

# Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática  
**Especialidad en Diseño de Sistemas en Chip**



## **Voltaje de Referencia BandGap y Módulo de Comunicación Serial para SAR ADC 10 Bits de Baja Potencia para Aplicaciones Biomédicas**

---

**TRABAJO RECEPCIONAL** que para obtener el **GRADO** de  
**ESPECIALISTA EN DISEÑO DE SISTEMAS EN CHIP**

Presenta: **LUIS ANGEL GONZÁLEZ ORNELAS**

Director: **JOSE LUIS CHÁVEZ HURTADO, ESTEBAN MARTINEZ  
GUERRERO Y CUAUHTÉMOC RAFAEL AGUILERA GALICIA**

Tlaquepaque, Jalisco. Octubre de 2021.

# ITESO – The Jesuit University of Guadalajara

Department of Electronic, System, and Informatics.  
Specialization Program in SoC Design.



## **BandGap Reference Voltage and Serial Communication Module for a Low Power 10 Bits SAR ADC to Biomedical Applications**

Thesis/Project to achieve the degree of SoC Design Specialist

Present: Luis Angel González Ornelas

Thesis Director: Jose Luis Chávez Hurtado, Esteban Martinez Guerrero and Cuauhtémoc Rafael  
Aguilera Galicia.

Tlaquepaque, Jalisco. October 2021



## **Acknowledgments**

I principally would like to thank my family, especially my mother that who has always supported me in every stage of my life.

Then thanks to NXP Semiconductors and CONACYT, both entities help me financially to achieve this goal in my personal and professional life.

I want to thank Esteban Martinez Guerrero for the design of the Bandgap voltage reference circuit that is presented in this work.

Finally, I would like to thank ITESO that has guided us in a better way to reach this new goal.



# Abstract

The document presents two designs a BandGap Reference Voltage, and a Communication Serial Module for a 10 bits SAR ADC for low-power applications. Designs were implemented using TSMC 0.18  $\mu\text{m}$  CMOS technology with 1.8 V supply voltage.

The BandGap Reference Voltage was designed to provide a reference voltage of 900 mV  $\pm 500 \mu\text{V}$ . The bandgap was tested at simulation level under different temperature conditions to ensure constant output in a temperature range from  $-40 \text{ }^\circ\text{C}$  to  $85 \text{ }^\circ\text{C}$ . The Communication Serial Module is designed using the hardware description language Verilog. This module receives the 10 bits parallel output of the SAR ADC and retransmits the conversion result into a serial format using the SPI format. The Communication Serial Module was tested under a simulator, where multiple test cases were applied to stimulate in different ways the module.

Both circuits were designed to accomplish the SAR ADC requirements in which BandGap supplies the reference voltage to the capacitor array in the SAR ADC and the Serial Module sends the data values after the conversion is finalized.



## Acronyms and Abbreviation

ADC	Analog to Digital Converter.
BGR	BandGap Reference.
DAC	Digital to Analog Converter.
SAR	Successive Approximation Registers.
PVT	Process, Voltage, Temperature
DRC	Design Rule Check
LVS	Layout vs Schematic
IDE	Integrated Development Environment
RTL	Register Transfer Level
DSP	Digital Signal Processor





# Contents

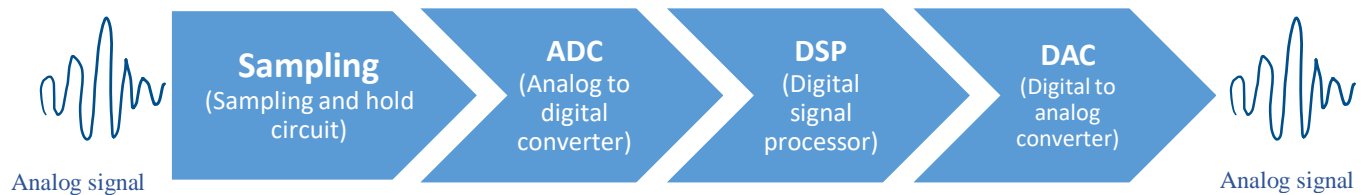
<b>Abstract</b> .....	<b>vi</b>
<b>Acronyms and Abbreviation</b> .....	<b>viii</b>
<b>Contents</b> .....	<b>x</b>
<b>1. ADCs For Biosensor Applications</b> .....	<b>1</b>
1.1. ANALOG TO DIGITAL CONVERSION.....	1
1.1.1 Sample and Hold.....	1
1.1.2 Analog to Digital conversion.....	2
1.1.3 Digital Signal Processor (DSP) .....	2
1.1.4 Digital to Analog Conversion.....	3
1.2. SAR ADC ARCHITECTURE.....	3
1.3. CAPACITIVE SPLIT – ARRAY DAC ARCHITECTURE.....	4
1.4. SAR ADC BLOCK DIAGRAM.....	7
1.5. BLOCK DIAGRAM FUNCTIONAL DESCRIPTION .....	7
<b>2. BandGap Reference Voltage</b> .....	<b>9</b>
2.1. INTRODUCTION .....	9
2.2. BGR VOLTAGE TOPOLOGIES .....	10
2.2.1 Topologies Optimization.....	13
2.3. BGR PVT ANALYSIS .....	15
2.3.1 PVT Corner Implementation .....	16
2.3.2 PVT Corner Results .....	17
2.4. BGR PHYSICAL DESIGN .....	18
2.4.1 Layout and Verification .....	18
2.4.2 Design Rule Checking (DRC).....	21
2.4.3 Layout vs Schematic (LVS) .....	22
<b>3. Serial Peripheral Interface Module</b> .....	<b>23</b>
3.1. INTRODUCTION .....	23
3.2. SPI MODULE DEVELOPMENT.....	23
3.2.1 SPI Module.....	24
3.2.2 PISO Master .....	25
3.2.3 SPI Counter .....	25
3.3. SPI MODELSIMS SIMULATION RESULTS .....	26
3.4. SPI MODULE INTEGRATION WITH SAR.....	28
<b>4. ADC Integration</b> .....	<b>29</b>
4.1. LOGIC SYNTHESIS .....	29
4.2. PHYSICAL SYNTHESIS.....	34
<b>5. Conclusions</b> .....	<b>38</b>
<b>6. Appendix A</b> .....	<b>39</b>
6.1. LUP.6 40	
<b>7. Bibliography</b> .....	<b>43</b>

# 1. ADCs For Biosensor Applications

## 1.1. Analog to Digital Conversion

Analog to Digital converters (ADCs) translate analog signals, which are characteristic of most phenomena in the world, into a digital signal, to be processed in digital systems (Floyd T. L., 2014). Since the “real world” is analog and processing is digital, data converters are used in electronic circuits at the interface between analog and digital world, and play a fundamental role in most of the applications, such as industrial, telecommunications, automotive, medical, etc.

The basic operation of an ADC in signal processing flux can be explained in four steps (Fig. 1.1): sampling, analog to digital conversion, digital signal processing, and digital to analog conversion (Floyd T. L., 2006). These steps are shown in the next figure:



**Figure 1.1** : Analog to digital conversion process

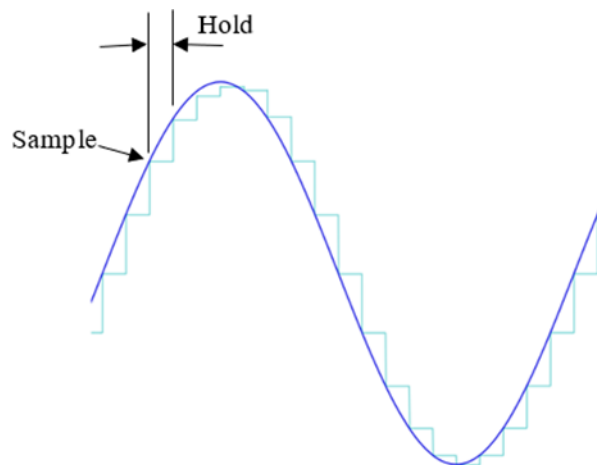
### 1.1.1 Sample and Hold

Sampling is the process to take the value of the input signal sufficient times to have enough information of the input signal. This process converts an analog signal into a series of impulses, each one representing the amplitude of the signal at an instant in time. The more samples are taken, the more accurate is the waveform.

The holding operation ensures the sampled value must be held constant for an instant defined until the next sample is taken. This is necessary for the ADC to have time to process the sampled value. This sample and hold process results in a stairstep waveform that approximates the analog input signal (Fig. 1.2).

### 1.1.2 Analog to Digital conversion

It is the process of converting the output of the sample and hold circuit into a series of binary codes that represent the amplitude of the analog input signal at each of the sample times. The ADC makes the codification in the time between sample pulses, or the time that sample and hold circuit is holding the sampled value. This process is called quantization.



**Figure 1.2** Analog signal and its stairstep approximation.

### 1.1.3 Digital Signal Processor (DSP)

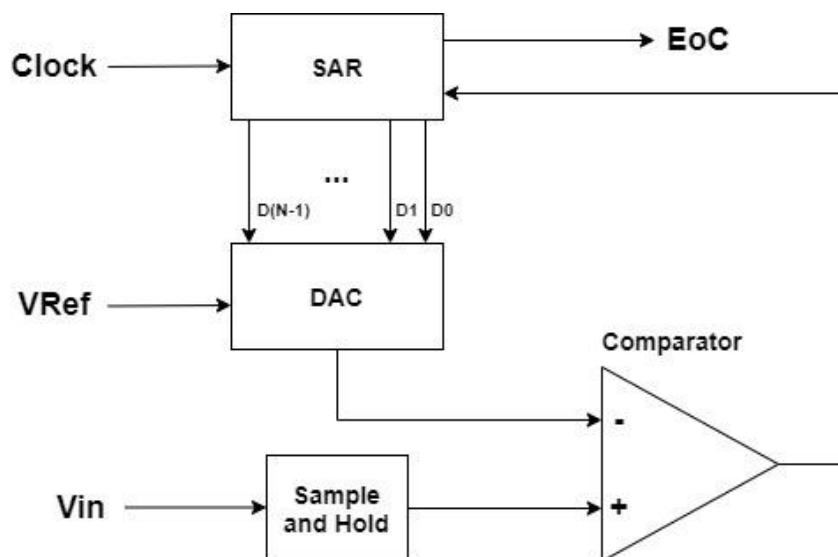
The Digital Signal Processor (DSP) consists of a specialized microprocessor chip which architecture has been optimized for processing the digital signal received from the previous step. The goal of the DSP is to process data in real-time.

### 1.1.4 Digital to Analog Conversion

It is the process of transforming data or the result of the digital processing, back to an analog signal to send it back to the “real world”. Most of the DACs perform two basic functions: conversion of the digital input into an equivalent analog signal and reconstruction of the signal.

## 1.2. SAR ADC Architecture

There are many conversion techniques to change analog signals to digital. In our proposal, a SAR ADC technique is selected. Fig. 1.3 shows the main components of a SAR ADC. The operation principle is as follows: the circuit samples an input signal and compares it to several voltages that are generated by a Digital to Analog Converter (DAC). The Successive Approximation Register (SAR) controls the voltage of the DAC and saves the results of the comparison.



**Figure 1.3** Diagram of a SAR ADC.

In Figure 1.3, the block diagram of a basic SAR ADC is shown. The main modules of this architecture are a sample and hold circuit, a comparator, a DAC, and a SAR.

As stated before, the SAR controls the voltage that the DAC outputs. It starts by assigning a 1 to the most significant bit (MSB). This is done because that is the halfway value of the voltage that the DAC works with. The next step is to compare the DAC voltage with the sampled input voltage. If the analog input is higher than the voltage of the DAC, the Comparator has an output of 1 (or the equivalent analog voltage) that the SAR saves, and then the SAR will assign a 1 to the next significant bit, repeating the process. But if the sampled voltage is smaller than the original DAC's voltage, then the SAR saves a 0, turns the MSB to 0, and assigns a 1 to the next significant bit, starting the cycle again.

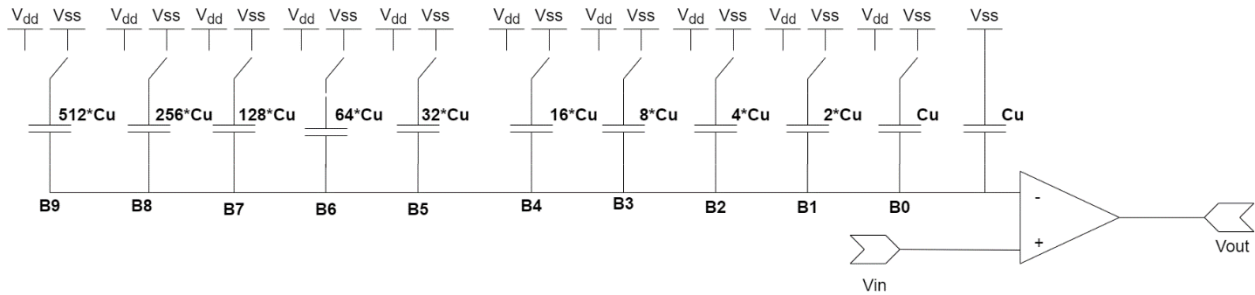
A SAR ADC converts a bit in each cycle, so depending on the resolution of the ADC is the minimum number of cycles needed to finish an analog conversion.

### **1.3. Capacitive Split – Array DAC Architecture**

As explained before, a SAR ADC needs a Digital to Analog Converter (DAC) to generate the voltages that will be compared against the input voltage. Since this project wants to be used in a Biosensor application, the proposal that we choose must consume the least amount of power and must be as small as possible. For this reason, we choose Capacitive Split-Array Architecture for our DAC.

There are many architectures for implementing a DAC and most of them use either resistors or capacitors to generate voltage as their main component. The difference between using one or the other component is that resistors are components that are constantly consuming power since they only work in a state, meanwhile, capacitors consume power depending on the state the capacitor is in. When the capacitor is working then it consumes power, but if it is not working then it won't. This behavior is what pushes us to use an architecture based on capacitors.

The most commonly used capacitor architecture is called Charge-Scaling Capacitors, which can be seen in Figure 1.4.

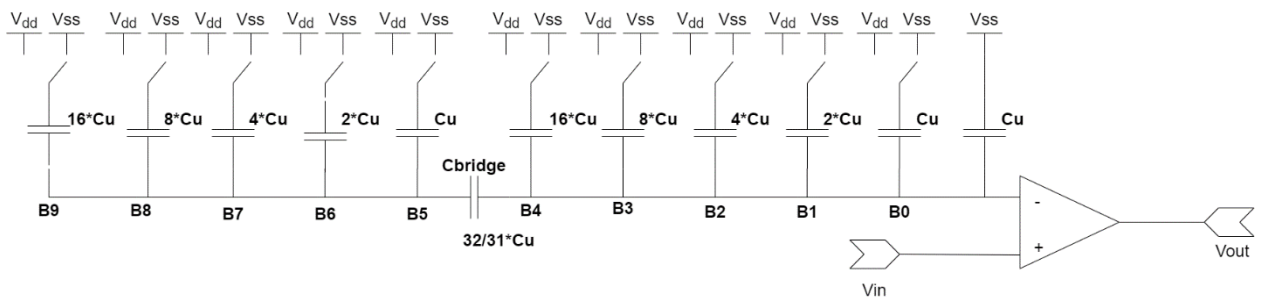


**Figure 1.4** Schematic diagram of Charge-Scaling Capacitor topology for a 10-bit resolution DAC (Baker, 2010).

The number of capacitors of Charge-Scaling Capacitor DAC defines the bits in the DAC. Figure 1.4, shows the schematic of a 10 bits DAC. Each of the capacitors represents a bit, except for the last one that is used to provide the correct divisor factor in the inverting node. Each capacitor will switch between VSS and VDD, depending on what the SAR will assign to the control signals. This process will start from the MSB to the LSB.

The main problem with this architecture remains in the capacitor value of the MSB; where  $n$  is the DAC resolution. In the example of Figure 1.4, the MSB capacitor is 512 times bigger than the smallest one, i.e., the unity capacitance  $C_u$ . Having large capacitors is not desirable because they are area-consuming, and this is an important tradeoff when designing a chip. The larger they are, the more expensive they are, and they consume more power.

To reduce the power consumption of DAC, the Capacitive Split-Array topology is utilized. It is shown in Figure 1.5.



**Figure 1.5** Schematic diagram of Capacitive Split-Array Architecture (Baker, 2010).

The architecture called “Capacitive Split-Array” reduces power consumption by dividing the capacitors into two arrays, one for the least significant bits (LSB) b0 to b4 and the other one for the most significant bits (MSB) b5 to b9. Both arrays have capacitors of the same magnitude, and this works because of the bridge capacitor between both arrays ( $C_{bridge}$ ). The only difference between the array is that the one for the LSB also has an extra capacitor to provide the correct divisor factor in the inverting node. The value of the bridge capacitor can be calculated using Equation 1:

$$C_{bridge} = \frac{\text{Sum of LSB Capacitance}}{\text{Sum of MSB Capacitance}} * C_u$$

Eq 1: Equation for calculation the bridge capacitor in Capacitive Split-Array architecture.

The DAC Capacitive Split-Array has all the benefits of the DAC Charge-Scaling Capacitors but uses smaller capacitors. The only disadvantage is that by adding the  $C_{bridge}$ , the system will not have a linear behavior which could complicate fixing mistakes if the capacitor arrays are not properly balanced (Yan Zhu, 2014).



## 1.4. SAR ADC Block Diagram

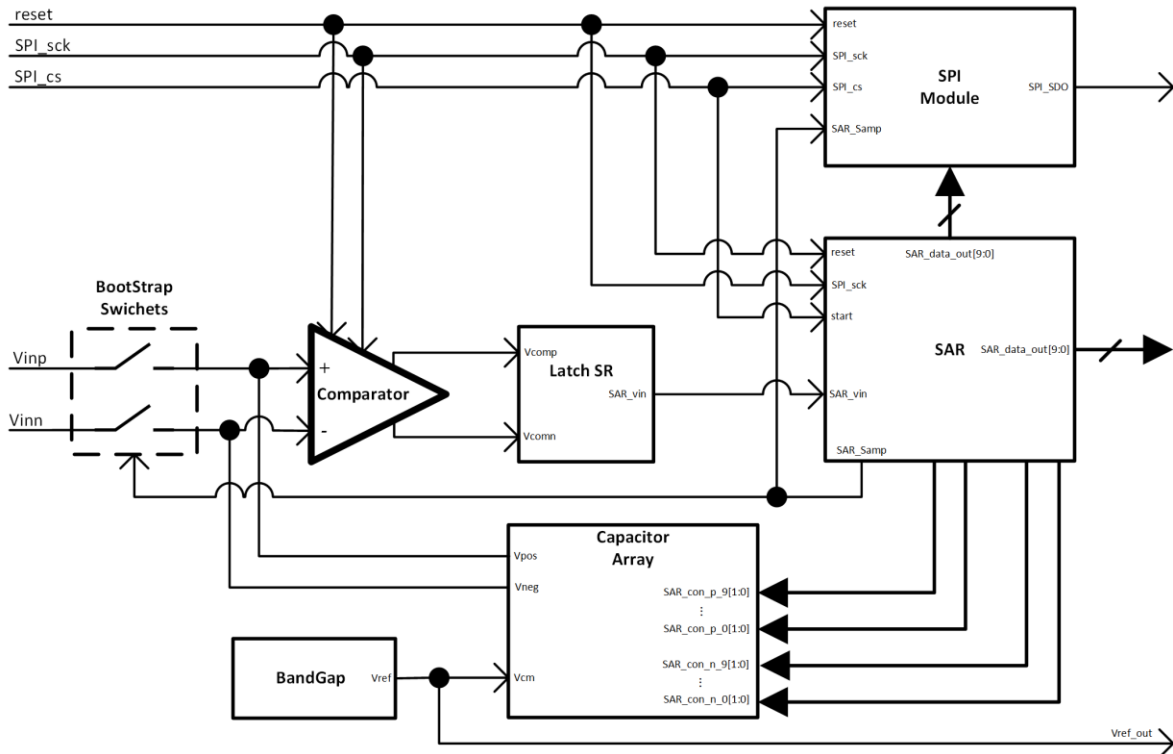


Figure 1.5 SAR ADC Block Diagram.

## 1.5. Block Diagram Functional Description

The SAR ADC is made by different functional blocks, shown in Fig 1.5, which are described in the following sentences:

- **Serial Peripheral Interface (SPI) module:** the SPI module is responsible to send the conversion result from the SAR module to an external signal in a serial format.
- **BandGap Reference module:** This module supplies a reference voltage to the capacitors array. This module is supplied by 1.8 V and provides a stable output of  $900\text{ mV} \pm 500\text{ }\mu\text{V}$ .
- **SAR module:** It converts the analog signal into a digital signal. The digital signal contains 10 bits, named  $SAR\_data\_out [9:0]$ , that are sent directly to the SPI module and the exterior.
- **Capacitor Array module:** This module consists on a Digital to Analog block that generates voltages to be compared against two external signals,  $V_{ip}$  and  $V_{inn}$ .

- Comparator module: It compares the two external signals,  $V_{ip}$  and  $V_{inn}$  after they have been sampled by the BootStrap Switches.
- Latch SR module: once the comparison between  $V_{ip}$  and  $V_{inn}$  has been finished, the Latch SR module converts the pair signals to a single wire.

## 2. BandGap Reference Voltage

In this section, the BandGap Reference (BGR) Voltage is described in detail. The BGR is designed in TSMC 0.18  $\mu\text{m}$  CMOS technology with 1.8 V supply voltage. The BGR works with an input voltage of 1.8 V as supply and an output voltage or  $V_{\text{ref}} = 900 \text{ mV}$ , from  $-40 \text{ }^\circ\text{C}$  to  $85 \text{ }^\circ\text{C}$ , across PVT corners with a variation of  $\pm 500 \text{ } \mu\text{V}$ .

### 2.1. Introduction

An ideal voltage reference is a circuit used to generate a fixed voltage,  $V_{\text{ref}}$ , that is independent of the power supply voltage  $V_{\text{DD}}$  (where  $V_{\text{ref}} < V_{\text{DD}}$ ), the temperature, and the process variations. In other words, the ideal reference voltage is independent of PVT variations. In some cases, we want to design a reference that varies with temperature. When  $V_{\text{ref}}$  increases with temperature, as shown in Fig. 2.1a, we say that the reference voltage is proportional to the absolute temperature (PTAT), meanwhile, if the reference voltage decreases when the temperature increases, as shown in Fig. 2.1b, we say that the reference is complementary to the absolute temperature (CTAT) (Baker, 2010). Then the BGR circuit works on the compensation of the two PTAT and CTAT curves.

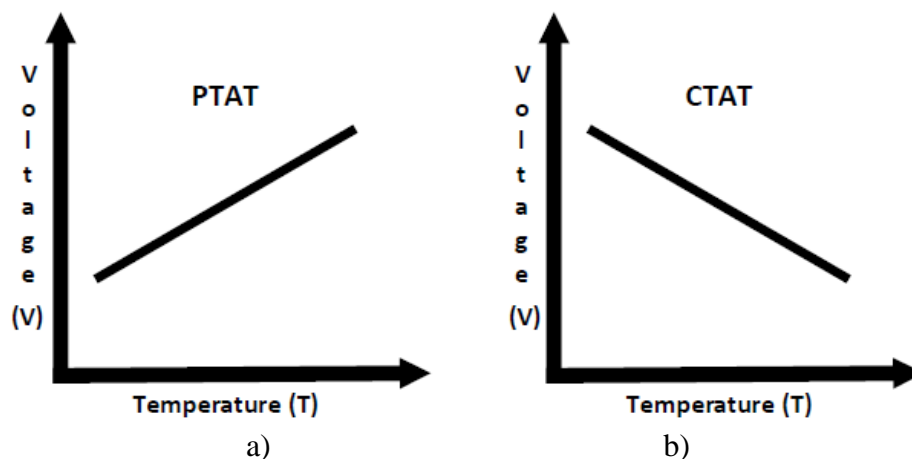
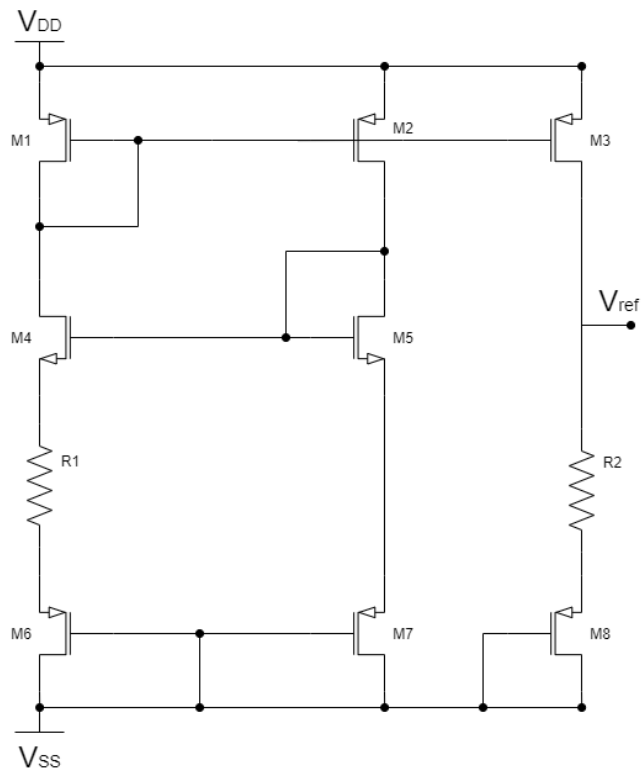


Figure 2.1. a) PTAT and b) CTAT voltages references.

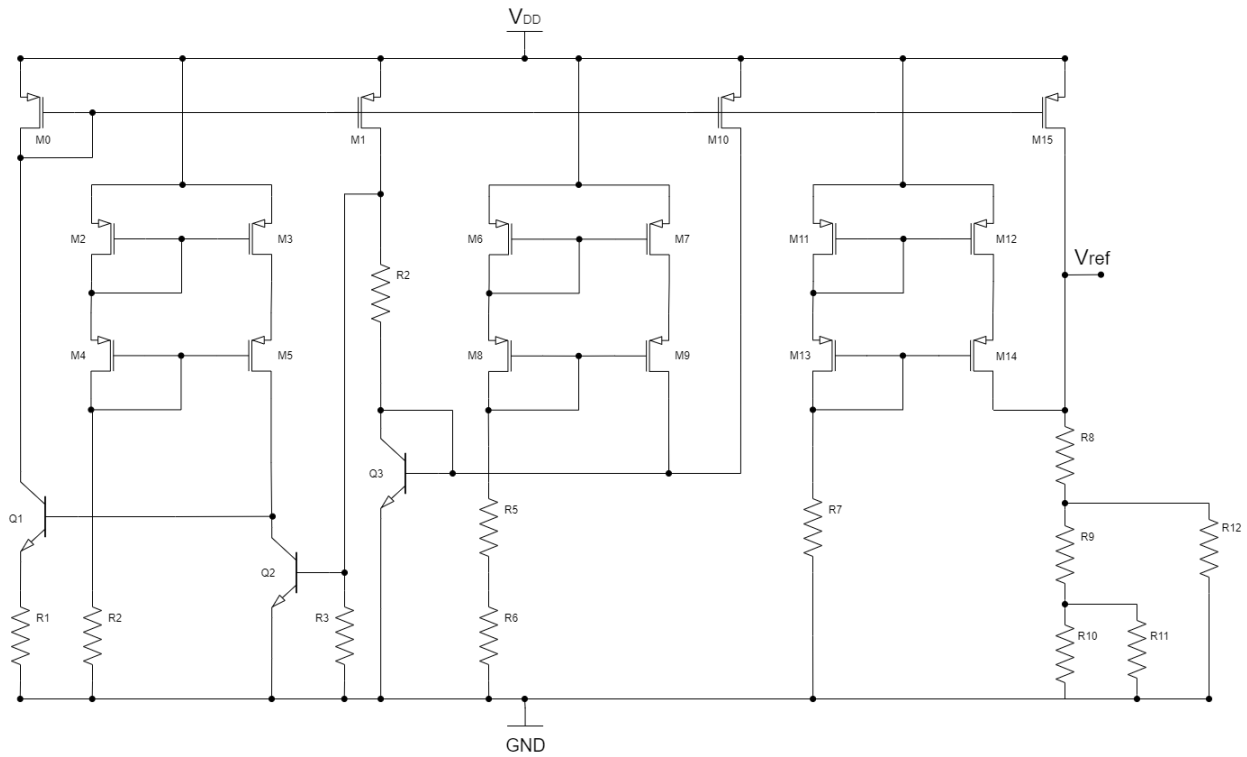
## 2.2. BGR Voltage Topologies

In this work, two BGR topologies were analyzed: a full CMOS transistor topology (Fig. 2.2) (Ribeiro, Gama, Costa, Neves, & Horta) and a hybrid CMOS – BJT transistor topology which was designed by (Martinez Guerrero, 2021) from the conceptual approach reported in (Naganadhan, 2019) (Fig. 2.3).

Both topologies were calculated to obtain a voltage reference  $V_{ref} = 900$  mV, and are based on current mirrors to deal with the current consumption and to equilibrate the temperature variation. To validate the correct functionality, each circuit was calculated, simulated, and optimized.



**Figure 2.2** Full CMOS transistor topology.



**Figure 2.3** Hybrid CMOS – BJT transistor topology.

The first simulation draft was performed to observe the effect of the temperature on  $V_{ref}$ . The simulation was performed on the Virtuoso environment using the Spectre simulator and the values shown in Table 2.1 and Table 2.4.

A DC analysis was done, taking the temperature as a sweep variable from  $-40\text{ }^{\circ}\text{C}$  to  $120\text{ }^{\circ}\text{C}$ . Corresponding responses for both circuits are shown in Table 2.2, Fig. 2.4, and 2.5.

Topology	M1, M2, M6, M7 [ $\mu\text{m}$ ]	M4, M5 [ $\mu\text{m}$ ]	M3 [ $\mu\text{m}$ ]	M8 [ $\mu\text{m}$ ]	R1 [ $\text{k}\Omega$ ]	R2 [ $\text{k}\Omega$ ]
FULL CMOS	28.8	2	180	2	25	22

**Table 2.1** Values to be testes on Virtuoso for the Full CMOS Technology.

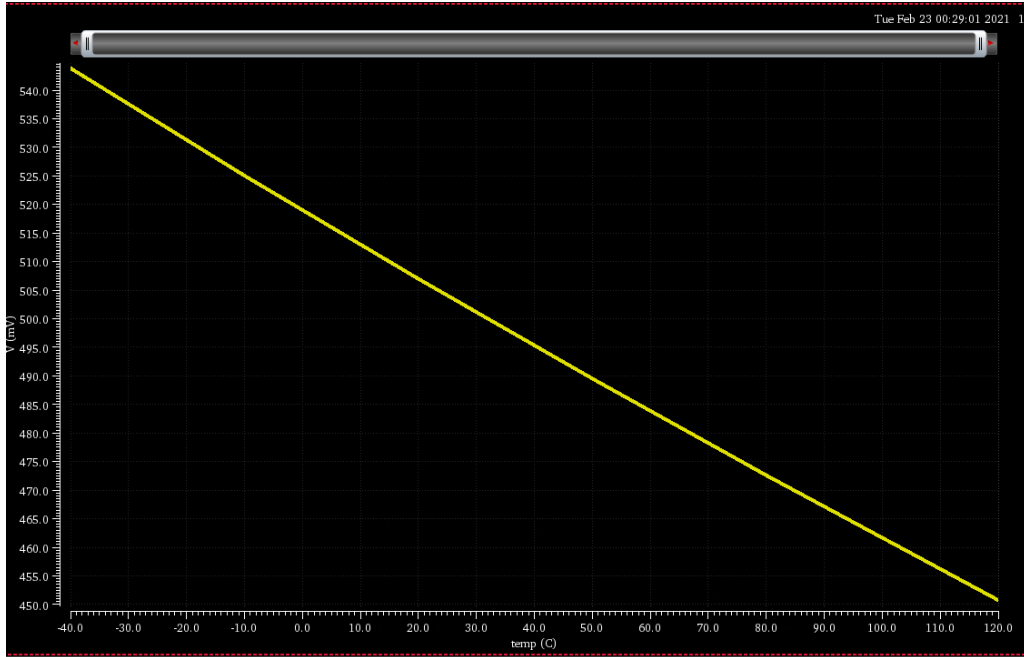


Figure 2.4  $V_{ref}$  simulation response. Full CMOS topology.

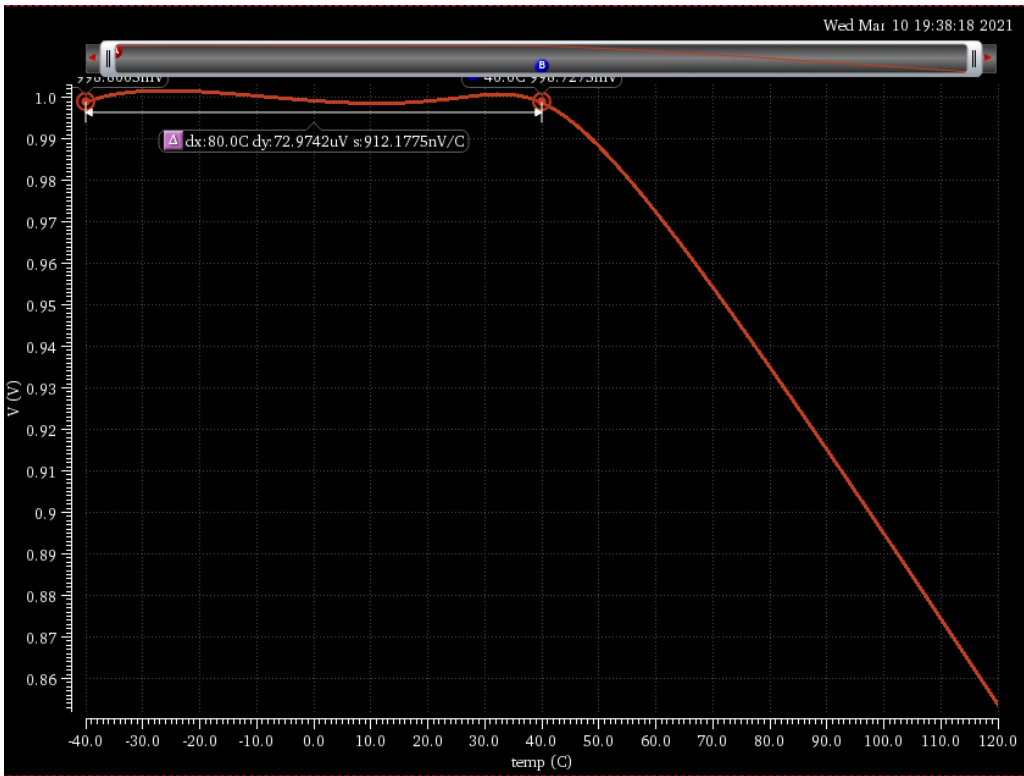


Figure 2.5  $V_{ref}$  simulation response. CMOS - BJT topology.

Topology	V <sub>ref</sub> Max [mV]	V <sub>ref</sub> Min [mV]	Min Temp [ °C ]	Max Temp [ °C ]	Voltage Variation [mV]
First	540	450	-40	120	90
Second	990	860	-40	120	130

**Table 2.2** V<sub>ref</sub> results after first simulation draft on Virtuoso.

### 2.2.1 Topologies Optimization

After the simulation draft, the full CMOS topology (first one) was optimized. Even with the optimization process, the V<sub>ref</sub> response exhibits a big variation between V<sub>ref</sub> MAX and V<sub>ref</sub> MIN. For this reason, the full CMOS topology was no longer used.

Model	V <sub>ref</sub> Max [mV]	V <sub>ref</sub> Min [mV]	Voltage at Room [mV]	Min Temp [ °C ]	Max Temp [ °C ]	Voltage Variation [mV]
OPT - 1	1380	1060	1190	-40	120	320
OPT - 2	830	755	776	-40	120	75
OPT - 3	1090	910	975	-40	120	180
OPT - 4	1070	900	965	-40	120	170
OPT - 5	1120	920	995	-40	120	200
OPT - 6	1170	940	1020	-40	120	770
OPT - 7	701	687	688	-40	120	4
OPT - 8	670	661	662	-40	120	9

**Table 2.3** V<sub>ref</sub> results after optimization were applied on Full CMOS topology.

As mentioned previously, the main purpose of this BGR is to be part of an ADC for biosensors applications, for this reason, the stability of V<sub>ref</sub> is critical. A maximum variation on V<sub>ref</sub> must be under  $\pm 500 \mu\text{V}$ . Simulation results have shown that hybrid CMOS – BJT topology performs better than the Full CMOS topology, showing a voltage variation of  $\pm 72 \mu\text{V}$ , during the early temperature range, from -40 °C to 40 °C. Then, optimization techniques were applied to the CMOS – BJT topology to obtain the desired V<sub>ref</sub>.

To optimize the hybrid CMOS – BJT topology, a surrogate-based optimization technique was applied. First, a set of training points was obtained to develop a surrogate model. Five surrogate model techniques were implemented: polynomial surrogate models, response surface methodology, support vector machines, Kriging, and generalized regression neural networks.

By using the best-performed surrogate model, a direct optimization technique (such as nelder mead, pattern search, and genetic algorithms) was applied to obtain the desired  $V_{ref}$  response. The optimization techniques used on the CMOS – BJT topology was the one called surrogate based optimization (Chávez-Hurtado, Rayas-Sánchez, & Brito-Brito, 2016).

After each surrogate-based optimization process, the final value was used as a seed value to generate a new surrogate model and restart the optimization process. This was done consecutively until the specification design was achieved. These groups of values were called "Center" and are shown in Table 2.4 (Chávez-Hurtado & Rayas-Sánchez, 2016).

Center	M0, M1, M10, M15 [μm]	M2, M3, M4, M5, M6, M7, M8, M9 [μm]	Q1, Q2, Q3 [μm]	R1 [kΩ]	R2 [kΩ]	R3 [kΩ]	R4 [kΩ]	R5, R6 [kΩ]	R7 [kΩ]	R8, R9, R10, R11 [kΩ]	R11 [kΩ]	R12 [kΩ]
Center 1	2.866	10.9847	10	13.717 9	59.58	12.014 4	43.085 8	156.52	54.621 6	7.8185	19.8 9	31.2 5
Center 2	3	10	X	14.5	X	13.200 2	45.148 7	X	57.015 4	8.2307	X	X
Center 3	2.85	9.5	X	13.775	X	12.540 1	42.891 2	X	54.193 1	7.8192	X	X
Center 4	2.8785	9.595	X	13.912 7	X	12.665 5	43.320 1	X	54.735 0	7.8974	X	X

**Table 2.4** Transistors and resistors values taken as center after simulation techniques.

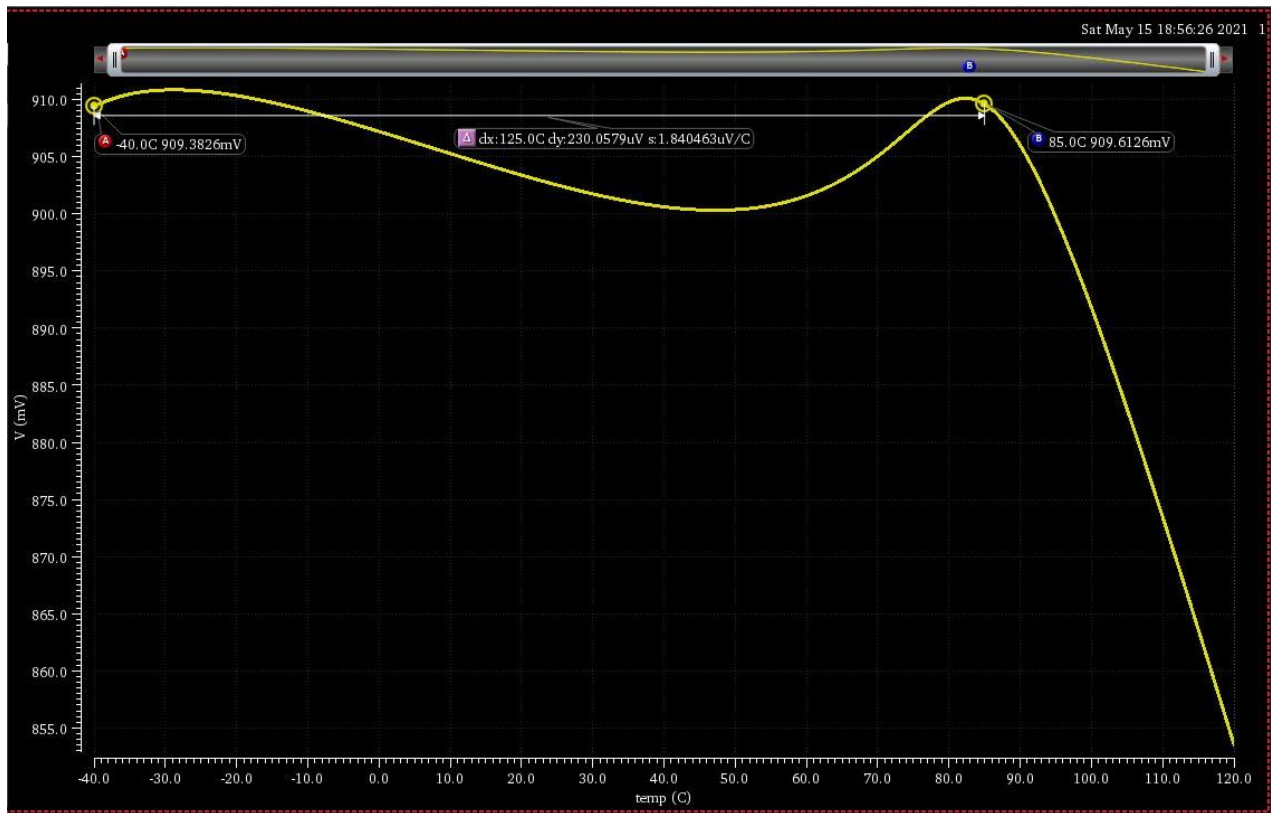
After the first optimization process, it was found that components Q1, Q2, Q3, R2, R5, R6, R11, and R12 are in their ideal value, then these components were no longer included in the following optimization process.

Due to the long temperature range, from  $-40\text{ }^{\circ}\text{C}$  to  $120\text{ }^{\circ}\text{C}$ , it was not possible to find an optimal solution, then, the maximum temperature was reduced to  $85\text{ }^{\circ}\text{C}$ , since this new temperature range still complies with the biosensor applications. The optimal component values, shown in Table 2.5, correspond to the best response on  $V_{ref}$  for this topology.

Center	M0, M1, M10, M15 [μm]	M2, M3, M4, M5, M6, M7, M8, M9 [μm]	Q1, Q2, Q3 [μm]	R1 [kΩ]	R2 [kΩ]	R3 [kΩ]	R4 [kΩ]	R5, R6 [kΩ]	R7 [kΩ]	R8, R9, R10, R11 [kΩ]	R11 [kΩ]	R12 [kΩ]
BGR	2.90	9.69	10	14.05	59.58	12.79	43.75	126.52	55.28	7.97	79.89	31.25

**Table 2.5** CMOS – BJT topology values for the best respond on  $V_{ref}$ .





**Figure 2.6** Graph from  $V_{ref}$  using the finals BGR values.

Model	$V_{ref Max}$ [mV]	$V_{ref Min}$ [mV]	Voltage at Room [mV]	Min Temp [ °C ]	Max Temp [ °C ]	Voltage Variation [ $\mu$ V]
BGR	909.6126	909.3826	230	-40	85	230.0579

**Table 2.6**  $V_{ref}$  respond using the values on BGR model.

### 2.3. BGR PVT Analysis

In order to guarantee that our chip properly works in all the possible operable conditions, we must simulate it at different values of process, voltage, and temperature. These conditions are called PVT corners (TemplatesYard, 2020).

These variations will effect operational conditions like threshold voltage since PVT corners affect parameters like doping concentration, surface potential, channel length, oxide thickness, temperature, source-to-body voltage, implant impurities, among others (TemplatesYard, 2020).

### 2.3.1 PVT Corner Implementation

For the PVT corner analysis, the supply voltage was not modified due to the complete BGR circuit was calculated to give a  $V_{ref}$  of  $900 \text{ mV} \pm 500 \text{ } \mu\text{V}$  using a supply voltage of  $1.8 \text{ V}$ . For that reason the PVT corner analysis includes only variations on process and temperature.

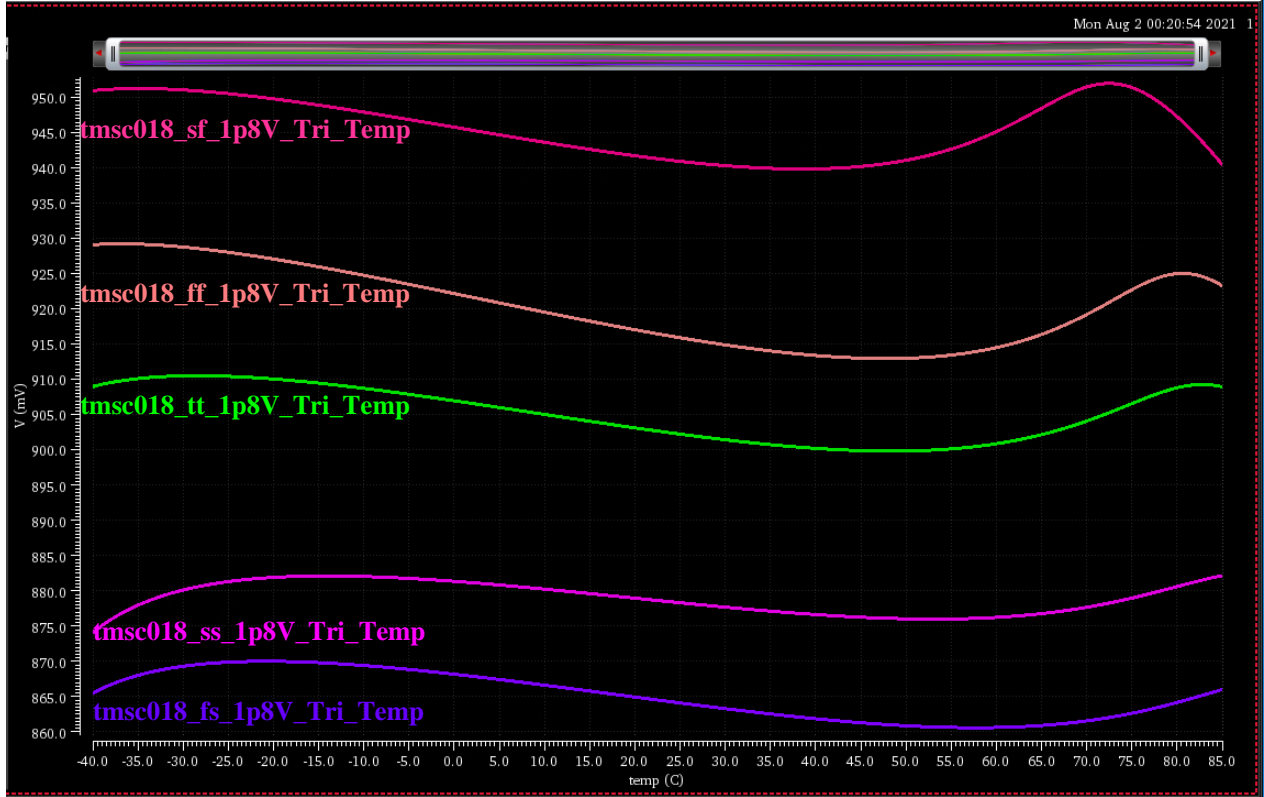
Corner Name	Process	Temperature [ °C ]	Voltage [ V ]
tmsc018_tt_1p8V_Tri_Temp	tt	-40	1.8
tmsc018_tt_1p8V_Tri_Temp	tt	27	1.8
tmsc018_tt_1p8V_Tri_Temp	tt	85	1.8
tmsc018_ff_1p8V_Tri_Temp	ff	-40	1.8
tmsc018_ff_1p8V_Tri_Temp	ff	27	1.8
tmsc018_ff_1p8V_Tri_Temp	ff	85	1.8
tmsc018_ss_1p8V_Tri_Temp	ss	-40	1.8
tmsc018_ss_1p8V_Tri_Temp	ss	27	1.8
tmsc018_ss_1p8V_Tri_Temp	ss	85	1.8

**Table 2.7** PVT Corner performed on the CMOS – BJT topology for tt, ff, and ss process.

Corner Name	Process	Temperature [ °C ]	Voltage [ V ]
tmsc018_sf_1p8V_Tri_Temp	sf	-40	1.8
tmsc018_sf_1p8V_Tri_Temp	sf	27	1.8
tmsc018_sf_1p8V_Tri_Temp	sf	85	1.8
tmsc018_fs_1p8V_Tri_Temp	fs	-40	1.8
tmsc018_fs_1p8V_Tri_Temp	fs	27	1.8
tmsc018_fs_1p8V_Tri_Temp	fs	85	1.8

**Table 2.8** PVT Corner performed on the CMOS – BJT topology for sf, and fs process.

### 2.3.2 PVT Corner Results



**Figure 2.7**  $V_{ref}$  variation under sf, ff, tt, ss and fs PVT corners and -40 °C – 85 °C temperature range.

As shown in Fig 2.7, the temperature variation does not affect the circuit performance, it can be seen that the three temperature plots are overlapped. However, the model variation has a huge effect on the  $V_{ref}$  voltage, where models such as slow-fast and fast-slow correspond to the worst performance case.

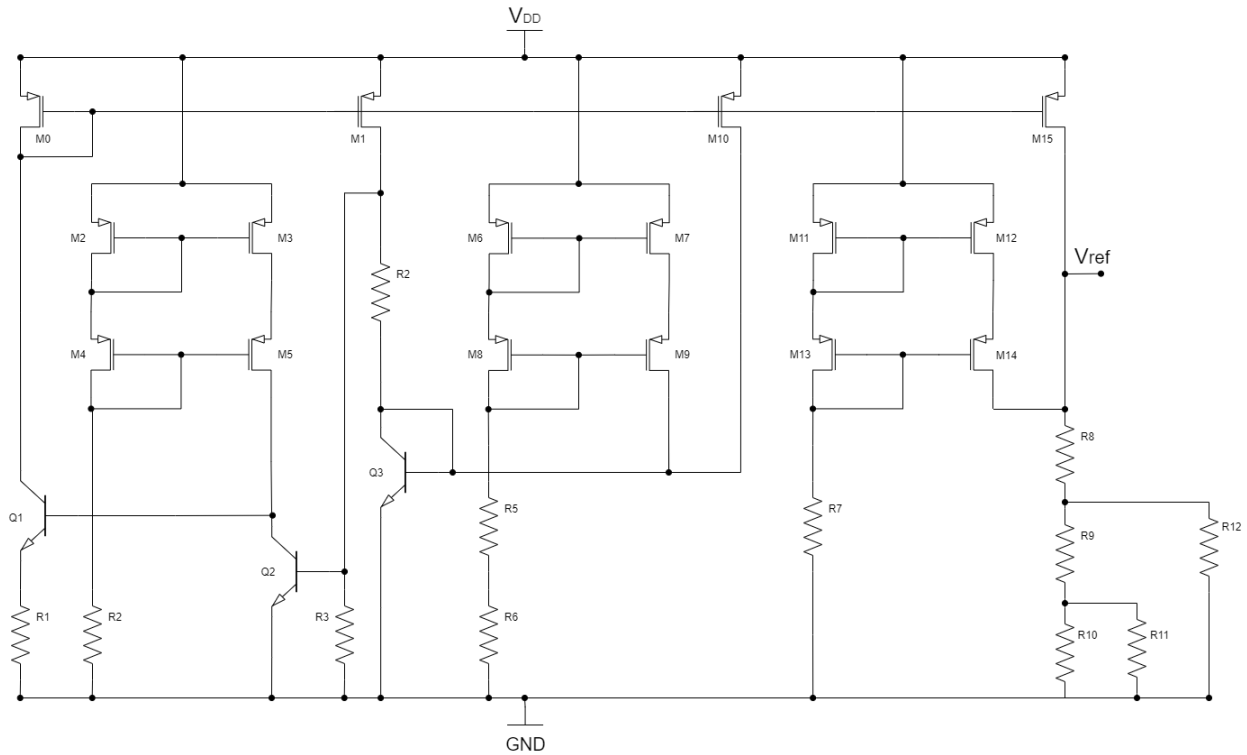
The voltage variation between models slow-fast and fast-slow is 81.7735 mV. The fast-fast model is the closest one to the typical case, exhibiting a voltage variation of 18.2763 mV.

Taking into consideration the PVT results, and how they affect the  $V_{ref}$  voltage on the BGR, only the typical model will be taken into consideration for the test inside the SAR ADC system.

## 2.4. BGR Physical Design

### 2.4.1 Layout and Verification

The Bandgap Layout was developed using Virtuoso Tools. The Layout XL imports the cell view of each component from the Bandgap schematic shown in Fig. 2.8.

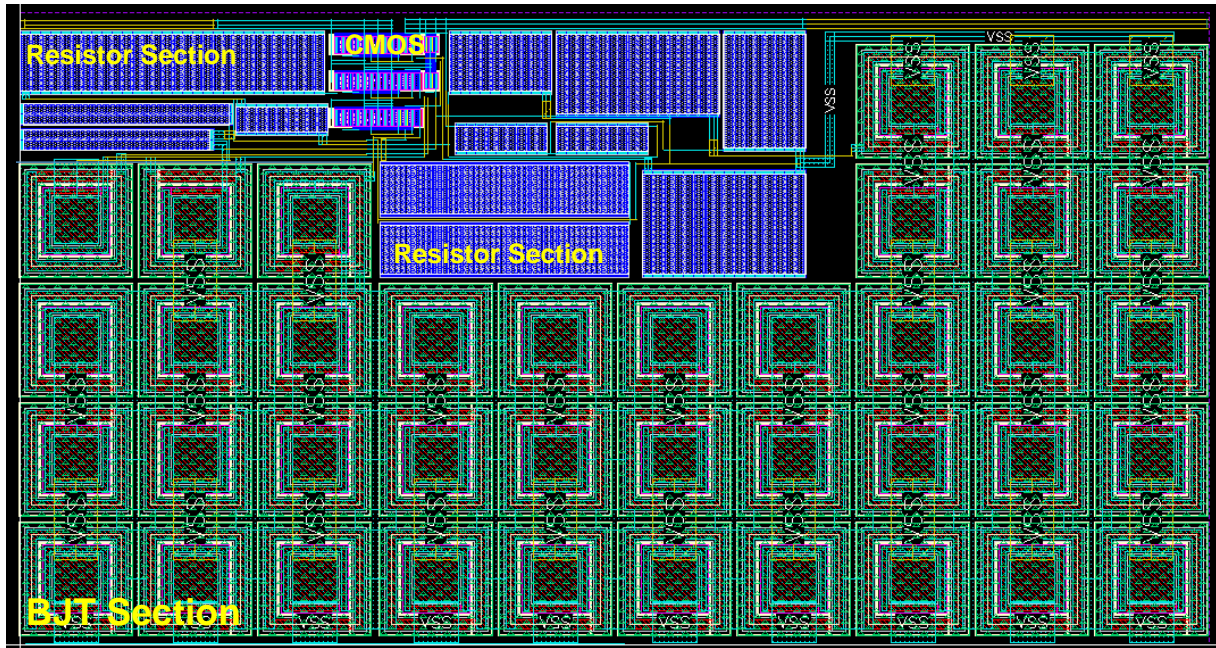


**Figure 2.8** Bandgap reference schematic.

The placement of the components and routing was made manually. The height for the BGR were chosen considering the height of a standard cell in 0.18 $\mu\text{m}$  CMOS technology, shown in Table 2.9. The BGR has a height of 140.48  $\mu\text{m}$  which is 32 times a standard cell height.

	Height [ $\mu\text{m}$ ]	Base [ $\mu\text{m}$ ]	Times
Standard Cell	4.39	N/A	1
BGR	140.48	264.18	32

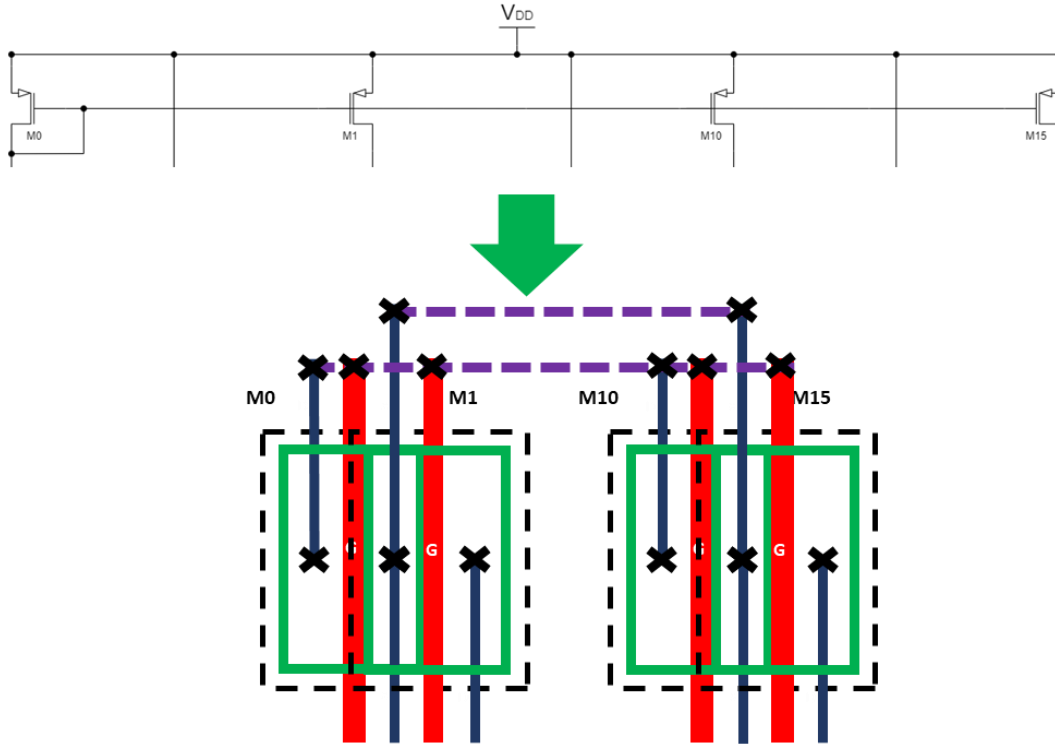
**Table 2.9** BandGap Reference dimension and comparation with standard cell.



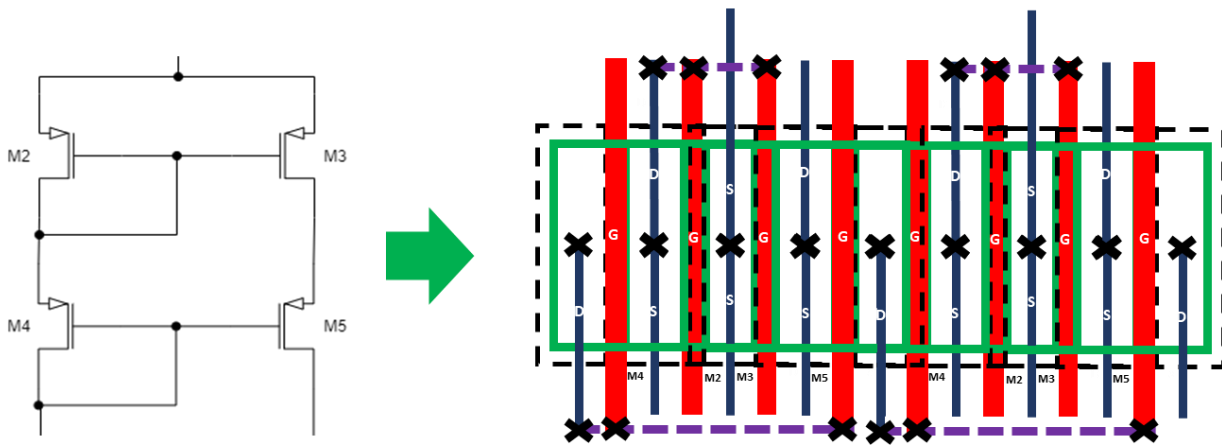
**Figure 2.9** BandGap layout after placement and routing.

For routing components, metals 1, 2, and 3 were used. The pins were made using metal 3. As shown in Fig. 2.9, a big amount of the area corresponds to BJT transistors. Resistors were implemented using high resistance polysilicon resistor (nrhpoly) available in the TSMC18 process, and a finger technique was applied to them to reduce the final length.

Before performing the layout of the BGR circuit, a stick diagram was developed to visualize the placement of CMOS transistors. In order to improve matching, the CMOS transistors were divided on fingers and arrayed using interdigitated techniques, as shown in Fig. 2.10 and 2.11.



**Figure 2.10** Stick diagram for the first part of CMOS transistors.



**Figure 2.11** Stick diagram for the second part of CMOS transistors.

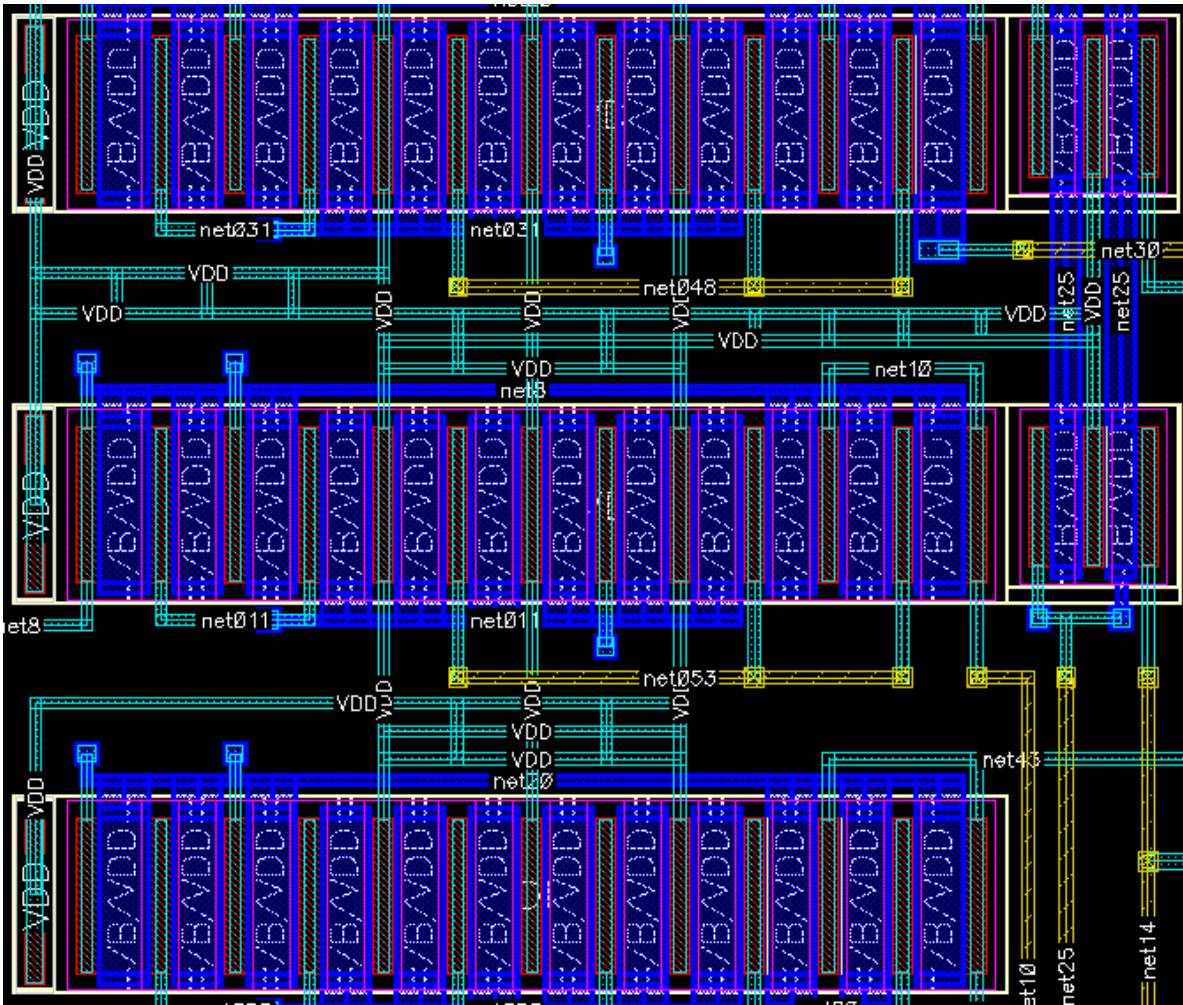


Figure 2.12 CMOS Section routed and using interdigitated techniques on transistors.

### 2.4.2 Design Rule Checking (DRC)

The layout must be drawn according to strict design rules. After the routing and placement have been finished, an automatic program will check each and every polygon in the BGR against the design rules. This process is called Design Rule Checking (DRC) and MUST be done for the BGR layout to ensure it will function properly when fabricated (AMSaC Lab Group, 2021). All the DRC errors were corrected during the layout design. One of these errors found during DRC checking is described in Appendix A. This is reported because it is not a common error in the layout of analog cells and its documentation can serve other designers as a guide for correcting this error.

### 2.4.3 Layout vs Schematic (LVS)

The next step in debugging the layout is to compare the netlist extracted from the layout with the schematic to ensure the BGR layout is an identical match to the cell schematic (AMSAc Lab Group, 2021). Just a few errors were reported by the LVS tool but were corrected without problems. Figure 2.13 shows the final LVS report in which we can appreciate that the BGR layout is LVS error free.

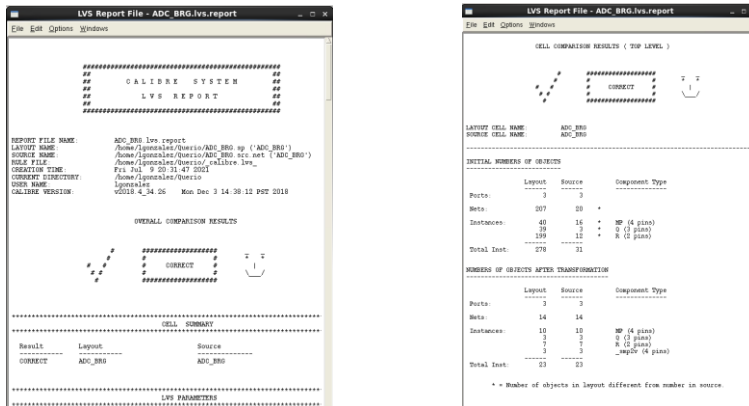


Figure 2.13 LVS report once the errors were corrected.



## **3. Serial Peripheral Interface Module**

The serial peripheral interface (SPI) module was designed by describing the hardware on Verilog language using the Quartus IDE by Intel. Simulations of the description were performed on ModelSim. This SPI module is part of a low-power ADC for Biosensor applications.

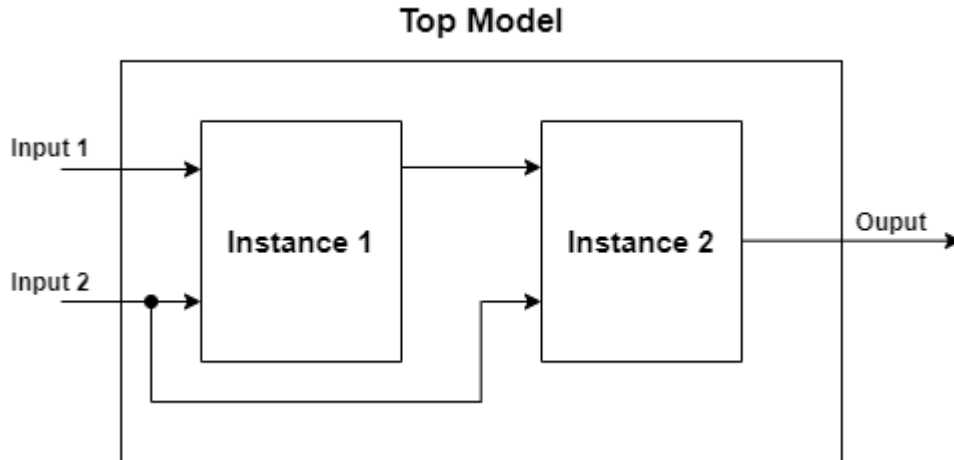
### **3.1. Introduction**

The SPI is one of the most widely used interfaces between the microcontroller and IC peripherals, such as sensors, ADCs, DACs, shift registers, SRAM, among others. The SPI module is asynchronous communication, full-duplex master-slave interface (Dhaker, 2018).

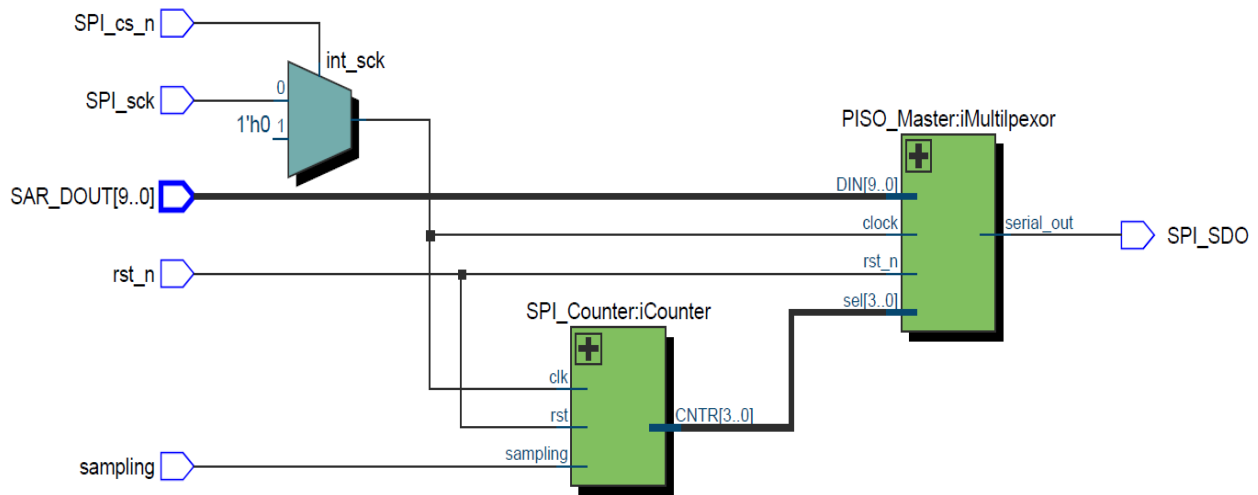
### **3.2. SPI Module Development**

The SPI module was designed using Quartus IDE. The SPI module has two instantiations inside, one called PISO Master, and a second one called SPI Counter (Fig. 3.1).

Once the code has been finished, the IDE gives a graphic representation of each instance and its connection inside a top model. This is called Register Transfer Level (RTL) Block Diagram. It is shown in Fig. 3.2.



**Figure 3.1** Graphic representation of instantiation.



**Figure 3.2** RTL Block Diagram of the SPI Module.

### 3.2.1 SPI Module

This is the main module, also called Top Model. It receives the main signal to start to work such as `SPI_cs_n`, `SPI_sck`, `rst_n`,...etc. As mentioned before this module works similar to SPI communication, therefore once the chip – select signal (CS) is in a low state, the clock signal (SCK) is available for the rest of the instances. Both signals are handled externally and should comply with SPI communication standards.

### 3.2.2 PISO Master

The PISO Master is a module that handles the data that will be sent. After the clock signal is available, the module receives data composed of 10 bits from the SAR module, through the SAR\_DOUT [9...0] signal, in a parallel format (Fig. 3.4). The PISO receives the data and allocates the most significant bit (MSB) to the serial data out (SDO) signal. This process continues until the PISO allocates the low significant bit (LSB) on the SDO signal (Fig. 3.5). Once the bits are on the SDO, they can be managed externally in a serial format.

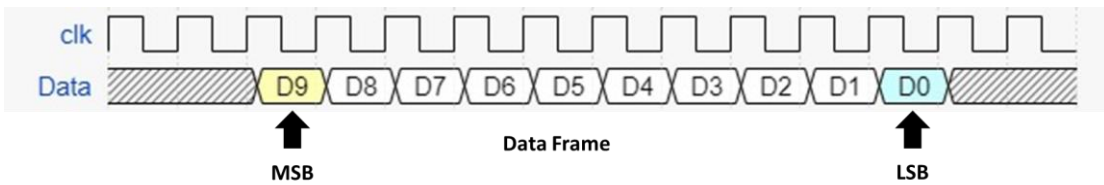


Figure 3.3 MSB and LSB in a data frame.

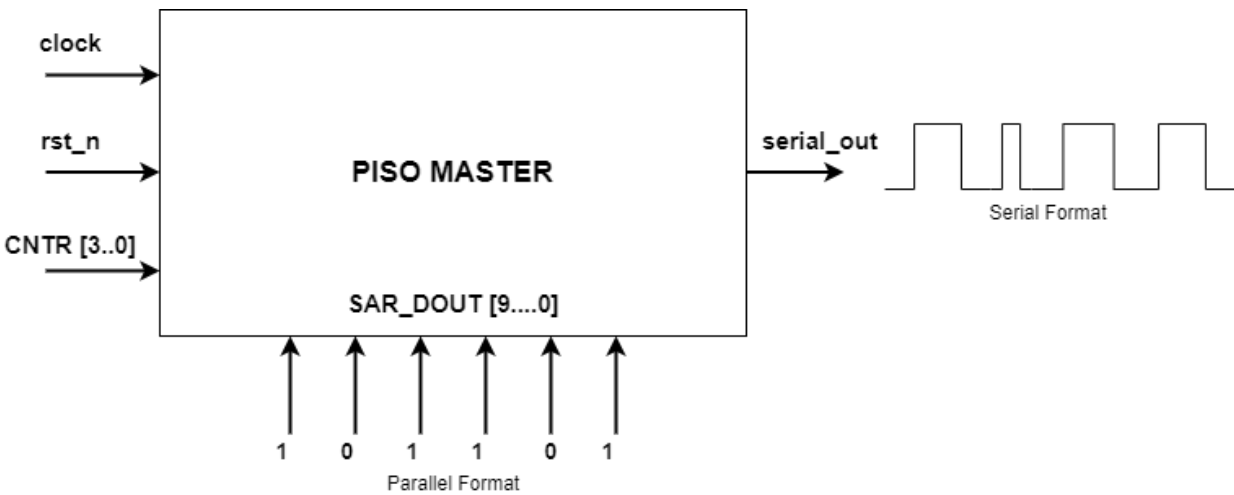
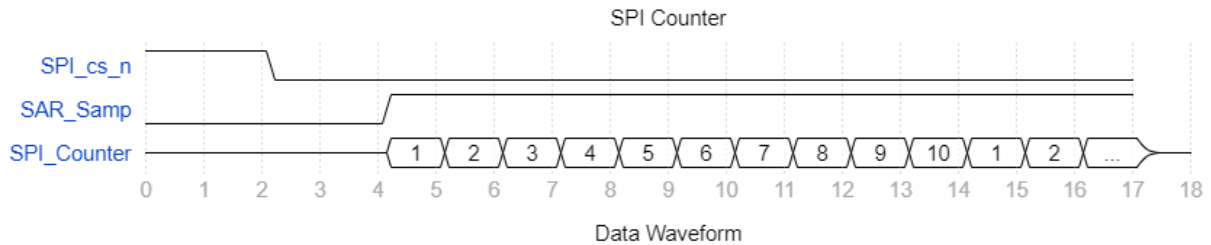


Figure 3.5 PISO Master block diagram functionality.

### 3.2.3 SPI Counter

The SPI counter counts the number of bits for the PISO Master. This module is active after the CS is in a low state and the clock signal is available. After the clock is available the module

waits until the sampling signal is in a high state (Fig. 3.6). The sampling is in a high state once the SAR has started with the conversions.



**Figure 3.6** SPI Counter data waveform description.

### 3.3. SPI ModelSims Simulation Results

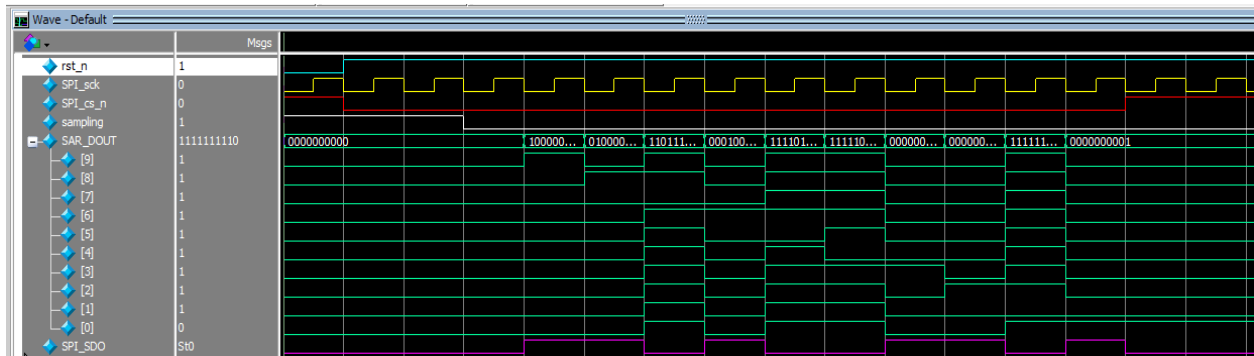
The simulation of the SPI Module was performed on ModelSim, a tool from Siemens that makes a representation of the bench test. The bench test code was developed in Verilog. All the signals have an initial value or reset state to have a known state.

During the simulation, ten different data were sent. All data changes just the bit that will be sent e.g. the first data has just the D9 on 1 and the rest of them in 0, the second one has the D8 on 1 and the rest on 0. this sequence continues until the LSB is sent (Fig. 3.7).

```
#10 SAR_DOUT = 10'b10_0000_0000;
#10 SAR_DOUT = 10'b01_0000_0000;
#10 SAR_DOUT = 10'b11_0111_1111;
#10 SAR_DOUT = 10'b00_0100_0000;
#10 SAR_DOUT = 10'b11_1101_1111;
#10 SAR_DOUT = 10'b11_1110_1111;
#10 SAR_DOUT = 10'b00_0000_1000;
#10 SAR_DOUT = 10'b00_0000_0100;
#10 SAR_DOUT = 10'b11_1111_1101;
#10 SAR_DOUT = 10'b00_0000_0001;
```

**Figure 3.7** Sequence of data to be sent.

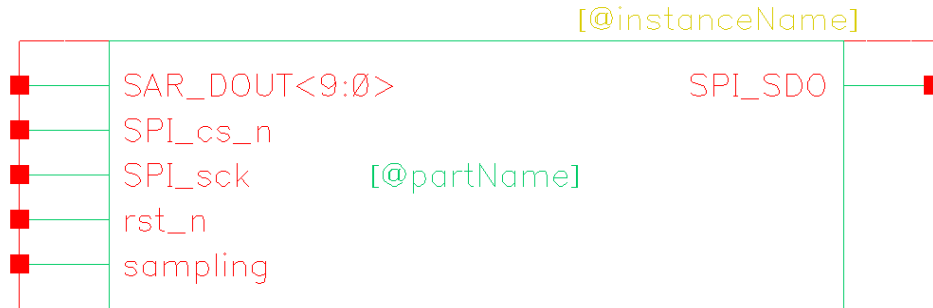
Before running the simulation, ModelSim reviews the code to identify no typos or any other issues. Once the design has no errors the simulation can be performed (Fig. 3.8).



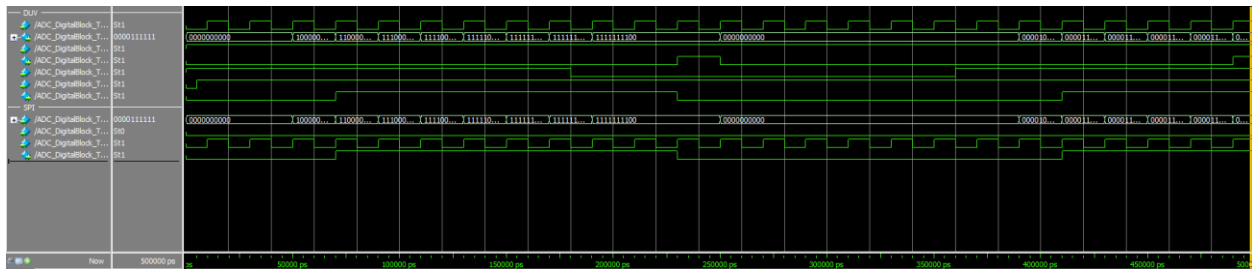
**Figure 3.8** SPI Module simulation.

### 3.4. SPI Module Integration with SAR

After the simulation is working as expected, the SPI Module is imported to Virtuoso. Using the Verilog option on the cell view, Virtuoso creates a black box that recreates the functionality of the Verilog code in the Virtuoso environment (Fig. 3.9).

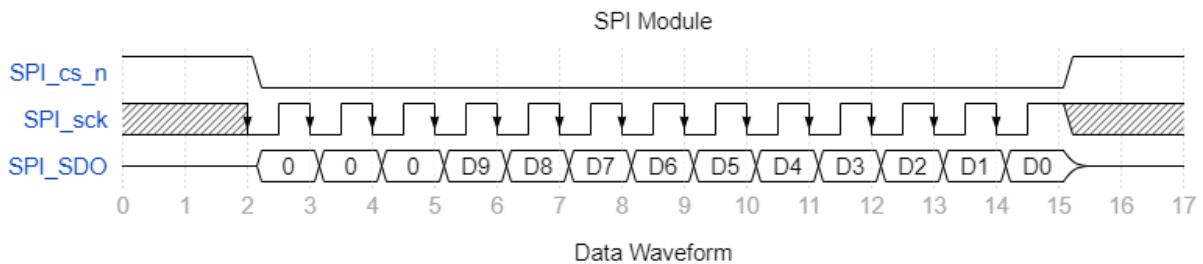


**Figure 3.9** SPI Module Figure created by Virtuoso



**Figure 3.10** SPI and SAR integration waveform.

Once the integration was made, and the SAR and SPI's waveforms were reviewed (Fig. 3.10), the results indicate that after three clock cycles the first bit, means the MSB, will be sent through SPI SDO to be handled externally (Fig. 3.11).



**Figure 3.11** SPI and SAR integration results.

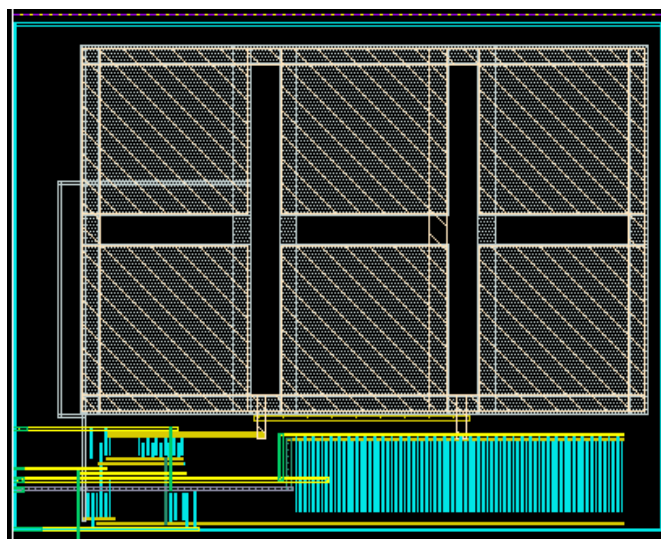
## 4. ADC Integration

### 4.1. Logic Synthesis

Logic synthesis transforms behavioral hardware description language (HDL) code into a *netlist* describing the hardware as a model represented by logic blocks and the connections between them. In this project, logic synthesis was done by Encounter RTL Compiler (RC) which needs several files as inputs so it can do the synthesis of the design. The output of the RC tool is a netlist and some reports with parameters like total area, fanout, total power consumption, number of used cells, etc.

To synthesize the full custom modules as the bandgap reference, dynamic comparator, bootstrap switch, and DAC capacitive, it is necessary to generate the Library Exchange Format (LEF) files of each cell. LEF Files contain information about the area of the cell, metal connection, and vias. It is the representation of the layout of the cell using coordinates.

The LEF Files are generated from the abstract view of the layout once it is DRC and LVS verified. Figures 4.1 – 4.4 show the abstracts of the full custom modules of the ADC.



**Figure 4.1** Bootstrap Switch Abstract.

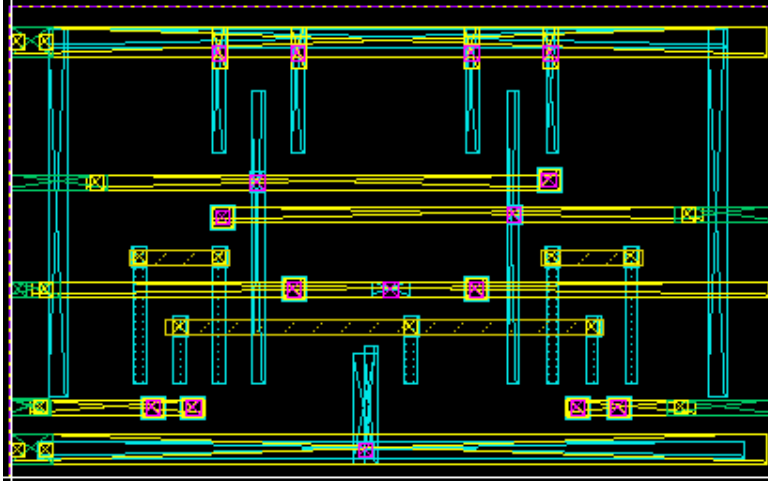


Figure 4.2 Dynamic Comparator Abstract.

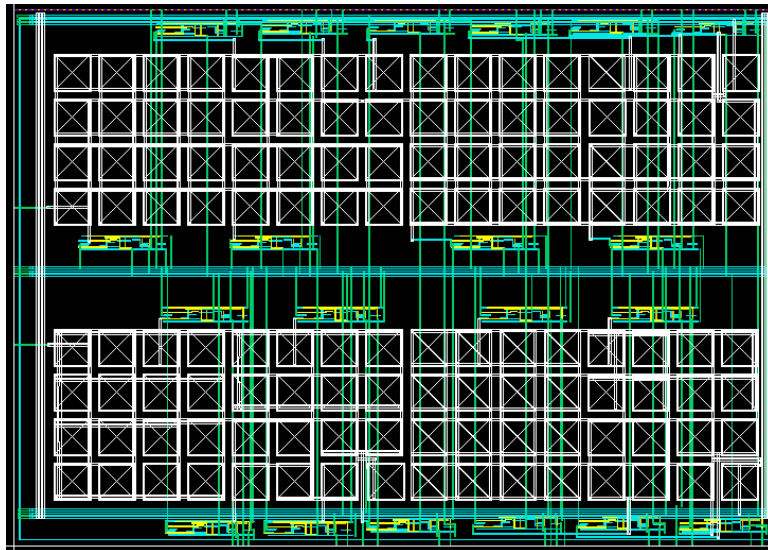


Figure 4.3 DAC Capacitive Abstract.

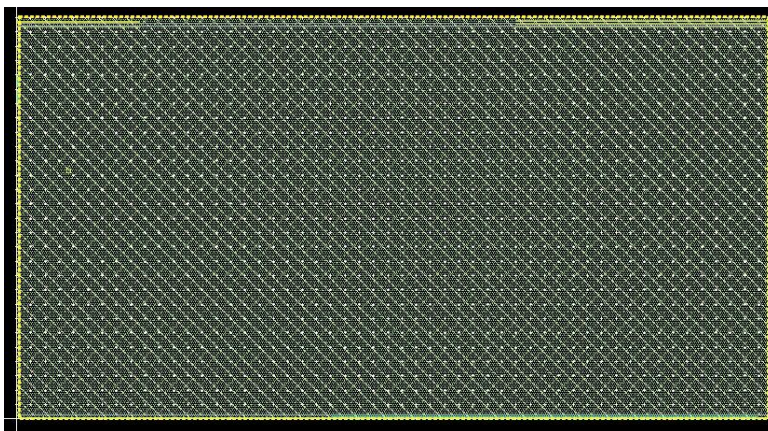


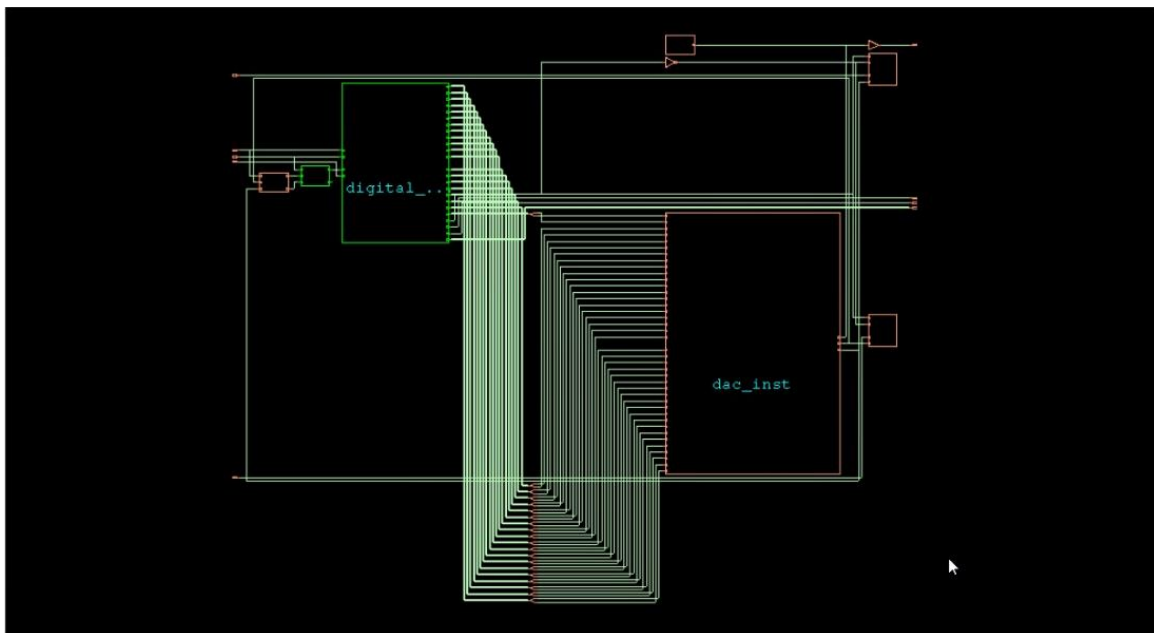
Figure 4.4 BandGap Abstract.



The SAR ADC was synthesized using the RC tool. For doing a logic synthesis, constraints were defined. This information can be found completely in a sdc file, but some of the most important are the next ones:

- There's only 1 clock (Main\_CLK) that works with a period of 4000 picoseconds
- Both the input and the output delay where 10% of the main clock's period
- The external driver that was used as the buffer BUFFD12BWP7T using the pin Z
- The max transition was 65% of the clock's period
- The max capacitance is 6000 femtofarads
- The maximum fanout is 50

Figure 4.5 shows the schematic of the design. There is a top module that connects all the modules that conform to the ADC, full custom modules are shown in orange color. The RTL diagram shows all connections are correct.



**Figure 4.5** Schematic of Synthesized RTL.

The following reports were generated by the RC tool for the typical and worst-case after the SAR ADC were synthesized.

**Resultados Typ:**

```

Module: ADC_LP_bb
Technology libraries: tcb018gbwp7twc 270
                    tpd018nvwc 280a
                    physical_cells
Operating conditions: WCCOM
Interconnect mode: global
Area mode: physical library

```

Timing  
-----

```

Clock Period
-----
Main_Clk 4000.0

```

Cost Group	Critical Path Slack	TNS	Violating Paths
C2C	947.2	0	0
C2O	1490.4	0	0
default	3683.9	0	0
I2C	3329.5	0	0
I2O	No paths	0	
Total		0	0

Instance Count  
-----

```

Leaf Instance Count          249
Sequential Instance Count     72
Combinational Instance Count  177
Hierarchical Instance Count   44

```

Area  
-----

```

Cell Area                    227032.622
Physical Cell Area           0.000
Total Cell Area (Cell+Physical) 227032.622
Net Area                      2232.648
Total Area (Cell+Physical+Net) 229265.270

```

Power  
-----

```

Leakage Power                0.139 uW
Dynamic Power                 2148.675 uW
Total Power                   2148.814 uW

```

```

Max Fanout                    68 (ADC_SPI_sck)
Min Fanout                    0 (digital_inst/sar_lp_inst/DobleRegs/sw9n/SW_ctrl[0])
Average Fanout                2.5
Terms to net ratio            3.3
Terms to instance ratio       3.6
Runtime                       14.996 seconds
Elapsed Runtime                31 seconds
RC peak memory usage:         175.00
EDI peak memory usage:        no_value
Hostname                       FV00
Final Runtime & Memory.

```

```

=====
The RUNTIME after FINAL is 15 secs
and the MEMORY_USAGE after FINAL is 173.00 MB
=====

```

**Results WC:**

Module: ADC\_LP\_bb  
Technology libraries: tcb018gbwp7twc 270  
tpd018nvwc 280a  
physical\_cells  
Operating conditions: WCCOM  
Interconnect mode: global  
Area mode: physical library

Timing  
-----

Clock Period  
-----  
Main\_CLK 4000.0

Cost Group	Critical Path Slack	TNS	Violating Paths
C2C	934.7	0	0
C2O	1348.1	0	0
default	3574.9	0	0
I2C	3250.1	0	0
I2O	No paths	0	
-----			
Total		0	0

Instance Count  
-----

Leaf Instance Count 249  
Sequential Instance Count 72  
Combinational Instance Count 177  
Hierarchical Instance Count 44

Area  
-----

Cell Area 227032.622  
Physical Cell Area 0.000  
Total Cell Area (Cell+Physical) 227032.622  
Net Area 2232.648  
Total Area (Cell+Physical+Net) 229265.270

Power  
-----

Leakage Power 0.139 uW  
Dynamic Power 2150.416 uW  
Total Power 2150.556 uW

Max Fanout 68 (ADC\_SPI\_sck)  
Min Fanout 0 (digital\_inst/sar\_lp\_inst/DobleRegs/sw9n/SW\_ctrl[0])  
Average Fanout 2.5  
Terms to net ratio 3.3  
Terms to instance ratio 3.6  
Runtime 13.993 seconds  
Elapsed Runtime 26 seconds  
RC peak memory usage: 175.00  
EDI peak memory usage: no\_value  
Hostname FV00  
Final Runtime & Memory.

From this synthesis report, the main clock, Main\_CLK, was used for both cases with a 4000ps period, therefore working with a frequency of 250 MHz. The cost group did not show any path violation in the typical and worst case.

In this report, we can also see the number of instances that are in the design and the area that was used for design. Something that we notice is that the physical cell area is 0. To see the Physical cell area, we had to look for another report, that can be found in the next figure. We can see the area of our modules in the first 5 rows of the report.

Generated by: Encounter(R) RTL Compiler v14.10-s022\_1 (Sep 3 2014)  
 Generated on: Aug 03 2021 22:04:26  
 Module: ADC\_LP\_bb  
 Technology libraries: tcb018gbwp7twc 270  
 tpd018nvw 280a  
 Operating conditions: WCCOM  
 Interconnect mode: ple

Gate	Instances	Area
TOTAL	255	227362.23
Bootstrap_switch_split	2	21166.82 physical_cells
ADC_BGR	1	37113.41 physical_cells
DAC	1	162415.64 physical_cells
Dyn_comp_StrongArm_2in	1	124.68 physical_cells
INVD0BWP7T	59	388.55 tcb018gbwp7twc
MUX2D0BWP7T	51	1007.60 tcb018gbwp7twc
DFCNQD1BWP7T	39	1883.48 tcb018gbwp7twc
DFSND0BWP7T	20	1053.70 tcb018gbwp7twc
AN2D0BWP7T	11	120.74 tcb018gbwp7twc
BUFFD12BWP7T	11	531.24 tcb018gbwp7twc
EDFCNQD2BWP7T	10	658.56 tcb018gbwp7twc
INR2D0BWP7T	9	98.78 tcb018gbwp7twc
HA1D0BWP7T	5	164.64 tcb018gbwp7twc
NR2D0BWP7T	5	43.90 tcb018gbwp7twc

**Figure 4.6** Report of Area utilized in the gates that were used in the project.

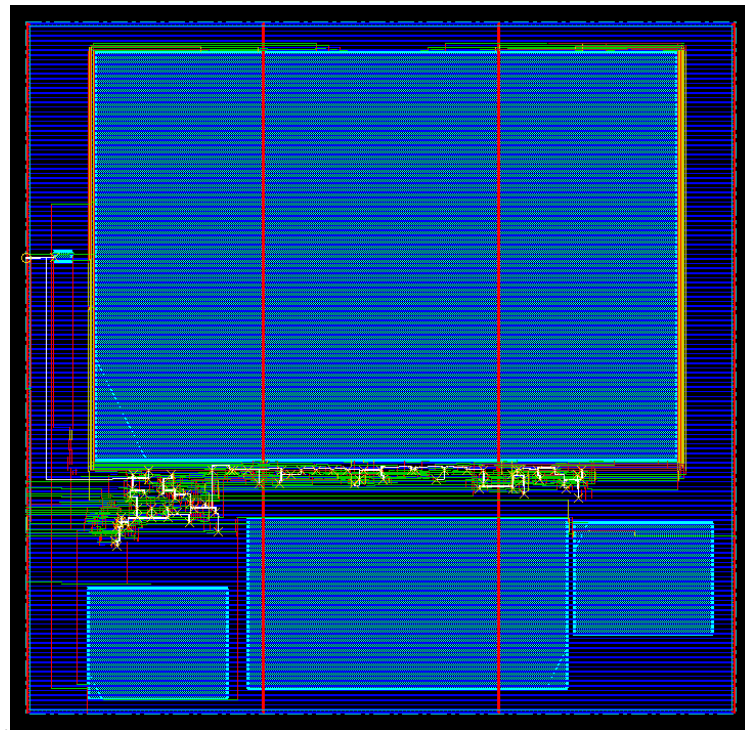
The total power that the ADC will consume is 6587.202 uW. This is high for the functionality that we want to give. And finally, the maximum fanout is 68 and the clock ADC\_SPI\_sck has it. This is the main clock of the system, so it makes sense that it goes to all those instances since we have not added a clock tree.

## 4.2. Physical Synthesis

The output of the logic synthesis is given to a Place and Route Tool, in this case, Encounter Digital Implementation (EDI) to perform Physical Synthesis. The objective of the Physical synthesis is to optimize the design in terms of area, routing, and timing.

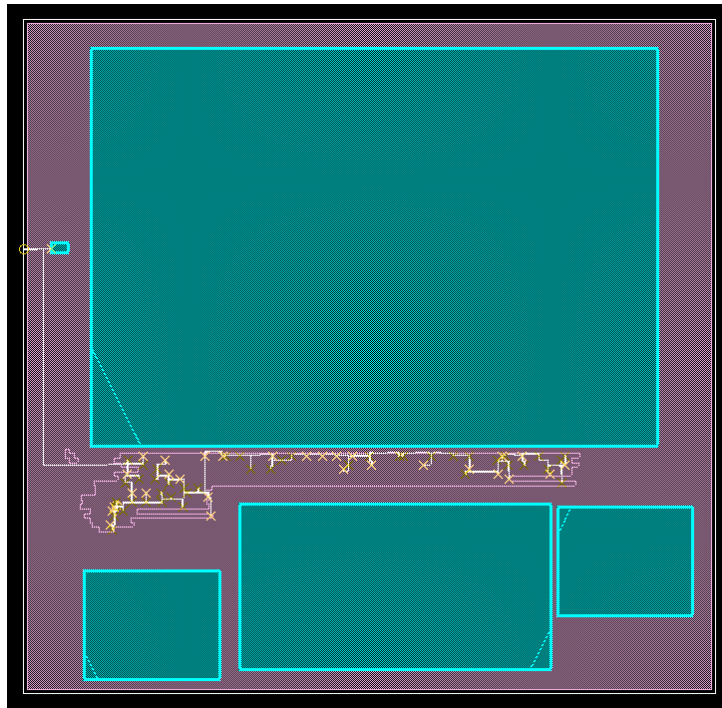
Physical synthesis needs several inputs to be performed: Verilog netlist, LEF Libraries, timing libraries, and timing constraints. The netlist is the result of the logic synthesis and describes the connections among all the components as logic gates, macros, and pins. Every cell must have a LEF library that contains detailed information of area and routing. In this project, timing libraries were not generated.

The process of the Physical synthesis that is performed in this design consists of six steps: floorplan definition, power ring creation, placement, clock tree synthesis, and routing. Figure 4.7 shows the placement of the different modules (DAC, dynamic comparator, bootstrap switches, and bandgap), standard cells, and connections.



**Figure 4.7** Physical view of the ADC.

Figure 4.8 shows the result of display only the Clock Tree.



**Figure 4.8** Clock Tree View.

The next report shows geometry verification, this test check for violations of wiring, shorts between nets, the overlap between cells, etc.

```
*** Starting Verify Geometry (MEM: 884.1) ***

VERIFY GEOMETRY ..... Starting Verification
VERIFY GEOMETRY ..... Initializing
VERIFY GEOMETRY ..... Deleting Existing Violations
VERIFY GEOMETRY ..... Creating Sub-Areas
..... bin size: 8320
VERIFY GEOMETRY ..... SubArea : 1 of 1
VERIFY GEOMETRY ..... Cells           : 0 Viols.
VERIFY GEOMETRY ..... SameNet         : 0 Viols.
VERIFY GEOMETRY ..... Wiring          : 182 Viols.
VERIFY GEOMETRY ..... Antenna         : 0 Viols.
VERIFY GEOMETRY ..... Sub-Area : 1 complete 182 Viols. 0 Wrngs.
VG: elapsed time: 1.00
Begin Summary ...
Cells           : 0
SameNet        : 0
Wiring         : 2
Antenna        : 0
Short          : 180
Overlap        : 0
End Summary
```

Verification Complete : 182 Viols. 0 Wrngs.

\*\*\*\*\*End: VERIFY GEOMETRY\*\*\*\*\*

## Connectivity verification

\*\*\*\*\* Start: VERIFY CONNECTIVITY \*\*\*\*\*

Start Time: Sun Aug 8 22:12:17 2021

Design Name: ADC\_LP\_bb

Database Units: 2000

Design Boundary: (0.0000, 0.0000) (587.7050, 572.3200)

Error Limit = 1000; Warning Limit = 50

Check all nets

Net vref\_gnd\_w: no routing.

Net vref\_vdd\_w: no routing.

Begin Summary

2 Problem(s) (ENCVFC-98): Net has no global routing and no special routing.

2 total info(s) created.

End Summary

End Time: Sun Aug 8 22:12:18 2021

Time Elapsed: 0:00:01.0

\*\*\*\*\* End: VERIFY CONNECTIVITY \*\*\*\*\*

## DRC verification

\*\*\* Starting Verify DRC (MEM: 959.9) \*\*\*

VERIFY DRC ..... Starting Verification

VERIFY DRC ..... Initializing

VERIFY DRC ..... Deleting Existing Violations

VERIFY DRC ..... Creating Sub-Areas

VERIFY DRC ..... Using new threading

VERIFY DRC ..... Sub-Area : 1 of 1

VERIFY DRC ..... Sub-Area : 1 complete 182 Viols.

Verification Complete : 182 Viols.

## 5. Conclusions

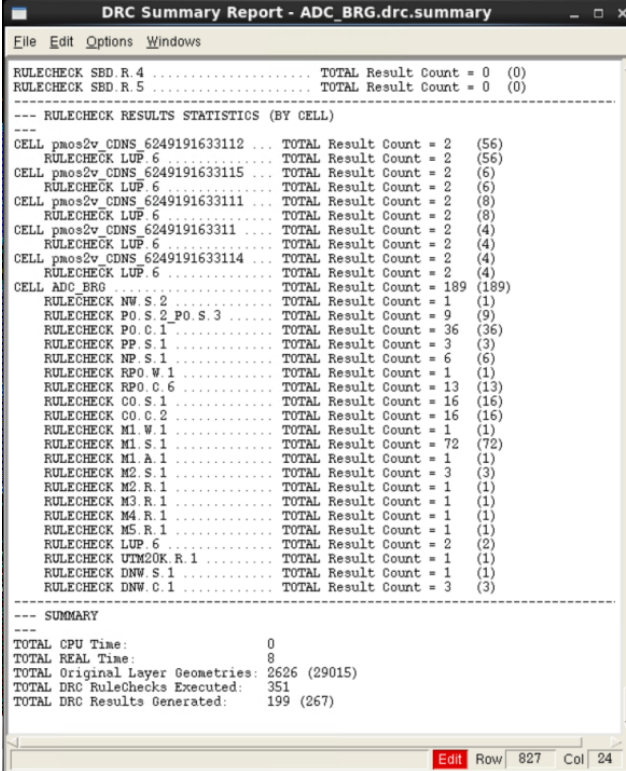
Designing an ADC, or any IC module is a big challenge. Even if the tasks are divided among all the team members. Each block or module requires a great level of understanding of the full design functionality. Through this document, the journey to design two blocks was described, an SPI Communication Module, and a BandGap Reference of a SAR ADC. In both modules were faced different issues. These inconveniences were handled in the best possible way to comply with the 10 bits SAR ADC requirements, applying everything learned during the specialty course.

In the beginning, project requirements for the BandGap were set, such as the temperature range from  $-40\text{ }^{\circ}\text{C}$  to  $120\text{ }^{\circ}\text{C}$  with a variation of  $\pm 500\text{ }\mu\text{V}$ . Not all the initial requirements were accomplished for the CMOS – BJT topology that was chosen. Then, the temperature range was decreased from  $-40\text{ }^{\circ}\text{C}$  to  $85\text{ }^{\circ}\text{C}$  to get the desired reference voltage. PVT corner analysis shows that not all the corners are reached on this design and the BGR is capable to work under the typical corner. On the other hand, the communication serial module showed fewer complications. The requirements on this module were achieved easier due to the number of bits that the module converts from parallel data to serial SPI format. It is adjusted according to the 10 bits SAR ADC specifications.



## 6. Appendix A

In this section, the DRC errors faced in chapter 2.4 will be described. After finish, the layout, the first route on the BGR generated 199 DRC errors (Fig. 5.1). Almost all the errors were related to density and spacing, except for LUP.6.



```
File Edit Options Windows
DRC Summary Report - ADC_BRG.drc.summary
-----
RULECHECK SBD R. 4 ..... TOTAL Result Count = 0 (0)
RULECHECK SBD R. 5 ..... TOTAL Result Count = 0 (0)
-----
--- RULECHECK RESULTS STATISTICS (BY CELL)
---
CELL pmos2v CDNS 6249191633112 ... TOTAL Result Count = 2 (56)
  RULECHECK LUP 6 ..... TOTAL Result Count = 2 (56)
CELL pmos2v CDNS 6249191633115 ... TOTAL Result Count = 2 (6)
  RULECHECK LUP 6 ..... TOTAL Result Count = 2 (6)
CELL pmos2v CDNS 6249191633111 ... TOTAL Result Count = 2 (8)
  RULECHECK LUP 6 ..... TOTAL Result Count = 2 (8)
CELL pmos2v CDNS 624919163311 ... TOTAL Result Count = 2 (4)
  RULECHECK LUP 6 ..... TOTAL Result Count = 2 (4)
CELL pmos2v CDNS 6249191633114 ... TOTAL Result Count = 2 (4)
  RULECHECK LUP 6 ..... TOTAL Result Count = 2 (4)
CELL ADC BRG ..... TOTAL Result Count = 189 (189)
  RULECHECK NW S. 2 ..... TOTAL Result Count = 1 (1)
  RULECHECK PO S. 2_PO S. 3 ..... TOTAL Result Count = 9 (9)
  RULECHECK PO C. 1 ..... TOTAL Result Count = 36 (36)
  RULECHECK PP S. 1 ..... TOTAL Result Count = 3 (3)
  RULECHECK NP S. 1 ..... TOTAL Result Count = 6 (6)
  RULECHECK RPO W. 1 ..... TOTAL Result Count = 1 (1)
  RULECHECK RPO C. 6 ..... TOTAL Result Count = 13 (13)
  RULECHECK CO S. 1 ..... TOTAL Result Count = 16 (16)
  RULECHECK CO C. 2 ..... TOTAL Result Count = 16 (16)
  RULECHECK M1 W. 1 ..... TOTAL Result Count = 1 (1)
  RULECHECK M1 S. 1 ..... TOTAL Result Count = 72 (72)
  RULECHECK M1 A. 1 ..... TOTAL Result Count = 1 (1)
  RULECHECK M2 S. 1 ..... TOTAL Result Count = 3 (3)
  RULECHECK M2 R. 1 ..... TOTAL Result Count = 1 (1)
  RULECHECK M3 R. 1 ..... TOTAL Result Count = 1 (1)
  RULECHECK M4 R. 1 ..... TOTAL Result Count = 1 (1)
  RULECHECK M5 R. 1 ..... TOTAL Result Count = 1 (1)
  RULECHECK LUP 6 ..... TOTAL Result Count = 2 (2)
  RULECHECK UTM20K R. 1 ..... TOTAL Result Count = 1 (1)
  RULECHECK DNW S. 1 ..... TOTAL Result Count = 1 (1)
  RULECHECK DNW C. 1 ..... TOTAL Result Count = 3 (3)
-----
--- SUMMARY
---
TOTAL CPU Time: 0
TOTAL REAL Time: 8
TOTAL Original Layer Geometries: 2626 (29015)
TOTAL DRC RuleChecks Executed: 351
TOTAL DRC Results Generated: 199 (267)
-----
Edit Row 827 Col 24
```

Figure 5.1 Total DRC errors after first route on the BGR,

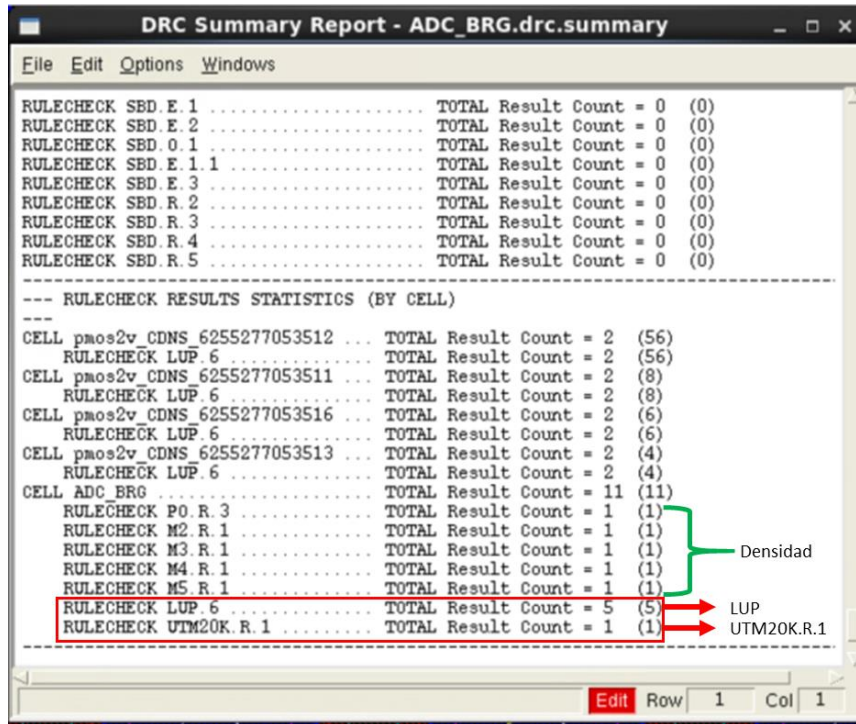


Figure 5.2 Spacing errors corrected.

## 6.1. LUP.6

The Calibre RVE reported that LUP.6 that any point inside NMOS/PMOS source/drain space to the nearest PW/NW STRAP in the same PW/NW  $\leq 30 \mu\text{m}$ .

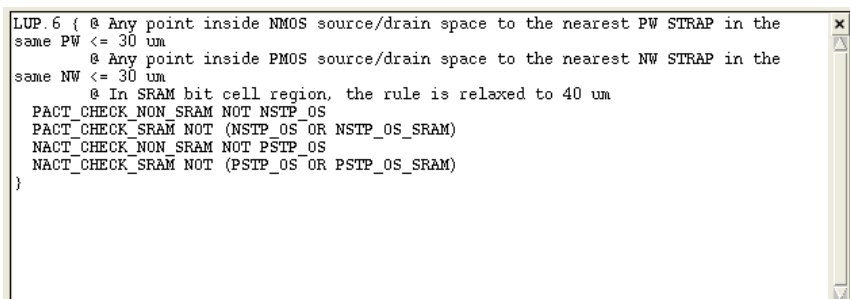
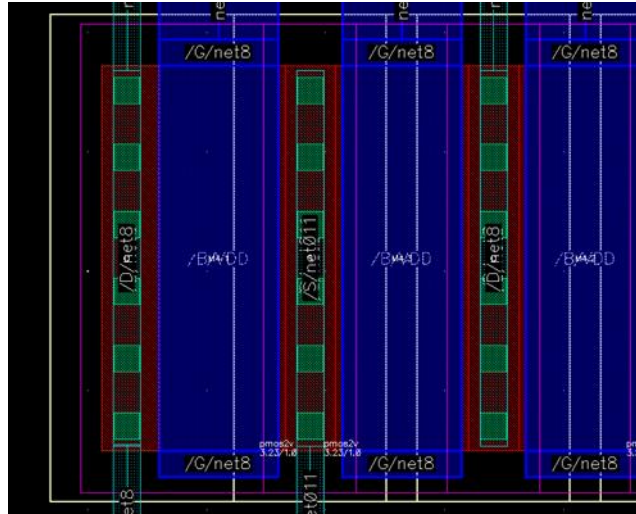
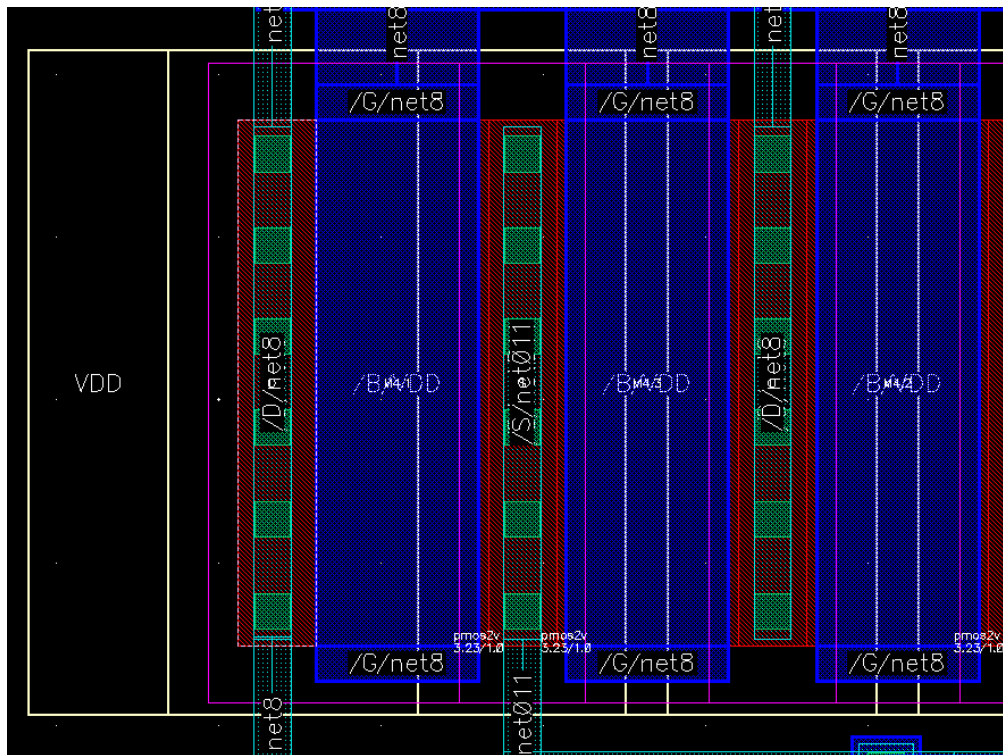


Figure 5.3 Calibre RVE report.

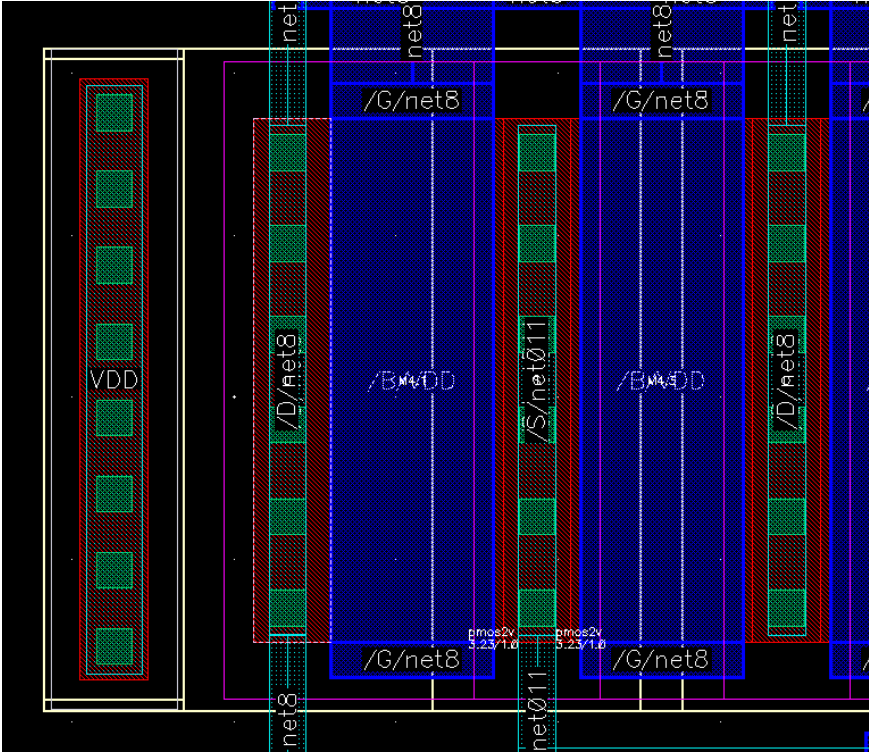
The error is shown in coordinate. That coordinate points to the left side of the transistor arrangement. After the localization of the error, due to all the transistors being P- Channel extra frames of NWell were connected to the transistors and connected to VDD troughs vias.



**Figure 5.4** LUP.6 Error pointing the error on the left of the transistor arrange.



**Figure 5.5** Extra figure of NWell connected to the transistors.



**Figure 5.6** Adding vias on the NWell to fix the LUP.6 error

Once the LUP.6 errors were corrected, there were no more DRC errors.

## 7. Bibliography

- AMSAc Lab Group. (2021). Cadence Tutorial B: Layout, DRC, Extraction, and LVS. *MSU VLSI program*.
- Baker, R. J. (2010). *CMOS Circuit Design, Layout, and Simulation*. Hoboken, New Jersey: John Wiley & Sons Inc.
- Chávez-Hurtado, J. L., & Rayas-Sánchez, J. E. (2016). *Polynomial-Based Surrogate Modeling of RF and Microwave Circuits in Frequency Domain Exploiting the Multinomial Theorem*. Tlaquepaque, Jalisco, Mexico.
- Chávez-Hurtado, J. L., Rayas-Sánchez, J. E., & Brito-Brito, Z. (2016). Multiphysics Polynomial-Based Surrogate Modeling of Microwave Structures in Frequency Domain. Tlaquepaque, Jalisco, Mexico.
- Dhaker, P. (2018, September). *Analog Dialogue*. Retrieved from Introduction to SPI Interface: <https://www.analog.com/en/analog-dialogue/articles/introduction-to-spi-interface.html>
- Floyd, T. L. (2006). *FUNDAMENTOS DE SISTEMAS DIGITALES*. Madrid: PEARSON EDUCACIÓN S.A.
- Floyd, T. L. (2014). *Digital fundamentals: A systems approach*. Pearson.
- Khot, P. (2015). Design of Area efficient and Low Power Bandgap Voltage Reference using Sub-threshold MOST Transistor.
- Martinez Guerrero, E. (2021). *Hybrid CMOS – BJT transistor topology*. Guadalajara: Unpublished.
- Martinez-Nieto, A., Sanz-Pascual, M. T., & Rosales-Quitero, P. (2013). A bandgap voltage reference in 0.18um CMOS technology. Puebla.
- Naganadhan, V. (2019). *Low temperature coefficient bandgap voltage reference generator*. Iowa: Iowa State University.
- Ribeiro, A., Gama, R., Costa, T., Neves, R., & Horta, N. (n.d.). A Bandgap Voltage Reference With Only MOS Transistors. 2009. Lisboa, Portugal.
- TemplatesYard. (2020). *VLSI Physical Design for Freshers*. Retrieved from PVT (Process, Voltage, Temperature): <https://www.physicaldesign4u.com/2020/07/pvt-process-voltage-temperature.html#:~:text=PVT%20is%20the%20Process%2C%20Voltage,These%20conditions%20are%20called%20corners>
- Yan Zhu, C.-H. C.-F.-W.-P. (2014, February). *Split-SAR ADCs: Improved Linearity With Power and Speed Optimization*. Retrieved from IEEE Transactions on Very Large Scale Integration (VLSI) Systems: <https://ieeexplore.ieee.org/document/6462023/authors#authors>