MÓDULO ANALÓGICO PARA UM CONVERSOR A/D POR APROXIMAÇÕES SUCESSIVAS

IAGO SERGIO PINHEIRO PEREIRA

TRABALHO DE GRADUAÇÃO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

Universidade de Brasília
Faculdade de Tecnologia
Departamento de Engenharia Elétrica

MÓDULO ANALÓGICO PARA UM CONVERSOR A/D POR APROXIMAÇÕES SUCESSIVAS

Iago Sergio Pinheiro Pereira

Trabalho final de graduação submetido ao Departamento de Engenharia Elétrica da Faculdade de Tecnologia da Universidade de Brasília, como parte dos requisitos necessários para a obtenção do grau de Engenheiro Eletricista.

APROVADA POR:

Prof. José Camargo da Costa, D.Sc. (ENE-UnB)
(Orientador)

Prof. Daniel Chaves Café, D.Sc. (ENE-UnB)
(Examinador Interno)

Prof. Sandro Augusto Pavlik Haddad, Ph.D. (FGA-UnB)
(Examinador Interno)

Brasília/DF, Março de 2018.

AGRADECIMENTOS

Primeiramente, quero agradecer aos meus pais, minha irmã e minha esposa por todo o apoio que me deram durante meu caminho até aqui.

Quero agradecer também a todos os amigos que fiz durante a graduação, em especial, ao Gabriel de Melo, Paulo Egídio, Jeremy Paule e Guilherme Campbell por terem sido meus companheiros desde meu primeiro semestre. Agradeço também aos amigos da graduação Carlos Eduardo Salustiano e ao Pedro Tolentino por terem contribuído diretamente neste trabalho.

Agradeço ao professor e amigo José Camargo pela oportunidade de realizar este trabalho, pela orientação e por todos os conselhos. Agradeço ainda ao professor Daniel Café pela ajuda no LPCI.

RESUMO

Palavras-chave: Conversor analógico-digital, SAR ADC, single-ended, aproximações sucessivas, CMOS.

Este trabalho apresenta o desenvolvimento dos módulos analógicos de um conversor analógico-digital por aproximações sucessivas de 10 bits construído em tecnologia CMOS 0.18 μm. Neste trabalho foi implementado um modelo comportamental do conversor em Matlab e a partir do algoritmo definido, partiu-se para o desenvolvimento do circuito analógico do conversor. O conversor foi concebido a partir de uma divisão do circuito em blocos funcionais. Dentre esses blocos destacam-se os circuitos *sample and hold*, comparador e conversor digital-analógico. Além da implementação completa da parte analógica do conversor, a sua validação foi feita através de uma análise e validação de cada bloco em separado. A validação do sistema completo do conversor foi efetuada por simulação, sendo que todos os requisitos das especificações iniciais foram satisfeitos.

ABSTRACT

Keywords: Analog to Digital Converter, SAR ADC, single-ended, successive aproximations, CMOS.

This work presents the development of analog modules for a 10 bit successive approximation analog to digital converter, built in 0.18μm CMOS technology. In this work, a behavioral model of the converter was implemented in Matlab, and from the algorithm defined in the model, the converter analog circuit was developed. The converter was conceived from the division of the circuit in blocks. Among those blocks there are a sample and hold, a comparator and a digital to analog converter. Besides the complete implementation of the converter analog circuit, its validation was done by analysis and validation of each block separately. The validation of the entire converter was also realized by simulation, and all the initial specifications were satisfied.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE SÍMBOLOS

Vin	Sinal de entrada do bloco.
$Vout$	Sinal de saída do bloco.
V_{GS}	Tensão entre porta e fonte do transistor.
V_{SG}	Tensão entre fonte e porta do transistor.
V_{DS}	Tensão entre dreno e fonte do transistor.
f_{S}	Frequência de amostragem.
K'	Parâmetro de Transcondutância.
λ	Parâmetro de modulação de canal.
W	Largura do canal do transistor.
L	Comprimento do canal do transistor.
V_{T}	Limiar de tensão de um transistor.
C_{OX}	Capacitância entre Porta e canal por unidade de área.
μ	Mobilidade.
VDD	Potencial positivo de alimentação do circuito.
GND	Potencial de referência.
VSS	Potencial negativo de alimentação do circuito.
Vcm	Potencial de modo comum.
$Vref$	Referência de tensão.
ω	Frequência angular fundamental do sinal.

GLOSSÁRIO

ADC	Conversor Analógico-Digital
DAC	Conversor Digital-Analógico
A/D	Analógico-Digital
D/A	Digital-Analógico
S/H	Amostrador *Sample and Hold*
SAR	Registrador de Aproximações Sucessivas
THD	Distorção harmônica total
SNDR	Razão de Sinal-Ruído mais distorção
SNR	Razão de Sinal-Ruído
ENOB	Número de bits efetivo
SFDR	Faixa dinâmica livre de espúrios
DNL	Erro diferencial de não linearidade
DNL	Erro integral de não linearidade
UnB	Universidade de Brasília
R	Resolução do conversor em Volts
Q	Carga em Coulombs
LSB	Menor bit significativo
MSB	Maior bit significativo
CMOS	Tecnologia Metal-Óxido semicondutor complementar
NMOS	Metal-Óxido semicondutor de canal n

PMOS	Metal-Óxido semicondutor de canal p
MS/s	Mega amostras por segundo
f_s	Frequência de amostragem
V_f	Faixa dinâmica em Volts

CAPÍTULO 1

INTRODUÇÃO

1.1 MOTIVAÇÃO

O desenvolvimento tecnológico tem causado grandes impactos na sociedade, revolucionando a forma como trabalhamos, nos comunicamos e cuidamos de nossa saúde. A democratização da informação, proporcionada por esse desenvolvimento, tornou-se também uma importante ferramenta social em benefício da humanidade, tornando a informação mais acessível a todos. São inúmeros os benefícios e aplicações oferecidos pelos avanços tecnológicos. Grande parte dessas aplicações advém das vantagens proporcionadas pelo processamento digital.

O processamento digital de sinais oferece muitas vantagens em relação ao processamento analógico, como a programabilidade, maior confiabilidade, redução dos efeitos do envelhecimento de componentes e redução da dependência da temperatura na performance. Os sistemas eletrônicos modernos processam e armazenam informação digitalmente, entretanto, devido a natureza analógica do mundo, a conversão analógica-digital e a conversão digital-analógica possuem um papel muito importante no processamento de sinais e no desenvolvimento de novas aplicações e tecnologias. Os conversores analógico-digital (A/D) e digital-analógico (D/A) são os responsáveis por fazer essa ligação entre o mundo analógico e os sistemas de processamento digital.

Nesse cenário, os conversores Analógico Digitais (ADC) ganham cada vez mais importância, sendo muitas vezes cruciais e limitantes no desempenho de várias aplicações, como por exemplo, em sistemas de áudio, sensores, sistemas de controle de plantas industriais e sistemas automotivos.

O número de aplicações de processamento e controle de sinais é muito extenso e está sempre crescendo. Essa diversidade de aplicações exige que sejam desenvolvidos ADCs capazes de satisfazer especificações bastante distintas entre si. Assim, visando atender a diversas especificações

de desempenho, são desenvolvidas várias arquiteturas.

A escolha da melhor arquitetura para uma determinada aplicação está atrelada ao casamento entre as características da arquitetura e as especificações que devem ser privilegiadas na aplicação. Certamente, uma das mais antigas e conhecidas arquiteturas de ADCs é a de conversão por aproximação sucessiva, mais conhecida como SAR ADC. Apesar de ter sido substituída por outras arquiteturas no passado, vários trabalhos recentes têm mostrado que o SAR ADC é uma arquitetura de boa resolução, baixo consumo e média velocidade comparado à arquiteturas tradicionais (MANGANARO, 2011).

Tendo em vista a disponibilidade de tempo para a realização deste trabalho e o nível de conhecimento do autor na área de microeletrônica e projeto de circuitos integrados, optou-se por realizar a implementação de um conversor analógico-digital por aproximações sucessivas em tecnologia CMOS, visando-se assim o aprendizado de técnicas de projeto de circuitos integrados. O projeto do conversor completo foi dividido em duas frentes, o projeto dos módulos analógicos do conversor, realizado neste trabalho, e o projeto do módulo digital do conversor, realizado em outro trabalho pelo estudante Pedro Tolentino. Apesar da divisão do projeto em duas frentes, foi decidido que ambos os estudantes participariam do desenvolvimento do projeto dos módulos analógicos e do módulo digital do projeto.

1.2 OBJETIVO

Neste trabalho, o objetivo é realizar o projeto dos módulos analógicos de um conversor analógico-digital por aproximações sucessivas. O conversor será projetado na tecnologia XFAB CMOS 0.18 μm. A proposta é construir um conversor de 10 a 12 bits com taxa de amostragem de até 1 MS/s e baixo consumo. A fim de se definir especificações iniciais para o projeto do conversor SAR, foi feito um estudo bibliográfico de projetos de implementação de conversores SAR. A partir do estudo bibliográfico, e ainda levando em conta as condições de tempo para realização do trabalho, escolheu-se as especificações iniciais mostradas na tabela 1.2 para dar início ao projeto do conversor.

Especificações	
Vdd	1.8 V
Número de bits	10 - 12
Taxa de amostragem	1 MS/s
Temperatura	0 - 80 °C
Tecnologia	0.18 μm

Tabela 1.1. Tabela de especificações

1.3 ORGANIZAÇÃO

O trabalho está organizado como segue. O capítulo 2 faz uma breve introdução às arquiteturas clássicas de conversores analógico-digitais, às métricas de performance destes conversores e ao algoritmo de funcionamento de um conversor analógico-digital SAR. O capítulo 3 apresenta a metodologia utilizada no desenvolvimento do projeto, bem como o detalhamento de cada passo do fluxo de projeto seguido. O capítulo 4 descreve o desenvolvimento do modelo de referência do conversor SAR realizado em Matlab, o estudo dos blocos analógicos do conversor e por último apresenta o funcionamento esperado da máquina de estados do conversor. O capítulo 5 apresenta os resultados do projeto e a discussão acerca desses resultados. Por fim, o capítulo 6 apresenta as conclusões e trabalhos futuros propostos.

CAPÍTULO 2

FUNDAMENTAÇÃO TEÓRICA

Este capítulo trata de aspectos importantes na caracterização de conversores analógico digitais e ainda faz uma breve introdução às arquiteturas clássicas de conversores analógico-digitais. Tradicionalmente, as principais especificações que caracterizam ADCs e DACs são a resolução e a frequência de amostragem f_s. Entretanto, como veremos a seguir, outros parâmetros podem ser mais significativos na caracterização da performance desses conversores, dependendo do contexto e aplicação (MANGANARO, 2011).

A frequência de amostragem de um conversor é a frequência com que o mesmo pega uma amostra do sinal a ser digitalizado. De acordo com o critério de Nyquist, para que um sinal possa ser convertido sem perda de informação, a taxa de amostragem mínima deve ter o dobro da frequência do sinal amostrado. Na prática, é comum utilizar uma taxa de amostragem pelo menos dez vezes maior que a frequência do sinal amostrado, para que se obtenha bons resultados na representação de um sinal em sua forma digital.

A resolução de um conversor analógico digital diz respeito a quantidade de valores discretos que um conversor é capaz de representar dentro de uma faixa de tensão. O termo resolução pode também ser usado para designar apenas a quantidade de bits de um conversor. Neste trabalho, será usado apenas a definição de resolução identificada na equação (2.1), em que n é o número de bits do conversor, V_f representa a faixa dinâmica que esse conversor atua e R é a resolução desse conversor.

$$R = \frac{V_f}{2^n - 1}\ Volts \tag{2.1}$$

A medida que a tensão de entrada aumenta dentro de um nível discretizado de tensão, o erro de conversão também aumenta, como pode ser visto na figura 2.1. Isso acontece porque a tensão de entrada varia, mas a resposta continua sendo representada por um mesmo nível de tensão. Até que a valor da entrada alcance o próximo nível de tensão discretizado pelo

conversor, o erro só aumenta. Esse erro causado pela incerteza na quantização é conhecido como erro de quantização e em situações normais não deve exceder 1 LSB. Este tipo de erro é inerente a qualquer processo de conversão analógico-digital e digital-analógico.

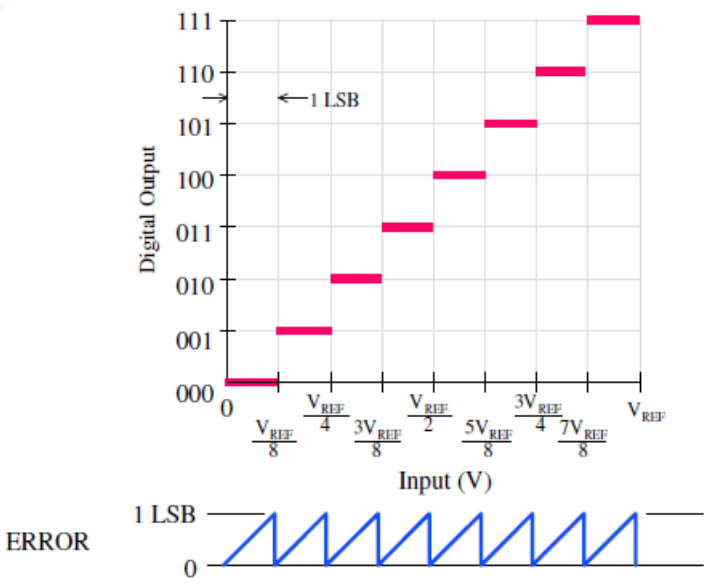

Figura 2.1. Erro de Quantização (GRAY, 2003)

A seguir, será apresentado um resumo de importantes características de performance de conversores analógico-digitais. Primeiro, é importante distinguir essas características entre estáticas e dinâmicas. As características estáticas do conversor são medidas a baixas frequências, preferencialmente usando sinais DC, portanto, referem-se a performance do conversor a baixas frequências. Já as características dinâmicas devem ser avaliadas a partir da resposta do conversor em seu espectro de frequências, costumam ser testadas utilizando sinais AC variando ao longo de toda a faixa dinâmica do conversor.

2.1 CARACTERÍSTICAS ESTÁTICAS

Devido as imperfeições de dispositivos à nível de circuito, durante o processo de conversão, as transições entre códigos podem ocorrer em pontos inesperados, degradando a resposta do conversor. Essas imperfeições são as principais fontes de erros estáticos em conversores.

2.1.1 Erro diferencial de não linearidade

Em um conversor ideal, os pontos de conversão entre um código e outro devem estar separados por exatamente 1 LSB, que corresponde a menor variação na tensão de entrada que o conversor é capaz de perceber. A diferença entre o LSB ideal e o pior caso, em que o desvio entre os níveis do conversor se afastam de 1 LSB, é chamado de DNL ou erro diferencial de não linearidade.

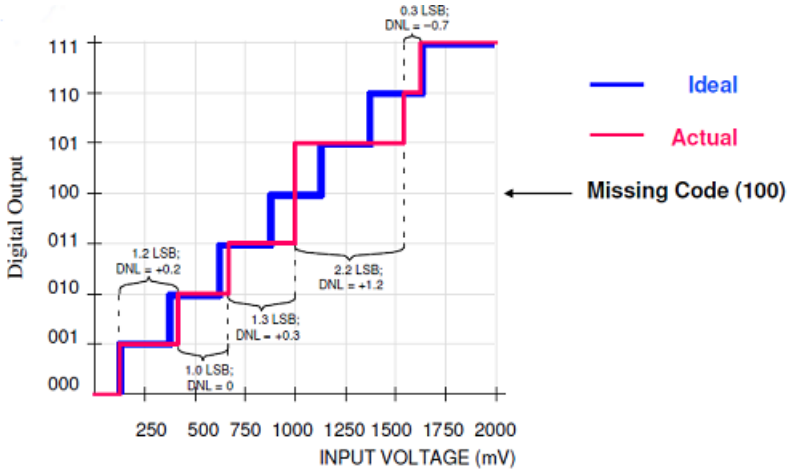

Figura 2.2. Análise do DNL de um ADC de 3 bits (GRAY, 2003)

Na figura 2.2, usando a função de transferência de um ADC de 3 bits, verifica-se a forma com que o DNL de um conversor é identificado. O DNL representa a diferença entre um passo ideal e um passo real do conversor e é medido em relação ao LSB do conversor. Quando o passo real é maior que o passo ideal, DNL > 1, já quando o passo real é menor, DNL < 1. Em termos práticos, pode-se medir o DNL de um conversor inserindo em sua entrada um sinal em forma de rampa, que passe por toda a faixa dinâmica do conversor, deste modo, é obtida a função de transferência do conversor. A partir da função de transferência, pode-se fazer a mesma análise realizada na figura 2.2 e obter o DNL do conversor.

2.1.2 Erro integral de não linearidade

O erro integral de não linearidade, ou INL, descreve o desvio acumulado entre a função de transferência real e a ideal de um ADC. O INL não inclui erros de quantização, ganho e offset, é uma medida do quão reta é a função de transferência. O desvio entre a função de transferência

real e ideal ocorre por conta dos erros de DNL, assim, o INL é resultado do acúmulo de DNL.

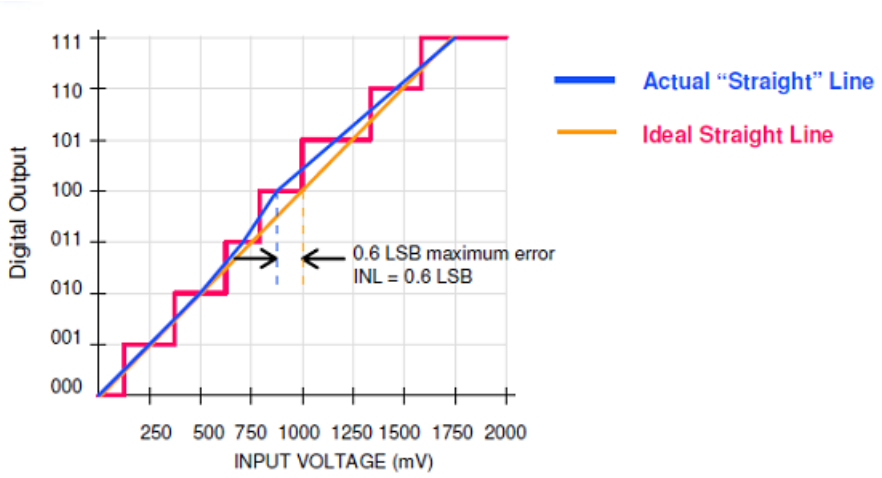

Figura 2.3. Análise do INL de um ADC de 3 bits (GRAY, 2003)

Já que o INL representa o desvio acumulado da função de transferência ideal do ADC, este pode ser representado, para um código de saída i, pela equação (2.2).

$$\mathsf{INL}_{(i)} = \Sigma_{i=1}^{N} DNL \tag{2.2}$$

2.1.3 Erro de Offset e Ganho

O erro de *offset* de um ADC quantifica o quanto a função de transferência real está linearmente afastada de sua posição original na base da função de transferência ideal. Para a determinação do *offset* de um conversor deve-se comparar as bases das funções de transferência real e ideal. O erro de *offset* será dado pela diferença entre essas bases e pode ser medido em LSB. A figura 2.4 ilustra o efeito do erro de *offset*.

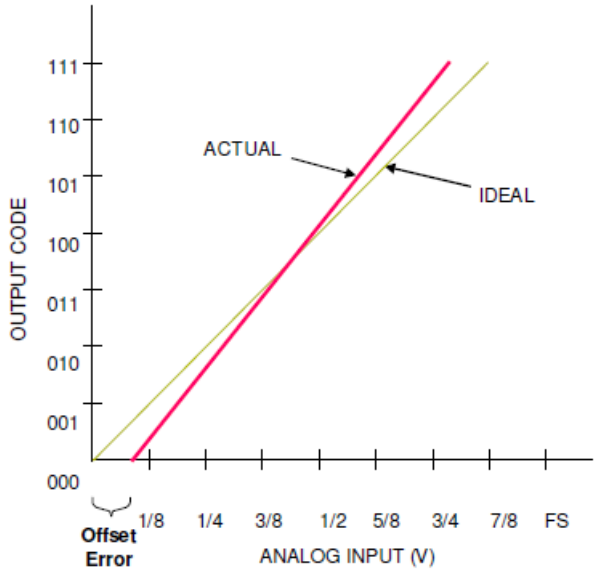

Figura 2.4. Análise do erro de *offset* de um ADC de 3 bits (GRAY, 2003)

O erro de ganho é uma medida do quanto a inclinação da função de transferência real se afasta da função de transferência ideal. Para realizar o cálculo do erro de ganho podemos deslocar a função de transferência real para eliminar o erro de *offset* e medir a distância entre a função de transferência ideal e a real deslocada em LSB.

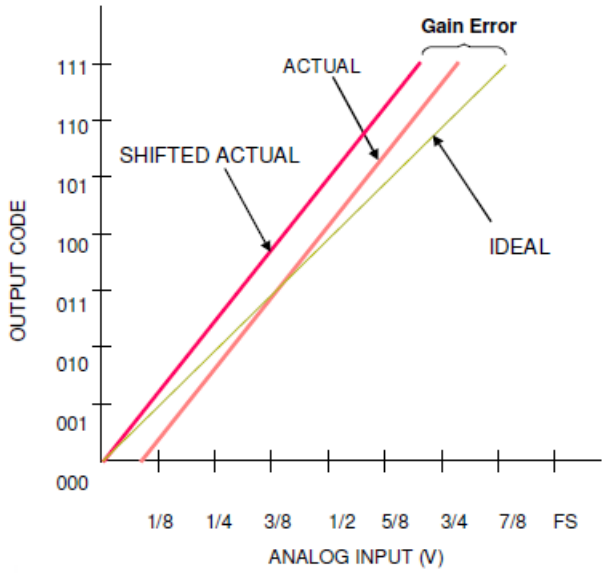

Figura 2.5. Análise do erro de Ganho de um ADC de 3 bits (GRAY, 2003)

2.2 CARACTERÍSTICAS DINÂMICAS

A medida que o conversor passa a trabalhar em frequências mais altas, efeitos que eram desprezíveis a baixas frequências começam a aparecer, degradando a resposta do conversor. Estes efeitos são classificados como erros dinâmicos do conversor e costumam ser dependentes em amplitude e frequência. A seguir, serão apresentados algumas das mais importantes medidas de performance dinâmica de ADCs.

2.2.1 Razão Sinal Ruído e Distorção

A especificação dinâmica mais importante de um conversor é sua Razão Sinal-Ruído (SNR) (PLASSCHE, 2013). A SNR depende da resolução do conversor e automaticamente inclui especificações de linearidade, distorção, ruído e tempo de acomodamento.

Geralmente são usados sinais senoidais para caracterizar a SNR de conversores. A SNR de um conversor é a razão entre a potência do sinal de saída e a potência do ruído. Já que o erro de quantização é inerente ao processo de conversão, a SNR de um conversor ideal pode ser determinada pela razão entre a potência do sinal de saída e a potência do erro de quantização. Para uma entrada senoidal, o SNR ideal de um conversor é dado pela equação (2.3), em que N é o número de bits do conversor.

$$SNR_{(\mathrm{dB})} = 6.02 \times N + 1.76 \ dB \tag{2.3}$$

O SNR de um conversor real, incluindo seus efeitos não ideais, pode ser avaliado através da resposta espectral do sinal de saída do conversor. Para um sinal senoidal, a SNR é a razão entre a potência do harmônico fundamental e a potência do ruído total dentro da banda de interesse, sem levar em conta a potência dos harmônicos.

Uma resposta dinâmica ainda mais exigente pode ser obtida considerando a distorção harmônica total (THD) do conversor. O THD é a razão entre a potência dos harmônicos e a potência do harmônico fundamental.

$$THD = \frac{P_{\text{harmônicos}}}{P_{\text{sig}}} \tag{2.4}$$

Ao considerar ruído e distorção, pode-se analisar a especificação de performance dinâmica conhecida como Razão Sinal Ruído e Distorção (SNDR). Essa especificação representa a relação entre a potência do sinal e a soma do ruído com a potência dos harmônicos.

$$SNDR = \frac{P_{\text{sig}}}{P_{\text{ruído}} + P_{\text{harmônicos}}} \tag{2.5}$$

2.2.2 Faixa Dinâmica Livre de Espúrios

A faixa dinâmica livre de espúrios (SFDR) é a faixa de tensão de entrada utilizada pelo conversor antes da interferência de qualquer ruído ou espúrio (componente de distorção). O SFDR pode ser calculado através da razão entre a potência do maior componente espectral desejável e o maior espúrio encontrado ao longo do espectro utilizável pelo conversor.

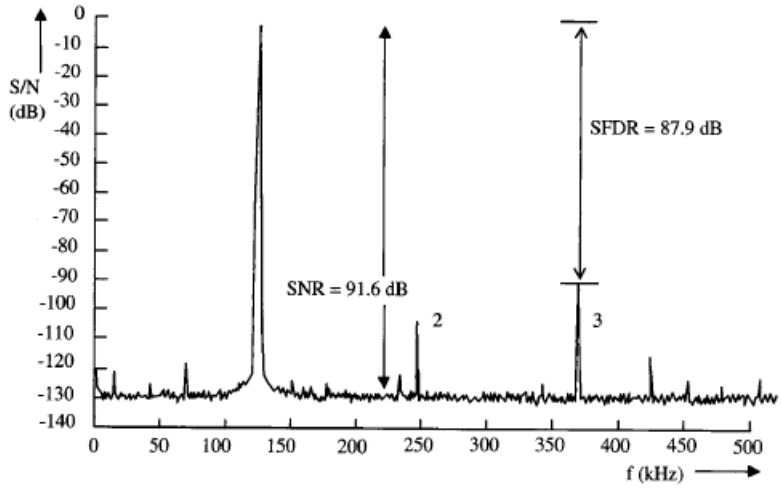

Figura 2.6. Exemplo de SFDR (PLASSCHE, 2013)

Na figura 2.6, a SFDR é calculada entre o ponto que representa o harmônico fundamental do sinal de entrada e o ponto em que a maior componente de distorção é definida.

2.2.3 Número de bits efetivo

Um conversor com n bits não precisa necessariamente funcionar com toda a precisão implicada pelos n bits, não porque o conversor não está trabalhando sob a performance adequada, mas porque o desempenho desejado pode estar relacionado a outros parâmetros do conversor. Um método para se comparar a performance de conversores que possuem o mesmo número de bits, mas que devido a diferentes topologias possuem desempenho diferente, é a comparação entre o Número de Bits Efetivo (ENOB) de cada um. O ENOB é o número de bits efetivo calculado a partir da SNDR medida no conversor.

$$ENOB = \frac{SNDR_{\text{dB(medido)}} - 1.76dB}{6.02dB} \tag{2.6}$$

2.3 ARQUITETURA DE ADCS

2.3.1 ADC Flash

Entre as arquiteturas clássicas, o ADC *Flash* é o tipo de conversor mais rápido (BAKER, 2008). Como visto na figura 2.7, a referência de tensão é dividida entre 2^N valores, sendo N o número de bits do conversor. O sinal de entrada Vin é comparado com cada um dos valores divididos pelos resistores, o resultado da comparação forma um código termométrico que pode ser decodificado em outro código, por exemplo, o código binário, e apresentado na saída do sistema.

A grande vantagem dessa arquitetura é a velocidade na qual uma amostra pode ser convertida, todos os bits do conversor são determinados em apenas um ciclo de *clock*. Entre as desvantagens dessa arquitetura estão o alto consumo e a relação entre área e resolução. No diagrama da figura 2.7 fica claro que um conversor *Flash* de N bits requer um circuito com (2^N-1) comparadores e 2^N resistores, portanto, para se aumentar um 1 bit na resolução do conversor deve-se dobrar o número de resistores e praticamente dobrar o número de comparadores. Por conta dessa desvantagem, os conversores *Flash* costumam ter resoluções limitadas a 8 bits (MANGANARO, 2011).

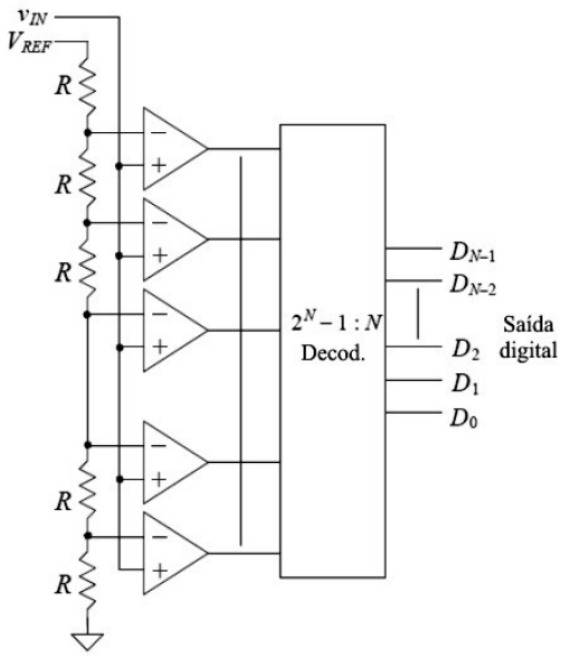

Figura 2.7. Diagrama de blocos do ADC *Flash* (BAKER, 2008)

2.3.2 ADC Pipeline

O conversor *Pipeline* é um conversor capaz de alcançar altas resoluções à velocidades relativamente altas (BAKER, 2008). A figura 2.8 mostra o funcionamento dessa arquitetura, que é composta por circuitos S/H, comparadores, somadores e amplificadores de ganho 2. O algoritmo do conversor começa com o sinal de entrada sendo amostrado e retido pelo S/H, em seguida, este sinal é comparado com metade da referência de tensão. Se a resposta do comparador indicar que o sinal de entrada é maior que a metade da referência de tensão, então o MSB será 1 e o sinal amostrado será subtraído da metade da referência de tensão, caso a resposta do comparador indique o contrário, o MSB será 0 e o sinal não é subtraído no somador.

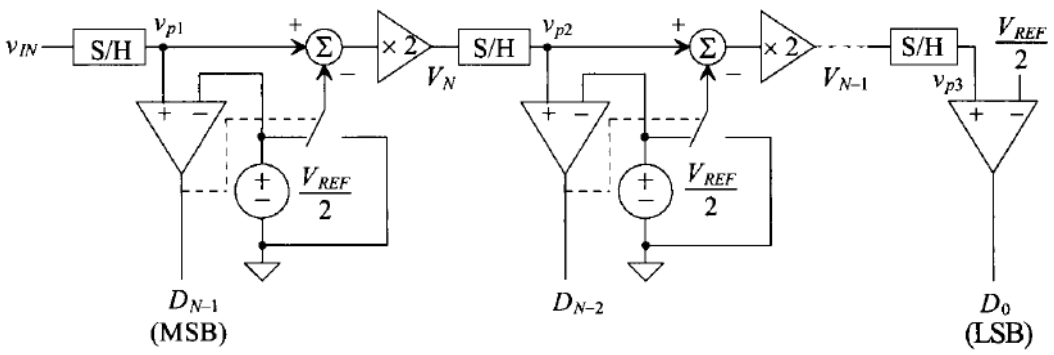

Figura 2.8. Diagrama de blocos do ADC *Pipeline* (BAKER, 2008)

Uma grande vantagem dessa topologia é que ao final da conversão da primeira amostra, o conversor passa a ser capaz de realizar as próximas conversões em apenas um único ciclo de *clock*. Isso acontece pois após o conversor decidir o valor de um bit de uma amostra, este já é capaz de decidir o mesmo bit da próxima amostra no próximo ciclo de *clock*. Essa característica faz com que o conversor Pipeline possua um alto *throughput*, ou seja, uma alta taxa de entrada de amostras, fazendo com que este conversor seja relativamente rápido em suas conversões.

2.3.3 ADC Sigma-Delta

Os conversores *Flash*, *Pipeline* e SAR são classificados como conversores de Nyquist, pois estes convertem sinais a uma taxa de amostragem próxima à taxa de Nyquist destes sinais. O conversor Sigma-Delta é classificado como conversor super amostrador, pois esse tipo de conversor amostra os sinais de entrada a taxas muito maiores do que a banda destes sinais. A vantagem dos conversores super amostradores é a alta resolução que estes conversores são capazes de alcançar, geralmente maiores que 16 bits.

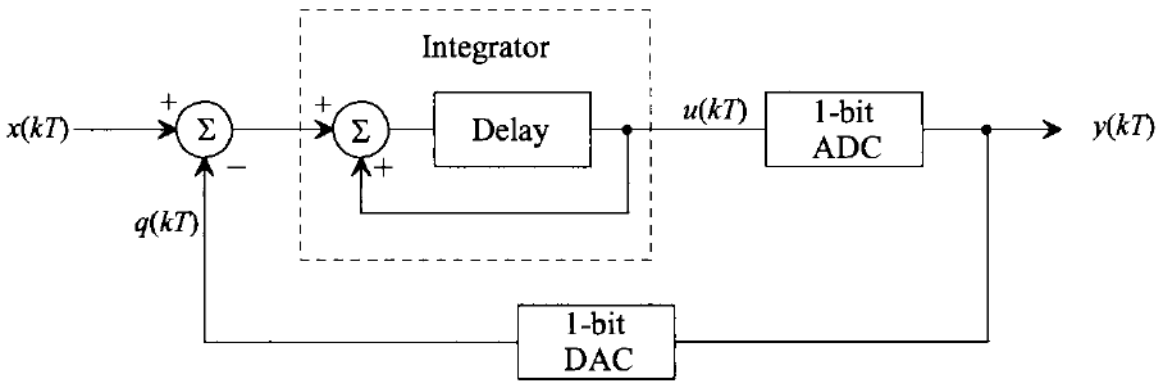

Figura 2.9. Diagrama de blocos do ADC *Sigma-Delta* (BAKER, 2008)

O conversor Sigma-Delta possui esse nome por utilizar modulação Sigma-Delta (ZRILIC, 2005) durante o seu processo de conversão. O sinal de saída do modulador é um sinal modulado por pulsos que representam a média do sinal de entrada. O circuito costuma utilizar a técnica de capacitores chaveados no somador, por isso, não é necessário o uso de S/H. A desvantagem desse tipo de conversor é a baixa velocidade de conversão devido a necessidade do conversor super amostrar o sinal de entrada. Portanto, o conversor Sigma-Delta costuma ser usado apenas em aplicações que exigem alta resolução e baixa velocidade, como é o caso em aplicações de

sistemas de áudio.

2.3.4 O ADC por aproximação sucessiva

O conversor por aproximações sucessivas, conhecido como ADC SAR (Successive Approximation Register), é uma das topologias mais antigas e conhecidas de ADCs. Esses conversores costumam ser usados em aplicações de baixa ou média velocidade e possuem resolução de média para alta, geralmente entre 8 e 16 bits. A grande vantagem deste tipo de conversor é o baixo consumo de energia, que o faz muito interessante para aplicações em dispositivos remotos.

Na arquitetura do SAR os principais circuitos podem ser divididos entre blocos, entre eles, o S/H, o comparador, o conversor digital analógico e a máquina de estados. Tendo conhecimento do papel de cada um desses blocos, fica mais fácil compreender o algoritmo de funcionamento do conversor SAR.

No conversor, o circuito S/H, conhecido como *sample and hold*, é responsável por capturar valores de Vin (sinal analógico de entrada) e guarda-los em um elemento de memória analógica, como um capacitor, por um período determinado. O comparador é um circuito de entrada analógica e saída digital, este circuito compara os sinais analógicos de entrada e apresenta um bit na saída que indica qual dos dois sinais é maior. O DAC é o bloco que converte um sinal digital, representado por símbolos, em um sinal analógico. A máquina de estados é o bloco responsável por fazer o controle de todos os circuitos dentro do conversor. No capítulo 4, o funcionamento de cada um destes blocos é abordado com mais profundidade.

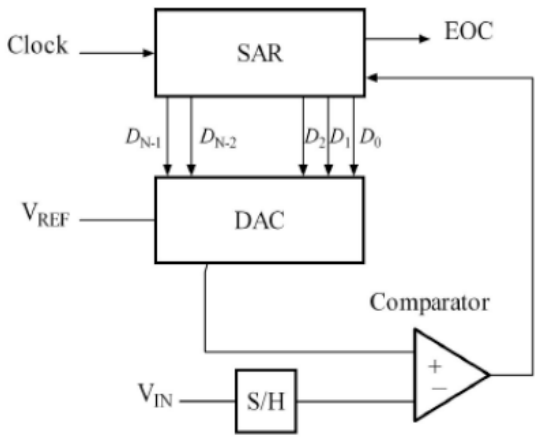

Figura 2.10. Diagrama de blocos ADC SAR (BAKER, 2008)

O funcionamento do conversor SAR é bem simples e intuitivo, como mostrado na imagem (2.10). O algoritmo genérico do SAR segue os seguintes passos:

1. O sinal Vin é amostrado no S/H.
2. A máquina de estados envia para o DAC a primeira aproximação do resultado final, que corresponde a palavra binária com MSB igual a 1 e o resto dos bits sendo 0.
3. No DAC, a palavra digital é convertida em analógica.
4. O sinal convertido no DAC é comparado com o sinal amostrado pelo S/H.
5. A partir da resposta do comparador, a máquina de estados decide o valor do primeiro bit do resultado final.

Após a decisão do primeiro bit da resposta, os passos são repetidos de 2 a 5, até que todos os bits sejam determinados.

2.3.4.1 Single-Ended

Para um conversor SAR single ended, o sinal de entrada do conversor é carregado na entrada negativa do comparador, enquanto a entrada positiva do comparador permanece ligada, durante todo o processo de conversão em uma tensão de modo comum V_{cm}. Um critério importante que deve ser levado em consideração na escolha de V_{cm} é o ICMR (*Input Common Mode Range*) do comparador. O processo completo de conversão pode ser dividido em três fases: amostragem, contenção e comparação.

Na fase de amostragem, a entrada analógica V_{in} é amostrada na chave S/H. Em seguida, a tensão de entrada analógica amostrada é conectada aos terminais negativos do array de capacitores, sendo esses os terminais que não estão conectados à entrada do comparador.

A seguir, inicia-se a fase de contenção. A tensão de modo comum (mesma conectada na entrada positiva do comparador) é conectada ao terminal positivo do array de capacitores. Neste momento, o array de capacitores deve estar carregado com uma carga igual a $2 \times C \times (V_{cm} - V_{in})$. Em que C representa a capacitância do maior capacitor.

Agora, na fase de comparação, a máquina de estados gera a primeira palavra binária a ser comparada com a tensão de entrada. Cada bit irá controlar a tensão ligada ao terminal negativo de cada capacitor, com exceção de um dos capacitores de menor capacitância, que vai permanecer ligado à gnd. O capacitor que receber um bit 1 será conectado à V_{ref}. O capacitor que receber um bit 0 será conectado à gnd. Assim, o array de capacitores funcionará como um divisor de tensão capacitivo, subindo a tensão do array para uma tensão igual a parcela de V_{ref} representada pelos bits.

Neste ponto, a carga e a tensão sob os capacitores são iguais à:

$$Q = 2 \times C \times (V_{\text{cm}} - V_{\text{in}} + V_{\text{DAC}})\ \ Coulombs \tag{2.7}$$

$$V_{\text{-}} = V_{\text{cm}} - V_{\text{in}} + V_{\text{DAC}}\ \ V \tag{2.8}$$

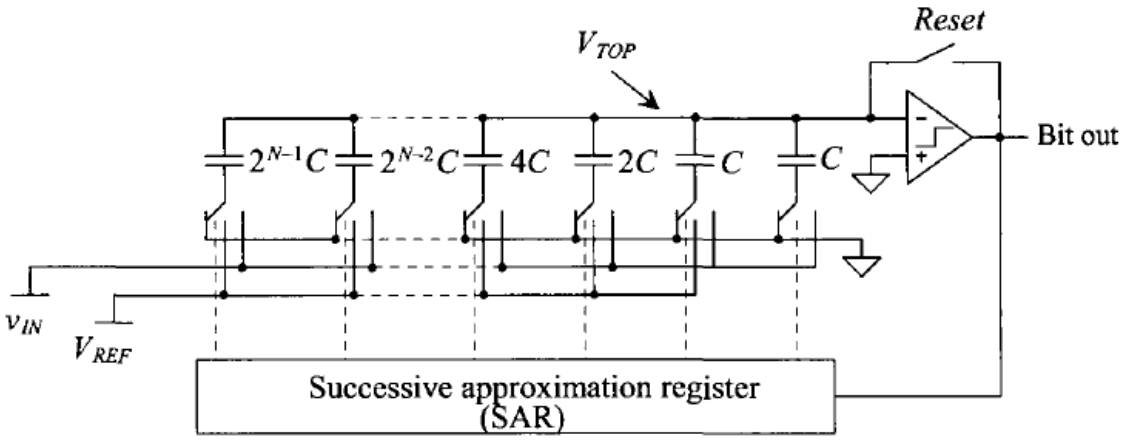

Figura 2.11. Conversor Analógico-Digital *single ended* (BAKER, 2008)

Agora, o comparador já pode atuar comparando a tensão de modo comum com a tensão proveniente do array de capacitores. Olhando para a equação (2.8), observe que a comparação está sendo realizada entre V_{in} e V_{DAC}. Pois caso $V_{DAC} - V_{in}$ for menor que zero, o comparador apresenta 1 na saída, caso contrário o comparador apresenta um bit 0. O resultado da comparação indica o valor do bit que está sendo comparado neste momento.

CAPÍTULO 3

METODOLOGIA

Este capítulo apresenta todas as etapas do planejamento de execução do projeto, passando pela metodologia e fluxo de projeto utilizados.

Neste trabalho, antes de se iniciar o projeto do conversor SAR, fez-se um planejamento das etapas a serem realizadas para a conclusão do projeto. Utilizou-se uma metodologia *bottom-up* para o desenvolvimento deste projeto, em que o circuito foi primeiro implementado bloco a bloco e por último avaliado o conjunto de blocos por completo. As etapas do fluxo de projeto são:

- Definição das especificações
- Implementação de modelo de referência
- Escolha das topologias
- Escolha dos dispositivos
- Teste dos blocos isoladamente
- Teste do circuito completo

O primeiro passo no fluxo de projeto de um conversor SAR geralmente consiste na determinação das especificações do conversor. Entretanto, nesta primeira fase, optou-se por começar pela implementação de um modelo de referência, visto que ainda haviam algumas métricas de performance do conversor a serem definidas.

Quando se está desenvolvendo um conversor para uma aplicação específica, é importante possuir um bom entendimento das especificações desta aplicação para que seja possível privilegiar certos aspectos do conversor. Por exemplo, caso o conversor precise trabalhar em ambientes

Parâmetros	(MINH, 2015)	(MARJONEN, 2007)	(ZHU, 2015)	(ZHU *et al.*, 2015)
Vdd	1.8	1	0.6	0.9
Resolução (bits)	10	10	10	10
ENOB (bits)		9.17	9.4	9.07
Taxa de amostragem (MS/s)	25	0.001	0.02	2
Potência (μW)	3834	0.043	0.038	22.12
DNL (LSB)	0.7		0.46	0.5
INL (LSB)	3.6		0.44	1
SNDR (dB)		57.16		56.36
FoM (fJ/conv)	374.4	73	2.8	20.6
Ano	2015	2015	2015	2014
Aplicação		Biomédica	Biomédica	WSN

Tabela 3.1. Tabela comparativa de ADCs SAR em tecnologia 0.18 μm

em que a temperatura varie bastante, é importante que o projetista escolha topologias de circuito que sejam mais robustas à variação de temperatura. Quase todas as informações sobre a aplicação do conversor podem ser úteis na determinação do conjunto de especificações.

Existem também, conversores de propósito genérico, que podem ser usados em uma grande variedade de aplicações. O SAR é uma topologia muito boa para conversores de aplicação genérica, já que esta topologia permite um alto desempenho em quase todas as medidas de performance, sendo um conversor preciso, rápido e econômico em área e energia, em comparação com outras topologias clássicas de ADC.

Neste trabalho, optou-se por realizar um conversor para aplicações genéricas. A partir dessa escolha, surge a dificuldade em se determinar as especificações do conversor. Para isso, realizou-se uma pesquisa bibliográfica com a intenção de avaliar projetos de conversores SAR recentes e destinados a aplicações diversas. Neste momento, a única especificação determinada era a tecnologia a ser utilizada no projeto, que era a XFAB XC018. Portanto, buscou-se apenas conversores construídos em tecnologia CMOS 0.18μ.

Analisando as especificações dos trabalhos de desenvolvimento de ADCs SAR na tabela

(3.1) pode-se observar que enquanto algumas características são bastante comuns em todos os trabalhos, outras variam bastante de acordo com a aplicação. Nota-se que nessa tecnologia é comum trabalhar com alimentação igual a 1.8 Volts e com resolução de 10 bits. Visto essas características decidiu-se fazer um conversor SAR de 10 bits, frequência de amostragem igual 1.10^6 amostras por segundo e alimentação igual a 1.8 Volts. Portanto, inicialmente pretende-se fazer um projeto que alcance essas especificações principais e que ainda seja possível otimizar o consumo e as respostas estática e dinâmica do conversor.

Após definidas as especificações do sistema, começa-se a estudar as topologias de circuitos analógicos que permitem chegar às especificações desejadas. As topologias dos circuitos analógicos devem ser escolhidas de modo a respeitar a precisão e a frequência de operação definidas. Tanto o DAC quanto o comparador e a chave S/H devem ter uma precisão que supere o valor de 1 LSB, por segurança, é aconselhável projetar estes circuitos para uma precisão de 0.5 LSB. Esses três blocos são críticos para que o conversor atinja o desempenho esperado.

Com a tecnologia já determinada, deve-se escolher os dispositivos de acordo com o VDD estabelecido, transistores, capacitores e resistores disponíveis. Além de VDD é interessante observar a limitação destes componentes quanto a corrente e *mismatch*.

Utilizando todas essas informações, especificações, topologias e dispositivos deve-se iniciar o projeto de cada bloco separadamente, garantindo que estes atendam as especificações com uma certa folga.

Após o design dos blocos, deve-se realizar uma série de simulações para a verificação do comportamento dos blocos avaliados separadamente. Essas simulações devem ser realizadas a nível de transistor para cada esquemático envolvido no projeto. Em seguida, deve-se realizar simulações ligando um bloco a outro, observando o comportamento do sinal de resposta dos blocos em conjunto.

Por último, deve-se validar o comportamento do conversor por completo. Nesta fase, deve-se fazer simulações que validem o comportamento do conversor ao longo de toda a faixa de entrada especificada sob várias condições de frequência dentro das especificações. É importante avaliar DNL, INL, SNDR, potência consumida e ENOB para que seja possível comparar com outros trabalhos.

3.1 MODELO DE REFERÊNCIA

O modelo de referência consiste na descrição do sistema através de uma linguagem de alto nível. Apesar de neste trabalho ter sido desenvolvido em Matlab, este tipo de modelo também pode ser feito utilizando Simulink, Verilog-A ou mesmo circuitos ideais em simuladores do tipo Spice. Neste trabalho, escolheu-se desenvolver o modelo de referência utilizando scripts de Matlab, pois a implementação nesta ferramenta garante bastante flexibilidade ao modelo, permitindo testá-lo rapidamente sob diferentes condições. Além disso, a ferramenta possui várias funções úteis ao projeto e uma excelente documentação, que ajuda a reduzir o tempo gasto na implementação do modelo.

O trabalho em (AOUIZERATE, 2017) realiza um estudo da implementação de diferentes arquiteturas de ADC SAR utilizando Matlab, em que cada arquitetura é implementada e testada quanto a sua performance estática. O trabalho em (AOUIZERATE, 2017) foi utilizado como base para o desenvolvimento do modelo de referência deste projeto. Primeiro, buscou-se validar o modelo, em (AOUIZERATE, 2017), através do estudo de cada parte do código e da realização de testes, que consistiam na execução do código e na análise dos resultados.

Em (AOUIZERATE, 2017), o algoritmo utilizado para a implementação do ADC SAR *single-ended* é um pouco diferente do algoritmo explicado no capítulo 2, que inclui a tensão de modo comum no procedimento. Assim, o código foi modificado de modo a executar exatamente o algoritmo descrito em 2. Outra importante mudança realizada no código foi a inclusão de parâmetros da tecnologia XFAB 0.18 μ m na implementação do *array* de capacitores do conversor, permitindo prever efeitos das imperfeições da tecnologia na performance do conversor, que ajudam a definir certas especificações durante o projeto, como a escolha do valor de capacitância unitária do *array* e a resolução do conversor.

O código foi concebido a partir da ideia de se ter um modelo representativo de um conversor testando sua lógica de implementação e avaliando a performance do algoritmo de acordo com métricas pré-definidas. De modo geral, o código representa, em linguagem de alto nível, a concepção do ADC SAR e as ferramentas necessárias para a avaliação do conversor e suas medidas de performance DC. Para a realização dessa tarefa, definiu-se os seguintes passos: criar um sinal de entrada para o conversor, criar o próprio conversor (permitindo a inserção de

parâmetros da tecnologia utilizada), observar a resposta de saída do conversor e por último, avaliar o conversor a partir das métricas definidas.

Para realizar essa tarefa, o código foi implementado da seguinte forma:

- **Variáveis iniciais do modelo**: Esta parte do código declara cinco variáveis iniciais, sendo elas o número de bits do conversor, a referência de tensão, a tensão de modo comum, a inserção de erro de offset no conversor e a escolha entre um sinal de entrada em forma de rampa ou senoide.

- **Sinal de entrada**: Esta parte cria um dos tipos de sinais de entrada especificados na variável "sinal"(rampa ou senoide).

- **Array de Capacitores**: Nesta parte do código, é criado o *array* de capacitores do DAC. É possível escolher entre um *array* ideal e um *array* que segue uma distribuição gaussiana baseada em especificações da tecnologia adotada. Dessa forma, o código permite a simulação do efeito de *mismatch* de capacitores na curva de transferência do conversor.

- **Processo de Conversão Single Ended**: Nesta parte, é realizada a conversão de cada amostra do sinal de entrada em um sinal digital representado por um *array* de bits. O processo de conversão utilizado segue o mesmo algoritmo do conversor SAR *Single Ended* explicado no capítulo 2.

- **Recomposição do Sinal**: O sinal digital obtido é recomposto em um sinal de forma analógica, para que seja possível realizar uma comparação entre o sinal de entrada e o sinal digital de saída.

- **Análise do Conversor**: Nesta parte, são plotados os sinais de entrada e saída do conversor. Além disso, caso o sinal de entrada tenha sido a rampa, é feita uma análise das características estáticas do conversor DNL e INL. Caso o sinal de entrada seja uma senoide, é plotada a resposta espectral do conversor.

O modelo de referência possui variáveis como o número de bits, referência de tensão e parâmetros da tecnologia que permitem um estudo da correlação entre esses parâmetros e a resposta de saída do conversor. Assim, o modelo de referência foi aproveitado para validar as

especificações do projeto dado as características da tecnologia referentes às imperfeições dos capacitores. Esse aspecto foi fundamental no dimensionamento do *array* de capacitores do conversor. O código do modelo de referência descrito encontra-se no apêndice.

3.2 FLUXO DE PROJETO DE CADA BLOCO

A metodologia aplicada neste projeto é do tipo *bottom-up*. Desta forma, parte-se do desenvolvimento de cada bloco do circuito separadamente para ao final juntar todos esses blocos em um único bloco e assim testar e validar o comportamento do circuito por completo.

Dentro do projeto de cada bloco utilizou-se o seguinte fluxo de projeto para o desenvolvimento de cada um:

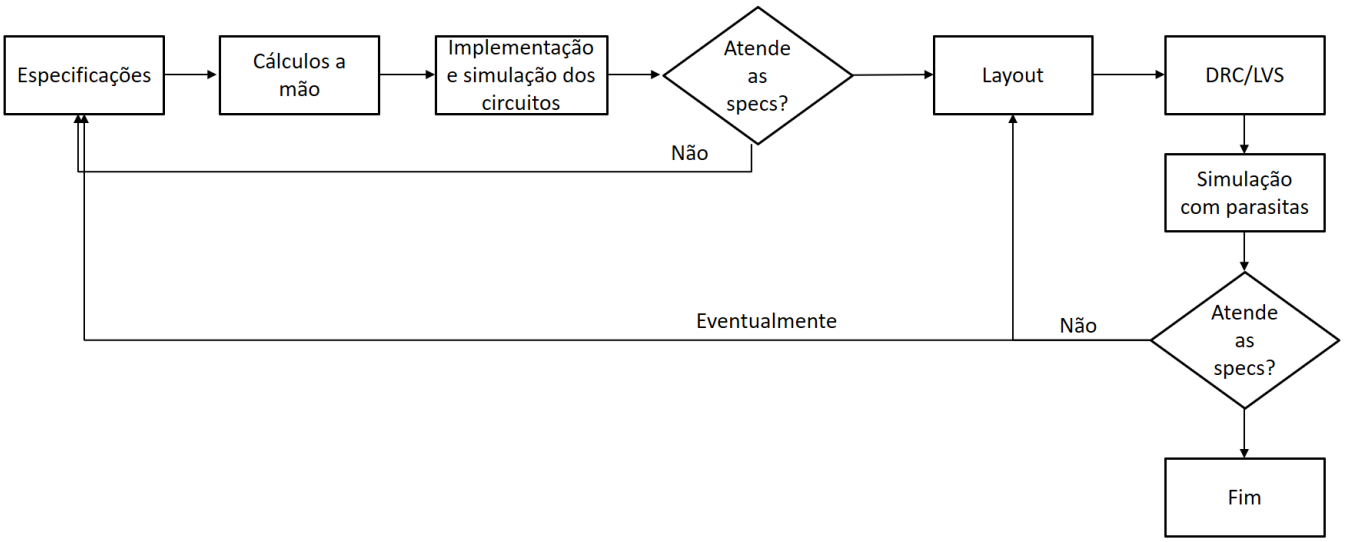

Figura 3.1. Fluxo de projeto

O fluxo de projeto começa com a determinação das especificações do bloco. Essas especificações de cada bloco dependem das especificações completas do conversor e ainda do algoritmo de implementação do conversor. Cada bloco possui características específicas no processo de conversão, por isso, são implementados de forma distinta e possuem especificações que dependem da sua função no processo de conversão. No próximo capítulo as especificações de cada bloco serão discutidas com profundidade.

Após a definição das especificações do bloco, inicia-se os primeiros cálculos a mão para dimensionar o tamanho dos transistores e outros dispositivos a serem implementados. Nesta

fase, é importante ter conhecimento de certos parâmetros da tecnologia, como K', λ e V_T dos transistores escolhidos. A documentação de XFAB traz informações referentes aos dispositivos da tecnologia, entretanto, ainda foi necessário realizar uma caracterização dos transistores da tecnologia para a obtenção de parâmetros que são utilizados na fase de cálculos a mão.

Em seguida, o bloco deve ser implementado e testado utilizando uma ferramenta de simulação de circuitos. Neste projeto, os circuitos foram implementados e testados utilizando o software Cadence. A partir deste ponto, inicia-se um processo iterativo com o objetivo de adequar o circuito às especificações determinadas inicialmente. Só avança-se para a próxima etapa depois da confirmação de que o circuito projetado atende as especificações esperadas. Eventualmente, pode-se chegar a conclusão de que não é possível chegar às especificações esperadas com a topologia utilizada, neste caso, deve-se repensar a topologia ou até mesmo as especificações do conversor completo.

Neste trabalho, depois de realizar as simulações de um bloco e validar o seu funcionamento, prossegui-se para o desenvolvimento do próximo bloco, antes da realização de layout. Desta forma, foi possível realizar simulações a nível de transistor com o circuito completo. Optou-se portanto em validar o circuito completo a nível de transistor antes de iniciar o desenvolvimento do layout dos blocos.

Apesar de não ter sido realizado layout dos blocos, vale a pena comentar sobre essa parte do fluxo de projeto. Após a validação do circuito completo a nível de transistor, seria iniciado o layout dos circuitos de cada bloco individualmente. Seriam executadas as ferramentas de DRC e LVS, a primeira é utilizada para conferir se as distâncias entre os dispositivos no layout respeitam as regras impostas pela tecnologia, a segunda é usada para comparar o circuito feito a nível de layout com o circuito feito a nível de transistor. Caso o layout não seja equivalente ao esquemático, a ferramenta de LVS aponta as disparidades entre ambos. Após esta fase, entraria-se novamente em um processo iterativo em que o circuito agora passaria por uma simulação que incluiria os efeitos parasitários. Assim, o circuito seria modificado até que este fosse capaz de novamente atender as especificações.

Após a realização e validação do layout de cada bloco, deveria ser realizada uma simulação do circuito completo a nível de layout. Depois que o circuito estivesse validado por completo através de simulações que incluíssem efeitos parasitas, o projeto estaria pronto para fabricação.

CAPÍTULO 4

PROJETO

Este capítulo apresenta o desenvolvimento de cada bloco especificado na metodologia do projeto. Como dito no capítulo anterior, o tipo de metodologia adotada seguiu um padrão *bottom-up*, em que cada bloco foi construído e validado separadamente. Entretanto, antes do desenvolvimento de cada bloco é importante obter um modelo capaz de caracterizar os dispositivos da tecnologia utilizada para que assim seja possível realizar os primeiros cálculos a mão do dimensionamento dos transistores. Este capítulo também apresenta o processo de caracterização de dispositivos utilizado.

O capítulo ainda mostra o desenvolvimento de um modelo de referência utilizado para estudo do algoritmo empregado na implementação do conversor A/D.

4.1 MODELO DE REFERÊNCIA

O intuito em desenvolver um modelo de referência é validar o comportamento esperado na saída do sistema e de cada subsistema, dado um conjunto de entradas conhecidas. Para tanto, desenvolveu-se um modelo funcional puramente ideal e genérico do algoritmo do conversor. Em (AOUIZERATE, 2017), foi realizado um estudo de vários algoritmos de topologias SAR através de modelos descritos em Matlab. Neste trabalho, realizou-se um modelo de referência baseando-se nos modelos de (AOUIZERATE, 2017).

O modelo de referência foi implementado em um script de Matlab e a topologia de SAR escolhida para sua implementação foi a single-ended, já que é esta topologia que pretende-se implementar neste projeto. O script segue o mesmo algoritmo de SAR single-ended descrito anteriormente. Além de permitir a construção do modelo ideal, o script permite a inserção de algumas fontes de erro como offset e mismatch de capacitores.

O script começa escolhendo o valor de algumas variáveis como o número de bits do conversor,

sua frequência de amostragem e sua tensão de referência. Caso deseja-se inserir erro de offset no conversor, a variável V_{off} pode ser alterada para inserir offset em função do LSB do conversor.

Outra forma de simular imperfeições no modelo de referência do conversor é inserindo mismatch entre os capacitores do array de capacitores. A fim de observar os efeitos do mismatch de capacitores na resposta do conversor, criou-se a variável C_mismatch, que permite escolher entre um array de capacitores sem mismatch, um array de capacitores com mismatch gaussiano ou uma array com mismatch fixo. Na opção de mismatch gaussiano, foi implementado um modelo que caracteriza o desvio padrão do mismatch de capacitores fornecido pela tecnologia, que varia com o tipo de capacitor e a área ocupada por esse.

No script, através da variável signal, é possível escolher entre uma rampa ou uma senoide como sinal de entrada do conversor. Fez-se esta opção para que fosse possível analisar a performance estática, caracterizada por um sinal em forma de rampa, e a performance dinâmica, caracterizada pela senoide.

Em seguida, o script determina o valor dos capacitores do array de capacitores, baseado na escolha da variável Cmismatch.

O processo de conversão é iniciado carregando a primeira amostra do sinal de entrada no array de capacitores, fazendo com que a tensão nos capacitores seja igual a tensão de modo comum menos a tensão do sinal de entrada amostrado. Em seguida, o nível de tensão no array sobe de acordo com a palavra binária colocada no DAC. O sinal é comparado no comparador e o bit mais significativo é escolhido. O processo é repetido até que todos os bits sejam escolhidos.

Com o intuito de visualizar o sinal digital de saída do conversor, o script converte o sinal digital em um sinal analógico equivalente. Logo após, os sinais de entrada e saída são plotados juntos para efeito de comparação.

Para avaliar melhor a qualidade da conversão, ao final, caso o sinal de entrada escolhido seja uma rampa, é feita uma análise das características estáticas DNL e INL do conversor. Assim, pode-se observar os efeitos de mismatch de capacitores e do offset no DNL e INL. Caso o sinal de entrada tenha sido a senoide, o espectro do sinal de entrada e do sinal de saída serão plotados, permitindo uma análise das características dinâmicas do conversor. Vários ensaios foram realizados para a validação de todas as funcionalidades do modelo de referência.

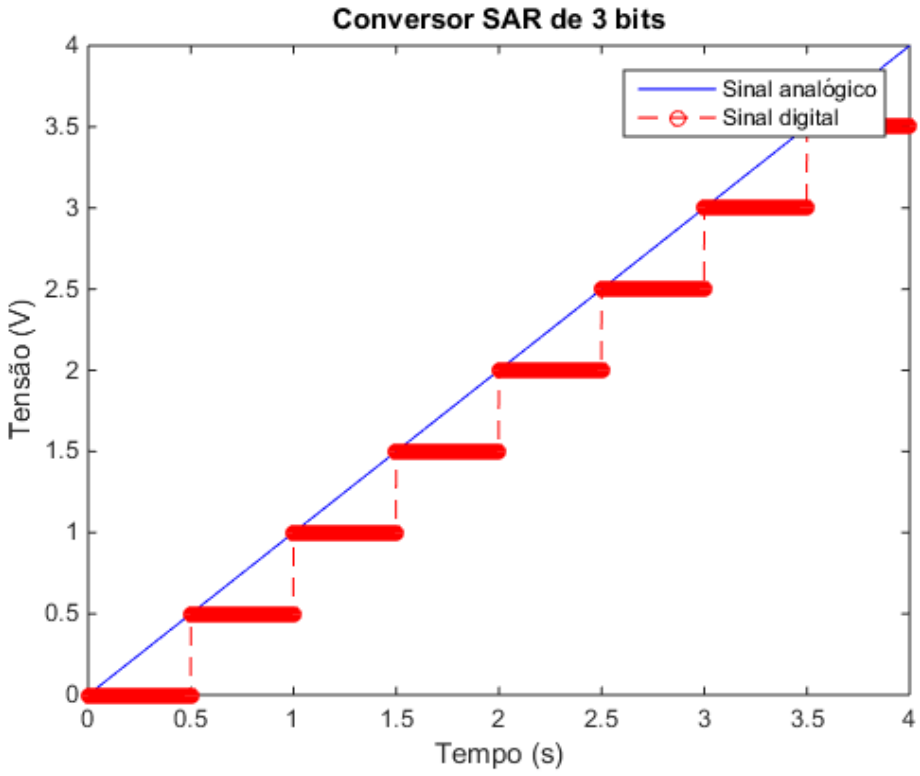

Figura 4.1. Simulação de conversor SAR ideal de 3 bits

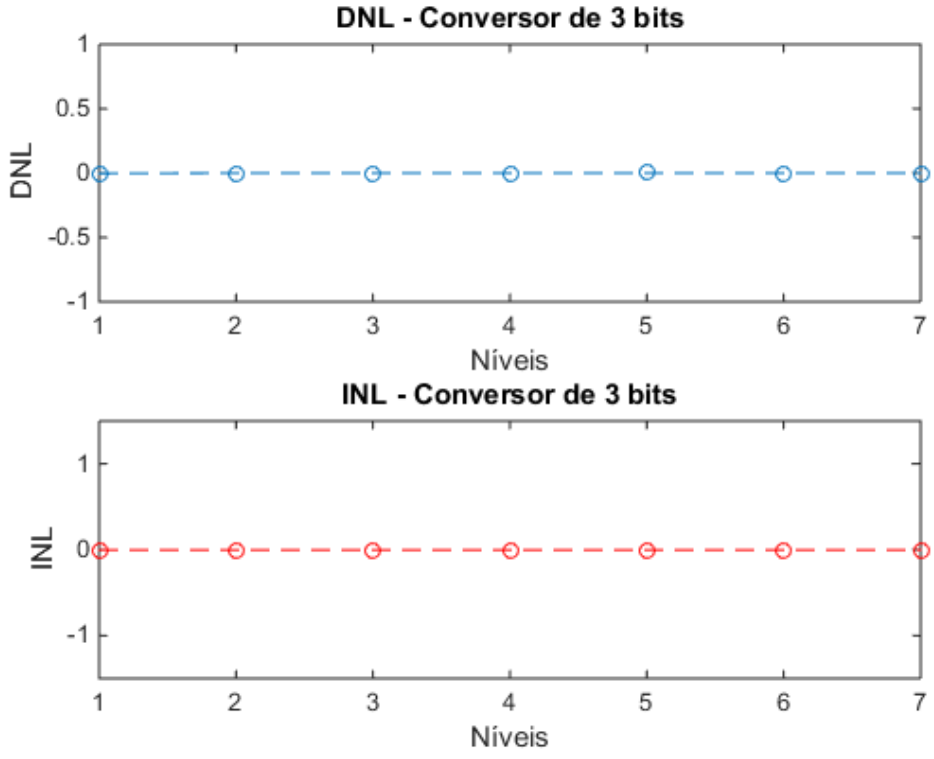

Figura 4.2. Análise de DNL e INL de conversor SAR ideal de 3 bits

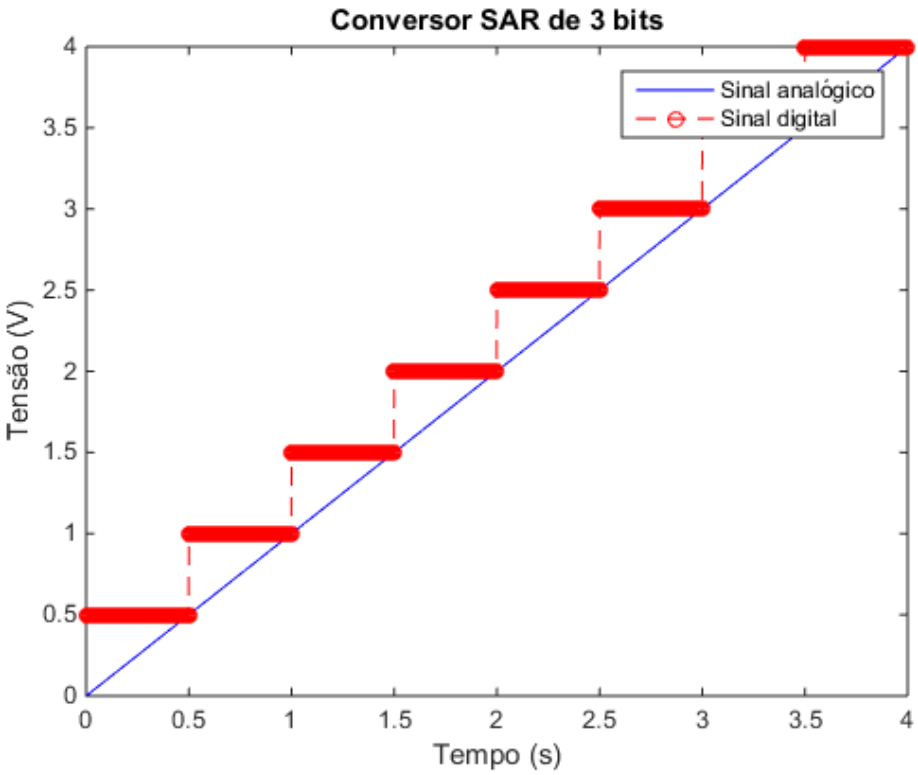

Figura 4.3. Simulação de conversor SAR de 3 bits com erro de offset

As figuras (4.1) e (4.2) mostram os resultados do ensaio de um conversor A/D de 3 bits ideal. Como esperado, o conversor apresentou degraus uniformes, espaçados de acordo com a resolução ideal de um conversor de 3 bits, dentro do intervalo de 0 a 4 volts. As características DNL e INL apresentaram pequenos erros devido a amostragem do sinal de entrada em Matlab. Esses pequenos erros são reduzidos a medida em que se aumenta a taxa de amostragem para a construção do sinal analógico no Matlab.

As figuras (4.3) e (4.4) mostram o ensaio com 1 LSB de offset. No ensaio, como esperado, o offset não causou alteração na largura dos passos, apenas deslocou a função de transferência do conversor. O offset também não surtiu nenhum efeito no DNL ou INL, já que essas são medidas diferenciais dos passos do conversor.

As figuras (4.5) e (4.6) são da simulação de um conversor de 5 bits com mismatch de até 10% do valor dos capacitores. Fica nítido a degradação da resposta de saída devido a inserção de mismatch nos capacitores. Nas figuras (4.7) e (4.8) é interessante observar a relação do número de bits do conversor e sua performance estática de saída considerando o erro de mismatch de capacitores. Observa-se que a medida em que se aumenta o número de bits a resposta de saída

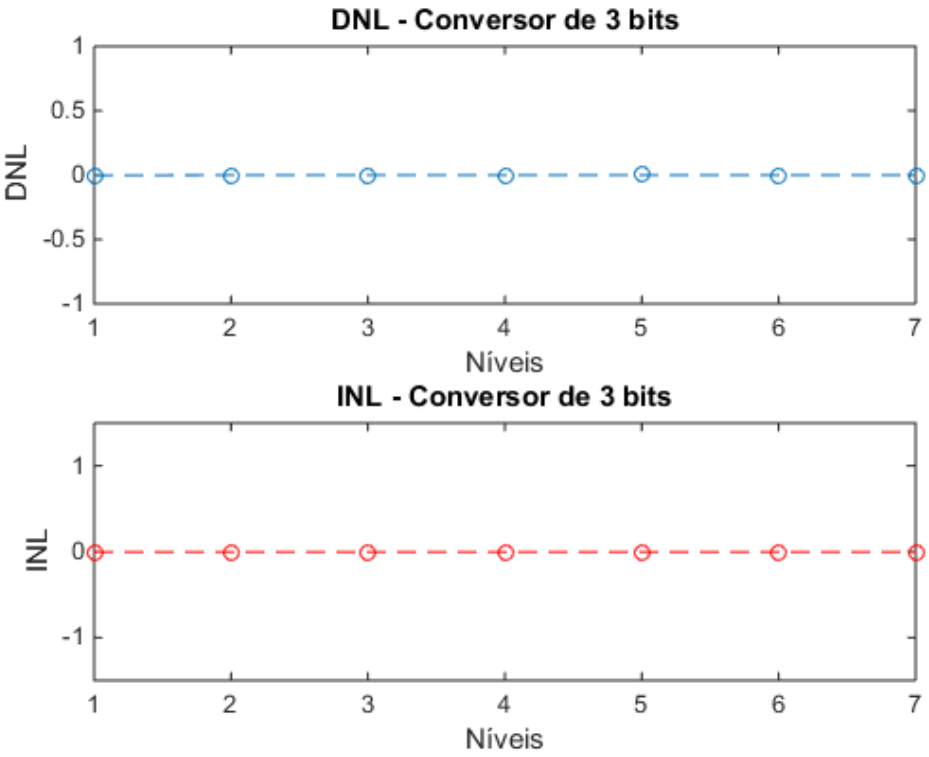

Figura 4.4. Análise de DNL e INL no conversor com erro de offset

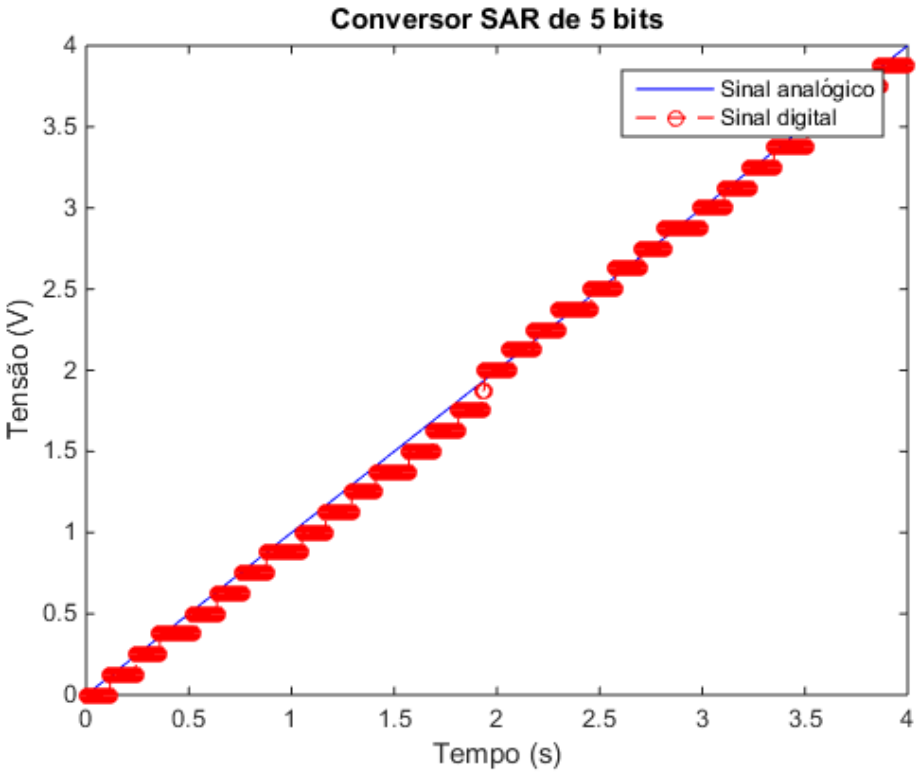

Figura 4.5. Simulação de conversor SAR de 5 bits com mismatch nos capacitores

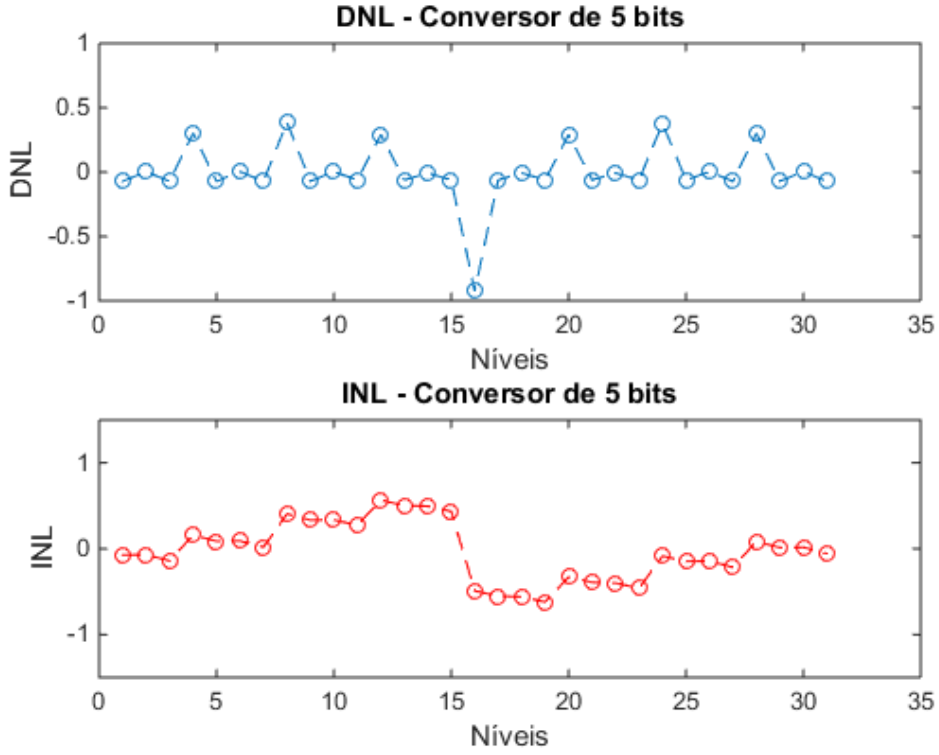

Figura 4.6. Análise de DNL e INL no conversor com mismatch nos capacitores

representa com mais fidelidade a resposta de entrada, entretanto, quanto mais bits o conversor possui, mais capacitores são necessários e mais erro de mismatch é adicionado.

4.2 CARACTERIZAÇÃO DA TECNOLOGIA

Antes de se desenvolver projetos de circuitos integrados é necessário obter um modelo que descreva o comportamento dos componentes disponíveis para o projeto. Um modelo pode ser descrito por equações matemáticas, circuitos ou tabelas. É importante entender que existe um grau de exatidão na representação de qualquer modelo. Portanto, para o desenvolvimento de projetos é importante escolher modelos que se adequem às necessidades e limitações determinadas por tal projeto.

Os transistores de efeito de campo (FET) são dispositivos não lineares de difícil caracterização, alguns modelos de transistores chegam a conter mais de 100 parâmetros. Realizar projetos usando modelos desta complexidade pode ser bastante desafiador, pois realizar cálculos com mais de 100 variáveis é no mínimo dispendioso em termos de tempo. Felizmente, existem ferra-

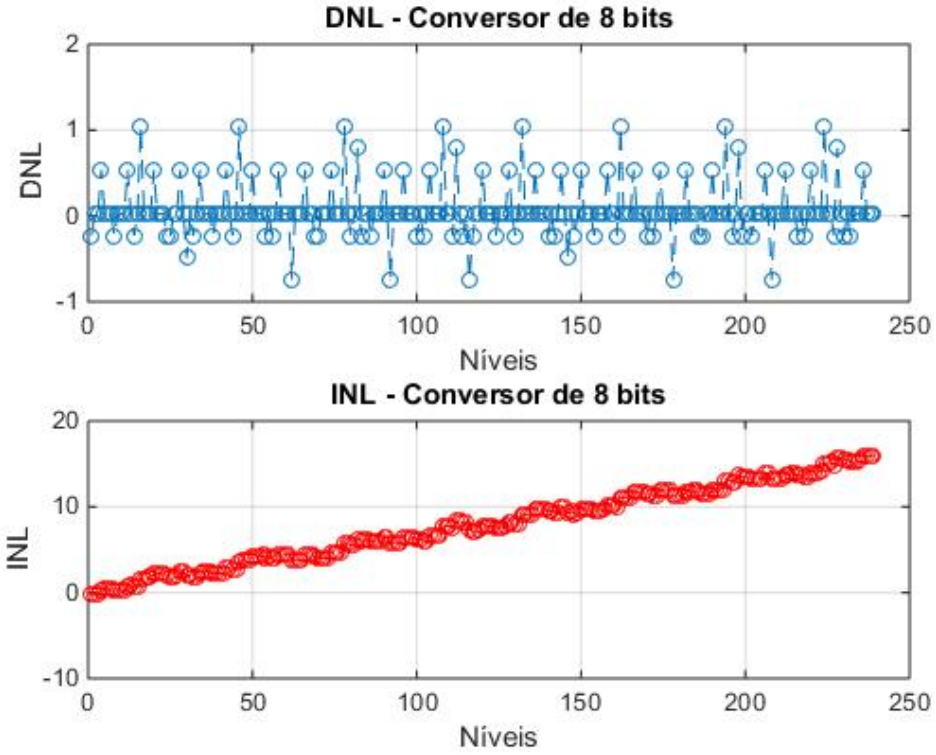

Figura 4.7. Simulação de conversor SAR de 8 bits com mismatch nos capacitores

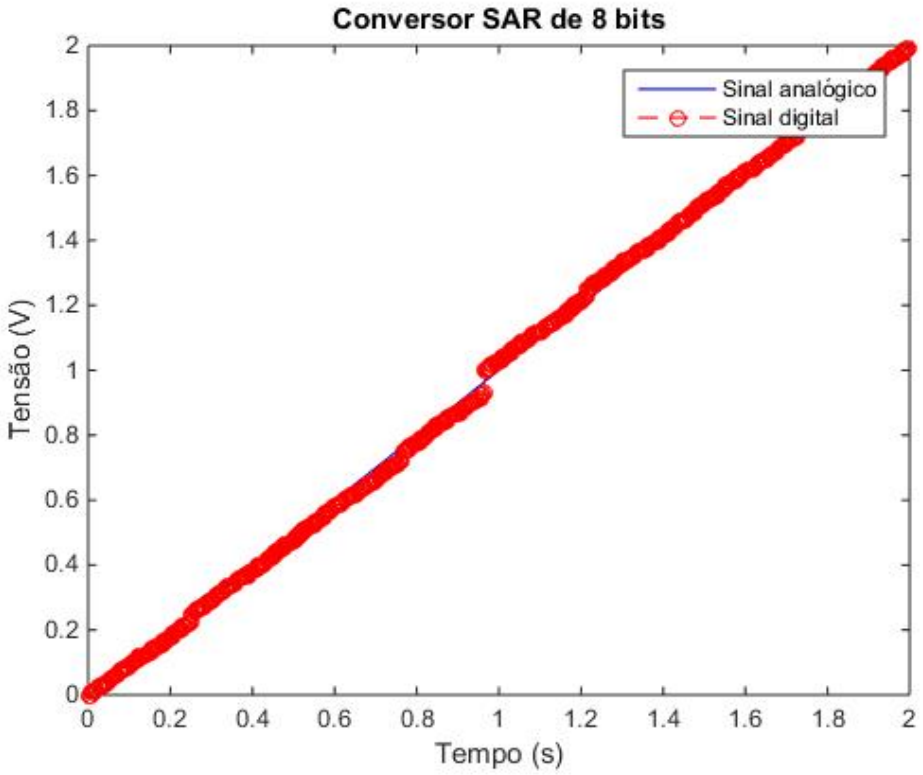

Figura 4.8. Análise de DNL e INL no conversor de 8 bits com mismatch nos capacitores

mentas de simulação desenvolvidas sob modelos de bastante precisão, permitindo ao projetista, simular o comportamento de circuitos complexos em segundos.

Ainda assim, mesmo com o uso de ferramentas de simulação pode ser muito trabalhoso projetar circuitos utilizando modelos muito complexos. Portanto, o projetista deve utilizar modelos que descrevem com menos precisão os componentes utilizados, mas que em compensação são mais simples e intuitivos. À vista disso, o projetista pode utilizar um modelo simples para inicialmente dimensionar os dispositivos de interesse e em seguida utilizar uma ferramenta de simulação para observar o comportamento obtido segundo um modelo mais preciso. Caso o projeto esteja divergindo um pouco do esperado, o projetista pode realizar um processo iterativo ao variar alguns parâmetros de circuito e simulá-los, até que encontre um resultado satisfatório.

Este projeto foi realizado utilizando a tecnologia XFAB XC018. Essa tecnologia é de 0.18μm construída em única camada de poly e em até seis camadas de metal. A biblioteca da tecnologia possui transistores MOS, capacitores metal-isolante-metal, transistores de junção bipolar e poly de alta resistência.

Usa-se um modelo de nível 1 para os transistores durante os primeiros cálculos de projeto. Para isso, deve-se determinar alguns parâmetros da tecnologia a ser utilizada na fabricação do projeto. Na documentação da tecnologia não são informados todos os parâmetros que caracterizam o modelo de nível 1 dos transistores. Portanto, será necessário fazer uma caracterização do modelo de nível 1 para a determinação destes parâmetros do modelo. A seguir, será mostrado o modelo utilizado, bem como seu procedimento de caracterização.

4.2.1 Modelo

As equações do modelo de nível 1 de um transistor MOS nas regiões de saturação e triodo, em inversão forte, são dadas respectivamente pelas equações 4.1 e 4.2 e 4.3.

$$i_{\mathrm{D}} = \frac{1}{2}K'\frac{W}{L}(v_{\mathrm{GS}} - V_{\mathrm{T}})^2(1 + \lambda v_{\mathrm{DS}}) \tag{4.1}$$

$$i_{\mathrm{D}} = K'\frac{W}{L}(v_{\mathrm{GS}} - V_{\mathrm{T}})v_{\mathrm{DS}} - \frac{{v_{\mathrm{DS}}}^2}{2} \tag{4.2}$$

$$V_{\mathrm{T}} = V_{\mathrm{T0}} + \gamma[\sqrt{2|\phi| + v_{\mathrm{SB}}} - \sqrt{2|\phi|}] \tag{4.3}$$

A princípio, os parâmetros de interesse são K', V_T e λ. Inicialmente, será mostrado como estimar o valor de K'. É importante saber que o valor de K' é diferente para as regiões de saturação e triodo ((ALLEN; HOLBERG, 2011)), portanto, este parâmetro precisa ser caracterizado separadamente para cada região.

Para calcular K' na região de saturação começa-se assumindo que o termo λv_{DS} é muito menor que 1 e portanto pode-se fazer a seguinte simplificação na equação 4.1:

$$i_{\mathrm{D}} = \frac{1}{2}K'\frac{W}{L}(v_{\mathrm{GS}} - V_{\mathrm{T}})^2 \tag{4.4}$$

Tirando a raiz quadrada dos dois lados da equação (4.4), obtém-se a equação (4.5) que possui uma característica linear em relação as varáveis $\sqrt{i_D}$ e v_{GS}. Considerando esta equação como uma reta, observa-se que o coeficiente angular da reta é dado por $\sqrt{\frac{K'W}{2L}}$. Logo, conhecendo os valores de W e L, consegue-se chegar em uma boa aproximação para K' na região de saturação neste modelo.

$$\sqrt{i_{\mathrm{D}}} = \sqrt{\frac{K'W}{2L}}v_{\mathrm{GS}} - \sqrt{\frac{K'W}{2L}}V_{\mathrm{T}} \tag{4.5}$$

Para a implementação do modelo de caracterização, realizou-se uma simulação DC no circuito ..., em que observou-se a corrente no transistor i_D ao variar a tensão na porta de 0 a 1.8 V. Os resultados da simulação foram exportados para o Matlab a fim de realizar o procedimento necessário para a obtenção do parâmetro de trasncondutância K'. No Matlab, foi tirado a raiz quadrada de i_D e então foi plotada a curva raiz de i_D versus v_{GS}, como mostra a figura 4.9.

Observa-se que o gráfico apresenta três comportamentos distintos, no início da curva o transistor encontra-se na região de corte, na região seguinte o transistor entra em saturação e por último o transistor entra na região de triodo. Dado que se está interessado na caracterização do transistor na saturação, realiza-se uma regressão linear da região de interesse, no script de Matlab é possível escolher os intervalos de regressão baseados na região de interesse.

Com os coeficientes da reta determinados pelo processo de regressão e com os valores de W e L usados na simulação realiza-se o cálculo de K'. O procedimento é análogo para transistores

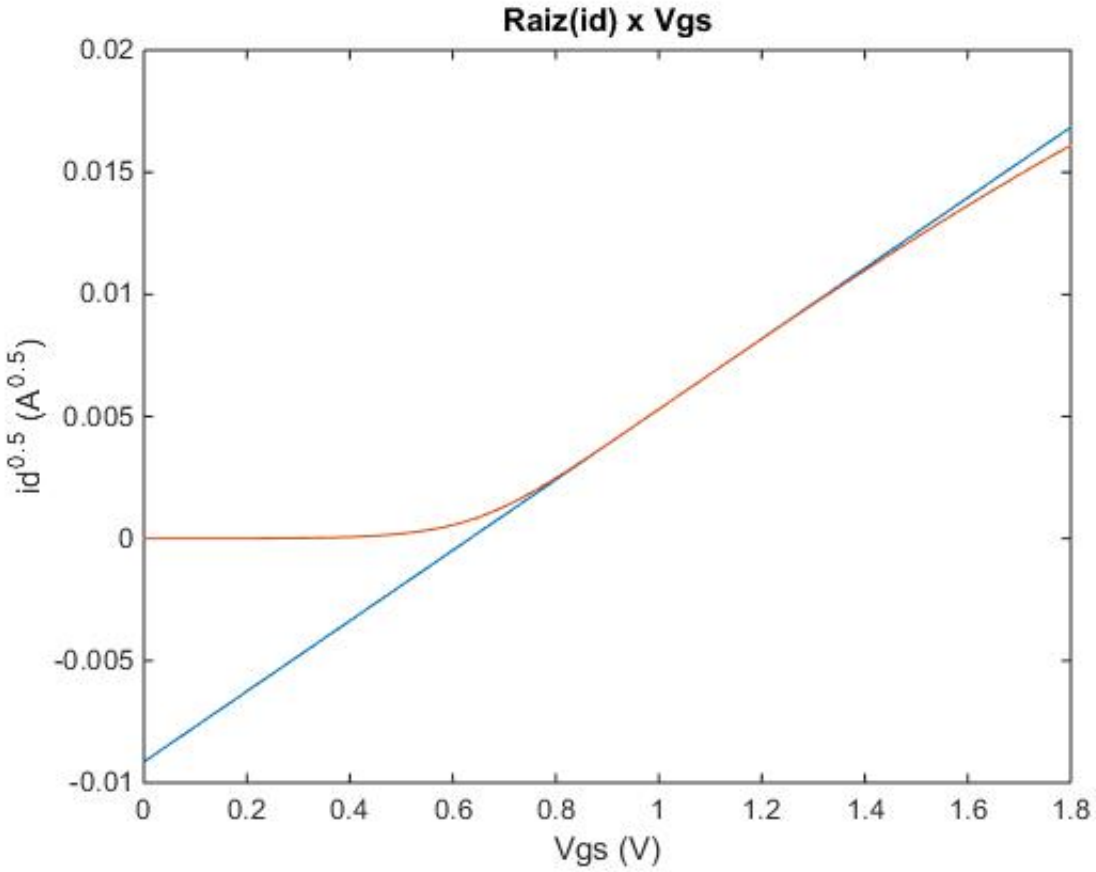

Figura 4.9. Identificação do parâmetro de transcondutância K' no transistor NMOS

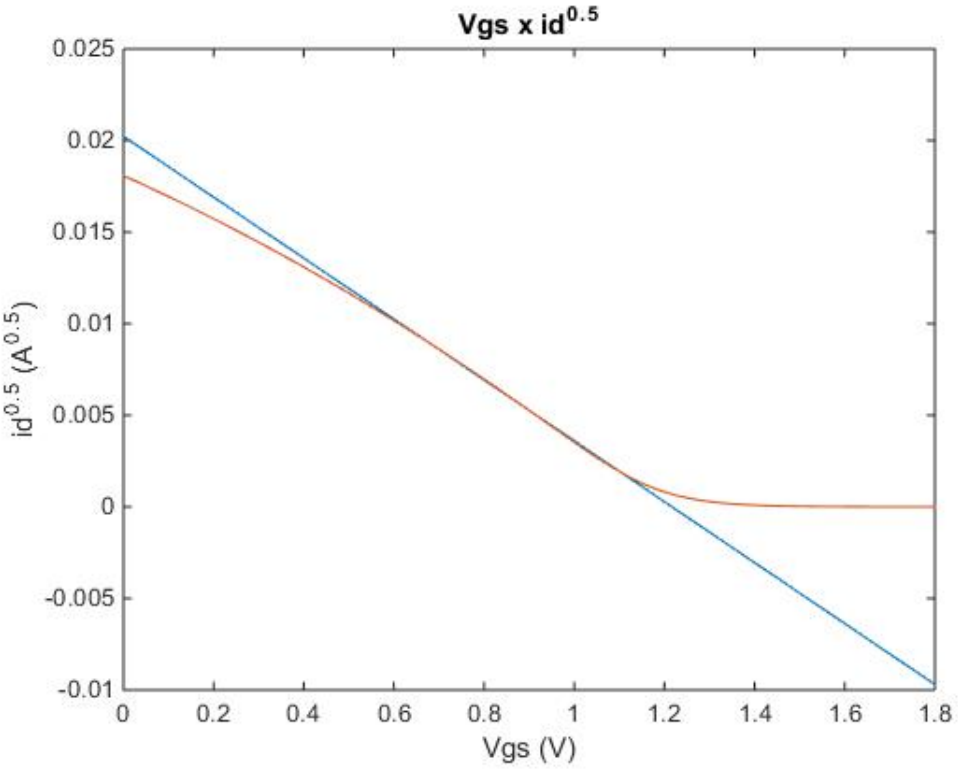

Figura 4.10. Identificação do parâmetro de transcondutância K' no transistor PMOS

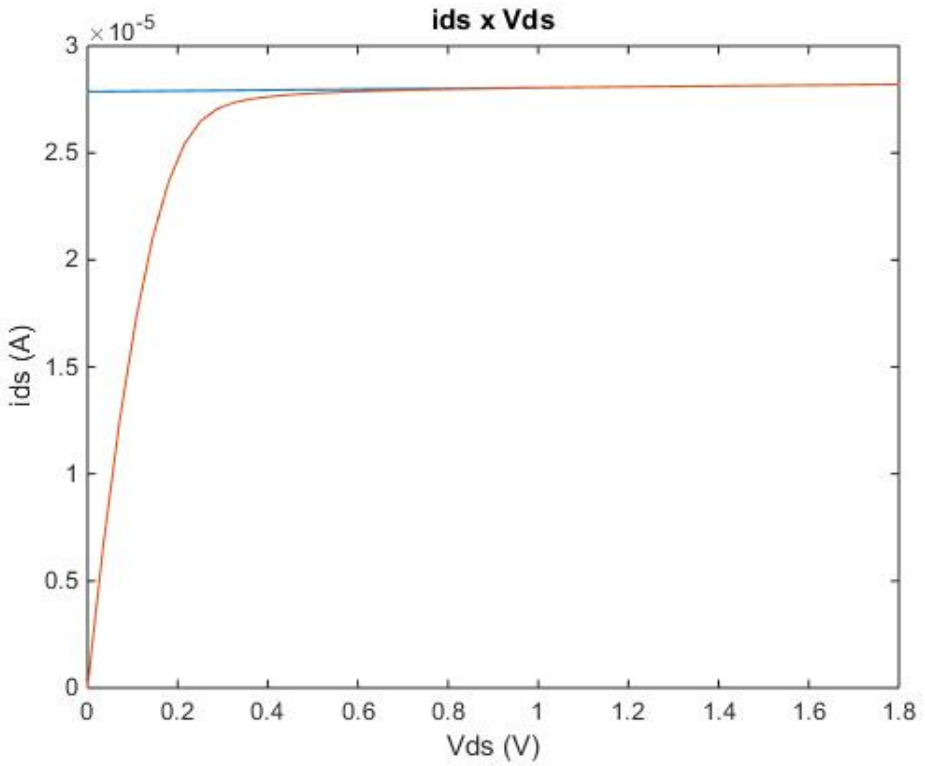

Figura 4.11. Identificação do parâmetro lambda do transistor NMOS

Transistores	K' ($\mu A/V^2$)	V_T (V)	λ (S)
ne	166.80	0.63	0.0068
pe	32.593	0.65	0.00019

Tabela 4.1. Caracterização dos transistores pe e ne com W/L = 5/2

nmos e pmos.

A partir da equação 4.5, nota-se que além do termo K', é possível determinar também a tensão de threshold V_T do transistor a partir da mesma curva, já que este termo multiplicado por um termo conhecido representa o coeficiente linear da reta.

Após a determinação dos parâmetros K' e V_T calcula-se o parâmetro de modulação de canal λ do transistor, que pode ser obtido através da curva característica do transistor de i_D por v_{DS}. Assim, realiza-se uma simulação DC variando a tensão entre dreno e fonte do transistor e mantendo uma tensão constante entre porta e fonte.

A partir da equação (4.1) sabe-se que o crescimento de i_D com v_{GS} é proporcional a λ. Tendo conhecimento da inclinação da reta característica do efeito de modulação de canal, calcula-se o λ usando a equação (4.1).

Para a tecnologia utilizada, chegou-se aos valores da tabela 4.1:

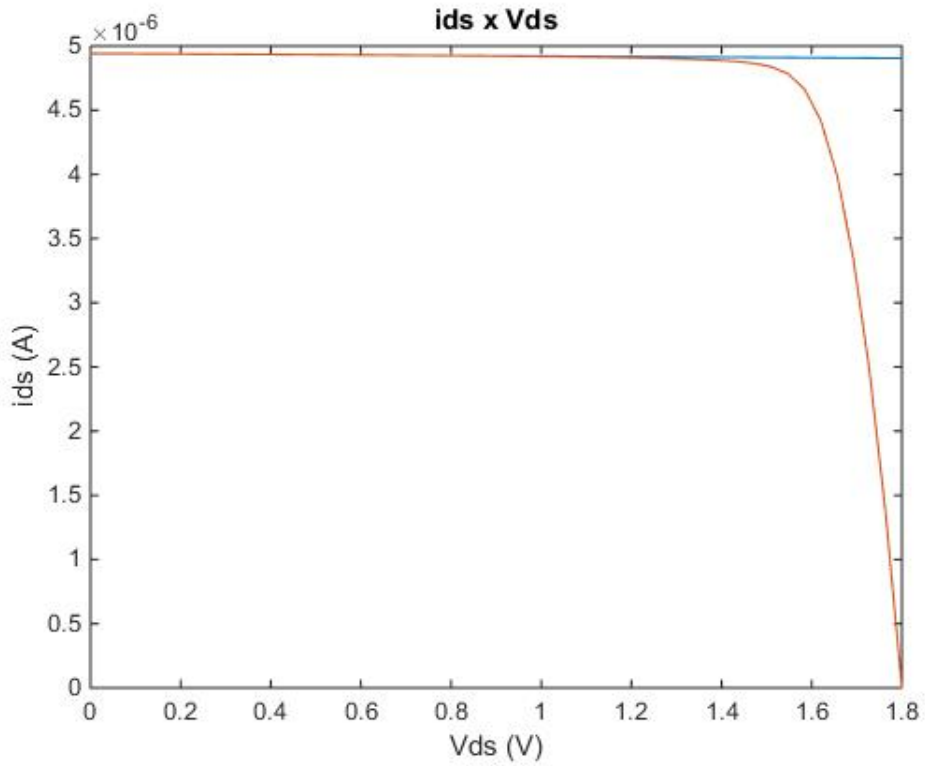

Figura 4.12. Identificação do parâmetro lambda do transistor PMOS

4.3 SAMPLE AND HOLD

Um circuito amostrador amostra um sinal analógico e o salva em um elemento de memória até o momento de sua próxima amostragem ((RAZAVI, 1995)). Este tipo de circuito é geralmente realizado por um transistor MOSFET e um capacitor, como mostra a figura (4.13). Este tipo de circuito não representa um circuito S/H (*Sample and Hold*), mas sim um circuito T/H (Track and Hold) que apenas segue o sinal de entrada e salva este sinal em um capacitor. O circuito S/H é implementado com um circuito T/H seguido de um *buffer* que dá ao circuito a capacidade de alimentar cargas capacitivas. Esta seção discutirá aspectos do S/H, mas com um foco principal no circuito de T/H. Assim, quando o termo S/H for mencionado no texto, deve-se entender o termo como o conjunto T/H e *buffer*.

Neste circuito, o transistor funciona como uma chave, operando nas regiões de corte e de triodo, o capacitor é o elemento de memória que armazena o sinal de entrada por um período determinado. O funcionamento deste tipo de circuito pode ser dividido em duas fases "sample"e "hold"que são controladas através do sinal V_{CLK} no terminal da porta do transistor.

Na fase "sample", o sinal de controle está em nível alto, o canal do transistor funciona

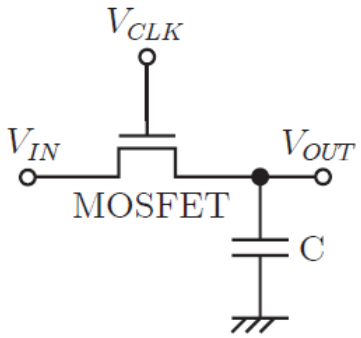

Figura 4.13. S/H com chave NMOS (BAKER, 2008)

idealmente como um curto circuito e o sinal de saída segue o sinal de entrada. Nesta fase, é interessante que o transistor possua uma tensão relativamente alta na porta para que a resistência de canal seja baixa fazendo com que a queda de tensão entre V_{in} e V_{out} seja desprezível. Durante a fase "hold", o sinal de controle está em nível baixo, o canal do transistor funciona como um circuito aberto e o sinal de saída é dado pelo sinal armazenado no capacitor.

O circuito da figura (4.13) é bem simples e intuitivo, entretanto, existem algumas limitações que o tornam ineficiente. Primeiro, para que a chave funcione durante a fase "sample"é necessário que a tensão entre porta e fonte seja maior que a tensão de threshold do transistor para garantir que o transistor não entre em corte. Isso implica na seguinte relação para uma chave tipo n:

$$Vin_{\text{máx}} = V_{\text{CLK}} - V_{\text{T}} \tag{4.6}$$

Dessa forma, quando o transistor é do tipo NMOS, a excursão máxima do sinal de entrada será limitada pela tensão de threshold do transistor. Para uma chave do tipo PMOS o mesmo acontece, mas a excursão mínima da entrada fica limitada pelo threshold. Uma possível solução é usar uma chave NMOS em paralelo com uma PMOS, dessa forma, uma chave atua na região que a outra chave não é capaz de atuar, aumentando a faixa de operação da chave sample and hold. Essa é a chave CMOS e deve ser implementada usando sinais complementares de clock para o seu correto funcionamento.

Além do aumento da excursão de entrada, a chave CMOS possui a vantagem, em relação as chaves NMOS e PMOS, de possuir uma resistência de canal que depende muito menos da tensão de entrada (RAZAVI, 1995).

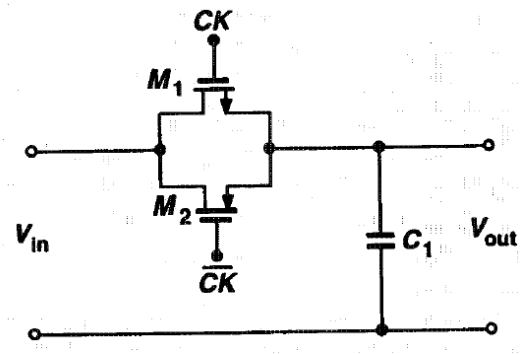

Figura 4.14. T/H CMOS, (RAZAVI, 1995)

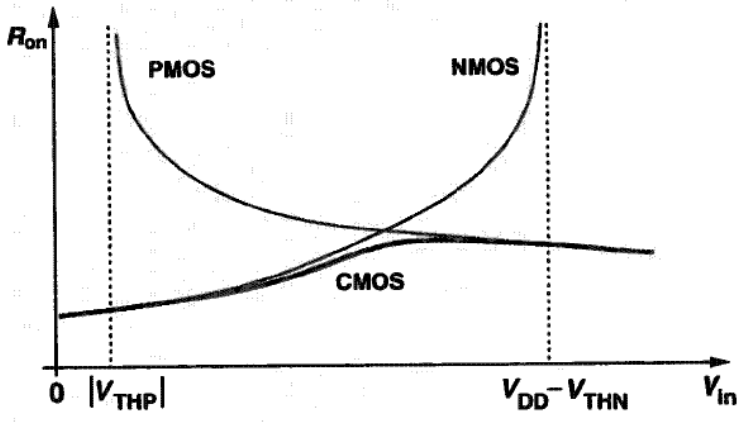

Figura 4.15. Comparação entre as resistências de canal das chaves CMOS, NMOS e PMOS, (RAZAVI, 1995)

A resistência de canal do transistor NMOS operando na região de triodo é expressada pela equação 4.9, em que μ é a mobilidade do elétron no canal, C_{ox} é a capacitância porta óxido por unidade de área, W e L são a largura e comprimento efetivos do dispositivo.

$$R_{on} = \frac{1}{\mu C_{ox} \frac{W}{L}(V_{GS} - V_T)} \tag{4.7}$$

Apesar de apresentar menor variação na resistência de canal, ao longo da excursão do sinal de entrada, a chave CMOS ainda apresenta uma variação considerável. Devido a problemas de linearidade, a chave CMOS é inadequada para aplicações que exigem precisões maiores do que 6 bits (RAZAVI, 2015).

Em circuitos que exigem maior precisão do sample and hold, o *bootstrapping* é uma técnica interessante a ser utilizada. Como foi discutido anteriormente, um dos fatores que degradam a resposta do sample and hold é a variação da resistência de canal provocada por V_{in}. Essa variação, em sua maior parte, é provocada pela alteração de V_{GS} ao variar a tensão de entrada. Portanto, para mitigar a variação de R_{on} pode-se buscar um circuito capaz de manter V_{GS}

constante ao variar V_{in}. Essa é exatamente a função do circuito de *bootstrapping*.

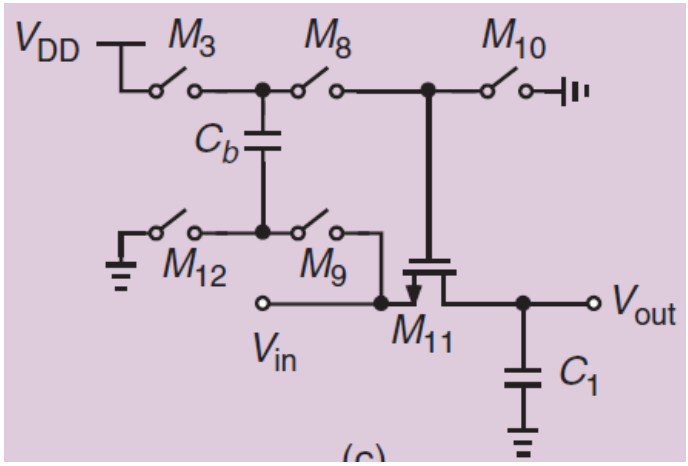

Figura 4.16. *Bootstrapped T/H* (RAZAVI, 2015)

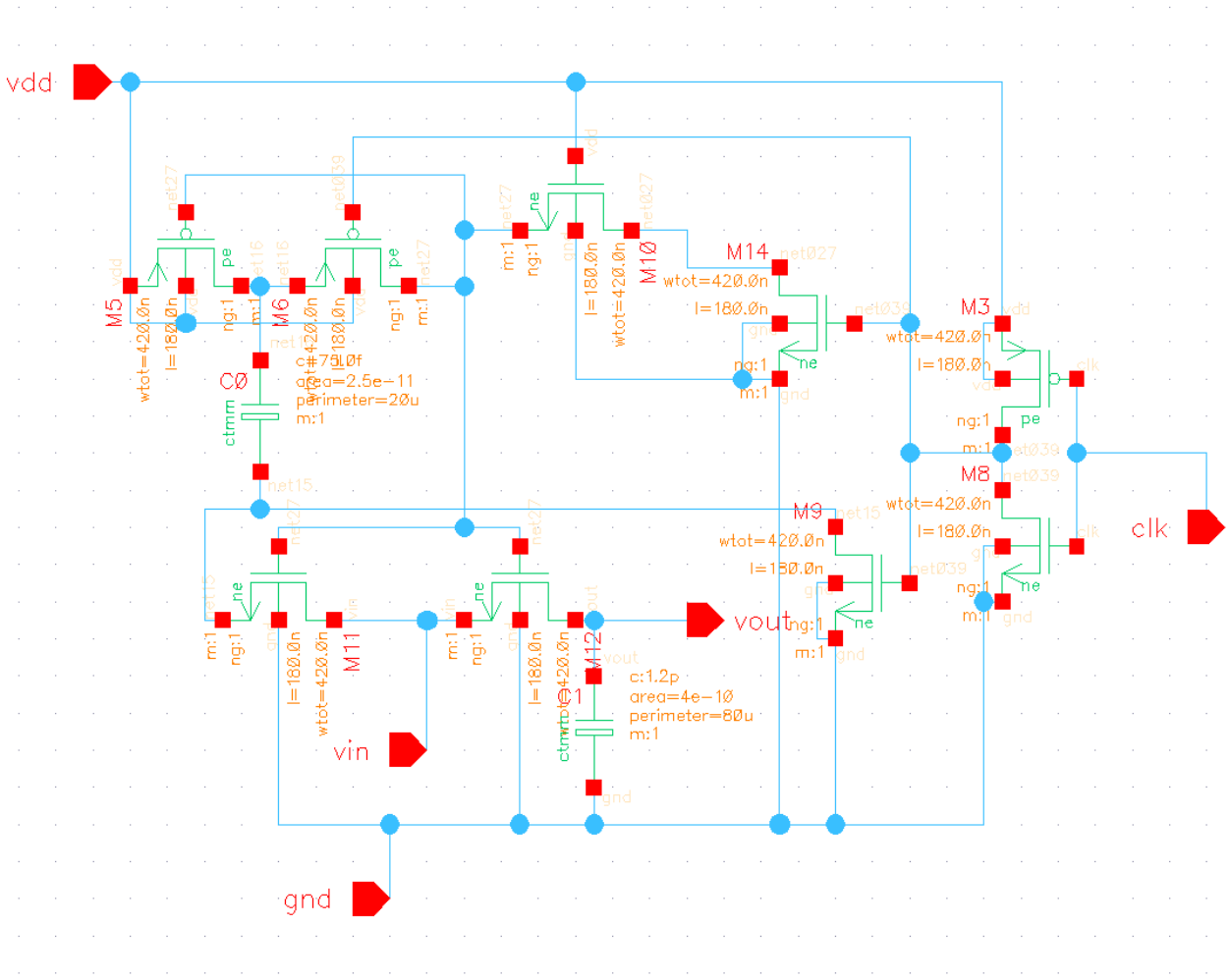

Figura 4.17. *Bootstrapped T/H*

Uma alternativa para manter-se constante o V_{GS} da chave é posicionar um capacitor carre-

gado entre os terminais de porta e fonte do transistor chave. Desta maneira, a medida que o sinal na fonte do transistor variar, o sinal na porta varia a mesma proporção, mantendo V_{GS} constante. A figura 4.16 mostra o esquema de um sample and hold *bootstrapped*.

Na figura 4.16, durante a fase hold, o capacitor C_b deve ser carregado com V_{DD} entre seus terminais e estar desacoplado do transistor M_{11}. Portanto, M_3 e M_{12} estarão ligados, M_8 e M_9 estarão desligados. Além disso, o transistor M_{11}, que representa a principal chave do circuito, deverá estar desligado, logo, M_{10} estará ligado, conectando o sinal de terra até a porta de M_{11}. Na fase sample, o capacitor, carregado com uma tensão igual a V_{DD}, é acoplado entre a porta e fonte do transistor M_{11}. Assim sendo, os transistores M_8, M_9 e M_{11} estarão ligados e os outros transistores desligados.

Utilizou-se o circuito da figura para a implementação do sample and hold. Foi adicionado a chave M_{14} para proteção do transistor M_{10}, pois a tensão no ponto X pode chegar a valores maiores do que a tensão de ruptura do transistor M_{10}. Desta forma, o M_{14} provoca uma queda de tensão protegendo M_{10}.

4.4 COMPARADOR

Os comparadores são bastante utilizados em topologias de conversores A/D. De certa forma, o comparador ideal pode ser considerado um conversor de 1 bit. O comparador possui duas entradas v_p (positiva) e v_m (negativa) e uma saída v_{out}. Os sinais de entrada são comparados, caso v_p seja maior que v_m, o sinal de saída se torna VDD, caso o contrário aconteça, o sinal de saída vai para VSS. A figura 4.18 mostra o símbolo de um comparador.

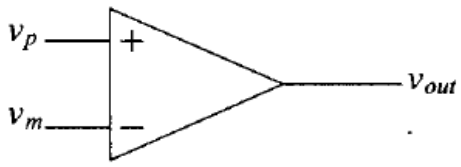

Figura 4.18. Símbolo do Comparador

Quanto menor for a diferença entre os sinais de entrada, mais difícil é para o comparador escolher o bit de saída. Uma importante característica do comparador para aplicações de conversão A/D é a resolução, a menor diferença entre os sinais de entrada que é capaz de alterar

o valor de saída do comparador. Neste projeto, baseou-se inicialmente em uma resolução para conversor A/D de 12 bits dentro de uma faixa de 1.5 V, portanto, fez-se o seguinte cálculo:

$$V_{\text{mín}} = \frac{1.5}{2^{12}} = 366\mu V \tag{4.8}$$

$$t_{\text{p}} = \frac{\Delta V}{SR} \tag{4.9}$$

Desta forma, projetou-se o comparador para uma resolução que alcançasse pelo menos uma precisão igual a 366 μ V. Para garantir essa resolução projetou-se um comparador com capacidade para comparar sinais com 150 μV de diferença. Para alcançar essa resolução, o comparador pode comparar as duas entradas utilizando um par diferencial e em seguida amplificar a diferença entre os sinais para que se alcance valores altos o suficiente para caracterizar o bit de saída como 0 ou 1.

Assim, utilizou-se a topologia em que consiste em um par diferencial com poço de corrente e um espelho de corrente como carga, um segundo estágio sendo um amplificador fonte de corrente e um terceiro estágio sendo um inversor.

O ganho necessário para cumprir a especificação da resolução pode ser calculado a partir da equação 4.10. Este ganho pode ser dividido entre os três estágios do comparador, ou seja, pode-se projetar o primeiro estágio para um ganho igual a 20, o segundo estágio para um ganho igual a 30 e o terceiro para um ganho igual a 20, dessa forma, obtém-se o ganho de 12000 desejado para o sistema completo.

Outra especificação importante é a frequência de operação do comparador, que é determinada pela frequência de amostragem do conversor, o número de bits e o algoritmo do conversor tipo SAR. Na seção sobre a máquina de estados, a tabela 4.2 mostra o tempo de cada procedimento no algoritmo. Nesta tabela, o comparador possui um ciclo de clock para realizar a comparação, que na tabela está representado com 20 ns de período. Portanto, o clock deve obter um tempo de resposta menor que 20 ns para satisfazer as especificações.

$$A_{\text{v}} = \frac{V_{\text{OH}} - V_{\text{OL}}}{V_{\text{min}}} = \frac{1.8V}{150\mu V} = 12000 \tag{4.10}$$

Baseando-se nos tempos de conversão da tabela 4.2, definiu-se que o comparador poderá ter

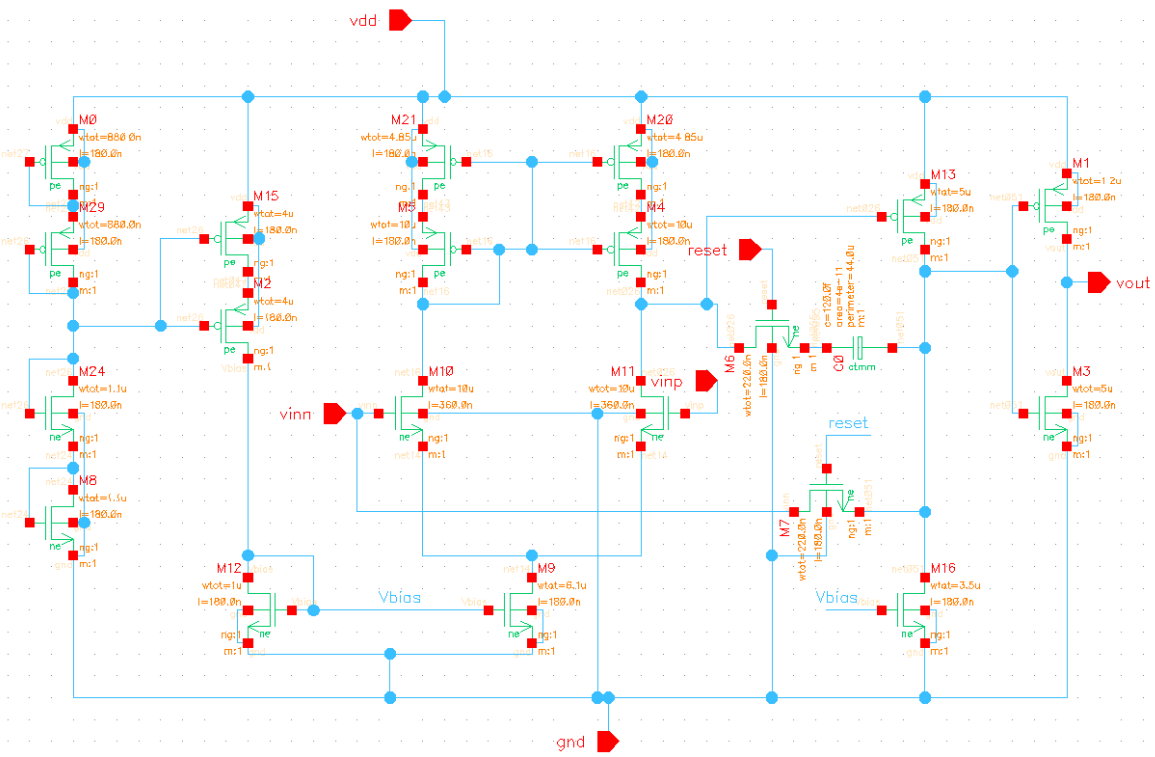

Figura 4.19. Esquemático do comparador

um tempo de resposta de 10ns. O tempo de resposta do circuito está relacionada ao *slew rate* e a composição dos polos do comparador, que possui um polo por estágio e estes dependerão da capacitância dos transistores utilizados e consequentemente de suas dimensões físicas. A partir deste tempo de resposta foi possível determinar o *slew rate* do comparador, esse dado foi usado para determinar a corrente desejada em cada estágio do conversor.

O tamanho dos transistores influencia no modelo a ser utilizado para dimensionar o circuito. Como estão sendo usados transistores com o menor tamanho de comprimento possível para a tecnologia, as equações que modelam o comportamento dos transistores são mais complexas do que as do modelo proposto na seção de caracterização do transistor. Entretanto, iniciou-se o projeto utilizando o modelo para transistores maiores e em seguida corrigiu-se os valores fazendo simulações de ponto de operação e transiente.

No algoritmo do ADC proposto, antes da fase de comparação existe um momento em que os capacitores do DAC são carregados com uma tensão igual a (Vcm - Vin), em que Vcm é a tensão de modo comum na entrada positiva do comparador. Para carregar os capacitores do DAC com Vcm seria necessário a utilização de um buffer. Todavia, decidiu-se aproveitar os

primeiros dois estágios do comparador para realizar a função do buffer necessário. Deste modo, adicionou-se um sinal de controle no comparador para transformá-lo em um buffer de ganho unitário chaveando seu circuito de compensação.

Essa ideia de transformar o comparador em buffer, além de evitar a inserção de um buffer a mais no circuito, acaba realizando também uma forma de cancelamento de offset no comparador.

Após o dimensionamento dos transistores deve-se avaliar a performance do comparador através de simulações DC e transiente. Assim, é possível avaliar o comportamento estático e dinâmico do comparador.

4.5 CONVERSOR DIGITAL/ANALÓGICO

Existem várias arquiteturas de conversores D/A e cada uma possui vantagens específicas que justificam seus usos em diferentes aplicações. A primeira diferença que pode se notar é o tipo de código digital utilizado. É comum que os conversores utilizem o código binário tradicional ou o código de Gray, pois esse possui a vantagem de variar apenas um dígito entre um nível de quantização e outro, o que em certas aplicações pode levar a redução do consumo de energia do conversor. Como neste projeto, está sendo realizado um conversor de aplicação genérica, escolheu-se utilizar o código binário tradicional para o ADC. Assim, deve-se também utilizar o mesmo código para o DAC.

Entre as arquiteturas de DAC mais populares estão DAC R-2R construído com resistores, o DAC condutor de corrente e o DAC baseado no carregamento de capacitores. Escolheu-se utilizar a arquitetura de DAC baseada no carregamento de capacitores, já que essa é a arquitetura mais utilizada para ADCs do tipo SAR por apresentarem baixo consumo, serem rápidas e obterem uma boa resolução.

O DAC baseado no carregamento de capacitores consiste em um *array* de capacitores, como o da figura 4.20, em que cada capacitor representa um bit, com exceção do capacitor *dummy* de valor unitário C. Os outros capacitores possuem o dobro de capacitância do bit anterior. A capacitância total do circuito será dada por $2^{N}C$, em que N é o número de bits do DAC.

Os terminais superiores do capacitor estão todos ligados no mesmo nó, já os terminais inferiores são ligados em V_{REF} ou ground dependendo do valor do bit ligado a cada capacitor.

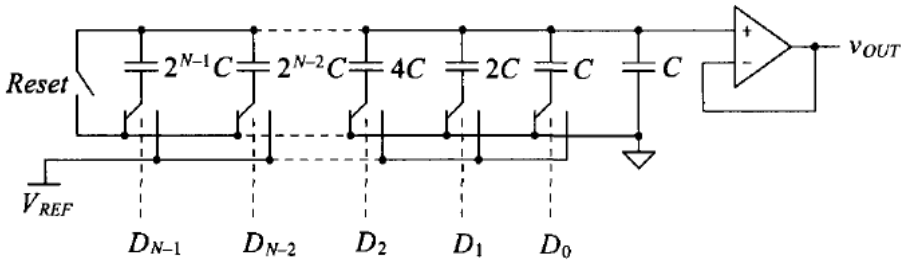

Figura 4.20. DAC baseado no carregamento de capacitores (BAKER, 2008)

Dessa forma, o circuito da figura 4.20 pode ser simplificado como mostra a figura 4.21.

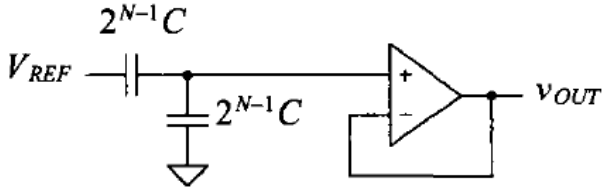

Figura 4.21. DAC simplificado (BAKER, 2008)

Como é apresentado na figura, a tensão nó conectado ao terminal positivo do buffer de saída é resultado do divisor de tensão capacitivo formado pelos capacitores ligados em V_{REF} e os capacitores ligados em ground. É importante notar que o capacitor *dummy* estará sempre conectado ao ground para que a conversão aconteça de forma correta. Para exemplificar o funcionamento do DAC a equação mostra a tensão de saída de um conversor de 4 bits tendo como entrada a palavra binária 1100.

$$v_{\text{out}} = V_{\text{REF}} \times \frac{(8+4)C}{(1+1+2+4+8)C} = V_{\text{REF}} \times \frac{3}{4} \tag{4.11}$$

A grande desvantagem deste tipo de configuração de DAC é que o valor dos capacitores crescem exponencialmente com o número de bits, o que provoca um aumento muito grande no consumo e na área ocupada pelo chip. Uma forma de reduzir o tamanho dos capacitores é utilizar uma arquitetura semelhante chamada de Split Array Capacitor. Esta arquitetura funciona com o mesmo algoritmo da anterior e os capacitores não crescem exponencialmente com o número de bits.

No *split array*, existe um capacitor que divide o array de capacitores em dois sub arrays, o MSB e o LSB. Desta forma, o crescimento dos capacitores só é exponencial dentro do sub array. O valor do capacitor de atenuação é calculado a partir da equação 4.12, em que o numerador

corresponde a soma dos capacitores do sub array LSB e o denominador corresponde a soma dos capacitores do sub array MSB.

$$C_{\mathrm{at}} = \frac{\sum C_{\mathrm{LSB}}}{\sum C_{\mathrm{MSB}}} \times C \tag{4.12}$$

A grande desvantagem desta arquitetura de DAC é a linearidade. O valor do capacitor de atenuação requer uma precisão de casas decimais em relação a capacitância unitária. Portanto, ao fazer a escolha entre uma dessas arquiteturas deve ser levado em consideração os limites de precisão garantidos pela tecnologia, pois dependendo do número de bits do conversor e do mismatch de capacitores o uso do split array pode causar problemas de linearidade que o torne uma solução inviável.

No ADC SAR, o DAC costuma ser o bloco que mais consome energia e essa energia gasta vai ser proporcional a capacitância total do DAC. Desse modo, a arquitetura split array é significativamente mais eficiente em termos de consumo. Comparando a capacitância total dos DACs, baseados no carregamento de capacitores, tradicional e split array, a capacitância do split array é consideravelmente menor. Para o mesmo valor de capacitância unitária C_{un}, em um conversor de 12 bits, o split array possui capacitância aproximadamente igual a $65C_{un}$, já o DAC tradicional possui capacitância igual a $4096C_{u}$.

Utilizou-se o modelo de referência, que é baseado na arquitetura tradicional do DAC de carregamento de capacitores, em Matlab para prever os efeitos do mismatch de capacitores na linearidade do conversor. Nas referências da tecnologia é dado uma equação para a modelagem do desvio padrão da variação dos capacitores. No script de Matlab, implementou-se o erro de mismatch de capacitores utilizando uma distribuição normal com desvio padrão calculado a partir do modelo fornecido. A partir dos resultados obtidos escolheu-se usar uma capacitância unitária igual 15 fF.

4.6 MÁQUINA DE ESTADOS

A máquina de estados é responsável por controlar todos os blocos do circuito, fazendo com que o algoritmo do conversor seja executado de forma correta. A máquina de estados do conversor deste projeto está sendo realizada pelo aluno Pedro Tolentino. A máquina de estados

deste trabalho consiste em um circuito digital que foi sintetizado a partir de um código VHDL utilizando as ferramentas de projeto do Cadence.

Apesar de não ter sido desenvolvida neste trabalho, é importante ter conhecimento do funcionamento da máquina de estados para o desenvolvimento de especificações dos blocos analógicos desenvolvidos neste projeto.

Em um alto nível de abstração, podemos dividir a máquina de estados em duas fases principais, *sample* e *hold*. Durante a fase de *sample*, o S/H estará seguindo o sinal de entrada e este será salvo no capacitor de saída do S/H, nesta fase, a função da máquina de estados é apenas habilitar o S/H.

Temporização do processo de conversão	
Fase	**Tempo (ns)**
. Sample	200
. Hold	800
- Carregamento de capacitores	200
- Chaveamento do DAC	40
- Comparador	20

Tabela 4.2. Temporização do processo de conversão

A fase de *hold* é dividida em algumas etapas, a primeira é carregar os capacitores do DAC com uma tensão igual a Vcm-Vin. Para isso, o máquina de estados manda um sinal de controle para o DAC e outro para o comparador para que esse entre em modo buffer de ganho unitário. Em seguida, a máquina de estados deve enviar para o DAC a primeira sequência binária a ser testada. Por último, a máquina de estados observa a resposta do comparador e decide a próxima palavra a ser testada. Após testar todos os bits, a máquina volta para o seu estado inicial.

Como o ADC possui taxa de amostragem de até 1M amostras/s, o processo inteiro tem que ocorrer dentro do período de 1 μs. É importante notar que cada estado da máquina tem um período diferente. Para evitar a necessidade de usar múltiplos sinais de clock, optou-se por utilizar apenas um sinal e usar a contagem de ciclos deste sinal para simular os outros sinais de períodos superiores. O clock escolhido foi o de período igual a 20ns. A distribuição do tempo de conversão de uma amostra está apresentada na tabela 4.2.

CAPÍTULO 5

RESULTADOS E DISCUSSÃO

5.1 SAMPLE AND HOLD

Como já foi mencionado no capítulo anterior, a função do S/H é amostrar o sinal analógico de entrada do ADC e salvá-lo em um elemento de memória. Existem várias topologias de S/H, uma das mais simples consiste apenas em uma chave CMOS seguida de um capacitor, outra um pouco mais sofisticada é a chave *bootstrapped*, ambas podem ser utilizadas em projetos de conversores A/D.

Visto as especificações propostas neste projeto, foram feitos testes com ambas as chaves visando identificar qual delas seria mais apropriada. Obviamente, espera-se um melhor desempenho do S/H *bootstrapped*, entretanto, o S/H com chave CMOS é menor e mais simples, portanto, se esta topologia atende as especificações, não existe necessidade de usar outra chave mais sofisticada e mais cara em termos de área.

Como já dito no capítulo anterior, a vantagem da chave *bootstrapped* sobre a chave CMOS é manter a resistência de canal aproximadamente constante ao longo de toda a faixa de tensão. Essa vantagem aumenta a linearidade da chave *bootstrapped* em comparação com o circuito S/H realizado com chave CMOS.

Foram realizadas algumas simulações com o S/H com chave CMOS com o propósito de avaliar o nível de distorção da chave ao longo da faixa de tensão de entrada do conversor. Para isso, escolheu-se 3 pontos ao longo dessa faixa e observou-se a resposta do S/H para uma entrada senoidal variando 1 LSB em torno de cada ponto. As figuras 5.2, 5.3 e 5.4 mostram as simulações realizadas, em que é aplicado na entrada do circuito de teste um sinal senoidal com amplitude de 300 μV e offset igual a 0.1 V, 0.9 V e 1.4 V respectivamente. A partir das figuras, observa-se a diferença entre os sinais de entrada e saída aplicados. O resultado esperado para essas simulações é que Vin seja igual a Vout durante a fase de *sample* e que a diferença entre

estes sinais aumente gradualmente na fase de *hold*.

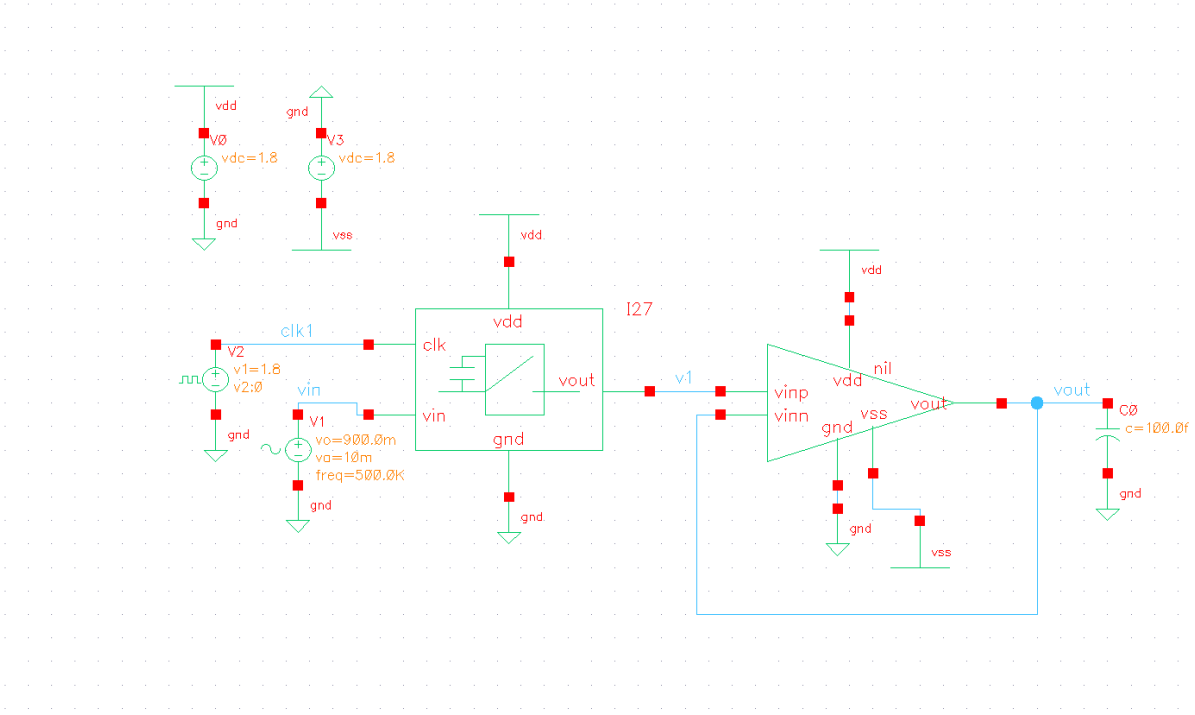

Figura 5.1. Circuito para teste da chave CMOS

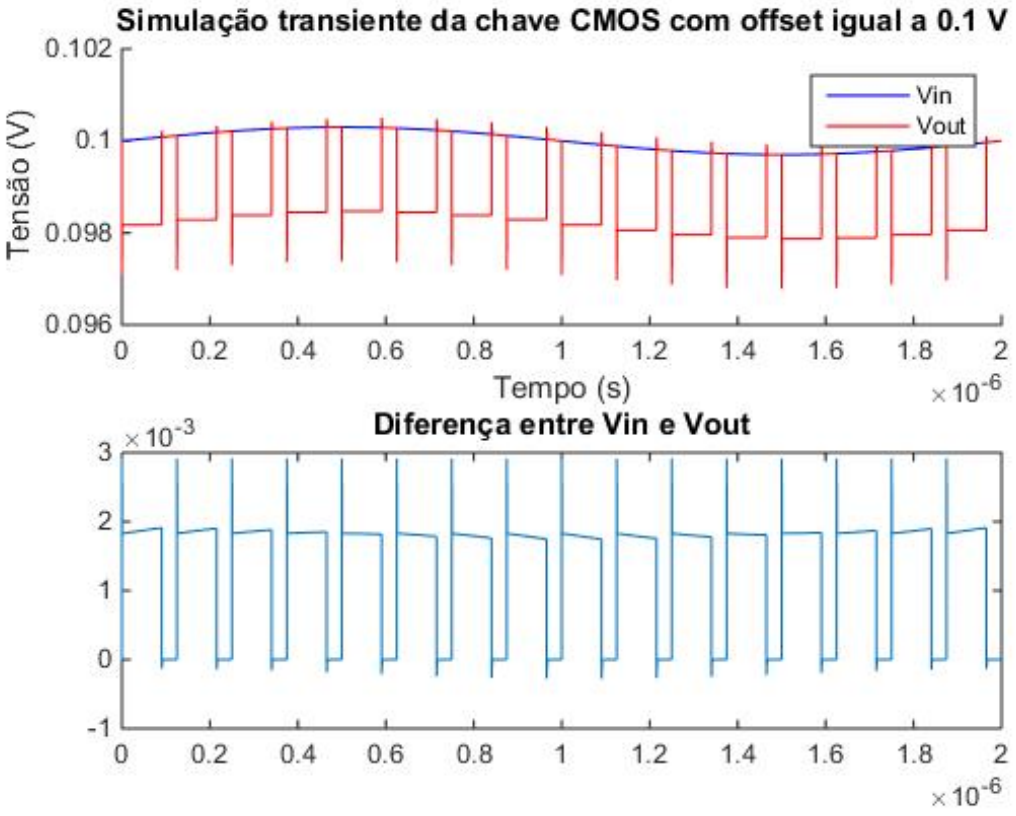

Figura 5.2. Simulação transiente do S/H em torno de 0.1 V

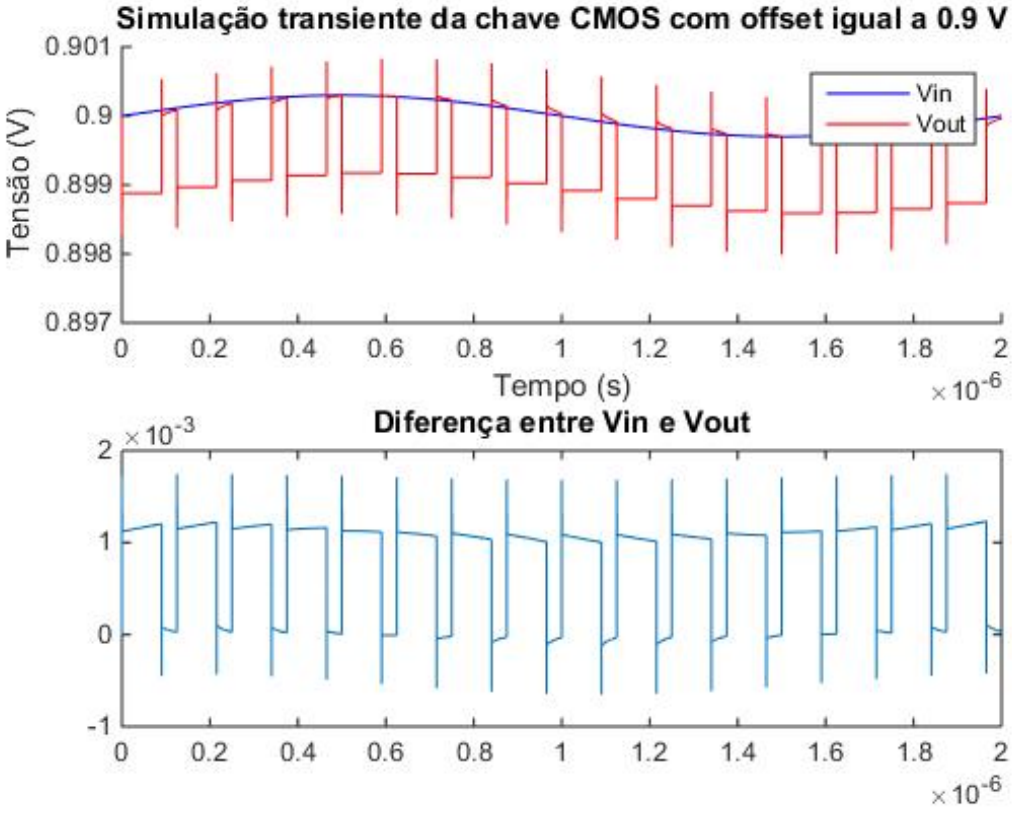

Figura 5.3. Simulação transiente do S/H em torno de 0.9 V

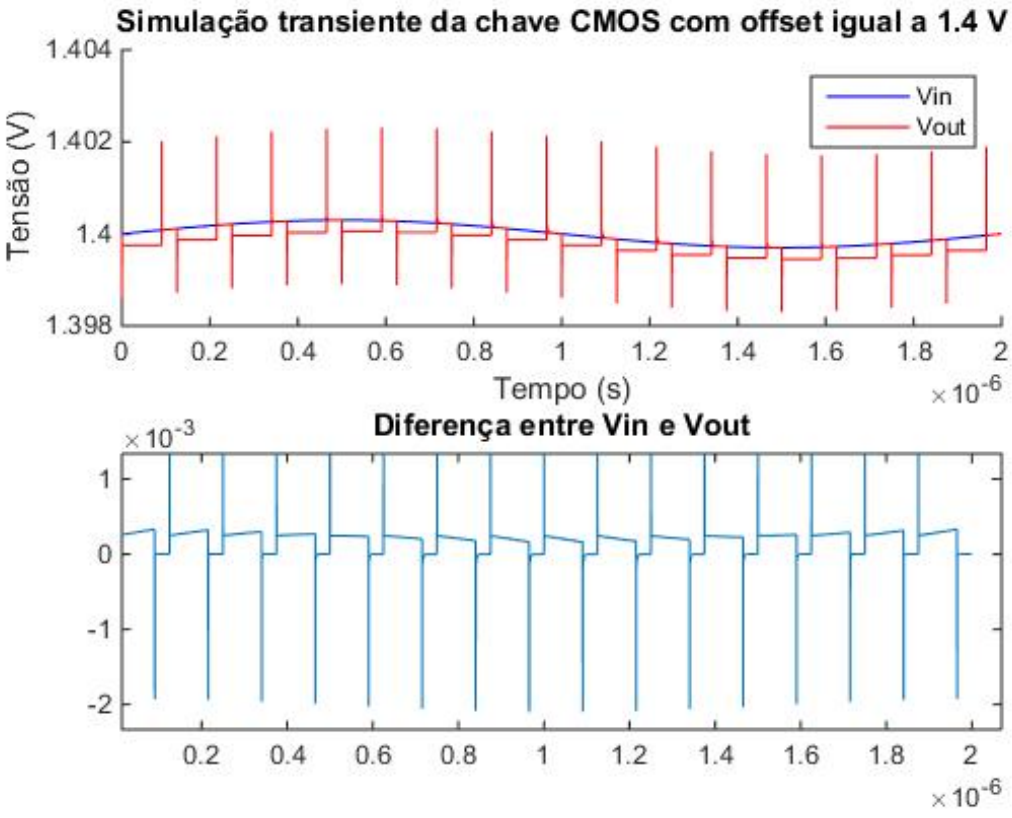

Figura 5.4. Simulação transiente do S/H em torno de 1.4 V

A partir das figuras 5.2, 5.3 e 5.4, observa-se que os resultados não corresponderam ao esperado, isso porque as simulações apresentaram um offset na saída do circuito durante a fase de *hold*. Esse offset é bem maior do que a gradual diferença entre os sinais Vin e Vout idealmente esperada, por isso, não foi possível observar na imagem essa diferença esperada. Além disso, as simulações mostram que o offset varia ao longo da faixa dinâmica de entrada do S/H, resultando em distorção.

Em busca de se corrigir este problema de offset, as chaves foram redimensionadas e simuladas. Nas figuras 5.5 e 5.6 observou-se que com o ajuste do tamanho dos transistores consegue-se variar o offset indesejado, entretanto, mesmo com os ajustes feitos o offset continua variando ao longo da faixa dinâmica do S/H causando distorção. Caso essa distorção fosse menor que o LSB do conversor, o S/H com chave CMOS poderia ser utilizado no projeto, contudo, a distorção causada está na casa de unidades de milivolts, o que inviabiliza o seu uso em um conversor de 10 bits.

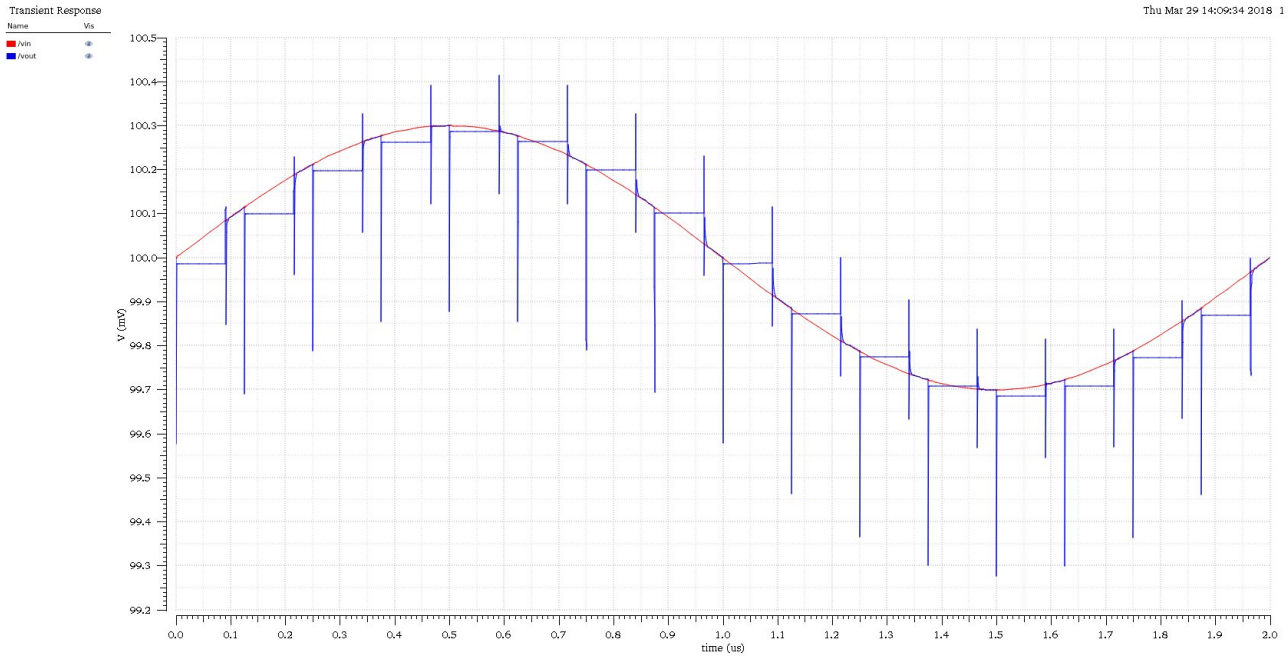

Figura 5.5. Simulação transiente do S/H em torno de 0.1 V

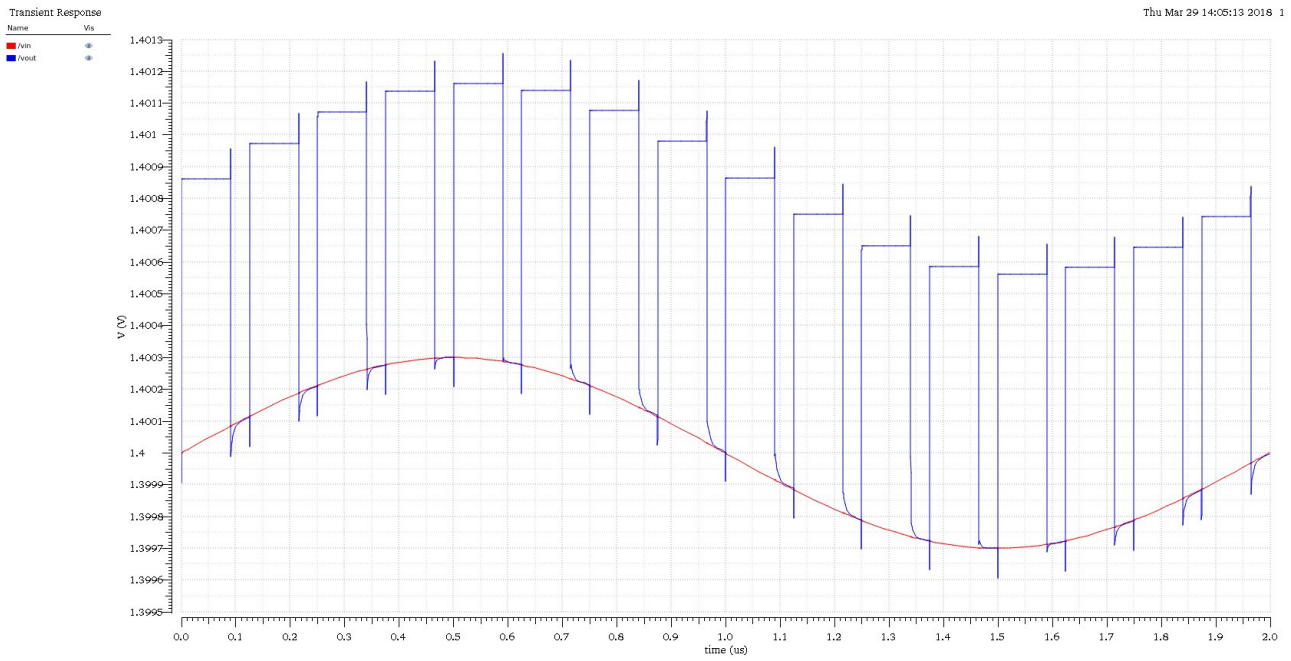

Figura 5.6. Simulação transiente do S/H em torno de 1.4 V

Dado os resultados do S/H com chave CMOS, inicia-se os testes com o S/H *bootstrapped* fazendo uma simulação transiente com Vin sendo um sinal senoidal de amplitude menor que 1 LSB do conversor. A figura 5.7 mostra o resultado da simulação.

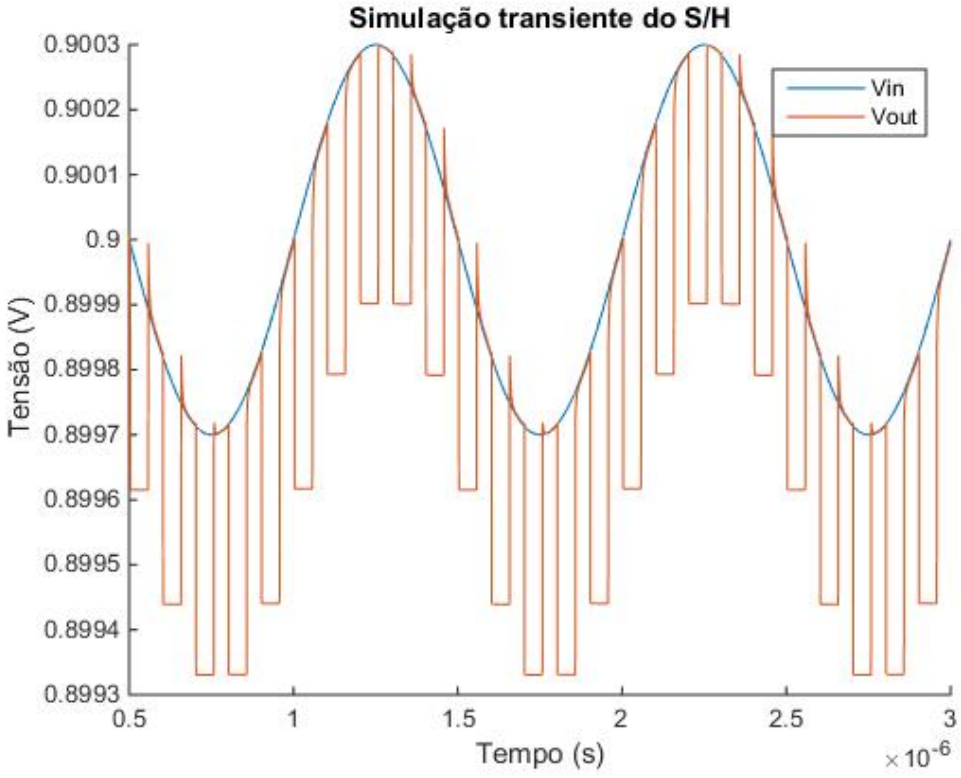

Figura 5.7. Simulação transiente do S/H bootstrapped

Observa-se que na fase *sample* o circuito segue a saída com uma performance adequada. Entretanto, durante a fase hold o circuito apresenta um offset em relação a resposta esperada, assim como apresentou o S/H com chave CMOS.

O fenômeno causador de offset nestes circuitos é conhecido como injeção de carga e acontece devido ao acoplamento capacitivo entre a porta do transistor e os terminais de dreno e fonte. Durante a fase *sample*, a tensão na porta do transistor é aproximadamente igual a Vdd e existe uma forma de divisor capacitivo entre o acoplamento porta dreno e o capacitor da saída do S/H. Quando o circuito entra na fase *hold*, o canal do transistor é fechado e a carga acumulada entre porta e dreno só tem um caminho, que é o do nó de saída ligado ao capacitor do S/H. Como a chave é construída com um NMOS, um dispositivo baseado na corrente de elétrons, são elétrons ao invés de buracos que migram do acoplamento porta dreno para o capacitor de saída gerando uma queda de tensão. Felizmente, a topologia *bootstrapped* faz com que a diferença de tensão entre porta e dreno seja sempre Vdd, independente da tensão de entrada do circuito, fazendo com que a carga acumulada no acoplamento não varie com a tensão de entrada. Assim, a injeção de carga não provoca distorção no S/H *bootstrapped*, apenas offset.

$$Q = C \times V \tag{5.1}$$

Ainda assim, é interessante se livrar deste offset ou pelo menos torná-lo invisível na resolução do conversor. A partir da equação do capacitor 5.1, sabe-se que a capacitância é inversamente proporcional a variação de tensão no capacitor, portanto, aumentando o valor da capacitância de saída espera-se reduzir o offset causado pela injeção de carga. Para avaliar essa hipótese, foram realizadas algumas simulações com diferentes valores de capacitância de saída. A figura 5.8 mostra a comparação entre as simulações feitas com diferentes capacitâncias de saída.

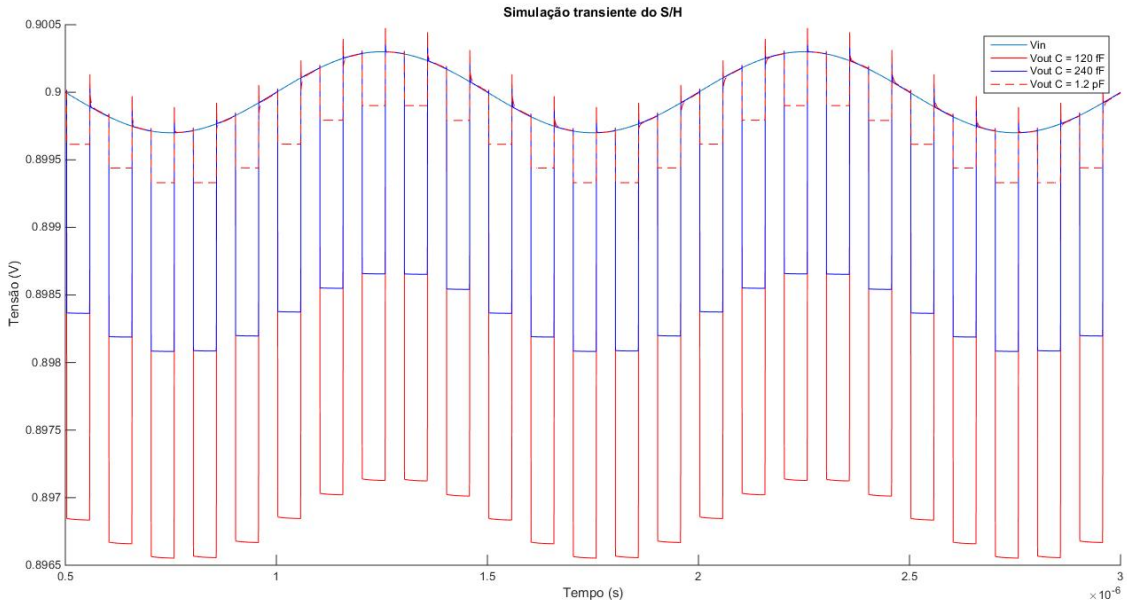

Figura 5.8. Avaliação do valor do capacitor de saída sobre o valor do *offset* de saída

A partir das simulações, confirma-se a hipótese de que o aumento da capacitância de saída reduz o offset na saída do sample and hold. Entretanto, existem outros fatores que estão sendo influenciados também, o principal deles é a constante de tempo do circuito, que está aumentando junto com a capacitância. Uma possível solução seria aumentar o tamanho do transistor, usado como chave, para reduzir a resistência do canal do transistor e assim reduzir a constante de tempo.

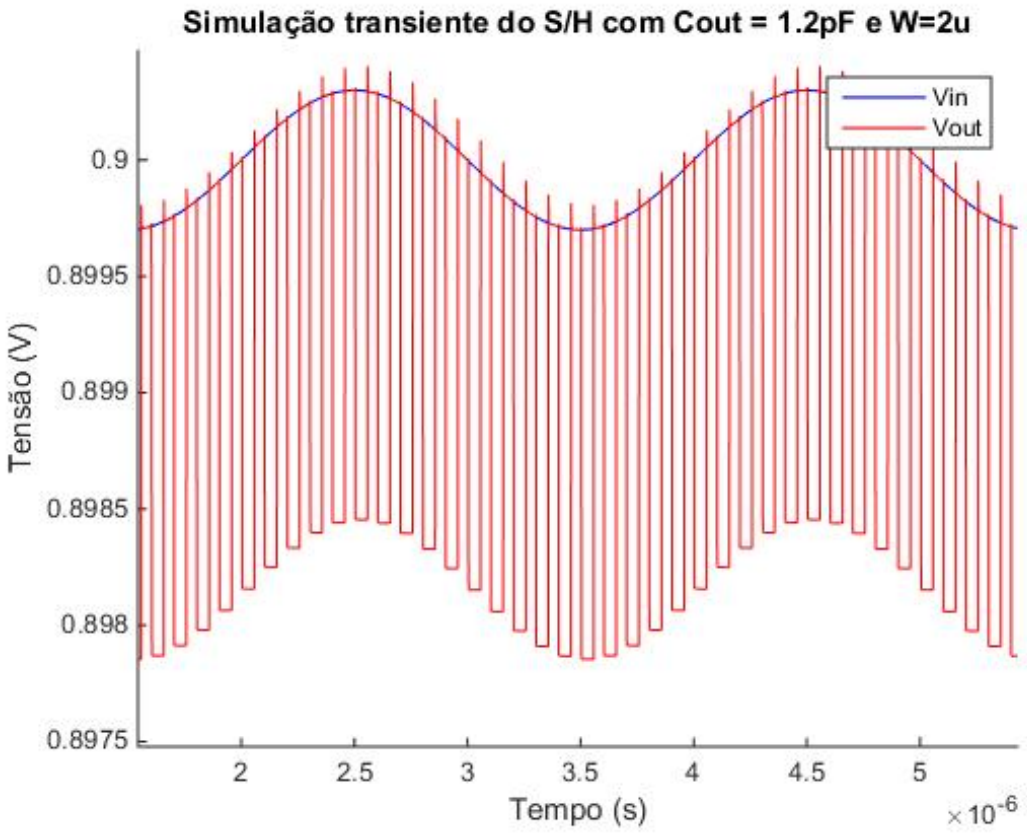

Figura 5.9. Avaliação da variação da relação W/L no offset

Algumas simulações foram realizadas para avaliar a opção de aumentar a relação W/L do transistor visando uma redução da constante de tempo do sistema. A figura 5.9 mostra uma das simulações realizadas. Analisando os gráficos, chegou-se a conclusão de que aumentando-se o W do transistor aumenta-se também a capacitância do acoplamento porta dreno do transistor. Dessa forma, mais carga é acumulada no acoplamento e maior fica o offset indesejado. Além disso, como a capacitância de saída está predominando sobre a constante de tempo do circuito, a variação da resistência de canal foi insignificante na constante de tempo. Portanto, aumentar a relação W/L não foi uma boa escolha.

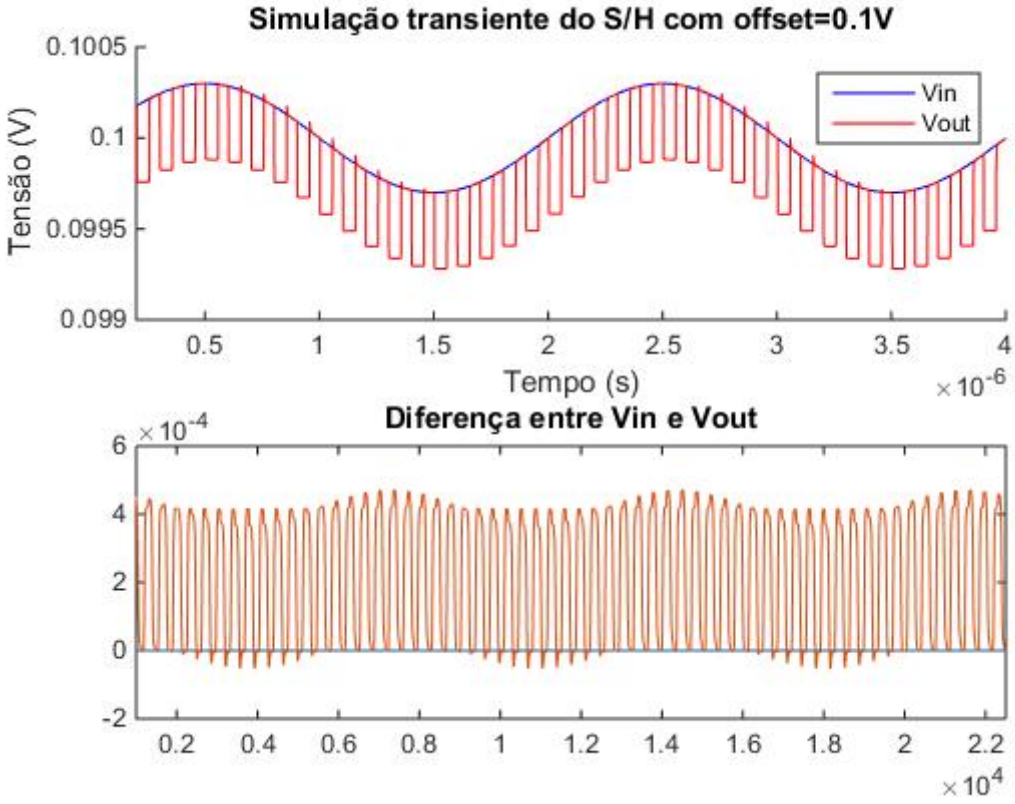

Figura 5.10. Análise do offset para Vin = 0.1 V

Dado as análises realizadas, optou-se por utilizar um capacitor de 1.2pF na saída do sample and hold e utilizar como chave um transistor com W pequeno e L pequeno. Desta forma, manteve-se uma boa fidelidade ao sinal de entrada, mas ainda com um offset um pouco menor que 1 LSB. Ainda foram realizadas algumas simulações para avaliar se o offset, como esperado, se mantém constante ao longo da faixa de tensão usada pelo conversor. Obteve-se um bom resultado de 0 V à 1.5 V. As figuras 5.10, 5.11, 5.12 mostram algumas simulações feitas para avaliar a variação de offset. Nestas figuras, observa-se que o offset varia entre 400 e 200 μV, o que satisfaz as especificações para um conversor de 10 bits.

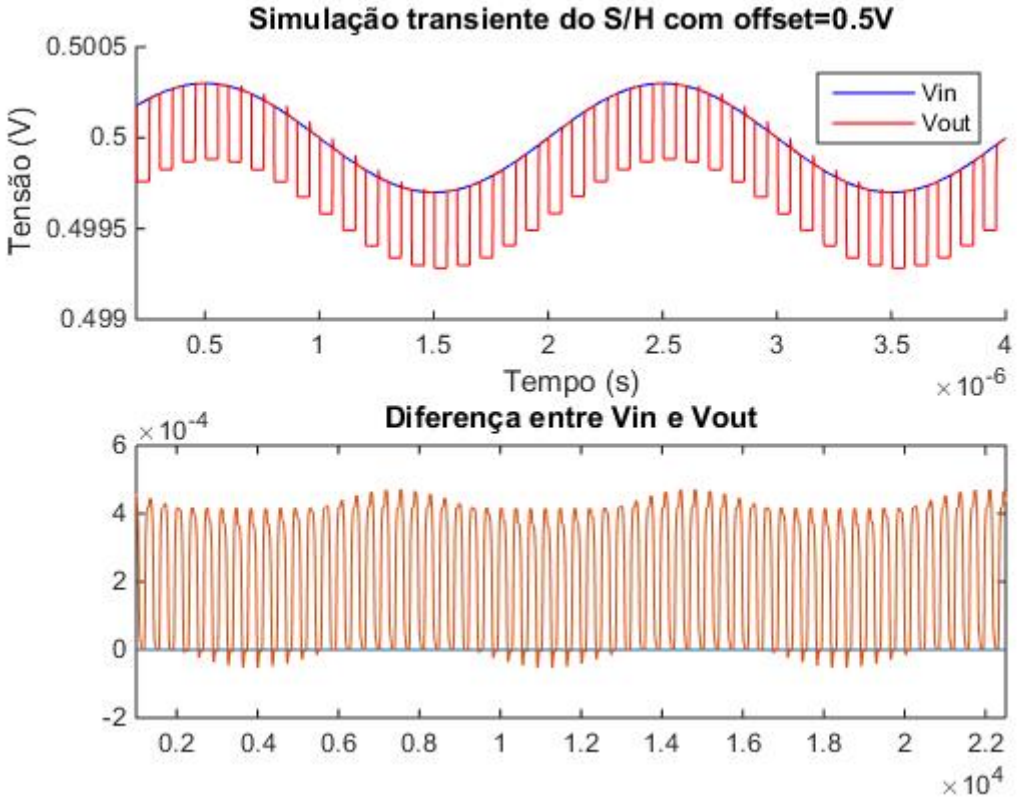

Figura 5.11. Análise do offset para Vin = 0.5 V

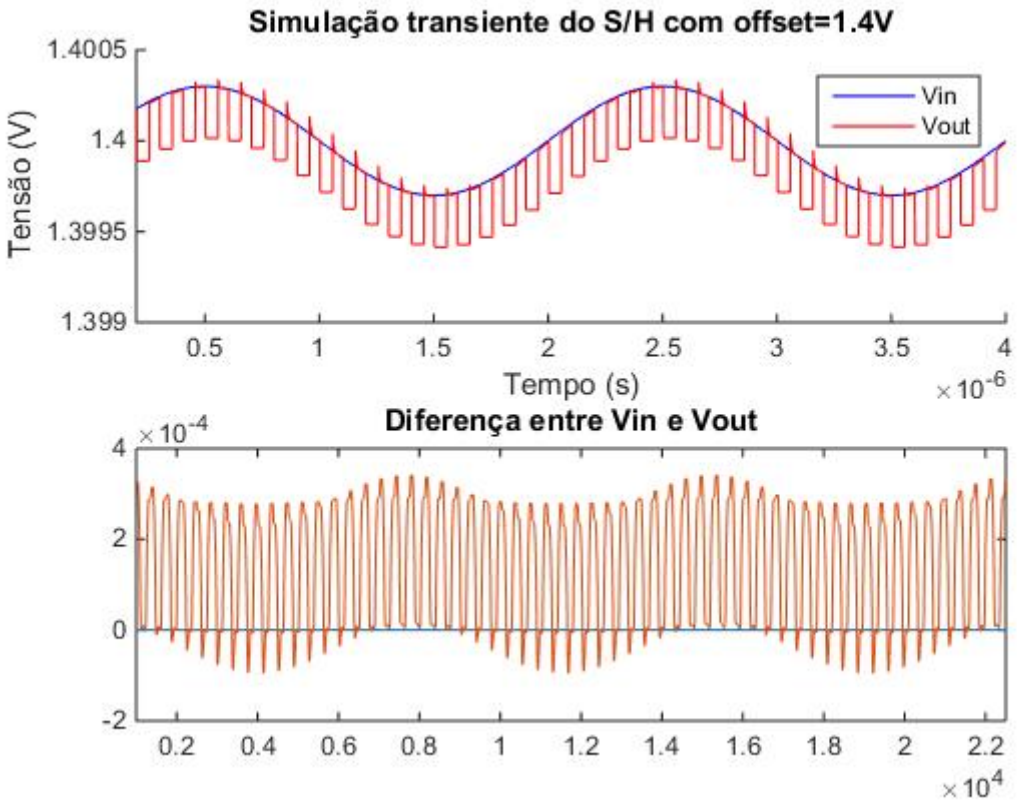

Figura 5.12. Análise do offset para Vin = 1.4 V

Ainda foi realizada uma simulação transiente para observar a resposta do conversor ao longo de toda a escala de tensão do conversor. A figura 5.13 mostra o resultado da simulação com tensão de entrada passando por toda a escala de tensão.

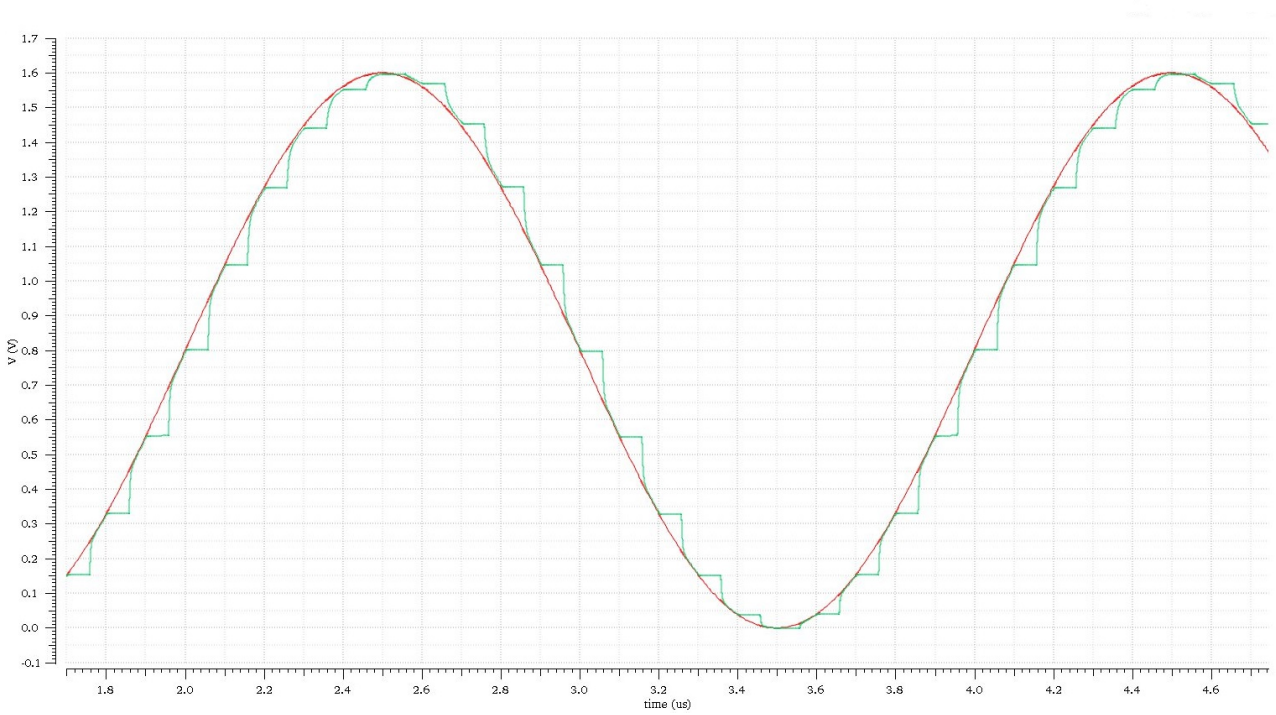

Figura 5.13. Simulação do Sample and Hold com tensão de entrada passando por toda a escala de tensão.

Por último, foi realizado uma simulação de corners no sample and hold. No total foram realizadas 225 simulações de corners. Na tecnologia existem cinco tipos de corner para transistores e três para capacitores. Além dos corners que simulam variações nas dimensões dos componentes do circuito, testou-se também a robustez do circuito com relação a temperatura e variação em Vdd. O circuito apresentou bom desempenho dentro da faixa de temperatura de 0 °C a 80 °C.

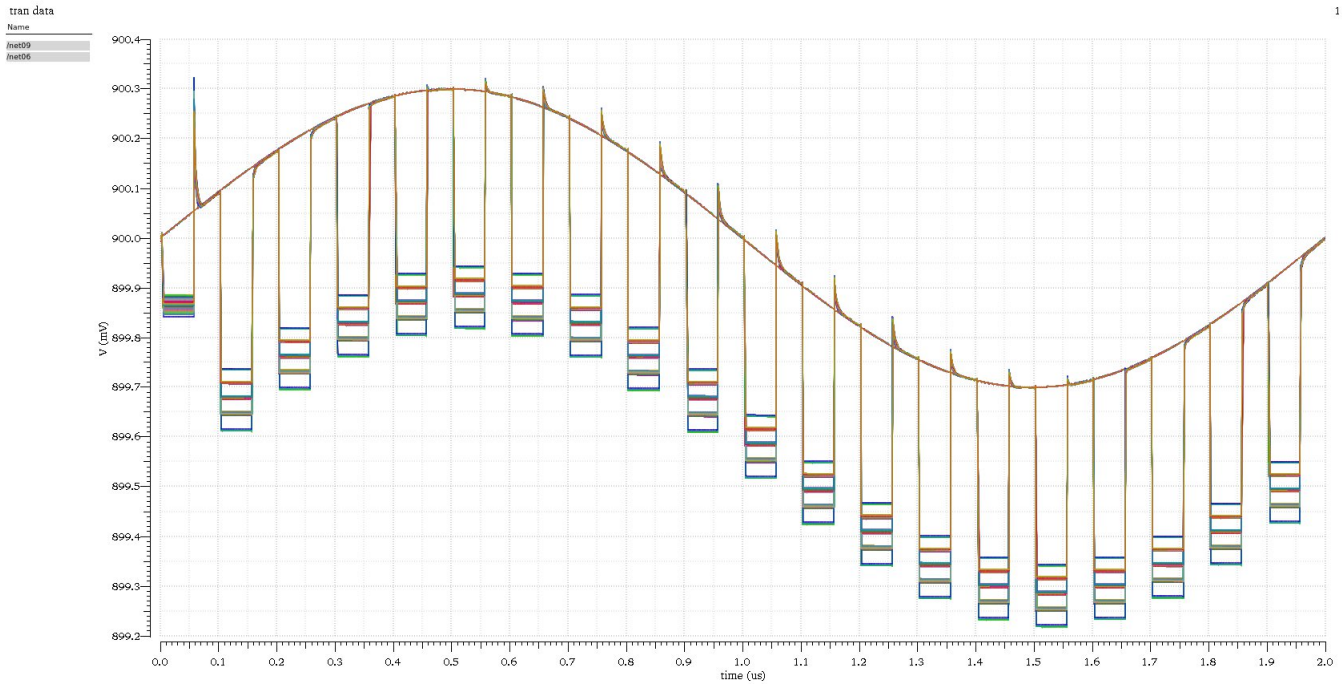

Figura 5.14. *Corners do Sample and Hold*

5.2 COMPARADOR

O comparador é um dos blocos mais importantes para o desempenho do conversor. Por isso deve-se garantir com bastante rigor que ele atenda às especificações esperadas. Primeiro, avalia-se o desempenho deste bloco em uma simulação transiente em que V_+ se mantém próximo à 1 V e V_- sai de 300 μV abaixo de 1 V para 300 μV acima de 1 V. Nesta simulação deseja-se observar se o comparador consegue perceber essa diferença na entrada do circuito e fornecer o bit adequado na saída do circuito. Também observa-se o tempo de reação do comparador para essa variação.

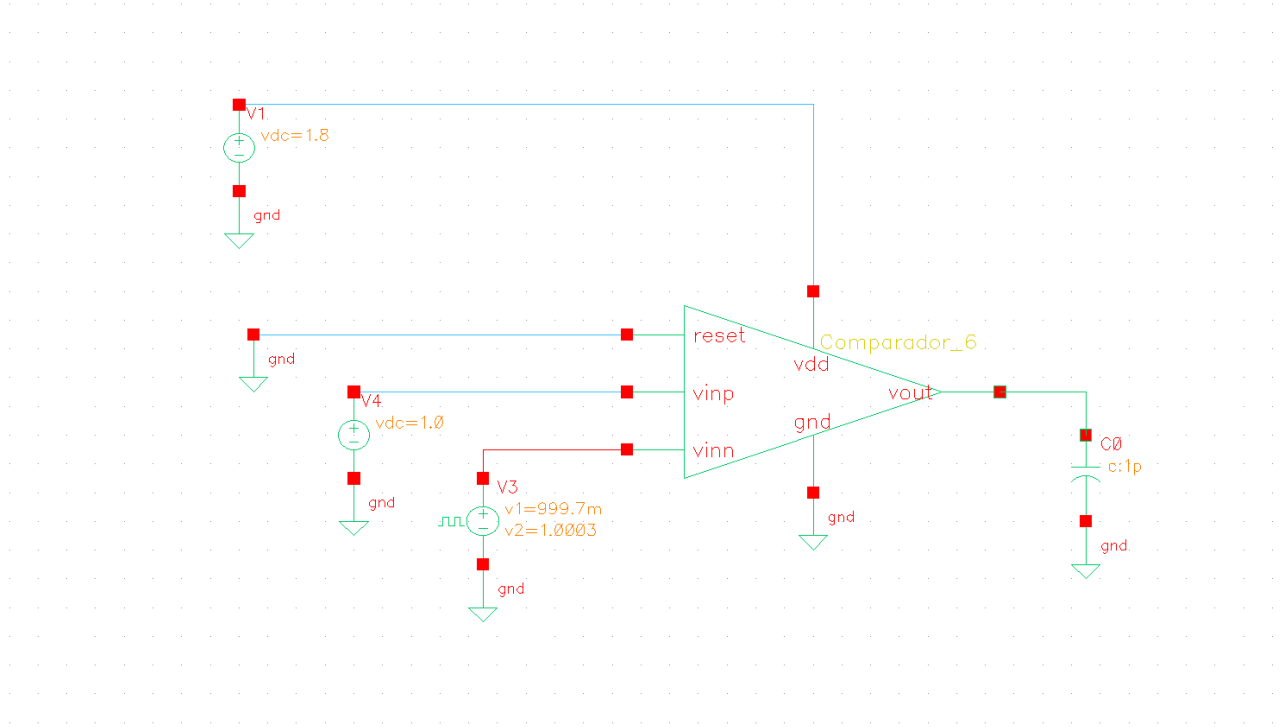

Figura 5.15. Circuito de teste do comparador

As figuras 5.16 e 5.17 os resultados da simulação transiente. A partir dos resultados, observa-se que o comparador demorou menos de 20 ns para reagir a variação de 300 μ V. Em outras simulações observou-se que a medida que essa variação aumenta, o tempo de reação do comparador diminui, assim, para menores resoluções o comparador terá um menor tempo de reação. Como mostrado na tabela 4.2, o comparador possui 20 ns para decidir o valor do bit de saída. Portanto, o comparador atende às especificações desejadas, em situações normais.

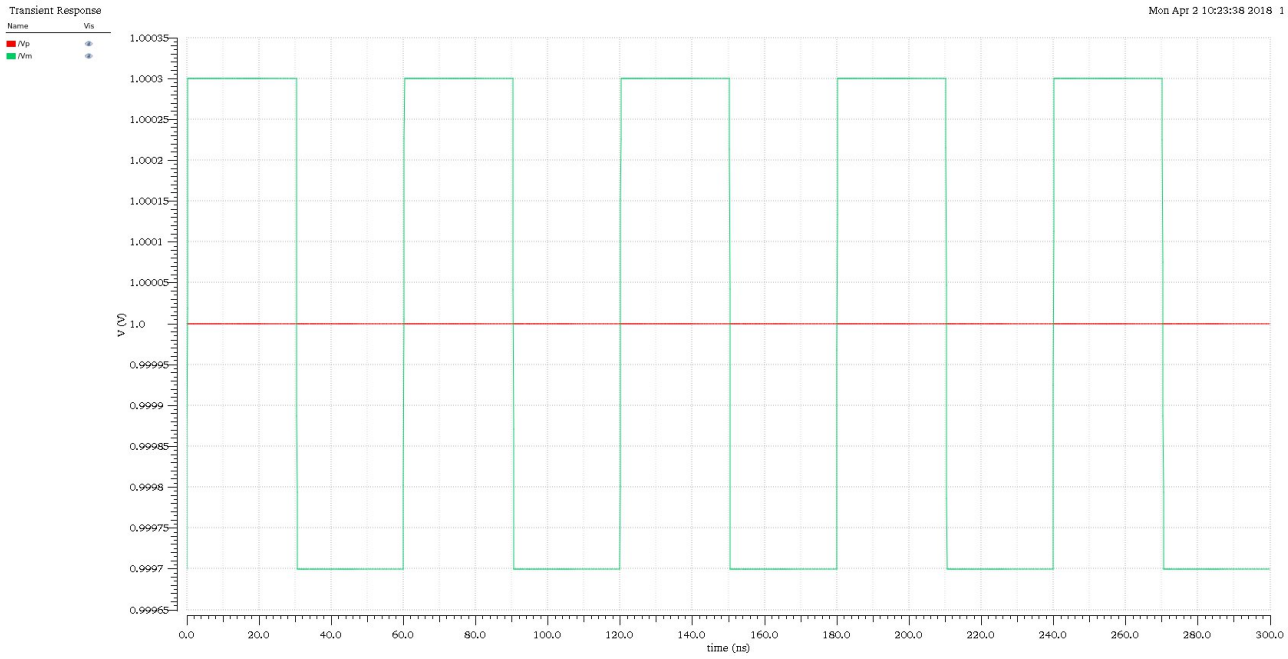

Figura 5.16. Sinais de entrada do comparador na simulação transiente do circuito

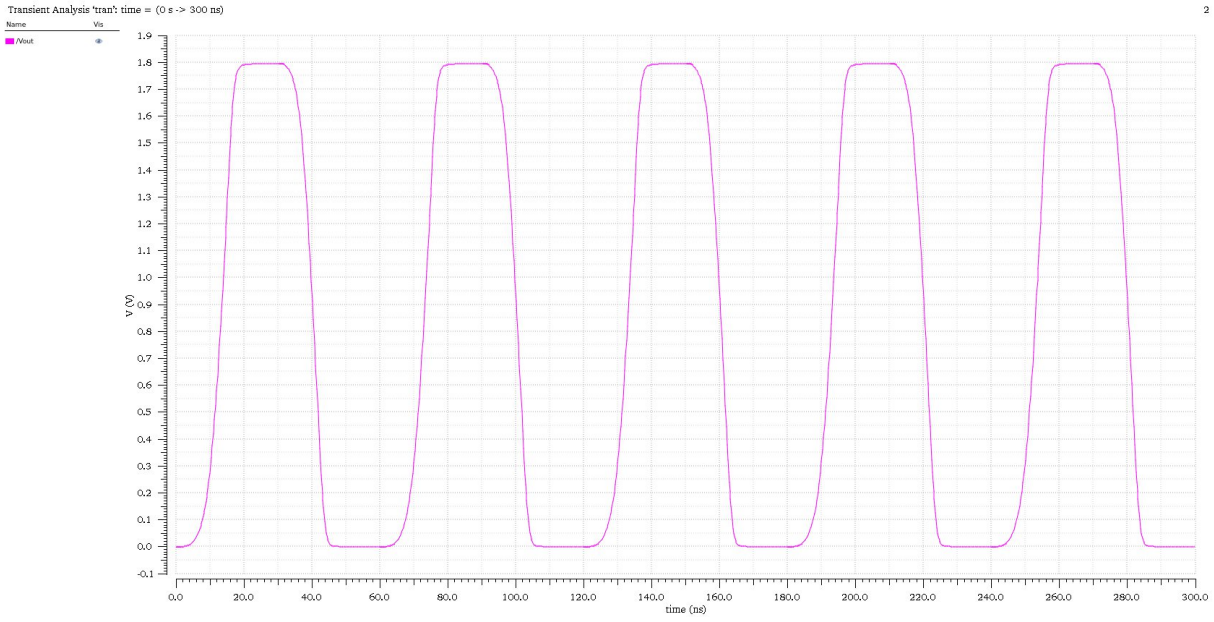

Figura 5.17. Sinal de saída do comparador na simulação transiente do circuito

Foi ainda avaliado em uma simulação AC, o desempenho de ganho e de fase do comparador. A figura 5.18 mostra o desempenho AC do comparador.

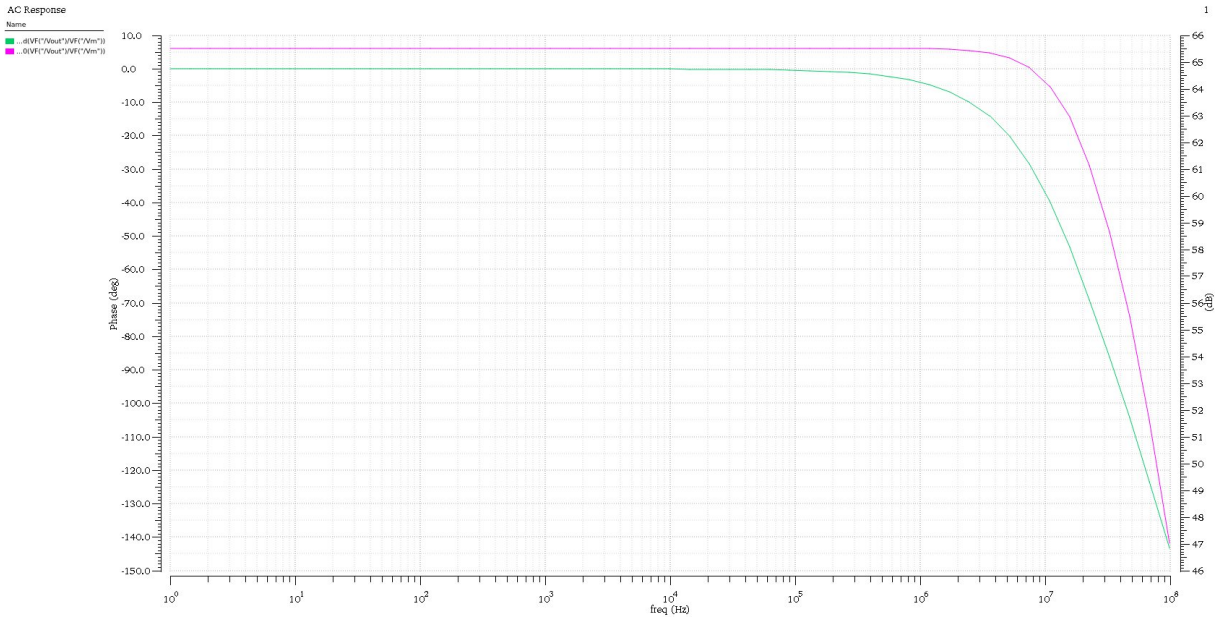

Figura 5.18. Simulação AC do comparador

Assim como foi feito para o S/H, foi necessário verificar o comportamento do circuito sob diferentes condições. Pretendia-se fazer uma simulação com os mesmos *corners* do circuito S/H, contudo, erros inesperados ocorreram no software utilizado para simulação e foi possível executar apenas os *corners* de temperatura. A especificação de temperatura do comparador atendeu as especificações do projeto funcionando de 0 °C a 80 °C.

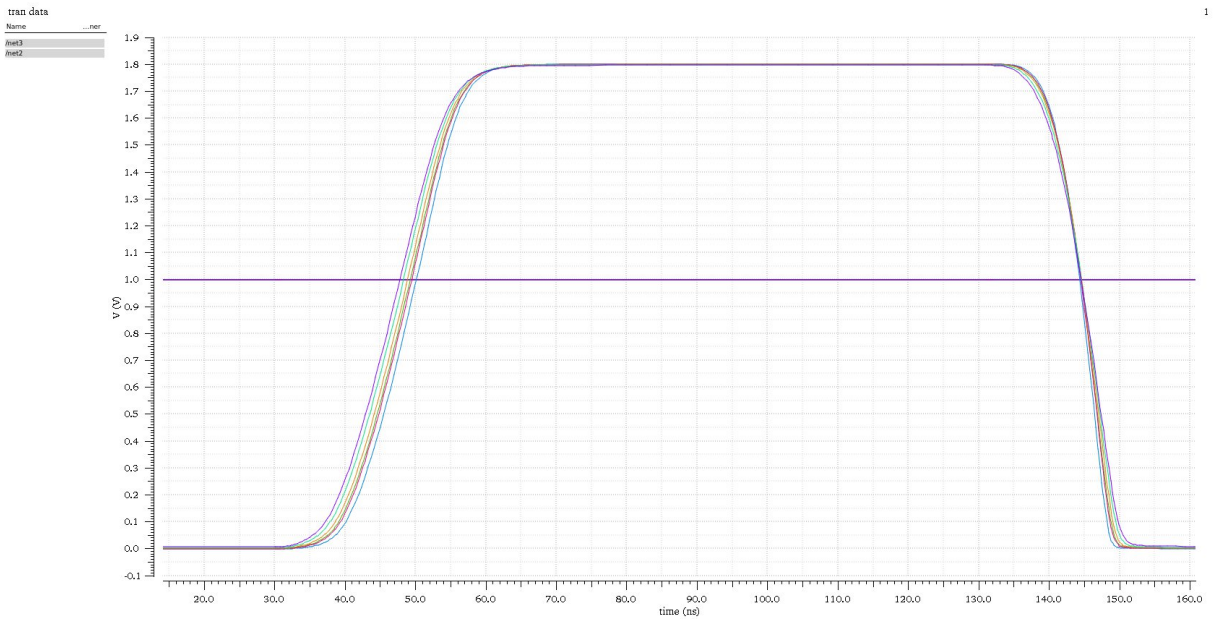

Figura 5.19. Simulação transiente de corners do comparador

Uma das ideias deste projeto foi utilizar o comparador como buffer de ganho unitário para carregar o terminal superior do array de capacitores com a tensão de modo comum Vcm. Além

disso, a ligação do comparador como buffer ainda apresenta a vantagem de se realizar um processo de cancelamento de offset no comparador. Para fazer com que o comparador pudesse trabalhar como buffer e comparador, adicionou-se um sinal de controle ao circuito que controla a função do comparador através do chaveamento de um capacitor de compensação no circuito do comparador.

Para avaliar o desempenho do comparador como buffer, utilizou-se o circuito do teste completo do conversor, mostrado na figura 5.25. A figura 5.20 mostra o resultado da simulação durante o período em que o comparador atua como buffer. Na figura, o sinal de cor verde representa o nó do terminal superior do array de capacitores e o sinal de cor verde representa Vcm. Na simulação, o buffer leva menos de 90 ns para carregar os capacitores com o nível adequado de tensão. Na tabela 4.2, o buffer possui 200 ns para realizar esta operação de carregamento, portanto, o buffer atende a essa especificação com folga. Ao final da fase de buffer, a diferença entre as duas entradas é aproximadamente igual a 1 LSB, o que pode causar erros de DNL no conversor. Contudo, este tipo de erro, por ser pequeno pode ser tratado na calibração do DAC.

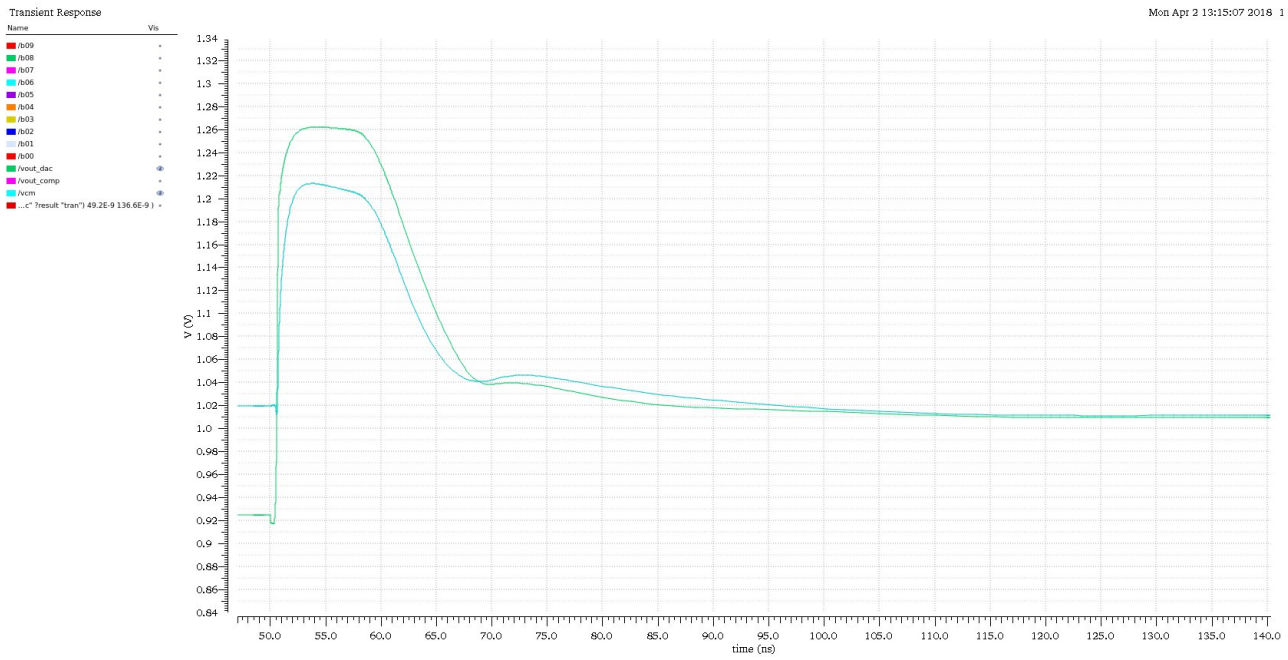

Figura 5.20. Simulação transiente do comparador como buffer de ganho unitário

5.3 CONVERSOR D/A

A princípio pensou-se em duas possibilidades para a topologia do DAC, o DAC tradicional de capacitores binários e o DAC *split array capacitor*. O compromisso que existe nesta escolha envolve principalmente a área do chip, consumo e linearidade do conversor. A capacitância total do DAC tradicional é igual a $2^{N}C$, em que N é o número de bits do conversor e C é o valor da capacitância unitária. Portanto, o valor da capacitância do circuito cresce exponencialmente com o número de bits, o que significa que a área ocupada pelo DAC e o consumo vão crescer exponencialmente também. Já o *split array capacitor* possui uma capacitância equivalente bem menor, porém, esta topologia apresenta maiores problemas de linearidade.

Dado o compromisso entre as topologias, escolheu-se primeiro testar a topologia *split array capacitor* e analisar sua função de transferência. Para a topologia *split array capacitor*, escolheu-se um valor de capacitância unitária que fosse alto o suficiente para se obter uma boa precisão no valor do capacitor de atenuação, já que o valor ideal deste possui várias casas decimais. No caso do DAC tradicional, utilizou-se o modelo de referência no Matlab para simular o mismatch de capacitores e ter um controle sobre o quanto isso iria afetar a performance estática de DNL e INL. No Matlab, foram obtidos os resultados mostrados nas figuras 5.21 e 5.22. Não foi possível fazer a previsão do efeito do mismatch de capacitores no caso do *split array capacitor* pois o modelo de referência é construído com base em um array tradicional, entretanto, foi possível utilizar um capacitor unitário consideravelmente maior na topologia *split array capacitor* e ainda obter uma capacitância equivalente bem menor que a do ao array tradicional. Como a capacitância unitária do *split array capacitor* é maior, espera-se que os efeitos de mismatch não sejam muito mais graves, apesar de que a capacitância equivalente do circuito *split array capacitor* é muito menor.

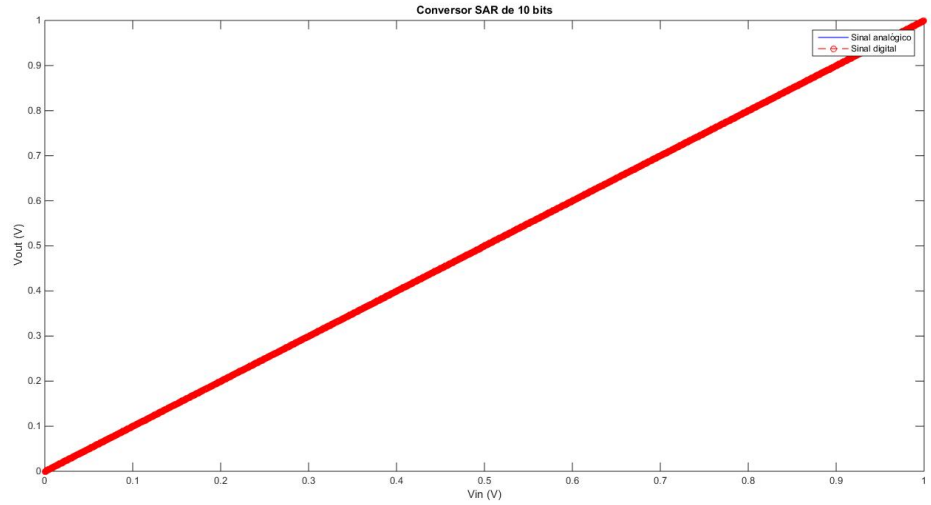

Figura 5.21. Análise do DAC do modelo de referência.

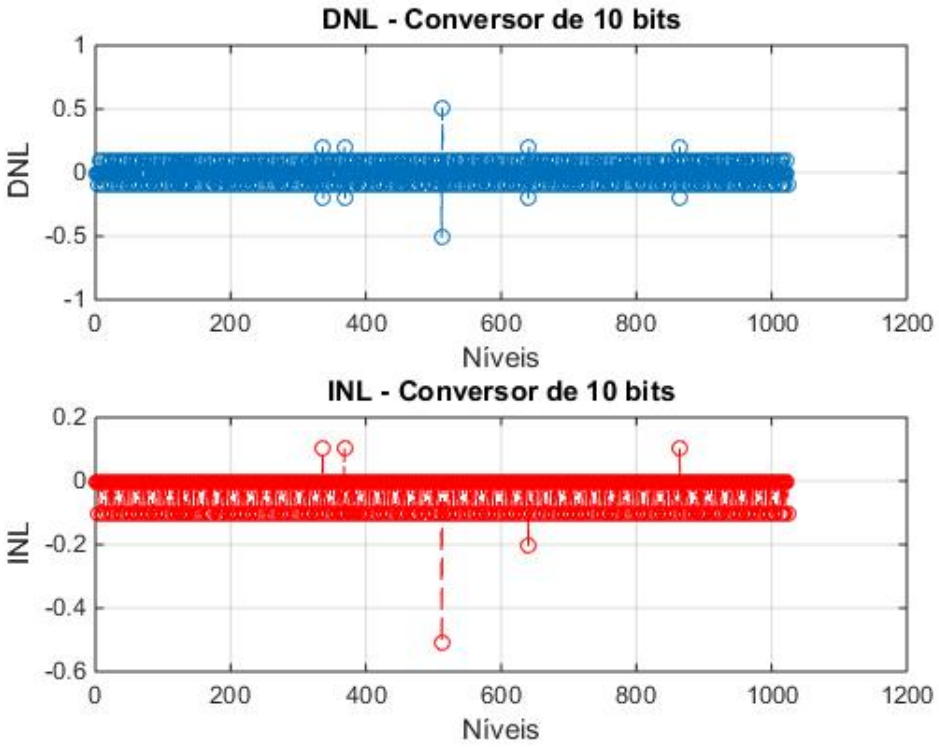

Figura 5.22. Análise da performance estática do modelo de referência.

A figura 5.23 mostra o circuito utilizado para se obter a função de transferência do DAC. Este circuito apresenta um bloco de referência de tensão seguido de um buffer de ganho unitário. A referência de tensão utilizada no bloco é de 1.5 V. O comparador foi colocado na saída do DAC, já que este é o circuito subsequente do DAC no modelo do conversor A/D SAR. Nesta simulação, as fontes ligadas nas entradas do DAC foram ajustadas para representar todas as

1024 combinações possíveis de entrada em um DAC de 10 bits.

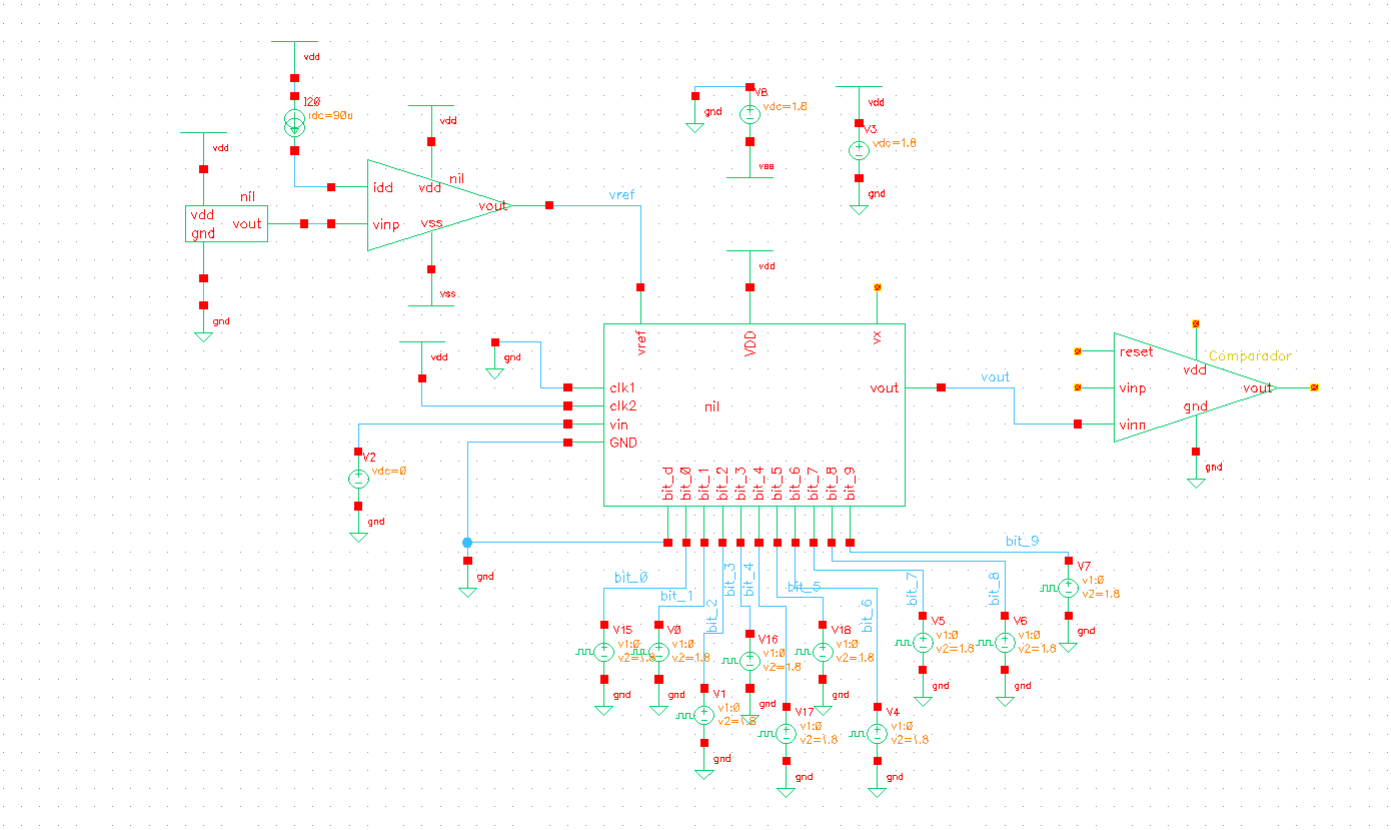

Figura 5.23. Circuito para teste do DAC.

Na simulação do DAC no Cadence, observou-se que a função de transferência apresentava vários pontos em que a função decrescia inesperadamente. Dado isso, procurou-se investigar o problema verificando primeiro o funcionamento das chaves do DAC, que estavam funcionando de acordo com o esperado. Uma possível hipótese é a variação da capacitância com a tensão sobre os capacitores. Visto essa hipótese, inicia-se um processo de calibração manual do array de capacitores. Após o processo de calibração, houveram melhoras significativas, mas a curva ainda não se mantém completamente linear. A função de transferência após calibração é mostrada na figura 5.24.

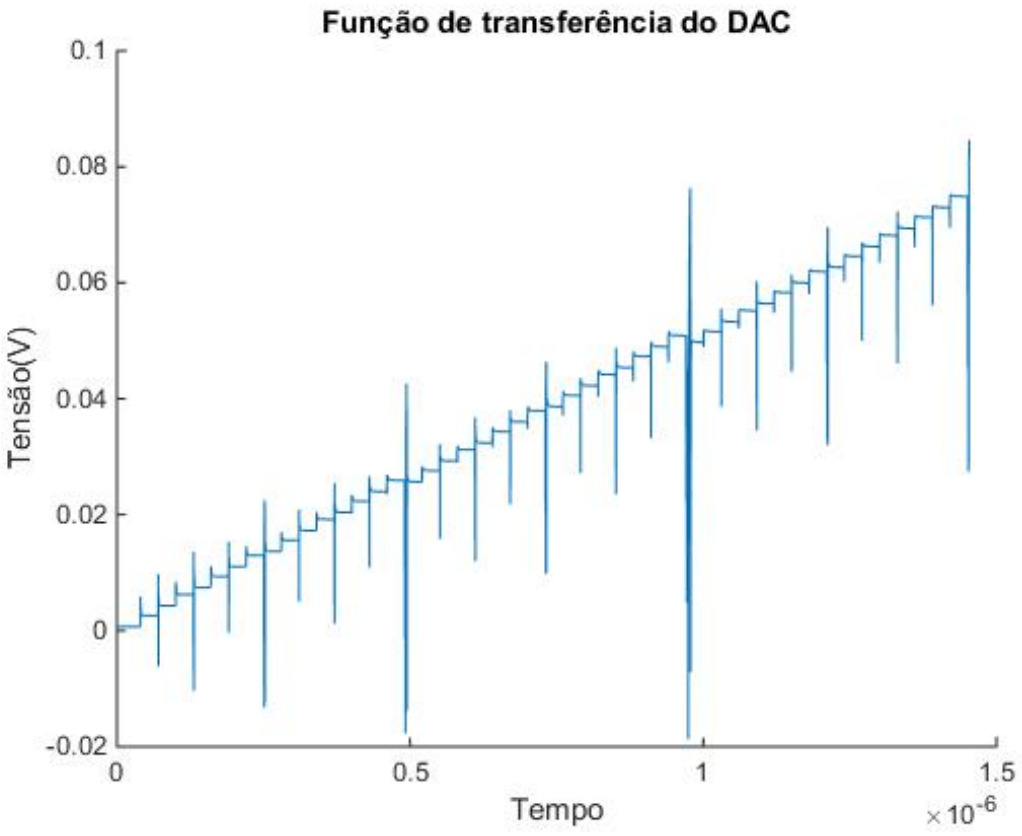

Figura 5.24. Função de transferência do DAC.

Na figura 5.24, os transientes entre um passo e outro na função de transferência acontecem devido ao chaveamento do DAC, não foi possível evitar esse efeito, a figura também não está representando toda a faixa de tensão testada, pois não foi possível plotar com boa resolução o gráfico inteiro, mas a curva completa segue o mesmo comportamento mostrado. Para avaliar se o transitório do chaveamento estava distorcendo a função de transferência, avaliou-se o mesmo circuito separadamente, porém com valores fixos representando os bits. Como resultado, observou-se que a resposta do DAC convergiu para o mesmo valor obtido na simulação anterior, o que sugere que o transitório do chaveamento não interfere no valor da resposta final.

Realizou-se também uma simulação com o DAC tradicional e este apresentou um comportamento similar ao do DAC *split array capacitor*, incluindo o problema de transientes entre os passos e a deformação da função de transferência. Como não houve vantagens em utilizar o DAC tradicional, decidiu-se utilizar o DAC *split array capacitor* no projeto do conversor.

A partir dos resultados sugere-se a hipótese de que os problemas de deformação da função de transferência sofridos pelo *split array capacitor* são decorrentes da não linearidade dos capa-

citores quanto a variação de tensão e portanto, podem ter seus resultados melhorados através da calibração do array. Buscou-se na documentação da tecnologia, uma caracterização do capacitor utilizado neste projeto, mas não foi encontrado nenhuma informação quanto a linearidade do capacitor sob variação de tensão.

5.4 SIMULAÇÃO DO MÓDULO ANALÓGICO COMPLETO

Como neste trabalho não foi desenvolvida a máquina de estados, implementou-se um circuito, mostrado na figura 5.25 para simular uma máquina de estados atuando sob o circuito analógico. Na entrada de cada bit do DAC foi colocada uma fonte de pulsos para simular as palavras binárias, além disso, foram utilizadas duas fontes de pulsos complementares para realizar o controle das fases de *sample* e *hold*. Ao dar início na simulação transiente, o conversor entra na fase de *sample* e amostra o sinal de entrada. Em seguida, o conversor entra na fase de *hold* carregando o array de capacitores com Vcm-Vin volts. O DAC testa a primeira palavra binária fornecida pelas fontes e o comparador indica se o bit deve se manter em 1 ou 0. De acordo com a resposta do comparador, o duty cycle das fontes de pulsos é ajustado manualmente, até que se obtenha a palavra completa.

A figura 5.26 mostra o resultado da simulação do circuito completo mostrado na figura 5.25. Na figura 5.26, pode-se observar todo o processo de conversão analógico digital descrito no capítulo de fundamentação teórica, vários sinais foram abstraídos da imagem para que fosse possível obter uma boa visualização dos principais sinais. O sinal de cor azul (vcm) representa a tensão de modo comum Vcm, o sinal de cor laranja (vout_dac) representa a tensão no terminal superior do array de capacitores e o sinal amarelo (vout_comp) representa a saída do comparador.

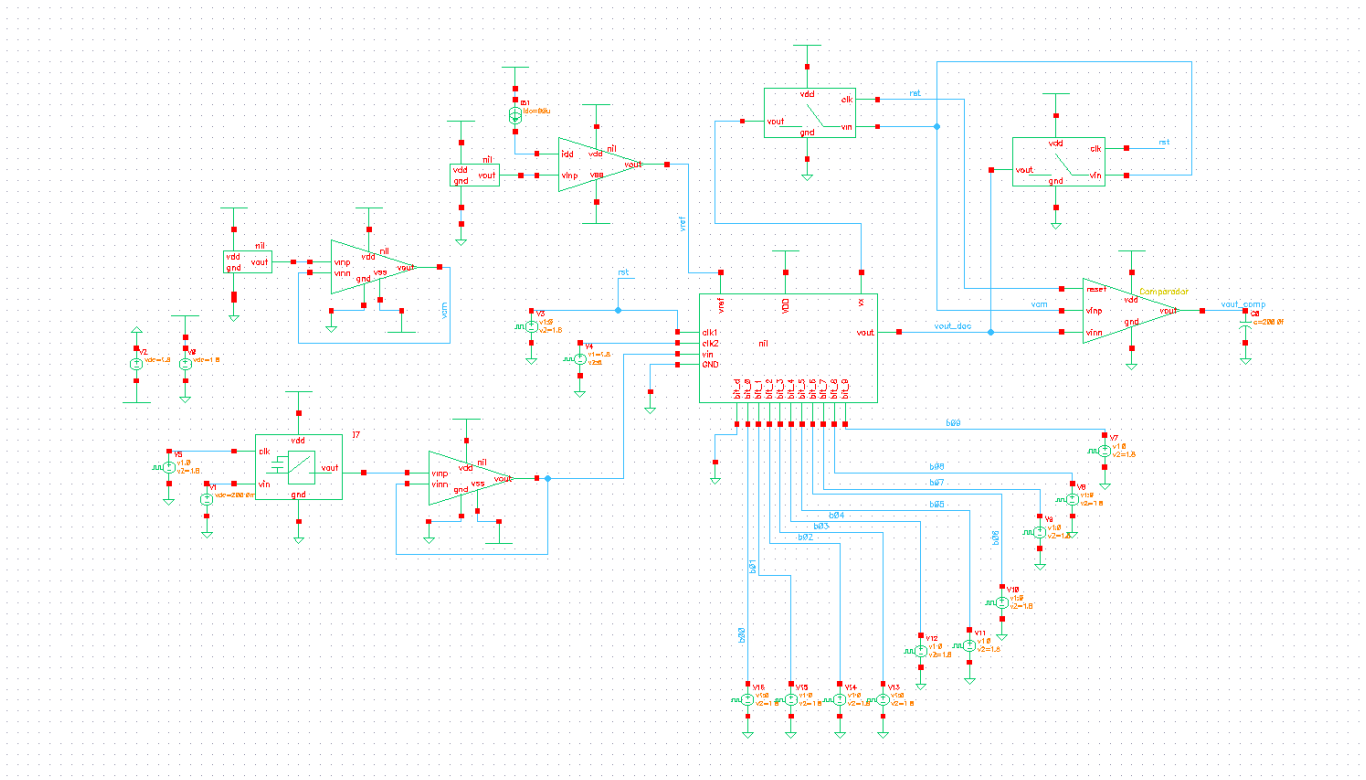

Figura 5.25. Circuito para teste dos módulos analógicos em conjunto.

Durante os primeiros 50 ns de simulação ocorre a fase de *sample*, em que o sinal de entrada é amostrado no S/H. A partir dos 50 ns, o conversor entra no estado *hold*, entre 50 e 140 ns o conversor se encontra na fase de carregamento de capacitores. Nesta fase, o comparador entra em modo buffer com o objetivo de fazer com que o sinal vout_dac seja igual a Vcm, observa-se que ao fim desta fase os sinais vout_dac e vcm se encontram praticamente sobrepostos na figura, como era desejado. É importante comentar que até este momento, a saída do comparador não está sendo lida pela máquina de estados, por isso seu valor não tem importância até este momento.

A partir dos 140 ns começa-se o processo de aproximações sucessivas, neste momento a primeira palavra binária é enviada ao DAC. As palavras binárias foram abstraídas da figura, mas cada palavra é mantida por 30 ns até a próxima palavra ser testada. No primeiro ciclo é testada a palavra com MSB igual a 1 e com os outros bits em 0. Neste momento, o sinal vout_dac fica acima de Vcm e o comparador vai para 1, como o sinal vout_comp está invertido, este bit é entendido como 0. Assim, a próxima palavra a ser testada é a 0100000000, que também apresenta como resultado um bit igual a 1 na saída vout_comp, por isso, o segundo bit também é 0. Agora, a palavra testada é 0010000000 e dessa vez o bit na saída do comparador é igual

a 0, por isso, o terceiro bit é 1 e a próxima palavra a ser testada será 0011000000. O processo continua até que todos os bits sejam testados.

A medida que os bits vão sendo determinados, a saída do DAC vai se aproximando cada vez mais de Vcm. Este é o comportamento esperado pelo conversor já que a tensão de modo comum está sendo comparada com ela mesma acrescida da diferença entre o sinal de entrada e o sinal proveniente do DAC, que representa o valor da aproximação. Portanto, isso significa que a aproximação feita pelo conversor está cada vez mais próxima do valor de entrada, o que é o objetivo do conversor.

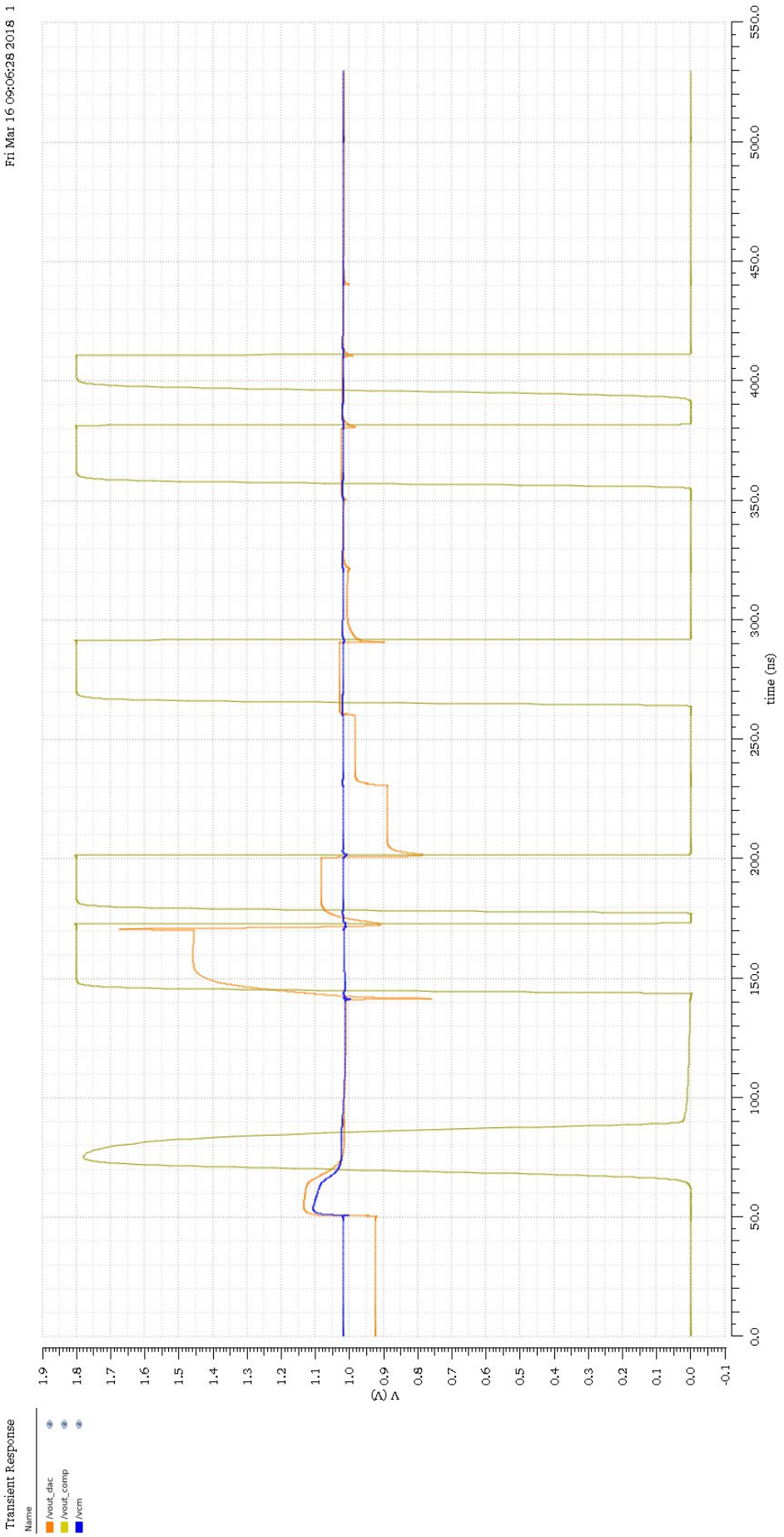

Figura 5.26. Resultado da simulação dos módulos analógicos para uma entrada igual a 0.3 V.

A ausência da máquina de estados dificultou a execução de testes e completa validação do conversor A/D. Entretanto, utilizou-se o circuito da figura 5.25 para se fazer alguns testes e obter a função de transferência do conversor A/D. Foram convertidos valores de 0 à 0.9 V, em seguida, foi realizada uma regressão desses valores. A partir dos coeficientes obtidos na regressão, foi plotada uma função de transferência do conversor. A figura 5.27 mostra a função de transferência obtida e também a função de transferência ideal para valores de 0 à 0.9 V.

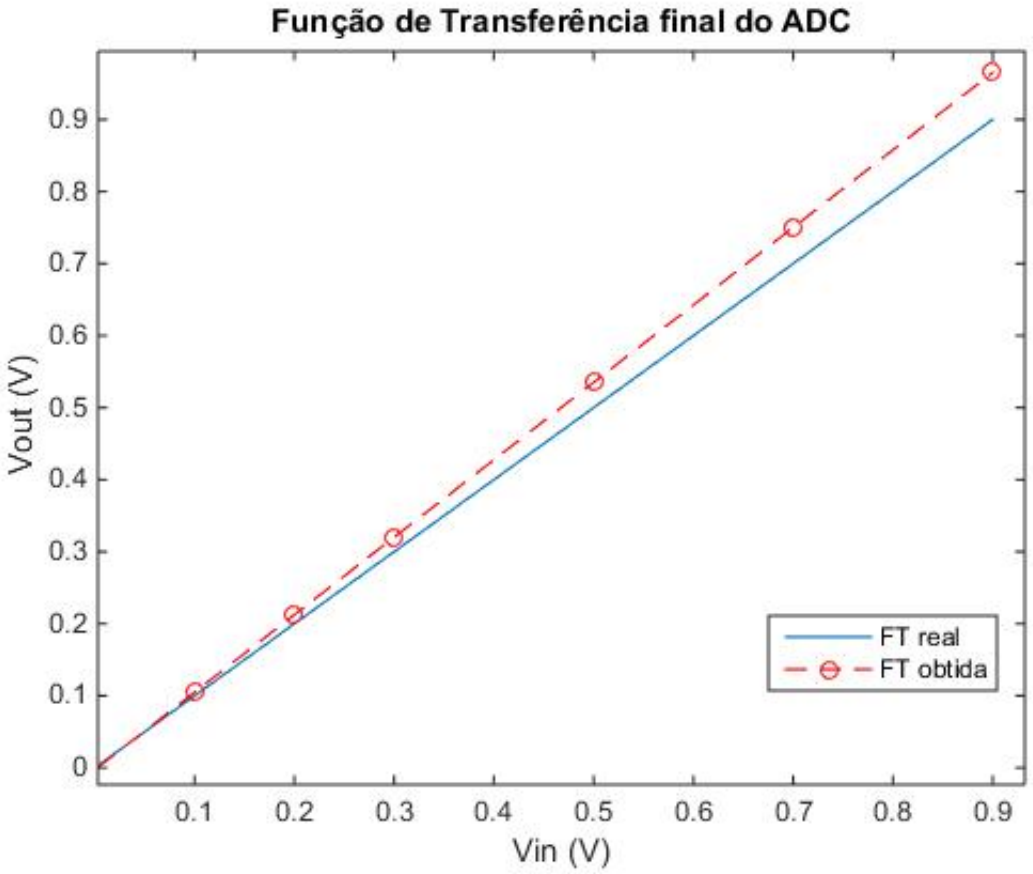

Figura 5.27. Função de transferência do conversor A/D.

A partir da figura 5.27, observa-se que o conversor apresenta erro de ganho. Esse tipo de erro pode ser corrigido com a calibração do array de capacitores. Como não foi possível realizar mais simulações do conversor A/D devido à falta da máquina de estados, não foi possível realizar testes que permitissem avaliar o desempenho dinâmico do conversor. Apesar de tudo, a função de transferência obtida em Matlab fornece uma boa aproximação do comportamento estático do conversor. Como seria necessária ainda a calibração do mesmo, a resposta estática vai apresentar erros de DNL e INL muito altos.

CAPÍTULO 6

CONCLUSÃO

Neste trabalho foi proposto o desenvolvimento do circuito analógico de um conversor analógico-digital por aproximações sucessivas. Foi realizado um estudo comportamental do conversor, em que foi avaliado principalmente o efeito do mismatch de capacitores na resposta final do conversor. Após o desenvolvimento do modelo de referência, foi realizado um estudo bibliográfico em que buscou-se definir especificações características de conversores realizados em tecnologia 0.18 μ m. Foi decidido realizar o projeto de implementação de um conversor de 10 a 12 bits, com taxa de amostragem de 1.10^6 amostras por segundo e baixo consumo.

Durante o projeto, foram estudadas e comparadas várias topologias de circuitos analógicos. Buscou-se validar todas as escolhas realizadas, através de simulações e análise de resultados. O circuito foi dividido em blocos e cada bloco foi projetado tendo em vista as especificações do projeto e o algoritmo proposto. Por fim, cada bloco teve seu desempenho testado e analisado separadamente. Ainda foi realizada uma análise do circuito completo com a inserção de um circuito de teste que simulou o funcionamento da máquina de estados.

Os resultados indicam que os blocos analógicos desenvolvidos são adequados para a implementação de um conversor analógico-digital por aproximações sucessivas de 10 bits e 1 MS/s. Entretanto, não foi possível fazer uma validação mais rigorosa do projeto por completo, já que sem uma máquina de estados para controlar o conversor, os testes se tornaram muito dispendiosos em tempo.

6.1 TRABALHOS FUTUROS

No mundo atual, os conversores analógico-digital são fundamentais no processamento de sinais, por isso, é importante dar continuidade no melhoramento destes dispositivos através do estudo de novos algoritmos e novas formas de implementação. Com base no trabalho de-

senvolvido, existem várias oportunidades de melhoria do projeto atual, sugere-se os seguintes trabalhos futuros:

- Realizar testes mistos junto a máquina de estados, possibilitando testes mais ágeis e também a avaliação da performance dinâmica do conversor;
- Continuar os próximos passos do fluxo de projeto do conversor A/D, passando por layout e simulação com parasitas;
- Otimizar o consumo do conversor através da implementação de outros algoritmos utilizando os mesmos blocos analógicos desenvolvidos nesse projeto;
- Implementar um conversor A/D SAR em modo diferencial utilizando os blocos analógicos desenvolvidos neste trabalho;

Como sugestão de trabalhos futuros, fica a elaboração de um banco de testes para validação e análise precisa de todas as características estáticas e dinâmicas deste conversor, a elaboração do layout do conversor e a implementação de um conversor SAR em modo diferencial.

REFERÊNCIAS BIBLIOGRÁFICAS

ALLEN, P.; HOLBERG, D. *CMOS Analog Circuit Design*. 3. ed. [S.l.]: OUP USA, 2011. (The Oxford Series in Electrical and Computer Engineering). ISBN 9780199765072. 33

AOUIZERATE, M. *Study for the improvement of a SAR interleaved ADC*. [S.l.], 2017. 21, 25

BAKER, R. J. *CMOS: circuit design, layout, and simulation*. 3. ed. [S.l.]: John Wiley & Sons, 2008. v. 1. iv, v, 12, 13, 14, 15, 17, 37, 44

GRAY, N. Abcs of adcs. *National Semiconductor Corporation, Aug*, v. 4, 2003. iv, 6, 7, 8, 9

MANGANARO, G. *Advanced data converters*. 1. ed. [S.l.]: Cambridge University Press, 2011. 3, 5, 12

MARJONEN, J. An 8 bit 10ks/s 0.18 μm cmos sar adc for rfid applications with sensing capabilities. *Norchip, 2007*, IEEE, Aalborg, Denmark, n. 6, nov. 2007. 19

MINH, H. N. A design of 10-bit 25-ms/s sar adc using separated clock frequencies with high speed comparator in 180nm cmos. p. 133–138, Oct 2015. ISSN 2162-1020. 19

PLASSCHE, R. J. Van de. *CMOS integrated analog-to-digital and digital-to-analog converters*. 2. ed. [S.l.]: Springer Science & Business Media, 2013. v. 742. iv, 10, 11

RAZAVI, B. *Principles of data conversion system design*. 1. ed. [S.l.]: IEEE Press, 1995. ISBN 9780780310933. v, 36, 37, 38

RAZAVI, B. The bootstrapped switch [a circuit for all seasons]. *IEEE Solid-State Circuits Magazine*, v. 7, n. 3, p. 12–15, Summer 2015. ISSN 1943-0582. v, 38, 39

ZHU, Z. A 0.6-v 38-nw 9.4-enob 20-ks/s sar adc in 0.18-μm cmos for medical implant devices. *IEEE Transactions on Circuits and Systems I: Regular Papers*, v. 62, n. 9, p. 2167–2176, Sept 2015. ISSN 1549-8328. 19

ZHU, Z.; QIU, Z.; LIU, M.; DING, R. A 6-to-10-bit 0.5 v-to-0.9 v reconfigurable 2 ms/s power scalable sar adc in 0.18 μm cmos. *IEEE Transactions on Circuits and Systems I: Regular Papers*, v. 62, n. 3, p. 689–696, March 2015. ISSN 1549-8328. 19

ZRILIC, D. *Circuits and Systems Based on Delta Modulation: Linear, Nonlinear and Mixed Mode Processing*. 1. ed. [S.l.]: Springer, 2005. (Signals and Communication Technology). ISBN 9783540237518. 14

APÊNDICE A

MODELO DE REFERÊNCIA DE CONVERSOR A/D SAR

Este apêncice contém o código de Matlab usado como modelo de referência para o estudo do conversor Analógico-Digital SAR. Este código e os códigos utilizados para realizar a caracterização da tecnologia encontram-se disponíveis no link abaixo:

https://github.com/iagospereira/Projeto-Conversor-A-D-SAR

```
% Conversor A/D SAR single-ended
% 11/03/2018
% TCC 2
% Universidade de Brasília
% Autor: Iago Sergio Pinheiro Pereira
% Matrícula: 11/0148959
% versão 0.7
%
clear all;
clc;

%% Variáveis do modelo
n_bits = 12; % número de bits do conversor
Vref = 1;  % Tensão de referência
Vcm = 1;    % Tensão de modo comum
Voff = 0*(Vref/(2^n_bits)); % Tensão de offset, em função de LSB, na saída
sinal = 2; % Sinal de entrada: 1=rampa, 2=senoide
```

```
%% Sinal de entrada
switch sinal
case 1
tmax = 1;   % tempo do processo inteiro em segundos
sp = 10000; % número de amostras
fs = sp/tmax; %taxa de amostragem
t = 0:(1/fs):(tmax-(1/fs));
Vin = t;
case 2
tmax = 1;   % tempo do processo inteiro em segundos
sp = 20000; % número de amostras
fs = sp/tmax; %taxa de amostragem
t = 0:(1/fs):(tmax-(1/fs));
f = fs/30;
Vin = (Vref/2) + (Vref/2)*cos(2*pi*f*t);
end;

%% Array de capacitores
C = 15e-15;  % Capacitância unitária
C_cte = 3;  % Capacitor da Tecnologia: cmm=1 , cdmm=2 ,ctmm=3
AC = 0.0023;  % Capacitor da tecnologia: cmm=0.004 , cdmm=0.0028 , ctmm =0.0023
C_mismatch = 1; % 0 = sem mismatch, 1 = mismatch gaussiano, 2 = mismatch fixo
C_misfixo = 0.1*C; % mismatch fixo

switch C_mismatch
case 0  %sem mismatch
for i=1:n_bits
C_ar(i) = C*(2^(n_bits-i));
end;
C_ar(n_bits+1) = C; %dummy
case 1  %com mismatch gaussiano
```

```
WL = (C/C_cte)*1e15;   % Área do capacitor
C_mis_sig = AC/sqrt(WL); % desvio padrão
for i=1:n_bits
C_ar(i) = C*(normrnd(1,C_mis_sig))*(2^(n_bits-i));
end;
C_ar(n_bits+1) = C*(normrnd(1,C_mis_sig))*(2^(n_bits-i));   %dummy
case 2  %com mismatch fixo
for i=1:n_bits
C_ar(i) = C*(2^(n_bits-i))*(1+C_misfixo);
end;
C_ar(n_bits+1) = C*(1+C_misfixo);   %dummy

end;
C_total = sum(C_ar);

%% Proceso de Conversão Single Ended Tradicional
for i=1:sp
bit = zeros(1,n_bits);
% Fase 1: Distribuição de (Vcm-Vin) sobre os capacitores
Qi = C_total*(Vcm - Vin(i));
% Fase 2: Entrada dos bits no DAC
for a=1:n_bits
%         Q_total = 0;
bit(a)=1;
for b=1:n_bits
if bit(b)==1
Vdac(b) = (C_ar(b)/C_total)*Vref;
else
Vdac(b)=0;
end;
end;
```

```
%         Q_total = (sum(Vdac)*C_total) + Qi;
Vcomp = Vcm - (sum(Vdac) + Qi/C_total);
if(Vcomp < 0)
bit(a)=0;
end;
end;
Vout_b(i,:) = bit;
end;

%% Recomposição do sinal
for i=1:sp
for a=1:n_bits
Vout_r(a) = (Vout_b(i,a)*(Vref/(2^a)));
end;
Vout(i) = sum(Vout_r) + Voff;
end;

%% Plotando os sinais
figure;
if(sinal==1)
plot(Vin,Vin,'-b',Vin,Vout,'--ro'); % Plotando rampa
else
plot(t,Vin,'-b',t,Vout,'--ro'); % Plotando senoide
xlim([0 0.01]);
end;
legend('Sinal analógico','Sinal digital');
title(['Conversor SAR de ',num2str(n_bits),' bits']);
xlabel('Vin (V)');
ylabel('Vout (V)');

%% Cálculo do DNL e INL (apenas para entrada rampa)
```

```
if(sinal==1)
figure(2);
edges = [0:(Vref/(2^n_bits)):Vref];
h = histogram(Vout,edges);
counts = h.Values;
DNL(1)=0;
for i=2:(2^n_bits)
DNL(i) = (counts(i)-counts(i-1))/(sp/((2^n_bits)-1));
end;
%% Cálculo do INL
INL(1)=DNL(1);
for i=1:(length(DNL)-1)
INL(i+1) = INL(i) + DNL(i+1);
end;
%% Plotando DNL e INL
figure(2);
subplot(2,1,1);
plot(DNL,'--o');
grid on;
%     ylim([-3 3]);
title(['DNL - Conversor de ',num2str(n_bits),' bits']);
xlabel('Níveis');
ylabel('DNL');
subplot(2,1,2);
plot(INL,'--ro');
grid on;
%     ylim([-3 3]);
title(['INL - Conversor de ',num2str(n_bits),' bits']);
xlabel('Níveis');
ylabel('INL');
end;
```

```
%% Análise espectral dos sinais de entrada e saída do conversor
if(sinal==2)
analise_espectral(Vin,Vout,fs);
end;
```