

**DISSERTAÇÃO DE MESTRADO**

**PROJETO DE UMA NOVA ARQUITETURA  
PARA CONVERSORES DE SINAIS  
ANALÓGICOS PARA DIGITAIS**

**YURI CESAR ROSA DE TOLEDO**

Brasília, 20 de Março de 2015

**UNIVERSIDADE DE BRASÍLIA**

FACULDADE DE TECNOLOGIA

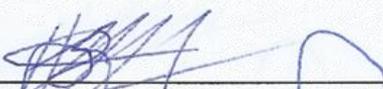
**UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

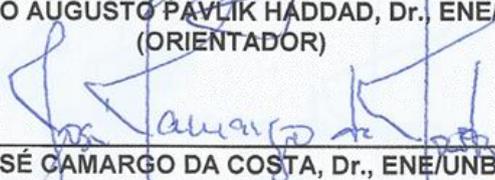
**PROJETO DE NOVA ARQUITETURA PARA CONVERSORES DE  
SINAIS ANALÓGICOS PARA DIGITAIS**

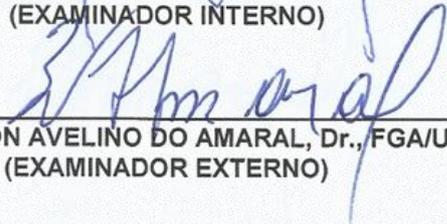
**YURI CESAR ROSA DE TOLEDO**

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE.

APROVADA POR:

  
\_\_\_\_\_  
SANDRO AUGUSTO PAVLIK HADDAD, Dr., ENE/UNB  
(ORIENTADOR)

  
\_\_\_\_\_  
JOSÉ CAMARGO DA COSTA, Dr., ENE/UNB  
(EXAMINADOR INTERNO)

  
\_\_\_\_\_  
WELLINGTON AVELINO DO AMARAL, Dr., FGA/UNB  
(EXAMINADOR EXTERNO)

Brasília, 20 de março de 2015.

**UNIVERSIDADE DE BRASÍLIA**  
**FACULDADE DE TECNOLOGIA**  
**DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**PROJETO DE UMA NOVA ARQUITETURA PARA  
CONVERSORES DE SINAIS ANALÓGICOS PARA  
DIGITAIS**

**YURI CESAR ROSA DE TOLEDO**

**ORIENTADOR: SANDRO AUGUSTO PAVLIK HADDAD**

**DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA**

**PUBLICAÇÃO: 590/2015 DM PGEA**

**BRASÍLIA/DF: MARÇO – 2015**



## FICHA CATALOGRÁFICA

Toledo, Yuri Cesar Rosa  
PROJETO DE UMA NOVA ARQUITETURA PARA CONVERSORES DE SINAIS  
ANALÓGICOS PARA DIGITAIS,

[Distrito Federal] 2015.

xxv, 111p, 210 X 297 mm (ENE/FT/UnB, Mestre, Dissertação de Mestrado) –  
Universidade de Brasília. Faculdade de Tecnologia.

1. Sistemas Analógicos

2. Conversores de Sinais

3. Circuitos Integrados

4. Processamento de Sinais

I. ENE/FT/UnB

II. Título (série)

## REFERÊNCIA BIBLIOGRÁFICA

TOLEDO, YURI C.R, (2015). PROJETO DE UMA NOVA ARQUITETURA PARA CONVERSORES DE SINAIS ANALÓGICOS PARA DIGITAIS Dissertação de Mestrado em Engenharia Elétrica, Publicação 590/2015 DM PGEA Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 111p.

## CESSÃO DE DIREITOS

**AUTOR: Yuri Cesar Rosa de Toledo.**

**TÍTULO: PROJETO DE UMA NOVA ARQUITETURA PARA CONVERSORES DE SINAIS ANALÓGICOS PARA DIGITAIS. GRAU: Mestre ANO: 2015**

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

---

Yuri Cesar Rosa de Toledo  
SQS 103 Bloco E Apartamento 102 – Asa Sul.  
70.342-050 Brasília – DF – Brasil.



## **Dedicatória**

*A minha mãe, a minha mulher e a minha tia*



## **AGRADECIMENTO**

Em primeiro lugar ao meu Orientador, pelo acompanhamento e pela paciência.

Aos meus familiares pela compreensão e auxílio durante o período de dedicação.



## RESUMO

O objetivo deste trabalho é o desenvolvimento e projeto de uma nova arquitetura para ADC o Conversor do Módulo da Diferença (CMD). O Conversor CMD proposto se caracteriza por ser assíncrono, sem oversample e com apenas um ciclo para conversão. Sua topologia não apresenta realimentação, não utiliza capacitores nem resistores, não utiliza circuito de Sample-and-Hold (S/H) e possui apenas uma referência para conversão.

O projeto foi desenvolvido em modo corrente, o que permitiu não utilizar Amp-Op no circuito destinado à conversão. E o projeto também possui um circuito digital com apenas uma porta XNOR por bit. Podemos destacar que os bit são convertidos simultaneamente por um processo contínuo e que seu código de conversão é estável por variar apenas um bit entre palavras digitais adjacentes.



## **ABSTRACT**

The objective of this work is the development and design of a new architecture for the ADC Difference Module Converter (CMD). The proposed converter CMD is characterized by being asynchronous without oversample and with only one cycle for conversion. Its topology doesn't presents feedback, does not use capacitors nor resistors, does not use Sample-and-hold circuit (S / H) and has only one reference to conversion.

The project was developed in current mode, which not use Op-amp in the circuit for the conversion. And the project also has a digital circuit with only one XNOR gate per bit. We can highlight that each bit is simultaneously converted by a continuous process and that your conversion code is stable, i.e., varying only one bit between adjacent digital words.



# SUMÁRIO

<b>1.</b>	<b>INTRODUÇÃO</b>	<b>1</b>
1.1.	Prefácio	1
1.2.	Contextualização Geral	1
1.3.	Contexto Local	4
1.4.	Motivação	4
1.5.	Objetivos	5
1.6.	Divisão da Dissertação	6
<b>2.</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>8</b>
2.1.	CONTEXTO GERAL	8
2.1.1.	FLASH	9
2.1.1.1.	Projeto do Flash	10
2.1.1.2.	Vantagens	11
2.1.1.2.1.	Todas as comparações ocorrem simultaneamente em paralelo;	11
2.1.1.2.2.	Baixo Ruído De Quantização Na Conversão	11
2.1.1.2.3.	Monotonicidade Garantida	12
2.1.1.3.	Desvantagens	12
2.1.1.3.1.	Capacitância de Entrada	12
2.1.1.3.2.	Área em SoC	13
2.1.1.3.3.	Resolução da Referência	13
2.1.1.3.4.	Consumo	13
2.1.1.3.5.	Interface Analógico Digital	13
2.1.2.	SAR	14
2.1.2.1.	Blocos Básicos do SAR	15
2.1.2.2.	Vantagens	15
2.1.2.2.1.	Pouco Aumento de Circuitos com a Resolução	16
2.1.2.2.2.	Estrutura de Projeto Simples	16
2.1.2.2.3.	Poucas Referências	16
2.1.2.3.	Desvantagens	16
2.1.2.3.1.	Número de Ciclos de Conversão	16
2.1.2.3.2.	Capacitores Grandes	17
2.1.3.	PIPELINE	17

2.1.3.1.	Projeto Tradicional	17
2.1.3.2.	Vantagens	18
2.1.3.2.1.	CONVERSÃO POR ETAPAS PARALELAS	18
2.1.3.3.	Desvantagens	18
2.1.3.3.1.	RESOLUÇÃO COMPROMETIDA	18
2.1.3.3.2.	INTERFACE OPERACIONAL	19
2.1.4.	SIGMA DELTA	19
2.1.4.1.	Projeto Tradicional	19
2.1.4.2.	Vantagens	19
2.1.4.2.1.	NOISE SHAPING	20
2.1.4.2.2.	BITS NÃO DEPENDEM DE VALORES CONVERTIDOS	20
2.1.4.3.	Desvantagens	20
2.1.4.3.1.	OVER SAMPLE	20
2.1.4.3.2.	USO DE CAPACITORES	20
2.2.	COMENTÁRIOS	21
<b>3.</b>	<b>METODOLOGIA</b>	<b>23</b>
3.1.	DEFINIÇÃO DE OBJETIVOS	23
3.2.	ESTUDOS DAS ARQUITETURAS	24
3.3.	DEFINIÇÃO DE UM MODELO COMPORTAMENTAL	24
3.4.	CARACTERIZAÇÃO LÓGICA DOS BLOCOS	25
3.5.	PROJETO	25
3.6.	ANÁLISE DOS RESULTADOS	26
<b>4.</b>	<b>PROJETO DO CONVERSOR CMD</b>	<b>28</b>
4.1.	DEFINIÇÕES INICIAIS E PROPRIEDADES CONSAGRADAS	28
4.2.	ALGORITMO DE CONVERSÃO DO CMD	29
4.3.	MODELO MATEMÁTICO	31
4.4.	COMPORTAMENTO BINÁRIO	33
4.4.1.	GERANDO OS BITS	34
4.4.2.	PARTICULARIDADES DOS VALORES BINÁRIOS DO CMD	36
4.4.2.1.	ESTABILIDADE ENTRE PALAVRAS DIGITAIS	36
4.4.2.2.	SIMETRIA	37
4.4.2.3.	CODIGO DE GRAY	38
4.4.3.	LÓGICA DO TRADUTOR	39
4.5.	MODELO EM SOFTWARE	42

4.5.1.	O SOFTWARE	42
<b>4.6.</b>	<b>BLOCOS BÁSICOS</b>	<b>45</b>
4.6.1.	Bloco CMD	46
4.6.1.1.	BY-PASS E SOMADOR	48
4.6.1.2.	Espelhos	52
4.6.1.3.	Comparador	54
4.6.1.4.	FLUXO DA CORRENTE	55
4.6.1.4.1.	COMPORTAMENTO ESPERADO	56
4.6.2.	Bloco PGA	57
4.6.2.1.	Resumo PGA	57
4.6.2.2.	Descrição dos Blocos Básicos do PGA	58
<b>4.7.</b>	<b>CIRCUITOS EM CADENCE</b>	<b>61</b>
4.7.1.	BLOCOS BÁSICOS	62
4.7.2.	BLOCO BÁSICO DO CMD	62
4.7.3.	OFFSET	67
4.7.4.	ERRO DE GANHO	69
4.7.5.	DNL – DIFFERENTIAL NONLINEARITY	70
4.7.6.	INL – INTEGRAL NONLINEARITY	72
4.7.7.	SNR – SIGNAL-TO-NOISE RATIO	75
4.7.8.	ENOB – EFFECTIVE NUMBER OF BITS	78
<b>5.</b>	<b>RESULTADOS E DISCUÇÕES</b>	<b>85</b>
5.1.	RESULTADOS DO PROJETO DE 10 BITS	85
5.2.	ULTRA LOW POWER	90
5.2.1.	TECNOLOGIA UTILIZADA	90
5.2.2.	TOPOLOGIA GERAL DO CONVERSOR	90
5.2.3.	RETIRADA DA MULTIPLICAÇÃO POR 2 NA SAÍDA	91
5.2.4.	CARACTERIZAÇÃO DO PROJETO	92
5.3.	CONSOLIDAÇÃO DA NOVA ARQUITETURA	94
5.3.1.	CARACTERÍSTICAS PRINCIPAIS	94
5.3.1.1.	O SAR	94
5.3.1.2.	O FLASH (PARALELO)	94
5.3.1.3.	O PIPELINE	94
5.3.1.4.	O SIGMA DELTA	95
5.3.1.5.	O CMD	95
5.3.1.5.1.	SIMETRIA	96
5.3.1.5.2.	CORRESPONDÊNCIA BIUNÍVOCA	96

5.3.1.5.3.	CONTINUIDADE DOS SINAIS DE CADA BIT CONVERTER O PADRÃO DO CMD NO PADRÃO MATEMÁTICO	97
5.3.1.5.4.		97
5.3.1.1.1.	TAXA DE AMOSTRAGEM	98
<b>5.4.</b>	<b>CONSIDERAÇÕES</b>	<b>99</b>
<b>6.</b>	<b>CONCLUSÃO</b>	<b>101</b>
	<b>REFERÊNCIAS</b>	<b>104</b>
	<b>APÊNDICE I BY-PASS</b>	<b>109</b>



## ÍNDICE DE IMAGENS

Imagem	1.1-1:	Tabela comparativa entre o SAR, o $\Sigma\Delta$ , o FLASH e o CMD.	1
Imagem	1.2-1:	Relação das publicações no IEEE [6]. Os conversores representam 20%.	2
Imagem	1.2-2:	Arquiteturas principais mapeadas pela energia por conversão em relação à taxa de amostras por segundo [1].	3
Imagem	2.1.2-1:	correspondência entre faixas analógicas e digitais para o Flash.	10
Imagem	2.1.2.1-1:	Blocos Básicos de um Conversor FLASH	10
Imagem	2.1.1.2.3-1:	Divisor de tensão para o conversor Flash	12
Imagem	2.1.3-1:	correspondência entre faixas analógicas e digitais para o SAR.	14
Imagem	2.1.2.1-1:	Blocos Básicos do SAR.	15
Imagem	2.1.3.1-1:	Retirada de: [22].	18
Imagem	4.3-1:	Interpretação gráfica do modelo matemático	31
Imagem	4.4.1-1:	Comportamento dos bits para as faixas analógicas	34
Imagem	4.4.1-2:	Dado analógico fluindo de Bloco a Bloco.	35
Imagem	4.4.2-1:	Comportamento dos bits para as faixas analógicas	36
Imagem	4.4.2.3-1:	Código Binário e Código de Gray. Retirada de [2].	38
Imagem	4.4.2.3-2:	Código Binário e Código de Gray.	38
Imagem	4.4.3-1:	Tabela Verdade para o circuito Tradutor	39
Imagem	4.4.3-2:	Circuito Não Simplificado de 5 bits para o Tradutor	40
Imagem	4.4.3-3:	Tabela Verdade da Porta Lógica XNOR	40
Imagem	4.4.3-4:	Resposta Lógica da associação em série de duas XNOR	41
Imagem	4.4.3-5:	Circuito Digital do Tradutor	41
Imagem	4.5.2-2:	Simulações em software dos conversores SAR e CMD em maior Frequência.	45
Imagem	4.5.2-3:	Simulações do Comportamento Binário.	45
Imagem	4.6-1:	Blocos Básicos do Conversor	46
Imagem	4.6.1-1:	Detalhamento do Bloco Básico do CMD	47
Imagem	4.6.1-2:	Arranjo de blocos básicos do CMD	48
Imagem	4.6.1.1-1:	Esquemático do circuito do By-Pass	49
Imagem	4.6.1.1-2:	Circuito Somador e Referência de Corrente	50
Imagem	4.6.1.1-3:	Curva de transferência do Somador	50
Imagem	4.6.1.1-4:	Seleção automática do By-Pass	51
Imagem	4.6.1.1-5:	Caminhos do By-Pass	52
Imagem	4.6.1.2-1:	Esquemático do circuito dos espelhos	53
Imagem	4.6.1.4-1:	Corrente de Entrada do 1º ao 4º Bloco	55
Imagem	4.6.1.4-1:	Fluxo de corrente entre os blocos básicos do bloco CMD	57
Imagem	4.6.2.2-1:	Diagrama dos espelhos de Corrente do PGA – SPLITTER	58
Imagem	4.6.2.2-2:	Blocos Básicos do PGA	58
Imagem	4.6.2.2-3:	AIC do PGA	58
Imagem	4.6.2.2-4:	RETIFICADOR DE TENSÃO DO DRENO	60
Imagem	4.6.2.2-5:	Circuito de Eliminação do Modo DC	61

Imagem	4.7.1-1:	Top Level do Projeto Completo dentro do Ambiente CADENCE	62
Imagem	4.7.2-1:	Esquemático do Bloco Básico em Ambiente CADENCE de Projeto	63
Imagem	4.7.2-2:	Comparador do Bloco Básico do CMD	64
Imagem	4.7.2-3:	Curvas do Comparador Com e Sem o Inversor	65
Imagem	4.7.2-4:	Esquemático do Circuito do Tradutor	65
Imagem	4.7.2-5:	Esquemático do Circuito da XNOR.	66
Imagem	4.7.2-6:	Circuitos das portas NOR e NAND	67
Imagem	4.7.3-1:	Erro de Offset. [41]	67
Imagem	4.7.3-2:	Simulação Com ERRO de OFFSET	68
Imagem	4.7.4-1:	Erro de Ganho. [41]	69
Imagem	4.7.4-2:	Erro de Ganho devido ao excesso de Corrente.	70
Imagem	4.7.5-1:	ERRO DE DNL. [41]	70
Imagem	4.7.5-2:	Comportamento dos Sinais dos Módulos B0 é o mais significativo	71
Imagem	4.7.5-3:	Resultado na escada de valores convertidos do ERRO DE DNL	72
Imagem	4.7.5-1:	ERRO DE INL. [41]	72
Imagem	4.7.5-2:	Resultado na escada de valores convertidos dos ERROS DE INL e DNL, que possuem extremos nas mesmas faixas.	73
Imagem	4.7.5-3:	Pontos críticos de INL e de DNL	73
Imagem	4.7.5-4:	Atrasos e Falha no Ponto de Inversão do Módulo em Baixa Frequência.	74
Imagem	4.7.5-5:	Inversão do módulo em Alta Frequência Vgs do By-Pass em níveis baixos.	74
Imagem	4.7.5-6:	Inversão do módulo em Alta Frequência Vgs em níveis adequados.	74
Imagem	4.7.7-1:	Bloco By-Pass Operando com Entrada de + 5 $\mu$ A.	76
Imagem	4.7.7-2:	Bloco By-Pass Operando com Entrada de - 5 $\mu$ A.	77
Imagem	4.7.7-3:	Distribuição do ruído em torno do sinal.	78
Imagem	4.7.8-1:	Rampa acompanhada por uma conversão Ideal	79
Imagem	4.7.8-2:	Sinal do Ruído de Quantização Ideal	80
Imagem	5.1-1:	100 Primeiros Valores da Escada Gerada para Entrada de 2 MS/s	85
Imagem	5.1-2:	Escada Gerada com 2 MS/s	86
Imagem	5.1-3:	Bits Gerado Pelo CMD	86
Imagem	5.1-4:	Máximo do DNL	87
Imagem	5.1-5:	Máximo do INL	87
Imagem	5.1-6:	Quadrado da Diferença do Sinal Analógico e do Digital.	88
Imagem	5.1-7:	Quadrado da Diferença do Sinal Analógico e do Digital.	88
Imagem	5.2.2-1	Topologia Geral do Circuito Low Power.	90
Imagem	5.2.3-1	Topologia do Circuito PMOS	91
Imagem	5.2.4-1	FFT do Sinal do Conversor Recebendo o Sinal de Interesse em 5 Hz	93
Imagem	5.3.1.5.1-1:	Código Binário e Código de Gray (Imagem 4.4.2-4).	96
Imagem	5.3.1.5.3-1:	Descontinuidade do PIPELINE (Imagem 2.1.3.1)	97

Imagem	5.3.1.5.3-1:	Comportamento do CMD e do SAR (Imagem 4.4.2-1)	97
Imagem	5.3.1.5.4-1:	Tradutor (Imagem 4.4.3-5)	97
Imagem	5.3-1:	Bits menos significativos continuam alterando independente do mais significativo	98
Imagem	6-1:	Tabela comparativa entre o SAR, o $\Sigma\Delta$ , o FLASH e o CMD.	101
Imagem	6-2:	O SAR é o conversor mais rápido e o mais eficiente dentro do mapeamento do estado da arte de 2011 [1].	102



## LISTA DE SÍMBOLOS, NOMENCLATURAS E ABREVIACÕES

AMS	AUSTRIA MICROSYSTEMS
ADC	ANALOGIC DIGITAL CONVERTER
CMOS	COMPLEMENTARY METAL OXIDE SEMICONDUCTOR
$C_{ox}$	OXIDE CAPACITANCE PER UNIT GATE AREA
DAC	DIGITAL ANALOGIC CONVERTER
$ddp$	POTENTIAL DIFFERENCE
$G_m$	TRANSCONDUTÂNCIA
Hz	HERTZ
$I_{ds}$	DRAIN TO SOURCE CURRENT
IEEE	INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS
LSB	LESS SIGNIFICANT BIT
ms	MILISEGUNDOS
RMS	ROOT MEAN SQUARE
SNR	SIGNAL NOISE RELATION
SoC	SYSTEM ON CHIP
S/H	SAMPLE AND HOLD
$T_h$	THRESHOLD LEVEL
V	VOLT
$V_{bs}$	BULK TO SOURCE VOLTAGE
$V_{ds}$	DRAIN TO SOURCE VOLTAGE
$V_{gs}$	GATE TO SOURCE VOLTAGE
$V_{th}$	THRESHOLD VOLTAGE
$\mu A$	MICRO AMPÈRE
$\mu o$	MOBILIDADE ELETRÔNICA
nA	NANO AMPÈRE



# 1. INTRODUÇÃO

## 1.1. Prefácio

Este trabalho introduz uma nova arquitetura de conversor analógicos para digitais, batizada de CONVERSOR DO MÓDULO DA DIFERENÇA (CMD), capaz de minimizar os principais inconvenientes apresentados pelos modelos consagrados no mercado e no meio científico. No trabalho relacionamos os principais problemas das arquiteturas dominantes no mercado e meio científico para minimizar esses efeitos na arquitetura de ADC CMD, conforme a imagem 1.1-1.

DESvantagens				
	SAR	$\Sigma\Delta$	FLASH	CMD
ÁREA EM SOC	POUCO AFETADOS	POUCO AFETADOS	MAIS AFETADOS	POUCO AFETADOS
CAPACITORES	MAIS AFETADOS	PARCIALMENTE AFETADOS	PARCIALMENTE AFETADOS	POUCO AFETADOS
CONSUMO	POUCO AFETADOS	PARCIALMENTE AFETADOS	MAIS AFETADOS	POUCO AFETADOS
INTERFACE ANALÓGICO DIGITAL	POUCO AFETADOS	POUCO AFETADOS	MAIS AFETADOS	POUCO AFETADOS
OPERAÇÃO	POUCO AFETADOS	MAIS AFETADOS	MAIS AFETADOS	POUCO AFETADOS
CICLOS	MAIS AFETADOS	POUCO AFETADOS	POUCO AFETADOS	POUCO AFETADOS
OVER SAMPLE	POUCO AFETADOS	MAIS AFETADOS	POUCO AFETADOS	POUCO AFETADOS
RESOLUÇÃO	PARCIALMENTE AFETADOS	POUCO AFETADOS	MAIS AFETADOS	POUCO AFETADOS
REFERÊNCIA	PARCIALMENTE AFETADOS	POUCO AFETADOS	MAIS AFETADOS	POUCO AFETADOS



Imagem 1.1-1: Tabela comparativa entre o SAR, o  $\Sigma\Delta$ , o FLASH e o CMD.

Para maior compreensão da motivação desta pesquisa e de sua estrutura como um todo, iniciaremos uma abordagem sobre o contexto histórico dos conversores de dados e o estado da arte, visando inserir o leitor no cenário evolutivo dos circuitos. Na seção seguinte apresentaremos o modelo mais antigo de conversor de dados que encontramos a partir dos referenciais teóricos abordados. E faremos uma breve abordagem do telégrafo, precursor da telefonia como conhecemos, pois foi a ideia de transmitir a voz humana, através de um sinal elétrico, a grande predecessora do desenvolvimento da tecnologia de conversão de dados (embasado em [2]).

## 1.2. Contextualização Geral

A data do primeiro conversor de dados analógicos para digitais ou vice-versa não pode ser determinada. O registro mais antigo que temos conhecimento está descrito em [2] e é um sistema de controle de vazão de uma represa turca do século 18. Já era utilizada a lógica binária para gerar diversos valores, ou seja, um DAC. Esse conversor serve apenas de curiosidade para esse trabalho, pois o nosso principal foco está nas áreas de comunicação, processamento de dados, registro de dados e armazenamento de dados.

Entre os anos 1825 e 1875 [2] o telegrafo é o protagonista dos processos de comunicação e ganhou grande desenvolvimento na busca de um melhor aproveitamento do meio de transmissão e maior imunidade ao ruído. As tecnologias de transmissão de multiplexação pelo tempo (TDM) e pela frequência (FDM) estavam em desenvolvimento, sendo a FDM mais desafiadora. A busca por melhorias na FDM abriu portas inexploradas para o processamento de dados, sejam analógicos ou digitais. Tendo seu ápice, em 1875, A.G. Bell percebeu que a tecnologia já havia evoluído o suficiente para transmitir a voz como um sinal elétrico. A invenção do telefone contribuiu, principalmente para despertar o interesse de grandes investidores. Essa época é relatada como o início da teoria da informação [2]. O teorema da amostragem seria publicado apenas em Certain Topics in Telegraph Transmission Theory em 1928 [3], por Harry Nyquist, ou seja, uma das principais teorias utilizadas atualmente não era nem praticada.

O primeiro conversor, com características similares aos conversores atuais, surge em 1921, possui 5 bits e era similar a um conversor FLASH [2] [4], possui 32 níveis e para cada nível é gerado um código binário com correspondência direta, e seu inventor foi Paul M. Rainey. O sistema realizava a transmissão em série para cada bit, aumentando a imunidade ao ruído no processo de transmissão, passou a ser transmitido em apenas dois níveis, similar ao PCM, mas não atingiu abrangência. O trabalho que realmente lançou o PCM surgiu com Alec Harley Reeves, em 1937, com o intuito de obter um sistema com maior imunidade ao ruído.

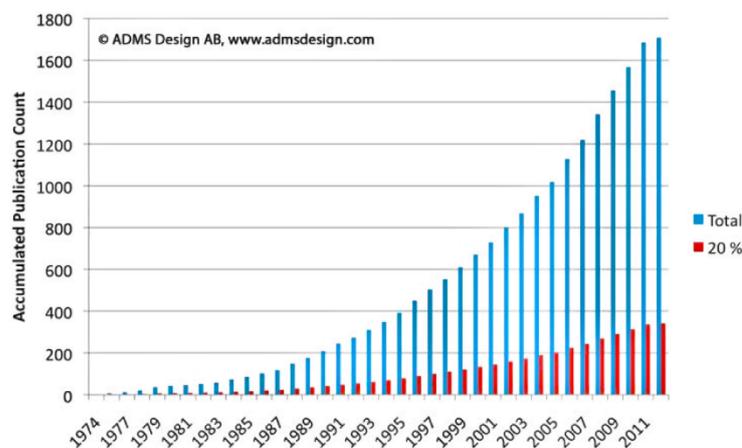


Imagem 1.2-1: Relação das publicações no IEEE [6]. Os conversores representam 20%.

Notamos que o campo da comunicação foi um dos principais estimuladores ao desenvolvimento dos conversores de dados. Hoje não é muito diferente, temos telefonia móvel, internet, rádio, televisão e etc. Contudo, surgiram diversos outros campos: controle automático; sensores para robótica; instrumentação de dados; drives para HDs; medicina; equipamentos de guerra eletrônica; sensoriamentos geofísicos; equipamentos e sistemas de calibração; radares e sonares; analisadores de espectros; equipamentos de gravação de vídeos; telescópios; equipamentos fotográficos e outros [5]. Permitindo que os conversores representem 20% das publicações no IEEE [6]. Os conversores são utilizados para amostras e registrar dados, desde simples imagens do dia-dia até sinais para

monitoramento, controle e calibração. Isso é possível devido à evolução de outra área: a da computação, que para alguns surge com um esboço muito além da tecnologia da época feito em 1822 por Charles Babbage [7] [8]. Para outros surge com o primeiro computador programável, com lógica booleana, com Konrad Zuse em 1938 [9] [10]. Mas o primeiro computador digital eletrônico de grande escala, surge durante a segunda guerra em 1946 [11] [12], o ENIAC. Assim, em 1946, o mundo vive uma grande disputa tecnológica: a computação e a comunicação passam a se desenvolver de forma desenfreada. Os recursos fluem por interesses bélicos, políticos, econômicos e científicos, que permitiram atingirmos o patamar tecnológico atual.

Dessa forma, hoje os conversores de dados são amplamente utilizados em diversas áreas, seja para tornar a transmissão de dados mais robusta, para registrar informações de forma mais eficaz, para processar sinais de forma mais fácil e muitas outras aplicações. Isso é claro quando analisamos os impactos dos conversores no meio científico, 20% das publicações deixam claro a relevância do assunto ao mundo. Importância que permite o surgimento de trabalhos científicos dedicados exclusivamente aos conversores, ou seja, não apenas para uma grande área que necessita de um conversor. E, esse é o contexto do trabalho em questão, ou seja, trata-se de um trabalho dedicado exclusivamente para o conversor propriamente dito.

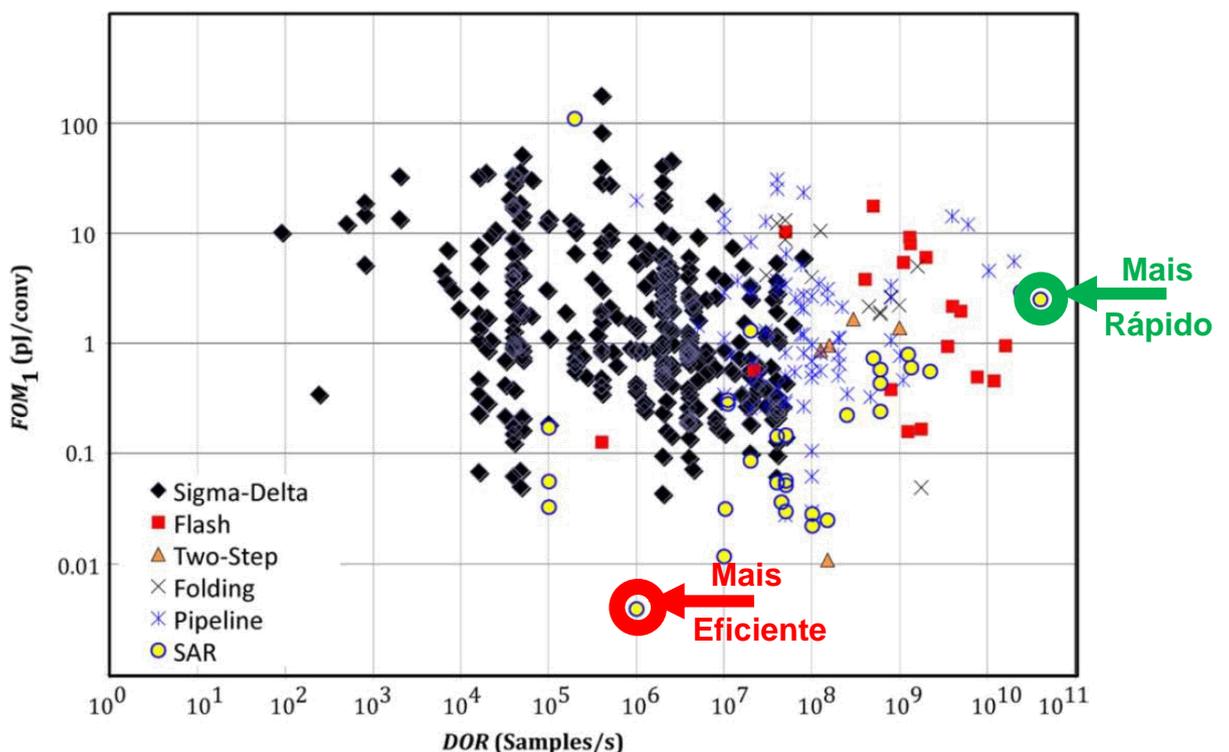


Imagem 1.2-2: Arquiteturas principais mapeadas pela energia por conversão em relação à taxa de amostras por segundo [1].

- A energia por conversão, eixo das ordenadas da imagem 1.2-2, relaciona a energia em pJ por conversão, ou seja, está normalizado com a resolução do conversor.
- O eixo das abscissas da imagem 1.2-2 relaciona a taxa de amostragem por segundo.

Nesse contexto surge o trabalho [1] no qual é apontado o estado da arte do desenvolvimento dos conversores de dados. Na imagem 1.2-2, retirada de [1], temos diversas arquiteturas de grande aplicação e relevância no ano de 2011. Trata-se de uma realidade bem atual e nela podemos verificar que o conversor mais rápido e o conversor mais eficiente no estudo de [1] foi projetado na arquitetura SAR.

### 1.3. Contexto Local

O trabalho foi desenvolvido dentro da Universidade de Brasília (UnB), que conta com softwares de desenvolvimentos industrial CADENCE para ensinar e treinar os alunos dentro da área de projetos, testes, desenvolvimento e fabricação de circuitos integrados. Os dois principais laboratórios são o LPCI e o LDCI. O trabalho também contou com o apoio de uma empresa encubada dentro da UnB, a DFCHIP - fundada com auxílio do programa CI Brasil.

O CI Brasil surge por um estudo realizado pelo Ministério da Ciência, Tecnologia e Inovação (MCTI) em 2011 identificando *“que menos de 30% dos produtos exportados pelo Brasil apresentam algum componente tecnológico. A maioria é composta por produtos de baixa intensidade tecnológica, como commodities, produtos intensivos em mão de obra e recursos naturais, o que impacta diretamente no equilíbrio da balança comercial brasileira”* [11]. O Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), a empresa pública brasileira Financiadora de Estudos e Projetos (FINEP) e o Banco Nacional de Desenvolvimento Econômico e Social (BNDES) dão apoio ao programa. Dessa forma, a empresa DFCHIP se encaixa dentro de um contexto de desenvolvimento de projetos de circuitos integrados dentro da UnB, que passa a contar com um programa federal de apoio ao desenvolvimento da eletrônica no Brasil.

### 1.4. Motivação

Em 2007, começam os primeiros contatos do autor com a área da eletrônica com sistema de RDS (Rádio Definido por Software [12]), que possui grande dependência de conversores de dados analógicos para digitais. Trabalho desenvolvido dentro do laboratório do Grupo de Processamento Digital de Sinais (GPDS) da UnB, na ocasião apoiado pelo Programa Institucional de Bolsas de Iniciação Científica (PIBIC) do CNPq. O trabalho foi desenvolvido em uma plataforma FPGA a USRP™ (Universal Software Radio Peripheral) desenvolvida pela Ettus Research [13], atualmente pertencente à National Instruments Company, vinculada ao grupo GNU Radio [14].

Em 2008, surge o primeiro contato com o CADENCE do Laboratório de Projeto de Circuitos Integrados (LPCI) da UnB, com as plataformas de projetos VISTUOSO [15] e ENCOUNTER [16], e no meio profissional; com a equipe da empresa DFCHIP - encubada na UnB. Onde eram desenvolvidos conversores de dados analógicos para digitais SAR e Sigma Delta. Durante o projeto do Conversor  $\Sigma\Delta$ , deparamo-nos com o trabalho [1], e verificamos que o conversor SAR ainda era a arquitetura com menor consumo por conversão dentro do estado da arte de 2011 e, também, o conversor com maior

taxa de amostragem; imagem 1.2-2. O detalhe de uma das primeiras arquiteturas desenvolvidas, atingir duas das características relevantes para um conversor de dados analógicos para digitais, desperta grande interesse.

Então, a ideia inicial foi estudar o SAR, suas principais características e obter uma arquitetura com potencial a ser a melhor arquitetura em termos de taxa de amostragem e consumo por conversão [1]. Nesse processo o foco central foi a necessidade do SAR, adicionar um ciclo de processo para cada bit adicionado, o que indubitavelmente adiciona atraso e consumo [11] [21]. Ou seja, elaborar um conversor que possuísse todas as características da lógica do SAR, mas sem adicionar um ciclo de tempo por bit de resolução. Seria melhor que o SAR em termos de consumo e taxa de amostragem. Consequentemente, o conversor já teria o potencial necessário para ser o melhor conversor dentro do estado da arte de 2011 em termos de consumo e taxa de amostragem.

A ideia se concretiza observando que o SAR precisa de um ciclo por bit, pois há uma tomada de decisão dependente do valor do bit adjacente mais significativo. A questão passa a ser a eliminação dessa dependência do valor digital do bit adjacente mais significativo. A resposta para eliminar essa dependência é obtida pela padronização do processo analógico de cada bit independentemente do valor digital do bit anterior. O processo passa a ser:

- A. Sempre subtrair da entrada uma referência fixa, que é metade do fundo de escala da entrada;
- B. Obter o módulo do resultado do processo de subtração;
- C. Comparar a entrada com a referência. Se maior o bit é “1” e “0” caso contrário.

Trata-se de um processamento analógico, pois qualquer dependência em estabilizar um sinal digital é um atraso desnecessário. Veja que no processo “C” já temos a informação equivalente à tomada de decisão do SAR e que no processo “B” o valor será sempre uma saída positiva. O sinal gerado em “B” é a entrada para o próximo bit, que recebe o sinal sem a depender do valor obtido em “C”. Assim, haverá um circuito independente para cada bit. O processo define o nome do conversor, batizado de Conversor do Módulo da Diferença (CDM), pois, exceto pelo bit mais significativo, todos bits convertem o módulo da diferença obtida do processamento do bit adjacente mais significativo.

## 1.5. Objetivos

O objetivo geral é desenvolver uma NOVA ARQUITETURA DE CONVERSORES DE SINAIS ANALÓGICOS EM SINAIS DIGITAIS. Desenvolvendo maior imunidade aos efeitos de segunda ordem, com um projeto simples e mesclando os conceitos de conversores de Nyquist e conversores de Sobre Amostragem, para que possa ser aproveitado em um grande range de projetos.

Para atingir o objetivo geral desse trabalho é necessário avaliar os impactos a que se propõe, pois para que o projeto tenha maior imunidade aos efeitos de segunda ordem, ampliados com a

evolução tecnológica dos processos de fabricação, já que a redução do canal deixa o dispositivo mais sensível. Serão necessários circuitos menos complexos, de projeto fácil, em termos de layout e esquemático, e com pouca dependência com as variações de tensão.

Assim, os objetivos parciais para atingir o objetivo geral passam por:

- O Estudo das principais arquiteturas existentes no mercado, fazendo a comparação entre elas - apontando suas vantagens e desvantagens;
- Modelar o comportamento desejado para uma validação adequada tanto em softwares quanto em ambiente de projeto;
- Caracterizar o comportamento desejado para o conversor e verificar os seus impactos dentro das diversas validações de um conversor em relação ao comportamento esperado de um conversor dentro de um contexto universal;
- Projetar circuitos que possam reproduzir o modelo desenvolvido com as características desejadas e caracterizadas.
- Comparar os resultados obtidos da nova arquitetura e projeto para apontar os cenários em que o projeto tenha maior aplicação;
- Apresentar os passos necessários para que a arquitetura atinja o efeito desejado na área.

## **1.6. Divisão da Dissertação**

O trabalho é dividido em 6 capítulos:

- O primeiro é uma introdução apresentando o desenvolvimento e a evolução da área.
- No segundo temos uma revisão bibliográfica apenas com o conteúdo pertinente para a compreensão do trabalho, sem muito detalhamento, mas fornecemos fontes ricas em relação ao conteúdo apresentado;
- O terceiro capítulo descreve a metodologia utilizada para o desenvolvimento do trabalho, do projeto, dos resultados, das validações e das características do CMD;
- No Quarto Capítulo detalhamos o processo do CMD, validamos o processo, detalhamos o seu comportamento, descrevemos suas características e descrevemos o projeto com seus blocos;
- No quinto capítulo apresentamos, como resultados, dois projetos em ambiente CADENCE, sendo um conversor de 10 bits e outro um conversor Ultra Low Power de 14 bits, finalizando com um resgate das características abordadas nos capítulos anteriores para comparar as arquiteturas já consagradas na com o CMD;
- No sexto capítulo, apresentamos as conclusões referentes ao CMD.



## 2. REVISÃO BIBLIOGRÁFICA

Esse capítulo tem o objetivo de apresentar os conversores consagrados na academia e no mercado. Os conversores serão contextualizados em suas características gerais, ou seja, não serão apresentadas particularidades de projetos específicos, pois não é intuito desse trabalho apontar a melhor topologia para cada tipo de conversor, mas apresentar os conceitos, os processos, as dificuldades e facilidades inerentes às diversas arquiteturas.

### 2.1. CONTEXTO GERAL

Como vimos na introdução, existem diversas arquiteturas de conversores de analógico para digital, entretanto, eles podem ser divididos em dois grupos principais, que são baseados na frequência de Nyquist e os conversores baseados na sobre amostragem [17] [18]. O primeiro grupo é o mais antigo, são conversores que realizam um processo direto, ou seja, os valores são convertidos em seu formato final. Os conversores baseados em sobre amostragem, por outro lado, realizam um processo indireto que reflete um comportamento estatístico que depende da entrada.

Uma das principais características dos conversores de Nyquist é a necessidade de um circuito de sample-and-hold [18] para estabilizar o sinal durante o processo do sinal analógico [19]. O processo insere ruído inevitavelmente, mas o ruído não se torna um fator limitante para o seu uso [18]. Esses conversores trabalham com amostragem de 6 a 40 vezes o valor da frequência do sinal de interesse. Dentro desse grupo se destacam os ADC Successive approximation register (SAR), Integrating, Flash, Averaging, Folding and interpolating, Sub-ranging, Pipelined, Algorithmic e Time-interleaved [17] [20]. Não é necessário apresentar todos os conversores nesse trabalho, apenas alguns são suficientes para o objetivo a ser alcançado. As referências que trata com grande riqueza os diversos conversores é a [20].

O grupo de conversores de sobre amostragem operam em frequências de amostragem superiores aos valores anteriores, sendo de 40 a 1024 vezes a frequência do sinal de interesse. O processo de sobre amostragem é uma forma para aumentar a relação entre o sinal e o ruído, o SNR, do processo como um todo.

Algumas das saídas que vem sendo empregadas para os processos atuais é o processamento dos dados em domínio analógico como o realizado em [18], [28], [29] e [30], ou seja, são realizados processos matemáticos ainda no domínio analógico para não gastar energia nos processos de conversão de dados ou perder tempo em conversão de dados para realização de processamento para transferir comandos analógicos para o meio com intensidade controlada pelo sinal de interesse. Esses processamentos são em sua maioria tratamento de dados com baixa frequência, pois os circuitos que trazem grande facilidade para o desenvolvimento de processos, equações e funções de transferência

fazem o uso de dispositivos em inversão fraca, que, por estarem em estado de baixa condução, apresentam dispositivos lentos [31].

Contudo já vem surgindo a tendência em atender processamento analógico em alta frequência promovido pelas inovações dos dispositivos móveis para comunicação utilizando processos paralelos e modulação sigma delta [32]. Os trabalhos com moduladores sigma delta vem atingindo alta frequência, mas o ruído gerado para conseguir operar com baixo consumo é significativo [33], [34] e [35]. Os circuitos para os moduladores também exigem interfaces que possam inibir glitches, mas suas vantagens vêm compensando a utilização dessas interfaces.

A tendência de circuitos dedicados também tem espaço no processamento digital. São os circuitos digitais de processadores dedicados a determinados processos como para processamento em vídeo, jogos, emuladores e outros [36]. As duas áreas, tanto a analógica, quanto a digital, vêm se desenvolvendo, mas circuitos mistos demandam conversores com melhores características. Com o grande emprego e desenvolvimento de circuitos digitais vem reduzindo rapidamente o tamanho dos circuitos. Para os circuitos analógicos a redução dos componentes vem exigindo mais pelo aumento dos efeitos secundários, como o efeito de modulação de canal [17], [37] e [38]. Dessa forma, as inovações tecnológicas facilitam inicialmente os circuitos digitais, pois os códigos em verilog, muitas vezes, facilmente adaptáveis, o que promove a migração rápida para tecnologias menores e retira rapidamente as tecnologias antigas de fabricação do mercado.

Esse trabalho tem como objetivo contribuir de forma significativa a um setor de vital importância para o desenvolvimento de novas tecnologias. O conversor desenvolvido nesse trabalho é um projeto analógico com grande vínculo digital. E também nos preocupamos com os efeitos de segunda ordem, pois faz o uso de circuitos mais robustos ao efeito de modulação de canal [37] e [38]. Também utiliza topologias que buscam reduzir a necessidade de layouts complexos, com o objetivo de simplificar os circuitos básicos e, conseqüentemente, acelerar o projeto da arquitetura como um todo.

### **2.1.1. FLASH**

Trata-se da aplicação pura e simples do processo de conversão. Com a característica de ser o mais rápido dos conversores, pois sua conversão é baseada na obtenção de cada um dos valores digitais, a partir de valores analógicos, pré-definidos, que dividem as faixas analógicas de interesse. O conversor também recebe outros nomes no mercado como “Instantâneo” e “Paralelo” [20], [21] e [22].

Para explicarmos a conversão, consideremos uma faixa analógica de 0 a 10V e um processo com 2 bits para converter valores dentro dessa faixa. Teríamos:

BINÁRIOS EQUIVALENTES	
10	1 1
7,5	1 0
5	0 1
2,5	0 0
0	

Imagem 2.1.2-1: correspondência entre faixas analógicas e digitais para o Flash.

Dessa forma, seriam gerados no mínimo três valores analógicos que seriam comparados simultaneamente com a entrada da seguinte forma: quando as três comparações fossem 000 teríamos “00<sub>2</sub>”; quando os resultados das saídas dos comparadores fossem 001 (comparador de 7,5V = 0; comparador de 5V = 0 e comparador de 2,5V = 1), ou seja, os comparadores com os valores de 7,5V e 5V estariam comparando 0, portanto o valor estaria abaixo dessas tensões, mas o comparador com referência em 2,5V apresenta saída igual a 1, portanto, o valor é maior que 2,5V e menor que 5V, estando o valor dentro da faixa correspondente à “01<sub>2</sub>”. Os valores seguintes seriam 011 e 111, para as saídas dos comparadores, que corresponderia às faixas dos valores binários “10<sub>2</sub>” e “11<sub>2</sub>” respectivamente. Dessa forma, o conversor faz o uso de três comparadores para representar dois bits. Essa característica é ideal para conversores de poucos bits, mas, para conversores com quantidades elevadas de bits, esses valores de tensão se tornam muito próximos. O número de comparadores se torna muito elevado e, portanto, o circuito passa a consumir muita potência e o offset dos comparadores se torna crítico.

### 2.1.1.1. Projeto do Flash

Dos fatores apontados na secção anterior, temos a topologia genérica do Conversor Flash:

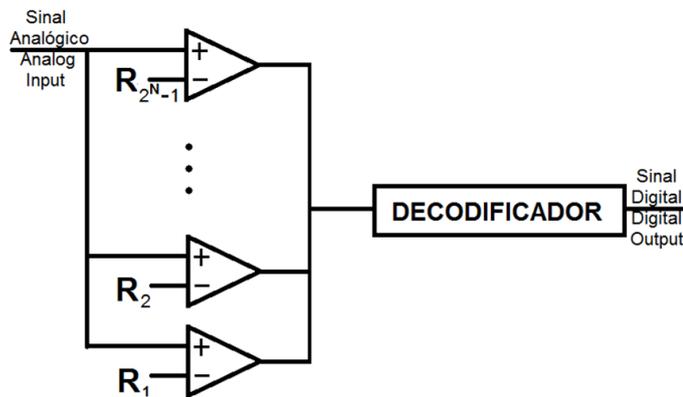


Imagem 2.1.2.1-1: Blocos Básicos de um Conversor FLASH

O “FLASH” faz uso de “ $2^N - 1$ ” comparadores, sendo “N” o número de bits desejados para o valor analógico de interesse. Para interpretar os sinais dos comparadores é utilizado um decodificador, que recebe os sinais de todos dos bits e apresenta, em sua saída, o valor binário correspondente equivalente à soma das saídas dos comparadores, por exemplo, caso todos os comparadores tenham

como saída “0” (zero), o valor binário correspondente será “0000” (zero) e, da mesma forma, se a soma das saídas dos comparadores for, por exemplo, “15” (quinze) a saída correspondente será “1111<sub>2</sub>”, ou seja, o valor é superior à 15 valores de referência, portanto, passou dos valores de “0000<sub>2</sub>”, “0001<sub>2</sub>”, ..., e “1111<sub>2</sub>”, portanto, está entre a 15<sup>o</sup> e o 16<sup>o</sup> valor de divisão dentre as faixas analógicas. Outro método para obter o resultado final é utilizar portar XOR para encontra quando a saída é 1 [20].

É importante observar que ao todo serão gerados pelos comparadores do Flash, “ $2^N - 1$ ” bits, que darão origem à apenas “N” bits. Pode-se afirmar que o Flash trata o sinal analógico como um todo e sem causar atrasos desnecessários com a formatação dos valores binários durante o processo de amostragem do valor analógico, a ideia é armazenar a informação necessária para reproduzir o valor digital equivalente à faixa analógica, à qual pertencia o valor analógico de interesse.

### **2.1.1.2. Vantagens**

Serão apresentadas as vantagens que contribuem para a elaboração da TESE.

#### **2.1.1.2.1. Todas as comparações ocorrem simultaneamente em paralelo;**

É a principal vantagem do conversor, mas também será a responsável pelos diversos problemas presentes na arquitetura. Essa qualidade do conversor não permite o acúmulo de atrasos na geração de cada bit para o conversor, portanto, ele conta apenas com o atraso do comparador mais lento, dentre os, “ $2^N - 1$ ”. Ainda, conforme veremos, também no próximo conversor, há a necessidade de estabilizar o sinal durante o processo de conversão, pois cada bit gerado, seja ele em seu formato final, ou no formato gerado pelo conversor (no caso do Flash as saídas dos comparadores), há a necessidade de segurar a variação do sinal analógico enquanto é obtido um valor fiel para o valor binário em questão [20].

No flash todos os bits são processados de forma simultânea e por mais que um bit demore mais que o outro para ser gerado - seja por questões de falhas no processo de fabricação, ou por proximidade entre os valores comparados - que deixariam o comparador mais lento, o tempo máximo para cada conversão será o período destinado à estabilização de qualquer conversor em seu pior ponto de operação. Com isso, o Flash poderia reduzir ao máximo o intervalo dentre amostras, quando comparado com as arquiteturas atuais [20].

#### **2.1.1.2.2. Baixo Ruído De Quantização Na Conversão**

O conversor não faz uso de processos de subtração, multiplicação, comparações em série, modulação e outros diversos processos presentes nos demais conversores, portanto, seu ruído de quantização é derivado do processo compressão de uma faixa de um seguimento de valores analógicos

para a correspondência de um único valor digital, ou seja, pelo ruído de um conversor ideal e pelos erros entre os valores analógicos gerados para referência [20].

### 2.1.1.2.3. Monotonicidade Garantida

Realizando a divisão da escala analógica com dispositivos em série, pode-se garantir os valores das referências, que são as responsáveis por determinar que as faixas analógicas correspondentes a cada valor binário, sejam diferentes e crescentes, quando tomadas do menor valor até o maior valor de referência.

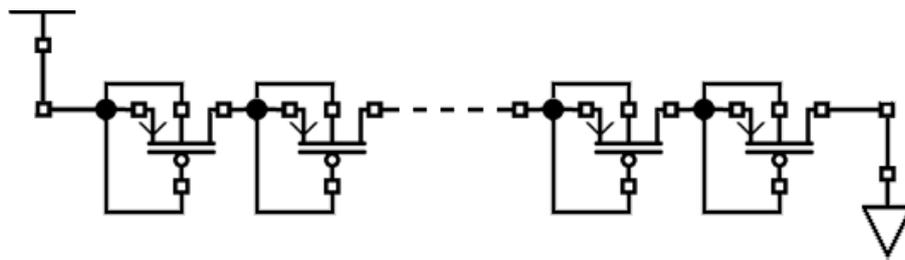


Imagem 2.1.1.2.3-1: Divisor de tensão para o conversor Flash

No circuito da imagem acima, verifica-se que os dispositivos em questão são do tipo P-MOS, normalmente fabricados sobre uma poça de substrato N, que é isolado do substrato do circuito como um todo, portanto, esses dispositivos possuem corpo isolado dos demais dispositivos do sistema como um todo. Assim, os dispositivos são polarizados em corte que caracteriza valores elevados de impedância. O grande problema desses circuitos divisores de tensão é a baixa corrente que eles são capazes de fornecer quando realizados conforme a imagem acima, ou seja, com impedância elevada. E outro problema está relacionado aos circuitos com baixa impedância, normalmente os circuitos resistivos apresentados em muitos artigos e livros [20] [23]. Esses dispositivos são suscetíveis aos ruídos térmicos, consumo elevado e grandes áreas em SoC, apesar de serem capazes de fornecer corrente para o circuito responsável pela leitura dos valores de referência. Entretanto, de qualquer forma, como os dispositivos estão em série, é garantido que haverá queda de tensão de um dispositivo para o outro, portanto, a monotonicidade pode ser garantida com facilidade, em conversores de pouco bits.

### 2.1.1.3. Desvantagens

Serão apresentadas as principais desvantagens desse conversor: a grande capacitância de entrada e a área em silício [20] e [24].

#### 2.1.1.3.1. Capacitância de Entrada

A capacitância, praticamente, dobra a cada bit, pois, por menores que sejam os capacitores dos Gates dos comparadores, há uma capacitância parasita que é multiplicada pelo número de valores

binários – 1. Por exemplo, para um conversor de 10 bits, seria a capacitância do GATE do comparador, utilizado para cada faixa definida por uma referência, multiplicada por “ $2^{10} - 1$ ”, ou seja, 1023 vezes o valor da capacitância do GATE do comparador, considerando que seja o mesmo projeto de comparador para cada uma dessas faixas, o que normalmente é inviável devido à necessidade de operação para uma grande faixa de valores de entrada.

#### **2.1.1.3.2. Área em SoC**

O grande problema no desenvolvimento do conversor em questão está no desenvolvimento de seu Layout, pois os comparadores devem possuir comportamento similar o quanto for necessário, conforme a resolução desejada, entretanto, para números elevados de comparadores, qualquer casamento é inviável, como exemplo, um conversor de 10 bits possuiria 1023 comparadores, que devem operar de forma similar em relação ao seu offset, além desse fator, os valores de referência devem chegar dentro faixas estreitas de erro, por exemplo, para um conversor de 10bits operando com 2,7V aplicados ao circuito geral, a diferença para cada referência de tensão seria de aproximadamente 2,64mV, portanto, a variação de 1mV já comprometeria 37,93%. Assim, há a necessidade de planejar toda a distribuição garantindo a mesma queda de tensão ou corrigindo a queda de tensão, demandando tempo de projeto, área de circuito e consumo.

#### **2.1.1.3.3. Resolução da Referência**

Dando continuidade ao item anterior, a ressalva na seção “2.1.1.2.3” desse trabalho, relativo à monotonicidade, que fica comprometida em conversores Flash [19] e [23]. O valor da diferença de tensão é reduzido à metade a cada bit adicionado, comprometendo a fidelidade da referência e exigindo topologias que interfiram ao mínimo nas referências [20].

#### **2.1.1.3.4. Consumo**

O aumento do número de dispositivos de forma, praticamente, proporcional exponencialmente ao número de bits da resolução, além de aumentar o consumo pelo quantitativo de dispositivos, também impacta no consumo de cada dispositivo, que deverá operar com uma precisão maior que implica em interfaces melhores, como a fonte de tensão e corrente para o sistema e os sinais de polarização dos componentes [20].

#### **2.1.1.3.5. Interface Analógico Digital**

Devido ao elevado número de comparadores, surge a necessidade de diversos circuitos para realizar os ajustes dos níveis de tensão entre o analógico e o digital, pois, em muitas tecnologias, nos deparamos com circuitos digitais operando em níveis baixos de tensão e circuito analógicos necessitando de níveis elevados de tensão para conseguir operar dentro da especificação. Assim surge

a necessidade de vários circuitos do tipo Bootstrapped e / ou Level Shifter [25] e [26], que consomem quantidades significativas de corrente por possuir componentes operando em dois níveis de alimentação.

### 2.1.2. SAR

BINÁRIOS EQUIVALENTES	
10	1 1
7,5	1 0
5	0 1
2,5	0 0
0	

Imagem 2.1.3-1: correspondência entre faixas analógicas e digitais para o SAR.

É baseado em processo similar ao de busca binária dentro de um conjunto de dados ordenados. Dessa forma, há um valor de referência que divide o conjunto em duas partes, aproximadamente iguais, mas a seleção da parte em que a busca continuará, é definida pelo próprio sinal de analógico de interesse e os extremos das partes divididas determinam onde será mantida a busca, pois conhecemos o valor que é procurado, mas no conversor, conhecemos as partes em questão e o objetivo encontrar a faixa conhecida que contém o valor de interesse. Retomando o exemplo para dois bits, utilizados para o conversor Flash.

Ao comparar o se sinal verificando é maior ou menor que a referência, define-se que o bit é 0, caso seja menor ou 1, caso seja maior. Assim, considerando o valor analógico de 5,7V, o primeiro valor de referência é 5V, pois dividirá toda a faixa de 0 até 10V em duas faixas aproximadamente iguais, e, ao comparar 5,7V com 5, constatará que é maior- teremos “1” para o primeiro bit e saberemos que o valor está contido na faixa de 5 até 10V. Alguns conversores subtraem o valor da referência do sinal e comparam com a mesma referência, ou seja, seria 0,7V comparados com 5V e constatar-se-ia que o valor é menor, portanto o bit valeria “0”, resultando na faixa analógica caracterizada pelo valor binário “10<sub>2</sub>”. O outro processo seria possuir o valor de 7,5V, comparando o valor de 5,7 com 7,5 o que também resultaria, de forma equivalente, no resultado anterior, ou seja, “10<sub>2</sub>”. O segundo processo implicaria na mesma desvantagem vista na seção “2.1.2.3.2.”, que trata da necessidade do Flash em gerar diversos valores de referência.

Conforme foi verificado, o SAR realiza as comparações sucessivamente até determinar a faixa que contém o valor analógico. A princípio poder-se-ia dividir o quanto queira, conseguindo a precisão que desejar Entretanto, há inserção de ruído a cada aproximação e perde-se tempo a cada comparação, portanto, ao adicionar cada bit, a frequência é reduzida, o circuito necessitará de maior precisão e o sinal analógico pode variar, durante o processo.

A última observação exige do SAR o uso de um circuito que “segure” o valor analógico durante a conversão, um circuito de sample and hold (S/H) - O S/H aplica ruído de quantização, mas, por ser frequentemente concebido com o uso de um circuito capacitivo que estará sujeito a inserção de erros durante o chaveamento para a amostragem. Contudo, o capacitor dá origem à topologia de redistribuição de cargas que realiza o processo de “subtração” ou determinação de uma nova referência através de processos de equivalência capacitiva que não será abordada pela complexidade e por não ser parte integrante da TESE.

### 2.1.2.1. Blocos Básicos do SAR

O SAR é feito com blocos básicos de S/H, que mantém o valor da entrada fixo durante a conversão; DAC que converte um sinal digital composto pelos “bits”, que já foram confirmados / comparados, por uma referência analógica, em ciclos anteriores do processo de aproximação sucessiva; “Comparador” que realiza a comparação do sinal de entrada com a referência gerada pelo DAC; e o “Registro” que armazenas os “bits” convertidos para serem utilizados pelo DAC e para apresentar o valor digital convertido [21].

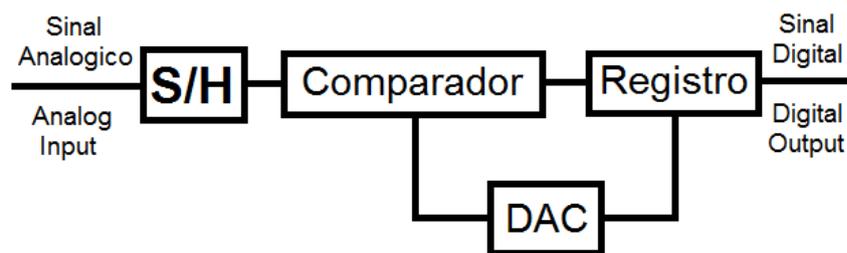


Imagem 2.1.2.1-1: Blocos Básicos do SAR.

No caso do segundo processo, descrito na seção 2.1.3, o DAC seria o gerador dos sinais de referência para a conversão continuada a cada etapa, ou seja, ele poderia ser substituído por referências fixas, poderia ser um circuito de subtração ou um circuito de redistribuição de cargas. Independente da topologia, verifica-se que o circuito não terá o número de componentes aumentando de forma drástica como no caso do Flash, mas a complexidade em obter referências fieis continua. Além da complexidade com as referências, há a questão do atraso que não ocorreria de forma drástica com o aumento de bits [20].

Conforme apontado na introdução, o SAR é tido como um conversor baseado no teorema de Nyquist. Assim, a frequência limite de operação estaria próxima aos valores dos ciclos completos do conversor. O SAR é um dos conversores que atinge as maiores frequências de conversão, mas o aumento do número de bits implica na redução da frequência máxima que o conversor poderá converter.

### 2.1.2.2. Vantagens

Serão apresentadas as vantagens do conversor SAR.

#### **2.1.2.2.1. Pouco Aumento de Circuitos com a Resolução**

O número de componentes (não considerando os circuitos de interface) do “SAR” é implementado de forma mínima com o aumento da resolução, pois haveria apenas o aumento do número de capacitores e chaves, no caso da topologia baseada em redistribuição de cargas e os circuitos aumentariam de tamanho apenas para atingir a resolução necessária. Entretanto, haveria apenas um comparador, que é um circuito com grande impacto no consumo. Assim, por ter apenas um no comparador, pode-se “investir” em topologias de maior complexidade para atender à resolução em questão [21].

#### **2.1.2.2.2. Estrutura de Projeto Simples**

Devido ao emprego de apenas um comparador, ao contrário do Flash, há uma liberdade maior disponível para o consumo do comparador. Assim, o projeto se torna mais simples, apesar de possuir blocos com funções diferenciadas em relação ao conversor, o bloco de registros é mais simples que o circuito de decodificação do Flash. A grande complexidade do projeto estará no DAC, mas esse processo é simplificável pelo uso do circuito de redistribuição de cargas. Entretanto, o circuito de redistribuição implica no uso de capacitores e será discutido na seção de desvantagens [21].

#### **2.1.2.2.3. Poucas Referências**

O uso de uma quantidade reduzida de referências reduz o tamanho do circuito como um todo, pois, como foi visto na seção “2.1.1.” a grande quantidade de referências aumenta a área do circuito, permitindo que interferências eletromagnéticas e térmicas atinjam o circuito. Também surge a dificuldade em distribuir diversas referências mantendo de forma fiel as características entre elas. Os circuitos necessários, para gerar as referências para a conversão, aumentam o consumo inevitavelmente, pois aumentam pela adição de componentes ou pelos circuitos que realizam a leitura e os processos que faram uso dessas referências [21].

### **2.1.2.3. Desvantagens**

#### **2.1.2.3.1. Número de Ciclos de Conversão**

Devido ao processo de aproximação sucessiva, o “SAR” precisará de “ $n$ ” ciclos para gerar um sinal de “ $n$ ” bits e, portando, sua frequência de operação é limitada por “ $n$  intervalos” de tempo de conversão para cada bit. As tecnologias são limitadas em termos de frequência, pois os componentes possuem parasitários devidos às suas imperfeições e também aos seus princípios fundamentais. Dessa forma, o SAR não consegue atender aos processos que precisam de grandes frequências e resolução [20].

### **2.1.2.3.2. Capacitores Grandes**

Devido ao grande número de capacitores [19] o circuito possui dispositivos maiores para atender a demandas solicitada por esses componentes durante o acionamento e distribuição de carga entre eles. Esses componentes são sujeitos aos erros de fabricação, ruídos térmicos e perturbações elétricas de outros dispositivos, todos esses efeitos também são associados à dificuldade de casar os componentes dentro de um bom Layout [27] [37], o que gera um erro acumulado durante os processos de transferência de carga. Esse fato ocorre mesmo com o processo de redistribuição de cargas, mas é menor devido à redução significativa da área demandada por esses dispositivos [20].

### **2.1.3. PIPELINE**

O “Conversor PIPELINE” (também chamado de “subranging quantizer”). Os dois nomes dão um entendimento a respeito do conversor. É um conjunto de conversores com baixa resolução, mas, normalmente, com grande velocidade. Com o resultado dessa conversão de baixa resolução, gera-se um resíduo dessa conversão, que é amplificado para gerar uma nova conversão. Para os blocos básicos, são usados, geralmente, conversores Flash, pois são circuitos com poucos bits e, portanto, esses blocos possuem poucas referências a cada bloco e menos circuitos, permitindo trabalhar com maiores vantagens [20].

O conversor realiza, uma aproximação do valor a cada etapa, portanto, sua lógica é muito similar à do SAR, pois a cada conversão, consegue-se definir uma faixa mais estreita para a localização da faixa que contém o valor analógico de interesse. A grande diferença é que as etapas passam a gerar uma aproximação com um número maior de bits, lembrando que no caso do SAR era apenas 1 bit a cada ciclo. Outro aspecto que diferencia do SAR é o fato dos circuitos estarem em série, portanto, há um circuito conversor com seus comparadores a cada etapa. Dessa forma, quando um desses conversores termina o processo de conversão e passa para o próximo, esse bloco já poderá receber uma nova referência para ser convertida.

O processo perde menos tempo tratando cada valor amostrado por S/H, mas há a necessidade de estabilizar os valores ao transferir o resíduo de um conversor para outro, principalmente em casos de grandes resoluções, pois, retornamos ao problema inicial do FLASH, que é a falha no processo envolver resíduos de cargas que representam valores dentro de mV, ou até mesmo  $\mu\text{V}$ . Por exemplo, um sinal de 2,7V em um conversor de 10 bits, a diferença entre as referências analógicas seria de 2,7V dividido por 1024, ou seja, 2,64mV. Além desse fato, o mercado exige e possui carência de conversores com alta resolução, como conversores de 16 a 24 bits, obtidos em baixas frequências, em sua maioria com conversores Sigma Delta.

#### **2.1.3.1. Projeto Tradicional**

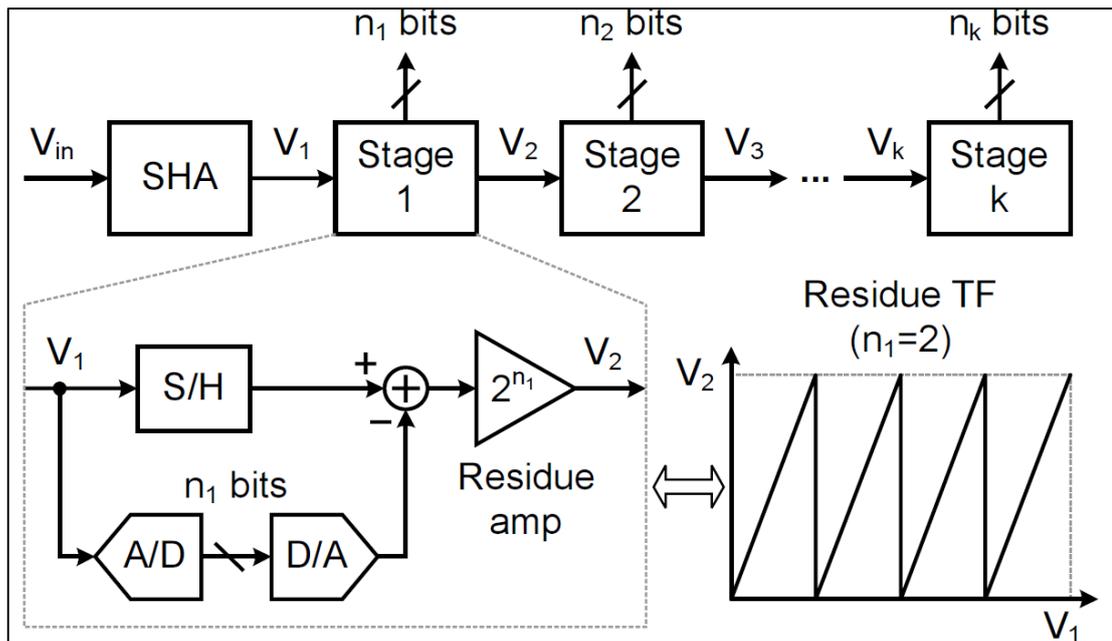


Imagem 2.1.3.1-1: Retirada de: [22].

O projeto tradicional é constituído de blocos independentes com um SH, um ADC, um DAC, um bloco de subtração de sinais analógicos.

O processo de subtração recebe o sinal do S/H e do DAC para gerar o sinal analógico, a ser amplificado e entregue para o próximo bloco. E o ADC recebe o sinal do S/H para conversão da correspondente etapa de conversão [22].

## 2.1.3.2. Vantagens

### 2.1.3.2.1. CONVERSÃO POR ETAPAS PARALELAS

As etapas de conversão são simples, rápidas e independentes (quando possui blocos de “S/H” entre as etapas, armazenando os resíduos das conversões anteriores). Dessa forma, o processo atinge grande taxa de conversão sem a necessidade de grandes áreas de.

Como as etapas são paralelas, o PIPELINE mantém um ciclo de curta duração para o processo de conversão, uma vez que continua a converter o valor armazenado enquanto amostra outro e assim por diante de acordo com o número de etapas paralelas que o conversor tiver [22].

## 2.1.3.3. Desvantagens

### 2.1.3.3.1. RESOLUÇÃO COMPROMETIDA

A grande desvantagem do processo é gerada pela forma como é obtido o resíduo da conversão, pois tanto a subtração do sinal analógico gerado pelo DAC, quando o processo de subtração do sinal analógico, geram erros e, portanto, o erro do primeiro processo define a resolução máxima que

o processo pode atingir. Outra questão é o tamanho do capacitor necessário para que o sinal seja amostrado e processado sem alterar de forma significativa a carga do capacitor em questão [20].

### **2.1.3.3.2. INTERFACE OPERACIONAL**

Elevado número de interfaces para tratar o controle operacional e ajustes para o casamento lógicos dos intervalos de conversão [39].

Dessa forma, o processo é simples em questões de número de passos para a conversão, mas os circuitos de quantização devem possuir alta precisão, pois o erro é amplificado para a próxima entrada e assim sucessivamente.

## **2.1.4. SIGMA DELTA**

### **2.1.4.1. Projeto Tradicional**

No Sigma Delta a entrada analógica passa por um modulador, cuja saída é conectada a um comparador que determinará uma sequência de valores de zeros e uns, quando a saída do modulador for menor que uma determinada referência o valor é ZERO e, quando for maior que a referência, o valor da saída é UM. A oscilação ocorre com a realimentação negativa do sistema através de um DAC e devido ao processo não ser fechado, o que dá instabilidade e satura a saída, hora no mínimo e hora no máximo. A realimentação do circuito poderia ser fechada, mas demanda circuitos de alto ganho e um controle mais refinado.

Da sequência, de ZEROS e UNS, é realizado o processamento adequado para extrair a informação, com determinada resolução, do comportamento do sinal analógico da entrada. Observe que, no caso de uma palavra de 4 bits, por exemplo, precisaríamos no mínimo de 4 oscilações para garanti-los, portanto, o trem de zeros e uns do Sigma Delta. A questão, entretanto, está no formato desses valores, pois eles não possuem uma forma padrão, o que eles possuem é um comportamento gerado pelo sinal de entrada e, por isso, necessitamos de uma sobre amostragem (geralmente definida como "OverSampling") que será responsável por apresentar uma quantidade maior pulsos válidos para processar e obter a faixa analógica correspondente a aquele padrão [20].

### **2.1.4.2. Vantagens**

A ideia central do conversor é atingir alta resolução com a modulação do ruído de quantização para altas frequências (noise shaping). O ruído de quantização está presente até mesmo no processo ideal de conversão. O processo de modulação gera o trem de zeros e uns com o comportamento estatístico derivado do nível de tensão em que o sinal se encontra. E esse ruído é facilmente filtrado por filtros digitais simples, pequenos e consomem pouca potência quando comparados com filtros analógicos.

#### **2.1.4.2.1. NOISE SHAPING**

A grande vantagem do conversor “ $\Sigma\Delta$ ” é a eliminação de grande parcela do ruído de quantização, pois é inevitável o surgimento desse ruído, limitante da resolução dos conversores em geral. Essa característica é possível pelo processo de sobre amostragem do sinal. Permitindo o tratamento do sinal para uma caracterização mais preciso do sinal

Com a eliminação do ruído de quantização, já no domínio digital, que possui facilidade de processamento e precisão, é possível aumentar o ENOB com baixo consumo de potência pelo número de bits e pela taxa de amostragem, o que caracteriza uma figura de mérito de potência em ascensão. Essas características permitem projetos com grande resolução e baixo consumo.

#### **2.1.4.2.2. BITS NÃO DEPENDEM DE VALORES CONVERTIDOS**

O conversor realiza as operações necessárias apenas para registrar a informação, ou seja, não trata o sinal para processamento futuro, pelo contrário, demanda a necessidade de interpretação e processamento do sinal digital gerado. Porém é mais simples tratar o sinal no domínio digital, atualmente, no lugar de realizar todo o processo no analógico.

#### **2.1.4.3. Desvantagens**

A grande limitação do sigma delta está na frequência de amostragem, pois a frequência do processo e a frequência do sinal devem possuir uma diferença que é a frequência para a sobre amostragem - “Over Sample” – necessária para armazenar a informação para a demodulação em domínio digital. Assim, quanto menor a frequência menor é o ruído, conseqüentemente, maior será a resolução. E de forma inversa, quanto maior a frequência do sinal, menor a sobre amostragem e menor será a resolução.

##### **2.1.4.3.1. OVER SAMPLE**

Devido à necessidade de realizar uma amostragem muito superior à frequência do sinal. O conversor fica limitado para processos que necessitam de alta frequência. A solução adotada para esse processo é a inserção de moduladores em paralelo, aumentando de forma significativa o consumo e as interfaces para tratar o sinal.

##### **2.1.4.3.2. USO DE CAPACITORES**

O Modulador também é dependente dos capacitores para manter a sua operação, são componentes sensíveis aos ruídos térmicos e ocupam grandes áreas em SoC [37]. Os problemas relacionados aos capacitores são singulares para os conversores.

## 2.2. COMENTÁRIOS

A arquitetura mais explorada foi a do SAR, que é um dos conversores com maior simplicidade em seu desenvolvimento. As aproximações sucessivas para a conversão é um processo seguro, mas um ciclo depende do outro, o que atrasa o processo a cada ciclo e limitando a possibilidade de alta resolução e taxa de amostragem simultaneamente [5]. Apesar da grande maioria dos trabalhos apontarem o SAR como um conversor maior lento que o Flash, vimos na imagem 1.2-2 que o SAR é o conversor com maior taxa de amostragem para o estado da arte até 2011 [1].

O Flash seria o conversor ideal se não houvesse comprometimento de consumo, pois realiza todo o processo em paralelo, há atraso quanto à estabilização dos conversores, por melhor que seja o casamento, apresentar-se-ão discrepâncias entre si. Do Flash é desejado o atraso mínimo para cada conversão, mas sem o uso de um número elevado de circuitos, pois o conversor torna-se inviável para alta resolução e perde velocidade devido às capacitâncias parasitas inseridas pelos diversos componentes.

O Pipeline é uma arquitetura que divide o processo em etapas que ocorrem de forma serial, porem simultaneamente. O processo tem grande entrada de ruído nos processos de sincronismos e ampliação do resíduo de conversão entre as etapas. O processo de ampliação do ruído também é um processo lento. Mas é um dos conversores que atinge as maiores taxas de conversão para o atual estado da arte, estando entre o Flash e o SAR [1].

O Sigma Delta é um projeto que explora métodos matemáticos para reproduzir um sinal digital gerado por uma modulação do sinal de entrada, mas cria uma dependência da sobre amostragem para realizar o processo de noise shaping. É um dos conversores com maior número de inovações [17].



### **3. METODOLOGIA**

Esse capítulo descreve qual foi o método estruturado a partir da motivação sustentada pela possibilidade do desenvolvimento de um novo conversor de dados.

#### **3.1. DEFINIÇÃO DE OBJETIVOS**

Já tendo o objetivo geral do trabalho bem definido de desenvolver uma nova arquitetura de conversão de dados de sinais analógicos para digitais, surge a necessidade em desmembrar etapas para atingir o objetivo geral.

Dessa forma, o primeiro passo para dar sustância ao desenvolvimento do projeto foi o estudo das principais arquiteturas existentes e a busca pelos trabalhos de maior representatividade na área. E verificar se não existia nenhum conversor que realizava esse processo dentro dessas fontes.

Com o embasamento da revisão bibliográfica. Buscamos um modelo para atender à ideia de um processo com as características desejadas, ou seja, realizar a conversão com a resolução desejada sem a necessidade de ciclos de conversão envolvendo valores já convertidos. Entretanto, o desafio era, também, não utilizar várias referências para definir cada faixa como é feito no caso no Flash e não depender de sobre amostragem para esse processo. Assim, queríamos evitar circuitos com dimensões que inviabilizassem o projeto quando fosse necessário realizar para vários bits e / ou comprometer a frequência máxima com processo de sobre amostragem.

Validar o processo obtido pelo modelo estruturado com as características desejadas. A necessidade é de uma simples validação lógica. Assim essa etapa se destina a verificar se o modelo realmente atinge o objetivo mínimo para o processo, veja que até aqui ainda não está sendo colocado em questão se o processo é viável.

Com um processo válido surge a necessidade de levantar suas características dentro do que esperamos para esse novo conversor. Apenas com uma boa caracterização poderemos dar início a um projeto estruturado com um objetivo claro de início, meio e fim.

Nessa etapa damos início ao projeto do conversor já com a consciência do que buscamos. Esse projeto surge com as etapas simples:

- A definição de blocos de alto nível com características mínimas para definir um nível máximo (TopLevel);
- A definição das topologias que serão utilizadas para gerar cada um dos blocos de alto nível;
- A definição de como serão as interfaces entre os blocos de alto nível já conhecendo as topologias dos circuitos e os sinais necessários para colocar os blocos em operação;

- A definição das interfaces entre os sub-blocos dentro dos blocos de alto nível;
- O projeto de cada sub-bloco já imersos dentro do cenário de possibilidades e limitações da tecnologia adotada para o projeto;
- A validação de cada sub-bloco de forma individual e em conjunto para realização do fechamento dos blocos de alto nível;
- A validação dos conjuntos de blocos de alto nível;
- A documentação dos resultados obtidos com as limitações e as especificações atingidas pelo projeto como um todo.

Com o projeto finalizado apontamos suas qualidades e confrontamos o projeto final com as arquiteturas existentes para verificar se foi satisfeito o objetivo geral e, conseqüentemente, a motivação que levou à elaboração desse trabalho.

E com um trabalho finalizado buscar novos trabalhos para serem desenvolvidos, ou seja, verificar quais são os novos passos abertos e as possibilidades para explorar o que foi desenvolvido.

### **3.2. ESTUDOS DAS ARQUITETURAS**

Inicialmente foi necessário realizar um levantamento das características dos conversores dos projetos mais consagrados do setor, pois esses conversores já foram testados e melhorados ao longo dos anos, são arquiteturas antigas e, por isso trazem segurança em relação ao conhecimento apresentado nos textos, uma vez que trazem entre si semelhanças que permitem avaliar o comprometimento do trabalho desenvolvido em cima da arquitetura.

Foi realizada a revisão dos circuitos consagrados para desenvolver de forma clara os textos, equações e modelos apresentados no trabalho em questão, pois é interessante facilitar a compreensão dos assuntos tratados e conceitos aplicados para o desenvolvimento dessa nova arquitetura, garantindo que seja de fácil reprodução dentro da academia e do mercado.

Também foi levantado um pequeno histórico verificando como ocorreu o desenvolvimento dessas arquiteturas e, com isso, saber quais são os principais fatores que motivam o desenvolvimento do meio. Dessa forma, podemos saber também qual a importância de um novo desenvolvimento dentro desse setor.

Essa etapa é fundamental para sustentar se será um trabalho voltado para um grande público de interesses ou se o trabalho se destina apenas a um desenvolvimento isolado.

### **3.3. DEFINIÇÃO DE UM MODELO COMPORTAMENTAL**

O desenvolvimento desse modelo é um processo ainda matemático e logístico. Assim partimos da ideia que é a motivação do trabalho para criar uma lógica comportamental que atenda aos requisitos dessa motivação descrevendo passo a passo que é desejado em busca de uma conexão entre cada etapa. Definido o comportamento desejado partimos para um modelo matemático que estruture e permita a caracterização comportamental de forma fácil e fiel.

### **3.4. CARACTERIZAÇÃO LÓGICA DOS BLOCOS**

Trata-se de uma simples validação lógica, mas sem essa validação poderíamos dar continuidade no desenvolvimento de um sistema que não faria sentido algum. Dessa forma, trata-se de um processo simples, porém essencial para o desenvolvimento do trabalho.

Essa caracterização já se aprofunda com a previsão de como serão gerados os valores digitais a partir do processo analógico. No caso do trabalho essa etapa é fundamental, pois será visto no desenvolvimento que os valores digitais são convertidos em um formato particular do Conversor de Módulo da Diferença (CMD).

Sendo desenvolvida inicialmente pelo modelo matemático, sendo uma validação através de uma análise matemática em cima das equações que modelam o comportamento desejado e motivado pela ideia de criar um novo ADC.

A validação foi finalizada com algoritmos desenvolvidos em linguagem de alto nível com facilidade de desenvolvimento gráfico. Assim, dentro dessa etapa, também foram desenvolvidos algoritmos para descrever o comportamento do modelo gerado. Buscando uma avaliação comparativa também foi desenvolvido um algoritmo nesse software modelando outro conversor já consagrado. A principal ideia do algoritmo é estressar o processo em busca de falhas e comportamentos de maior complexidade.

### **3.5. PROJETO**

O projeto foi dividido em duas etapas principais. Primeiro temos o projeto dos blocos, topologias adotadas sub-blocos e outros. Segundo temos a implementação da primeira etapa em ambiente de simulação, no nosso caso o CADENCE. As duas etapas são distintas, mas não ocorrem em momentos distintos. Trata-se de um processo cíclico e uma etapa serve de estruturação para a outra etapa. Apesar de parecer obvio o retorno de uma etapa na outra, é crucial para um projetista saber quando deve trabalhar cada etapa. Há um momento em que a primeira etapa pode se tornar densa e confusa ao passo que a persistência de alguns processos na segunda etapa podem manter o projetista cego na tentativa de implementar algo que já estava errado. Dessa forma, um bom casamento dessas etapas principais dão fluxo a um projeto rápido.

Na primeira temos o projeto dos blocos de alto nível com características mínimas para definir um nível máximo que servirá de referência e amarração para do sistema como um todo. Seguindo para a definição de topologias a serem adotadas, das interfaces entre os blocos e sub-blocos e definição dos processos de validação.

A segunda etapa já está dentro das definições da tecnologia que será adotada para o processo e como as topologias definidas ainda na primeira etapa são afetadas. Assim o processo de validação das topologias, dos sub-blocos, dos blocos adotados dentro do ambiente de projeto pode condenar topologias por não funcionarem ou por não atenderem as necessidades dentro da especificação na tecnologia adotada.

Por fim, é gerada a documentação dos resultados obtidos com as limitações e as especificações atingidas pelo projeto como um todo.

### **3.6. ANÁLISE DOS RESULTADOS**

Com o projeto finalizado e caracterizado poderemos afirmar que o objetivo geral foi atingido e trata-se de uma nova arquitetura implementada. Em contrapartida, ainda resta saber os potenciais dessa nova arquitetura, que abre várias portas para o desenvolvimento de diversos circuitos dentro da área de conversão de dados analógicos para digitais.

Assim, será feito um resgate de características exploradas de algumas arquiteturas consagradas na revisão bibliográfica para conflitar e comparar com as características do conversor do módulo da diferença com as suas particularidades e semelhanças.



## **4. PROJETO DO CONVERSOR CMD**

Esse capítulo é dedicado ao projeto do conversor CMD. Apresentaremos as propriedades consagradas do CMD e de outros conversores que levaram à ideia por trás do CMD.

Nesse capítulo também serão vistos um algoritmo descrevendo o processo realizado pelo CMD, um modelo matemático do CMD e o comportamento binário dos valores convertidos pelo CMD. Munido das informações básicas do conversor, seguiremos com os detalhes relacionados ao processo de obtenção dos valores convertidos, como esses valores se comportam, quais as suas particularidades e uma lógica simples para tratamento dos valores convertidos. Assim, para uma validação simples, será apresentado um modelo de software com curvas comparativas.

O capítulo segue com a definição de uma estrutura básica para o projeto do conversor e como essas estruturas afetam os parâmetros básicos relacionados ao processo de conversão ideal, ou seja, os erros tradicionais como OFFSET, ERRO DE GANHO, DNL, INL, SNR E ENOB. Também será apresentada uma medida de consumo por bit convertido.

### **4.1. DEFINIÇÕES INICIAIS E PROPRIEDADES CONSAGRADAS**

A ideia do CMD surge ao avaliar o trabalho [1] e perceber que o SAR apresentava vantagens significativas em seu estado da arte dentre as principais arquiteturas de conversores. Assim, a ideia central do novo conversor é fundamentada com o intuito de evitar a pior entre as características do conversor SAR, sendo o atraso obrigatório para tomada de decisão. Entretanto, foram observadas outras arquiteturas dentre as mais relevantes.

O Conversor CMD realiza a estabilização de todos os bits simultaneamente, esta é a característica mais importante do conversor Flash, entretanto, não é totalmente paralelo, pois o sinal é propagado, ainda em sua forma analógica, para todos os blocos básicos. Do Sigma Delta o conversor faz uso da propriedade de registrar um padrão digital independente de sua interpretação, evitando o tempo gasto para a tradução, que pode e deve, no entendimento do autor, ser realizada após o registro do conteúdo do sinal analógico de forma suficiente para a resolução desejada. Do SAR o processo herdou a busca binária dentro das faixas analógicas, determinada pela granulosidade imposta pela resolução digital desejada. O conversor também parte do princípio de possuir processos paralelos, conforme a ideia de um conversor Pipeline.

Dessa forma, a arquitetura é fundamentada com o objetivo aproveitar a busca binária e eliminar por completo da tomada de decisão do conversor SAR. Entretanto, esse processo permitiu inserir características na arquitetura do CMD de outras arquiteturas como Flash, Sigma Delta e Pipeline

## 4.2. ALGORITMO DE CONVERSÃO DO CMD

O algoritmo de conversão é obtido observando o algoritmo de conversão do SAR, que é um algoritmo recorrente. O algoritmo descrito aqui procura ser o mais genérico possível, uma vez que buscamos encaixar a maiorias dos conversores SAR nele. Outro modelo de algoritmo em [20]. Iniciaremos com o algoritmo do SAR para colocar o leitor dentro de um cenário mais acolhedor, uma vez que já é um algoritmo de grande conhecimento na área.

### Algoritmo genérico do SAR

- >1: INICIAR;
- >2: SELECIONAR O BIT MAIS SIGNIFICATIVO;
- >3: AMOSTRAR E SEGURAR O VALOR DE INTERESSE;
- >4: COMPARAR AMOSTRA COM O VALOR MÉDIO DOS POSSÍVEIS VALORES PARA A AMOSTRA;
- >5: SE COMPARAÇÃO IGUAL A 1 → LIMITE OS POSSÍVEIS VALORES PARA A METADE SUPERIOR DOS POSSÍVEIS VALORES PARA A AMOSTRA E IR PARA ">7:";
- >6: LIMITAR OS POSSÍVEIS VALORES PARA A METADE INFERIOR DOS POSSÍVEIS VALORES PARA A AMOSTRA;
- >7: REGISTRAR O VALOR DA COMPARAÇÃO NO BIT ATUAL;
- >8: SE NÃO É O BIT MENOS SIGNIFICATIVO → AVANÇAR UM BIT E IR PARA ">4:";
- >9: INFORMAR FIM DE CONVERSÃO E ATUALIZAR REGISTRO DE SAÍDA;
- >10: RETORNAR PARA ">2:";

### Fim do algoritmo genérico do SAR

Temos que:

- ">1:" apenas informa que o processo iniciou;
- ">2:" seleciona o bit mais significativo para receber o valor da primeira comparação;
- ">3:" realizar a amostragem como um circuito de S/H;
- ">4:" é o processo de comparação do valor de interesse com a referência do ciclo;
- ">5:" verifica se a comparação é 1 para selecionar a faixa superior dos possíveis valores e pula a etapa ">6:" caso a comparação seja 1;
- ">6:" seleciona metade inferior para os ainda possíveis valores para a amostra no ciclo atual (só acontece se a comparação for 0);
- ">7:" registra o valor da comparação para o bit que está selecionado no ciclo atual;
- ">8:" verifica se já foram registrados todos os bits, ou seja, se está selecionado o último bit. Se não for o último ciclo, então seleciona o próximo bit e retorna para a comparação de ">4:";
- ">9:" se chegar aqui todos os bits foram devidamente registrados e o valor está pronto, portanto, é liberado para o registro adequado ou processamento;
- ">10:" retorna para o início do processo.

O algoritmo do conversor CMD é muito similar ao algoritmo do conversor SAR e ambos podem ser escritos como algoritmos recursivos. Então, para ficar bem evidente a diferença entre os dois algoritmos o do CMD também será escrita na forma recursiva, mas o processo do projeto a frente não é o de um sistema cíclico ou realimentado e será um processo assíncrono.

### Algoritmo genérico do CMD

- >1: INÍCIAR;
- >2: SELECIONAR O BIT MAIS SIGNIFICATIVO;
- >3: AMOSTRAR E SEGURAR O VALOR DE INTERESSE;
- >4: COMPARAR AMOSTRA COM A REFERÊNCIA;
- >5: SUBTRAIR DA AMOSTRA A REFERÊNCIA;**
- >6: OBTER O VALOR ABSOLUTO/MÓDULO DA AMOSTRA E MULTIPLICAR POR 2 (DOIS);**
- >7: REGISTRAR O VALOR DA COMPARAÇÃO NO BIT ATUAL;
- >8: SE NÃO É O BIT MENOS SIGNIFICATIVO → AVANÇAR UM BIT E IR PARA “>4.”;
- >9: INFORMAR FIM DE CONVERSÃO, TRADUZIR O VALOR CONVERTIDO E ATUALIZAR REGISTRO DE SAÍDA;**
- >10: RETORNAR PARA “>2.”;

### Fim do algoritmo genérico do CMD

Temos que:

- “>1.” apenas informa que o processo iniciou;
- “>2.” seleciona o bit mais significativo para receber o valor da primeira comparação;
- “>3.” realizar a amostragem como um circuito de S/H;
- “>4.” é o processo de comparação do valor de interesse com a referência;
- **“>5.” subtrair referência da “AMOSTRA” (nesse caso é o valor analógico que está sendo processado no ciclo atual);**
- **“>6.” se “AMOSTRA” é maior que “0” não faz nada, caso o contrário multiplique por “-1”;**
- “>7.” registra o valor da comparação para o bit que está selecionado no ciclo atual;
- “>8.” verifica se já foram registrados todos os bits, ou seja, se está selecionado o ultimo bit. Se não for o último ciclo, então seleciona o próximo bit e retorna para a comparação de “>4.”;
- **“>9.” se chegar aqui todos os bits foram devidamente registrados. O valor precisa ser traduzido, pois o formato gerado não está na forma de um número binário padrão. Após a tradução, o valor convertido está pronto para o registro adequado ou processamento;**
- “>10.” retorna para o início do processo.

Os algoritmos se diferenciam nas linhas “>5.”, “>6.” e “>9.”. Essa diferença foi realizada para eliminar o processo de tomada de decisão existente no SAR. Dessa forma, ao contrário do SAR que realiza uma operação quando a comparação é 1 e outra operação quando a comparação é 0. O CMD sempre realiza o mesmo processo, ou seja, não precisa interromper o processo analógico para

determinar qual o tipo de processo será realizado, pois no algoritmo do CMD sempre há a subtração do sinal pela referência e a obtenção do valor absoluto / módulo do resultado do processo de subtração. Dessa forma, o processo pode se tornar assíncrono com facilidade, mas no caso do SAR, o circuito precisaria de realimentação para se tornar viável, ou demandaria uma árvore de processos, cada um prevendo uma possível tomada de decisão, o que tornaria o circuito até maior que o circuito de um conversor Flash tradicional.

Note que não há referência à uma topológica específica para o SAR, pois o processo em questão está apenas atualizando a faixa na qual o valor analógico poderá se encontrar. Note que ao subtrairmos da amostra a referência utilizada no processo de comparação do SAR, também estamos alterando a faixa para os possíveis valores do sinal analógico. Entretanto, em outro sentido, uma vez que soubermos que o sinal analógico está na faixa inferior ou superior e, dessa forma, subtrairmos ou adicionarmos valor na referência, respectivamente, também estaremos considerando faixas distintas e limitadas para os possíveis valores para o sinal de interesse.

Da mesma forma, o processo para o CMD não fica preso ou deixa de ser descrito pelo algoritmo em questão e, com o intuito de preservar o entendimento, frisamos que o circuito do projeto do CMD não é um processo cíclico, pois a independência do processo analógico com o digital, permite a criação de poucos circuitos analógicos para processar o sinal. E o projeto que será descrito possui “N” circuitos em série, quando analisado o fluxo do sinal analógico, porém em paralelo, quando analisado o fluxo da informação digital, sendo “N” igual ao número de bits do projeto em questão. Isso permite gerar os bits de forma independente e simultaneamente. Também veremos à frente que o projeto não utilizou circuitos de S/H ou T/H.

### 4.3. MODELO MATEMÁTICO

O Modelo matemático para o “CMD” será denominado “ $CMD(x)_n$ ”, sendo “ $x$ ” o sinal analógico a ser convertido para sinal digital e “ $n$ ” o nível do bit em questão. Não será considerado o ajuste para o formato binário padrão, pois não é um formato obrigatório. O formato considerado será o numérico na base 2. A função de “ $CMD(x)_n$ ” é obtida através da função “ $cmd(x)$ ”, que determina o valor intensidade da corrente de saída do bloco básico que é definida como:

$$cmd(x)_1 = 2 * \left| x - \frac{R}{2} \right| \quad (4.3-1)$$

$$cmd(x)_2 = 2 * \left| cmd(x)_1 - \frac{R}{2} \right| \quad (4.3-2)$$

$$cmd(x)_n = 2 * \left| cmd(x)_{n-1} - \frac{R}{2} \right| \quad (4.3-3)$$

$$CMD(x_{in})_n \begin{cases} Se, cmd(x)_n > 0 ; 1 \\ Se, cmd(x)_n < 0 ; 0 \end{cases} \quad (4.3-4)$$

$$\{n \in I | n \subset [0, +\infty)\} \quad (4.3-5)$$

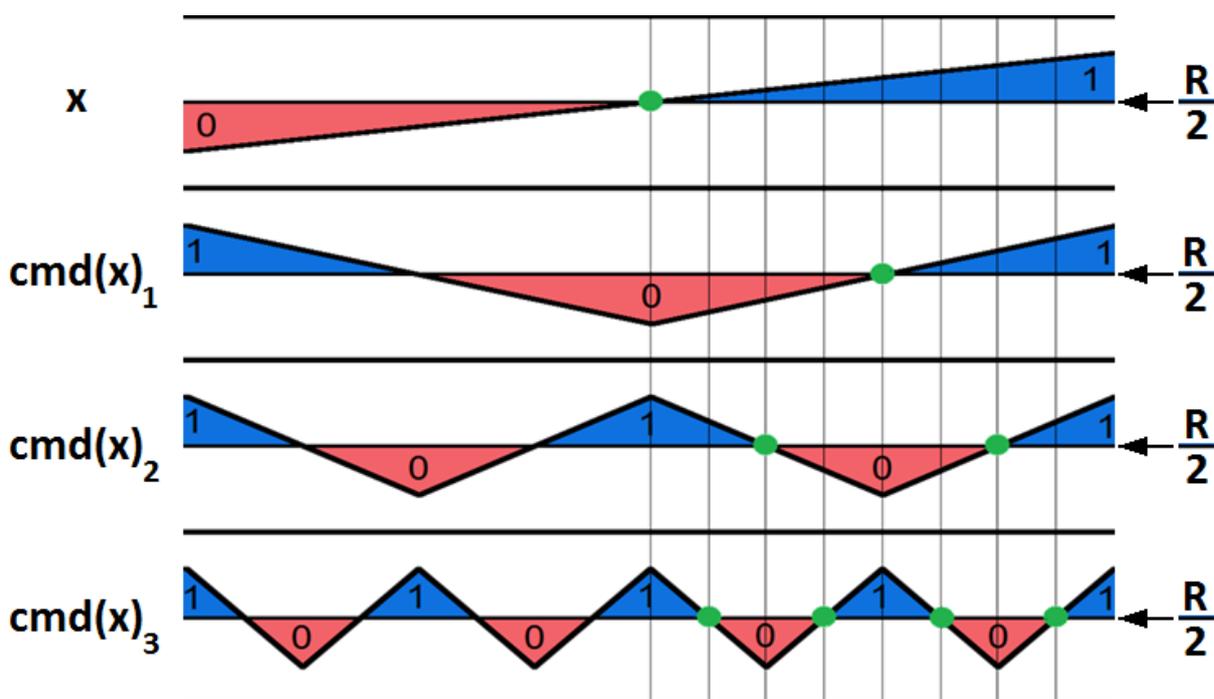


Imagem 4.3-1: Interpretação gráfica do modelo matemático

Na qual " $CMD(x_{in})_0$ " é o bit mais significativo; " $\llbracket x \rrbracket$ " é a função menor inteiro, cuja saída é o menor inteiro "maior ou igual" a " $x$ "; " $R$ " é o valor máximo da entrada; " $CMD(x)_n$ " corresponde ao valor do " $n$ -ésimo" bit; " $I$ " representa o conjunto dos Inteiros; e " $x_{in}$ " é o sinal analógico que será convertido. Para representar corretamente o comportamento do "SAR" o valor de " $R$ " é "1" e " $x_{in}$ " é positivo menor ou igual a " $R$ ", portanto:

$$R = 1 \text{ e } x_{in} \in (0,1) \quad (4.3-6)$$

Dessa forma a entrada para a função para a função " $\llbracket x \rrbracket$ " receberá apenas valores com módulo menor que "0,5" e, portanto, a saída da função será "0" ou "1", respectivamente para valores negativos e para valores positivos.

Na imagem 4.3-1 observamos o comportamento através das retas verticais traçadas de forma secante ao ponto de variação do valor de determinado bit, destacado pelos círculos verdes, não foram passadas retas ao longo de toda a imagem para evitar poluição e facilitar o entendimento, mas o padrão de distribuição seria o mesmo, uma vez que as funções são pares e, portanto, simétricas em relação ao eixo vertical central.

Essa característica implica em menor capacidade de operação para a fonte de alimentação, uma vez que não será solicitada por diversos circuitos simultaneamente, ou seja, a demanda de consumo é distribuída e otimizada. Veja que pelas faixas podemos observar o comportamento dos

valores, que estão fluindo da direita para a esquerda na sequência de “1111”, “1110”, “1100”, “1101”, “1001”, “1000”, “1010” e “1011”. A sequência vinda da esquerda para a direita é a mesma para os três bits menos significativos, mas o valor para o bit mais significativo é “0”, ou seja, o comportamento dos bits é espelhado em relação ao eixo central, conforme veremos adiante. Dessa forma, a sequência de dos valores dos bits vindos da esquerda para a direita até o 8º byte seria “0111”, “0110”, “0100”, “0101”, “0001”, “0000”, “0010” e “0011”. Como é verificado, se comporta de forma particular.

#### 4.4. COMPORTAMENTO BINÁRIO

Vimos que o CMD armazena os valores binário em um formato próprio, pois não houve preocupação em organizar os valores convertidos, mas apenas em obter um padrão digital para um sinal analógico de interesse, sendo essa ideia presente também nos conversores Flash e Sigma Delta. Esse comportamento diferenciado ocorre por obtermos o valor do módulo, que é um processo realizado para eliminar a necessidade de tomada de decisão do SAR. Dessa forma, surge a necessidade de interpretar / traduzir a informação gerada, o que já pode ocorrer no domínio digital e, portanto, não causa atrasos no processo de conversão.

Para entendermos o comportamento dos bits gerados, tomemos como exemplo um conversor de 4 bits. Na imagem 4.4-1 apresentamos a relação de correspondência entre os valores binários e as faixas de valores analógicos, ainda no formato binário padrão, a ideia é comparar e definir a lógica necessária para gerar uma correspondência entre os valores gerados pelo conversor e os números na base 2.

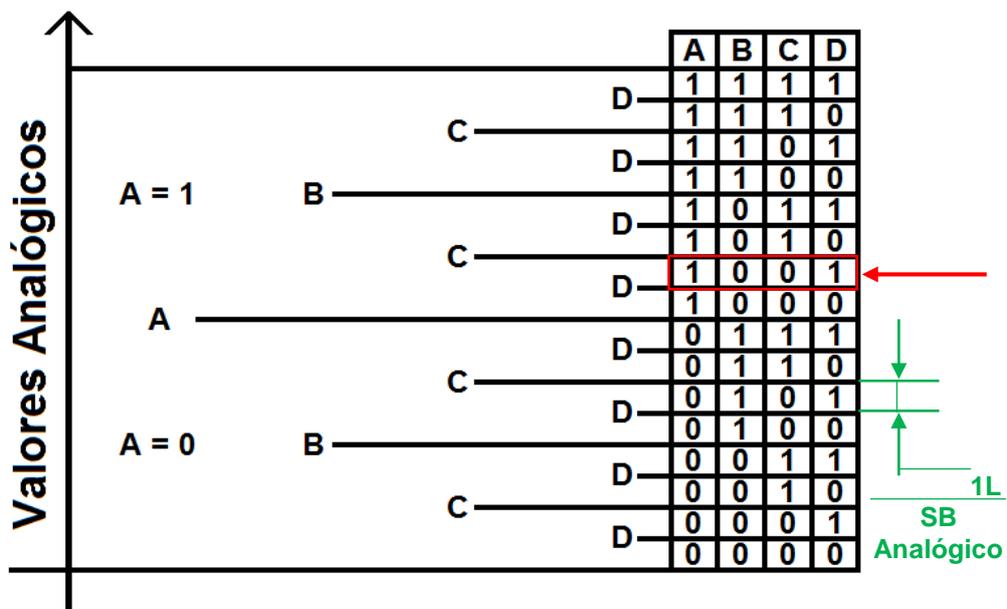


Imagem 4.4-1: Divisão das faixas Analógicas

As letras representam as palavras de 4 bits, ou seja, sendo o bit representado pela letra “A”, o mais significativo e o representado pela letra “D”, o menos significativo. Conforme a imagem acima,

pode-se observar que o bit “A” divide a faixa analógica em duas faixas iguais (a superior é a faixa determinada por “A = 1” e a inferior é definida por “A = 0”). O bit “B” divide as faixas determinadas por “A” em outras duas faixas e esse processo ocorre de forma sucessiva para cada bit.

Para observar e entender o comportamento do CMD, é necessário considerar o sinal analógico em cada uma dessas faixas e o valor binário correspondente a cada faixa. O novo conversor é viável, devido ao fato de cada uma das faixas, analógicas, possuir apenas um valor binário correspondente à essa faixa, ou seja, partindo de cada uma das faixas analógicas de interesses, haverá apenas um valor binário correspondente. Esse valor binário também corresponde apenas à mesma faixa analógica correspondida, ou seja, trata-se de uma correspondência biunívoca entre uma faixa de valores analógicos e um único valor digital. Se essa propriedade não fosse atingida, poderia ser necessário amostrar o valor analógico com um número maior de bits que o correspondente à resolução desejada ou até inviabilizar a arquitetura. No caso da sobre-amostragem, para processar o sinal e determinar o valor desejado, seriam descartando os bits excedentes, de forma similar ao Sigma Delta.

#### 4.4.1. GERANDO OS BITS

Palavra no formato binário padrão	Entrada do Bloco 1 (LSB analógico)			Entrada do Bloco 2 (LSB analógico)			Entrada do Bloco 3 (LSB analógico)			Entrada do Bloco 4 (LSB analógico)			Saída do Bloco 4 (LSB analógico)	Palavra no formato binário do CMD		
"1111"	15	até	16	14	até	16	12	até	16	8	até	16	0	até	16	"1111"
"1110"	14	até	15	12	até	14	8	até	12	0	até	8	0	até	16	"1110"
"1101"	13	até	14	10	até	12	4	até	8	0	até	8	0	até	16	"1100"
"1100"	12	até	13	8	até	10	0	até	4	8	até	16	0	até	16	"1101"
"1011"	11	até	12	6	até	8	0	até	4	8	até	16	0	até	16	"1001"
"1010"	10	até	11	4	até	6	4	até	8	0	até	8	0	até	16	"1000"
"1001"	9	até	10	2	até	4	8	até	12	0	até	8	0	até	16	"1010"
"1000"	8	até	9	0	até	2	12	até	16	8	até	16	0	até	16	"1011"
"0111"	7	até	8	0	até	2	12	até	16	8	até	16	0	até	16	"0011"
"0110"	6	até	7	2	até	4	8	até	12	0	até	8	0	até	16	"0010"
"0101"	5	até	6	4	até	6	4	até	8	0	até	8	0	até	16	"0000"
"0100"	4	até	5	6	até	8	0	até	4	8	até	16	0	até	16	"0001"
"0011"	3	até	4	8	até	10	0	até	4	8	até	16	0	até	16	"0101"
"0010"	2	até	3	10	até	12	4	até	8	0	até	8	0	até	16	"0100"
"0001"	1	até	2	12	até	14	8	até	12	0	até	8	0	até	16	"0110"
"0000"	0	até	1	14	até	16	12	até	16	8	até	16	0	até	16	"0111"

Imagem 4.4.1-1: Comportamento dos bits para as faixas analógicas

Primeiro consideraremos a faixa definida pela palavra “1001”, destacada pelo retângulo vermelho e seta na imagem 4.4.1-1. Veja na tabela que o valor contido nessa faixa varia dentro dos

valores analógicos pertencentes ao intervalo de 9 a 10 comprimentos de 1 LSB (o valor do LSB em questão está se referindo à faixa analógica definida por 1 LSB, conforme destacado na imagem 4.4-1). Assim, o menor valor analógico da faixa analisada é sempre maior que 8 LSB. Os valores analógicos possíveis para a entrada variam de 0 a 16 LSB e a referência que divide a faixa em duas faixas de mesmo tamanho corresponde a 8LSB. Assim, conforme a seção 4.3 do modelo matemático, a referência para comparação do modelo, conforme a equação 4.3-1 é de 8LSB.

Com a intensidade do sinal da faixa correspondente ao valor binário “1001” variando de 9 a 10 LSB, quando comparada à 8 LBS, a saída do comparador é 1. O valor do módulo variará de “1” “LSB” até “2” “LSB”, mas esse valor será multiplicado por dois, conforme as equações 4.3-1 e 4.3-3, gerando o cálculo da equação 4.4-1.

$$[cmd(9) = 2 * |9 - 8| = 2] \text{ até } [cmd(10) = 2 * |10 - 8| = 4] \quad (4.4-1)$$

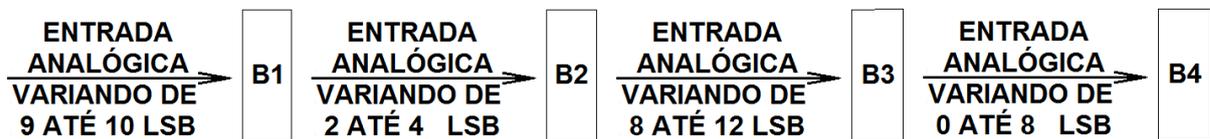


Imagem 4.4.1-2: Dado analógico fluindo de Bloco a Bloco.

Retomando ao processo de análise, do comportamento dos “bits”, na faixa da palavra “1001”. Considerando que a informação analógica flui de bloco a bloco do primeiro Bloco (B1) até o quarto Bloco (B4), da imagem 4.4.1-2. O valor de saída do primeiro circuito de módulo possuirá intensidade variando de 2 a 4 “LSB” analógicos, que é menor que 8 “LSB”, e, portanto, o segundo “bit” do novo conversor também seria “0”. Aplicando a equação “4.3.3” temos as saídas que serão as entradas para o 3º Bloco.

$$[cmd(2) = 2 * |2 - 8| = 12] \text{ até } [cmd(4) = 2 * |6 - 8| = 8] \quad (4.4-2)$$

Portanto, o valor da saída do segundo bloco variará de “8” à “12”. O processo é o mesmo para o terceiro e o quarto blocos, sendo a variação dos valores de saída de cada bloco, respectivamente, inscritas nas faixas de “0” à “8” e de “0” à “16”. Assim, a intensidade da corrente de saída do segundo bloco é maior que 8 e é a entrada do terceiro bloco, que terá o valor de bit em 1. Para o quarto bloco, portanto, teríamos como entrada a saída do terceiro bloco, então sua entrada variaria de 0 a 8 e, portanto, o bloco definiria o bit com o valor de 0. Dessa forma, poderíamos continuar o processo até obtermos como resultado a tabela da imagem 4.4.1-1 do início dessa seção.

$$[cmd(12) = 2 * |12 - 8| = 8] \text{ até } [cmd(8) = 2 * |8 - 8| = 0] \quad (4.4-3)$$

$$[cmd(8) = 2 * |8 - 8| = 0] \text{ até } [cmd(0) = 2 * |0 - 8| = 16] \quad (4.4-4)$$

## 4.4.2. PARTICULARIDADES DOS VALORES BINÁRIOS DO CMD

### 4.4.2.1. ESTABILIDADE ENTRE PALAVRAS DIGITAIS

Para entendermos essa estabilidade, consideremos um conversor SAR sem um S/H. Quando o ADC SAR fosse do valor “0111” para o valor “1000”, o primeiro bit a ser alterado seria o mais significativo, portanto, a palavra passaria a ser “1111”, depois o segundo, acarretando na palavra “1011”. Dessa forma, em resumo, os valores apresentados seriam respectivamente do “0111”, para o “1111”, para o “1011”, para o “1001” e, finalizando, no “1000”. Em termos decimais teríamos apresentados os valores de 7, para 15, para 11, para 9 e, finalmente, o 8. Esse mesmo processo no “CMD” representaria a variação direta da palavra “0011” para a palavra “1011”, ou seja, variaria apenas um bit, o que, em termos decimais, implicaria na variação de “7” para “8”, considerando a correspondência dos valores gerados pelo CMD com os valores binários. Isso também mostra um gasto menor de energia, uma vez que há apenas uma variação e não 4 (quatro). Na Imagem 4.4.2-1 o comportamento à esquerda é o do CMD para um sinal analógico variando de 0 ao fundo de escala linearmente e, do lado direito, o comportamento dos bits do SAR para o mesmo sinal.

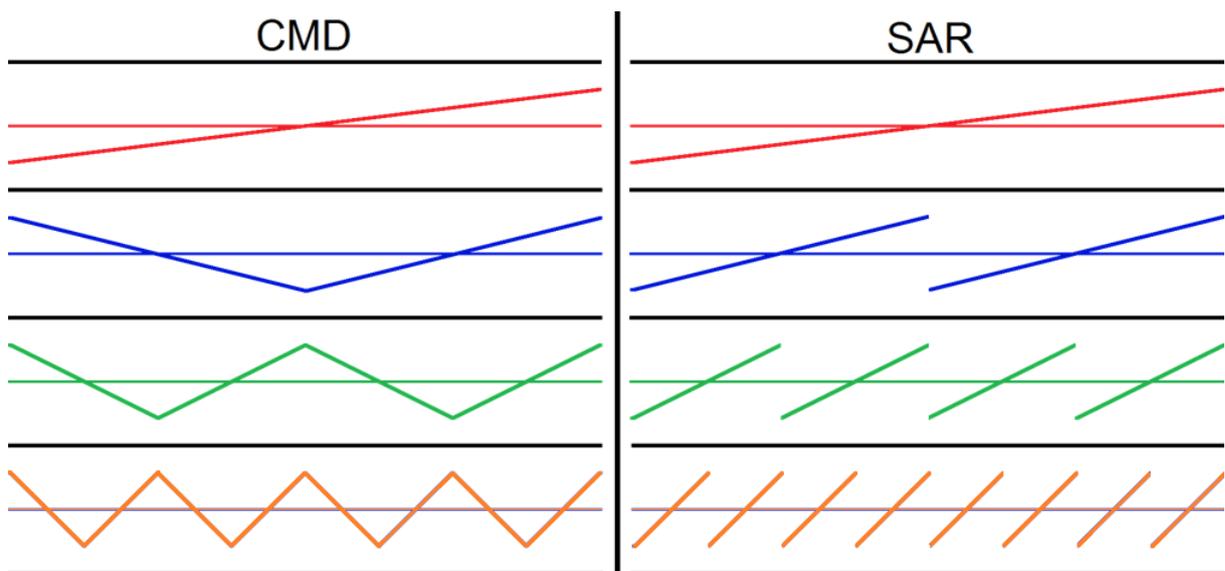


Imagem 4.4.2-1: Comportamento dos bits para as faixas analógicas

O primeiro sinal na parte superior da imagem 4.4.2-1 é o mais significativo, tanto para o SAR, quanto para o CMD. Verificamos pela imagem que o sinal do bit mais significativo é o mesmo para os dois conversores. Para o segundo bit mais significativo (curvas azuis) é descontínuo no SAR e a alteração de seu valor implica na alteração do valor de todos os bits menos significativos na sequência. Em contraste, do lado esquerdo, apenas um bit altera seu valor de cada vez, considerando a alteração do valor quando uma curva variável transita por valores em torno da curva horizontal. A continuidade do sinal gerado por cada bloco do CMD gera mais estabilidade no processo de conversão, que está

associado ao fato de o conversor não alterar dois bits simultaneamente em nenhuma transição dentre valores binários adjacentes quaisquer como vimos na seção 4.3 do modelo matemático.

Essa característica simplifica o circuito do conversor para processos de medição de temperatura e outros sinais de baixa frequência como sinais biológicos, por exemplo, temperatura, frequência cardíaca, respiração e outros. O conversor ocuparia uma área menor em silício, pois não possuiria um capacitor e possuiria alta resolução com poucas oscilações no processo, consumindo menos da fonte (deve-se ressaltar que há um pico de consumo durante as transições digitais e mudança de estados analógicos de trípode, saturação e corte dos MOSFETs).

#### 4.4.2.2. SIMETRIA

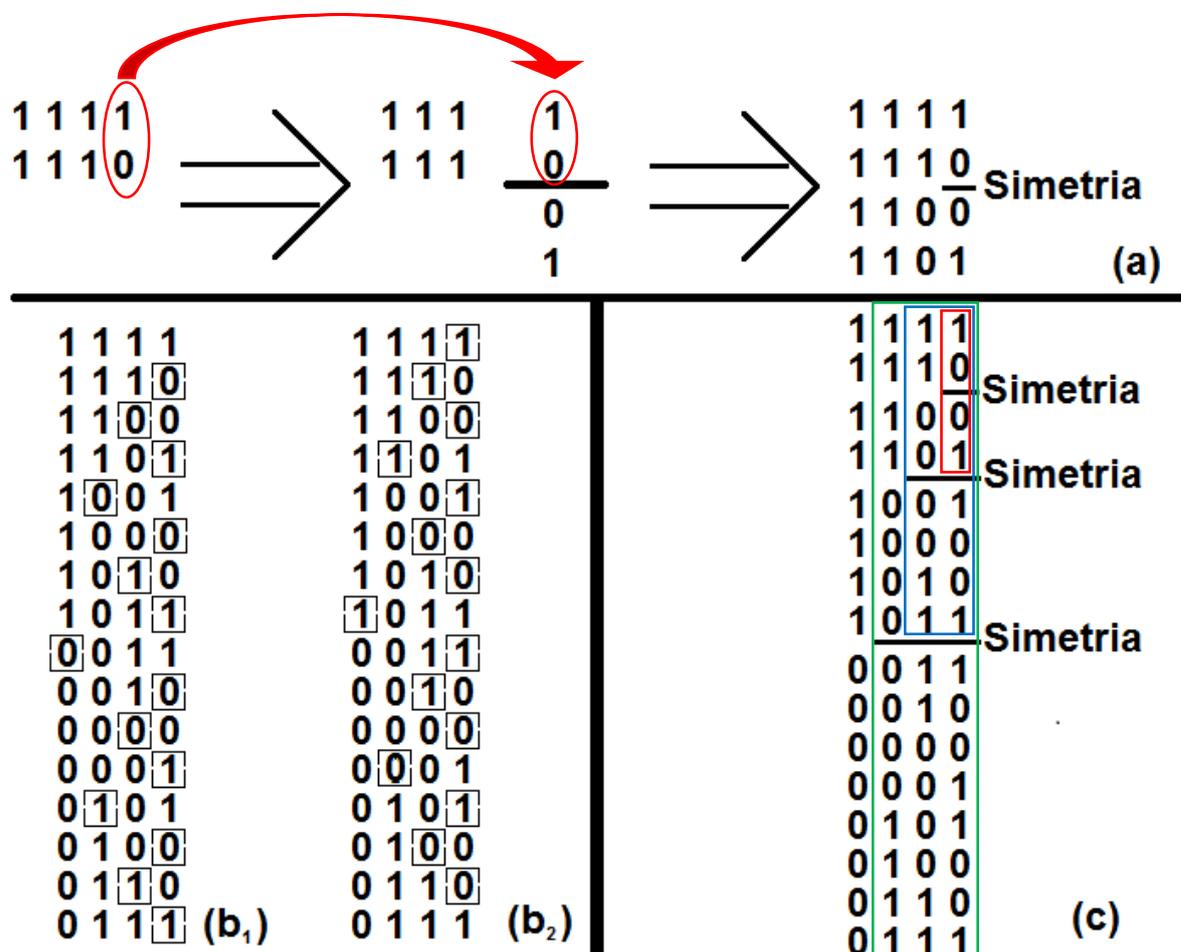


Imagem 4.4.2.2-1: Comportamento simétrico dos bits

Outra particularidade é o fato das sequências serem funções pares de comportamento dos bits, portanto, os bits se comportam como um reflexo dentre as palavras digitais, conforme a Imagem 4.4.2-2. A simetria facilita a previsão do comportamento dos "bits" e também a geração de sinais para a realização dos processos de simulação em ambiente de projetos para silício. O fato de alterar apenas um bit a entre valores adjacentes gera sinais estáveis. Veja que até o bit menos significativo não sofre alteração a cada transição. A simetria se repete dentro de cada família de retângulos em destaque na

imagem 4.4.2.2-1. O retângulo vermelho e menor contém um eixo de simetria, ou seja, o comportamento da parte superior é o reflexo da parte inferior, "10|01". O mesmo processo se repete nos outros dois retângulos em destaque, o azul (segundo menor) e o verde (retângulo maior) na imagem 4.4.2-2.

### 4.4.2.3. CODIGO DE GRAY

O processo realizado pelo novo conversor é similar ao processo realizado pelo código de Gray, que é mais simples de processar que o código binário habitual, possui apenas um bit variando a cada transição de valores e / ou estado lógico.

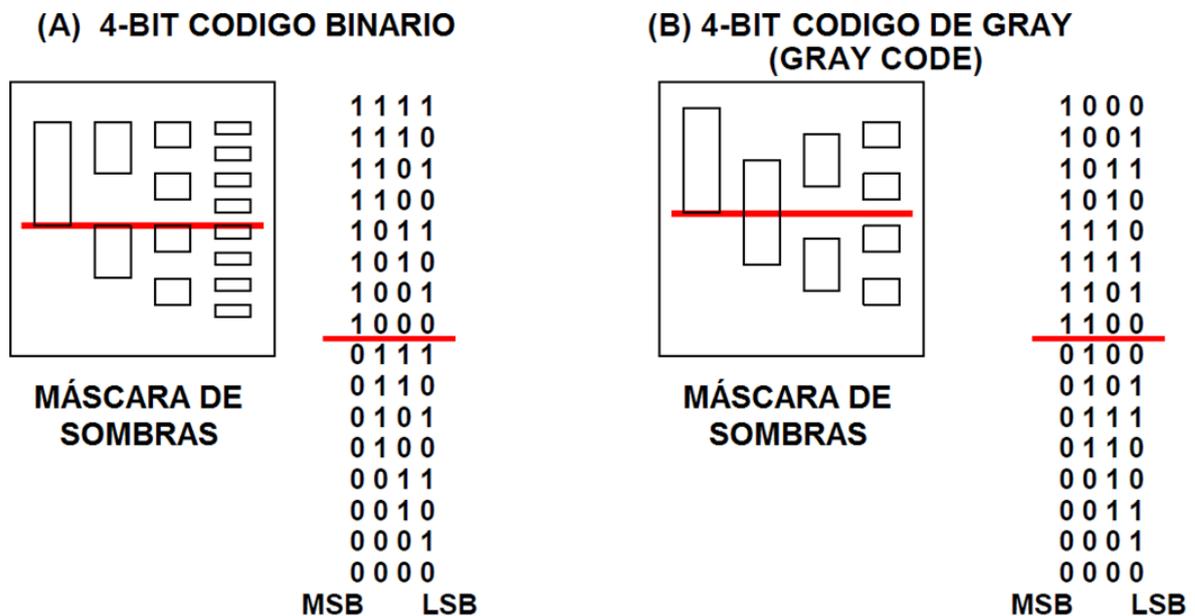


Imagem 4.4.2.3-1: Código Binário e Código de Gray. Retirada de [2].

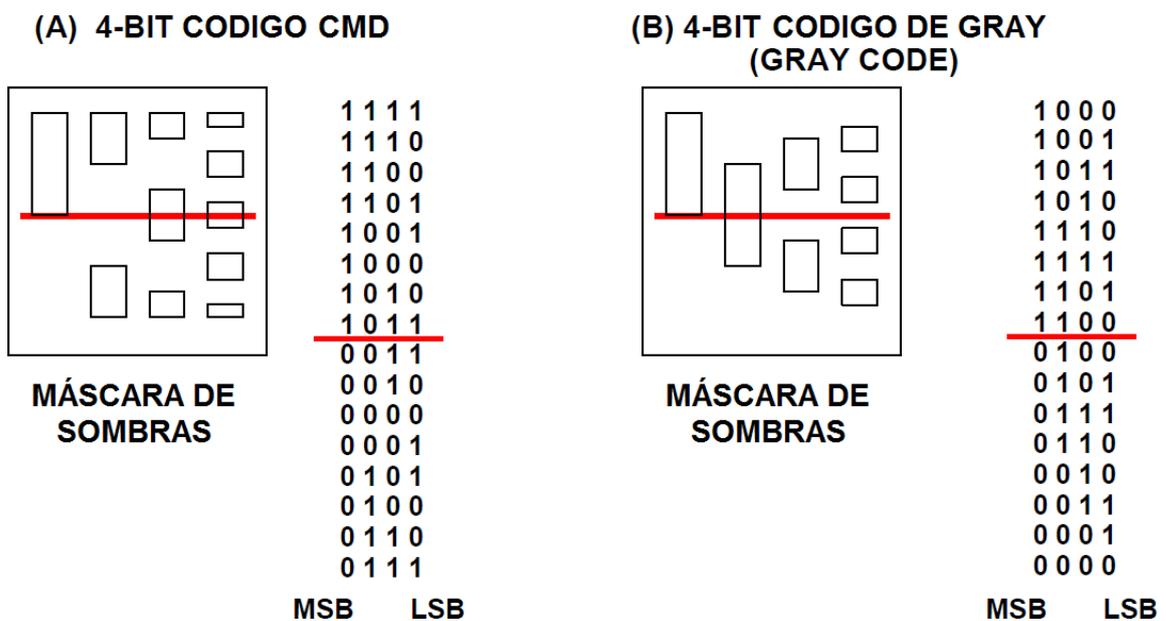


Imagem 4.4.2.3-2: Código Binário e Código de Gray.

Note pelas imagens que a que os três códigos possuem exatamente o mesmo comportamento para o primeiro bit, ou seja, o mais significativo. Entretanto, o código de Gray e o CMD são completamente compatíveis em relação ao comportamental entre eles, pois, caso sejam negados todos os bits do CMD, exceto pelo primeiro, os códigos se tornam iguais. Dessa forma, todas as particularidades do código de Gray são aplicáveis ao CMD.

#### 4.4.3. LÓGICA DO TRADUTOR

Com o mapeamento do comportamento dos bytes gerados pelo conversor “CMD” e a verificação de um padrão que é extensivo a novos bits a serem adicionados, pode-se definir qual o circuito digital necessário para converter as palavras do “CMD” no formato binário padrão. Conforme é observado, o comportamento dos demais bits vem da variação do mais significativo, ou seja, as duas faixas divididas pelo bit mais significativo adjacente implicam no comportamento simétrico do bit menos significativo adjacente, conforme visto no parágrafo anterior.

SAÍDA BINÁRIO		1	2	3	4	5																							
A	B	C	D	E	a	b	c	d	e	A	a	A	B	b	A	B	C	c	A	B	C	D	d	A	B	C	D	E	e
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1				
1	1	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
1	1	1	1	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
1	1	0	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
1	1	0	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
1	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0				
0	0	0	0	1	1																								

mesma lógica é observada para a correspondência dos valores de “d” e para os valores de “e”. Na imagem 4.4-6 temos os circuitos lógicos para obter cada bit traduzido.

O circuito lógico, portanto, para cada bit, é uma “XNOR” quando a posição do bit corresponde a uma posição par, ou seja, o 2º, o 4º, o 6º e etc. Quando o valor da posição do bit é um valor ímpar, a porta lógica correspondente é uma “XOR”. As entradas das portas lógicas são os valores dos “bits” de mesma significância com os de maior significância em relação ao valor do bit desejado, ou seja, para obter o valor correspondente do enésimo bit, será dado pela equação lógica:

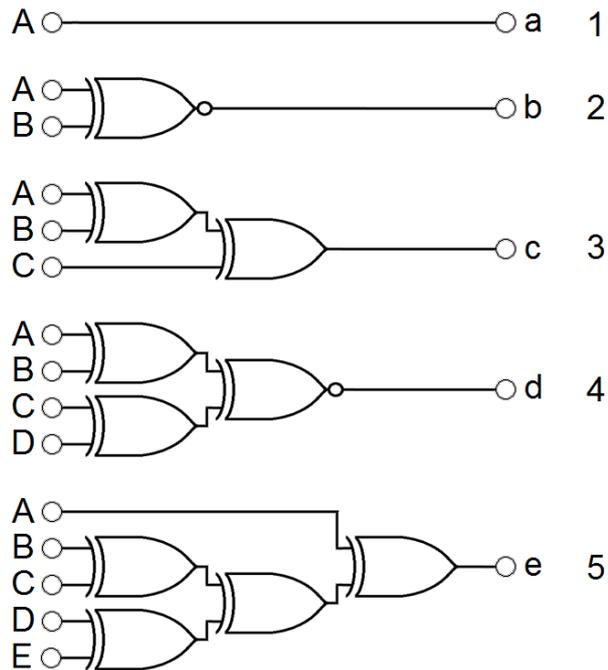


Imagem 4.4.3-2: Circuito Não Simplificado de 5 bits para o Tradutor

O circuito lógico pode ser obtido por portas lógicas do tipo XOR e negando a saída dos circuitos quando necessário. Os números em 4.4.3-2 estão associados aos números da tabela da imagem 4.4.3-1. Utilizando apenas portas XNOR temos:

A	B	XNOR
1	1	1
1	0	0
0	1	0
0	0	1

Imagem 4.4.3-3: Tabela Verdade da Porta Lógica XNOR

A	B	XNOR <sub>1</sub>
1	1	1
1	0	0
0	1	0
0	0	1

→

XNOR <sub>1</sub>	C	XNOR <sub>2</sub>
1	1	1
1	0	0
0	1	0
0	0	1

→

A	B	XNOR <sub>1</sub>	C	XNOR <sub>2</sub>
1	1	1	1	1
1	1	1	0	0
1	0	0	1	0
1	0	0	0	1
0	1	0	1	0
0	1	0	0	1
0	0	1	1	1
0	0	1	0	0

→

A	B	C	XOR
1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	1
0	0	1	1
0	0	0	0

Imagem 4.4.3-4: Resposta Lógica da associação em série de duas XNOR

$$b_n = XNOR(B_n; b_{n-1}) \tag{4.4-5}$$

Na equação, “ $b_n$ ” é o  $n$ ésimo bit a ser traduzido para a forma binária padrão, ou seja, um número binário conforme o formato padrão; “XNOR(A;B)” é a porta lógica “XOR(A;B)” negada; “ $B_n$ ” é o  $n$ ésimo bit gerado pelo conversor sem nenhum processamento; e “ $b_{n-1}$ ” é o bit traduzido mais significativo e adjacente ao “ $b_n$ ”. Ainda, conforme as tabelas, da imagem “4.4-8”, pode-se verificar a correspondência lógica.

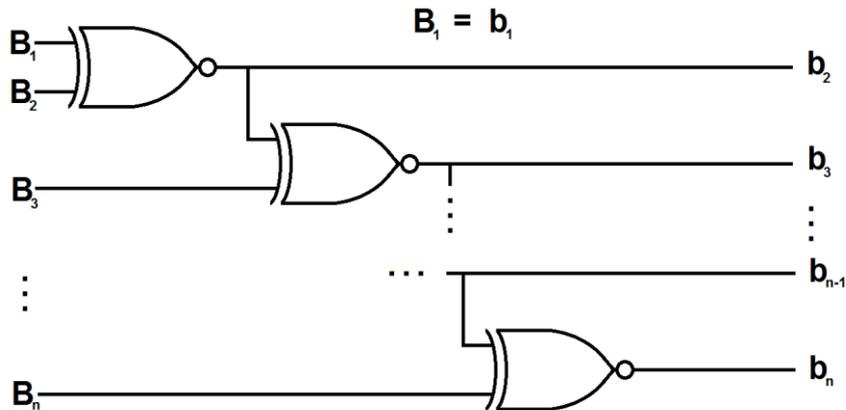


Imagem 4.4.3-5: Circuito Digital do Tradutor

$$XOR(A; B; C) = XNOR(XNOR(A; B); C) \tag{4.4-3}$$

Dessa forma, é satisfeita a condição lógica final, ou seja, para os bits pares, a serem gerados, é necessária a porta “XNOR” e, para os ímpares, é necessária a porta lógica “XOR”. Essa propriedade permite que o circuito do conversor aumente de forma linear conforme é adicionado um novo bit, tanto o circuito analógico, que já é composto por uma sequência de circuitos analógicos equivalentes, quanto para os circuitos digitais, conforme a imagem “4.4.3-5”, similar ao circuito de [12] para o código de Gray.

Conforme a seção 4.3, o comportamento dos bits poderia ser exatamente igual ao comportamento do código de Gray e, para isso, seria necessário apenas inverter as entradas o comparador, entretanto, qual seria o comportamento mais desejado? Aplicando o mesmo princípio que

levou à simplificação do circuito digital apenas para portas lógicas XNOR, ou seja, ao inverter uma das entradas da porta XOR obtemos uma porta XNOR, portanto, de forma correspondente, ao invertermos uma das entradas de uma porta XNOR, voltaremos à correspondência anterior, portanto, será obtida uma porta lógica XOR.

O restante dessa análise está na seção 5, na qual são apresentados os blocos e circuitos do projeto. Na seção seguinte veremos que o circuito das portas lógicas XNOR é mais simples que o das portas lógicas XOR. Devido à grande simplicidade dos circuitos analógicos que concebem esses circuitos digitais, o circuito mais simples geralmente é mais rápido, mais econômico e reduz a área em SoC.

## 4.5. MODELO EM SOFTWARE

O Software foi desenvolvido em “JAVA” com a ferramenta de desenvolvimento “Blue J” da “University of Kent”. A ideia é uma validação lógica.

### 4.5.1. O SOFTWARE

O conversor utilizado como maior referência para comparativo em software foi o “SAR”, pois se trata de um conversor com modelamento simples e normalmente utilizado como paradigma nos diversos livros de inicialização na área. O SAR também possui a característica de gerar os valores a partir de referências simples, o que é similar ao processo do CMD.

O CMD não utiliza S/H nesse trabalho, mas não há óbice para tal, além do fato de que descartar o S/H ocasiona vantagens como a redução do ruído gerado pelo conversor, de consumo, de trabalho com componentes sempre estabilizados, reduzindo o atraso para os processos de transição entre valores binários e dispensa o uso de capacitores para registrar os valores analógicos, o que reduz a área em SoC e também facilita o processo de validação do bloco devido à ausência de capacitores em todos os blocos do conversor. Para o SAR modelado, não está sendo considerada a existência de um S/H, portanto, a simulação também mostrará a vantagem da simetria entre as faixas adjacentes para os valores dos bits gerados pelo conversor, que vimos na seção 4.4.2.1.

O algoritmo do software que trata da simulação é bem simples:

(...)

***For (int i = 0; i < AMOS; i++){ → Repete o processo até gerar todo o array gráfico.***

Criando senóide para entrada

***IN[i] = 200000\*(1+Math.sin((Math.PI)\*(a/10000)\*i));***

Realizando a subtração para referência e lógica para o módulo. O valor de “200000” é a referência e também dá a coordenada para impressão do gráfico. O processo é repetido para todos os bits e, por isso, simplificamos.

```

IAC = IN[i] - 200000;
if(IAC >= 0){
  A = 1;//1 Lógico
}else{
  A = 0;//0 Lógico
  IAC = -IAC;
}
AX1 = 2*AX1 - 200000;
if(AX1 >= 0){
  B = 1;
}else{
  B = 0;
  AX1 = -AX1;
}
  (...)

```

Os valores são atualizados a cada ciclo de emulação para caracterizar o atraso dos espelhos

```

AX5 = AX4; → Sinal atual recebe o sinal do ciclo anterior e assim por diante.
AX4 = AX3;
AX3 = AX2;
AX2 = AX1;
AX1 = IAC; → Note que IAC é a entrada e só está chegando no primeiro bloco.

```

Processo do Tradutor para gerar a onda convertida. Também é repetido e por isso simplificamos. A ideia é verificar se o bit anterior está espelhando o não o bit atual.

```

if(A == B){
  BB = 1;
}else{
  BB = 0;
}
if(BB == 1){
  BC = C;
}else{
  if(C == 1){
    BC = 0;
  }else{
    BC = 1;
  }
}
  (...)

```

Atualizamos novamente a entrada para iniciar o processo com o SAR.

```

IAC = IN[i];

```

Como o bit menos significativo a cada ciclo apresentará seus valores de acordo com o sinal dos outros bits. Ele é o primeiro a ser atualizado. Assim, a equação já considera todas as comparações.

```

if((IAC - ASAR6*200000 - BSAR5*100000 - CSAR4*50000 - DSAR3*25000 -
ESAR2*12500 - 6250) > 0){
  FSAR = 1;
}else{
  FSAR = 0;
}

```

Atualizando os bits para a conversão de do próximo bit e mais significativo

```

ASAR6 = ASAR5; → Sinal atual recebe o sinal do ciclo anterior e assim por diante.

```

$BSAR5 = BSAR4;$   
 $CSAR4 = CSAR3;$   
 $DSAR3 = DSAR2;$   
 $ESAR2 = ESAR;$

(...)

O processo é repetido para cada bit mais significativo na sequência. Assim, os valores são somados e adicionados à uma posição do vetor para exibição gráfica

$$S1[i] = (16*A + 8*BB + 4*BC + 2*BD + BE + BF/2); \quad \text{CMD}$$

$$S3[i] = (16*ASAR + 8*BSAR + 4*CSAR + 2*DSAR + ESAR + FSAR/2); \quad \text{SAR}$$

#### 4.5.2. RESULTADOS DO SOFTWARE

Note que, devido à lógica do SAR ser baseada na alteração dos bits conforme o processo de geração do valor binário matemático. O sinal sofrer grandes oscilações durante a alteração dos bits do conversor. Caso a lógica para os valores binários do SAR já fossem alterados, para garantir maior estabilidade durante o processo de conversão, o sinal seria mais estável, mas ainda haveria a questão do atraso devido ao período de comparação e estabilização dos valores, o que não está sendo levado em conta no processo simulado. A primeira imagem mostra um processo para um sinal lento em relação ao atraso de estabilização dos blocos dos conversores simulados, entretanto, vejamos como ambos os sinais são prejudicados com o aumento da frequência do sinal de entrada, mas veja também que o CMD resiste de forma mais eficiente.

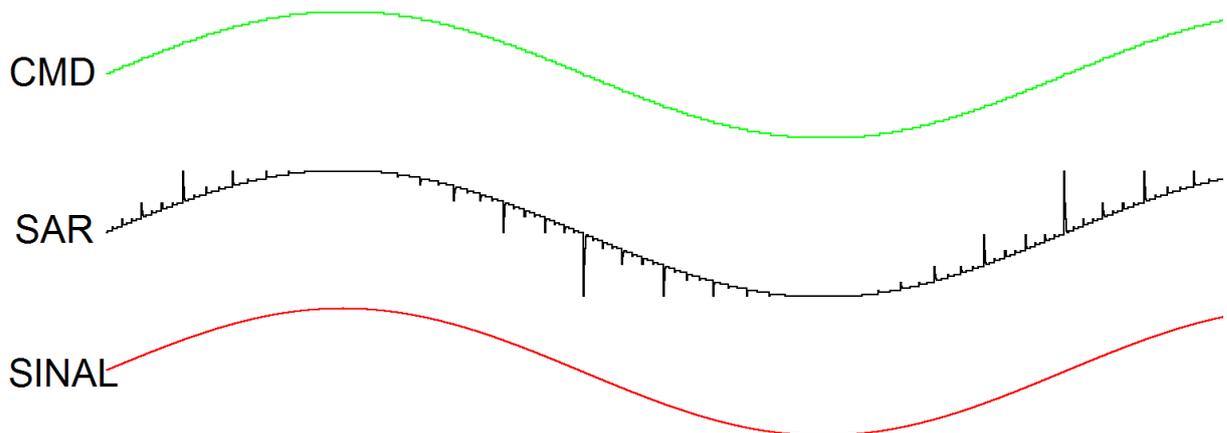


Imagem 4.5.2-1: Simulações em software dos conversores SAR e CMD baixa frequência.

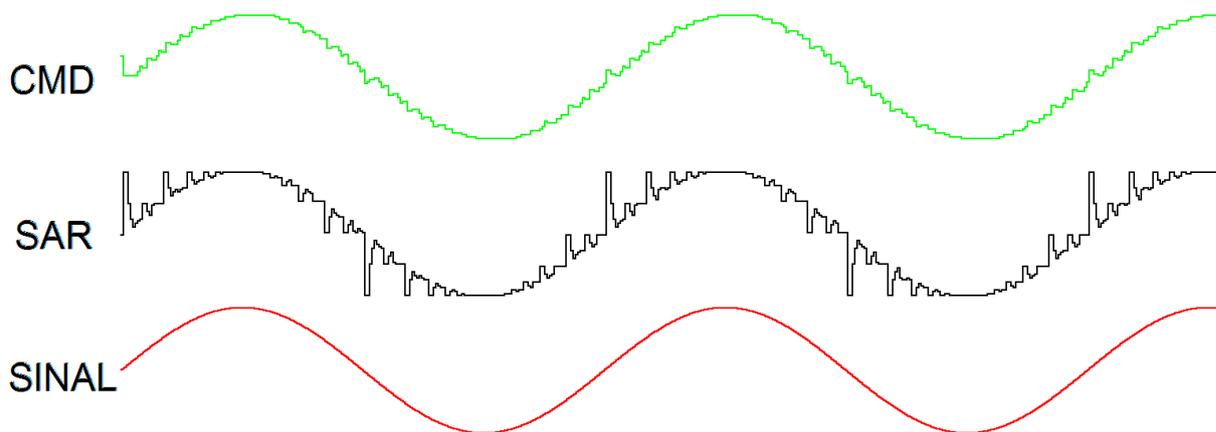


Imagem 4.5.2-2: Simulações em software dos conversores SAR e CMD em maior Frequência.

Ambos os conversores são prejudicados, mas a questão é qual o conversor que resiste mais ao processo degenerativo do sinal devido à impossibilidade de o conversor acompanhar a frequência do sinal. Veja na imagem que os bits do CMD são mais estáveis e não sofrem alteração ao mesmo, mas no caso do SAR é frequente a alteração de diversos bits simultaneamente, por isso a geração de ruídos na ausência de um S/H a cada transição de determinados valores binários

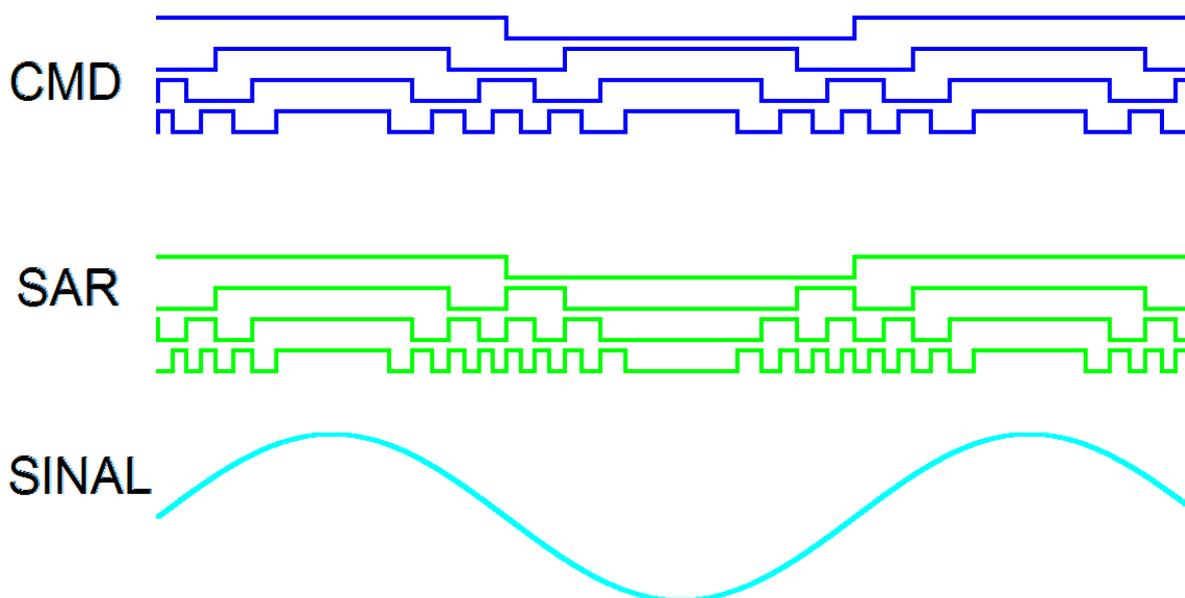


Imagem 4.5.2-3: Simulações do Comportamento Binário.

Na imagem 4.5.2-3 temos os valores binários gerados para um sinal senoidal utilizado como entrada para o software. Os gráficos foram obtidos pela biblioteca de Applet do Java.

#### 4.6. BLOCOS BÁSICOS

Recordemos à seção 4.3 do modelo matemático com as equações 4.3-1, 4.3-2, 4.3-3 e 4.3-4, respectivamente,  $\text{cmd}(x)_1 = 2 * \left\lfloor x - \frac{R}{2} \right\rfloor$ ;  $\text{cmd}(x)_2 = 2 * \left\lfloor \text{cmd}(x)_1 - \frac{R}{2} \right\rfloor$ ;  $\text{cmd}(x)_n = 2 * \left\lfloor \text{cmd}(x)_{n-1} - \frac{R}{2} \right\rfloor$ ; e  $\text{CMD}(x_{in})_n \{ \text{Se, } \text{cmd}(x)_n > 0 ; 1 \mid \text{Se, } \text{cmd}(x)_n < 0 ; 0$ . Essas equações servirão para fundamentar a seção 4.6.1. Na equação 4.3-1 temos o primeiro bit, que recebe o sinal customizado por um bloco PGA. Dessa forma o conversor CMD foi dividido em três blocos o do PGA, o Bloco CMD e o bloco TRADUTOR que vimos na seção 4.4.

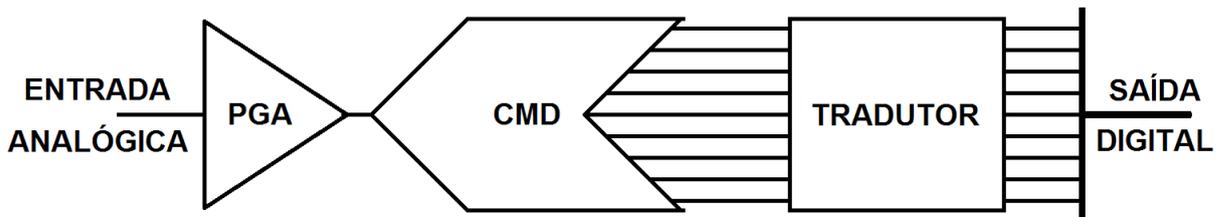


Imagem 4.6-1: Blocos Básicos do Conversor

#### 4.6.1. Bloco CMD

O Bloco “CMD” é a estrutura de maior interesse desse trabalho, pois é o bloco responsável pelo novo processo de conversão de sinais analógicos para sinais digitais. O bloco é dividido em três vetores de processos com o objetivo de reproduzir o modelo matemático. As equações realizam as operações de:

- A. Subtração de uma referência fixa;
- B. Módulo do resultado da subtração; e
- C. Multiplicação do resultado do módulo por dois.

Uma decisão de projeto foi implementar um conversor baseado em modo corrente, ou seja, o sinal de interesse é um sinal de corrente. Como em geral o conversor converte um sinal de tensão, o bloco do PGA recebe uma entrada de tensão e entrega um sinal de corrente dentro dos níveis adequados para o bloco CMD.

Com a escolha de operar em modo corrente, a solução do processo de subtração da letra “A” é apenas uma fonte de corrente que retira uma corrente fixa que serve de referência para o processo.

Para obter o módulo da corrente, note que após o processo “A”, o resultado é “negativo” ou “positivo”, mas isso é uma definição que devemos criar dentro do circuito. Assim, em nosso projeto, consideramos como positiva a corrente quando ela é injetada e negativa quando é retirada, ou seja, é relativa à sequência do circuito. Então, precisamos de um circuito que tenha a sua entrada conectada ao ponto em que ocorre a retirada da corrente de referência. Então a corrente positiva é injetada e se assim, **VAI** de um maior para um menor potencial elétrico, de forma inversa para o sinal negativo, ou seja, **VEM** de um maior para um menor potencial elétrico. Então temos dois caminhos: um que vem e

outro que vai de um maior para um menor potencial. No caso positivo o potencial do ponto de subtração é o maior potencial e, portanto, entregará corrente no sentido do GND (negativo da fonte de tensão) do circuito e no caso negativo está em menor potencial e retirará a corrente do VDD (positivo da fonte de tensão). Então, teremos dois caminhos para cada sinal e, portanto, por trabalharmos em modo corrente, basta espelhar essa corrente de modo a entregarmos a corrente, pois queremos que a saída seja sempre positiva, ou seja, o valor absoluto do sinal.

Assim, o processo “B” será obtido por um circuito de dois caminhos associado a um circuito que conduzirá os caminhos a um mesmo sentido de saída. O primeiro circuito / Bloco será chamado de By-Pass e o segundo de Espelhos. O processo “C” passa a ser a simples multiplicação por dois da largura dos dispositivos, que copiam a corrente dos caminhos positivos e negativo para a saída. Satisfazendo as equações 4.3-1, 4.3-2 e 4.3-3 com um circuito de soma de corrente (o circuito que subtrai a corrente de referência), um circuito de By-Pass (o circuito que destina a corrente positiva para um caminho e a negativa para outro) e um circuito de espelhos de corrente (o circuito que copia as correntes em um mesmo sentido e multiplica as correntes por 2).

Agora precisamos reproduzir a equação 4.3-4, que é uma comparação, essa equação foi reproduzida pela comparação dos níveis de tensão do VDS dos dispositivos nos dois caminhos destinados para reproduzir a cópia final para o módulo, sendo simplesmente um comparador duplo diferencial.

Terminamos com quatro sub-blocos para obter um bloco básico para o bloco CMD. Esses circuitos são: um SOMADOR; um By-Pass; um CIRCUITO DE ESPELHOS; e um CIRCUITO COMPARADOR. Veja a estrutura interna do BLOCO CMD na imagem 4.6.1-1.

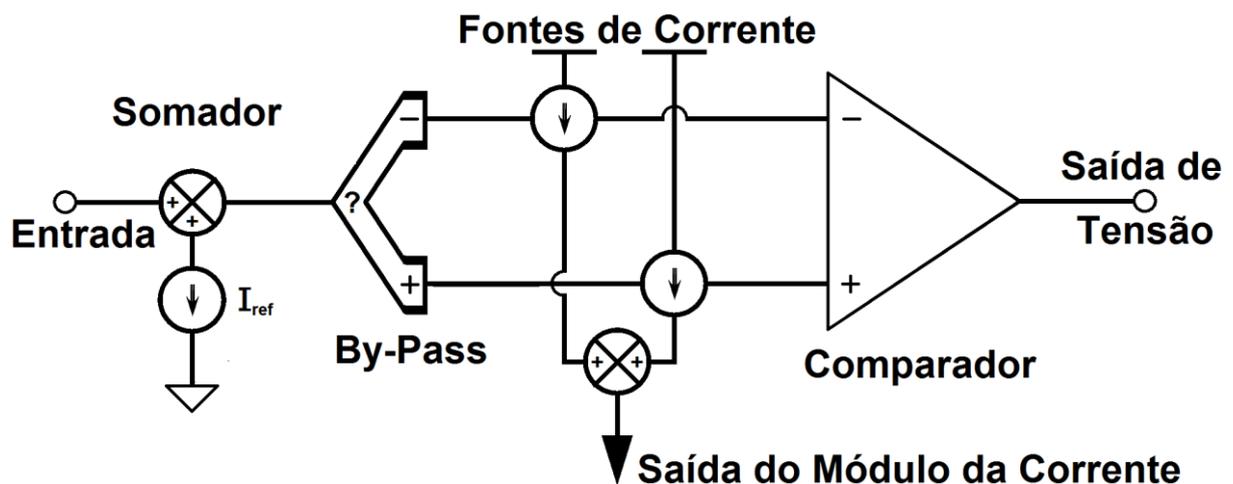


Imagem 4.6.1-1: Detalhamento do Bloco Básico do CMD

Dessa forma, a ideia central do Bloco Básico do CMD é preservar o comportamento equacional determinando o módulo da corrente de entrada relativo a uma referência fixa (do bloco Somador). Os sinais devem interferir de forma desprezível o suficiente para garantir o ENOB desejado no projeto final.

Cada um desses blocos básicos do CMD determina um bit, pois a ideia é que os bits são obtidos de forma paralela para obter um processo assíncrono e rápido.

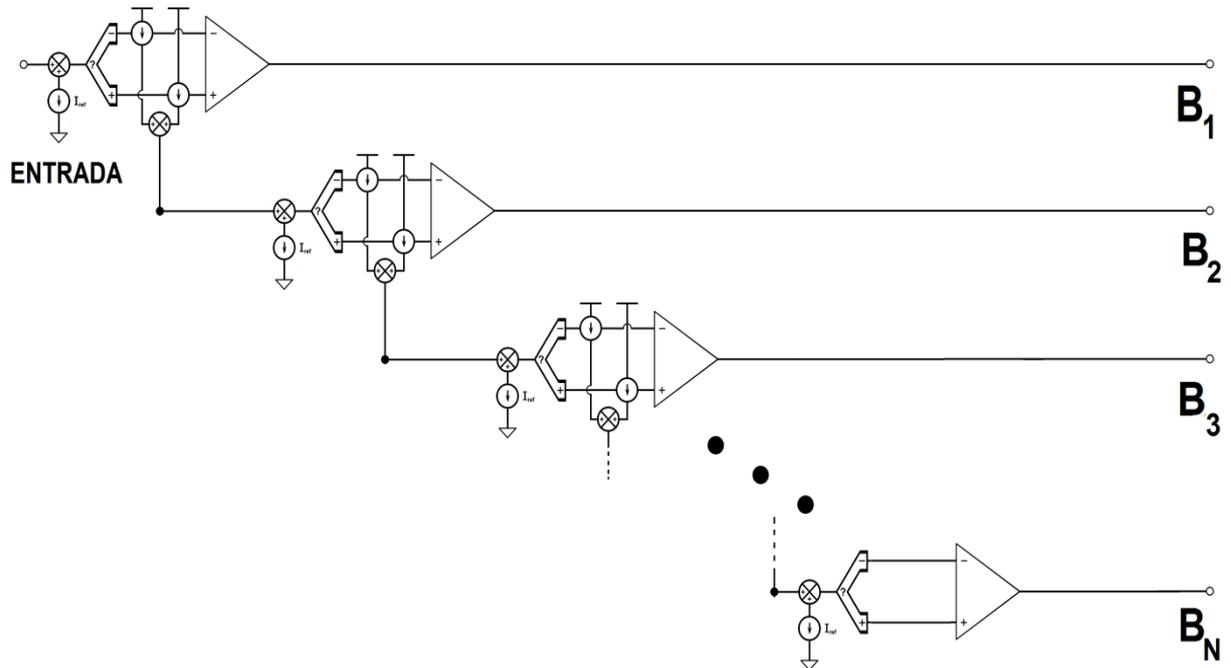


Imagem 4.6.1-2: Arranjo de blocos básicos do CMD

Na imagem 4.6.1-2, mostramos que o último bloco não precisa fornecer a corrente para o bloco seguinte. Os circuitos são controlados apenas pelos valores de tensão, gerados pelo By-pass, e, esses sinais, são amostrados pelos GATES dos outros blocos. Sendo assim, a retirada dos espelhos de corrente, do último bloco básico, não interfere nos demais blocos. Então, apenas o comparador realiza a leitura dos valores operacionais.

#### 4.6.1.1. BY-PASS E SOMADOR

O By-Pass tem grande importância na resolução final do conversor, pois as perdas de corrente do bloco implicam em perdas de bits e o Bloco deve entrar em corte quando a corrente de entrada for nula. A corrente residual, que venha a surgir quando a entrada de corrente no bloco for nula, será transmitida de bloco a bloco, podendo atingir significância suficiente para alterar os valores dos bits menos significativos e, portanto, reduzindo de forma irreversível a resolução final do CMD.

Conforme a imagem “4.2.1.1-1” os caminhos do circuito do By-Pass são em “P-MOS” e em “N-MOS”. Para determinar o sentido da corrente, no caso positivo e no caso negativo, há a fonte de corrente, conforme a imagem “4.2.1-2”, que puxa / retira corrente de um ponto comum, que é o circuito de soma de correntes, conforme a Primeira Lei de Kirchhoff, somatório das correntes em um nó. O circuito “Somador”, portanto, é apenas uma fonte de corrente retirando corrente do nó, que recebe a corrente de entrada e conecta-se na entrada do By-Pass imagem “4.2.1.1-2”, sendo uma das vantagens

de se trabalhar em modo corrente, pois no caso de uma soma em tensão, seria necessário copiar a tensão em um dispositivo ou um circuito, normalmente é utilizado um amplificador operacional para a realização desse processo em circuitos trabalhando com valores em tensão elétrica no lugar de corrente elétrica. No caso do Bloco Somador a sua simplicidade permite que seja explicado junto ao Bloco By-Pass.

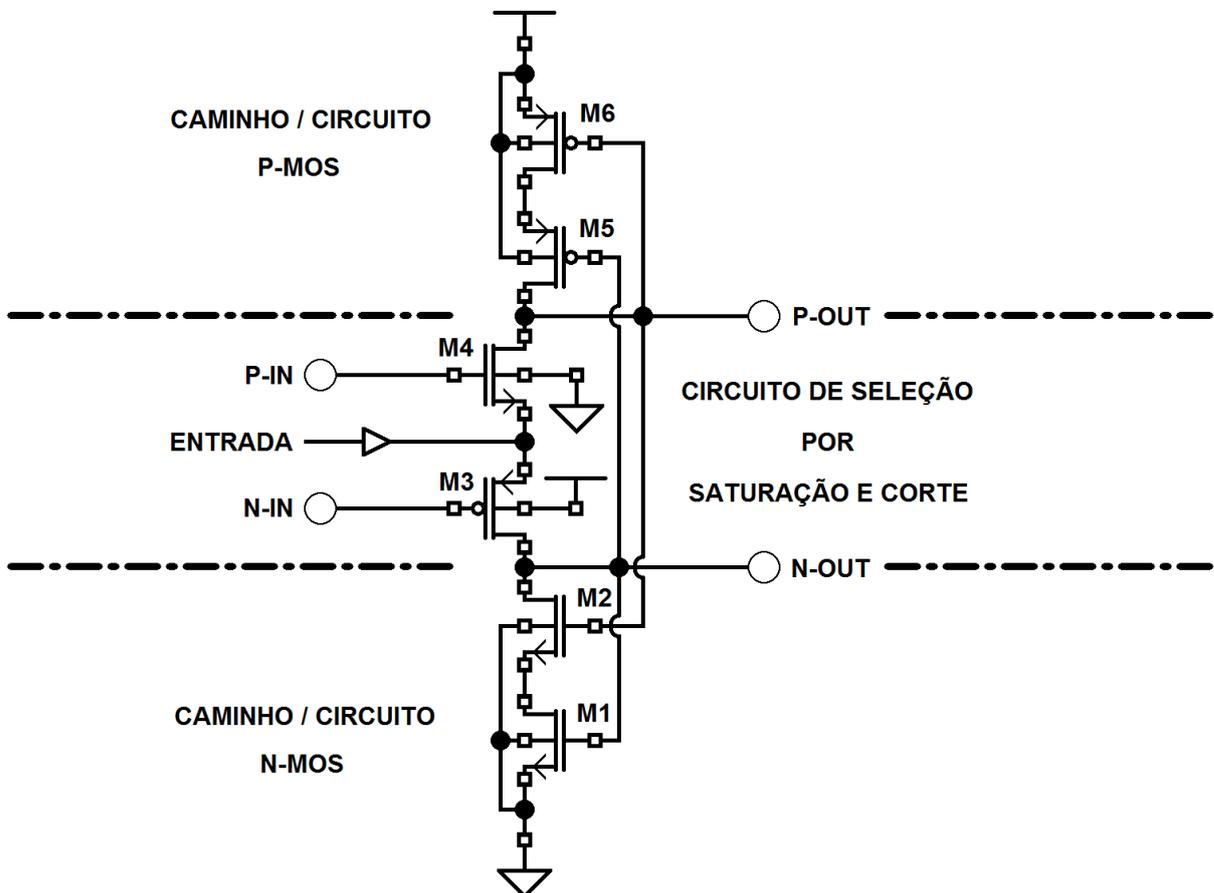


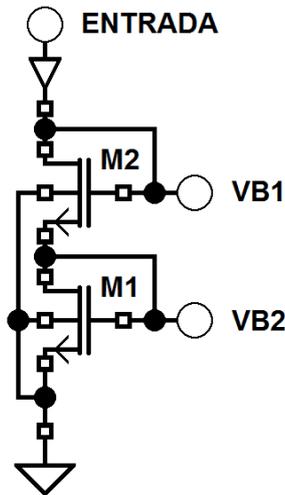
Imagem 4.6.1.1-1: Esquemático do circuito do By-Pass

O uso do circuito CASCODE é necessário, pois a precisão de todo o conversor dependerá da fidelidade dessa cópia de corrente, pois essa cópia é repetida para todos os blocos e é o circuito crítico para o casamento no Layout do circuito final, pois os circuitos dos comparadores e o circuito By-Pass são isolados, portanto, eles não são comprometidos pelos outros circuitos comparadores e / ou By-Pass que venha a ser somado ou subtraído do circuito. Dessa forma, o erro de uma fonte de corrente deve ser propagado para as outras fontes de corrente, pois é a forma de garantir um bom INL, prejudicando apenas o OFFSET, que é um erro mais simples a ser corrigido, podendo mesmo ser corrigido com a calibração da referência utilizada, ou seja, pode ser corrigido por circuito externo. Os erros dos conversores são tratados ao final desse capítulo.

Dessa forma, pode-se apresentar o comportamento da corrente de saída pela a corrente de entrada. A curva da imagem "4.6.1.1-3" está normalizada, portanto, "1" representa "100%". A referência deve ser 50%, pois é condição para o módulo dividir a escala em duas faixas iguais. A simplicidade do

bloco SOMADOR é importante para garantir um fácil casamento do Layout final, pois os espelhos de corrente sofrem grandes distorções com a diferença de dopagem ao longo do silício, efeitos térmicos inerentes aos circuitos vizinhos e interferências eletromagnéticas.

### REFERÊNCIA DE CORRENTE



### SOMADOR

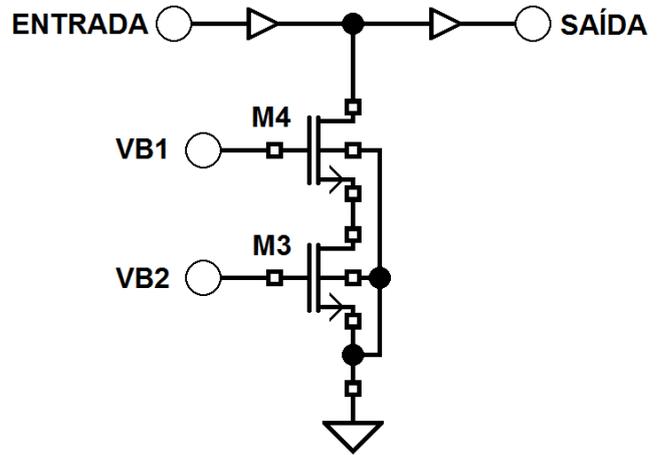


Imagem 4.6.1.1-2: Circuito Somador e Referência de Corrente

$$S(I_{in}) = I_{in} - I_{ref} \quad (4.2.1.1-2)$$

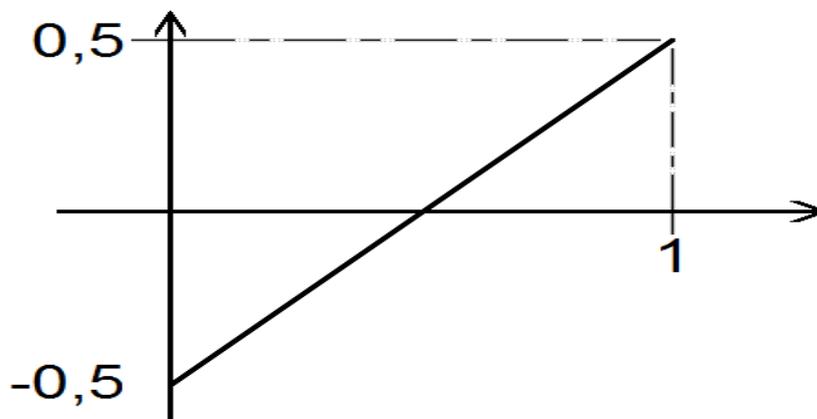


Imagem 4.6.1.1-3: Curva de transferência do Somador

Observe que haverá efeito de corpo no conjunto, por isso, devemos calcular a tensão necessária para a operação dos dispositivos em saturação com a corrente máxima à qual o sistema será submetido. Esse cálculo só é possível considerando a modulação de canal, e o efeito de corpo, pois os dispositivos do ByPass estarão com efeito de corpo significativo e prejudicarão a condução, caso o  $V_{gs}$  não satisfaça a condição de  $V_{gs} > V_{th}$ . Conforme as equações.

$$P_{IN} - V_{THM4} - \Delta(V_{THM4}) = V_{DD} - V_{DSM6} - V_{DSM5} - V_{DSM4} \quad (4.6.1.1-1)$$

$$N_{IN} + V_{THM3} + \Delta(V_{THM3}) = V_{DSM3} + V_{DSM2} + V_{DSM1} \quad (4.6.1.1-2)$$

Quando a entrada igual a referência, a corrente de entrada é totalmente desviada pela fonte de corrente do circuito somador e o valor do comparador somente será definido pela histerese do estado anterior, imagem “4.6.1.1-2”. Essa incerteza ocorre em diversos conversores, mas no caso do CMD, como será mostrado na seção “4.4”, é garantido que apenas um dos blocos básicos estará indeterminado. Por se tratar da transição de um valor binário para outro e apenas um bit é alterado entre valores binários adjacentes, para a lógica / comportamento binário do CMD, conforme a imagem “4.4-3” da seção “4.4”. Dessa forma, mesmo em um valor totalmente indefinido e sem um circuito de Sample and Hold, o valor binário definido pelo CMD não sofre variação maior que a de um LSB.

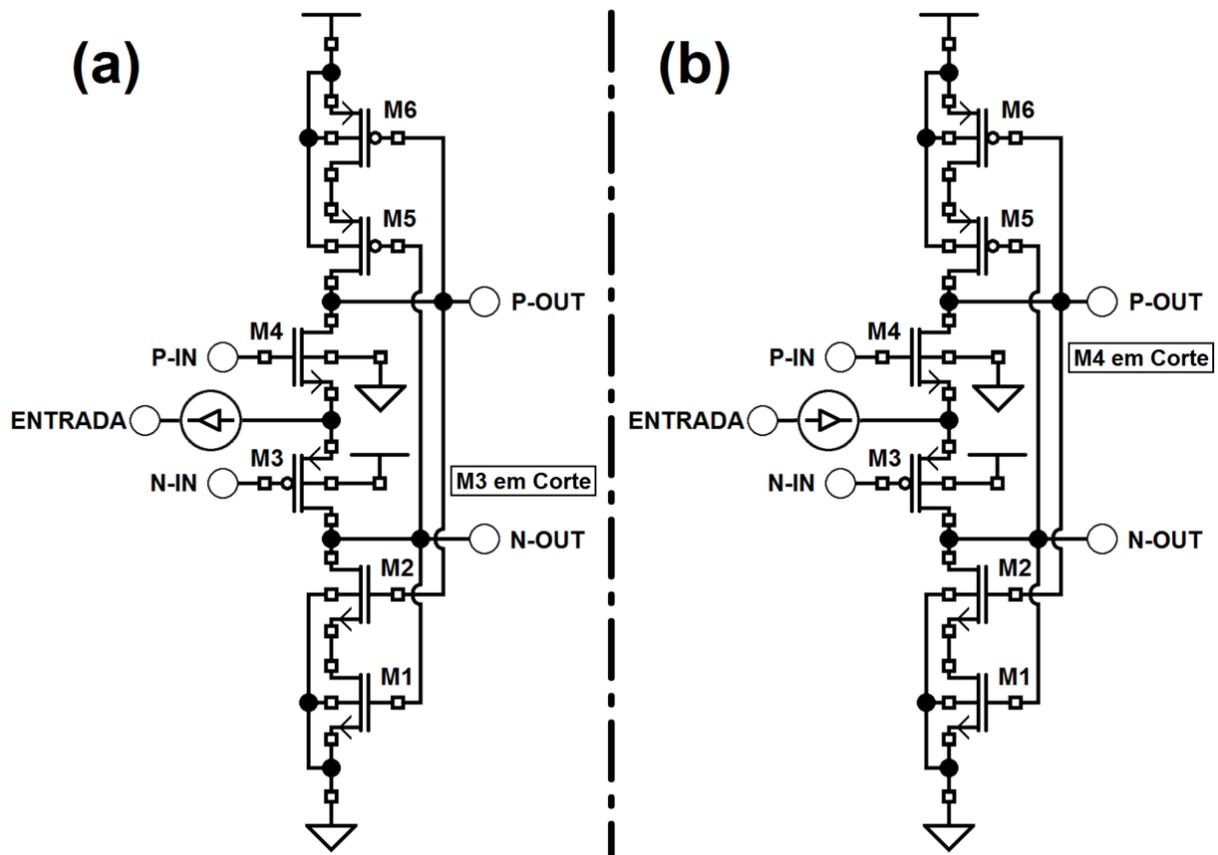


Imagem 4.6.1.1-4: Seleção automática do By-Pass

No momento em que a corrente de entrada se torna superior ao valor da corrente da fonte do bloco somador, parte da corrente é injetada no bloco do By-Pass, que passa a conduzir pelo caminho definido pelos dispositivos “N-MOS”, sendo os dispositivos “M1 e M2” da imagem “4.6.1.1-4 b”. E, nesse caso, “M3” entra em condução, provoca o aumento de tensão do nó de entrada para valores de “N-IN” mais a tensão de threshold e corta o dispositivo “M4”.

Quando a corrente é retirada do By-Pass, passa pelo caminho “P-MOS”, cortando o caminho “N-MOS”, portanto, o efeito de canal, nos dispositivos “M1 e M3”, com o aumento da tensão de Vds,

conforme ocorra aumento do módulo da diferença da corrente, acarretará valores baixos o suficiente, para desprezar a corrente que circula pelo caminho “N-MOS”. Contida no objetivo necessário para atingir a relação sinal ruído suficiente ao ENOB desejado. Os dois caminhos alternativos são destacados na imagem 4.6.1.1-5, na qual “a” representa o caminho de retirada de corrente pelos componentes “P-MOS” e “b” representa o caminho de injeção de corrente pelos componentes “N-MOS”.

Dessa forma, a ideia central dessa configuração é assegurar que quando um dos caminhos de condução estiver ativo, o outro estará em corte, garantindo uma maior resolução, uma vez que o dispositivo em corte tenderá a conduzir o mínimo de corrente e, o outro dispositivo a conduzir de forma natural, a que está dentro do sentido da corrente fluindo de VDD à GND. Assim, um dispositivo entra em saturação, alterando a tensão na entrada, e coloca o outro dispositivo em corte.

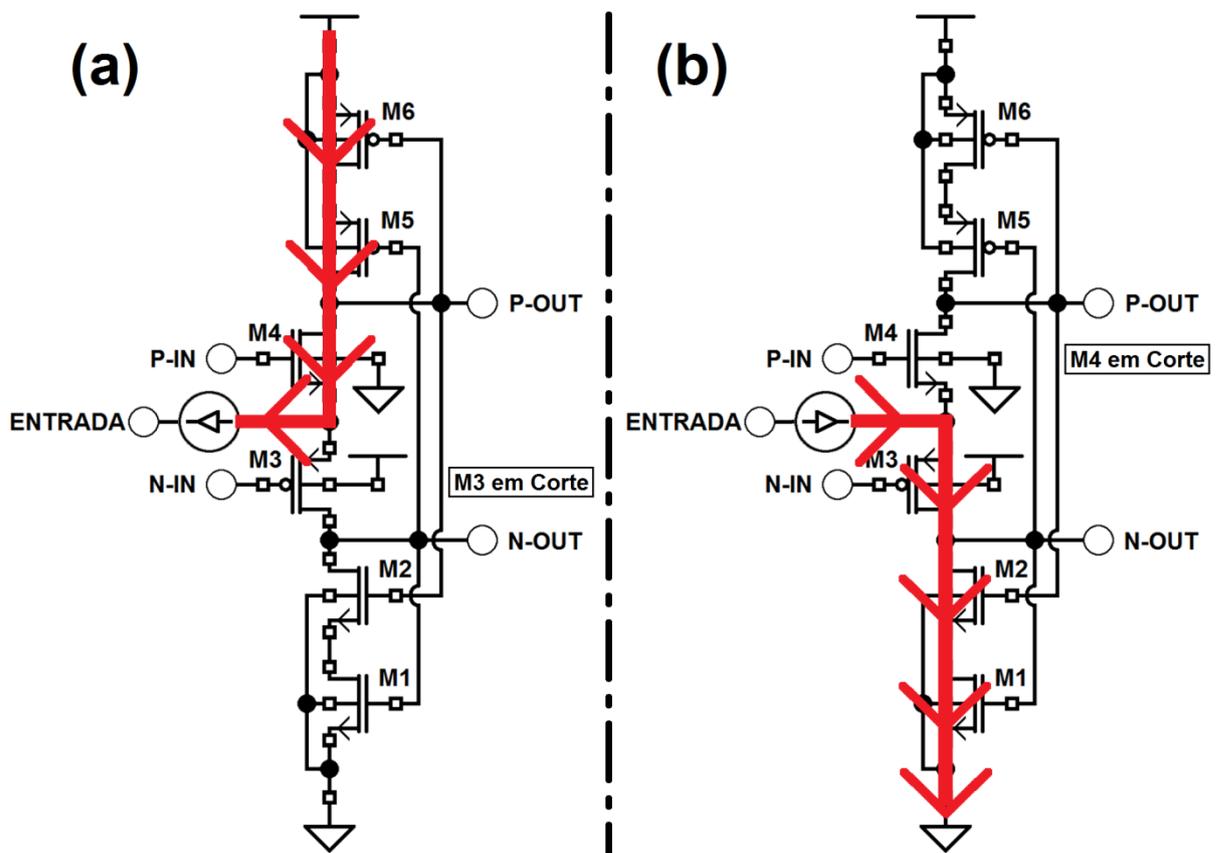


Imagem 4.6.1.1-5: Caminhos do By-Pass

No Apêndice I há maiores detalhes da matemática do processo.

#### 4.6.1.2. Espelhos

O SOMADOR define quando a corrente é “positiva” e quando ela é “negativa” enviando-a ao By-Pass que separa a corrente por dois caminhos distintos. Os espelhos, por sua vez, copiam as correntes e direcionando-as no mesmo sentido, independentemente de ser positiva ou negativa.

Entregando a corrente de saída com o mesmo valor da soma dos módulos dos caminhos do By-Pass para o próximo bloco, que recebe um sinal proporcional ao valor do módulo da diferença do sinal de entrada do bloco anterior. Esse sinal passará pelo somador do próximo bloco e dará continuidade ao processo, herdando os erros do bloco anterior e seus atrasos. Dessa forma, não há a interferência e/ou a necessidade de qualquer sinal digital convertido para tomada de decisão. Sendo, a propriedade de tomada de decisão, o que evitamos do SAR e do PIPELINE, uma vez que não desejamos a estabilizar um sinal digital para, só então, darmos continuidade à conversão.

No caso do CMD, a cada espelho ocorre a ampliação do sinal, uma vez que o valor do “W” (largura do MOSFET) dos dispositivos, de M1 a M9 do circuito da imagem 4.6.1.2-1, são duas vezes os valores de “W” para os componentes do By-Pass, portanto, há um ganho de 2 vezes no par de espelhos. Assim, o resíduo que é ampliado no PIPELINE, é ampliado a cada bloco do CMD, mas sem precisar de valores binários pré-definidos para realizar o processo.

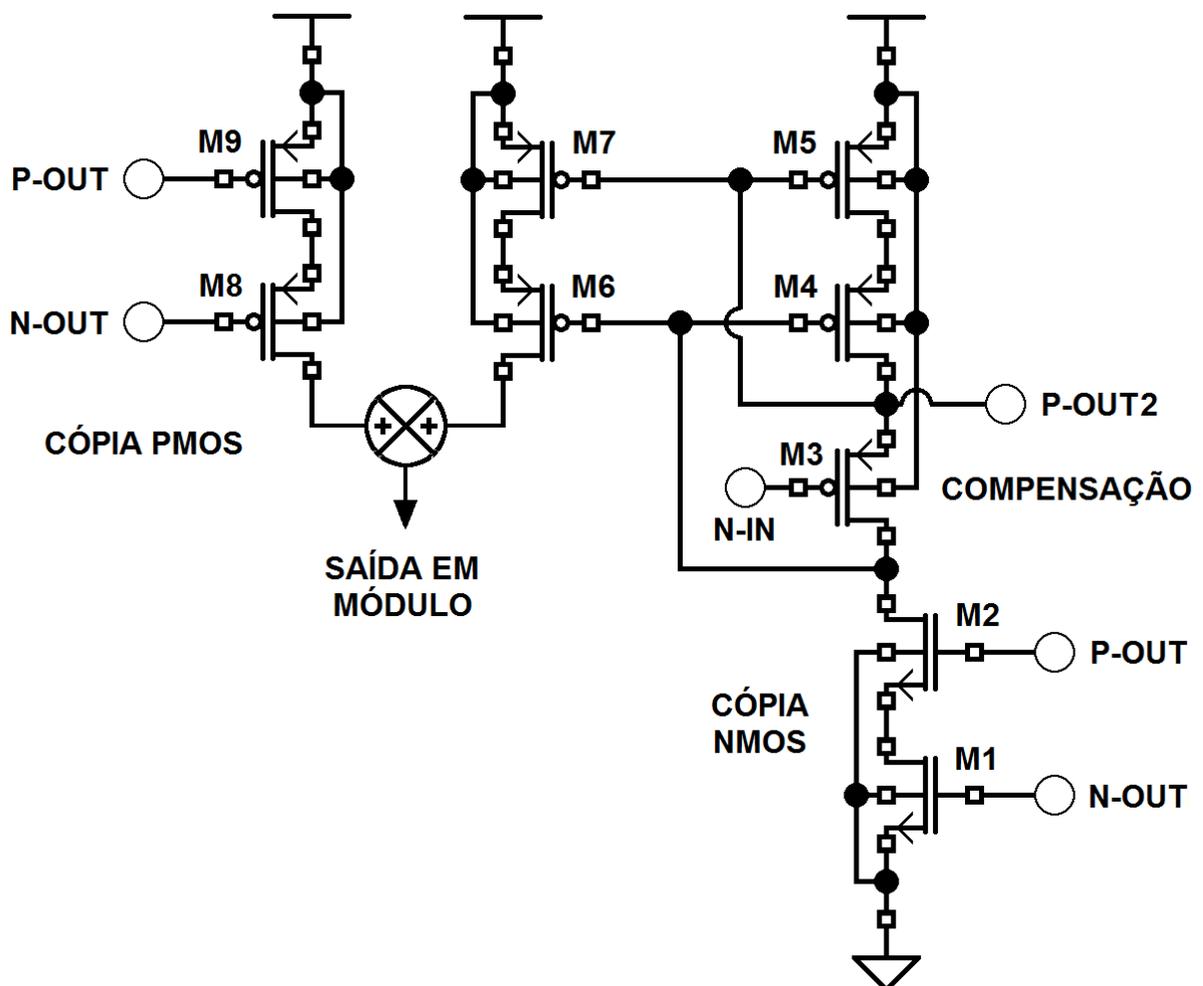


Imagem 4.6.1.2-1: Esquemático do circuito dos espelhos

Os espelhos geram erros relacionados ao efeito de modulação de canal, que obriga aos dispositivos copiados e aos dispositivos dos espelhos possuir valores de “L” (comprimento de canal do dispositivo) extenso o suficiente para garantir a fidelidade da cópia. Além do tamanho para “L” foi

adotada a topologia CASCODE, pois reduz o efeito de canal pelo fato de os dispositivos “M2”, “M4”, “M6” e “M8” sofrerem variações para os valores de  $V_{ds}$  no lugar dos dispositivos “M1”, “M5”, “M7” e “M9” garantindo maior fidelidade ao processo de cópia. O dispositivo “M3” está servindo de compensação à queda de tensão acarretada pelo dispositivo em COMUM GATE do By-Pass, pois essa queda de tensão é suficiente para prejudicar a proteção gerada pelos dispositivos “M2” e “M4”, que se destinam a mais uma cópia de sinal. Os espelhos também geram o sinal P-OUT2, que servirá para a comparação realizada pelo circuito comparador, portanto, há uma simetria gerada nesse processo que criará o parâmetro final necessário à comparação.

#### 4.6.1.3. Comparador

Trata-se de um simples par diferencial que possui seu funcionamento ampliado/melhorado pela grande variação do estado de operação de suas entradas, sendo os valores de P-OUT e P-OUT2 gerados, respectivamente, pelo By-Pass e pelo circuito dos Espelhos de Corrente. Os sinais variam de forma significativa devido ao estado de equilíbrios dos componentes do By-Pass, que produz grande variação de tensão nesses pontos por garantir que apenas um dos componentes em COMMUM GATE estará em saturação por vez, através dos valores de polarização. Assim, a fonte de corrente obtida com o disposto M1 da imagem 4.6.1.3-1 não precisa de características precisas e pode fornecer pouca corrente, já que há grande variação entre o  $V_{gs}$  de M2 e o  $V_{gs}$  de M5. O inversor serve para garantir um sinal limpo e também reduzir a necessidade de fornecer corrente através dos dispositivos “M4” e “M5”.

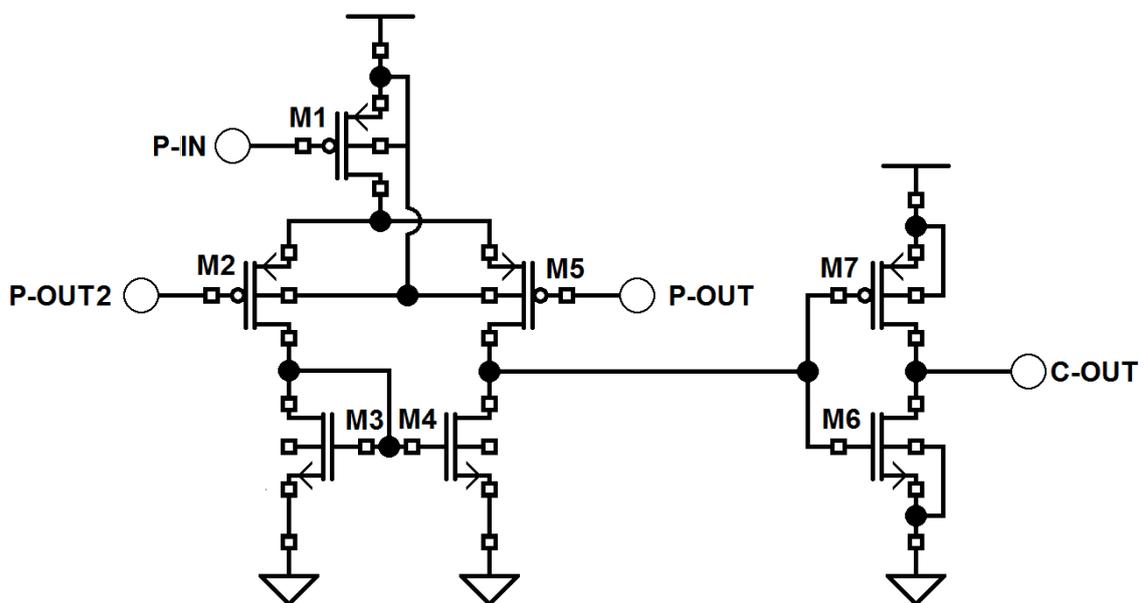


Imagem 4.6.1.3-1: Esquemático do Circuito do Comparador

Dessa forma, esse circuito foi projetado para consumir pouco e seu sinal é melhorado por um inversor. Como veremos adiante, os inversores atuaram em momentos distintos ao longo do processamento do sinal analógico na seção 4.4.

#### 4.6.1.4. FLUXO DA CORRENTE

O fluxo da corrente é basicamente a geração de um novo módulo da corrente de entrada a cada bloco, sendo que a entrada do bloco seguinte será a saída do bloco atual. Para vislumbramos o comportamento foi adicionada essa seção, que servirá de auxílio para a compreensão da seção 4.4, destinada à explicação do comportamento do geral do CMD.

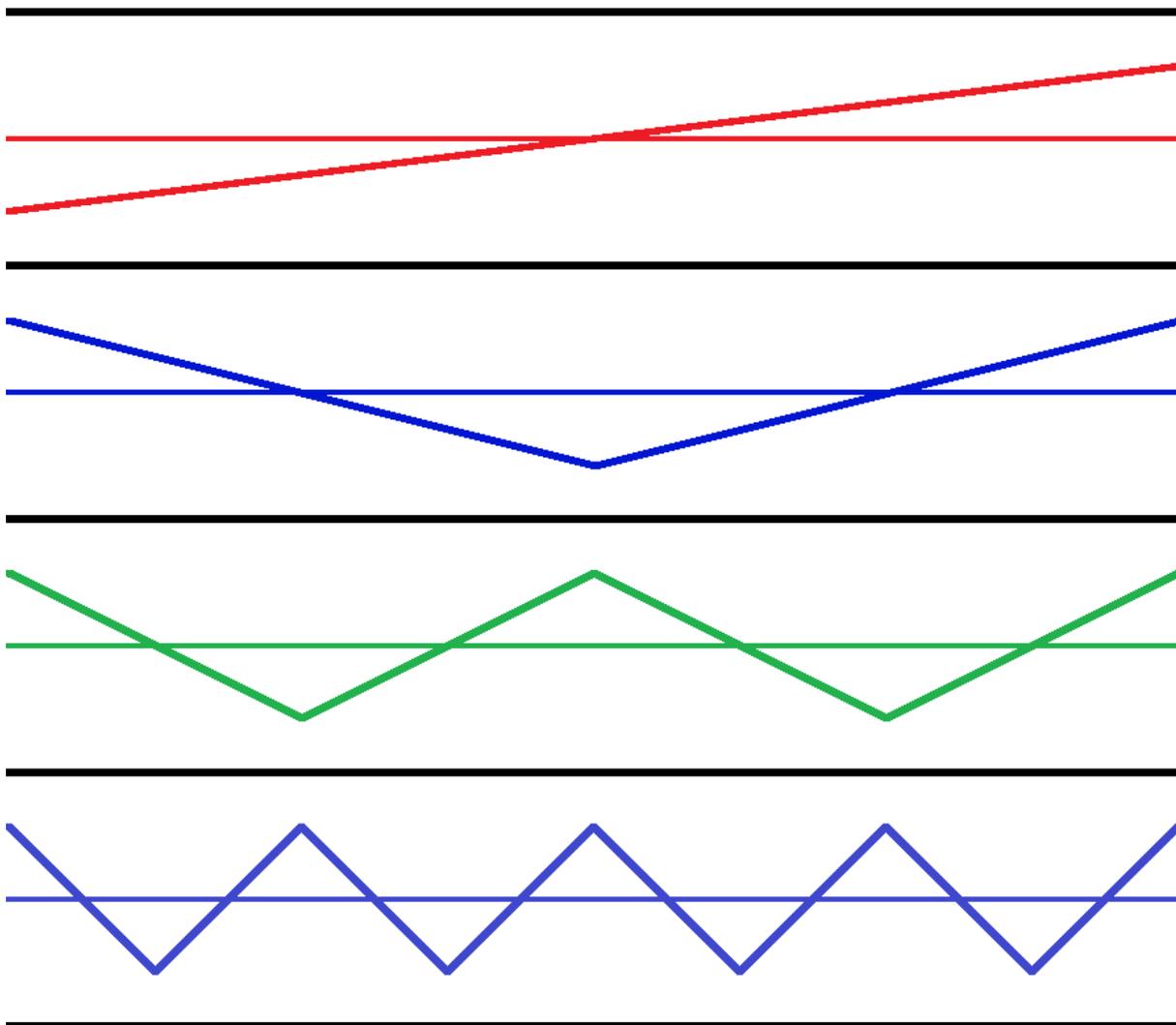


Imagem 4.6.1.4-1: Corrente de Entrada do 1º ao 4º Bloco

Na imagem 4.6.1.4-1 temos a representação do comportamento da corrente, de forma ideal, fluindo através dos blocos básicos do CMD. Em vermelho temos a corrente que está no primeiro bloco, pode-se verificar que a mesma varia de forma linear e crescente, sendo que a reta horizontal, também vermelha, representa a referência, que determinará os valores negativos e os valores positivos, sendo essa faixa definida pela corrente de referência inserida pelo bloco somador. A segunda curva é a saída do primeiro bloco e a entrada do segundo bloco, representada pela cor azul, da mesma forma que a curva anterior, a reta horizontal representa o valor da referência, mas por ser a saída do primeiro bloco já tem o comportamento do módulo do anterior e corta a reta horizontal em dois pontos. E assim, podemos observar o mesmo padrão na terceira e na quarta curvas. Os pontos aos quais as entradas

coincidem com os valores das referências serão pontos máximos das curvas filhas, ou seja, que são geradas pelo bloco correspondente.

Outra observação importante é que os pontos de interseção das curvas de entrada com as respectivas referências definem transições dos bits desses blocos, conforme a imagem verificamos que esses pontos não se coincidem entre cada bloco e, como será mostrado adiante, esse processo está relacionado com a simetria dos bits gerados pelo bloco CMD, além de estar relacionado com o fato de um bit nunca alterar o seu valor de forma simultânea com outro bit, sendo a observação que foi feita em relação ao inversor na saída do par diferencial, ou seja, não haverá momento em que dois inversores estejam submetidos à variação máxima de sua corrente, pois é coincidente com o ponto de inversão de um bit, mas os bits variam separadamente e sem coincidência.

#### **4.6.1.4.1. COMPORTAMENTO ESPERADO**

Retomando a Imagem 4.6.1-2, Arranjo de blocos básicos do CMD, e considerando o comportamento do circuito dentro da faixa definida pelo valor 1010 da conversão obtida pelo código do CMD, conforme a seção 4.1.1, GERANDO OS BITS. Veremos com a corrente fluir pela estrutura da imagem 4.6.1-2 na imagem 4.6.1.4.1-1, na qual cada processo descrito nas equações 4.4-1, 4.4-2, 4.4-3 e 4.4-4 é, respectivamente, representado pelos caminhos nas cores vermelho, azul, verde e roxo, ou seja, cada processo é representado por uma cor na imagem 4.4-2.

Note que, a cada processo, determinado por um bloco básico, o range de variação da corrente dobra, pois o valor da saída é multiplicado por dois. Assim, o valor de entrada do primeiro bloco varia de 9 a 10 LSBs, o do segundo bloco varia de 2 a 4 LSBs, o do terceiro bloco varia de 8 a 12 LSBs e o do quarto varia de 0 a 8 LSBs. Dessa forma, a amplitude do sinal de entrada para cada bloco, varia, respectivamente:  $10-9= 1$  LSB,  $4-2= 2$  LSB,  $12-8= 4$  LSB e  $8-0= 8$  LSB. Ainda, consideramos o valor de entrada, que caracterizamos por um byte de 4 bits, tomando como referência o valor binário padrão, que representa exatamente o valor decimal correspondente, assim, a faixa analógica em questão, iniciando a contagem de baixo para cima e tendo como valor correspondente a primeira faixa o zero, o valor decimal que corresponde ao da faixa selecionada é o  $9_{10} = 1001_2$ . Entretanto, a sequência de bits, obtida foi 1010, que corresponderia ao valor 10 em decimal, ou seja,  $10_{10} = 1010_2$ .

Verifique na imagem 4.4-2, que a saída do último bloco varia de 0 a 16 LSBs, isso ocorre para qualquer faixa definida por um valor binário, ou seja, essa corrente não trará caracterização alguma do valor analógico, por isso, conforme a imagem 4.2.1-2, apresentada no início do capítulo, não há a necessidade do espelho de corrente no último bloco básico. Entretanto, caso a corrente inicial variasse de 15,5 a 16 LSBs, o valor da saída do quarto bloco básico, ainda variaria de 8 a 16 LSB, portanto, seria adicionável outro bloco, que, conforme observado anteriormente, seria dobrada a faixa novamente e o quinto bloco básico, também passaria ter sua saída correspondendo a toda a faixa analógica em sua saída. Essa condição é, portanto, apenas um reflexo da resolução desejada, que implica na adição

de novos blocos, caso queria aumentar a resolução, pois estamos variando o valor da entrada do primeiro bloco para caracterizar toda a faixa definida pelo valor binário em questão.

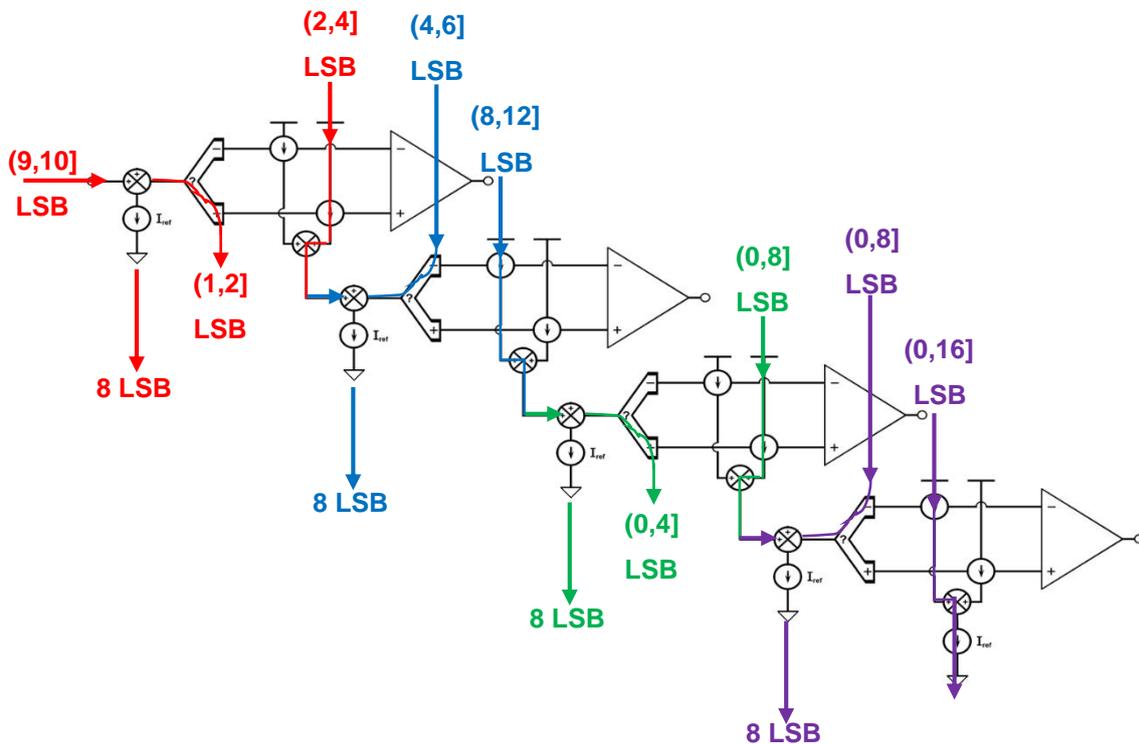


Imagem 4.6.1.4-1: Fluxo de corrente entre os blocos básicos do bloco CMD

## 4.6.2. Bloco PGA

Como o conversor foi desenvolvido totalmente em modo corrente e, geralmente, são colocados blocos de ganho programável na entrada dos conversores para que haja uma instrumentação com tratamento prévio do sinal entrada em níveis ótimos de operação para o conversor em questão. Assim, caso a corrente esteja com valores muito elevados - para o correto funcionamento do CMD - o PGA poderá ser ajustado e, da mesma forma, caso os valores de corrente estejam muito baixos, o PGA poderá ampliar esses valores a níveis adequados ao conversor.

Não há de interesse aprofundar no bloco do PGA, é apenas um circuito acessório para o conversor, portanto, serão mostrados os seus resultados sem um detalhamento prévio, como está sendo feito com o restante do conversor.

### 4.6.2.1. Resumo PGA

O PGA foi realizado ainda na DfChip, empresa vinculada à UnB, o bloco básico recebe um sinal de corrente e pode multiplicar por fatores de 1, 2, 3, 4, 5, 6, 7 e 8. O ganho é determinado por um sinal digital de três bits. E possui um circuito realimentado em corrente e tensão em todas as transferências internas de sinal. Frequência de 1MHz.

A ideia de um PGA com apenas 3 bits se deu pelo aumento exponencial do tamanho do bloco de acordo com o número de bits. Assim, o de 4 bits dobraria o tamanho do PGA, entretanto, colocando dois PGA de 3 bits em série, formaríamos um PGA de 6 bits do tamanho de um único PGA de 4 bits. O tamanho do circuito aumenta, pois o ganho do sistema é dado pela adição de espelhos de corrente, conforme o diagrama de blocos a seguir na imagem 4.6.2.2-1.

#### 4.6.2.2. Descrição dos Blocos Básicos do PGA

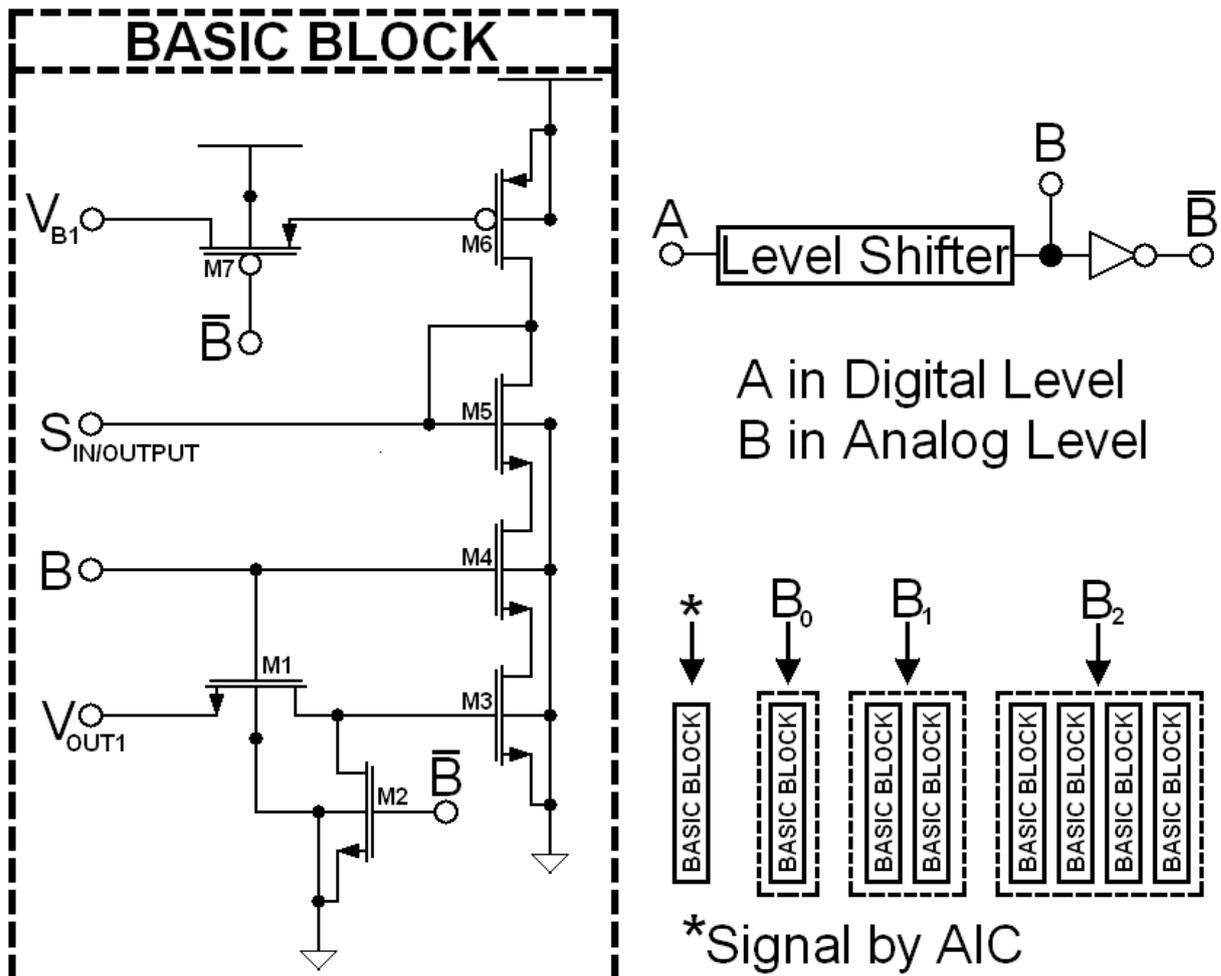


Imagem 4.6.2.2-1: Diagrama dos espelhos de Corrente do PGA – SPLITTER

Os dispositivos M1, M2, M4 e M7 são destinados apenas ao acionamento dos blocos básicos e também são utilizados nos blocos de entrada de sinal, independente da necessidade de comandar esses blocos, apenas para garantir a resolução e precisão para o processo final. Vemos, portanto, que para colocar um 4º Bit, seria necessário adicionar mais 8 "BASIC BLOCK" do PGA, portanto, é melhor colocar dois circuitos PGAs em série, caso seja necessário ampliar o sinal mais de 8 vezes. Esse bloco, conforme podemos observar, é apenas um repetidor, portanto, há, no PGA um bloco responsável pela recepção do sinal e estabilização dos níveis de tensão de entrada dessa corrente em um sinal fácil de copiar e, só então, inserir o sinal no conjunto de espelhos formados pelos "BASIC BLOCK". O bloco que recebe o sinal de corrente gera os sinais  $V_{OTU1}$ ;  $B$ ;  $S_{IN/OUTPUT}$  e  $V_{B1}$ , mostrados na

imagem 4.6.2.2-1, para garantir fidelidade no processo de cópia. A visualização dos blocos básicos que compõe o PGA está na Imagem 4.6.2.2-2.

O primeiro bloco, o AIC – Bloco de Aquisição, Instrumentação e Controle, recebe a corrente e é responsável por não fornecer impedância elevada à corrente de entrada para que o sinal de entrada não sofra distorções e também pela qualificação do sinal de entrada em um sinal de fácil cópia: imagem 4.6.2.2-3.

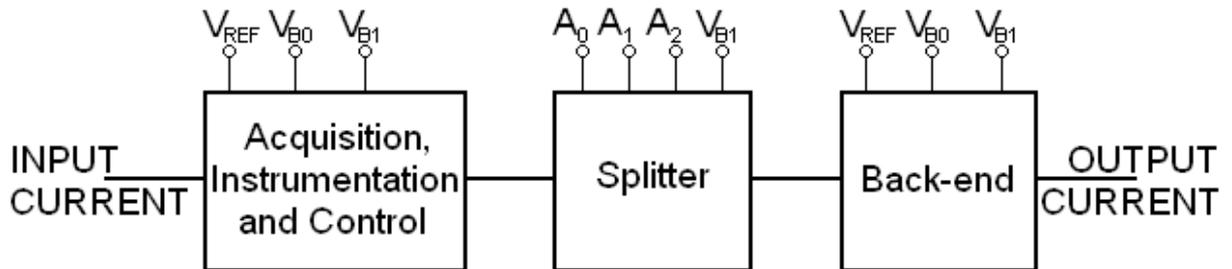


Imagem 4.6.2.2-2: Blocos Básicos do PGA

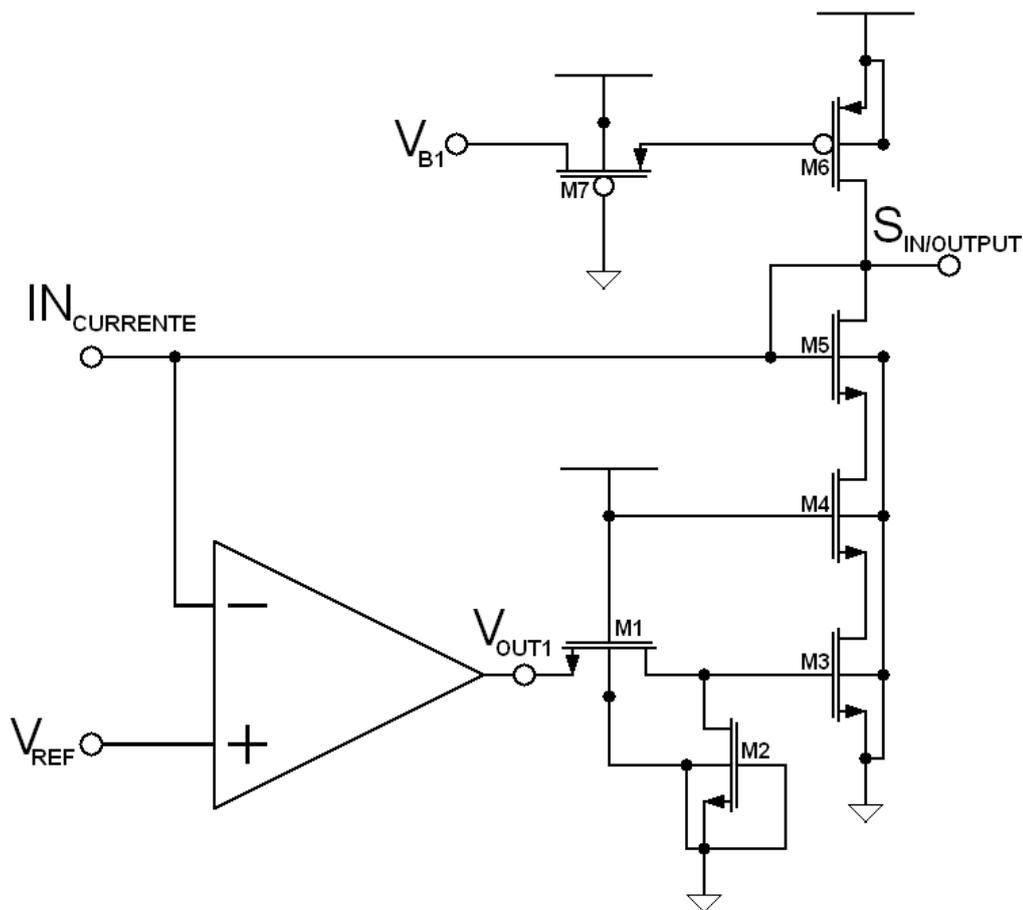


Imagem 4.6.2.2-3: AIC do PGA

Essa facilidade de cópia se deve à estabilização dos níveis de tensão da entrada, pois o amplificador possui um de seus terminais de entrada destinado à realimentação negativa do conjunto



signal de tensão da entrega do valor final da corrente tratada no PGA. Na imagem 4.6.2.2-5 temos a topologia do circuito de subtração do sinal DC resultante.

Assim, o circuito M1 é controlado para puxar a corrente de Modo DC com M4 operando com a tensão de  $V_{REF}$  em seu DRENO, tornando os dispositivos M3 e M2 condicionados aos níveis desejados de operação. O ideal será a entrada do circuito seguinte ter a tendência de equilibrar em níveis próximos de  $V_{REF}$ , que é o nosso caso, pois o bloco do By-Pass tende a operar próximo do valor de  $V_{REF}$  em questão, que é o valor de  $V_{dda}/2$ .

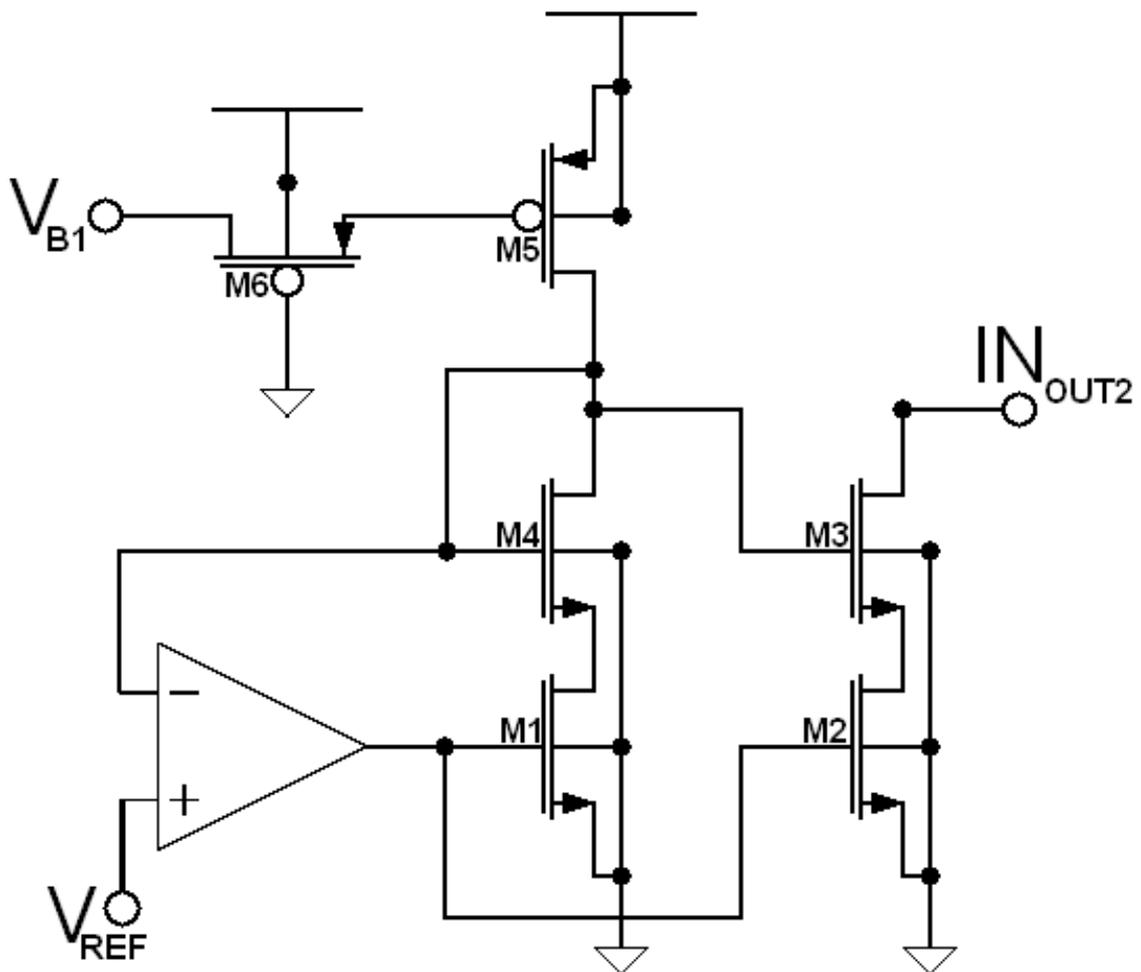


Imagem 4.6.2.2-5: Circuito de Eliminação do Modo DC

## 4.7. CIRCUITOS EM CADENCE

Serão apresentados conceitos, resultados e caracterização do conversor realizados imediatamente após o embasamento das características abordadas. Além de simulações com erros intencionais no circuito para apresentar os efeitos que surgiram durante o projeto e qual o motivo da distorção encontrada. A questão é demonstrar como os erros influenciam a resolução final do conversor e o impacto para reduzir o efeito em relação ao erro analisado.

A seção continua com o projeto de um circuito Ultra Low Power para explorar mais parâmetros da arquitetura. E apresentaremos os motivos que distinguem o CMD das outras arquiteturas. Demonstrando se tratar de um novo processo para determinação de valores digitais a partir de uma entrada analógica de interesse.

#### 4.7.1. BLOCOS BÁSICOS

Podemos observar que são os mesmos blocos básicos da imagem 4.2-1, que está na seção 4.2. Os circuitos apresentados nessa seção são os correspondentes aos anteriormente descritos, pois é o projeto que servirá, como base para explicarmos os efeitos inerentes aos principais erros de projetos ocasionados durante o desenvolvimento.

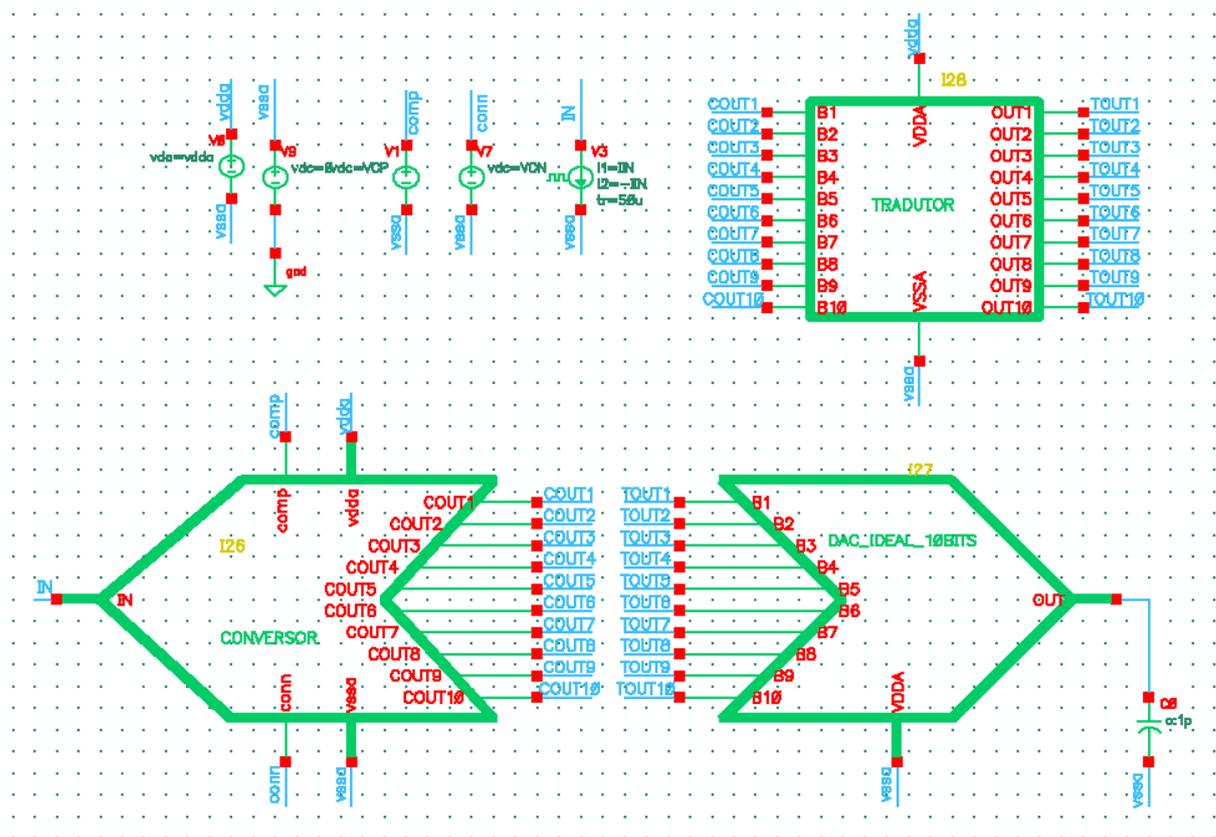


Imagem 4.7.1-1: Top Level do Projeto Completo dentro do Ambiente CADENCE

Foi utilizado um DAC ideal para caracterizar o projeto, no intento que o DAC promova um mínimo de desvios dentro do processo de caracterização do projeto como um todo. O circuito do Módulo Conversor está à esquerda na parte inferior da imagem 4.7.1-1; o conversor é composto por uma seqüência de blocos básicos, cujo interior é apresentado na seção seguinte.

#### 4.7.2. BLOCO BÁSICO DO CMD

Na imagem 4.7.2-1 observamos três divisões principais do bloco básico do CMD no By-Pass, nos Espelhos e no comparador.

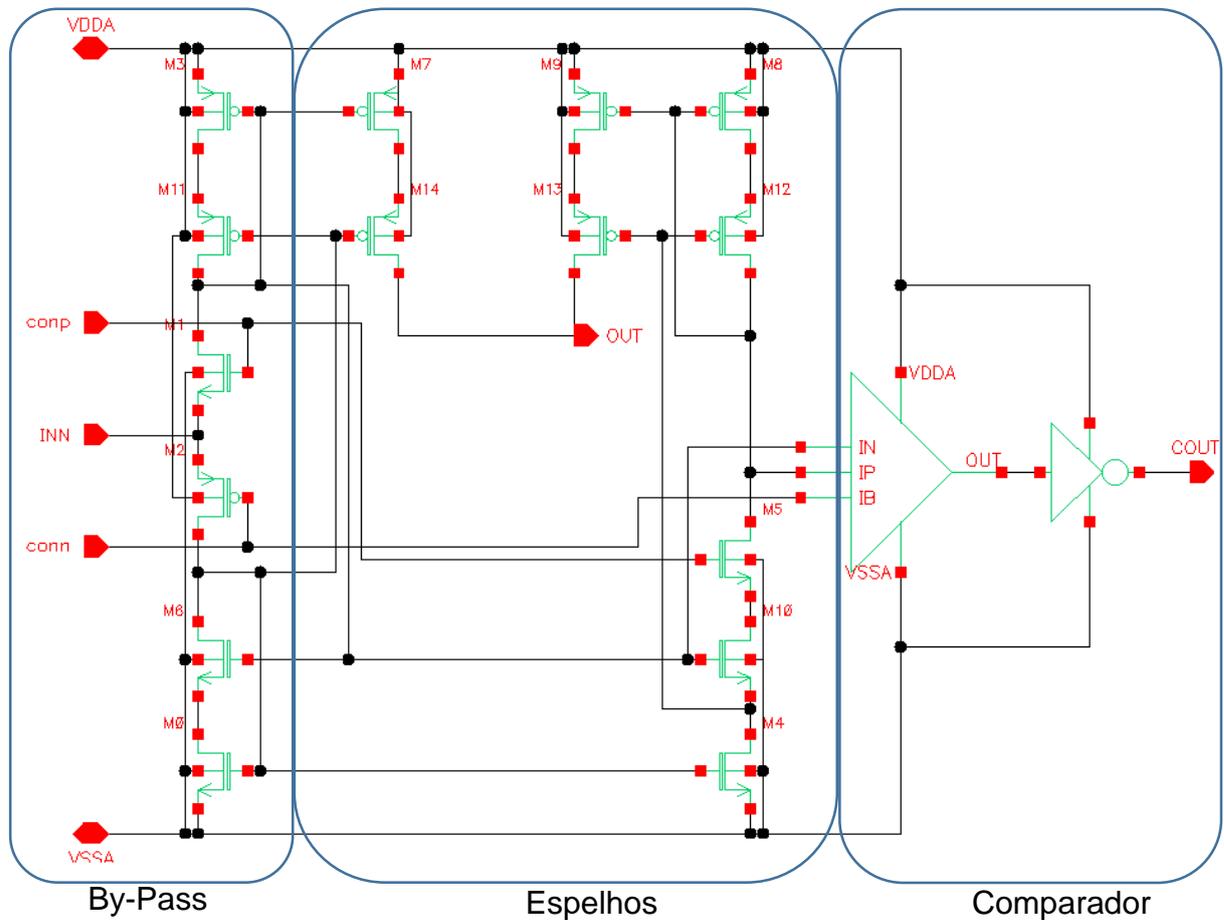


Imagem 4.7.2-1: Esquemático do Bloco Básico em Ambiente CADENCE de Projeto

A primeira divisão que compõe o circuito é o By-Pass que tratamos na seção 4.6.1, na qual tratamos do somador, que é apenas uma fonte de corrente conectada antes da entrada do By-Pass. A questão relacionada ao By-Pass de maior complexidade é o processo de saturação e corte entre os dispositivos.

Observando que o comparador, dentro do retângulo à direita, faz o uso de um inversor com a finalidade melhorar o sinal obtido pelo conversor. O inversor na saída de um comparador, constituído de um par diferencial foi utilizado para obter uma onda com um formato quadrado mais limpo, trazendo benefícios e exatidão ao processamento digital; o circuito interno do comparador encontrado na imagem 4.6-3.

O comparador é um circuito duplo diferencial simples e sua fonte de corrente opera com 500nA, pois o circuito recebe entradas que oscilam mais de 300mV, conforme as simulações, isso devido ao efeito causado pela saturação dos caminhos NMOS e PMOS do By-Pass e seus dispositivos conectados em COMUM GATE, conforme seção 4.6.1.1. Assim, o esquemático do comparador traz um

circuito de projeto fácil. É importante ressaltar que, caso o circuito fosse utilizado como um simples comparador e com referência fixa, sua especificação exigiria uma fonte de maior consumo e/ou de maior precisão, entretanto, tanto a entrada negativa do comparador, quanto a entrada positiva do comparador sofrem variações bruscas de tensão, sendo, portanto, escolhida a entrada positiva para sofrer menores variações e a negativa para sofrer maiores oscilações, uma vez que é a saída mais rápida.

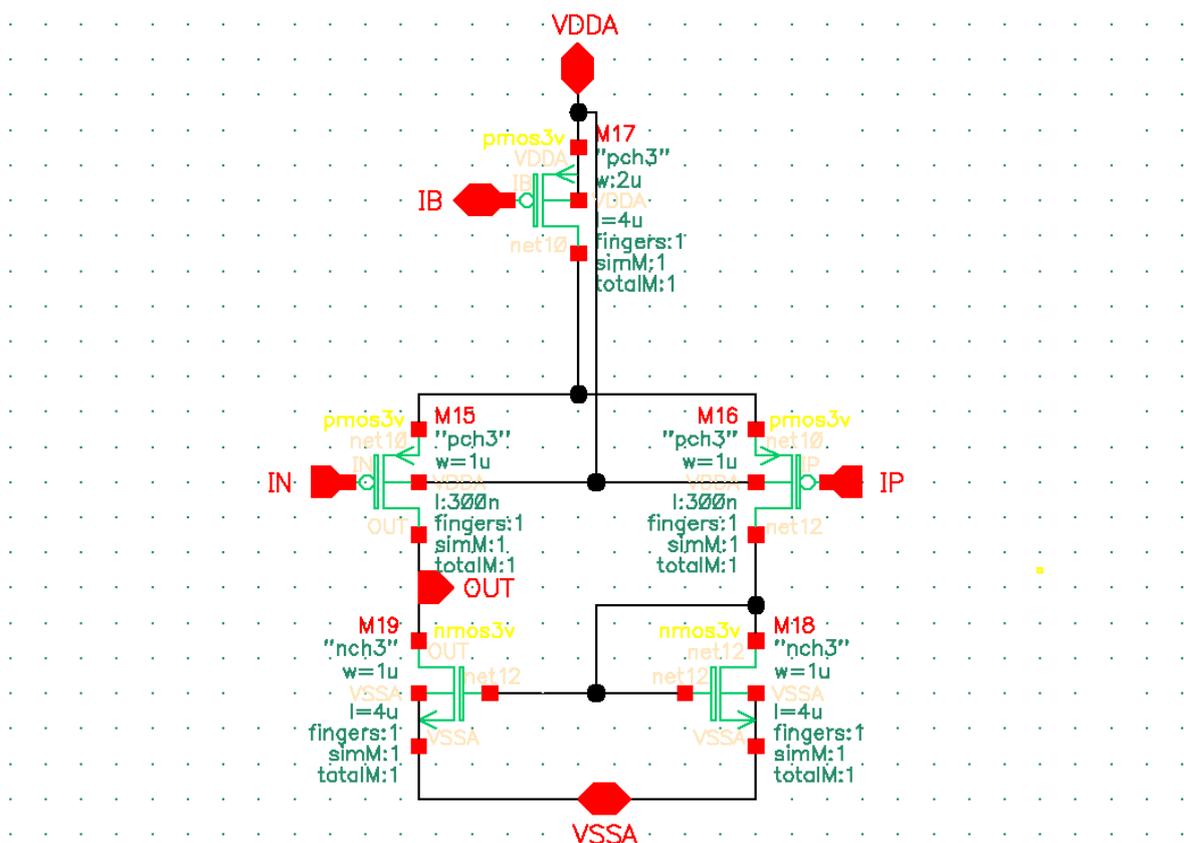


Imagem 4.7.2-2: Comparador do Bloco Básico do CMD

As grandes oscilações para os sinais de entrada do conversor, ocorrem graças ao conjunto de espelhos que compõem o módulo da corrente de entrada, gerada pelo bloco básico, pois os sinais de operação do comparador vêm do caminho PMOS do By-Pass e do caminho PMOS gerado pelo par de espelhos. Dessa forma, qualquer um desses sinais seria suficiente para inverter o comparador com o uso de uma referência fixa, mas o processo é intensificado com a variação das duas entradas, isso também economiza circuitos para interface e polarização de uma referência.

O sinal do Conversor inverte eficientemente, mas não consegue gerar uma onda adequada sem a necessidade de consumo significativo para o circuito, conforme a imagem 4.7.2-3; temos a curva do comparador durante o processo de caracterização individual do bloco básico do CMD, sendo: A curva verde claro (curva central) está consumindo 2µA por comparador. A curva em vermelho (curva mais à esquerda) passou pelo inversor. Dessa forma, a entrada da curva vermelha e a da curva verde

escuro (curva mais à direita) consomem apenas 500nA. Com a propriedade do CMA de alterar apenas 1 Bit por vez, devido à simetria no processo de conversão, haverá apenas um inversor consumindo por vez, mas todos os comparadores consomem a fonte de corrente de forma contínua.



Imagem 4.7.2-3: Curvas do Comparador Com e Sem o Inversor

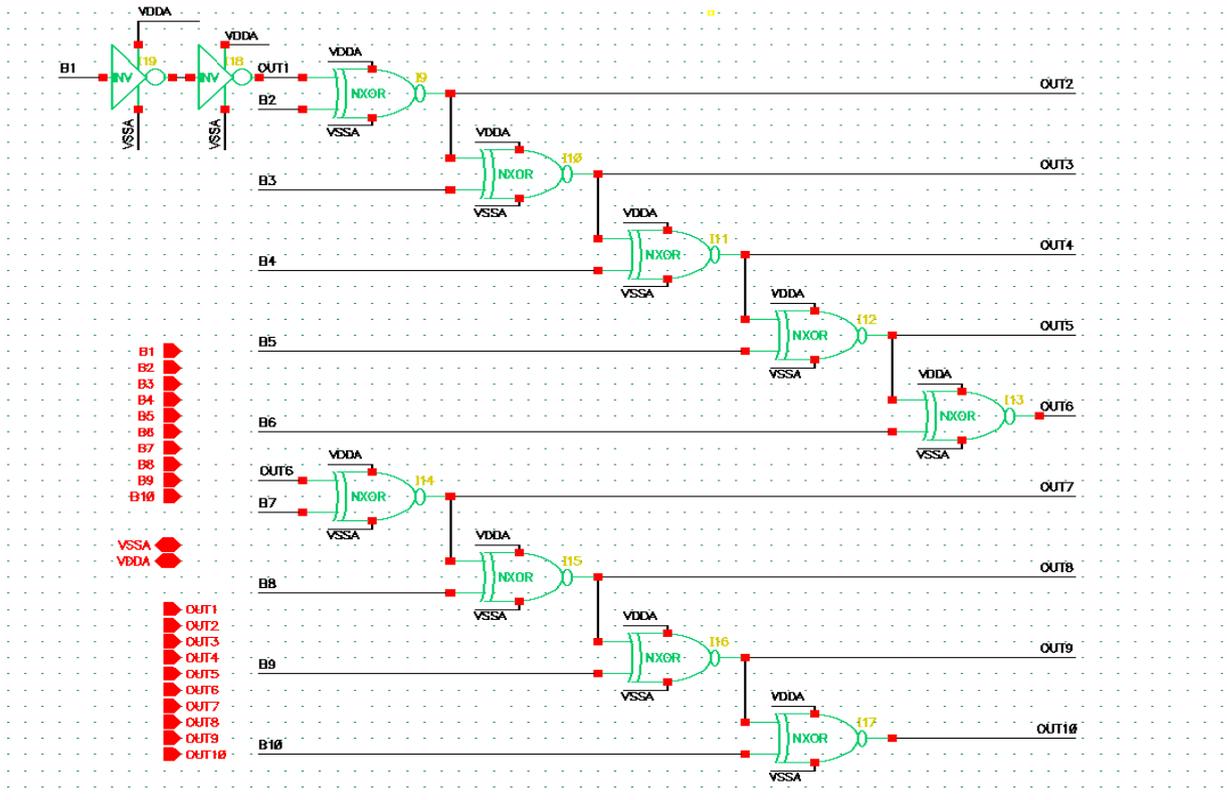


Imagem 4.7.2-4: Esquemático do Circuito do Tradutor

É importante notar que não há a necessidade de uma referência de tensão para o comparador, pois ele utiliza a própria variação das tensões nos GATES dos By-Pass e dos Espelhos, conforme seção 4.6.1.3, para realizar a operação. O fluxo de corrente quando injetado no bloco do By-Pass, fluirá pelo caminho NMOS e, caso seja retirado, fluirá pelo caminho PMOS, portanto, o ajuste da fonte de corrente do circuito somador seria suficiente para anular o OFFSET, uma vez que elas desviam o ponto de inversão do módulo.

O circuito tradutor não é um circuito essencial para o conversor, pois o circuito em questão realiza apenas uma interpretação dos dados já convertidos. Entretanto, os cálculos e processos já atuam com um formato digital padrão e, portanto, o uso de uma lógica para tratamento dos dados obtidos no formato primário do conversor seria frequente. O interessante do esquemático do tradutor, imagem 4.7.2-4, apresenta a configuração completa do circuito para converter a lógica de um conversor de 10 bits. Conforme a imagem, pode-se verificar que são utilizados apenas elementos lógicos, ou seja, apenas processamento digital do sinal que vem do conversor, uma vez que a informação que chega ao tradutor traz, consigo, a informação dentro da resolução desejada e, portanto, esses dados são suficientes para realizar os processamentos necessário e/ou desejado com esses valores.

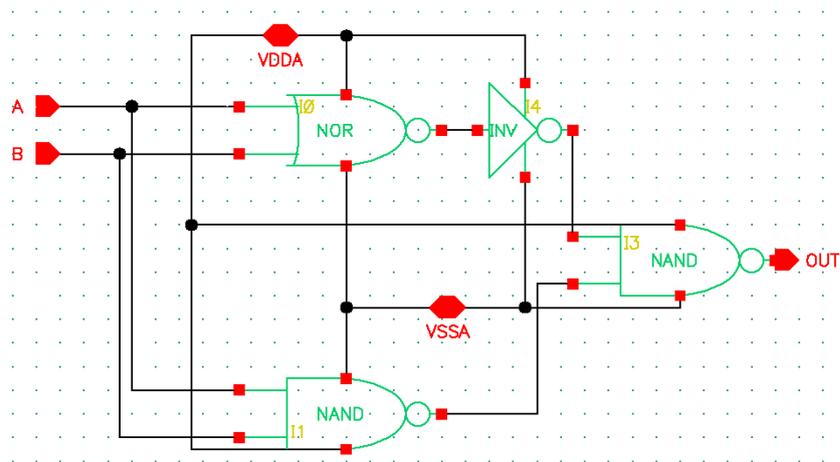


Imagem 4.7.2-5: Esquemático do Circuito da XNOR.

Cada uma das portas lógica XNOR, na imagem 4.7.2-4, são compostas por um circuito de portas lógicas simples, imagem 4.7.2-5. Essas portas básicas foram projetadas para o circuito e são representadas na imagem 4.7.2-6.

$$XNOR(A, B) = A * B + \bar{A} * \bar{B} \quad (4.7.2-1)$$

$$XNOR(A, B) = \overline{A * B + \bar{A} * \bar{B}} = \overline{A * B} * \overline{\bar{A} * \bar{B}} = \overline{A * B} * (A + B) \quad (4.7.2-2)$$

$$XNOR(A, B) = \overline{A * \bar{B} + \bar{A} * B} = \overline{(A * \bar{B})} * \overline{(\bar{A} * B)} \quad (4.7.2-3)$$

A diferença entre as equações 4.7.2-2 e 4.7.2-3 - que são duas resoluções para a mesma porta lógica - está no circuito a ser realizado para cada equação, pois na segunda verifica-se a necessidade de inverter cada uma das entradas, ou seja, já seriam adicionados dois circuitos inversores para cada entrada e, dessa forma, haveria a adição de mais duas portas inversoras. Veja que a necessidade de adicionar uma inversora para obter a porta OR, que possui topologia similar à topologia da porta AND, é compensada devido a porta AND ser obtida com a inversão da porta NAND [40]. As topologias das portas NAND e NOR estão na Imagem 4.7.2-6. Dessa forma, a equação 4.7.2-2 não é apenas uma solução, mas representa um circuito lógico otimizado.

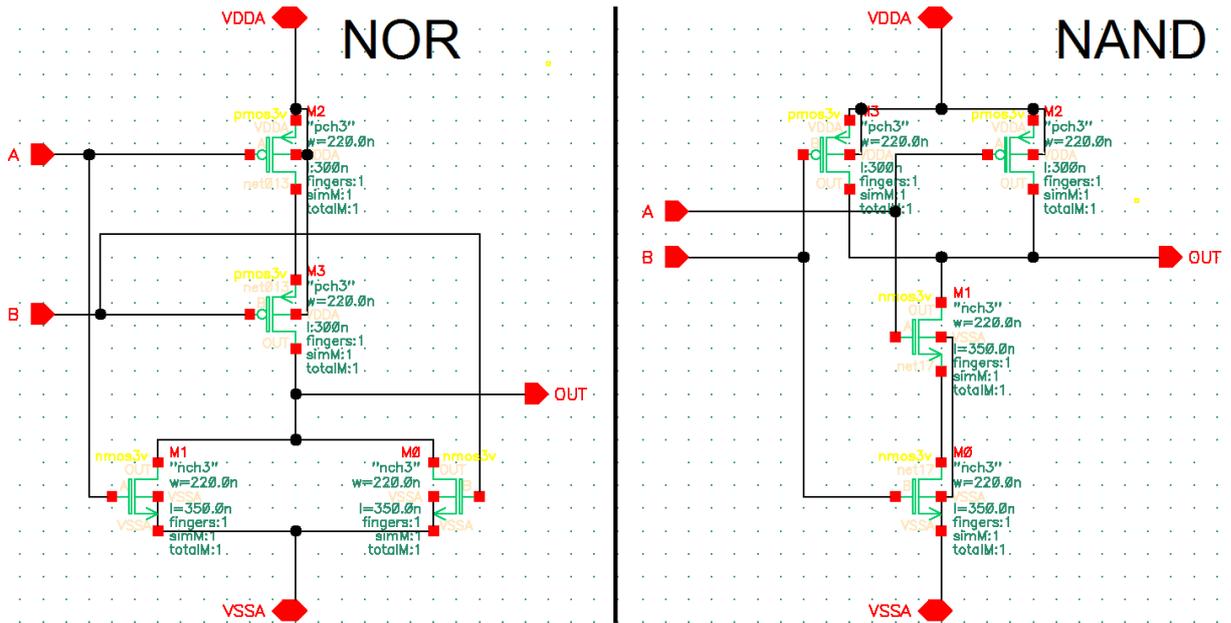


Imagem 4.7.2-6: Circuitos das portas NOR e NAND

### 4.7.3. OFFSET

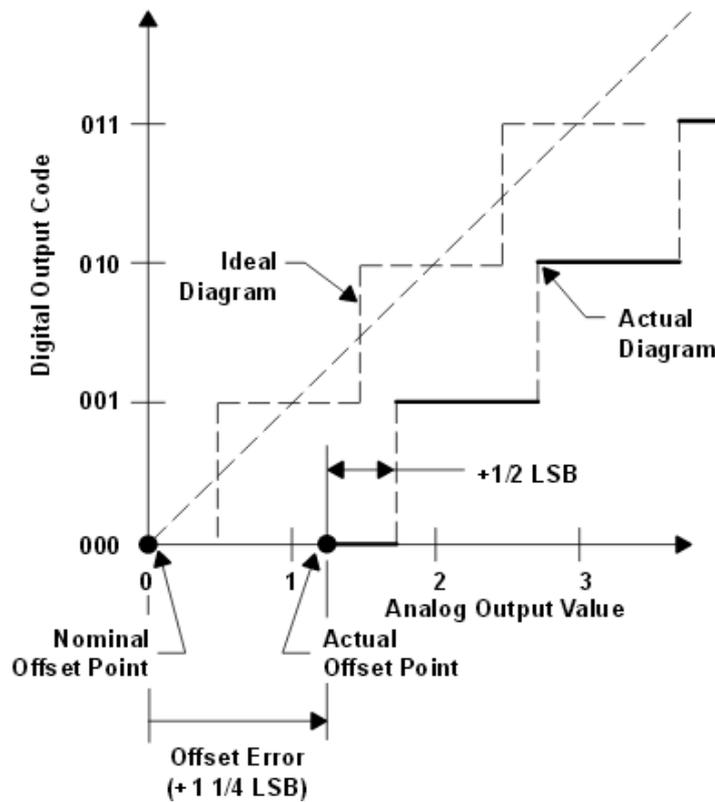


Imagem 4.7.3-1: Erro de Offset. [41]

O erro de offset, como mostrado na Imagem 4.7.3-1, é definido como a diferença entre os pontos nominais e reais do offset. Para um ADC, o ponto de deslocamento é o valor analógico quando a saída digital é zero. Ele afeta todos os códigos igualmente, podendo ser compensado por um

processo de corte. Sem a possibilidade do corte para compensar do erro, o offset passa a ser o fundo de escala do ADC.

O erro de OFFSET, no caso do conversor, ocorre por erros em espelhos de corrente e por fuga do circuito By-Pass. Sendo a fuga de corrente do By-Pass mais simples de tratar do que o processo de cópia da corrente. As variações do Vds podem aumentar ou reduzir a inclinação da curva do módulo, o que implica na perda de níveis ou na falta de níveis para atender toda a faixa analógica de interesse. Algumas simulações foram realizadas para valorizar os erros causados pelos espelhos de corrente e demonstrar como eles ocorrem no conversor.

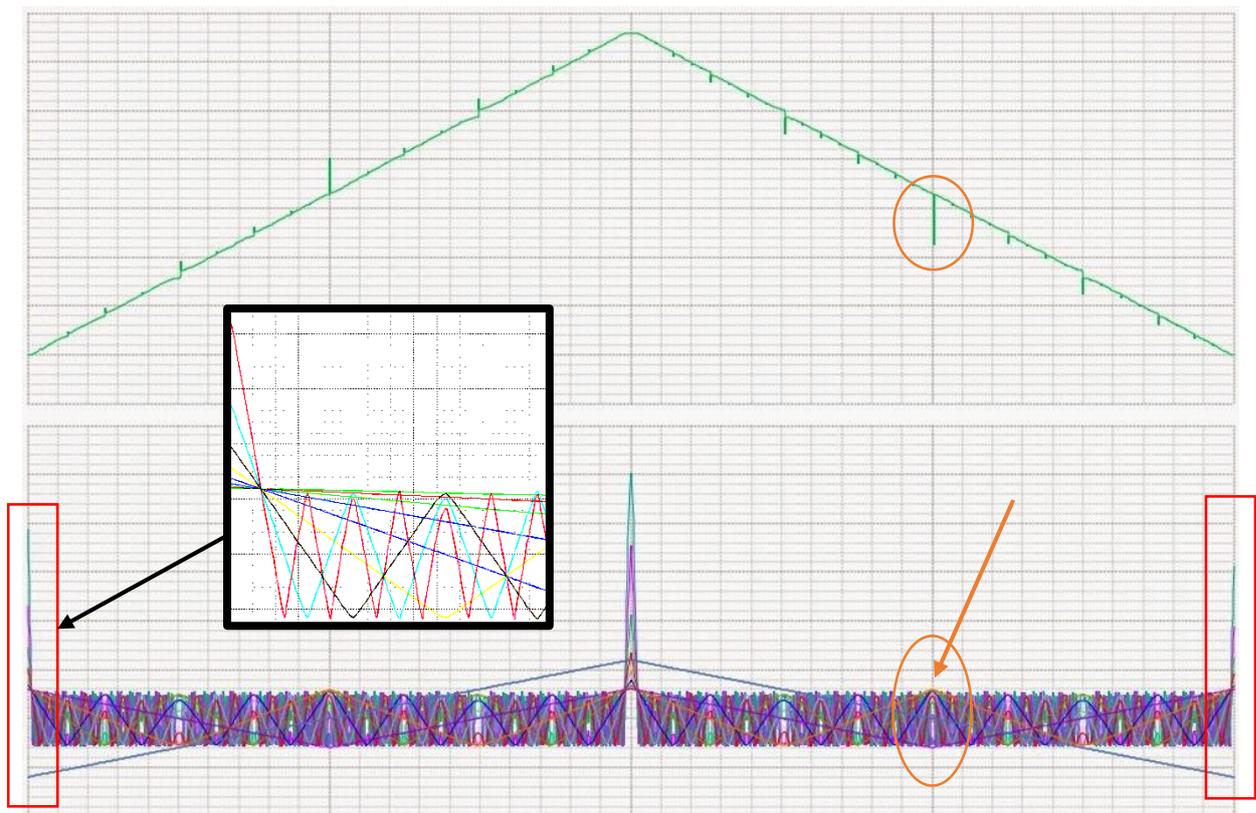


Imagem 4.7.3-2: Simulação Com ERRO de OFFSET

Conforme a imagem 4.7.3-2, pode-se verificar que os sinais dos módulos passam a crescer de forma contínua. Esse processo mostra que para os valores extremos, o conversor está saturando, ou seja, a corrente que está circulando entre os blocos é menor ou maior que a faixa em que o conversor consegue tratar o sinal. Por exemplo, digamos que o bloco básico, que é o gerador do módulo, opere com uma fonte para referência de corrente de  $5\mu\text{A}$ , mas o erro do espelho sofre variação no Vds e, portanto, passa a apresentar  $5,1\mu\text{A}$  o bloco seguinte, portanto, sobraram  $100\text{nA}$ . No caso de 10 bits, esse valor seria multiplicado por 1024 e passaria a ser  $1\mu$ . Finalizando, a corrente excederia em um  $1\mu$ , totalizando 20% da amplitude do sinal, que é a corrente vista na imagem 4.7.3-2, extrapolando valores de corrente apresentados na faixa normal de operação normal, esse processo ocorre na imagem para a faixa central e laterais destacadas em vermelho, uma vez que ocorrendo esse processo

na faixa central, não ficaram mapeados como ERRO de OFFSET, mas sim ERRO de GANHO. Como o conversor possui a lógica simétrica e a faixa central possui simetria com os extremos (observe pela imagem que as laterais são correspondentes aos valores mínimos do sinal de entrada, acima, sendo representado pela rampa gerada pelo conversor).

A seta laranja na imagem 4.7.3-2 mostra o erro gerador do OFFSET que também causa reflexos em pontos intermediários do processo de conversão, pois a mesma ocorrência existe em cada sinal e, apesar de não depender do valor binário definido pelos blocos mais significativos, o sinal se propaga de um bloco para outro, portanto, eles são independentes e repetem o mesmo erro ao longo da faixa, mas nesse caso passam a constituir erros de INL e DNL. Essa simulação foi realizada com diversos erros para observar os reflexos das falhas no comportamento final.

#### 4.7.4. ERRO DE GANHO

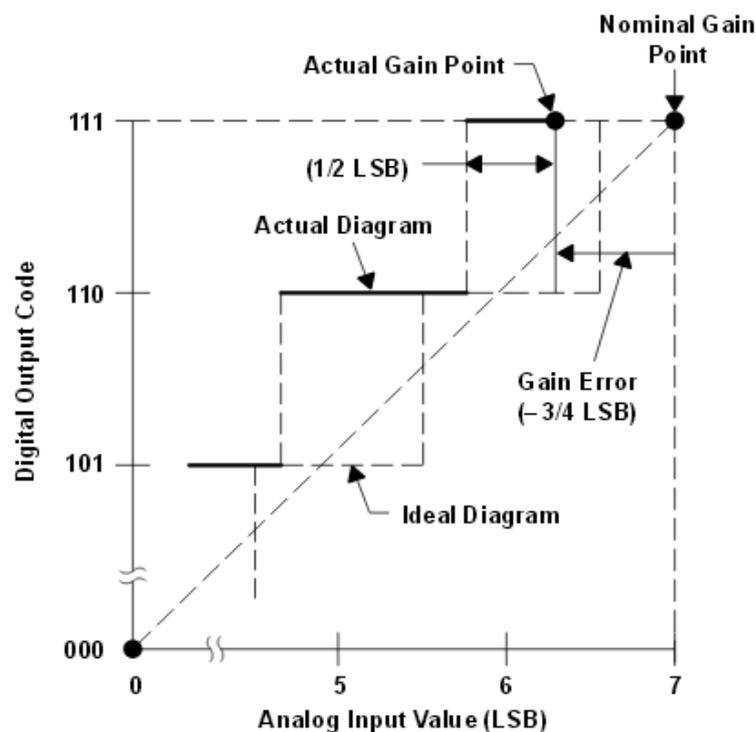


Imagem 4.7.4-1: Erro de Ganho. [41]

O erro de ganho representado na Imagem 4.7.4-1, definido como a diferença entre os pontos de ganho nominais e reais na função de transferência, após o erro de desvio tiver sido corrigido para zero, ou seja, eliminado o offset ou definindo-o como fundo de escala.

Para uma ADC, o erro do ganho é o valor quando a saída digital é o fundo de escala. Este erro representa uma diferença na curvatura das funções reais e ideais de transferência e, como tal, corresponde à mesma percentagem de erro em cada passo. Geralmente é de fácil ajuste, quando linear.

Da mesma forma que no OFFSET, os erros de cópia levam há uma corrente residual, que no caso de um erro dos espelhos, deve multiplicar o sinal por 1, pois é uma cópia. Entretanto estão multiplicando o sinal por valores maiores, resultando na visualização do processo ao centro da imagem 4.7.4-2. Observe que cada sinal sobe a uma altura diferente e que o valor tende a dobrar. Esse processo se deve à propagação do erro e da sua significância, ou seja, a significância aumenta em 2 bit a cada bit menos significativo que recebe o sinal em seu bloco básico. Assim, o desvio pode ser estimado utilizando a simulação de um módulo básico, pois o efeito é propagado de forma contínua.

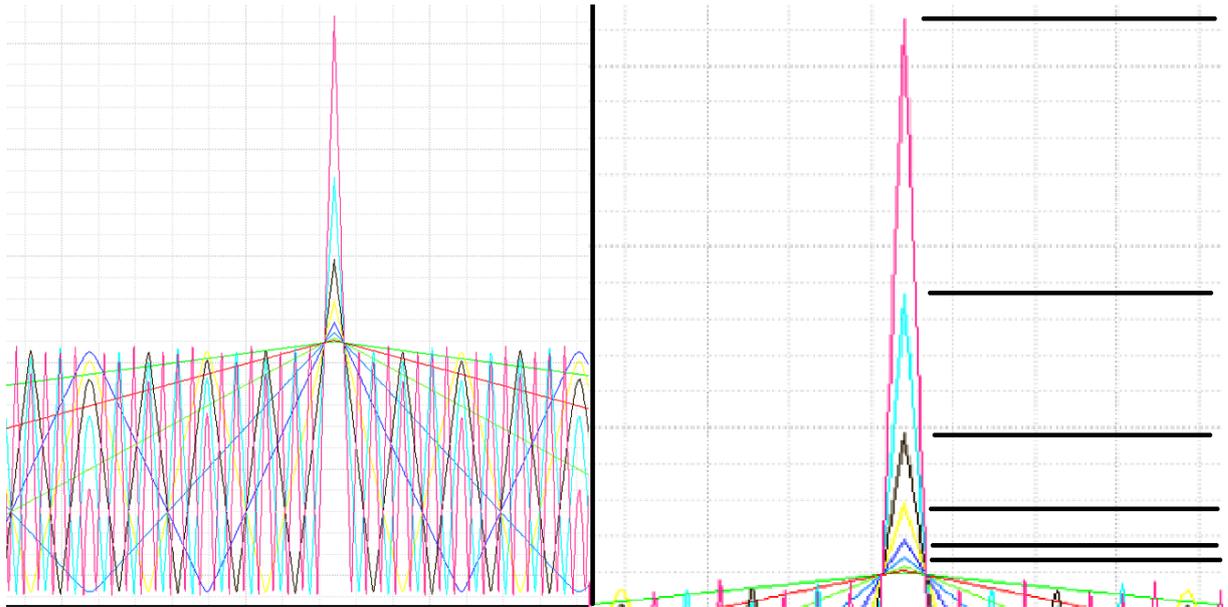


Imagem 4.7.4-2: Erro de Ganho devido ao excesso de Corrente.

#### 4.7.5. DNL – DIFFERENTIAL NONLINEARITY

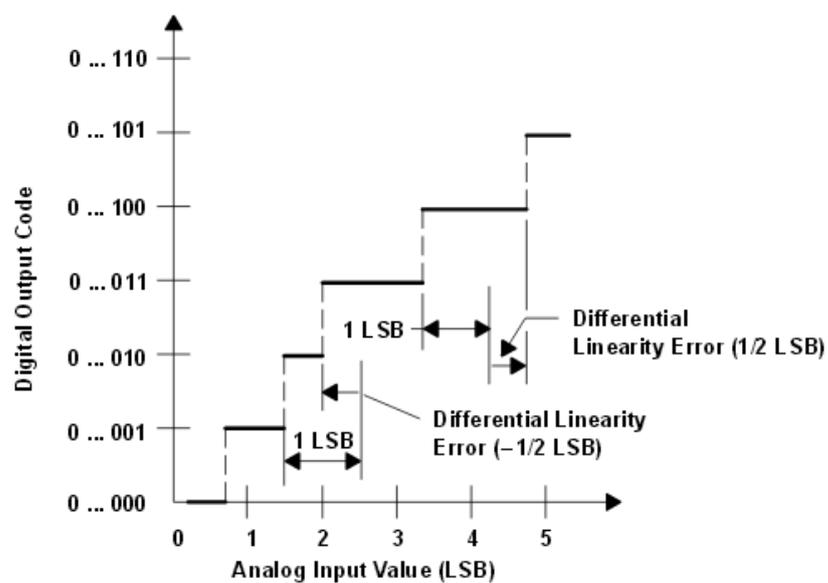


Imagem 4.7.5-1: ERRO DE DNL. [41]

O DNL mostrado na Imagem 4.7.5-1, é a diferença entre a largura do passo real (para um ADC) ou altura do degrau (para um DAC) e o valor ideal esperado para a variação de 1 LSB. Portanto, se a largura ou altura é exatamente a de 1 LSB, o DNL é zero. Se o LSB DNL excede 1, ou seja, um valor convertido está tomando parcialmente outra faixa de conversão, existe a possibilidade do conversor não ser monótono, variando a saída digital de forma desconexa com a entrada analógica. Surge uma grande possibilidade de algum dos valores digitais previstos não ser representado dentro da faixa analógica projetada para a conversão.

O CMD - por não utilizar o circuito de S/H - é sensível ao DNL e perde resolução com o aumento da frequência, portanto, a resolução do conversor é reduzida conforme aumenta a frequência de operação, pois o DNL, no CMD, fica caracterizado pelo atraso entre os blocos. Nesse estado crítico de operação, alguns valores representados por variações dos bits menos significativos deixam de ser representados, uma vez que são saltados, conforme imagem 4.7.5-2.

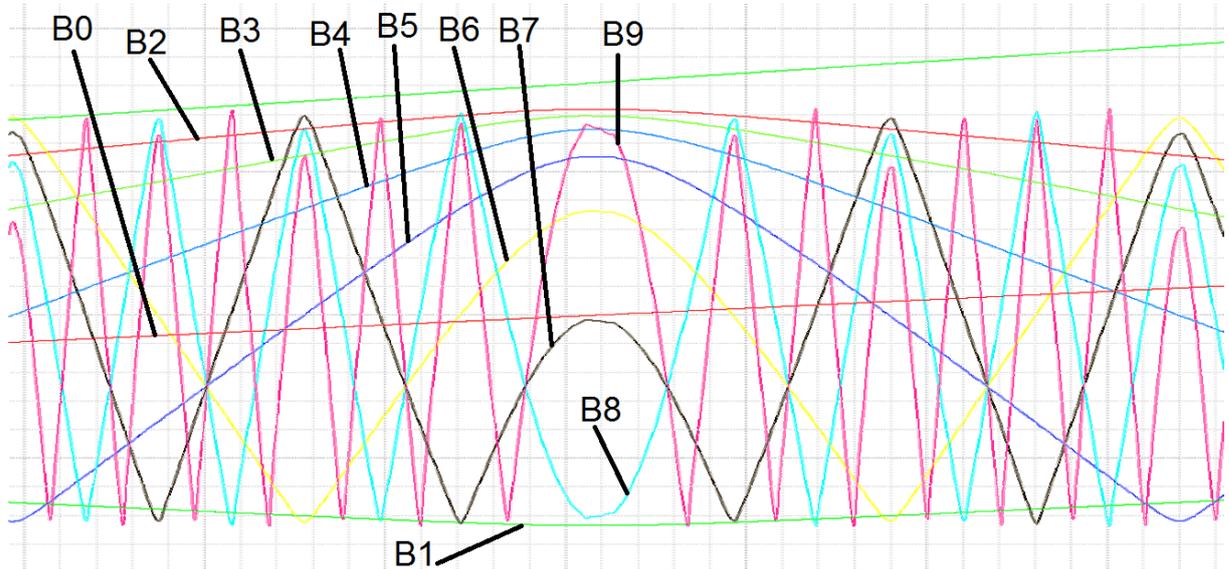


Imagem 4.7.5-2: Comportamento dos Sinais dos Módulos B0 é o mais significativo

Observe na imagem 4.7.5-2 que as curvas dos módulos se movimentam de forma independente, ou seja, o que ocorre com uma não interfere na outra. Veja que as curvas dos bits B6, B7, B8 e B9 não completaram o ciclo de operação para atender a resolução dentro da faixa. O bit B6 chega a cruzar metade da faixa, mas o 7, o 8 e o 9 não, ou seja, eles não operaram e não participaram da resolução nessa faixa.

Entretanto, devido à propriedade de simetria das faixas de conversão, ao retornarem aos níveis adequados de operação, esses bits voltam em valores válidos, imagem 4.7.5-3. Pode-se constatar que o bit B6 é o único dentre os quatro menos significativos que conclui a operação. Portanto, a diferença do nível é de 8 faixas analógicas. Aparentemente pela imagem, pode se confundir ao medir o comprimento dos degraus conforme a definição do próprio DNL, mas deve-se ressaltar que na Imagem

os degraus que realmente são válidos estão mais afastados do ponto de falha, devido ao atraso dos bits, o que reflete diretamente no DNL.

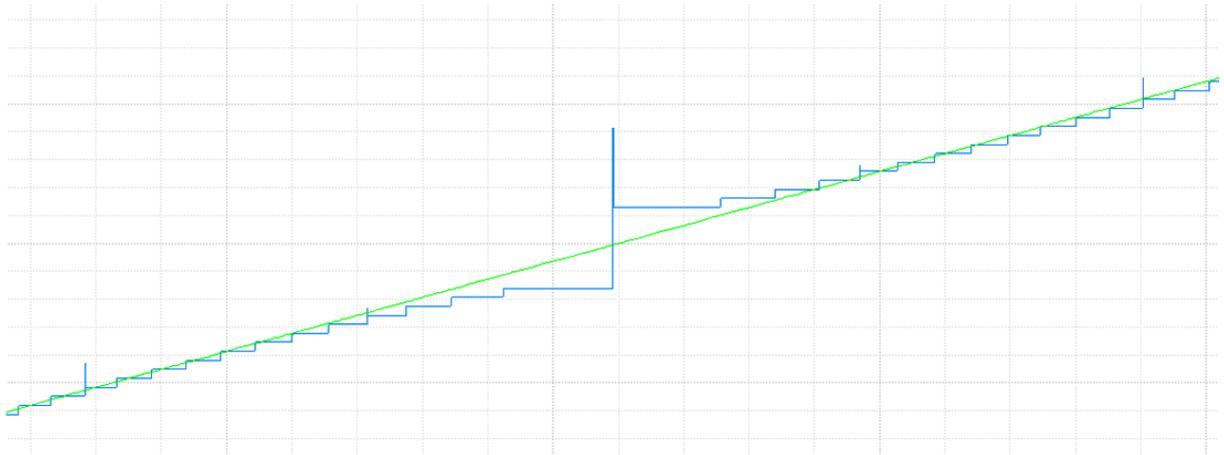


Imagem 4.7.5-3: Resultado na escada de valores convertidos do ERRO DE DNL

#### 4.7.6. INL – INTEGRAL NONLINEARITY

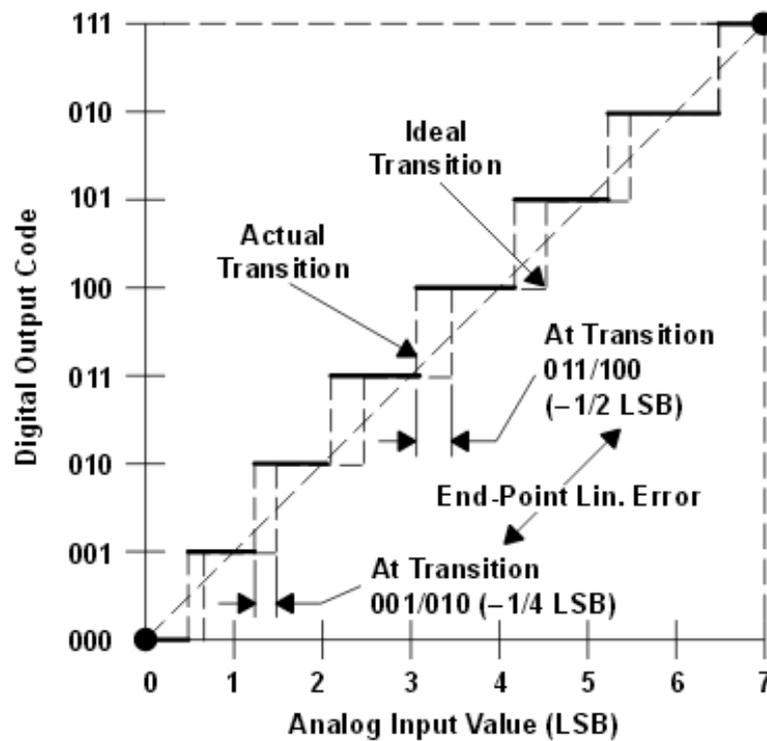


Imagem 4.7.5-1: ERRO DE INL. [41]

Voltando à Imagem 4.7.5-2, observe que os máximos para o INL e o DNL são próximos e o conversor retorna a operação adequada logo após o evento. As falhas em questão são repetidas de forma simétrica, destacando quando há coincidência das falhas de cada bloco básico do CMD dentro de um mesmo intervalo de tempo. Conforme a Imagem 4.7.5-3, na qual se verifica a repetição.

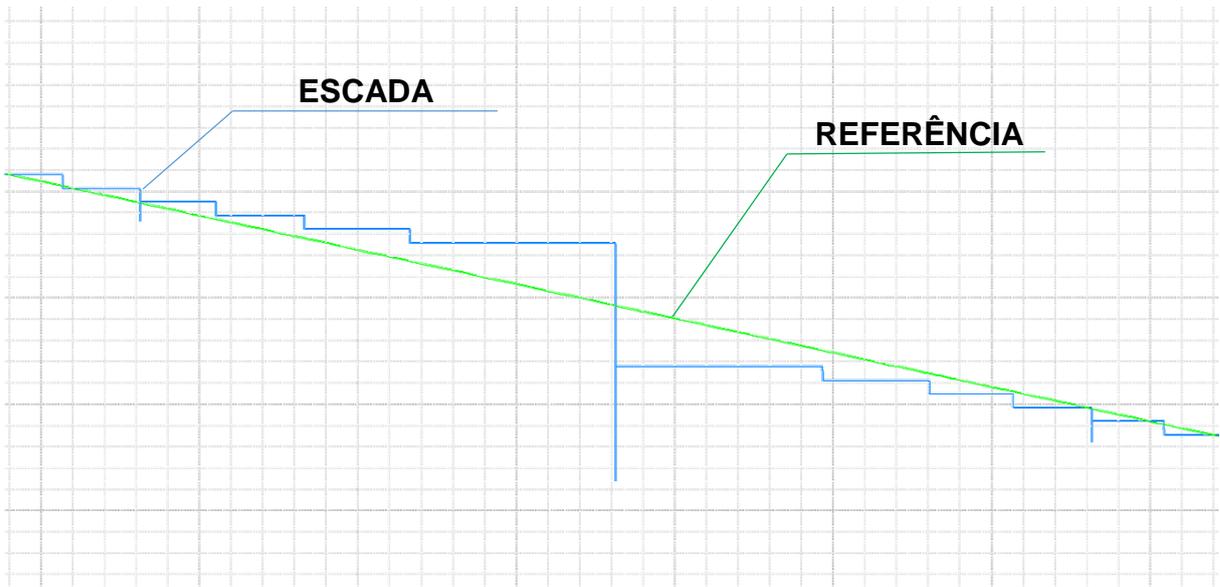


Imagem 4.7.5-2: Resultado na escada de valores convertidos dos ERROS DE INL e DNL, que possuem extremos nas mesmas faixas.

O INL (Integral Nonlinearity error) mostrado na “Imagem 4.7.5-1” (por vezes visto como simplesmente erro de linearidade) é o desvio dos valores sobre a função de transferência real de uma linha reta. Esta linha reta pode ser tanto uma linha reta contínua desenhada de modo a minimizar esses desvios ou pode ser uma linha traçada entre os pontos finais da função de transferência uma vez que o ganho e erro de deslocamento foram anulados. O segundo método é chamado de ponto final e linearidade é a definição usualmente adotada uma vez que pode ser verificada mais diretamente.

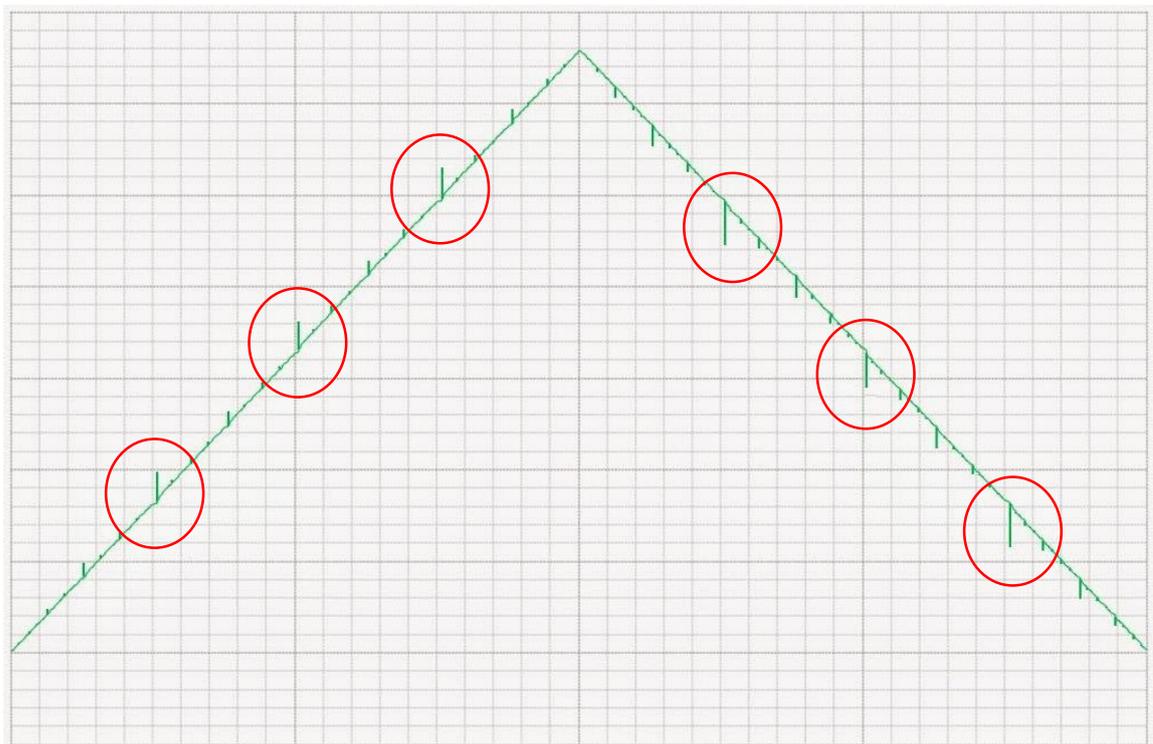


Imagem 4.7.5-3: Pontos críticos de INL e de DNL

Assim, o conversor perde parte de sua resolução mas de forma distribuída, pois, pela simetria e independência das curvas de módulo, os valores convertidos retornam aos pontos ótimos logo após a faixa de falha, conforme é observado na imagem 4.7.5-2, e também é visível o mesmo efeito na imagem seguinte 4.7.5-2, veja que as escadas à esquerda e à direita já estão próximas à referência.

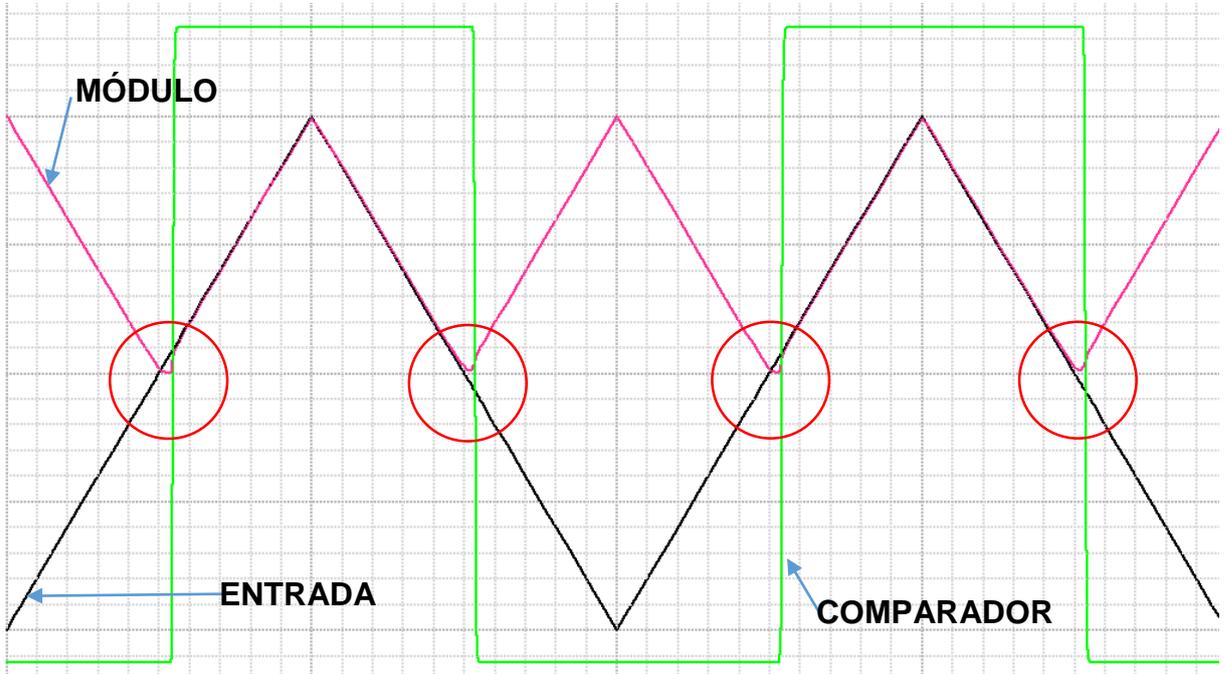


Imagem 4.7.5-4: Atrasos e Falha no Ponto de Inversão do Módulo em Baixa Frequência.

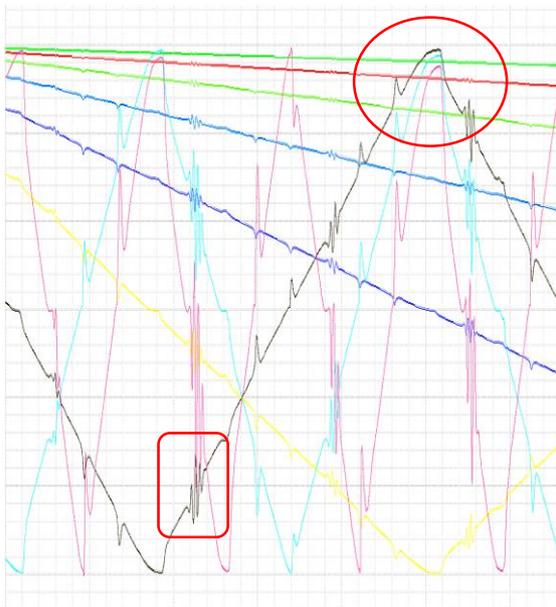


Imagem 4.7.5-5: Inversão do módulo em Alta Frequência Vgs do By-Pass em níveis baixos.

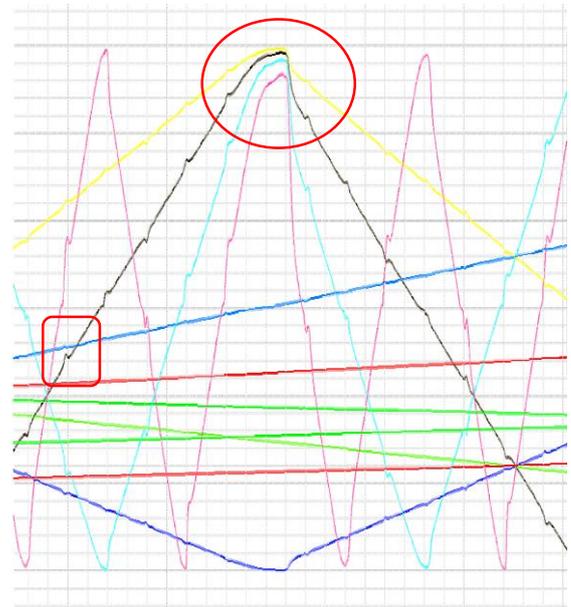


Imagem 4.7.5-6: Inversão do módulo em Alta Frequência Vgs em níveis adequados.

Os círculos vermelhos destacam os pontos de inversão, que são os pontos críticos do projeto, pois os erros que causam maiores impactos no ERRO de OFFSET e no ERRO de GANHO, estão

relacionados principalmente ao erro no conjunto de espelhos, entretanto, os erros relacionados ao resíduo do By-Pass limita a faixa de operação, pois elimina alguns valores digitais.

As imagens 4.7.5-5 e 4.7.5-6, tratam do mesmo circuito, entretanto, variando apenas o valor da polarização do Vgs, conforme seção 4.2.1.1, destinado aos dispositivos em COMUM GATE do By-Pass. Na primeira imagem, o ruído destacado nos retângulos vermelhos e inserido nos blocos devido à alta impedância aplicada pelo By-Pass, uma vez que os dispositivos passam a operar devido ao sentido obrigatório da corrente, enquanto o outro dispositivo ainda não descarregou completamente a carga armazenada em sua capacitância de GATE, prejudicando a saturação do dispositivo que deve entrar em operação. Na segunda imagem, a 4.7.5-6, a faixa está mais adequada à frequência de operação, ou seja, ainda ocorrem problemas relacionados ao conflito de operação dos dois dispositivos, o que gera ruído, mas o ruído é pequeno.

#### 4.7.7. SNR – SIGNAL-TO-NOISE RATIO

À Relação ou Razão SINAL RUÍDO, relação entre a potência do sinal de interesse com o ruído, pela a potência do ruído presente no sinal, ou seja, revela quantas vezes o sinal de interesse está superior ao ruído gerado e / ou aplicado sobre o sinal. No caso dos conversores não nos interessa saber o ruído que está presente no sinal a ser amostrado, pois o sinal de interesse também está com os ruídos deferidos pelos equipamentos de medição e pelo meio de propagação. O ruído, ao qual nos referimos nesse caso, é o gerado pelo processo de conversão e / ou o inserido durante o processo de conversão. O simples processo de utilizar um único valor digital para representar toda uma faixa analógica insere um ruído no processo, pois será mascarado quaisquer comportamentos que o sinal venha a ter dentro dos limites da resolução do processo ideal.

Assim, o ruído não é apenas um sinal adicionado ao processo, mas também surge do simples fato de não atendermos toda a resolução necessária para a leitura adequada do sinal de interesse, pois está perdida parte da informação e não se pode afirmar nada a respeito do comportamento do sinal dentro da faixa de erro aceitável para a resolução desejada. Dessa forma, não há a possibilidade de saber o que é sinal e / ou o que é exatamente ruído e, caso houvesse, o ruído já estaria eliminado e seria desprezível diante da resolução, portanto a relação é dada pela potência do sinal portanto o ruído pelo ruído mensurado.

$$SNR = \frac{POTÊNCIA_{SINAL} + POTÊNCIA_{RUÍDO}}{POTÊNCIA_{RUÍDO}} \quad (4.7.7-1)$$

Nos conversores, portanto, conforme o parágrafo anterior, o ruído limita a resolução do sistema, que já parte de um valor máximo possível para a resolução determinada pelo número de bits de projeto. Além desse limite, não há a possibilidade de gerar o circuito sem erro e, portanto, esses erros trarão desvios indesejados, que prejudicarão a resolução desejada. Assim, o nosso interesse em si, tratando

de conversores de sinais, é saber o quanto efetivamente está sendo obtido de resolução para o sinal amostrado por determinado conversor, surgindo a Imagem da seção seguinte, o ENOB.

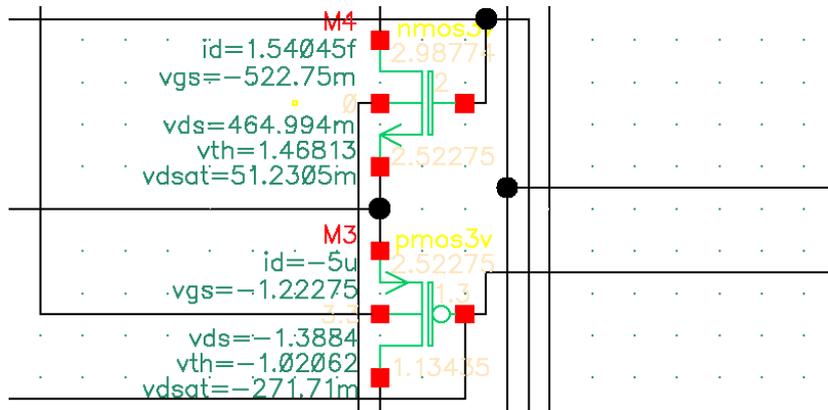


Imagem 4.7.7-1: Bloco By-Pass Operando com Entrada de + 5  $\mu$ A.

Na imagem 4.7.7-1 temos o bloco do By-Pass durante processo de validação. Estão sendo injetados 5  $\mu$ A e desviados para a região de menor potencial, no caso o GND. Note que para esse nível de corrente a operação está bem definida, uma vez o dispositivo M3 já está em saturação e o Vds é maior que a diferença entre Vgs e o Vth, ou seja, conforme a imagem, 1. Note que a tensão absorvida pelo Vds de M3 tira de operação o dispositivo M4, mas, mesmo em inversão forte, há uma corrente que flui pelo dispositivo M4, nesse caso, com o valor de 1,54 fA, que é desprezível, concluindo que o maior erro ocorre no processo de cópia entre os espelhos.

Na imagem 4.7.7-2 os dispositivos do By-Pass receberam um “B” antes do nome para que os dispositivos possuíssem a mesma numeração das imagens 4.6.1.1-1 e 4.6.1.2-1, que foram utilizadas para explicar o comportamento dessas topologias na seção 4.6. No circuito a corrente de 5  $\mu$ A é retirada do By-Pass e agora vem do VDDA para atender a retirada de corrente. Nesse caso a corrente de fuga já é 12,7 fA no By-Pass, veja também que a corrente nos dispositivos B M1 e B M2 já atingiram valores em pA, portanto, passam a receber corrente através do BULK do dispositivo B M3. Ao final de todo o processo, a corrente copiada é inserida ao bloco de espelho e processada até sua saída pelo dispositivo M6, que já está com 14,8 pA de fuga. Entretanto, essa não é a corrente que implica em maiores perdas de resolução, mas sim a corrente proveniente do erro de cópia entregue pelo dispositivo M8, que atinge 5 nA, portanto, o ruído inserido por esse bloco é cerca de 338 vezes maior.

O ruído inserido através do dispositivo M8 já limita a resolução de operação para o bloco ao valor máximo de 9,67 bits efetivos, como serão mostrados na seção posterior. Dessa forma, note que caso calculássemos o valor ignorando o ruído que foi adicionado ao próprio sinal, teríamos a relação direta 9,96 bits efetivos, pois nos transpareceria que o sinal está sendo efetivamente copiado, entretanto, isso não ocorre, uma vez que não se pode considerar características absolutas para o ruído inserido, pois nas faixas de transição, os valores podem ser considerados em mesmo sentido do sinal

de interesse quando em um nível e em sentido oposto quando no outro nível adjacente, conforme a imagem 4.7.7-2.

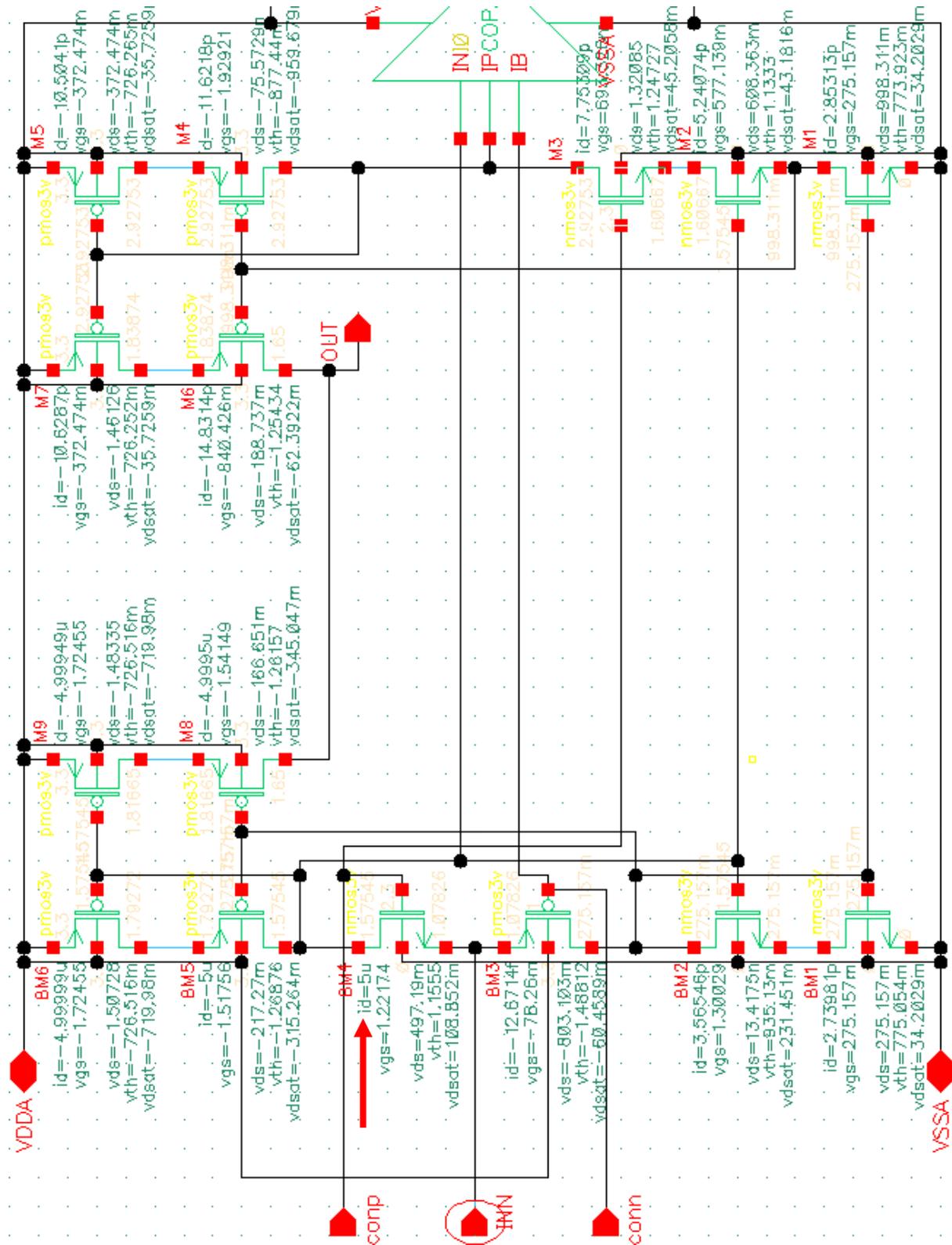


Imagem 4.7.7-2: Bloco By-Pass Operando com Entrada de - 5  $\mu A$ .

O ruído assimétrico em relação ao nível real do sinal. No caso do sinal de interesse 1, somado ao ruído, esse aproxima o sinal da faixa de transição de forma inoportuna. Mas no caso do sinal de interesse 2, o ruído somado não prejudica em mesma intensidade, assim, mas não faria sentido considerar apenas o evento oportuno para o ruído, uma vez que o ruído que ora é oportuno, em outro momento é prejudicial. Então, o ruído é somado ao sinal e não se pode separar o ruído do sinal. O valor de corrente em questão, atingiria a efetividade de 10 bits apenas quando a relação sinal ruído fosse 1254 vezes maior que o ruído e não simplesmente 1024 vezes, assim o maior de seus erros devido ao ruído deveria possuir o valor de 3,98 nA.

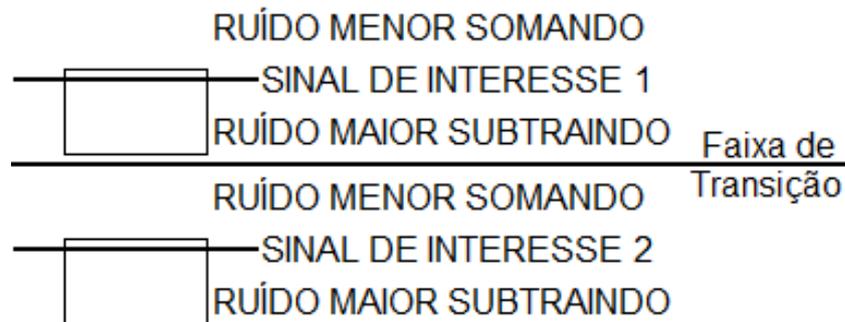


Imagem 4.7.7-3: Distribuição do ruído em torno do sinal.

#### 4.7.8. ENOB – EFFECTIVE NUMBER OF BITS

Conforme abordado na seção do SNR, mesmo o processo perfeito de conversão realizado por um conversor ideal adicionará ruído ao processo devido à condensação de toda a informação de uma faixa de valores analógicos para um único valor digital. Dessa forma, o processo em si já insere ruído e esse ruído está presente no sinal, conforme a equação 4.7.8-1. Assim, o número efetivo de bits o ENOB é calculado levando em consideração o ruído que é inserido pelo processo de conversão do sinal e considerando também o ruído gerado pelo processo analógico de tratamento do sinal em questão. A equação final para o ENOB fica:

$$ENOB = \log_2|SNR| - \log_2|POTÊNCIA_{QUANTIZAÇÃO}| \quad (4.7.8-1)$$

$$ENOB = \frac{SNR_{dB} - 10 * \log_{10}|POTÊNCIA_{QUANTIZAÇÃO}|}{20 * \log_{10} 2} \quad (4.7.8-2)$$

A equação para o número de bits fica, portanto, correspondente à equação 4.7.8-1, mas note que o SNR é, nesse caso, uma relação entre a intensidade do sinal, seja ele corrente ou tensão, pelo ruído em mesma grandeza física, entretanto, a Potência de Quantização está em grandeza bem definida e, portanto, não é correto equacionar como razão dentro do logaritmo do SNR. Dessa forma, resta avaliar qual seria o valor para esse ruído de quantização que é inserido pelo processo ideal de conversão.

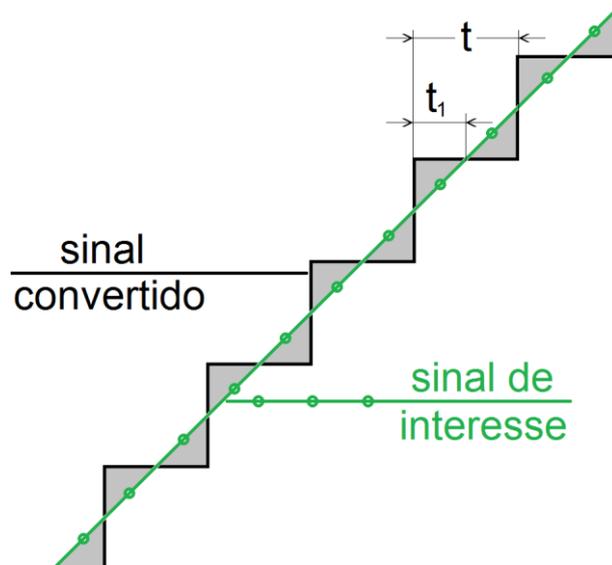


Imagem 4.7.8-1: Rampa acompanhada por uma conversão Ideal

Na imagem 4.7.8-1 temos uma rampa sendo acompanhada por um processo ideal de conversão, ou seja, não há erro de offset, não há erros de INL ou de DNL e não há erro de ganho, além disso o sinal convertido não possui atrasos ao processar o sinal de interesse e também não deforma o sinal de interesse nos processos de amostragem desse sinal. O erro de amostragem inserido pelo conversor ideal, portanto, possui potência proporcional à área pintada no interior dos triângulos gerados pelo erro de quantização. Note que caso o sinal de interesse não cruzasse exatamente no meio do degrau, gerado pela escada do sinal convertido, a área final seria maior, conforme as equações seguintes de 4.7.8-3 a 4.7.8-6, portanto o conversor ideal deve dividir o erro de quantização, por nível digital, em dois triângulos iguais para obter o erro mínimo.

$$A_{DEGRAU} = \frac{1}{2} \left( \frac{LSB \times t_1}{t} \times \frac{t_1}{2} + \frac{LSB \times (t - t_1)}{t} \times \frac{(t - t_1)}{2} \right) \quad (4.7.8-3)$$

$$\frac{\partial A_{DEGRAU}}{\partial t_1} = \frac{1}{2} \left( \frac{LSB}{t} \times t_1 - \frac{LSB}{t} \times (t - t_1) \right) = 0 \quad (4.7.8-4)$$

$$\frac{\partial A_{DEGRAU}}{\partial t_1} = 0 \quad \xrightarrow{\text{temos}} \quad t_1 = \frac{t}{2} \quad (4.7.8-5)$$

$$A_{DEGRAU_{mínima}} = \frac{1}{4} (LSB \times t) \quad (4.7.8-6)$$

Dessa forma, podemos calcular a potência inserida por cada um desses degraus conforme a área mínima calculada. O valor de interesse será o valor eficaz da potência gerada pelo ruído de quantização que será dada pela equação pelo valor RMS da curva do ruído que é apresentada na imagem 4.7.8-2.

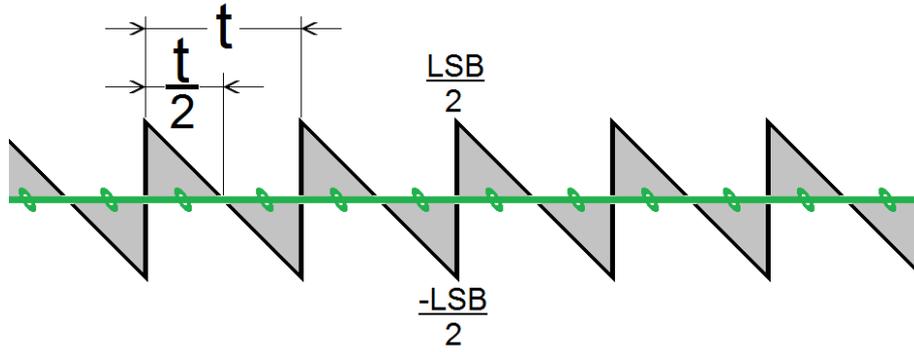


Imagem 4.7.8-2: Sinal do Ruído de Quantização Ideal

Para calcularmos o erro derivado do sinal gerado pelo conversor ideal, consideraremos o valor inicial de integração, referente de um ponto de abscissa com ordenada igual a zero, do intervalo de menos meio até meio período do ruído do sinal de quantização, portanto teremos a equação 4.7.8-7.

$$\text{RMS}[\text{RUÍDO}_{\text{QUANTIZAÇÃO}}] = \sqrt{\frac{1}{t} \int_{-\frac{t}{2}}^{\frac{t}{2}} \left[ \text{LSB} * \frac{x}{t} \right]^2 dx} = \sqrt{\frac{\frac{t}{2} \text{LSB}^2}{t^3} \times \frac{x^3}{3}} = \frac{\text{LSB}}{\sqrt[3]{12}} \quad (4.7.8-7)$$

$$\text{SNR}_{\text{C\_IDEAL}} = \frac{\frac{\text{FSR}}{2 \times \sqrt[3]{2}}}{\frac{\text{LSB}}{\sqrt[3]{12}}} = \frac{\frac{\text{FSR}}{2 \times \sqrt[3]{2}}}{\frac{\text{FSR}}{2^N \times \sqrt[3]{12}}} = \frac{2^N \times \sqrt[3]{12}}{2 \times \sqrt[3]{2}} = 2^N \times \sqrt[3]{\frac{3}{2}} \quad (4.7.8-8)$$

Tomando a potência de uma senóide com amplitude suficiente para atender toda a faixa de valores analógicos que são tratados pelo conversor ideal, podemos obter o SNR do conversor. Lembrando que o valor RMS para uma senóide é sua amplitude dividida por  $\sqrt[3]{2}$ , temos a equação 4.7.8-8. Da equação aplicamos o logaritmo para obtermos N de forma isolada.

$$\log|\text{SNR}_{\text{C\_IDEAL}}| = \log 2^N + \log \sqrt[3]{\frac{3}{2}} = N \times \log 2 + \frac{1}{2} \log \left| \frac{3}{2} \right| \quad (4.7.8-8)$$

$$N = \frac{2 * \log|\text{SNR}_{\text{C\_IDEAL}}| - \log \left| \frac{3}{2} \right|}{2 * \log 2} = \quad (4.7.8-9)$$

Relacionando as equações 4.7.8-2 com 4.7.8-9. Temos a equação que aplicaremos.

$$\text{ENOB} = \frac{\text{SNR}_{\text{dB}} - 10 \times \log \left| \frac{3}{2} \right|}{20 * \log_{10} 2} \quad (4.7.8-9)$$

Ainda, conforme [5], consideremos as distorções causadas pelos efeitos dinâmicos como atrasos causados pela frequência do sinal amostrado. Assim, apesar do ruído possuir pouca potência, há perda do sinal de interesse, convertendo em ruído, devido à distorção causada pela amostragem.

$$\text{ENOB}_{\text{SINAD}} = \frac{\text{SINAD} - 10 \times \log \left| \frac{3}{2} \right|}{10 * \log_{10} 2} \quad (4.7.8-10)$$

Assim, no lugar de considerarmos apenas a relação entre o sinal e o ruído, consideram-se também as distorções inerentes ao processo de amostragem e determinação do valor binário. O ENOB é referido como figura de mérito para alguns trabalhos como [5].

#### 4.7.9. FIGURA DE MÉRITO PARA EFICIÊNCIA

Acompanhando uma tendência de trabalhos bem referenciados da área [1] e [6], para os conversores de sinais analógicos para digitais, considera-se o consumo do circuito pelo número de faixas de valores caracterizados pelo conversor. Essa medida determina qual o gasto por conversão, ou seja, traz a independência com relação à frequência e / ou com a resolução, pois o consumo é dividido pelos dois, portanto, não há questões comparativas em relação à qual conversor possui maior resolução e / ou frequência de amostragem, sendo um valor fiel e independente de arquitetura para tratar da eficiência do conversor dentro do processo de conversão. Adotaremos a equação:

$$\text{FME} = \frac{I_{\text{MÉDIA}} \times VDD}{f \times 2^{\text{ENOB}}} \quad (4.7.9-1)$$



## 5. RESULTADOS E DISCUÇÕES

Nesse capítulo apresentaremos os resultados obtidos com os projetos e simulações em ambiente CADENCE. A ideia é apresentar quais foram os resultados obtidos e comentá-los em relação aos dois projetos básicos, sendo um conversor simples de 10 bits realizado apenas para ilustrar um projeto inicial com conceitos simplórios de projeto, ou seja, sem circuitos que viessem a exigir uma eletrônica mais aprofundada. O outro projeto é de um conversor com 14 bits e consumo menor que 500nV caracterizando como um projeto Ultra Low Power.

Por se tratar do primeiro trabalho do CMD (Conversor do Módulo da Diferença) não queremos saturar o trabalho de informações, mas apresentar um trabalho simples de fácil acesso. A tecnologia utilizada é a AMS 0.18.

### 5.1. RESULTADOS DO PROJETO DE 10 BITS

Na imagem 5.1-1 temos os primeiros valores do conversor para uma frequência de 2MS/s, mas devemos lembrar que o conversor é assíncrono, portanto, não possui nenhum parâmetro fixo de amostragem para o sinal. Assim, a taxa de 2MS/s é definida pelo intervalo de tempo que o sinal demora para sair de uma faixa para uma outra faixa analógica que definem valores digitais diferentes e o atraso dos espelhos de corrente para a propagação do sinal analógico entre os blocos básicos do CMD

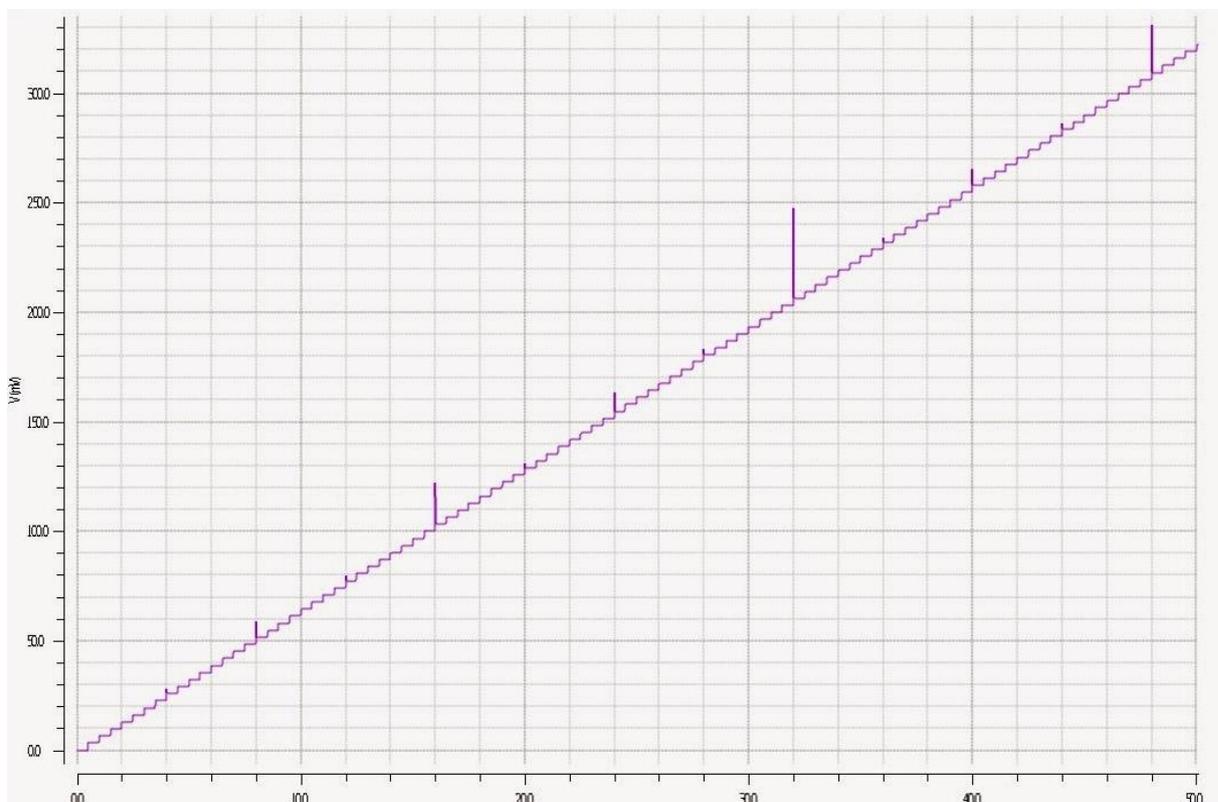


Imagem 5.1-1: 100 Primeiros Valores da Escada Gerada para Entrada de 2 MS/s



Imagem 5.1-2: Escada Gerada com 2 MS/s

Na imagem 5.1-2 mostrando os 1024 valores convertidos na rampa completa. Para avaliar o circuito não foram considerados os glitches, pois são e inerentes ao processo de conversão do DAC, que é ideal e oscila para representar os valores gerados pelo TRADUTOR, isso pode ser notado ao observarmos as ondas geradas por cada bit individualmente e percebermos que os bits não possuem variação simultaneamente ente si, portanto, não haveria razão para considerar que o glitch é gerado devido o processo do CMD. Os bits são apresentados na imagem 5.1-3.

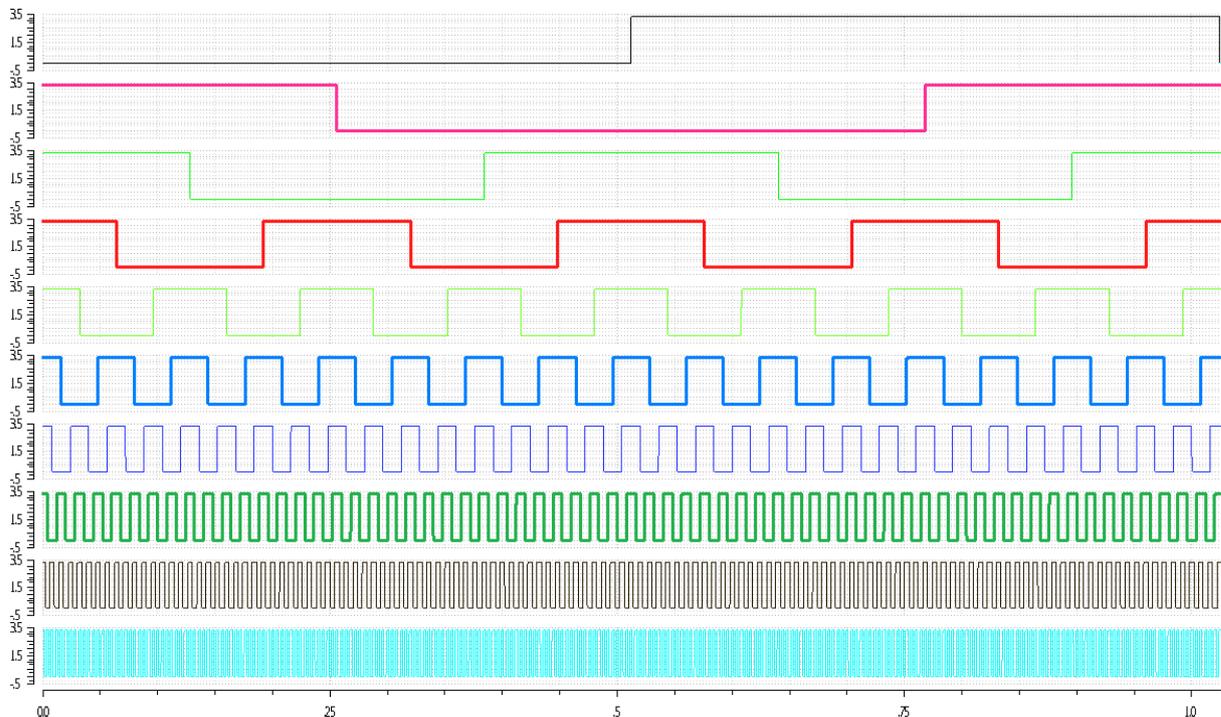


Imagem 5.1-3: Bits Gerado Pelo CMD

No intervalo de 768  $\mu$ s temos o maior glitch gerado, por isso, mostramos o sinal ampliado na Imagem 5.1-4. Nota-se a desproporcionalidade do sinal em relação ao degrau, pois o DAC utilizado, por ser um circuito criado com lógica a partir de dispositivos ideais, destaca de forma acentuada os atrasos por se tratar de um dispositivo ideal baseado na sequência dos números binários e não possui linearidade. Esse ponto também nos apresentou o maior valor para o DNL, o que já era esperado, pois, conforme vimos na seção 4.7.5, o processo possui maior possibilidade de falha na faixa central devido à corrente de módulo zero aplicada à entrada do By-Pass do bit mais significativo, que também é responsável pela propagação do sinal analógico para os demais bits e o erro derivado do bloco é propagado em dobro a cada bit menos significativo. Além desses fatores, o sinal também é mais suave, pois a cada bit a taxa de variação do sinal é dobrada, exceto do primeiro bit para o segundo, que possuem a mesma taxa de variação em termos da corrente de entrada em relação ao tempo.

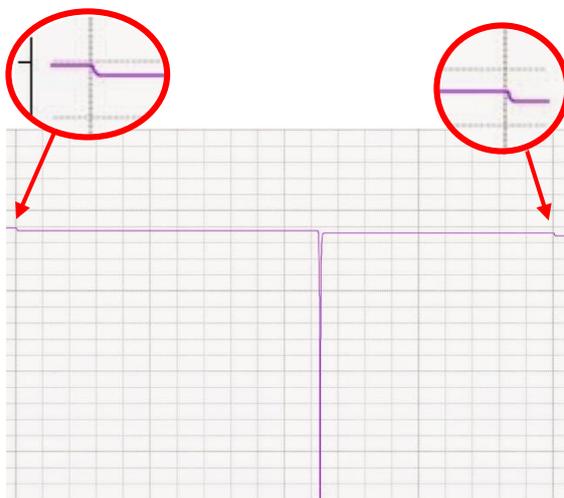


Imagem 5.1-4: Máximo do DNL

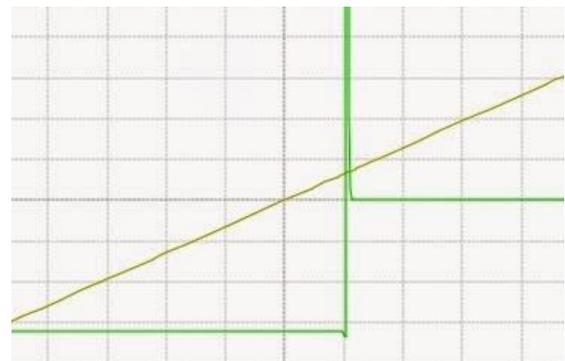


Imagem 5.1-5: Máximo do INL

Verifique na imagem que as outras duas variações ocorrem de forma precisa no intervalo de tempo desejado (destacado pelas setas na imagem 5.1-4), pois são geradas pela variação do bit menos significativo, que é o bit que está recebendo o sinal com maior taxa de variação, em relação ao tempo, em sua entrada, portanto, seus comparadores mudam de estado de forma proporcional. Na Imagem temos o degrau com maior duração e o degrau com menor duração, respectivamente, 557,3ns e 426,0ns, sendo o valor esperado para um degrau ideal de 500ns para a frequência de amostragem desejada no processo. Assim os erros são de 57,3ns e 74ns respectivamente, o que equivale a uma distorção de 11,46% e 14,8%, essas não chegam provocar perdas de valores, apenas de resolução.

O valor para determinar o INL foi prejudicado pelo Glitches, pois ocorrem nos extremos que determinam os valores para o INL. Assim, foi realizada o pegando o valor máximo do módulo da diferença entre a referência e a escada gerada pelo processo de conversão, mas considerando a alteração dos valores da imagem 5.1-3. O valor em questão está no ponto de inversão do segundo bit mais significativos sendo a transição do valor lógico do CMD de "1001010101" para "1101010101", que em termos binários seriam os valores de "1011111111" para "1100000000", ou, em decimal, de 767

para 768. A distorção chegou à 19,1% em relação ao valor de interesse, em termos de corrente seriam 933  $\mu$ A.

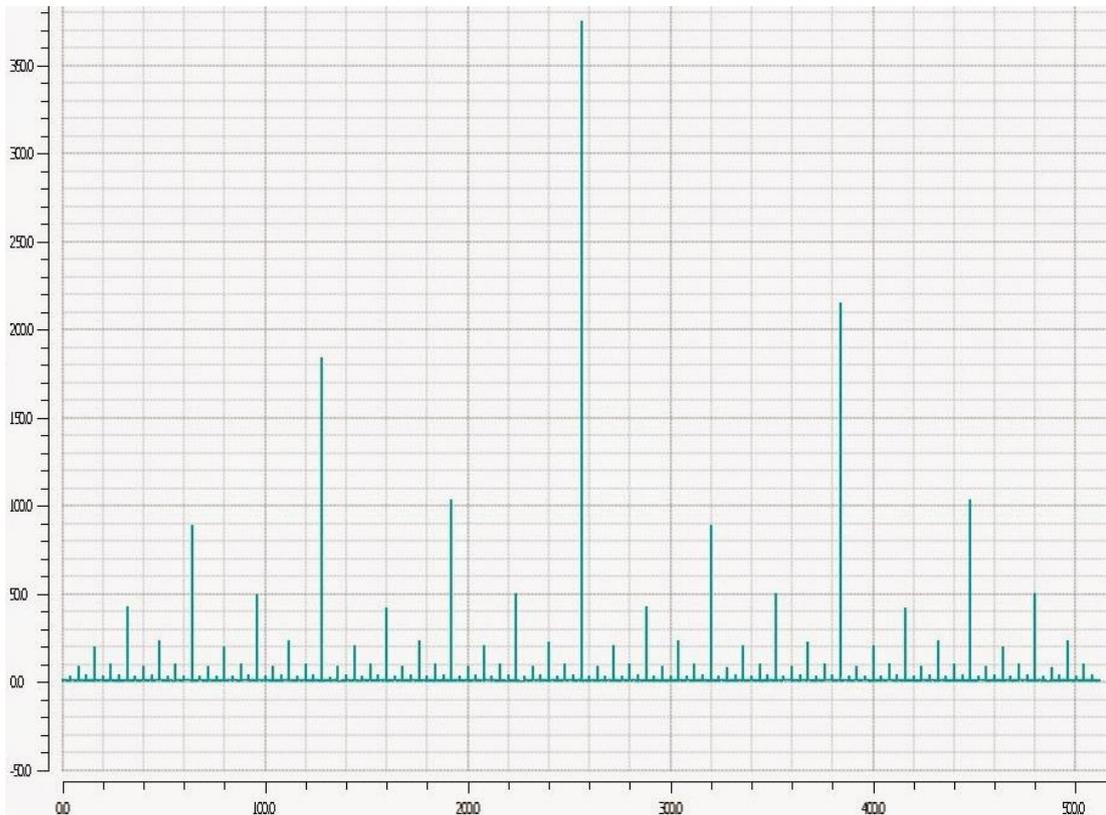


Imagem 5.1-6: Quadrado da Diferença do Sinal Analógico e do Digital.

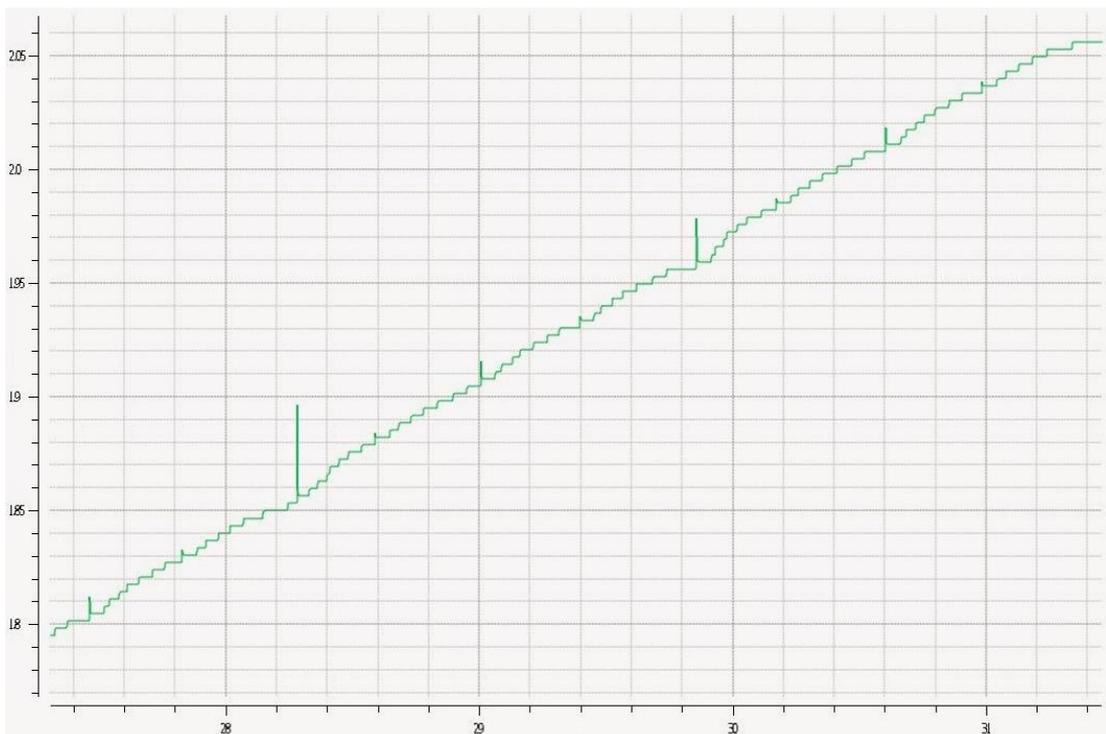


Imagem 5.1-7: Quadrado da Diferença do Sinal Analógico e do Digital.

Para o ENOB foi realizado o cálculo da potência do ruído diretamente da subtração do sinal pela referência, seguido da subtração pela média e finalmente a obtenção do RMS do sinal final. Nesse caso os Glitches contribuíram para o prejuízo do valor final. A curva em questão foi a da imagem 5.1-6. Veja que os Glitch tomam grande destaque na imagem, que é basicamente do sinal menos o sinal de entrada para as ondas da imagem 5.1-2, sendo tomados apenas os valores da descida da rampa, que apresentaram pior comportamento.

Os sinais foram transformados em valores de tensão para determinar a diferença entre o sinal de ainda do DAC e o padrão ideal. O valor para o RMS do ruído foi de 993uV e o RMS para o sinal é de 1,1667V. Assim a relação sinal ruído obtida é de 9,9 Bits e com o aumento da frequência de amostragem para 20MS/s o sinal já cai para 8,76 bits:

$$\text{ENOB} = \frac{20 \times \log\left(\frac{1166700}{993}\right) - 10 \times \log\left|\frac{3}{2}\right|}{20 * \log_{10} 2} \quad (5.1-1)$$

$$\text{ENOB} = \frac{61,3 - 1,761}{10 * \log_{10} 2} = \frac{59,61}{6,02} = 9,9 \quad (5.1-2)$$

Pela particularidade de não utilizarmos o S/H para amostrar o sinal, é possível avaliar a queda da resolução pelo aumento da frequência do sinal de interesse. Em nosso caso, a taxa de amostragem em 20MS/s representou uma queda de quase metade do valor, pois a relação entre o sinal e o ruído passou a ser 530,6:

$$\text{ENOB} = \frac{20 \times \log(530,6) - 10 \times \log\left|\frac{3}{2}\right|}{20 * \log_{10} 2} \quad (5.1-1)$$

$$\text{ENOB} = \frac{54,5 - 1,761}{10 * \log_{10} 2} = \frac{52,74}{6,02} = 8,76 \quad (5.1-2)$$

$$\text{FME} = \frac{633,6 \times 10^{-6}}{2 \times 10^6 \times 2^{9,9}} = 332fJ \quad (5.1-3)$$

$$\text{FME} = \frac{1,36 \times 10^{-3}}{20 \times 10^6 \times 2^{8,76}} = 157fJ \quad (5.1-4)$$

O consumo apresentado no primeiro caso foi de 192uA para 3.3V, correspondendo à 633,6uW e, para a taxa de 20MS/s foram 411 uA, ou seja, 1,36 mW. Sendo os parâmetros aplicados em 5.1-3 e 5.1-4. Assim, observamos que, apesar do aumento de consumo e perda de resolução, o processo se torna mais eficiente em termos energéticos com o aumento de frequência, o que mostra a possibilidade de aumentar o consumo para melhorar a resolução.

## 5.2. ULTRA LOW POWER

Para esse projeto resolvemos explorar pequenas variações que possam permitir um resumo do consumo final do conversor. O projeto possui uma quantidade maior de bits, pois são 14 bits de projeto para um sistema que possa ser utilizado em circuito de marca passo, por exemplo. A especificação é um conversor de 12 bits, 500nW e 10 Hz.

### 5.2.1. TECNOLOGIA UTILIZADA

A tecnologia para o projeto foi a XH 0.18 da XFAB, que é 180nm com fonte de alimentação de 1.62V à 1,98V; e de 350nm na faixa de 2,7V à 3,6V.

### 5.2.2. TOPOLOGIA GERAL DO CONVERSOR

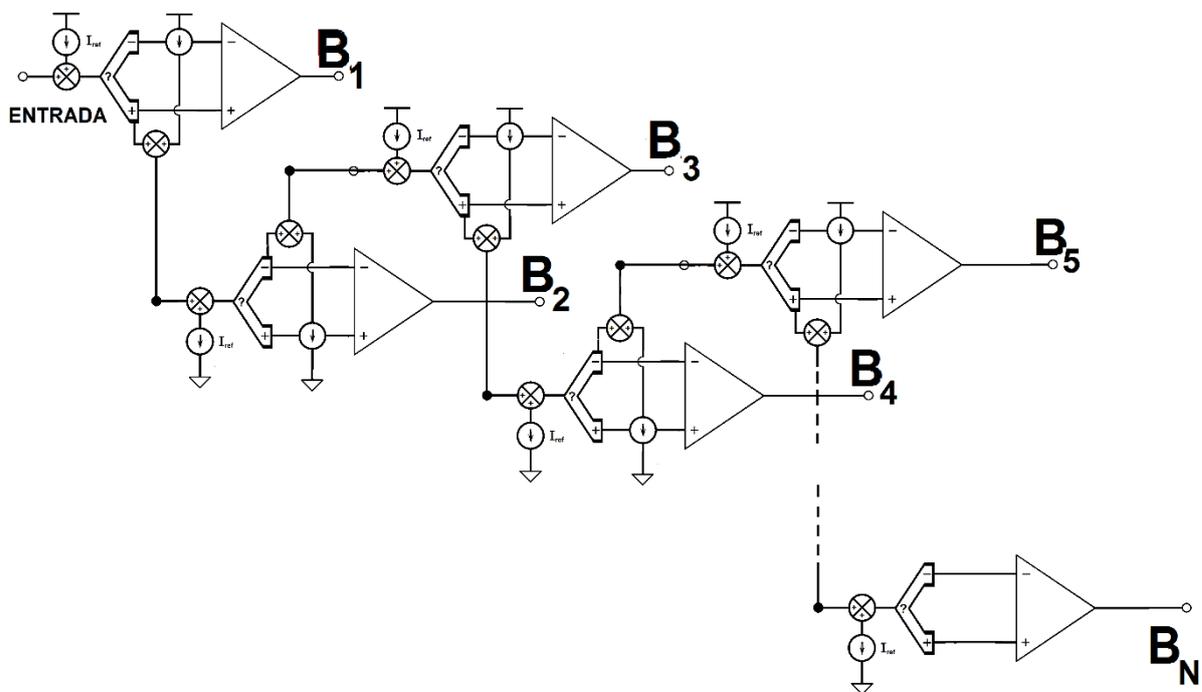


Imagem 5.2.2-1 Topologia Geral do Circuito Low Power.

A topologia reaproveita a corrente utilizada pelos blocos. Como vimos anteriormente, há um espelho de corrente que serve de fonte de corrente nos blocos somadores. Sendo assim, há, obrigatoriamente, no mínimo o consumo de corrente dessa fonte para cada bloco. E, essa é a corrente necessária também para a operação de outro bloco. Dessa forma, foram desenvolvidos circuitos que conseguem operar em metade da faixa de VDDA (tensão de alimentação do circuito analógico). Os circuitos que operam do VDDA para metade do valor do VDDA são prioritariamente em PMOS e os circuitos que operam na metade inferior são prioritariamente elaborados em NMOS. Os circuitos prioritariamente em PMOS fornecem a corrente para os circuitos prioritariamente NMOS que por sua vez, estarão sempre puxando corrente dos circuitos PMOS. Assim, além de economizar com a

reutilização da corrente, o sistema também se auto equilibra, pois tem a tendência de manter a tensão entre os dois conjuntos na metade do valor de VDDA. Entretanto, há um circuito de estabilização do nível de tensão com o intento do circuito retirar ou fornecer corrente para manter o valor central constante.

### 5.2.3. RETIRADA DA MULTIPLICAÇÃO POR 2 NA SAÍDA

Para reduzir o consumo o processo de multiplicação da saída por 2 foi eliminado e passamos a dividir a referência de corrente do bloco somador por dois, o que resulta no mesmo efeito. Essa propriedade também simplifica o processo da seção anterior, pois quer dizer que o bit mais significativo sempre gastará mais corrente que qualquer um dos menos significativos, pois ele no mínimo gasta a corrente máxima do menos significativo adjacente a ele. Assim, uma vez que o bit mais significativo começa no circuito PMOS, já garantimos que os circuitos PMOS sempre fornecerão corrente adequada aos circuitos baseados em NMOS. Finalmente, o circuito responsável pelo equilíbrio da tensão retirará a corrente fornecida em excesso pelos circuitos baseados em PMOS.

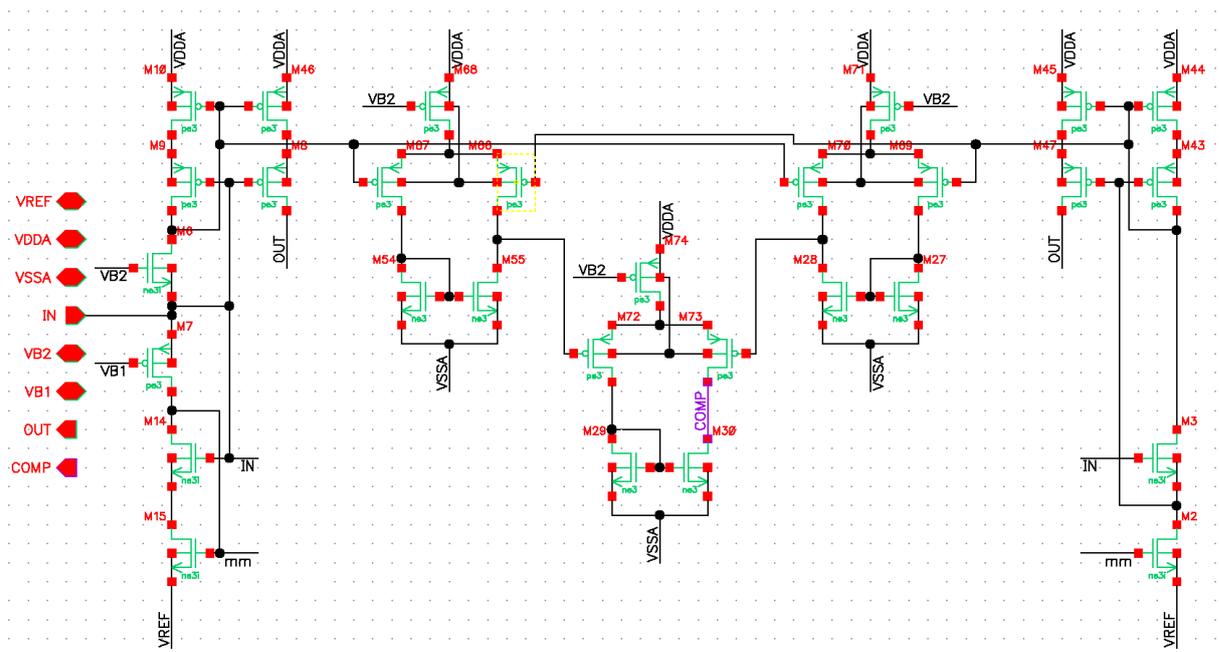


Imagem 5.2.3-1 Topologia do Circuito PMOS

Veja que no circuito PMOS as saídas dos blocos de Espelho vão para valores de VREF, no lugar de ser conectado ao VSSA (nível de referência para a tensão VDDA, seria o negativo da fonte de alimentação), como é o caso dos comparadores. Assim, a corrente consumida por esses blocos de espelho serve para realimentar os circuitos NMOS. Foram adicionados comparadores com corrente na faixa de 10nA para polarização, pois, devido à estrutura do sistema, não precisam de velocidade e nem fornece grande quantidade de corrente. O comparador final receberá o sinal já ampliado pelos dois primeiros, havendo a rápida inversão de qualquer forma.

#### 5.2.4. CARACTERIZAÇÃO DO PROJETO

O projeto teve sua especificação de consumo limitada a valores de 500nW. O consumo ficou em 468nW sem a conclusão de um PVT. Com número de 14 bits de projeto para atender uma especificação de 10 bits. O resultado pode ser visto através da FFT realizada para caracterizar a relação sinal ruído para o conversor como um todo. Pela FFT podemos avaliar a resolução obtida para diferentes frequências de operação. O Ruído para as frequências de 10Hz, 15Hz e 25Hz são de  $1 \times 10^{-4}$ ,  $1,82 \times 10^{-4}$  e  $2,155 \times 10^{-4}$ . O sinal de interesse aparece com potência de 0,5122. Assim temos os valores para ENOB.

$$\mathbf{ENOB} = \frac{74,19 - 1,761}{10 * \log_{10} 2} = \frac{72,43}{6,02} = 12,03 \quad (5.2.4-1)$$

$$\mathbf{ENOB} = \frac{68,99 - 1,761}{10 * \log_{10} 2} = \frac{67,23}{6,02} = 11,17 \quad (5.2.4-2)$$

$$\mathbf{ENOB} = \frac{67,52 - 1,761}{10 * \log_{10} 2} = \frac{65,76}{6,02} = 10,92 \quad (5.2.4-3)$$

$$\mathbf{FME} = \frac{468 \times 10^{-9}}{5 \times 2^{12,03}} = 22,38pJ \quad (5.2.4-4)$$

$$\mathbf{FME} = \frac{468 \times 10^{-9}}{5 \times 2^{11,17}} = 40,52pJ \quad (5.2.4-5)$$

$$\mathbf{FME} = \frac{468 \times 10^{-9}}{5 \times 2^{10,92}} = 48,31pJ \quad (5.2.4-6)$$

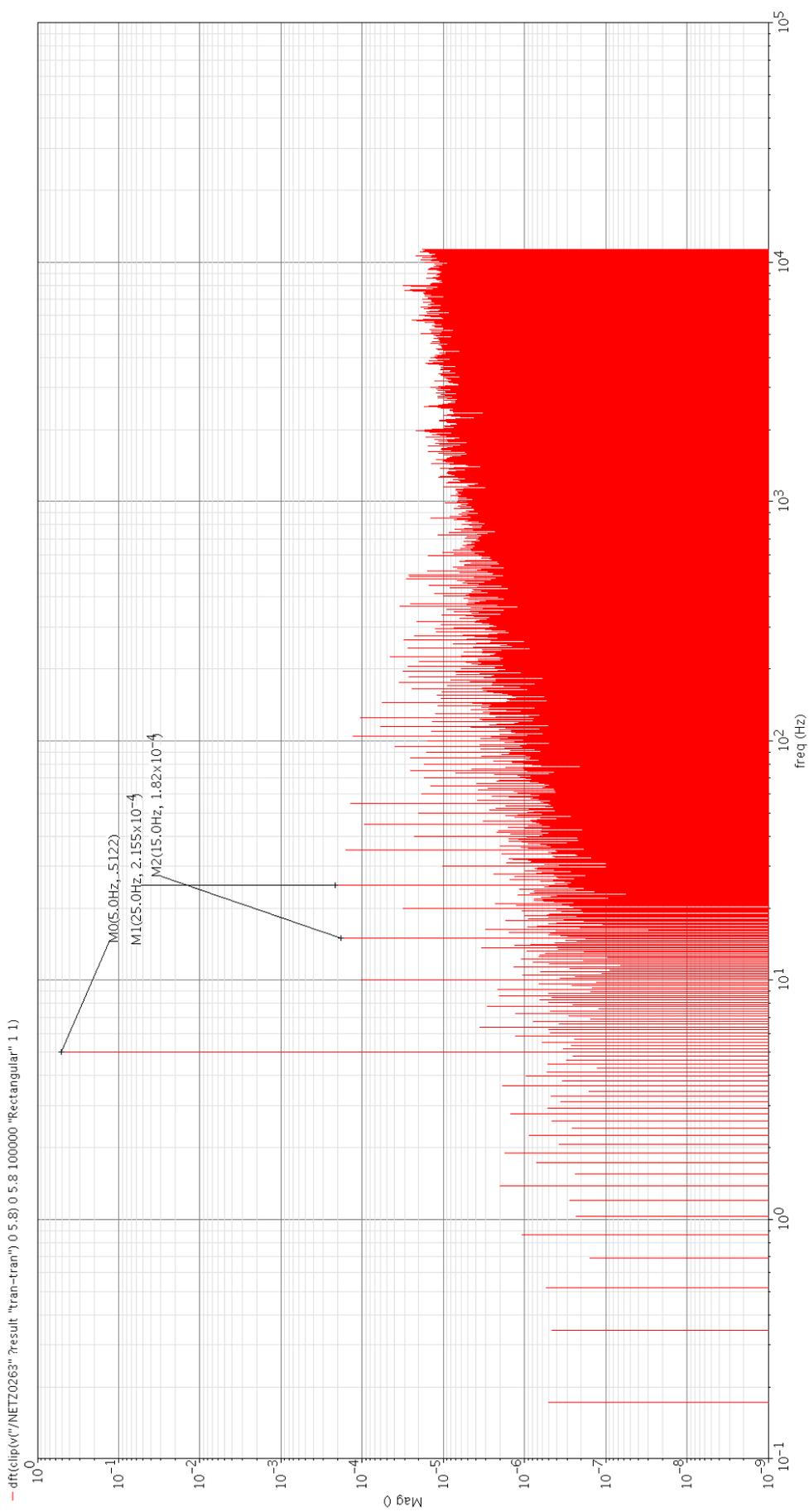


Imagem 5.2.4-1 FFT do Sinal do Conversor Recebendo o Sinal de Interesse em 5 Hz

## **5.3. CONSOLIDAÇÃO DA NOVA ARQUITETURA**

Esse capítulo objetiva apresentar as características observadas da nova arquitetura de conversor de sinais analógicos para sinais digitais, O CMD. Inicialmente apresentaremos os motivos de considerada uma *nova arquitetura* de conversores de sinais analógicos para sinais digitais, demonstrando que realmente se trata de um novo processo para determinação de valores digitais a partir de uma entrada analógica de interesse.

### **5.3.1. CARACTERÍSTICAS PRINCIPAIS**

O inovador é o processo de obtenção do valor digital. Para entendermos melhor, consideraremos, resumidamente, os processos dos conversores SAR, PARALELO, PIPELINE e SIGMA DELTA.

#### **5.3.1.1. O SAR**

No primeiro caso, o do SAR, verificamos se o sinal é maior ou se é menor que determinada referência e, apenas com essa questão solucionada, determinamos em qual faixa será feita uma nova comparação e assim por diante. A cada comparação é determinado o valor do bit, ou seja, 1 quando é maior que a referência e, 0 quando é menor que a referência.

Assim, o SAR precisa da informação do valor do bit para determinar qual será o procedimento para obter o próximo bit. De forma reversa, o bit que está sendo definido, caso não seja o primeiro bit, depende do valor do bit anterior para ser definido.

#### **5.3.1.2. O FLASH (PARALELO)**

No FLASH as faixas são pré-definidas pelas referências que dividem a faixa analógica de acordo com a resolução desejada. E, para cada faixa, é vinculado um valor digital correspondente. Assim, as faixas analógicas estão entre duas referências e o sinal é comparado, simultaneamente, com todas as referências. O valor digital é caracterizado avaliando o último comparador que está com resultado lógico 1 e o primeiro que está com sinal lógico 0, ou seja, respectivamente, o último que possui referência menor que a entrada e o primeiro que possui referência maior que a entrada. O sinal digital é obtido por um processo de decodificação direta.

#### **5.3.1.3. O PIPELINE**

No Pipeline temos mais um processo de conversão do que necessariamente uma arquitetura, pois a forma como o sinal analógico é obtido não interessa. Vejamos, o procedimento se dá com a determinação do valor digital, por um conversor de baixa resolução, normalmente Flash. O sinal digital

obtido é utilizado como entrada para um DAC, que gera um sinal para ser subtraído da entrada. E, posteriormente, amplificar o resíduo dessa subtração para realizar uma nova conversão. Essa nova conversão é feita por outro conversor e, portanto, o primeiro conversor já pode receber um novo valor para converter. A ideia do processo é utilizar-se de conversores rápidos e simples o suficiente para serem eficientes. Possui vários conversores em série para que cada um otimize a conversão, já que os conversores trabalham convertendo amostras diferentes do sinal, obtidas em intervalos sequenciais e distintos.

#### **5.3.1.4. O SIGMA DELTA**

O Sigma Delta parte da caracterização do sinal pela sua modulação da entrada em um sinal final saturado suficientemente para ser considerado digital, referindo-se ao trem de ZEROS e UNS, a modulação apresenta um valor perfeito e bem definido para cada bit, por isso, o sinal é processado já em domínio digital para caracterização de qual o valor analógico geraria o comportamento já armazenado.

O mais importante do Sigma Delta é a verificação de um novo conceito, mais que uma conversão de analógico para digital, pois nos processos anteriores a preocupação era a de representar o sinal analógico exatamente na forma digital a qual estamos acostumados a trabalhar. Entretanto, no caso do Sigma Delta, o valor analógico gera um padrão digital e estatístico. Desse padrão é determinada a faixa analógica, ou seja, o padrão possui informação suficiente para reconstruir o sinal. Apesar desse conceito ficar claro no caso do Sigma Delta, não deve-se esquecer que o processo seria similar no Flash, que é a primeira arquitetura a atingir aplicação comercial em circuitos eletromecânicos.

#### **5.3.1.5. O CMD**

Diferente do SAR, o CMD não utiliza o valor do bit mais significativo para determinar uma nova faixa para a próxima comparação. Diferente do FLASH, o CMD não faz o uso de valores de referência ao longo de toda a faixa analógica de interesse, dividindo a faixa de acordo com a resolução desejada. Diferente o PIPELINE, o CMD não realiza processos intermediários de conversão para ampliar a faixa analógica, não caracterizada, para realizar novas conversões até obter a resolução desejada. Diferente do Sigma Delta, o CMD não realiza modulação do sinal analógico em valores digitais para processamento futuro que determine as características do sinal originalmente analógico que provocou a modulação registrada.

Na primeira etapa do CMD, assemelha-se ao SAR, ou seja, comparando o sinal analógico de interesse com uma referência que divide a faixa analógica em duas faixas iguais. Mas, para não depender do resultado dessa comparação, passa para o processamento do valor de forma independente da primeira, assemelhando-se ao PIPELINE, o valor do módulo da diferença entre o sinal de interesse e a primeira referência com o valor do módulo em questão, realizada uma nova

comparação e gerado um novo valor de modulo inicia a próxima etapa até que seja atingida a resolução desejada.

Ao Sigma Delta, assemelha-se o processo do CMD quando não determina um valor digital dentro dos conformes do valor binário utilizado em cálculos matemáticos, o registrado é o efeito que o sinal analógico de interesse proporciona no processo de cada um. A grande questão é termos armazenado informação o suficiente para conseguirmos recuperar a faixa à qual o valor analógico pertencia.

Observe que o SAR determinava a faixa a partir do valor convertido, mas no CMD isso é ignorado por utilizar o valor do módulo. Note ainda que não é necessário utilizar o bit já determinado, pois ele informa à qual faixa o valor analógico pertence, por isso, essa informação não precisa ser utilizada na caracterização do próximo bit, pois seria uma redundância no processo.

Dessa forma, o CMD comprova ser uma nova lógica de conversão de valores analógicos em valores digitais.

A seguir, apresentaremos as características principais desse processo.

### 5.3.1.5.1. SIMETRIA

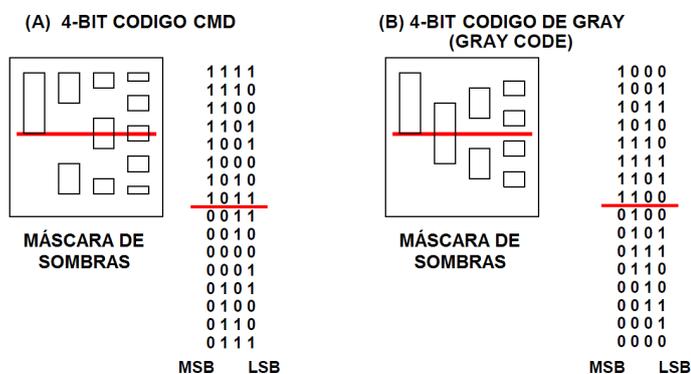


Imagem 5.3.1.5.1-1: Código Binário e Código de Gray (Imagem 4.4.2-4).

Vimos nas Seções anteriores dos capítulos anteriores, que assim como o código de Gray, o código gerado pelo CMD é simétrico. E que essa simetria permitia determinar o comportamento que teriam os novos bits que viessem a ser adicionados ao conversor e, portanto, poderíamos prever o comportamento do conversor, que é de grande valia no processo de caracterização e projeto.

### 5.3.1.5.2. CORRESPONDÊNCIA BIUNÍVOCA

*“A ideia criada pelo homem para lidar com as variações quantitativas foi denominada correspondência biunívoca. A correspondência biunívoca é a primeira criação matemática, e uma das mais importantes. A essência*

dessa operação é a comparação e equiparação entre dois conjuntos: um conjunto que conta - tomado como padrão - e um conjunto que é contado - do qual se quer controlar a variação quantitativa. No caso dos pastores da Antiguidade, sabemos que relacionavam um conjunto de pedras (conjunto que conta) com o conjunto de ovelhas (conjunto contado). Portanto a correspondência biunívoca exige a tomada de um conjunto como "padrão" - o conjunto que conta. É ele que guardará o registro da quantidade do conjunto que se quer contar". (Roberto P. Moisés).

O comportamento apresentado pelos bits para cada faixa analógica é único. E, da mesma forma, cada faixa analógica gera um único padrão de bits, o que permite reconhecer, apenas pelo formato binário registrado, a faixa analógica que gerou aquele determinado comportamento.

### 5.3.1.5.3. CONTINUIDADE DOS SINAIS DE CADA BIT

O sinal que gera cada bit se comporta de forma contínua na transição de valores, ou seja, de uma para outra faixa digital, os sinais gerados pelo sinal analógico não sofrem mudança abrupta, como ocorre com os sinais do PIPELINE e do SAR, pois eles utilizam os valores já determinados para realizar subtrações, mas esses valores geram sinais descontínuos, uma vez que eles já são sinais digitais. Recordemos as imagens que ilustram essa característica.

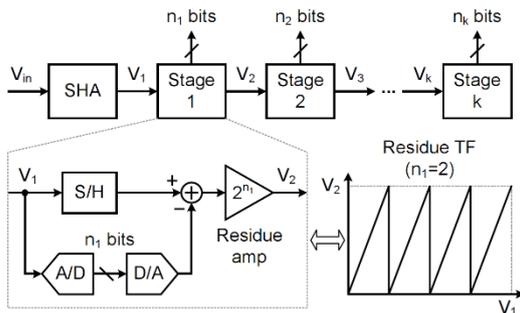


Imagem 5.3.1.5.3-1: Descontinuidade do PIPELINE (Imagem 2.1.3.1)

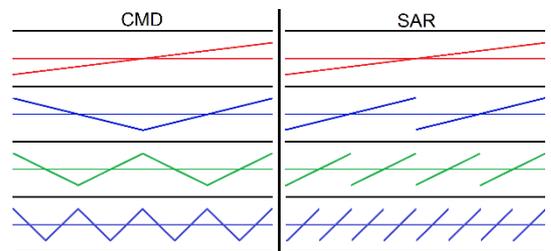


Imagem 5.3.1.5.3-1: Comportamento do CMD e do SAR (Imagem 4.4.2-1)

### 5.3.1.5.4. CONVERTER O PADRÃO DO CMD NO PADRÃO MATEMÁTICO

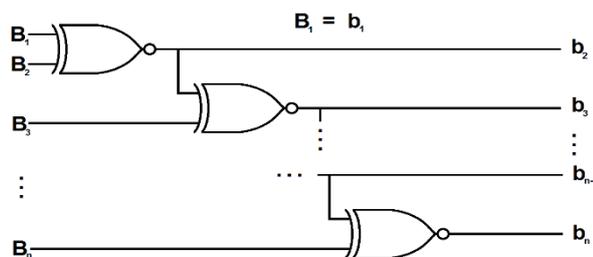


Imagem 5.3.1.5.4-1: Tradutor (Imagem 4.4.3-5)

Surpreendentemente o circuito para traduzir o comportamento binário gerado pelo CMD é realizado com apenas uma porta NXOR a cada bit adicionado, o que deixa o processo ainda mais eficiente e simples para o projeto de um conversor CMD.

#### 5.3.1.1.1. TAXA DE AMOSTRAGEM

Por se tratar de um processo assíncrono, a taxa de amostragem é caracterizada por três parâmetros básicos:

- Taxa de variação do sinal de interesse;
- Atraso do comparador de um dos bits;
- Atraso de propagação do sinal analógico entre os bits.

O primeiro é o mais simples, pois o sistema é assíncrono, portanto, caso o sinal fosse contínuo o conversor não variaria o valor e ficaria em um estado de espera. Apresentando o mesmo valor na saída durante todo o tempo.

A segunda depende do tempo de resposta do comparador. Vimos que afeta principalmente o bit mais significativo, pois o sinal é multiplicado por dois a cada conversão e, por isso, o sinal de cada bit menos significativo possui maior taxa de variação, o que gera uma variação mais rápida para os comparadores, conseqüentemente, para um sinal com taxa de variação maior que o slew rate do comparador, o bit menos significativo variar até mesmo após um dos bits menos significativos, o que gera perda de resolução e, portanto, o conversor conseguirá atuar na taxa de compatível com o sinal, mas terá sua resolução reduzida.



Imagem 5.3-1: Bits menos significativos continuam alterando independente do mais significativo

A terceira é similar à segunda, mas atinge os bits menos significativos, o processo é inerente ao atraso dos espelhos de corrente e trata-se de um processo limitado pela tecnologia e por capacitâncias parasitas dos GATES dos componentes.

Apenas a segunda possui soluções simples. Uma das soluções seria aumentar o slew rate do comparador do bit mais significativos e outra seria deixar de multiplicar a saída por dois, passando a dividir a referência por dois a cada bit, como foi feito no projeto do conversor Ultra Low Power. Uma solução em domínio digital poderia seria um processo que avaliasse cada bit de forma discreta, aplicando correções estatísticas que possam aumentar a resolução final do conversor, como é feito com o sigma delta, mas ainda não foi elaborado nesse trabalho. Na imagem 5.3-1 temos o processo, note que o bit mais significativo só altera seu valor após quase 6 vezes o intervalo necessário, mas os bits menos significativos continuam convertendo e, devido à simetria, passam a repetir os valores digitais na sequencia inversa.

#### **5.4. CONSIDERAÇÕES**

O processo e as características do CMD são divergentes dos outros conversores, uma vez que possui características bem diferentes em muitos aspectos, o que permite o uso do mesmo de forma diferenciada em relação aos outros conversores. Isso não coloca um conversor como melhor ou pior que outro, mas demonstra que ele possui características superiores em determinadas etapas, conforme será detalhado no capítulo 8.

Esse é o primeiro trabalho que trata do CMD e prevemos trabalhos futuros a fim de caracterizar com maior complexidade o conversor e explorar as possibilidades que essa nova arquitetura tem a fornecer. Já nesse trabalho foram apresentados dois projetos do conversor, mas existem ainda, inúmeros outros pretendidos que serão projetos voltados à exploração da frequência de operação do conversor; projetos para explorar a resolução que a nova arquitetura pode atingir, dado o enorme potencial para atingir grandes resoluções sem realizar processos descontínuos que inserem ruídos não desejados; aplicação de comportamento não linear no processo de conversão (na topologia utilizada de espelhos, é fácil alterar os ganhos dos espelhos de determinar faixas de maior precisão do que outras, muitos sinais como o do ECG teriam vantagens com esse processo); casamento dos atrasos de cada bloco de módulo; projetos em modo tensão e outros que poderão surgir.



## 6. CONCLUSÃO

Uma nova arquitetura de conversor A/D foi apresentada neste trabalho o CONVERSOR DO MÓDULO DA DIFERENÇA (CMD), atendendo a tabela abaixo.

DESvantagens				
	SAR	$\Sigma\Delta$	FLASH	CMD
ÁREA EM SOC	POUCO AFETADOS	POUCO AFETADOS	MAIS AFETADOS	POUCO AFETADOS
CAPACITORES	MAIS AFETADOS	PARCIALMENTE AFETADOS	PARCIALMENTE AFETADOS	POUCO AFETADOS
CONSUMO	POUCO AFETADOS	PARCIALMENTE AFETADOS	MAIS AFETADOS	POUCO AFETADOS
INTERFACE ANALÓGICO DIGITAL	POUCO AFETADOS	POUCO AFETADOS	MAIS AFETADOS	POUCO AFETADOS
OPERAÇÃO	POUCO AFETADOS	MAIS AFETADOS	MAIS AFETADOS	POUCO AFETADOS
CICLOS	MAIS AFETADOS	POUCO AFETADOS	POUCO AFETADOS	POUCO AFETADOS
OVER SAMPLE	POUCO AFETADOS	MAIS AFETADOS	POUCO AFETADOS	POUCO AFETADOS
RESOLUÇÃO	PARCIALMENTE AFETADOS	POUCO AFETADOS	MAIS AFETADOS	POUCO AFETADOS
REFERÊNCIA	PARCIALMENTE AFETADOS	POUCO AFETADOS	MAIS AFETADOS	POUCO AFETADOS

Imagem 6-1: Tabela comparativa entre o SAR, o  $\Sigma\Delta$ , o FLASH e o CMD.

Na tabela da imagem acima temos 9 linhas. Na 1ª o CMD se tornou pouco afetado pela área em SOC por ser constituído de circuitos simples e não utilizar capacitores ou resistores, que já é a vantagem da 2ª linha da tabela. Na 3ª o CMD é pouco crítico por adicionar apenas um grupo de circuito a cada bit, ou seja, não há um circuito para cada palavra digital. Na 4ª a interface de analógico para digital é simples, pois é necessário apenas alterar os níveis de operação dos bits gerados, caso sejam gerados em níveis diferentes. Na 5ª a operação do conversor é desnecessária, pois ele processa de forma assíncrona, sendo necessário apenas tratar os sinais quando necessário. Na 6ª o conversor realiza o processo sem ciclos, ou seja, de forma direta. No, 7 o conversor é caracterizado com Nyquist, portanto não faz o uso de Over Sample. No, 8 estamos nos referindo à resolução das referências, que no caso do conversor trabalhamos com poucos valores e não precisamos trabalhar com uma precisão complexa para o processo, como no caso do Flash ou trabalhar com os diversos erros de capacitores e resistores como no caso do SAR. Em relação à 9 não precisamos gerar diversas referências ou um circuito que gere referências com grande precisão como no caso do Flash, quando utilizando um número elevado de bits.

Assim, o CMD em resumo é um conversor sem over sampling, sem diversas referências, sem capacitores ou resistores, sem um circuito para cada divisão e sem dependência de entre bits adjacentes. Realizando a conversão de todos os bits em paralelo, ou seja, sem vários ciclos para aproximações que dependam dos ciclos anteriores. Dessa forma o resultado foi um conversor que surgiu da ideia de eliminar as falhas do processo de conversão do SAR, mas ainda mantendo os seus melhores aspectos lógicos.

O projeto do conversor CMD possui um algoritmo muito similar ao do SAR. Entretanto, o CMD realiza a conversão de forma contínua com três características básicas:

- A.** Sempre subtrair da entrada uma referência fixa, que é metade do fundo de escala da entrada;
- B.** Obter o módulo do resultado do processo de subtração;
- C.** Comparar a entrada com a referência. Se maior o bit é “1” e “0” caso contrário.

As etapas A, B e C são realizadas apenas em processamento analógico. Conseguindo gerar os bits de forma independentes um dos outros.

O CMD gera todas as possibilidades de valores binários conforme o número de bits do conversor, ou seja, se é de 3 bits gera 8 palavras e sendo de 5 gera 32 palavras. Entretanto, a sequência das palavras não é a sequência dos valores dos números binários.

As particularidades da sequência mostraram simetria com singularidade ao código de Gray. E essa similaridade garantiu grande estabilidade para os valores convertidos pelo fato de apenas um bit ser alterado entre valores digitais adjacentes convertidos pelo CMD. Possibilitando que o CMD não utilize circuito de S/H.

Por não utilizar S/H o conversor não possui taxa de amostragem fixa e por isso sua taxa utilizada para caracterizar o conversor dependia da taxa de variação do sinal de entrada. Assim o processo de conversor se adapta às características do sinal interesse. Entretanto essa característica depende da taxa de variação do sinal, ou seja, dos componentes de frequência e amplitude do sinal de entrada. Assim, uma taxa fixa de amostragem seria melhor implementada através de um filtro que trate-se os valores após a conversão.

O circuito digital para o CMD também utilizou circuitos simples baseados em apenas uma porta lógica XNOR para cada bit a ser gerado, sendo que o processo digital realiza apenas uma tradução dos valores digitais já convertidos pelo CMD. Apesar de serem os primeiros projetos do CMD, vemos na imagem 6-2 no círculo com um “X” no centro que já está dentro do mapa dos conversores dentro do estado da arte de 2011. Assim, o conversor abre novas portas para o desenvolvimento de circuitos mais eficiente e rápidos para realizar a conversão de analógico para digital.

Para trabalhos futuros pretendemos desenvolver quatro modelos, três que superem o estado da arte e um que melhores o CMD. Um com taxa de amostragem em 100GS/s para colocar o CMD dentro do grupo de conversores com as maiores taxa de amostragem. Segundo modelo de grande resolução para realizar uma comparação direta com os conversores Sigma Delta de 22bits. Terceiro um modelo com consumo inferior à 5 fJ por conversão. E um modelo com compensação do atraso de cópia entre os espelhos responsáveis pelo processamento analógico do sinal. O último é um desenvolvimento voltado apenas para a melhoria do CMD.



## REFERÊNCIAS

- [1] LA ROSA, José M. Sigma-Delta Modulators: Tutorial Overview, Design Guide, and State-of-the-Art Survey. **Circuits And Systems I: Regular Papers, IEEE Transactions On, Sevilla, v. 58**, n. 1, p.1-21, Jan. 2011.
- [2] **KESTER, Walt. Analog-Digital Conversion. USA:** Analog Devices, Inc, 2004. 1138 f.
- [3] H. Nyquist, "Certain Topics in Telegraph Transmission Theory," A.I.E.E. Transactions, Vol. 47, April 1928, pp. 617-644.
- [4] Paul M. Rainey, "Facimile Telegraph System," U.S. Patent 1,608,527, filed July 20, 1921, issued November 30, 1926. (although A. H. Reeves is generally credited with the invention of PCM, this patent discloses an electro-mechanical PCM system complete with A/D and D/A converters. The patent was largely ignored and forgotten until many years after the various Reeves' patents were issued in 1939-1942).
- [5] DALLET, Dominique; SILVA, José Machado da. **Dynamic Characterisation of Analogue-to-Digital Converters.** [s. L.]: Kluwer Academic Publishers, 2005. 280 p. (The Springer International Series in Engineering and Computer Science).
- [6] JONSSON, Bengt E.. **A/D-converter performance evolution (1974 – 2012) Rev: 1.1.** [s. L.]: Converter Passion/ADMS Design AB, 2013. 51 p.
- [7] Swade, Doron. The Difference Engine: Charles Babbage and the Quest to Build the First Computer. [S.I.]: Penguin, 2000. p. 84–87. ISBN 0-1420-0144-9
- [8] Tanenbaum, Andrew. Modern Operating Systems. 3<sup>a</sup> ed. [S.I.]: Prentice Hall, 2007. 1104 p. p. 7. ISBN 0-13600663-9
- [9] Priestley, Mark (2011). A Science of Operations: Machines, Logic and the Invention of Programming. Springer. ISBN 978-1-84882-554-3.
- [10] Rojas, Raúl (Spring 2006). "The Zuse Computers". RESURRECTION The Bulletin of the Computer Conservation Society ISSN 0958-7403 (37)
- [11] <http://www.ci-brasil.gov.br/index.php/pt/o-ci-brasil/cenario>
- [12] Roebuck, Kevin, Software-defined radio (SDR): High-impact Technology - What You Need to Know: Definitions, Adoptions, EMEREO PTY LTD, 01-Jun-2011 ISBN 10: 1743044119/ISBN 13: 9781743044117
- [13] <http://www.ettus.com/>

- [14] <http://gnuradio.squarespace.com/>
- [15] [http://www.cadence.com/products/cic/schematic\\_editor/pages/default.aspx](http://www.cadence.com/products/cic/schematic_editor/pages/default.aspx)
- [16] [http://www.cadence.com/products/di/edi\\_system/pages/default.aspx](http://www.cadence.com/products/di/edi_system/pages/default.aspx)
- [17] CUBAS, Heiner Grover Alarcón. **Projeto de um Modulador Sigma-Delta de Baixo Consumo para Sinais de Audio**. 2013. 167 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Universidade de São Paulo, São Carlos, 2013.
- [18] BAJDECHI, Ovidiu; HUIJSING, Johan. **Systematic Design of Sigma-Delta Analog-to-Digital Converters**. [S. L.]: Springer Us, 2003. 196 p. (The Springer International Series in Engineering and Computer Science).
- [19] CARUSONE, Tony Chan; JOHNS, David; MARTIN, Kenneth. **Analog Integrated Circuit Design**. [s. L.]: Wiley E-text, 2011. 706 p.
- [20] NDJOUNTCHE, Tertulien. **CMOS Analog Integrated Circuits: High-Speed and Power-Efficient Design**. [s. L.]: CRC Press, 2011. 926 p.
- [21] VASCONCELLOS, Rodrigo D.. **Projeto de um Conversor Analógico/Digital por Aproximações Sucessivas de 12 bits**. Departamento de Engenharia Elétrica, Programa de Pós-Graduação em Engenharia Elétrica, UFMG, dezembro 2011.
- [22] CHIU, Yun. **High-Performance Pipeline A/D Converter Design in Deep-Submicron CMOS**. 2004. 88 f. Dissertation – Engineering – Electrical Engineering And Computer Sciences, University Of California, Berkeley, 2004.
- [23] DEMLER, Michael J.. **High-Speed Analog-to-Digital Conversion**. [s. L.]: Academic Press, Inc., 1991. 182 p.
- [24] CAO, Zhiheng; YAN, Shouli. **Low-Power High-Speed ADCs for Nanometer CMOS Integration**. Austin, TX: Springer Netherlands, 2008. 95 p. (Analog Circuits and Signal Processing Series).
- [25] HAN, Sang-Keun; PARK, KeeChan; KONG, Bai-Sun; JUN, Young-Hyun. High-speed low-power bootstrapped level converter for dual supply systems. **Circuits and Systems (APCCAS), 2010 IEEE Asia Pacific Conference on**, vol., no., p. 871-874, 6-9 Dec. 2010.
- [26] YU, Chien-cheng; WANG, Wei-ping; LIU, Bin-da. **A New Level Converter for Low-Power Applications**. Conference: Circuits And Systems, 2001. Iscas 2001. The 2001 IEEE International Symposium On, Volume: 1, v. 1, n. 1, p.113-116, Maio 2001.

- [27] GAMBINI, S.; RABAEY, J.. Low-Power Successive Approximation Converter With 0.5 V Supply in 90 nm CMOS. **Solid-State Circuits, IEEE Journal of**, vol.42, no.11, p. 2348-2356, Nov. 2007.
- [28] Haddad, Sandro A. P, Serdijn, Wouter A, **Ultra Low-Power Biomedical Signal Processing - An Analog Wavelet Filter Approach for Pacemakers** Springer Science + Business Media B.V. 2009
- [29] CHRISOSTOMO., L. A. P.(2014). **Front-end para aparelhos auditivos analógicos utilizando Transformada Wavelet**. Dissertação de Mestrado em Engenharia Elétrica, Publicação PPGEA – 563/14 Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 111p.
- [30] MALLAT, Stephane; PEYRÉ, Gabriel. **Preface to the Sparse Edition**, p. xv. **Notations**, p. xix. In: **A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way**. Burlington, Ma: Academic Press, 2009. 700 p.
- [31] HUANG, Qiuting. Low voltage and low power aspects of data converter design. **Solid-State Circuits Conference, 2004. ESSCIRC 2004. Proceeding of the 30th European**, Zurich, vol., no., p. 29-35, 21-23 Sept. 2004
- [32] BOURDI, Taoufik; KALE, Izzet. **CMOS Single Chip Fast Frequency Hopping Synthesizers For Wireless Multi-Gigahertz Applications: Design Methodology, Analysis, and Implementation**. Dordrecht: Springer Netherlands, 2007. 208 p. (Analog Circuits and Signal Processing Series).
- [33] SILVA, Paulo G. R.; REIS FILHO, Carlos A. **Estudo dos Limites de Performance dos Moduladores SD Implementados com Circuitos e Capacitores Chaveados**. 2001. 227 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Universidade Estadual de Campinas, Campinas, SP, 2001.
- [34] MALCOVATI, P.; BRIGATI, S.; FRANCESCONI, Fabrizio; MALOBERTI, F.; CUSINATO, P.; Baschiroto, A. Behavioral modeling of switched-capacitor sigma-delta modulators. **Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on**, Pavia, vol. 50, no.3, p. 352-364, Mar 2003.
- [35] CHERRY, James A.; SNELGROVE, W. Martin. **Continuous-Time Delta-Sigma Modulators for High-Speed A/D Conversion: Theory, Practice and Fundamental Performance Limits**. Norwell: Kluwer Academic Publishers, 2000. 248 p.
- [36] VINK, J.; VAN RENS, J.. A CMOS multi-bit sigma-delta modulator for video applications. **Solid-State Circuits Conference, 1998. ESSCIRC '98. Proceedings of the 24th European**, Eindhoven, vol., no., p. 164-167, 22-24 Sept. 1998.

- [37] ALLEN, Phillip E.; HOLBERG, Douglas R.. **CMOS Analog Circuit Design**. [s. L.]: Oxford University Press, 2002. (The Oxford Series in Electrical and Computer Engineering).
- [38] RAZAVI, Behzad. **Design of Analog CMOS Integrated Circuits**. CA, USA: McGraw Hill, 2001. 676 p.
- [39] LIANG, Shang-Quan; YIN, Yong-Sheng; DENG, Hong-hui; SONG, Yu-kun; GAO, Ming-lun. A low power consumption, high speed Op-amp for a 10-bit 100MSPS parallel pipeline ADC. **Circuits and Systems, 2008. APCCAS 2008. IEEE Asia Pacific Conference on**, Hefei, vol., no., p. 818-821, Nov. 30 2008-Dec. 3 2008.
- [40] Ronald J.Tocci, Neal S.Widmer, Gregory L. Moss, **Sistemas Digitais - Princípios e Aplicações**, 8ªed PRENTICE HALL BRASIL
- [41] Texas Instruments, **Understanding Data Converters**, Texas Instruments Application Report SLAA013, Mixed-Signal Products, 1995.



## APÊNDICE I BY-PASS

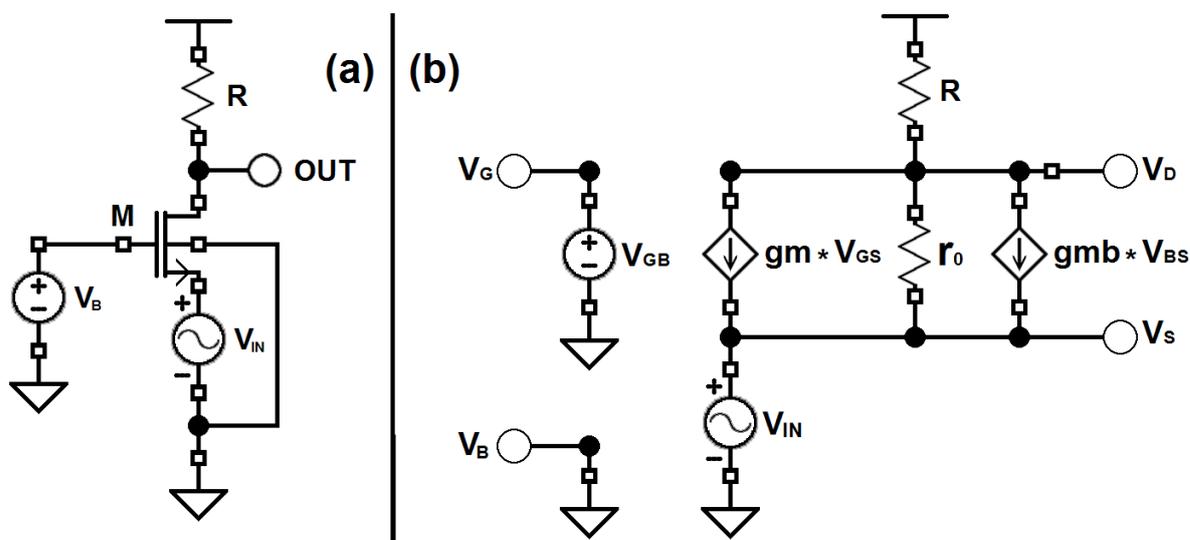


Imagem AI-1: COMUM GATE

É uma das topologias consagradas mais importante para esse trabalho. Tem como característica a tensão fixa no GATE e o sinal de entrada aplicado no SOURCE. Possui baixa impedância de entrada e ganho moderado. Geralmente as tecnologias possuem o substrato dopado com determinada carga, normalmente o substrato é dopado com portadores carentes de elétrons. Dessa forma, isolar o corpo do dispositivo do substrato custa área e, por isso, é evitado. Então, um dos problemas na configuração é a ocorrência do efeito de corpo pela dissociação do potencial do SOURCE e do BULK.

$$V_{DD} = V_{IN} + V_{DS} + R \times \left[ (gm \times V_{GS}) + (gm_B \times V_{BS}) + \frac{V_{DS}}{r_0} \right] \quad (AI-1)$$

$$V_{DD} - V_{IN} - R \times [(gm \times V_{GS}) + (gm_B \times V_{BS})] = V_{DS} \left( 1 + \frac{R}{r_0} \right) \quad (AI-2)$$

Diferenciando “AI-2” e observando as relações diferenciais “ $\partial V_{GS} = \partial V_{BS} = -\partial V_{IN}$ ”.

$$\frac{\partial V_{DS}}{\partial V_{IN}} = \frac{-r_0}{r_0 + R} + \frac{r_0 \times R \times (gm + gm_B)}{r_0 + R} \quad (AI-3)$$

Na equação: “R” representa a impedância do circuito de carga do COMOM GATE; “ $r_0$ ” representa a impedância do MOSFET; “ $gm_B$ ” é a transcondutância derivada da junção TBJ; e “ $gm$ ” é a transcondutância do MOSFET. Pela equação, o efeito de corpo fica favorável à transcondutância do MOSFET, entretanto, devesse observar que ao aumentar “ $V_{IN}$ ” o valor de  $V_{gs}$  reduz e o valor de  $V_{bs}$  aumenta. Para isso e para atender ao projeto, as equações serão tomadas em modo corrente, pois o projeto utiliza a configuração para o modo corrente.

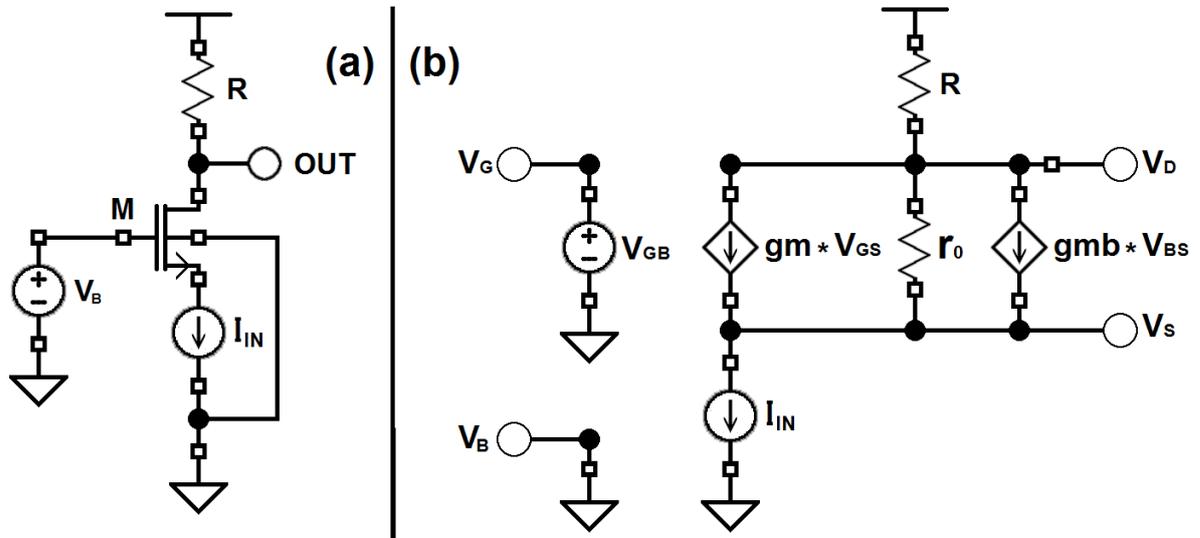


Imagem AI-1: COMUM GATE MODO CORRENTE

Do efeito de corpo temos a equação **AI-8** da seção anterior. E aplicando a primeira Lei de Kirchhoff:

$$\sum_{n=0}^N i_n = 0 \quad (\text{AI-4})$$

Na equação “**AI-4**”, “**N**” é o número total de diferentes componentes de correntes que derivam do nó ao qual é aplicada a “Primeira Lei”; “ $i_n$ ” é a  $n$ -ésima corrente elétrica que entra, quando positiva, ou sai, quando negativa, no nó. Portanto, uma soma de corrente em um circuito elétrico é a simples injeção e/ou retirada de corrente de um nó, que será toda a extensão de um metal, por exemplo, que conecta diversos dispositivos entre si. Inicialmente será considerado que a corrente “ $I_{IN}$ ” será retirada do SOURCE, portanto:

$$I_{IN} = gm \times V_{GS} + \frac{V_{DS}}{r_0} - \eta \times gm \times V_{BS} \quad (\text{AI-5})$$

$$V_{GS} = V_G - V_{DD} + R \times I_{IN} + V_{DS} \quad (\text{AI-6})$$

$$V_{GS} - V_{BS} = V_G \quad (\text{AI-7})$$

Das três equações anteriores, pode-se verificar que:

$$I_{IN} \times [1 + (\eta - 1) \times R \times gm] = \left[ (1 - \eta) \times gm + \frac{1}{r_0} \right] \times V_{DS} + gm \times [V_G - (1 - \eta) \times V_{DD}] \quad (\text{AI-8})$$

$$V_{GS} \times [1 + r_0 \times (\eta + 1) \times gm] = I_{IN} \times [r_0 - R] + [V_{DD} + V_G \times (r_0 \times \eta \times gm - 1)] \quad (\text{AI-9})$$

Portanto, temos as relações diferenciais:

$$\frac{\partial V_{DS}}{\partial I_{IN}} = r_0 \times \frac{[1 + (\eta - 1) \times R \times gm]}{[(1 - \eta) \times r_0 \times gm + 1]} \quad \& \quad \frac{\partial V_{GS}}{\partial I_{IN}} = \frac{\partial V_{BS}}{\partial I_{IN}} = \frac{[r_0 - R]}{[(\eta + 1) \times r_0 \times gm + 1]} \quad (\text{AI-10})$$

De forma similar, considerando a corrente como sendo injetada ao SOURCE:

$$\frac{\partial V_{DS}}{\partial I_{IN}} = -r_0 \times \frac{[1 + (\eta - 1) \times R \times gm]}{[(1 - \eta) \times r_0 \times gm + 1]} \quad \& \quad \frac{\partial V_{GS}}{\partial I_{IN}} = \frac{\partial V_{BS}}{\partial I_{IN}} = \frac{-[r_0 - R]}{[(1 - \eta) \times r_0 \times gm + 1]} \quad (\text{AI-11})$$

Das equações verificamos a condição de corte que será aplicada nos capítulos posteriores. O resultado da diferencial das diferenciais da equação **AI-10** apresentarem a mesma taxa de variação relativa à corrente de entrada para Vgs e para Vbs não significa que os dois estão aumentando em relação à variação de corrente, pois observemos que a relação foi obtida considerando o valor de Vbs menor que zero, pois Vb está conectado entre a fonte e o SOURCE, portanto, Vbs < 0. Assim, o que Vbs aumenta Vgs aumenta, mas se Vbs aumenta, seu valor, em módulo, diminui, pois o valor tende a zero, doravante o valor de Vgs aumenta em módulo, portanto, reduz o efeito de corpo e aumenta o valor de Vgs. Essa condição só é realizada para " $r_0 > R$ ". Já no caso da equação **AI-11** essa condição nunca ocorre, ou seja, a tendência é de esmagar o valor de Vgs, em termo de módulo, pois o valor é positivo, e ampliar ainda mais o valor de Vgs, portanto, o circuito corta a injeção de corrente, como era esperado, pois essa possibilidade seria conseguir fluir a corrente do polo positivo de uma fonte ao próprio polo positivo da fonte.

É importante observar, portanto, que para valores elevados de R, o circuito permanecerá sem conduzir até que o valor de corrente seja suficiente para atingir um " $r_0 > R$ ". Assim, quando o circuito não está conduzindo, sua tendência é a de não conduzir, mas ao entrar em condição suficiente para condução, a tendência é a de conduzir e saturar o dispositivo por um processo realimentado, ou seja, o valor de Vbs tende a zero e o valor de Vbs aumenta em mesma proporção, mas a condução aumenta com o quadrado do valor de Vgs, conforme as equações **AI-4** e **AI-5**, ou seja, o efeito ocorre tanto com o dispositivo em trípode quanto para o dispositivo em saturação. O processo também empurra o dispositivo para a saturação.