Low Power Analog-to-Digital Converter for Visual Prosthesis

- ADC de Baixo Consumo para Próteses Visuais -

Ivanildo José Pereira Gomes

Dissertation submitted for obtaining the degree of
Master in Electrical and Computer Engineering

Jury

Supervisor: Prof. Jorge Manuel dos Santos Ribeiro Fernandes
Co-Supervisor: Doctor Miguel Andrade Martins
President: Prof. Marcelino Bicho dos Santos
Members: Prof. José António Beltran Gerald

October 2009
"When life knocks you down you have two choices: stay down or get up."

(Tom Krause)
Acknowledgements

First I would like to thank Professor Jorge M. R. Fernandes for allowing me to write this thesis under his supervision. His continuous support and encouragement were essential to the development of this work. I would like also to thank him for receiving me at Instituto de Sistemas de Engenharia e Computadores (INESC), group of Analog and Mixed-Signal, providing me all the necessary working conditions in such a friendly environment. Finally I thank him for being my tutor from whom I acknowledged wise advices and orientation from the very beginning of my college life.

Next, I would like to thank my co-supervisor Doctor Miguel A. Martins for his work guidance through the whole project. Without his knowledge, dedication, patience and persistence this work would not have been possible. I would also like to thank him especially for the humour and good feeling that he spreads to everyone around him.

I would like to thank FCT (Fundação para a Ciência e Tecnologia) who supported this work through project ICONS.

I would like to thank Professor António C. Pinto for his availability and great timing on his help to this work and Rui Duarte and Pedro Jesus for all the healthy conversations on our shared workplace.

I would also like to thank, my dear friends, André Chee, Bruno Vieira, Carlos Calisto, Carla Almeida, Emanuel Paz, Gonçalo Codinha, Gonçalo Correia, Gualter Monteiro, Hugo Leal, Hugo Varandas, João Duarte, Mauro Sardinha, Paulo Ferreira, Ricardo Preguiça, Rodrigo Chiolas, Rui Ramalho, Tiago Raposo, to all RAG 2007/2008 residents, and all my friends that weren’t mentioned but not forgotten.

I would like to thank Ana Lúcia for make the world a better place.

And mostly I would like to thank my family: my father, Francisco, for representing an example of courage and generosity that inspires me every day to be a better person; my mother, Paula, for her unconditional love and support through all my life; and my sister, Ineida, for being my best friend.
Abstract

It is easily argued that profound vision loss is the most debilitating of all human senses. Project icons is developing prototypes in the visual rehabilitation field to return sight for blind people. The integrated circuit developed by this research program requires an Analog to Digital Converter (ADC).

This thesis presents the project of a Successive Approximation Register (SAR) ADC, using a CMOS inverter as a comparator. The main goal is to achieve a low power consumption, low frequency and low area solution.

In the first part of this work, the switched capacitor array is studied along with the capacitor sizing considerations. Then the design of the SAR control logic block is made using a VHDL approach. The CMOS inverter used as comparator was selected by its attractive advantages which better suit the required specifications; no static power consumption, no reference voltage and low area.

A 5-bit SAR ADC is implemented and simulated after all the blocks were selected. Time and frequency domain analysis were made confirming the feasibility of the circuit. The ADC core reached a power consumption of approximately 1.1 mW and an estimated area of 0.06886 mm².

In the second part of this work, a 5-bit 2-bit/step SAR ADC is developed with the main purpose of reducing the required number of clock cycles to complete a conversion, which was successfully achieved.

Keywords

Integrated Circuits, Analog to Digital Converter (ADC), Successive Approximation Register (SAR), Comparator, CMOS Inverter.
Resumo

É facilmente argumentável que a perda profunda de visão é a maior debilitadora entre os sentidos humanos. O projecto *Icons* está a desenvolver protótipos no campo da reabilitação visual para retornar a visão aos cegos. O circuito integrado desenvolvido por este programa de investigação requerer um Conversor Analógico-Digital (ADC).

Esta tese apresenta o projecto de um Conversor Analógico-Digital com a topologia de registo de aproximações sucessivas, usando um inversor CMOS como comparador. Esta tem como principal objectivo obter uma solução de baixo consumo, baixa frequência e área reduzida.

Numa primeira versão do ADC, o vector de condensadores comutados foi estudado, tal como as considerações para o seu dimensionamento. O desenho do bloco de controlo lógico do registo de aproximações sucessivas foi abordado utilizando a linguagem de descrição de *hardware* VHDL. O inversor CMOS usado como comparador, foi escolhido pelas suas vantagens que se enquadram nas especificações propostas. Este comparador não tem consumo estático, não requer tensões de referência e tem área reduzida com ganho elevado.

Um ADC de topologia registo de aproximações sucessivas foi implementado e simulado após os blocos serem selecionados. O circuito global tem um consumo de aproximadamente 1.1 mW e a área estimada de 0.06886 mm².

Um segundo ADC de 5-bits com 2-bits por ciclo, foi desenvolvido tendo como objectivo principal reduzir o número de ciclos de relógio para completar uma conversão, o que foi conseguido com sucesso.

Palavras-chave

Circuitos Integrados, Conversor Analógico-Digital, ADC por aproximações sucessivas, Comparador, Inversor CMOS.
Table of Contents

Acknowledgements ........................................................................................................... v
Abstract ............................................................................................................................ vii
Resumo ............................................................................................................................... viii
Table of Contents .............................................................................................................. ix
List of Figures .................................................................................................................. xii
List of Tables .................................................................................................................. xiv
List of Acronyms .............................................................................................................. xv
List of Software ............................................................................................................... xvii

1 Introduction ................................................................................................................... 1
  1.1 Project Icons ............................................................................................................ 2
  1.2 Scope and Objectives ................................................................................................. 2
  1.3 Outline ...................................................................................................................... 5

2 Basic Concepts ............................................................................................................... 7
  2.1 Analog to Digital Converter ...................................................................................... 8
  2.2 DC Accuracy ............................................................................................................. 10
     2.2.1 Offset error, full-scale error .............................................................................. 10
     2.2.2 Non-linearity .................................................................................................... 11
     2.2.3 Absolute error .................................................................................................. 13
  2.3 Dynamic Performance .............................................................................................. 14
  2.4 Traditional ADC Architectures ................................................................................. 16
2.4.1 Flash ADC ................................................................. 16
2.4.2 Successive approximation register ADC (SAR ADC) ....................... 17
2.4.3 Pipelined ADC ................................................................ 18
2.4.4 Integrating ADC ................................................................ 19
2.4.5 Sigma-delta ADC ................................................................ 19
2.5 Comparison between ADC architectures ............................................. 20
2.6 Why Choose a SAR ADC? ............................................................... 21
3 5-bit SAR ADC using an inverter as comparator ............................. 23
  3.1 Implemented SAR ADC ............................................................ 24
  3.2 Capacitor Array ......................................................................... 24
  3.3 Switches .................................................................................... 28
  3.4 Control Logic Block .................................................................... 30
    3.4.1 SAR logic block ..................................................................... 30
    3.4.2 Sample & hold block ............................................................... 32
    3.4.3 Non-overlapping block ............................................................ 33
    3.4.4 Bit register .............................................................................. 33
  3.5 Comparator .................................................................................. 34
  3.6 Simulations and Results ................................................................. 35
    3.6.1 Ramp response ....................................................................... 35
    3.6.2 Dynamic performance ............................................................. 36
    3.6.3 Corner analysis ....................................................................... 38
    3.6.4 Monte Carlo analysis ............................................................... 39
    3.6.5 Layout ................................................................................... 40
4 5-bit 2-bit/step SAR ADC using inverters as comparators .......... 43
  4.1 Proposed SAR-ADC ..................................................................... 44
  4.2 SAR Control Logic Block ............................................................... 45
  4.3 Simulations and Results ................................................................. 48
5 Conclusions ..................................................................................... 51
  5.1 Summary ....................................................................................... 52
  5.2 Future work .................................................................................. 52
6 References ....................................................................................... 54
Appendix A – VHDL Code ................................................................. 56
Appendix B - Tutorial ....................................................................... 61
List of Figures

Figure 1.1 Artistic representation of *phosphenes* (extracted from [5]). ................................................. 2
Figure 1.2 Example of the implementation of a microelectrode array on a patient (extracted from [6]). ................................................................. 3
Figure 1.3 Block diagram of project Icons (extracted from [6]). ................................................................. 3
Figure 1.4 Microelectrode array. Left: Intracortical electrode array. Right: illustration of a single electrode in neutral tissue. ........................................... 4
Figure 1.5 Electrode stimulator module. ........................................................................................................ 4
Figure 2.1 Generic ADC. ............................................................................................................................ 8
Figure 2.2 Ideal transfer function of a 3-bit ADC. ....................................................................................... 9
Figure 2.3 3-bit ADC transfer function with 1/2 LSB offset. ................................................................ 9
Figure 2.4 Offset error. ............................................................................................................................... 10
Figure 2.5 Full-scale error. ........................................................................................................................ 11
Figure 2.6 Differential non-linearity error. ............................................................................................... 12
Figure 2.7 Integral non-linearity error. ...................................................................................................... 12
Figure 2.8 Fast Fourier Transfer of an ADC output codes. ................................................................ 14
Figure 2.9 Spurious-free dynamic range. ................................................................................................. 15
Figure 2.10 Flash ADC Architecture. ...................................................................................................... 17
Figure 2.11 SAR ADC Architecture. ....................................................................................................... 17
Figure 2.12 Pipelined ADC architecture. ................................................................................................. 18
Figure 2.13 Pipelined Stage. ...................................................................................................................... 18
Figure 2.14 Single slope ADC architecture. ............................................................................................. 19
Figure 2.15 Sigma Delta ADC architecture............................................................................................ 20
Figure 2.16 Conversion time tradeoff (based on [9]). ............................................................................ 20
Figure 2.17 Component matching tradeoff (based on [9]). ................................................................ 21
Figure 2.18 Complexity tradeoff (based on [9]). .................................................................................. 21
Figure 3.1 Implemented SAR ADC block. ............................................................................................. 24
Figure 3.2 SAR ADC with the capacitor array in the sample phase. ....................................................... 25
Figure 3.3 DAC voltage references according to the work specifications. ...................................... 25
Figure 3.4 SAR ADC with the capacitor array in the hold phase. ........................................................... 26
Figure 3.5 Charge redistribution phase (after the conversion process for the digital word 111100). 26
Figure 3.6 Charge redistribution phase step 1 equivalent circuit .......................................................... 27
Figure 3.7 Transmission gate. Left: Ideal switch representation. ............................................................ 28
Figure 3.8 Ron representation. Left: ON and OFF mode representation. Right: Ron variation versus Control signal........................................................................ 28
Figure 3.9 Implemented switch. Left: Ideal switch representation. Right: Transistor level switch....... 29
Figure 3.10 Overlapped switch control signals. ..................................................................................... 29
Figure 3.11 Charge injection effect (based on [13]). ............................................................................. 30
Figure 3.12 Dummy circuit used to minimize charge injection (base on [13]). .................................. 30
Figure 3.13 Implemented SAR ADC logic flow chart. ..................................................................... 31
Figure 3.14 Control logic block (detailed) .............................................................................................. 31
Figure 3.15 Sample and Hold block signals .......................................................................................... 33
Figure 3.16 Representation of the bit register block.................................................................34
Figure 3.17 Left: CMOS inverter based comparator. Right: CMOS inverter based comparator voltage transfer curve.................................................................34
Figure 3.18 ADC output response to a ramp waveform input ..................................................35
Figure 3.19 ADC output to a 7 period sinusoidal waveform .....................................................37
Figure 3.20 FFT of the codes output (without DC component)..............................................37
Figure 3.21 FFT of the codes output (Zoomed) (without DC component) ..............................38
Figure 3.22 Monte Carlo output codes of 5 iterations ............................................................39
Figure 3.23 SAR control logic layout.........................................................................................40
Figure 3.24 Analog components disposed for area consideration...........................................41
Figure 4.1 5-bit 2-bit/step SAR ADC.......................................................................................44
Figure 4.2 Designed ADCs conversion time diagram............................................................44
Figure 4.4 5-bit 2-bit/step SAR ADC control logic block flowchart .......................................46
Figure 4.3 5-bit 2-bit/step SAR ADC comparator block ........................................................47
Figure 4.5 5-bit 2-bit/step SAR ADC ramp response.............................................................48
List of Tables

Table 3.1 Dynamic parameters ................................................................. 38
Table 3.2 Corner analysis with transistor conditions variations ....................... 39
Table 4.1 5-bit 2-bit/step SAR ADC comparator block 1 transistor sizing .......... 47
Table 4.2 5-bit 2-bit/step SAR ADC comparator block 2 transistor sizing .......... 48
List of Acronyms

ADC  Analog-to-Digital Converter
CMOS Complementary Metal-Oxide Semiconductor
DAC Digital-to-Analog Converter
DNL Differential non-linearity
DC Direct Current
ENOB Effective Number of bits
INL Integral Non-Linearity
L Transistors gate Length
LSB Least Significant Bit
MSB Most Significant Bit
MOS Metal-Oxide Semiconductor
N Number of bits - Resolution
NMOS n-channel MOS Transistor
PMOS p-channel MOS Transistor
R Resistor
SNR Signal-to-Noise Ratio
SAR Successive Approximation Register
SC Switched Capacitor
SFDR Spurious-Free Dynamic Range
SiNAD Signal-to-Noise and Distortion
S&H Sample and Hold
SNR Signal-to-Noise Ratio
THD Total Harmonic Distortion
VHDL "VHSIC Hardware Description Language", VHSIC "Very High Speed Integrated Circuits"
VTC Voltage transfer curve
W Transistors gate width
T Temperature
WO Worst case one process corner
WP Worst case power process corner
WS Worst case speed process corner
WZ Worst case zero process corner
c Capacitor
$C_{ox}$ Gate oxide capacitance per unit area
\( f_{\text{in}} \)  
Input frequency

\( f_s \)  
Sampling frequency

\( k \)  
Boltzmann Constant

\( M_{\text{samples}} \)  
Number of samples in the coherent sampling

\( N_{\text{cycles}} \)  
Number of cycle in the coherent sampling

\( R_{\text{ON}} \)  
Equivalent Resistance ON

\( V_A \)  
Analog voltage

\( V_{\text{DD}} \)  
Voltage Supply

\( V_{\text{FS}} \)  
Full scale voltage

\( V_{\text{in}} \)  
Input Voltage

\( V_{\text{noise}} \)  
Noise voltage

\( V_{\text{LSB}} \)  
Voltage equivalent to one LSB

\( V_{\text{DAC}} \)  
DAC output voltage

\( V_{\text{REF}} \)  
Reference Voltage

\( V_{\text{REFP}} \)  
High voltage in the input voltage range on the ADC

\( V_{\text{REFN}} \)  
Lowest voltage in the input voltage range of the ADC

\( V_{\text{th}} \)  
CMOS inverter quantization level

\( V_{\text{tn}} \)  
NMOS transistor threshold voltage

\( V_{\text{tp}} \)  
PMOS transistor threshold voltage

\( \mu_n \)  
Carrier mobility type n

\( \mu_p \)  
Carrier mobility type p
<table>
<thead>
<tr>
<th>Software</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cadence</td>
<td>Electronic design automation tool</td>
</tr>
<tr>
<td>Design Vision</td>
<td>Logic synthesis tool</td>
</tr>
<tr>
<td>SoC Encounter</td>
<td>RTL-to-GDSII system</td>
</tr>
<tr>
<td>Linux</td>
<td>Operative system</td>
</tr>
<tr>
<td>MATLAB</td>
<td>Numerical computing environment</td>
</tr>
<tr>
<td>Microsoft Excel</td>
<td>Spreadsheet assisted calculation tool</td>
</tr>
<tr>
<td>Microsoft Visio</td>
<td>Design tool (e.g. flowcharts, diagrams, etc)</td>
</tr>
<tr>
<td>Microsoft Windows</td>
<td>Operative system</td>
</tr>
<tr>
<td>Microsoft Word</td>
<td>Text editor tool</td>
</tr>
<tr>
<td>Xilinx ISE</td>
<td>VHDL programming environment</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

This chapter gives a brief overview of a neuroprosthesis prototype in the field of visual rehabilitation. Before establishing objectives the scope and motivations of the project are brought up. At the end of the chapter, the work structure is provided.
1.1 Project *Icons*

In the last decades, visual information has shown a significant evolution, which has lead to a world strongly dominated by visual communication. The loss of sight brings innumerable obstacles to individuals’ life quality. Those affected are no longer capable of living a normal and independent life. Until now, no treatment has been provided to restore vision to these patients, but recent studies showed that electrical stimulation of almost any location along the visual cortex can induce the perception of visual images called *phosphenes* (Figure 1.1), bringing new hope for those who suffer from sight loss [1-4].

![Figure 1.1 Artistic representation of phosphenes(extracted from [5]).](image)

Project *Icons* goal is to develop prototypes in the field of visual rehabilitation and to demonstrate feasibility of a cortical neuroprosthesis, interfaced with the visual cortex, which may restore a limited but useful visual perception to blind people. Even though the full restoration of vision is not possible for now, the distinction of shapes, location of objects could allow these patients to walk in familiar environments and to read enlarged text, providing a substantial improvement in the everyday living of blind people.

1.2 Scope and Objectives

The work presented in this thesis is integrated in the research program described earlier. The research program consists on developing a circuit to stimulate the optic nerves to produce *phosphenes*. This circuit is connected to a microelectrode which, in turn, is connected directly to the optic nerve near the visual cortex (Figure 1.2). The stimulation consists in charging and discharging the nerve with electric charges.
Figure 1.2 Example of the implementation of a microelectrode array on a patient (extracted from [6]).

The main blocks of project icons are shown in Figure 1.3.

The module of interest for this work is the Electrode Stimulator. The three major blocks of the Electrode Stimulator are:

- Digital Controllers Array;
- Digital-to-Analog Converter (DAC);
- Analog-to-Digital Converter (ADC).
This module is connected to an array of microelectrodes which, in turn, is connected to the optic nerves. The array contains microelectrodes, which resembles to small needles (Figure 1.4), that will be place inside the patients head and are connected directly to the optic nerves.

Figure 1.4 Microelectrode array. Left: Intracortical electrode array. Right: illustration of a single electrode in neutral tissue.

The chip containing the array of digital controllers and DACs will be placed on the other side of the array. Holes are made on the chip to connect the current sources directly to the microelectrodes. This technique is called flip-chip. Every microelectrode is intended to have a distinct digital controller and a DAC. Each digital controller controls one DAC. The use of such controller allows the converters to work independently after the reception of valid data through a 12-bit data bus, freeing the bus to address other converters after only one clock cycle. A simplified diagram of the Electrode Stimulator module is presented in Figure 1.5.

Figure 1.5 Electrode stimulator module.

A 10-bit address bus selects one of 1024 microelectrodes. Signal Data Valid goes high when the 12-bit data bus has valid data to be read by the controller. When a digital controller is active, this data is
loaded into registers. Then, the DAC is activated and generates a series of output levels, where each level is a result of a single 7-bit input word and different output levels represent different currents. The remaining 5-bit represent the duration of such stimulation. The ADC collets information (voltage/current) about the state of the microelectrode, to allow the debugging and monitoring of the microelectrode stimulation.

The design of an ADC for this application is the goal of this thesis. The application requires only a 5-bit resolution and a sampling frequency of 1.25 MHz. Due to the relaxed ADC specifications, the main objectives are now to reduce the area and power consumption as much as possible. Therefore a SAR architecture was chosen, because for the lower resolution it can be implemented in its simpler form, without needing to have more advanced techniques like resistor ladders or capacitor segmentation techniques\cite{7}.

The ADC will be implemented using the Austria Micro System (AMS) CMOS (Complementary-Metal-Oxide-semiconductor) 0.35 µm technology and a 3.3 V supply voltage.

1.3 Outline

The organization of this thesis is the following. In Chapter 2, Basic Concepts, general concepts of ADCs are presented. In Chapter 3, 5-bit SAR ADC using an inverter as comparator, the implementation of a SAR ADC is presented, covering all its composing blocks; capacitor array, control logic and comparator. The chapter concludes with the presentation of simulations and results. In Chapter 4, 5-bit 2-bit/step SAR ADC, presents an alternative to the implementation of chapter 3 SAR ADC, with a decrease of clock cycles to complete a conversion cycle. In Chapter 5, Conclusions, summarizes the achievements of the present study and explains future work to further improve the present design. In appendix the VHDL code, schematics and some tutorials that describe the design flow procedure used in the work process, and that are vital for the results achieved, are presented.
Chapter 2

Basic Concepts

This chapter provides an overview of the general concepts on ADCs, mainly focusing on its basic concepts, architectures and tradeoffs. In section 2.1 the basic concepts and specifications of an ADC are presented. In section 2.2, the architectures of some popular ADCs are presented and briefly explained. Finally, in section 2.3, a comparison between ADC architectures is made.
2.1 Analog to Digital Converter

In this chapter, ADC basic concepts are explained. Generally, ADC performance specifications are categorized in two ways: DC accuracy and dynamic performance. These aspects will be covered during this section.

An ADC is a device that converts an analog input (voltage or current) into a digital word output (Figure 2.1). The analog input signal is submitted to sampling operations (discrete in time) and quantification (having $2^N$ possible output values, where $N$ is the converter resolution). The resolution of the converter indicates the number of discrete output values that can be produced over the range of the analog input values. The output values are stored in digital form, and the resolution is expressed in bits.

![Figure 2.1 Generic ADC.](image)

Typically, the transfer function of an ADC should be a straight line, as shown in Figure 2.2, where a 3-bit ideal ADC transfer function is represented. The output code is the lowest (000) when the input is less than 1/8 of the full-scale (the size of this 3-bit ADC's code width) and it reaches its full-scale ($V_{FS}$) output code (111) at 7/8 of full scale. Notice that the last transition occurs at one code width less than full-scale input voltage (reference voltage).
The transfer function can also be implemented with an intentional \(-1/2\) least significant bit (LSB) offset, as shown in Figure 2.3. When this intentional offset is implemented the ADC is referred to as *mid-tread*, being the previous case referred to as *mid-rise* \([9]\) (Figure 2.2).
This shift of the transfer function to the left, shifts the quantization error from a range of (-1 to 0 LSB) to (-1/2 to +1/2 LSB).

Due to circuit limitations, fabrication process variations and other factors, ADCs don’t have this perfect transfer function. Thus other quantities were developed to quantify the ADC’s performance. Those can be divided into DC accuracy and dynamic performance.

2.2 DC Accuracy

2.2.1 Offset error, full-scale error

The ADC ideal transfer function line will intersect the origin of the plot. An offset error can be observed if there is a shift of the entire transfer function to the left or right along the input voltage axis, as shown in Figure 2.4.

![Offset error](image)

The full-scale error is the difference between the ideal code transition to the highest output code and the actual code transition to the highest output code when the offset error is zero. This is observed as a change in slope of the transfer function line as shown in Figure 2.5.
2.2.2 Non-linearity

Ideally, each code width on an ADC's transfer function should be uniform in size. For example, all codes in Figure 2.2 should represent exactly 1/8th of the ADC's full-scale voltage reference. The code width or LSB of an ADC is obtained by (2.1).

\[ V_{LSB} = \frac{V_{2^{N-1}} - V_1}{2^N - 1} \quad [V] \quad (2.1) \]

The difference in code widths from one code to the next is known as the differential non-linearity (DNL). This can be observed as uneven spacing of the code "steps" on the ADC's transfer function plot. In Figure 2.6, a selected digital output code width is shown as larger than the previous code's step size. This difference is the DNL error and is calculated by (2.2) for each code \( i \), being the DNL the maximum of DNL\((i)\).

\[ \text{DNL}(i) = \frac{V_{i+1} - V_i}{V_{LSB}} - V_{LSB} \quad (2.2) \]

The integral non-linearity (INL) is the deviation of an ADC transfer function from a straight line. This line is often the best-fit line among the transition points in the transfer function. INL is determined by measuring the voltage where all code transitions occur and comparing them to the ideal transfer function.
INL error at any given point in an ADC's transfer function is the accumulation of all DNL errors of all previous ADC codes, therefore, it's called integral non-linearity. This is observed as the deviation from a straight-line transfer function shown in Figure 2.7.

**Figure 2.6** Differential non-linearity error.

**Figure 2.7** Integral non-linearity error.
Because integral non-linearity will cause distortion, it will also affect the dynamic performance of an ADC. INL is calculated as shown in Equation 2.3 for each code $i$, being the INL the maximum of INL($i$).

$$\text{INL}(i) = \frac{V_i - (i - 1) \times V_{LSB} - V_1}{V_{LSB}} \text{ [LSB]}$$

(2.3)

2.2.3 Absolute error

The absolute error is the total DC measurement error and is characterized by the offset, full-scale, INL, and DNL errors. Quantization error also affects accuracy, but it’s inherent in the analog-to-digital conversion process (it doesn’t vary from one ADC to another of the same resolution). Offset and full-scale errors can be reduced by calibration at the expense of dynamic range and the cost of the calibration process itself. Offset error can be minimized by adding or subtracting a constant number to or from the ADC output codes. Full-scale error can be minimized by multiplying the ADC output codes by a correction factor.
2.3 Dynamic Performance

ADC’s dynamic performance is specified using parameters obtained via frequency-domain analysis and is measured by performing a fast Fourier transform (FFT) on the output codes of the ADC. In Figure 2.8, the fundamental frequency is the input signal frequency. This is the signal measured with the ADC. Everything else is noise to be characterized with respect to the desired signal. This includes harmonic distortion, thermal noise, and quantization noise. Notice that some sources of noise may not derive from the ADC itself, the distortion and thermal noise caused by the circuit connected at the input to the ADC.

![Figure 2.8 Fast Fourier Transfer of an ADC output codes.](image)

The parameters that measure the dynamic performance of an ADC are described next.

The **Signal-to-Noise Ratio** (SNR) is the ratio of the signal power to the noise power corrupting the signal. The SNR is often expressed in dB, and it is given by:

\[
\text{SNR} = 10 \log_{10} \left( \frac{P_s}{P_n} \right) \text{ [dB]} \tag{2.4}
\]

Where \(P_s\) is the signal power and \(P_n\) is the noise power.
For ADC’s it is also common to calculate the SNR using (2.5).

\[
\text{SNR} = 6.02N + 1.76 \text{ [dB]}
\]  

(2.5)

Where \(N\) is the converter resolution, or the number of bits.

The **Total Harmonic Distortion** (THD) of a signal is a measurement of the harmonic distortion and is defined as the ratio of the sum of the powers of all harmonic frequencies to the power of the fundamental.

\[
\text{THD} = 20 \log_{10} \left( \frac{\sqrt{V_2^2 + V_3^2 + \ldots + V_N^2}}{V_1} \right) \text{ [dB]}
\]  

(2.6)

**Signal-to-noise and distortion** (SiNAD) gives a description of how the measured signal will compare to the noise and distortion. The SiNAD ratio can be calculated using (2.6).

\[
\text{SiNAD} = 20 \log_{10} \left( \frac{V_1}{\sqrt{V_2^2 + V_3^2 + \ldots + V_N^2 + V_{\text{noise}}^2}} \right) \text{ [dB]}
\]  

(2.7)

The **Spurious-free dynamic range** (SFDR) is the difference between the magnitude of the measured signal and its highest spur peak. This spur is typically a harmonic of the measured signal. SFDR is shown in Figure 2.9.

![Figure 2.9 Spurious-free dynamic range.](image-url)
The **Effective number of bits** (ENOB) is given by (2.8), and is obtained with (2.5) and (2.7). By simulation a SiNAD is determined, and from its value an equivalent non integer value for the equivalent resolution of the ADC is determined.

\[
\text{ENOB} = \frac{\text{SiNAD} - 1.76}{6.02} \text{ [Bit]}
\]  

(2.8)

### 2.4 Traditional ADC Architectures

Many ADC architectures have been proposed till now. A possible classification for these architectures is:

- Comparator (1-bit ADC);
- Nyquist ADC Architectures.
  - Flash Converters;
  - SAR Converter;
  - Sub ranging, Error Corrected and Pipelined ADCs;
  - Serial Bit-per-stage Binary and Gray coded (Folding ADC’s).
- Counting and Integrating ADC Architectures;
  - Single Slope ADC/Multi-slope ADCs;
- Sigma Delta ADC.

A brief overview on the most popular ADCs will be presented in the following sections.

#### 2.4.1 Flash ADC

Flash ADCs (Figure 2.10), also known as full-parallel ADCs, are the fastest to convert an analog signal to a digital signal. These ADCs are ideal for applications requiring large bandwidth, however, they typically consume more power than other ADC architectures. The Flash ADC is formed by a series of comparators, each one comparing the input signal, \( V_{in} \), to a unique reference voltage. The comparators output connects to the input of an encoder circuit, which then produces a binary output.
This ADC requires \(2^N - 1\) comparators for an \(N\)-bit conversion. The size and cost of all those comparators makes flash converters generally impractical for precisions greater than 8 bits (255 comparators).

### 2.4.2 Successive approximation register ADC (SAR ADC)

Although there are many variations in the implementation of a SAR ADC, the basic architecture is presented in Figure 2.11.

In this architecture, the analog input voltage, \(V_{IN}\), is held on a sample-and-hold (S&H). Then, to implement the binary search algorithm, the \(N\)-bit register is first set to mid-scale (that is, 100...00, where the most significant bit, MSB, is set to ‘1’). This forces the DAC output, \(V_{DAC}\), to be \(V_{REF}/2\), where \(V_{REF}\) is the reference voltage provided to the ADC. A comparison is then performed to determine if \(V_{IN}\) is smaller or greater than \(V_{DAC}\). If \(V_{IN}\) is greater than \(V_{DAC}\), the comparator output is a logic high or ‘1’
and the MSB of the \( N \)-bit register remains at '1'. Otherwise, if \( V_{IN} \) is smaller than \( V_{LSB} \), the comparator output is a logic low or '0' and the MSB of the register is cleared to logic '0', and that bit will keep its value until the end of conversion. The SAR control logic then moves to the next bit down, forces that bit to a logic high, and does another comparison. The sequence continues all the way down to the LSB, and once this is done, the conversion is complete, and the \( N \)-bit digital word is available in the registers output. Typically SAR ADC's require \( N+2 \) cycles to complete a conversion.

### 2.4.3 Pipelined ADC

The Pipelined ADC basic architecture is show in Figure 2.14. The architecture presents \( N \) cascaded stages, each having the resolution of \( B \)-Bits. In Figure 2.13 a block diagram of a single pipelined stage is presented. Within each stage, the analog input is sampled by the S&H circuit. Then it is quantized by a sub-ADC to resolve \( B \)-bits. Using a sub-DAC the quantized value of the sub-ADC is reconverted into analog voltage and is subtracted from the original input signal to yield the quantization error. The quantization error is then restored to original full scale by an amplifier of gain \( 2^B \). Stages operate concurrently; that is, at any time, the first stage operates on the most recent sample while all other stages operate on residues from previous samples. So the digital outputs of each stage are delayed by using delay elements so that their values correspond to the same input sample. The digital correction block is used to generate the required output bits.

![Figure 2.12 Pipelined ADC architecture.](image)

![Figure 2.13 Pipelined Stage.](image)
2.4.4 Integrating ADC

There are a few ways of designing analog-to-digital converters using an integrator, for example, Single-slope ADC. The Single-slope ADC architecture is shown in Figure 2.12. It uses an integrator circuit, which is basically formed by a capacitor, a resistor and an operational amplifier. The switch makes the necessary control circuit. The integrator produces a ramp waveform on its output, from zero to the maximum possible analog voltage to be sampled. While the waveform is generated, the counter starts counting from 0 to $2^N - 1$, where $N$ is the ADC’s resolution. When the voltage found at $V_{IN}$ (the analog signal) is equal to the voltage achieved by the ramp waveform generated by the integrator, the control circuit captures the last value produced by the counter (by setting the output buffers clock input), which will be the digital correspondent of the analog sample being converted. At the same time, it resets the counter and the integrator, starting the conversion of the next sample. This ADC has a variable conversion time.

![Figure 2.14 Single slope ADC architecture.](image_url)

2.4.5 Sigma-delta ADC

Sigma-delta converters, also called oversampling converters, consist of 2 major blocks, an analog modulator and a digital filter as shown in Figure 2.15. In its simple form an analog modulator, whose architecture is similar to a dual-slope ADC, includes an integrator and a comparator with a feedback loop that contains a 1-bit DAC. The modulator oversamples the input signal, transforming it to a serial bit stream with a frequency much higher than the required sampling rate. The output filter then converts the bit stream to a sequence of parallel digital words at the sampling rate. The delta-sigma converters perform high-speed, low resolution (1-bit) analog to digital conversions, and then remove the resulting high-level quantization noise by passing the signal through analog and digital filters.
2.5 Comparison between ADC architectures

ADCs can be implemented by employing a variety of architectures. However, specific limitations (conversion time, component matching requirements, die size, and others) of each architecture, lead to a different choice depending of the required specifications.

The conversion time, for flash ADCs, does not change materially with the increase in resolution, while for SAR, Pipelined and Sigma-Delta converters this time increases approximately linearly. For integrating ADCs, the conversion time doubles with every bit increase in resolution (Figure 2.16).

Component matching requirements in the circuit double with every bit increase in resolution, for flash, successive approximation and pipelined converters. For integrating and sigma-delta converters the component matching does not materially increase with an increase in resolution (Figure 2.17).
2.6 Why Choose a SAR ADC?

Comparatively, the SAR ADC will have a few advantages over the other ADCs to fulfill these work specifications. First, the SAR ADC consumes much less power since its structure consists of only one comparator, switched capacitors and digital circuitry, all consuming little power. Second it follows a very simple principle, the binary search algorithm, which makes it simpler to design and implement than for instance sigma-delta ADCs. Since the main purpose of this work is to achieve a low power and low area ADC the SAR architecture was chosen.
Chapter 3

5-bit SAR ADC using an inverter as comparator

This chapter provides an overview on the implementation of a 5-bit SAR ADC using a CMOS inverter as comparator. Its architecture and all its essential components are presented, as the approached solution for each step of its implementation. This ADC is composed by three main blocks; capacitor array, SAR control logic and comparator. In section 3.1 the capacitor array function is explained. In section 3.2 the switch implementation is covered. In section 3.3 an overview on the SAR control logic design flow is described and in section 3.4 the comparator is presented. This chapter concludes with the presentation of simulations and results.
3.1 Implemented SAR ADC

The architecture of the implemented SAR ADC has the following blocks:

- Capacitor array;
- SAR control (SAR control logic, S&H block, non-overlap block and bit register);
- Comparator.

Figure 3.1 shows the implemented architecture.

![Figure 3.1 Implemented SAR ADC block.](image)

The ADC uses a clock signal of 10MHz and issues 5 bits output in parallel corresponding to one sample every 700 ns. It has an input voltage range of 1 V, making the voltage step corresponding to one LSB equal to $V_{LSB} = \frac{1V}{2^5} = 31.25$ mV. The end of conversion is signaled by setting the 'eoc' flag. The main blocks will be described in the following sections.

3.2 Capacitor Array

SAR ADCs can be classified according to the DAC they use. The choice of this block is crucial since the converter linearity is heavily affected by its performance. The converter linearity depends on the code transition levels, which depends directly from the DACs reference voltage. The Charge-Redistribution or Switched Capacitor (SC) SAR ADC is by far the most popular variation of a SAR ADC. It combines both S&H and DAC in a single building block.
Figure 3.2 shows the 5-bit SAR ADC implemented in this work, with its binary-weighted parallel capacitor array, where each capacitor’s bottom plate is connected to a switch [10].

![Figure 3.2 SAR ADC with the capacitor array in the sample phase.](image)

Voltages $V_{REFF}$ and $V_{REFN}$ define the input voltage range of the ADC, $V_{FS} = V_{REFF} - V_{REFN}$, $V_C$ is the comparator reference voltage, which in turn is the mid-scale of the ADC input voltage range (Figure 3.3) and $V_A$ is the analog input signal to be converted. The capacitor $C_T$ is used in order to terminate the capacitor array, making the total capacitance equal to $2C$.

![Figure 3.3 DAC voltage references according to the work specifications.](image)

The operation of the SC SAR ADC can be divided into three distinct phases, regarding the capacitor array switch positions; sample phase, hold phase and charge redistribution phase. In the sample phase (Figure 3.2), switch $S_B$ is closed, connecting the top plate of all capacitors to $V_C$, setting $V_X = V_C$, while switch $S_A$ is connected to the analog input voltage, $V_A$. The voltage $V_A$ appears across the total capacitance $2C$, resulting in a stored charge $\Delta Q = 2CV_A$. During this phase, a sample of $V_A$ is taken and a proportional amount of charge is stored on the capacitor array.
During the Hold phase (Figure 3.4), switch $S_b$ is opened and switches $S_1$ to $S_3$ and $S_7$ are moved to $V_{REFN}$ side. The top plate of the capacitor array is open-circuited while their bottom plates are connected to $V_{REFN}$. Since no discharge path has been provided, the capacitor charges remain equal to the charge stored in the previous phase. It follows that the voltage at the top plate must become $-V_A - V_c$. Note that during the hold phase, switch $S_A$ moves to $V_{REFP}$ side in preparation for the charge-redistribution phase.

The last phase of the conversion process, charge redistribution phase (Figure 3.5), starts by moving switch $S_1$ to the $V_{REFP}$ side.

The circuit then consists of a capacitor $C$ connected to $V_{REFP}$, and a total capacitance of value $C$ connected to $V_{REFN}$. This capacitor divisor is represented in Figure 3.6.
Figure 3.6 Charge redistribution phase step 1 equivalent circuit

This capacitor divider causes a voltage increment of $V_{FS}/2$ to appear on the top plates ($V_x$). Now if, $V_A$ is greater than $V_{FS}/2$ the net voltage at the top plate will be smaller than $V_C$, which means that $S_1$ will remain connected to $V_{REFP}$ as the logic control moves on to switch $S_2$. On the other hand, if $V_A$ was smaller than $V_{FS}/2$ then the net voltage at the top plate would become bigger than $V_C$. The comparator will detect this situation and trigger the control logic to connect $S_1$ to $V_{REFN}$ position and then to move on to $S_2$. Then switch $S_2$ is connected to $V_{REFP}$ which causes a voltage increment of $V_{FS}/4$ to appear on the top plate. If the resulting voltage is still smaller than, $V_C$, $S_2$ is left in its new position; otherwise, $S_2$ is return to its $V_{REFN}$ position. Next the logic tries switch $S_3$ and so on until all the bit switches $S_1$ to $S_5$ have been tested.

It can be seen that during the charge-redistribution phase the voltage on the top plate will be incrementally reduced to $V_C$. The voltage at the top plate of the capacitors $V_x$ is a very important voltage node in this circuit, since it is also the voltage input of the comparator, and any inaccuracy of its value will strongly affect the ADC performance.

The display of the switches at the end of this phase represents the output digital word. A switch connected to $V_{REFN}$ side indicates a logic low ‘0’ and a switch connected to $V_{REFP}$ side indicates a logic high ‘1’.

To find the capacitors size the work specifications were considered. To improve capacitor matching, noise immunity and consequently the ADC accuracy, the capacitors values should be as large as possible. On the other hand, to decrease power consumption and increase speed, the capacitor value should be as small as possible. Also the minimization of the $kT/C$ noise was considered.

\[
V_{\text{noise}}^2 = \frac{kT}{C} \quad (3.1)
\]

To minimize thermal noise then.

\[
C > k.T.2^{2(N+P)} \quad (3.2)
\]

Where $N$ is the converter resolution and $P$ is the fraction of $V_{\text{LSB}}$ permitted by the thermal noise.
Assuming the fraction of $V_{\text{LSB}}$ permitted to the thermal noise is $\frac{1}{8} V_{\text{LSB}}$, $P=3$, meaning that $C > 0.27 \text{fF}$.

Since the minimum capacitor value available in this technology is approximately $10 \text{ fF}$, the thermal noise is much smaller and can be disregarded. In order to avoid minimum dimensions and at the same time reduce the influence of the capacitor input capacitance, it is considered a $C_{\text{min}} = 20 \text{ fF}$.

### 3.3 Switches

Voltages to be transmitted in the circuit vary from GND to $V_{\text{DD}}$. In order to guarantee conduction over this voltage range, CMOS transmission gates \[11\] are used (Figure 3.7). To reach a symmetric conduction characteristic, PMOS transistors are approximately 3 times wider than NMOS transistors in all transmission gates. All analog switches have minimum channel length, in order to reduce the equivalent on-resistance.

The switch may be seen as a voltage controlled resistor, $R_{\text{ON}}$ (Figure 3.8). When the control signal is a logic low ‘0’ the transmission gate (between A and B) is high (ideally infinite), and the circuit behaves like an open circuit (OFF mode). When the control signal is a logic high ‘1’ the resistance in the transmission gate is low (ON mode), and the current flows between A and B [ ].

![Figure 3.7 Transmission gate. Left: Ideal switch representation. Right: Transistor level switch representation.](image)

![Figure 3.8 Ron representation. Left: ON and OFF mode representation. Right: Ron variation versus Control signal](image)
The switch to implement in the capacitor array is represented in Figure 3.9.

![Implemented switch diagram](image)

Figure 3.9 Implemented switch. Left: Ideal switch representation. Right: Transistor level switch.

In order to make an alternate connection between two nodes (A-B and A-C and vice-versa), two complementary transmission gates in parallel were used. To achieve a correct functionality of the switch (minimizing a source of charge injection errors [12]), the signal controlling the switch should have special properties. The signal controlling each transmission gate must be overlapped (Figure 3.10) and the signal controlling each pair of transistors has to be complementary. A control signal with those properties, guarantees that when a switch alternates between positions, the switch that is ON goes OFF before its complementary goes ON. A waveform of the overlapped control signal is represented in Figure 3.10.

![Overlapped control signals](image)

Figure 3.10 Overlapped switch control signals.
The implementation of switch comports charge injection errors that should be minimized in order to achieve a better performance. The charge injection effect is shown in Figure 3.11.

![Figure 3.11 Charge injection effect (based on [13]).](image)

In order to minimize these effects, the common method used is implementation of the dummy transistor (Figure 3.12). This is a transistor that is shortcut, and placed next \( T_1 \) and controlled by a complementary signal. The dummy transistor will absorb the charge injection caused by switch \( T_1 \).

![Figure 3.12 Dummy circuit used to minimize charge injection (base on [13]).](image)

### 3.4 Control Logic Block

To approach the solution for the ADC’s logic blocks the hardware description language VHDL\(^1\) was chosen. The code developed to project the SAR logic block is listed in Appendix A, and a tutorial on how to import the VHDL code to Cadence environment is listed in Appendix B.

#### 3.4.1 SAR logic block

A flow chart of the SAR ADC logic block functionality is presented in Figure 3.13.

\(^1\) VHDL- (“VHSIC Hardware Description Language”, VHSIC “Very High Speed Integrated Circuits”)
A detailed view of the control logic block is represented in Figure 3.14.

This block implements the binary search algorithm. Its inputs are:

- **Clock**: provides the working frequency and synchronization of the circuit.
- **Reset**: commands the system's reset by setting its value to a logic ‘1’.
- **Enable**: allows the system to function as long as its value is a logic ‘1’.
- **Comp**: is the comparator response to the top plate of the capacitor array ($V_X$). Its response (logic low ‘0’ or logic high ‘1’) allows the logic block to implement the binary search algorithm.

The outputs of the logic block are:

- $S_A$, $S_B$, $S_T$ and $S_1$ to $S_5$: responsible for commanding every switch in the capacitor array according to the binary search algorithm. Notice that, due to switch implementations and circuit synchronization requirements, the signals must go through the S&H and non-overlapping block to achieve the correct functionality (both blocks will be explained in detail in the following sections).

- **eoc**: is a flag for the end of conversion. When its value is a logic low ‘0’, the system is busy, meaning it is in the middle of a conversion cycle. Otherwise, the system has reached the end of conversion cycle and the digital word converted is available at the bit register output.

- **sh**: is a specific control signal for the S&H block, that forces the sample and hold phases of conversion to occur in one clock cycle. This behavior will be explained in the next section.

### 3.4.2 Sample & hold block

This block applies a simple solution, to implement the sample and hold phase of the conversion, into a single clock period ($1/2 T_{CLK}$ sample phase, $1/2 T_{CLK}$ hold phase). For a better understanding, the signals waveforms are represented in Figure 3.15. The outputs of the control logic $S_A$, $S_B$, $S_T$, $S_1$ to $S_5$ and **sh** are directly connected to the sample and hold block. The combinatory logic present in this block guarantees that the sample and hold phases specific control signals are delivered to the capacitor array. When the signal **sh** is a logic high, the period of time specified in Figure 3.15 with the number 1, forces all the switch signals to the logic high value (‘1’) (commanding the DAC to go into the sample phase – see capacitor array section). In the period of time specified as 2, all the switch signals are forced to a logic ‘0’ (commanding the DAC to go into the hold phase – see capacitor array section). When the signal **sh** is a logic low, all the signals pass through the S&H block unaffected. The sample & hold circuit is listed in appendix C.
3.4.3 Non-overlapping block

This block is responsible for providing the non-overlapped signals to each switch (except $S_B$), and the overall switch synchronization to the capacitor array. As described earlier, the control of each switch requires a non-overlapped signal between both transmission gates. Although, another special care has to be taken regarding switch timing control of the capacitor array, which is $S_A$ has to be the last switch changing and $S_B$ the first in order to keep the correct functioning of the capacitor array. The block of switches $S_1$ to $S_5$ and $S_T$ are in between $S_B$ and $S_A$ action. The non-overlapping block circuit is listed in appendix C.

3.4.4 Bit register

This block provides the output synchronization and is commanded by the signal ‘eoc’. The output signals of the register are busy when a conversion cycle is occurring (eoc=0) and in every end of conversion the signal is active (eoc=1) indicating that the conversion has finished and the result is available in the output pins, its representation is shown in Figure 3.16.
3.5 Comparator

The comparators can be divided into two broad classes, with or without a latch circuit. Regarding the power consumption there are those that have static power consumption and those, which are often called dynamic latches, without static consumption [12]. In this thesis the main goal is to achieve a simple and lower power ADC. It was chosen the CMOS Inverter to be used as comparator [14], since it is a dynamic comparator that does not requires voltage references and its structure is shown in Figure 3.17.

The comparator structure consists in two cascaded CMOS inverters as shown in figure 3.17. The analog input signal quantization level is set in the first stage by changing the voltage transfer curve (VTC) via transistor sizing. Since the transistor channel length, $L$, is more effective than the channel width, $W$, in controlling the performance $L$ is kept constant and with minimum value while $W$ is changed during the design process. The second inverter stage is used for increased gain and logic level inversion so that the circuit behaves as an internally set comparator circuit. It can be shown that
the $V_{th}$ point on the VTC of a CMOS inverter, which is shown in Figure 3.17, can approximately be given by (3.3) [14].

$$V_{th} = \frac{V_{dd} - |V_{tp}| + V_{tn} \sqrt{\frac{K_n}{K_p}}}{1 + \sqrt{\frac{K_n}{K_p}}} [V]$$ (3.3)

Where $V_{tn}$ and $V_{tp}$ are the threshold voltages for NMOS and PMOS devices, respectively with:

$$K_n = \left( \frac{W}{L} \right)_n \mu_n C_{ox}$$ (3.4)

$$K_p = \left( \frac{W}{L} \right)_p \mu_p C_{ox}$$ (3.5)

This type of comparator presents attractive advantages for these work specifications. It has no static power consumption, excludes the need of reference voltages to establish the quantization level (since this is made via transistor sizing), and it is a low area solution.

### 3.6 Simulations and Results

#### 3.6.1 Ramp response

In Figure 3.16, the output of the ADC to a ramp input waveform is presented.

![Figure 3.18 ADC output response to a ramp waveform input.](image-url)
With this simulation it can be confirmed the correct functionality of the ADC. The current consumption is approximately 350μA, which corresponds to a power consumption of 1.1 mW.

3.6.2 Dynamic performance

To measure the dynamic performance of this ADC, the FFT of the output codes was calculated using coherent sampling [15]. Coherent sampling refers to a relationship between input frequency, \( f_{in} \) sampling frequency, \( f_s \), number of cycles in the sampled set (which has to be an integer), \( N_{cycles} \) and number of samples (which has to be a power of 2), \( M_{samples} \). With coherent sampling, it is assured that the signal power in an FFT is contained within one FFT bin, assuming single input frequency. The number of cycles, \( N_{cycles} \), should be chosen carefully. Typically it’s an odd prime number.

The condition of coherent sampling is given by:

\[
\frac{f_{in}}{f_s} = \frac{N_{cycles}}{M_{samples}}
\]  

(3.1)

and for this ADC case it’s considered:

\[
f_s = \frac{10\text{MHz}}{7} = 1.458571 \text{ MHz}
\]

\[
N_{cycles} = 7
\]

\[
M_{samples} = 2^{12}
\]

\[
f_{in} = 2.44141 \text{ KHz}
\]

With \( f_{in} \), frequency defined, it’s now necessary to characterize the amplitude and shape of the signal. Since all codes of the AD must be covered, the amplitude of \( f_{in} \) should cover the full scale range of the ADC, therefore, the shape and amplitude chosen: sinusoidal waveform, with 1.65 V offset voltage and 1V amplitude peak-to-peak. The result of the simulation is presented in Figure 3.19.
The output codes of the simulation were sampled every 700 ns, once all data was collected the FFT was calculated and its partially represented in Figure 3.120.

Figure 3.19 ADC output to a 7 period sinusoidal waveform

Figure 3.20 FFT of the codes output (without DC component)
After the FFT analysis the dynamic parameters of the ADC were calculated: Table 4.1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENOB</td>
<td>4.69 dB</td>
</tr>
<tr>
<td>SINAD</td>
<td>30 dB</td>
</tr>
<tr>
<td>SFDR</td>
<td>41.4 dB</td>
</tr>
</tbody>
</table>

3.6.3 Corner analysis

A corner analysis was made for the ADC using ramp waveform input. The corners considered are presented in Table 3.2.
Table 3.2 Corner analysis with transistor conditions variations

<table>
<thead>
<tr>
<th>Corner</th>
<th>NMOS Condition</th>
<th>PMOS Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Worst case power (WP)</td>
<td>FAST</td>
<td>FAST</td>
</tr>
<tr>
<td>Worst case speed (WS)</td>
<td>SLOW</td>
<td>SLOW</td>
</tr>
<tr>
<td>Worst case one (WO)</td>
<td>FAST</td>
<td>SLOW</td>
</tr>
<tr>
<td>Worst case zero (WZ)</td>
<td>SLOW</td>
<td>FAST</td>
</tr>
</tbody>
</table>

The corner analysis indicates that the ADC code steps are no longer equal. This means that the comparators quantization level changed. This effect is more visible in WO and WZ corners than in the others.

3.6.4 Monte Carlo analysis

The output of the ADC of a 5 iterations Monte Carlo analysis is represented in Figure 3.22.
The analysis of the ADC output codes for each iteration indicates that the process variations introduced in the circuit by Monte Carlo analysis heavily affect the linearity of the converter. The worst iteration output code has deviations in a max of 2 LSB. Like the corner analysis this is also due to the change of the quantization level of the Comparator.

### 3.6.5 Layout

To estimate the area of the implemented ADC, the digital and analog part of the circuit were obtained separately. Using the software SoC Encounter\(^2\), it was possible to synthesize the digital control block of the circuit. The layout of the digital blocks of the implemented ADC is presented in Figure 3.23.

\(\text{Figure 3.23 SAR control logic layout.}\)

\(^2\) Soc-Encounter: Design synthesis tool, RTL-to-GDSII
The estimated area achieved for this digital block is approximately 0.01822 \( \text{mm}^2 \).

To estimate the area of the implemented ADC analog blocks, the tool Virtuoso (Layout XL) from Cadence was used. Figure 3.24, represents the area filled by the circuit components that form the analog blocks (routing is not included).

![Figure 3.24 Analog components disposed for area consideration.](image)

The estimated area achieved for this block is approximately 0.05064 \( \text{mm}^2 \). An estimate of the ADC core area is 0.06886 \( \text{mm}^2 \).
Chapter 4

5-bit 2-bit/step SAR ADC using inverters as comparators

This chapter presents the implementation of a 5-bit 2-bit/step SAR ADC. Its global architecture and main changes regarding the conventional architecture are presented and explained. In section 4.1 the structure of the proposed ADC is presented. In chapter 4.2, both the control logic block and comparator block are explained. This chapter concludes with the presentation of simulation results and some considerations regarding the proposed architecture.
4.1 Proposed SAR-ADC

The architecture of the implemented 5-bit 2-bit/step SAR-ADC is shown in Figure 4.1.

The proposed architecture has the following blocks:

- Capacitor array;
- SAR control (SAR control logic, S&H block, non-overlap block and bit register);
- Comparator block 1;
- Comparator block 2;

Being a variation of the SAR ADC using an inverter as comparator, it has a main goal to reduce the \( N+2 \) clock cycles needed to complete a conversion in a conventional SAR ADC to \( N \) clock cycles conversion. Figure 4.2 exposes the improvement of the 5-bit 2-bit/step SAR ADC over the conventional 5-bit SAR ADC.

Figure 4.2 Designed ADCs conversion time diagram.

\[3 \text{ Assume the same capacitor array studied in the previous chapter.} \]
The ADC uses a clock signal of 10MHz and issues the 5 bits output in parallel corresponding to one sample every 500 ns. The signal \textit{ref\_control} commands the ADC to alternate between comparator blocks. The end of conversion is signaled by setting the ‘eoc’ flag.

To implement this architecture two major changes had to be made from the conventional architecture, which were, re-program the SAR control logic block and increase the number of comparators. These changes will be covered in the following section.

### 4.2 SAR Control Logic Block

A flow chart of the proposed SAR ADC logic block functionality is presented in Figure 4.4. The VHDL code of the control logic block is listed in Appendix A.

The main improvement of this ADC topology is the creation of multi-reference comparator blocks. With this new improvement it is possible to speed up the implementation of the binary search algorithm by a reasonable number of clock cycles. It is able to resolve up to 2-bits per step of conversion. Follows the explanation of the conversion process.

In the first step of conversion the control logic signal \textit{ref\_control} is a logic ‘0’ choosing the comparator block 1 to be active while the control block 2 is inactive. The voltage references for each comparator of the comparator block are $V_c - V_{FS}/4$, $V_c$, and $V_c + V_{FS}/4$. These increments of $V_{FS}/4$ allow the control logic to evaluate the combined comparators responses and resolves bits 5 and 4 of the output digital word as shown in the flowchart of Figure 4.4. It then sets the \textit{ref\_control} signal to a logic ‘1’ meaning that in the next cycle the comparator block 1 will be inactive while the comparator block 2 is active. The comparator block 2 has reference voltages of $V_c - V_{FS}/16$, $V_c$, and $V_c + V_{FS}/16$. In this second step, the control logic evaluates the combined comparators response, once the response is obtained the control logic resolves bits 3 and 2 of the output digital word. The conversion process is finished by making one last comparison with the comparator with reference voltage of $V_c$, resolving the least significant bit, bit 1.
Figure 4.3 5-bit 2-bit/step SAR ADC control logic block flowchart.
The generic comparator block for the 2-bit/step SAR ADC is represented in Figure 4.3.

![Comparator Block Diagram](image)

Figure 4.4 5–bit 2-bit/step SAR ADC comparator block.

The implementation of the algorithm presented in the previous section requires a precise transistor sizing for the comparator blocks. Using Equation 3.3 the transistor sizes where obtained and are presented in Table 4.1 and Table 4.2.

Table 4.1 5-bit 2-bit/step SAR ADC comparator block 1 transistor sizing.

<table>
<thead>
<tr>
<th>Comparator</th>
<th>PMOS</th>
<th>NMOS</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comparator 1</td>
<td>$W = 3.15\mu m$</td>
<td>$W = 1\mu m$</td>
<td>$V_c$</td>
</tr>
<tr>
<td></td>
<td>$L = 0.35\mu m$</td>
<td>$L = 0.35\mu m$</td>
<td></td>
</tr>
<tr>
<td>Comparator 2</td>
<td>$W = 7\mu m$</td>
<td>$W = 1\mu m$</td>
<td>$V_c + V_{FS}/4$</td>
</tr>
<tr>
<td></td>
<td>$L = 0.35\mu m$</td>
<td>$L = 0.35\mu m$</td>
<td></td>
</tr>
<tr>
<td>Comparator 3</td>
<td>$W = 1.5\mu m$</td>
<td>$W = 1\mu m$</td>
<td>$V_c - V_{FS}/4$</td>
</tr>
<tr>
<td></td>
<td>$L = 0.35\mu m$</td>
<td>$L = 0.35\mu m$</td>
<td></td>
</tr>
</tbody>
</table>
Table 4.2 5-bit 2-bit/step SAR ADC comparator block 2 transistor sizing.

<table>
<thead>
<tr>
<th>Comparator</th>
<th>PMOS</th>
<th>NMOS</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comparator 1</td>
<td>W = 3.15\mu m</td>
<td>W = 1\mu m</td>
<td>( V_c )</td>
</tr>
<tr>
<td></td>
<td>L = 0.35\mu m</td>
<td>L = 0.35\mu m</td>
<td></td>
</tr>
<tr>
<td>Comparator 2</td>
<td>W = 3.8\mu m</td>
<td>W = 1\mu m</td>
<td>( V_c + V_{FS}/16 )</td>
</tr>
<tr>
<td></td>
<td>L = 0.35\mu m</td>
<td>L = 0.35\mu m</td>
<td></td>
</tr>
<tr>
<td>Comparator 3</td>
<td>W = 2.6\mu m</td>
<td>W = 1\mu m</td>
<td>( V_c - V_{FS}/16 )</td>
</tr>
<tr>
<td></td>
<td>L = 0.35\mu m</td>
<td>L = 0.35\mu m</td>
<td></td>
</tr>
</tbody>
</table>

4.3 Simulations and Results

Figure 4.3 presents ADC response to a ramp input waveform.

Figure 4.5 5-bit 2-bit/step SAR ADC ramp response.

The time domain simulation of the proposed SAR ADC demonstrates the feasibility of the implementation. The power consumption of this ADC was measured and it’s equal to 3 mW.
Corner Analysis and Monte Carlo simulations were not performed, but considering the results obtained for the 5-bit SAR ADC in chapter 3, it is excepted that this circuit is also very sensitive to process variations and it will require also a redimensioning of the comparators. In terms of area occupation, the main addition was the extra comparators which have small dimensions, so the estimated area is approximately \(0.07 \, \text{mm}^2\), about the same has the first ADC designed.
Chapter 5

Conclusions

This chapter point out the main conclusion of this thesis, as well as some suggestion for future work.
5.1 Summary

In the present thesis the design and simulation of two 5-bit ADC with SAR topology was performed.

In Chapter 2, general concepts of ADCs were exposed. The explanation of some popular architecture and a brief study regarding tradeoffs between ADC architecture was also presented. The SAR ADC was chosen over the range of ADCs available, because it was the one that better met the required specifications of low power, low resolution and low frequency.

In Chapter 3, the implementation of a 5-bit SAR ADC using a switched capacitor array that performs both DAC and S&H function in a single building block and a cascade CMOS inverter as comparator was presented. The capacitor sizing was determined regarding, power consumption, speed and thermal noise. Transmission gates were used to implement the switches controlling the capacitor array. The control logic block of the SAR ADC was design using VHDL approach. A sample and hold and a non-overlapping blocks that served as complementary logic to complete the correct functionality of the ADC were designed. The CMOS inverter was used as a comparator considering its advantages: no static power consumption, independence of voltage references and low area. Its voltage transfer curve was adjusted by simulation. The functionality of the ADC was proved by a time domain analysis. This ADC has a power consumption of 1.1mW and a estimated area of 0.06886 mm² which proves adequate for the proposed application. The dynamic parameters obtained were a SINAD of 30 dB, ENOB of 4.69 bit and a SFDR of 41.4 dB. The ADC was submitted to process corner and Monte Carlo analysis revealing that the comparator (CMOS inverter) correct functionality is very sensitive by process variations and should be redesigned.

In chapter 4 a 5-bit 2-bit/step SAR ADC based on inverters as comparators was proposed. This architecture is an alternative to the previous chapter ADC, focusing in reducing the number of clock cycles to complete a conversion. In this design, it reduced the number of cycles from 7 to 5. The control logic block of the architecture was again implemented using VHDL. A time domain analysis confirms the feasibility of this implementation. This ADC power consumption was approximately 3 mW, higher than the single bit SAR ADC, due to the increase in the number comparators, although this power consumption may be reduced with some future work considerations.

5.2 Future work

The future work suggestions depend on the line of research that will be followed. For the application presented here the major improvement would be to study techniques in order to reduce the CMOS Inverter used as comparator sensitivity to process variation, keeping in mind that these techniques should have very low power consumption and extra circuitry in order to maintain the power
consumption at a very low level.

A second suggestion to improve this design is to use the work produced in this thesis, to increase the ADC specifications and use the design flow developed to produce digital blocks.
References


Appendix A – VHDL Code

In this appendix, the VHDL code of the 5-bit SAR ADC Logic control block is listed.

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity sar_logic is
Port ( enable : in STD_LOGIC;
      comp : in STD_LOGIC;
      clock : in STD_LOGIC;
      reset : in STD_LOGIC;
      sA : out STD_LOGIC;
      sB : out STD_LOGIC;
      s1 : out STD_LOGIC;
      s2 : out STD_LOGIC;
      s3 : out STD_LOGIC;
      s4 : out STD_LOGIC;
      s5 : out STD_LOGIC;
      s6 : out STD_LOGIC;
      eoc : out STD_LOGIC;
      sh : out STD_LOGIC);
end sar_logic;

architecture Behavioral of sar_logic is

type statetype is (state0, state1, state2, state3, state4, state5, state6);

signal state : statetype;

begin

-- Logica ----
-- Logica ----
-- process (clock,reset,enable)
begin

if reset = '1' then
  state <= state0;
  sA <= '1';
  sB <= '1';
  s1 <= '1';
  s2 <= '1';
  s3 <= '1';
  s4 <= '1';
  s5 <= '1';
  s6 <= '1';
  sh <= '0';
  eoc <= '0';

elsif (clock'event and clock='1' and enable='1') then
  case state is
    when state0 => state <= state1;
    sA <= '1';
    sB <= '1';
    s1 <= '1';
  when state1 => state <= state2;
    sA <= '1';
    sB <= '1';
    s1 <= '1';
  when state2 => state <= state3;
    sA <= '1';
    sB <= '1';
    s1 <= '1';
  when state3 => state <= state4;
    sA <= '1';
    sB <= '1';
    s1 <= '1';
  when state4 => state <= state5;
    sA <= '1';
    sB <= '1';
    s1 <= '1';
  when state5 => state <= state6;
    sA <= '1';
    sB <= '1';
    s1 <= '1';
  when state6 => state <= state0;
    sA <= '1';
    sB <= '1';
    s1 <= '1';
  when others =>
end case;
end process;
end Behavioral;
s2 <= '1';
s3 <= '1';
s4 <= '1';
s5 <= '1';
s6 <= '1';
sh <= '1';
eoc <= '0';
when state1 => state <= state2;
  sA <= '0';
  sB <= '0';
  s1 <= '1';
  s2 <= '0';
  s3 <= '0';
  s4 <= '0';
  s5 <= '0';
  s6 <= '0';
  sh <= '0';
when state2 => state <= state3;
  if comp='1' then
    s1 <='0';
    s2 <='1';
    elsif comp='0' then
      s2 <='1';
    end if;
when state3 => state <= state4;
  if comp='1' then
    s2 <='0';
    s3 <='1';
    elsif comp='0' then
      s3 <='1';
    end if;
when state4 => state <= state5;
  if comp='1' then
    s3 <='0';
    s4 <='1';
    elsif comp='0' then
      s4 <='1';
    end if;
when state5 => state <= state6;
  if comp='1' then
    s4 <='0';
    s5 <='1';
    elsif comp='0' then
      s5 <='1';
    end if;
when state6 => state <= state0;
  if comp='1' then
    s5 <='0';
    elsif comp='0' then
      s5 <='1';
eoc <= '1';
end case;
if enable = '0' then
  sA <= '1';
  sB <= '1';
  s1 <= '1';
In this appendix, the VHDL code of the 5-bit 2-bit/step SAR ADC Logic control block is listed.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
--- Uncomment the following library declaration if instantiating
--- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
entity sar_logic is
Port ( enable : in STD_LOGIC;
comp1 : in STD_LOGIC;
comp2 : in STD_LOGIC;
comp3 : in STD_LOGIC;
clock : in STD_LOGIC;
reset : in STD_LOGIC;
sA : out STD_LOGIC;
sB : out STD_LOGIC;
s1 : out STD_LOGIC;
s2 : out STD_LOGIC;
s3 : out STD_LOGIC;
s4 : out STD_LOGIC;
s5 : out STD_LOGIC;
s6 : out STD_LOGIC;
eoc : out STD_LOGIC;
sh : out STD_LOGIC;
ref_control_out : out STD_LOGIC);
end sar_logic;
architecture Behavioral of sar_logic is
type statetype is (state0, state1, state2, state3, state4);
signal state : statetype;
begin
--%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
--%% -- Logica -- %%
--%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
logica : process (clock,reset,enable)
begin
if reset = '1' then
state <= state0;
sA <= '1';
sB <= '1';
s1 <= '1';
s2 <= '1';
s3 <= '1';
s4 <= '1';
s5 <= '1';
s6 <= '1';
sh <= '0';
eoc <= '0';
end if;
end process;
end Behavioral;
```
s4 <= '1';
s5 <= '1';
s6 <= '1';
sh <= '0';
eoc <= '0';
ref_control_out <= '0';
elif (clock'event and clock='1' and enable='1') then
  case state is
    when state0 => state <= state1;
      sA <= '1';
      sB <= '1';
      s1 <= '1';
      s2 <= '1';
      s3 <= '1';
      s4 <= '1';
      s5 <= '1';
      s6 <= '1';
      sh <= '1';
      eoc <= '0';
      ref_control_out <= '0';
    when state1 => state <= state2;
      sA <= '0';
      sB <= '0';
      s1 <= '1';
      s2 <= '0';
      s3 <= '0';
      s4 <= '0';
      s5 <= '0';
      s6 <= '0';
      1, sh <= '0';
    when state2 => state <= state3;
      if (compl='1' and comp2='1' and comp3='1') then
        s1 <= '0';
        s2 <= '0';
        s3 <= '1';
      elsif (compl='1' and comp2='1' and comp3='0') then
        s1 <= '0';
        s2 <= '1';
        s3 <= '1';
      elsif (compl='1' and comp2='0' and comp3='0') then
        s1 <= '1';
        s2 <= '0';
        s3 <= '1';
      elsif (compl='0' and comp2='0' and comp3='0') then
        s1 <= '1';
        s2 <= '1';
        s3 <= '1';
      end if;
    ref_control_out <= '1';
    when state3 => state <= state4;
      if (compl='1' and comp2='1' and comp3='1') then
        s3 <= '0';
        s4 <= '0';
        s5 <= '1';
      elsif (compl='1' and comp2='1' and comp3='0') then
        s3 <= '0';
        s4 <= '1';
        s5 <= '1';
  end case;
elsif (comp1='1' and comp2='0' and comp3='0') then
  s3 <= '1';
  s4 <= '0';
  s5 <= '1';
elsif (comp1='0' and comp2='0' and comp3='0') then
  s3 <= '1';
  s4 <= '1';
  s5 <= '1';
end if;
when state4 => state <= state0;
  if (comp2='1') then
    s5 <= '0';
  elsif (comp2='0') then
    s5 <= '1';
  end if;
  eoc <= '1';
  ref_control_out <= '0';
end case;
if enable = '0' then
  sA <= '1';
  sB <= '1';
  s1 <= '1';
  s2 <= '1';
  s3 <= '1';
  s4 <= '1';
  s5 <= '1';
  s6 <= '1';
  sh <= '0';
  eoc <= '0';
  ref_control_out <= '0';
end if;
end process;
end Behavioral;
Appendix B - Tutorial

Creating Cadence Digital Cells from VHDL code

1. Run Synopsys’s Design Vision.Source through the command line. (e.g. disk/soft/Start/synopsys.init )
   ➢ Design_vision

2. In the menu File, choose the option Read. A Read Design window will pop up, where you can select the .VHD file to import.

3. In menu Design, select option Compile Design, all default options are fine. Click OK.
4. In menu File, select option Save Design As. Change the field Format and save your design in Verilog(.v) extension.
Cadence Environment

To import the Verilog file to Cadence follow these steps.

1. Open Library Manager.
   - Create a new library. (In menu File -> New -> Library)
   - Name the new library (Name: … -> OK)
   - When the pop up window appears select option:
     o Attach to an exiting techfile -> OK
     o Attach Design Library to Technology File:
       ▪ Technologic Library : TECH_C35B4 -> OK

2. Open icfb window.

   In file menu, select option Import -> Verilog.

3. A Verilog in dialog window will pop up. Fill in each step carefully, and both schematic and functional view of the Verilog file will appear in the library created earlier. The following fields of the Verilog in dialog windows are necessary to successfully import the code:
   a. Target Library Name. Mandatory
   b. Reference Libraries. Mandatory
   c. Import Structural Modules As: schematic and functional. Mandatory
   d. Verilog Cell Modules -> Import. Mandatory
   e. Power Net Names. Mandatory
### Verilog Compiler Configuration

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>File Filter Name</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Target Library Name</strong></td>
<td>tempdir</td>
</tr>
<tr>
<td><strong>Reference Libraries</strong></td>
<td>sample basic CORELIB</td>
</tr>
<tr>
<td><strong>Verilog Files To Import</strong></td>
<td>/home/ijpg/sys/b03.v</td>
</tr>
<tr>
<td><strong>-f Options</strong></td>
<td></td>
</tr>
<tr>
<td><strong>-v Options</strong></td>
<td></td>
</tr>
<tr>
<td><strong>-y Options</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Library Extension</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Library Pre-Compilation Options</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Pre Compiled Verilog Library</strong></td>
<td></td>
</tr>
<tr>
<td><strong>HDL View Name</strong></td>
<td>hdl</td>
</tr>
<tr>
<td><strong>Target Compile Library Name</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Compile Verilog Library Only</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Ignore Modules File</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Import Structural Modules As</strong></td>
<td>schematic and functional</td>
</tr>
<tr>
<td><strong>Structural View Names</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Schematic</strong></td>
<td>schematic</td>
</tr>
<tr>
<td><strong>Netlist</strong></td>
<td>netlist</td>
</tr>
<tr>
<td><strong>Functional</strong></td>
<td>functional</td>
</tr>
<tr>
<td><strong>Symbol</strong></td>
<td>symbol</td>
</tr>
<tr>
<td><strong>Log File</strong></td>
<td>/verilogIn.log</td>
</tr>
<tr>
<td><strong>Work Area</strong></td>
<td>/tmp</td>
</tr>
<tr>
<td><strong>Name Map Table</strong></td>
<td>/verilogIn.map table</td>
</tr>
<tr>
<td><strong>Overwrite Existing Views</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Verilog Cell Modules</strong></td>
<td>Create Symbol Only, Import, Import As Functional</td>
</tr>
<tr>
<td><strong>Global Nets</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Power Net Name</strong></td>
<td>vcc</td>
</tr>
<tr>
<td><strong>Ground Net Name</strong></td>
<td>gnd</td>
</tr>
<tr>
<td><strong>Global Signals</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Schematic Generation Options >>**
Appendix C - Schematics

In this appendix, the main schematics of the two ADCs projected in this thesis is listed.

5-bit SAR ADC schematic.
5-bit SAR ADC capacitor array schematic.
5-bit SAR-ADC control logic block schematic.
5-bit SAR ADC sample and hold block.
5-bit SAR ADC sample and hold circuit.
5-bit SAR ADC non overlapping block.
5-bit SAR ADC non-overlapping circuit.
5-bit 2-bit/step SAR ADC.
5-bit 2-bit/step SAR ADC comparator block 1.
5-bit 2-bit/step SAR ADC comparator block 2.