ref);
173
174 end
175 endmodule
```