

UNIVERSIDADE FEDERAL DE ITAJUBÁ

**PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA**

Uma arquitetura de conversão A/D baseada
na aproximação sucessiva de sinais PWM em
tecnologia CMOS

Gustavo Della Colletta

Itajubá, Março de 2012

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA

Gustavo Della Colletta

Uma arquitetura de conversão A/D baseada
na aproximação sucessiva de sinais PWM em
tecnologia CMOS

Dissertação submetida ao Programa de Pós-
Graduação em Engenharia Elétrica como parte
dos requisitos para obtenção do Título de Mestre em
Ciências em Engenharia elétrica.

Área de Concentração: Microeletrônica
Orientador: Tales Cleber Pimenta

Março de 2012

Itajubá - MG

Agradecimentos

Agradeço primeiramente a Deus por me dar a oportunidade de aprofundar meus estudos.

Aos meus pais pelo apoio incondicional e incentivo durante toda minha vida.

Agradeço também aos amigos, que muitas vezes à distância, nunca deixaram de manifestar seu apoio e amizade.

Também agradeço aos professores e novos amigos do Grupo de Microeletrônica pelas discussões referentes ao presente trabalho e também pelos momentos de descontração.

Por fim, agradeço ao CNPq, CAPES, FAPEMIG e MOSIS por promover o suporte financeiro necessário para o desenvolvimento deste trabalho.

A mente que se abre a uma nova ideia
jamais voltará ao seu tamanho
original.

Albert Einstein

Resumo

Esse trabalho propõe uma nova arquitetura para o estágio de conversão D/A presente em um conversor A/D por aproximação sucessiva. Uma das desvantagens desse tipo de conversor A/D é sua não linearidade, que degrada seus parâmetros de DNL e INL. Esses erros de linearidade são causados por descasamentos durante o processo de fabricação. Os conversores SAR convencionais utilizam redes resistivas ou capacitivas em seu estágio D/A, o que os torna bastante suscetíveis a tais descasamentos.

A arquitetura proposta almeja eliminar esses descasamentos, minimizando assim os erros de não linearidade através da utilização da técnica PWM em seu estágio de conversão D/A. Também é utilizado um circuito comparador inversor síncrono com o intuito de simplificar o circuito e diminuir seu consumo de potência.

Para validar a arquitetura, foi simulado um conversor de 4bits utilizando a ferramenta Spectre e modelos BSIM3v3 para um processo CMOS padrão de $0,5\mu\text{m}$. Alimentado com $2,5\text{V}$, o consumo de potência verificado foi de $16\mu\text{W}$. A taxa de conversão ficou limitada em 200 Hz devido ao projeto do circuito e à máxima frequência disponibilizada pelo processo de fabricação.

Os resultados das simulação comprovam a precisão alcançada pela arquitetura proposta, uma vez que os erros de DNL e INL são inferiores a $0,1\text{LSB}$. A relação sinal-ruído obtida é de $24,36\text{dB}$ e representa $3,75$ bits efetivos de resolução.

Abstract

This work is intended to present a new architecture for a SAR A/D converter which uses the PWM technique in the internal DAC converter. One of the main disadvantages of SAR ADCs is non-linearity, which degrades the DNL and INL parameters. These errors are caused by the fabrication process. Since conventional SAR converters use resistor or capacitor networks in their DAC stage, they are very susceptible to these errors.

The proposed architecture aims at eliminating these mismatches and then minimizing the above errors using the PWM technique in the DAC stage. A clocked inverter comparator is used in order to simplify the circuit complexity and decrease the power consumption.

In order to validate this architecture, a 4bit A/D converter has been simulated on Spectre simulator using BSIM3v3 model for a $0.5\mu m$ CMOS process. Supplied with $2.5V$, power consumption is only $16\mu W$. The sample rate was limited to $200Hz$, regarding the circuit design and the maximum frequency achieved by the CMOS process.

Simulation results show high accuracy achieved by the proposed architecture, once the DNL and INL errors are smaller than $0.1LSB$. The signal to noise and distortion ratio is $24.36dB$ which represents 3.75 effective bits resolution.

Lista de siglas

| | |
|------|--------------------------------------|
| A/D | Análogo-Digital |
| DC | Corrente Contínua |
| CDK | Cadence Desing Kit |
| CT | Ciclo de Trabalho |
| D/A | Digital-Analógico |
| DFT | Transformada Discreta de Fourier |
| DNL | Não Linearidade Diferencial |
| DUT | Dispositivo Sob Teste |
| ENOB | Número Efetivo de Bits |
| FFT | Transformada Rápida de Fourier |
| INL | Não Linearidade Integral |
| LSB | Bit Menos Significativo |
| MSB | Bit Mais Significativo |
| NCSU | Universidade da Carolina do Norte |
| nMOS | Transistor MOS do tipo n |
| PWM | Modulação por Largura de Pulso |
| RMS | Valor Eficaz |
| SAR | Registrador de Aproximação Sucessiva |
| SNR | Relação Sinal-Ruído |
| VHDL | Linguagem de Descrição de Hardware |

Lista de símbolos

| | |
|-----------------|-------------------------------------------------------|
| FS | Escala completa |
| \angle | Inclinação do ganho |
| $erro_{ganho}$ | Erro de ganho |
| V_{off} | Erro de offset |
| p | Ciclo de trabalho do sinal PWM |
| k | Amplitude do sinal PWM |
| T | Período do sinal PWM |
| A_0 | Frequência fundamental |
| A_n | Harmônicas de ordem ímpar |
| B_n | Harmônicas de ordem par |
| S_1, S_2, S_3 | Chaves do circuito comparador |
| ϕ_1 | Instante 1 |
| ϕ_2 | Instante 2 |
| C | Capacitor de entrada do circuito comparador |
| V_{in} | Tensão de entrada do circuito comparador |
| V_t | Tensão de limiar do transistor |
| V_{GS} | Tensão de porta-fonte |
| V_G | Tensão de porta |
| V_{pwm} | Tensão originada do modulador PWM |
| B | Banda de frequência do sinal amostrado |
| f_s | Frequência de amostragem |
| f_{sar} | Frequência do Registrador de aproximação sucessiva |
| N | Número de bits do registrador em anel |
| f_c | Frequência de corte do filtro |
| τ | Constante de tempo do filtro |
| h_1 | Amplitude do ripple do sinal PWM em função do cosseno |
| h_{1pp} | Amplitude do ripple do sinal PWM |
| A | Atenuação do filtro |
| A_{dB} | Atenuação do filtro em dB |
| f_{pwm} | Frequência do sinal PWM |
| f_{pwm}^{clk} | Frequência do modulador PWM |
| V_{REF} | Tensão de Alimentação |

Sumário

Lista de Figuras

Lista de Tabelas

| | | |
|----------|------------------------------------------------|-----------|
| 1 | Introdução | 14 |
| 1.1 | Visão geral | 14 |
| 1.2 | Organização do trabalho | 15 |
| 2 | Revisão bibliográfica | 16 |
| 2.1 | Conversão análogo-digital | 16 |
| 2.1.1 | Erro de quantização | 17 |
| 2.1.2 | Resolução | 17 |
| 2.1.3 | Número efetivo de bits | 18 |
| 2.1.4 | Erro de não linearidade diferencial | 18 |
| 2.1.5 | Erro de não linearidade integral | 19 |
| 2.1.6 | Função de transferência do conversor | 20 |
| 2.1.7 | Monotonicidade | 20 |
| 2.1.8 | Perda de código | 20 |
| 2.1.9 | Relação sinal-ruído | 20 |
| 2.2 | Arquitetura SAR | 23 |
| 2.3 | Modulação por largura de pulso (PWM) | 24 |
| 2.3.1 | Conceito | 25 |
| 2.3.2 | Caracterização do sinal PWM | 25 |

| | | |
|----------|----------------------------------------------------|-----------|
| 3 | Modelagem | 27 |
| 3.1 | Considerações iniciais | 27 |
| 3.2 | Sistema proposto | 28 |
| 3.3 | Registrador de Aproximação Sucessiva | 30 |
| 3.4 | Modulador PWM Digital | 33 |
| 3.5 | Filtro de primeira ordem | 37 |
| 3.6 | Comparador | 37 |
| 3.7 | Equacionamento | 41 |
| 3.8 | Validação do modelo | 46 |
| 3.8.1 | Dados | 46 |
| 3.8.2 | Cálculos | 46 |
| 3.8.3 | Simulações | 48 |
| 3.8.4 | Funcionamento da arquitetura | 49 |
| 4 | Layout e simulações | 51 |
| 4.1 | Layout | 51 |
| 4.2 | Parâmetros estáticos | 56 |
| 4.3 | Parâmetros dinâmicos | 59 |
| 4.4 | Análise comparativa | 59 |
| 5 | Conclusão e trabalhos futuros | 62 |
| 5.1 | Conclusão | 62 |
| 5.2 | Trabalhos futuros | 63 |
| | Referências | 64 |
| | Apêndice A – Modelos Digitais | 66 |
| A.1 | Registrador de Aproximação Sucessiva-SAR | 66 |

| | | |
|---------------------------------|----------------------------------|-----------|
| A.2 | Modulador PWM | 67 |
| A.2.1 | Registrador | 67 |
| A.2.2 | Contador | 68 |
| A.2.3 | Comparador | 69 |
| A.2.4 | Modulador PWM completo | 70 |
| Apêndice B – Publicações | | 71 |
| B.1 | Artigos | 71 |
| B.2 | Capítulo de Livro | 71 |

Lista de Figuras

| | | |
|----|---------------------------------------------------------------------------------------------------------|----|
| 1 | Função de transferência de um conversor A/D de 3 bits e seu erro de quantização. | 17 |
| 2 | Erros de DNL e INL em um conversor A/D de 3 bits. | 19 |
| 3 | Espectro de frequências de um sinal senoidal convertido por um ADC ideal. | 21 |
| 4 | Espectro de frequências de um sinal senoidal convertido por um ADC que possui não linearidades. | 22 |
| 5 | Diagrama de blocos da arquitetura SAR de conversão A/D. | 24 |
| 6 | Sinal PWM deslocado no tempo para a obtenção de uma função par. | 25 |
| 7 | Diagrama de blocos do sistema proposto. | 28 |
| 8 | Sinal <i>PWM</i> decomposto. | 29 |
| 9 | Procedimento de filtragem do sinal <i>PWM</i> | 29 |
| 10 | Esquema lógico do Registrador de aproximação sucessiva. | 31 |
| 11 | Operação do Registrador de Aproximação Sucessiva com a entrada SEL em nível lógico baixo. | 32 |
| 12 | Operação do Registrador de Aproximação Sucessiva com a entrada SEL em nível lógico alto. | 33 |
| 13 | Lógica do circuito modulador PWM. | 34 |
| 14 | Esquema lógico do Modulador de largura de pulso. | 35 |
| 15 | Sinal PWM para uma entrada digital 0001 e a tensão de quantização após a filtragem. | 36 |
| 16 | Sinal PWM para uma entrada digital 1000 e a tensão de quantização após a filtragem. | 36 |
| 17 | Esquemático do filtro passivo de primeira ordem. | 37 |

| | | |
|----|-------------------------------------------------------------------------|----|
| 18 | Resposta do filtro de primeira ordem. | 37 |
| 19 | Circuito do estágio comparador. | 38 |
| 20 | Circuito dobrador de tensão. | 39 |
| 21 | Sinais do circuito dobrador de tensão. | 40 |
| 22 | Funcionamento do circuito comparador. | 40 |
| 23 | Caracterização do ripple do sinal filtrado. | 45 |
| 24 | Determinação da frequência do sinal PWM em função da atenuação. | 45 |
| 25 | Diagrama de Bode do filtro projetado. | 47 |
| 26 | Resposta do conversor D/A para os dezesseis códigos possíveis. | 48 |
| 27 | Funcionamento do conversor A/D com registrador de aproximação sucessiva | 48 |
| 28 | Etapas da conversão A/D. | 50 |
| 29 | Layout do estágio comparador. | 52 |
| 30 | Layout do estágio de aproximação sucessiva. | 53 |
| 31 | Layout do estágio modulador PWM. | 54 |
| 32 | Layout completo do conversor. | 55 |
| 33 | Entrada analógica amostrada através da região de transição. | 56 |
| 34 | Curva de probabilidade de conversão de um novo código. | 57 |
| 35 | Visualização dos erros DNL e INL do conversor. | 58 |
| 36 | Resultado da FFT para 32 amostras. | 59 |

Lista de Tabelas

| | | |
|---|------------------------------------------------------------------------------|----|
| 1 | Variação do ciclo de trabalho para uma palavra digital de quatro bits. . . . | 30 |
| 2 | Sinais do registrador de aproximação sucessiva | 32 |
| 3 | Sinais do modulador de largura de pulso | 35 |
| 4 | Resultado da simulação com uma entrada em rampa lenta. | 58 |
| 5 | Quadro comparativo. | 60 |
| 6 | Característica da arquitetura simulada. | 61 |

1 *Introdução*

Esse capítulo tem o objetivo de mostrar a importância da conversão análogo-digital nos sistemas digitais atuais, bem como introduzir a ideia principal do trabalho proposto.

1.1 Visão geral

Na natureza os sinais são analógicos, entretanto, os sistemas de processamento são digitais. Assim se torna necessário um circuito capaz de converter os sinais analógicos em sinais digitais, os conversores análogo-digitais. Eles formam uma interface para que os microcontroladores sejam capazes de interpretar os sinais analógicos. Conseqüentemente esse circuito é muito importante em qualquer sistema digital.

As não linearidades diferencial (DNL) e integral (INL) influenciam de maneira negativa no número efetivo de bits (ENOB) do conversor, degradando a resolução do circuito, pois os bits menos significativos (LSBs) se tornam imprecisos. Assim, o circuito concebido sempre possui uma resolução efetiva menor que a resolução projetada.

Dessa maneira, se torna importante buscar uma arquitetura de projeto que melhore o desempenho do conversor, uma vez que as não linearidades DNL e INL são características particulares da arquitetura escolhida para o projeto.

O registrador de aproximação sucessiva (SAR) é uma arquitetura de conversão muito comum e pode alcançar taxas de conversão intermediárias (kS/s) com um consumo moderado de potência (μW). Atualmente, a arquitetura SAR é largamente utilizada, inclusive em sistemas de aquisição biomédicos [1, 2].

Esse trabalho apresenta uma arquitetura SAR de conversão análogo-digital que utiliza a técnica de modulação de largura de pulso (PWM) para melhorar o desempenho do conversor, bem como reduzir a área de silício necessária, uma vez que o circuito proposto é quase completamente digital.

1.2 Organização do trabalho

Esse trabalho está organizado em cinco capítulos, sendo o primeiro introdutório.

O segundo capítulo traz uma revisão teórica onde são abordados tópicos sobre conversores A/D e também sobre modulação por largura de pulso, proporcionando um melhor entendimento do trabalho proposto.

O terceiro capítulo mostra o sistema proposto, seu funcionamento e a modelagem de cada estágio do sistema.

No quarto capítulo são apresentadas as principais figuras de mérito da arquitetura, obtidas através de simulações de pós-layout.

O quinto capítulo encerra o trabalho mostrando as conclusões que foram observadas e sugerindo uma série de trabalhos que podem ser desenvolvidos à partir deste.

2 Revisão bibliográfica

Esse capítulo tem por objetivo promover uma revisão dos conceitos fundamentais utilizados nesse trabalho.

Inicialmente são abordados os conversores análogo-digitais, discutindo-se suas características estáticas e dinâmicas. Em seguida, a arquitetura Registrador de Aproximação Sucessiva (SAR), utilizada nesse trabalho, é explicada.

Existe também uma revisão dos conceitos básicos da modulação por largura de pulso (PWM), onde sua definição é explicada e as características do sinal modulado são detalhadas através de uma análise matemática.

Os tópicos relacionados aos conversores A/D tomam como base as obras [3] e [4] enquanto os tópicos sobre modulação PWM são embasados por [5] e [6].

2.1 Conversão análogo-digital

Hoje em dia são utilizadas diversas arquiteturas de conversão análogo-digital. A escolha de uma determinada arquitetura é baseada em sua aplicação. Dessa maneira, se faz necessário caracterizar detalhadamente o sinal analógico de entrada que será amostrado, para que parâmetros importantes do conversor, como taxa de amostragem, faixa dinâmica e precisão possam ser corretamente identificados.

Deve-se tomar muito cuidado na interpretação da folha de dados do componente, pois muitas vezes o fabricante não deixa claro sua definição de determinada especificação e utiliza definições que melhorem os resultados dos testes. Também deve-se atentar ao fato de que os testes detalhados nas folhas de dados são realizados sob condições diferentes das condições em que o dispositivo vai trabalhar em determinada aplicação.

Dessa maneira se faz necessário o entendimento das especificações do conversor A/D e de suas definições, bem como o entendimento dos testes realizados para medir essas

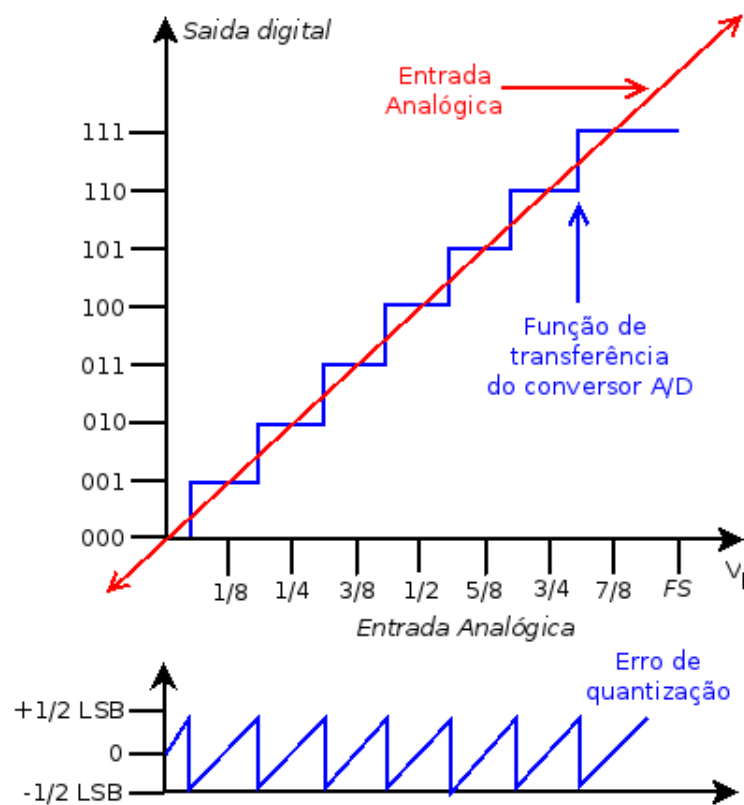


Figura 1: Função de transferência de um conversor A/D de 3 bits e seu erro de quantização.

especificações e também das condições de realização dos mesmos.

2.1.1 Erro de quantização

O erro de quantização é o erro natural que ocorre durante a digitalização de um sinal analógico. Ele decorre do fato de que um mesmo código digital de saída pode ser obtido à partir de uma faixa de sinais analógicos de entrada, fazendo com que as magnitudes do sinal de entrada e do código de saída sejam diferentes. Essa diferença expressa o erro de quantização. A Figura 1 mostra a função de transferência de um conversor A/D de 3 bits e seu erro de quantização.

2.1.2 Resolução

A resolução se refere ao número de níveis de quantização em que um sinal de entrada pode ser representado. Normalmente a resolução é expressa pela quantidade de bits do conversor.

2.1.3 Número efetivo de bits

O número efetivo de bits (ENOB) é a medida de desempenho do conversor sob condições dinâmicas. Devido ao ruído de quantização, um conversor real de n bits possui um ENOB inferior a n bits. Os efeitos acumulativos das muitas fontes de erros (não linearidades, ganho, códigos perdidos e outras) contribuem para diminuir o número efetivo de bits. A Equação (2.1) mostra como é calculado o ENOB do conversor.

$$ENOB = \frac{SNR - 1,76}{6.02} \quad (2.1)$$

Onde:

SNR é a relação sinal-ruído do conversor que será explicada posteriormente.

2.1.4 Erro de não linearidade diferencial

A não linearidade diferencial (DNL) é a medida do quão uniforme são os níveis de quantização da curva de transferência do conversor. Cada nível de quantização é comparado com o nível de quantização ideal e a magnitude da diferença entre eles denota o erro de DNL, como mostrado na Figura 2.

Um aspecto importante na determinação do DNL é a definição do nível de quantização ideal. Para alguns fabricantes o nível de quantização é definido por $\frac{FS}{2^n - 1}$, onde FS é o fundo de escala do conversor. Nessa definição, o nível de quantização será igual ao longo de toda a curva de transferência, uma vez que é calculado à partir de parâmetros independentes da função de transferência. Outros fabricantes utilizam uma definição conhecida por nível de quantização do dispositivo sob teste, que é definida como $\frac{\text{primeiro nível de quantização} - \text{ultimo nível de quantização}}{2^n - 2}$. Essa definição é utilizada para retirar os efeitos dos erros de ganho e offset. Dispositivos especificados segundo essa definição de nível de quantização possuirão um menor erro de DNL.

Algumas características importantes do conversor podem ser inferidas à partir de seu DNL.

- Se o erro de DNL for inferior a $1LSB$, o conversor não possuirá códigos perdidos.
- Se o erro de DNL for inferior a $0,5LSB$, o conversor será monotônico.

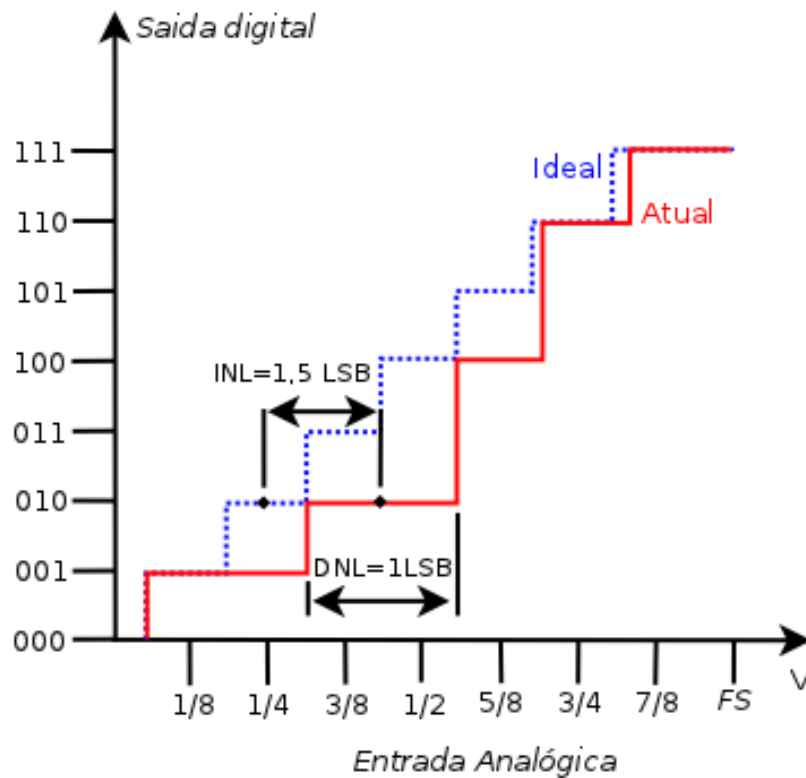


Figura 2: Erros de DNL e INL em um conversor A/D de 3 bits.

A Equação (2.2) define o erro de DNL para cada nível quantização.

$$\begin{aligned}
 DNL &= \text{Largura do LSB ideal} - \text{Largura do código} \\
 &= 1 - \frac{V(x) - V(x+1)}{LSB}
 \end{aligned}
 \tag{2.2}$$

Onde:

$V(x)$ e $V(x+1)$ são dois níveis de quantização adjacentes para o código x .

2.1.5 Erro de não linearidade integral

A não linearidade integral (INL) é o desvio dos pontos médios dos códigos das suas posições ideais, como ilustrado na Figura 2. São utilizados três métodos para se calcular a posição ideal dos pontos médios dos códigos.

- desenhar uma reta entre os pontos médios do primeiro e do último código.
- desenhar uma curva que melhor se adapte usando os pontos médios medidos de todos os códigos.
- desenhar uma reta entre a posição ideal dos pontos médios do primeiro e do último

código e usar o LSB ideal para calcular a localização do pontos médios dos outros códigos.

O primeiro método geralmente resulta em um menor valor para o erro de INL.

O erro de INL também pode ser entendido como a soma dos erros de DNL ao longo da curva de transferência do conversor, uma vez que o erro de DNL ocorre em cada um dos níveis de quantização.

2.1.6 Função de transferência do conversor

A função de transferência do conversor A/D mapeia a entrada analógica em um código digital de saída. A Figura 1 ilustra a função de transferência de um conversor unipolar de 3 bits e é idealizada no sentido em que a natureza dos níveis de transição é probabilística e não tão bem definida como ilustrado. Outro ponto importante a ser notado é que uma faixa de entradas analógica resulta em um mesmo código digital.

2.1.7 Monotonicidade

Essa característica assume que quando ocorrido um aumento na tensão de entrada, o código digital de saída não diminua. Essa característica é crítica em aplicações de controle pois pode resultar em uma operação instável.

2.1.8 Perda de código

Um conversor deve ser capaz de realizar a correspondência para todos os códigos digitais de saída. Se o conversor não é capaz dessa realização, é dito haver códigos perdidos.

2.1.9 Relação sinal-ruído

A relação sinal-ruído (SNR) é caracterizada através da amostragem de um sinal senoidal puro e da realização da transformada rápida de Fourier (FFT) nos dados coletados. Assim, SNR é a taxa da magnitude da frequência fundamental em relação ao valor RMS de todas as outras frequências, incluindo as harmônicas. Devido ao ruído de quantização, a SNR de um conversor A/D ideal é dada pela Equação (2.3).

$$SNR = 6,02ndB + 1,76dB \quad (2.3)$$

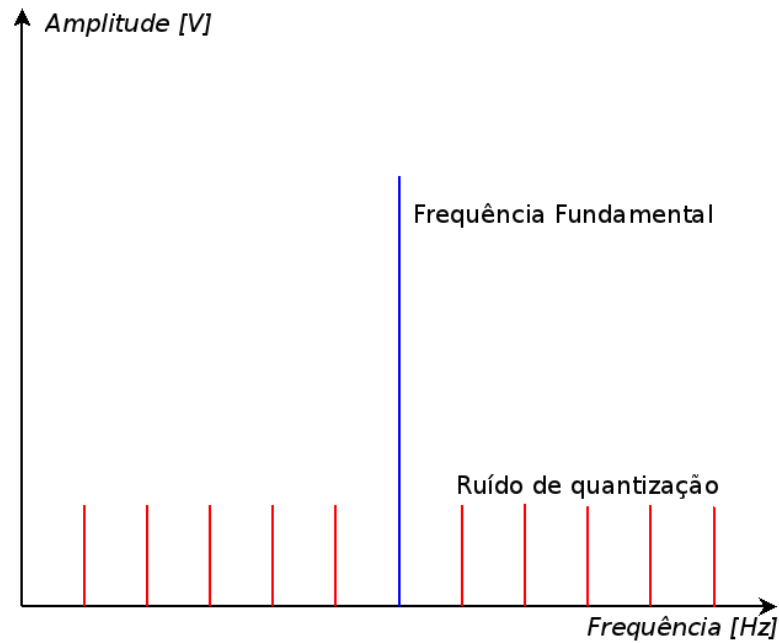


Figura 3: Espectro de frequências de um sinal senoidal convertido por um ADC ideal.

onde n representa o número de bits do conversor.

Com exceção do ruído de quantização, a transformada indica que a forma de onda discreta de saída do conversor possui apenas a componente harmônica fundamental correspondente ao sinal de entrada. Se o conversor possuir não linearidades, as mesmas serão indicadas como componentes harmônicas diferentes da fundamental. Dessa maneira, examinando-se o espectro de frequências da forma de onda discreta, pode-se determinar as não linearidades do conversor.

A Figura 3 mostra o espectro de frequências para um conversor ideal com um sinal senoidal ideal aplicado em sua entrada. Nota-se que mesmo no caso ideal, é possível notar a presença do ruído de quantização espalhado pelas faixas de frequência na forma de ruído branco aleatório.

A Figura 4 ilustra o espectro de frequências para um conversor que possui não linearidades. Como o sinal de entrada será distorcido, a DFT apresentará componentes harmônicas em frequências diferentes da fundamental e também um nível de ruído branco mais elevado.

Para a realização desse tipo de teste, alguns cuidados devem ser tomados. O primeiro é com relação à amplitude do sinal de entrada, que deve ser quase igual à faixa de entrada do conversor, sem exceder-la para que não haja distorção. Entretanto, a amplitude deve ser grande o suficiente para garantir que todos os códigos sejam avaliados. O segundo

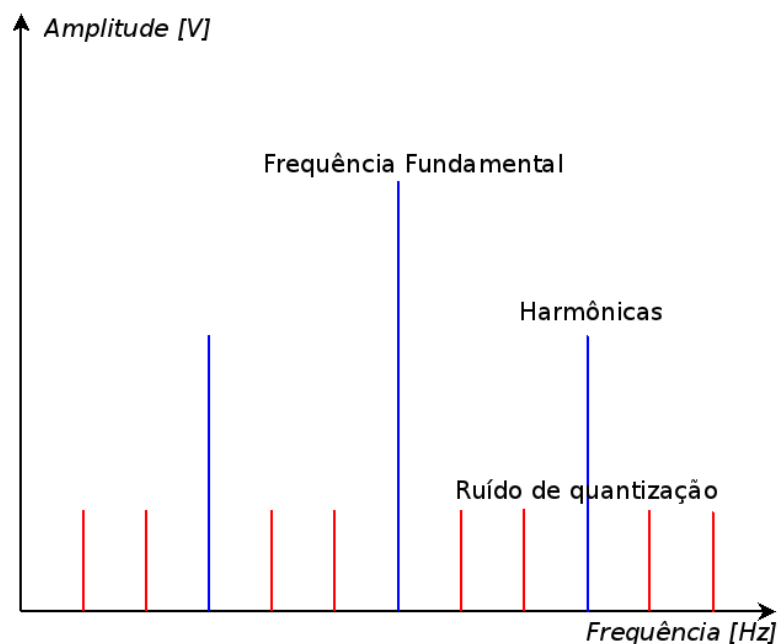


Figura 4: Espectro de frequências de um sinal senoidal convertido por um ADC que possui não linearidades.

cuidado a ser tomado se refere a quantidade de dados amostrados. Deve-se garantir que um número inteiro de ciclos do sinal de entrada seja amostrado, caso contrário, ocorrerão componentes harmônicas diferentes da fundamental devido à diferença de amostragem. Essas componentes poderão ser erroneamente interpretadas como não linearidades do conversor.

O terceiro cuidado a ser tomado refere-se à frequência de amostragem, que deve respeitar o critério de Nyquist para garantir que não ocorra efeito de aliasing. Como referência mais concreta, [7] traz as condições necessárias para a correta aplicação da DFT.

(**C₁**) O sinal a ser amostrado $x(t)$ deve ser periódico:

$$x(t) = x(t + \tau) \forall t \quad (2.4)$$

(**C₂**) O sinal deve ser amostrado com um período de amostragem T , que deve ser submúltiplo do período τ do sinal:

$$x_s(t) = x(t) \cdot d(t, T) T = \frac{\tau}{N} \quad (2.5)$$

Onde

x_s denota o sinal de controle de amostragem;

N é a relação de multiplicidade com o sinal a ser amostrado.

(**C₃**) A frequência de amostragem deve ser maior que o dobro da maior frequência do sinal a ser amostrado $x(t)$.

(**C₄**) Devem estar disponíveis N amostras do sinal.

Como regra prática, a Equação (2.6) pode ser utilizada para o cálculo da frequência do sinal de entrada. Dessa maneira, as frequências resultantes do efeito de aliasing serão múltiplas da frequência de entrada, como resultado, o erro de quantização possuirá a forma de tons discretos.

$$f_{in} = m \times \frac{f_s}{N_a} \quad (2.6)$$

Onde

m representa o número de janelas utilizadas para representar o sinal de entrada e deve ser um número inteiro e primo;

f_s representa a frequência de amostragem (em respeito a condição C_3);

N_a indica o número de amostras utilizadas para representar o sinal e deve ser uma potência de 2.

2.2 Arquitetura SAR

A Figura 5 ilustra o diagrama de blocos da conversão por aproximação sucessiva. Essa arquitetura é composta de um bloco SAR que representa a lógica e controle da aproximação sucessiva, de um bloco DAC que representa um estágio de conversão Digital-analógico e de um bloco S/H que representa um circuito de sample-hold. O diagrama também ilustra um circuito comparador que possui sua entrada positiva acoplada ao circuito de sample-hold e sua entrada negativa realimentada com o sinal oriundo do estágio DAC. O processo de conversão por aproximação sucessiva consiste em, iniciando-se pelo bit mais significativo, testar um a um os bits da palavra digital aplicada ao conversor digital-analógico (D/A). Esses testes iniciam-se sempre com o valor do bit em avaliação em nível lógico alto, e a cada passo do procedimento de conversão, o valor da saída do conversor D/A é comparado com o valor do sinal analógico de entrada. Se o valor de saída do conversor D/A for maior que o valor do sinal de entrada, o bit em teste recebe o valor lógico baixo e o procedimento tem continuidade aplicando-se um nível lógico no próximo bit que será avaliado. Se o valor do sinal de entrada for maior, o bit em teste continua em nível lógico alto. Esse procedimento ocorre até que o último bit da palavra digital seja testado e seu valor lógico seja corretamente determinado.

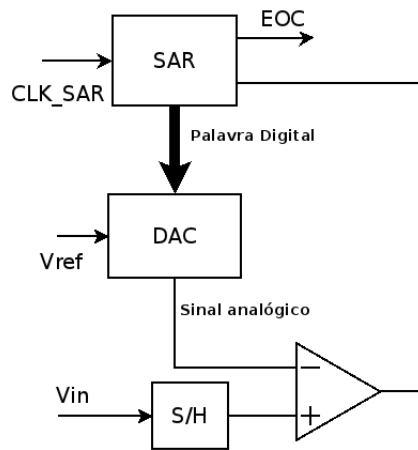


Figura 5: Diagrama de blocos da arquitetura SAR de conversão A/D.

A arquitetura SAR apresenta algumas características importantes:

- Necessita de n passos para realizar uma conversão de n -bits de um sinal analógico de entrada.
- A saída pode ser serial, quando tomada à partir da saída do circuito comparador, ou paralela, quando tomada à partir do registrador de aproximação sucessiva.
- Sua taxa de conversão (n ciclos de *clock*) é muito maior que a de uma arquitetura de contador em rampa (2^n ciclos de *clock*).

2.3 Modulação por largura de pulso (PWM)

A modulação por largura de pulso, mais conhecida por PWM (Pulse Width Modulation) é largamente utilizada em diversas áreas da engenharia elétrica, como por exemplo, em telecomunicações onde as diferentes larguras dos pulsos representam diferentes valores de dados codificados.

Outra aplicação encontra-se na área de transferência de potência, onde aplica-se esse tipo de modulação afim de minimizar as perdas em percursos resistivos.

Esse tipo de modulação também é empregada na regulação de tensão, onde se consegue o nível desejado de tensão através da variação do ciclo de trabalho do sinal modulado. Esse é o princípio utilizado nesse trabalho.

2.3.1 Conceito

Modulação significa a alteração de alguma característica de uma tensão ou corrente em resposta a uma alteração em outra tensão ou corrente [6]. Na modulação por largura de pulso, a largura dos pulsos de uma forma de onda quadrada de saída (sinal modulado) é alterada segundo a variação da amplitude da forma de onda de entrada (sinal modulante).

No presente trabalho, o sinal modulante é substituído por uma palavra digital. Assim, a largura do pulso do sinal modulado varia de acordo com o valor da palavra digital de entrada.

2.3.2 Caracterização do sinal PWM

A teoria de Fourier assegura que qualquer onda periódica pode ser decomposta na soma de suas infinitas harmônicas de frequências múltiplas inteiras. Assim, a representação em série de Fourier de um sinal modulado PWM pode ser simplificada posicionando-se arbitrariamente a origem do eixo das abcissas de maneira que o sinal se torne uma função matemática par, como mostrado na Figura 6, onde p representa o ciclo de trabalho do sinal, k representa sua amplitude e T o seu período.

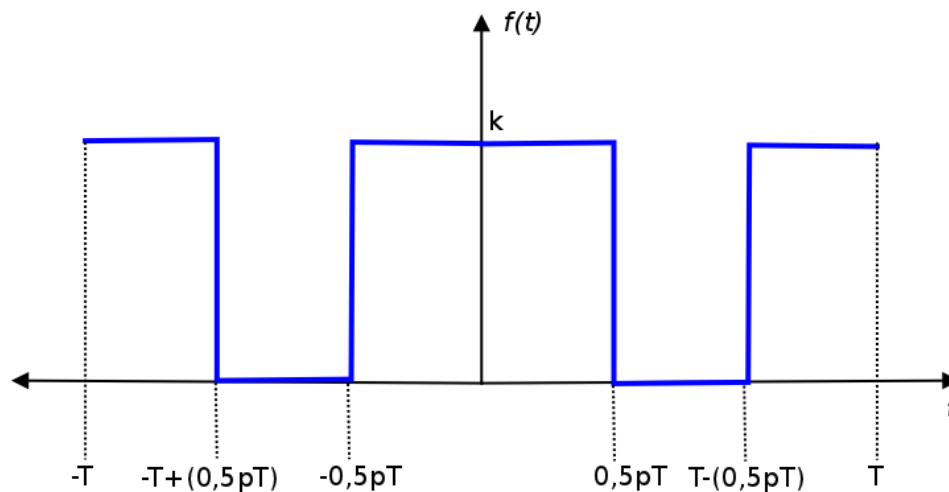


Figura 6: Sinal PWM deslocado no tempo para a obtenção de uma função par.

A representação em série de Fourier de uma função periódica par é dada pela Equação (2.7) [5].

$$f(t) = A_0 + \sum_{n=1}^{\infty} [A_n \cos(\frac{2n\pi t}{T}) + B_n \sen(\frac{2n\pi t}{T})] \quad (2.7)$$

sendo:

$$A_0 = \frac{1}{2T} \int_{-T}^T f(t) dt \quad (2.8)$$

$$A_n = \frac{1}{2T} \int_{-T}^T f(t) \cos\left(\frac{2n\pi t}{T}\right) dt \quad (2.9)$$

$$B_n = \frac{1}{2T} \int_{-T}^T f(t) \operatorname{sen}\left(\frac{2n\pi t}{T}\right) dt \quad (2.10)$$

onde:

A_0 representa o termo fundamental da Equação.

A_n representa as harmônicas de ordem ímpar.

B_n representa as harmônicas de ordem par.

Sendo k a amplitude do sinal PWM, as Equações (2.11), (2.12) e (2.13) representam os resultados das integração das Equações (2.8), (2.9) e (2.10), respectivamente.

$$A_0 = kp \quad (2.11)$$

$$A_n = k \frac{1}{n\pi} [\operatorname{sen}(n\pi p) - \operatorname{sen}(2n\pi(1 - \frac{p}{2}))] \quad (2.12)$$

$$B_n = 0 \quad (2.13)$$

O resultado de B_n já era esperado por se tratar de uma função par. A_0 (componente D.C.) pode ser representada pela multiplicação da amplitude do sinal PWM com o seu ciclo de trabalho p . Nota-se que escolhendo-se apropriadamente o valor do ciclo de trabalho, pode-se obter qualquer valor D.C. entre 0 e k . A componente A_n representa a amplitude das harmônicas de alta frequência, as quais ocorrem em frequências múltiplas inteiras da frequência fundamental.

3 Modelagem

Baseado nas revisões teóricas do capítulo anterior, esse capítulo traz a proposta de uma arquitetura de conversão A/D. Inicialmente será apresentada uma visão sistêmica através do seu diagrama de blocos. Posteriormente, serão feitas a análise e a modelagem do funcionamento de cada estágio do sistema, onde os estágios digitais são modelados em linguagem de descrição de hardware VHDL e os estágios analógicos são modelados com o auxílio da ferramenta MatLab.

O sistema completo é então validado através de simulações. Esse capítulo traz ainda considerações sobre a implementação do circuito integrado e também os esquemáticos utilizados em cada estágio do projeto.

3.1 Considerações iniciais

Antes de iniciar a modelagem do sistema, são necessárias algumas considerações que determinaram o modelo. A primeira delas trata do *design kit* disponível para o projeto. Foi utilizado um *design kit* gratuito desenvolvido pela Universidade Estadual da Carolina do Norte (NCSU).

Foi utilizada a versão NCSU CDK 1.5.1 compatível com a ferramenta Cadence Virtuoso 5.41. Nesta versão do *design kit* não estão disponíveis células básicas (*standard cells*) nem o conjunto de regras necessárias para o roteamento automático do circuito, o que possibilitaria maior agilidade e eficiência no projeto de circuitos digitais. Dessa maneira esse design kit é indicado para projetos analógicos ou digitais utilizando a metodologia *full custom* onde cada transistor é desenvolvido e posicionado individualmente. Sendo assim, os códigos em VHDL não puderam ser utilizados no projeto final do circuito, entretanto foram utilizados para uma validação inicial da arquitetura proposta através do MatLab, bem como para a verificação do equacionamento do modelo.

Outra consideração importante é sobre a tecnologia disponível. O projeto foi desen-

volvido utilizando o processo ON C5 (SCMOS_SUBM) que permite duas camadas de poly e três camadas de metalização. Esse processo foi escolhido pois permite a fabricação gratuita através de um convênio com a MOSIS que visa projetos educacionais. Nessa tecnologia, os circuitos alimentados com 2,5V conseguem trabalhar em uma frequência máxima em torno de 2 MHz, limitando assim o conversor proposto a trabalhar com uma taxa máxima de conversão de aproximadamente 200 S/s. Essa baixa taxa de conversão implica em capacitores grandes na etapa de filtragem, assim foi decidido que a etapa de filtragem seria implementada fora do circuito integrado para que testes de velocidade pudessem ser feitos variando-se a tensão de alimentação.

De acordo com essas considerações, as próximas seções apresentam os esquemáticos e a análise funcional dos circuitos utilizados no projeto final do conversor

3.2 Sistema proposto

Como visto no Capítulo 2, a arquitetura SAR necessita de uma etapa de conversão D/A. O projeto está diretamente ligado com as não linearidades diferencial (DNL) e integral (INL) [8], assim a ideia principal é desenvolver um estágio D/A em que essas não linearidades sejam minimizadas. Isso é obtido através de um modulador PWM trabalhando em conjunto com um filtro passa baixas de primeira ordem. A Figura 7 ilustra o sistema proposto, onde os blocos tracejados são puramente digitais e foram modelados em linguagem de descrição de hardware VHDL. A codificação de tais blocos se encontra no Apêndice A.

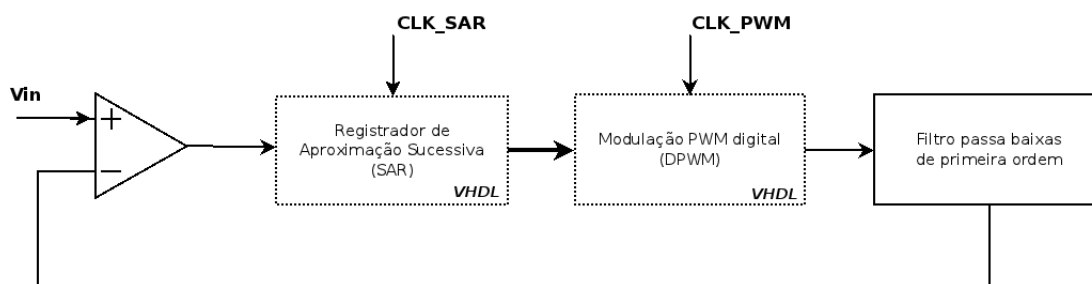


Figura 7: Diagrama de blocos do sistema proposto.

Como visto no capítulo anterior, o sinal PWM pode ser representado pela somatória das Equações (2.11) e (2.12), que representam a componente D.C. A_0 e as componentes harmônicas de alta frequência A_n , respectivamente. Foi visto também que selecionando-se adequadamente o valor do ciclo de trabalho do sinal modulado PWM, pode-se obter qualquer valor de A_0 , variando entre 0 e sua amplitude k .

De acordo com esses princípios, é possível decompor o sinal PWM em um nível D.C. e uma onda quadrada de valor médio zero [9], como mostrado na Figura 8, e filtrando-se apropriadamente a onda quadrada, tem-se um conversor D/A. A Figura 9 ilustra essa condição.

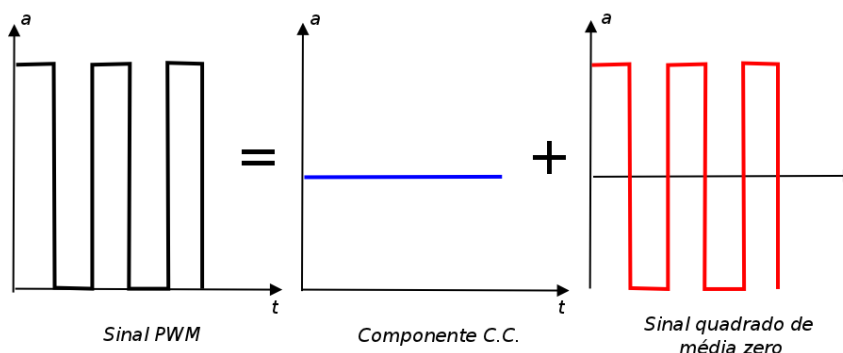


Figura 8: Sinal *PWM* decomposto.

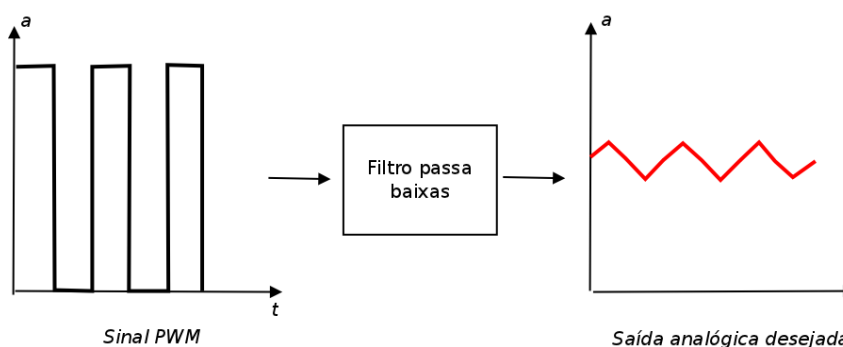


Figura 9: Procedimento de filtragem do sinal *PWM*.

Assim, a palavra digital que controla a largura dos pulsos é alterada a cada ciclo de clock do registrador de aproximação sucessiva, até que a conversão seja realizada. Essa palavra digital pode ser entendida como uma representação dos níveis D.C. obtidos após a etapa de filtragem. Dessa maneira, uma palavra com maior quantidade de bits consegue produzir uma maior quantidade de valores D.C., uma vez que o ciclo de trabalho pode ser variado em passos menores. A Tabela 1 ilustra os dezesseis níveis de variação do ciclo de trabalho possíveis para uma palavra de quatro bits.

Nota-se que o passo do ciclo de trabalho se relaciona com o número de bits da palavra digital segundo a Equação (3.1).

$$CT = \frac{1}{2^N} \times 100 \quad (3.1)$$

onde:

N representa o número de bits da palavra digital de controle.

Tabela 1: Variação do ciclo de trabalho para uma palavra digital de quatro bits.

| Palavra | Ciclo de Trabalho(%) |
|---------|----------------------|
| 0000 | 0,00 |
| 0001 | 6,25 |
| 0010 | 12,50 |
| 0011 | 18,75 |
| 0100 | 25,00 |
| 0101 | 31,25 |
| 0110 | 37,50 |
| 0111 | 43,75 |
| 1000 | 50,00 |
| 1001 | 56,25 |
| 1010 | 62,50 |
| 1011 | 68,75 |
| 1100 | 75,00 |
| 1101 | 81,25 |
| 1110 | 87,50 |
| 1111 | 93,75 |

A seguir será modelado um conversor A/D de quatro bits baseado na arquitetura proposta que possui a finalidade de validá-la por meio de simulações.

3.3 Registrador de Aproximação Sucessiva

A lógica de aproximação sucessiva faz com que cada bit da palavra digital de saída seja testado a cada ciclo do sinal de *clock*. Assim os bits vão sendo determinados um a um, iniciando-se sempre pelo bit mais significativo e seguindo a sequência de testes até o bit menos significativo.

Dessa maneira, para a implementação da lógica de aproximação sucessiva é proposto o circuito lógico da Figura 10. O mesmo é composto de um registrador de deslocamento [6], responsável por habilitar cada bit da palavra digital no seu ciclo de *clock* correspondente. Também há um conjunto de flip-flops do tipo D responsáveis por armazenar o valor atual entrada de seleção.

Ocorrendo um sinal de *reset*, o registrador de deslocamento é iniciado com o valor 10000 enquanto os flip-flops do tipo D são iniciados com valor 0000. A lógica combinacional OR garante que o valor inicial seja obtido na saída. Dado o primeiro pulso do

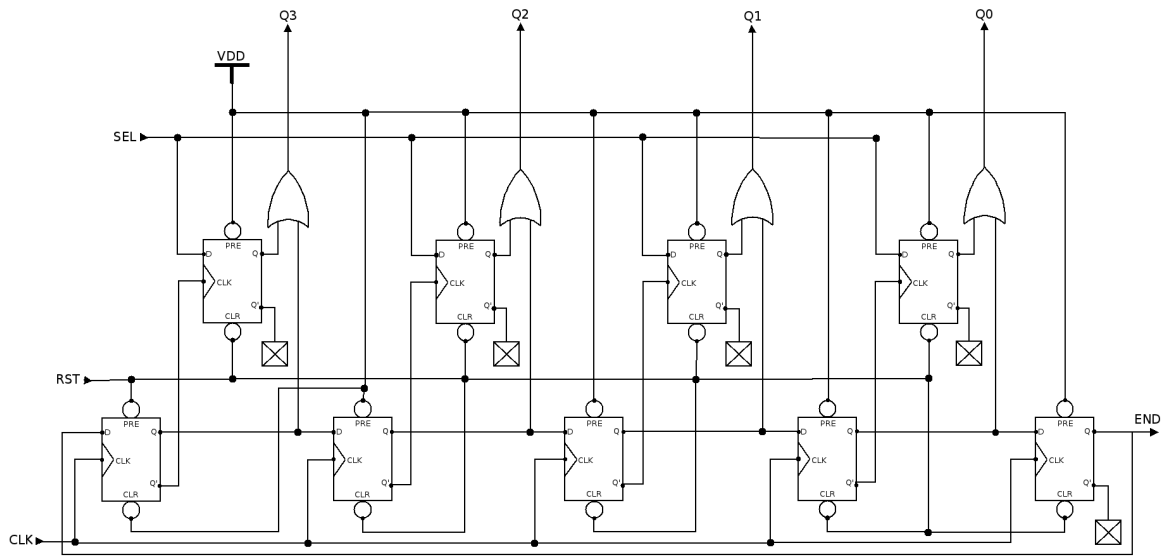


Figura 10: Esquema lógico do Registrador de aproximação sucessiva.

signal de *clock*, o valor do registrador de deslocamento é alterado para 01000 enquanto os flip-flops de saída possuem o mesmo valor inicial, exceto o flip-flop que representa o bit mais significativo, pois esse já teve seu valor avaliado e determinado.

Assim, a saída do registrador de aproximação sucessiva deve possuir o valor X100, onde o *X* indica o valor já determinado. A ideia é que em cada ciclo de *clock* necessário para a conversão, seja avaliado sempre o maior valor possível de saída do registrador.

A particularidade do SAR nessa aplicação, é a necessidade de um flip-flop adicional no registrador de deslocamento para indicar o final da conversão dos dados. Assim, o conversor modelado finaliza a conversão do dado na borda de subida do quinto pulso de *clock*, podendo o mesmo ser lido à partir desse momento.

A Tabela 2 resume os sinais envolvidos e suas funções. Enquanto as Figuras 11 e 12 ilustram o funcionamento do registrador para os valores 0 e 1, respectivamente.

Conforme o esperado, com a entrada SEL em nível lógico baixo, o valor 1 do bit mais significativo é deslocado até o bit menos significativo enquanto os outros bits são preenchidos com valor 0. Com a entrada SEL em nível lógico alto, o valor 1 do bit mais significativo também continua sendo deslocado até o bit menos significativo, entretanto os demais bits vão sendo preenchidos com o valor 1.

Tabela 2: Sinais do registrador de aproximação sucessiva

| Pino | Função |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| SEL | Entrada do bit de seleção. |
| CLK | Entrada do sinal de relógio. |
| RST | Sinal de inicialização dos flip-flops. Coloca a palavra 10000 nos registradores de deslocamento e a palavra 0000 nos flip-flops tipo D de saída. |
| Q₃ a Q₀ | Saída paralela do valor dos registradores. |
| END | Indica o final da conversão. |

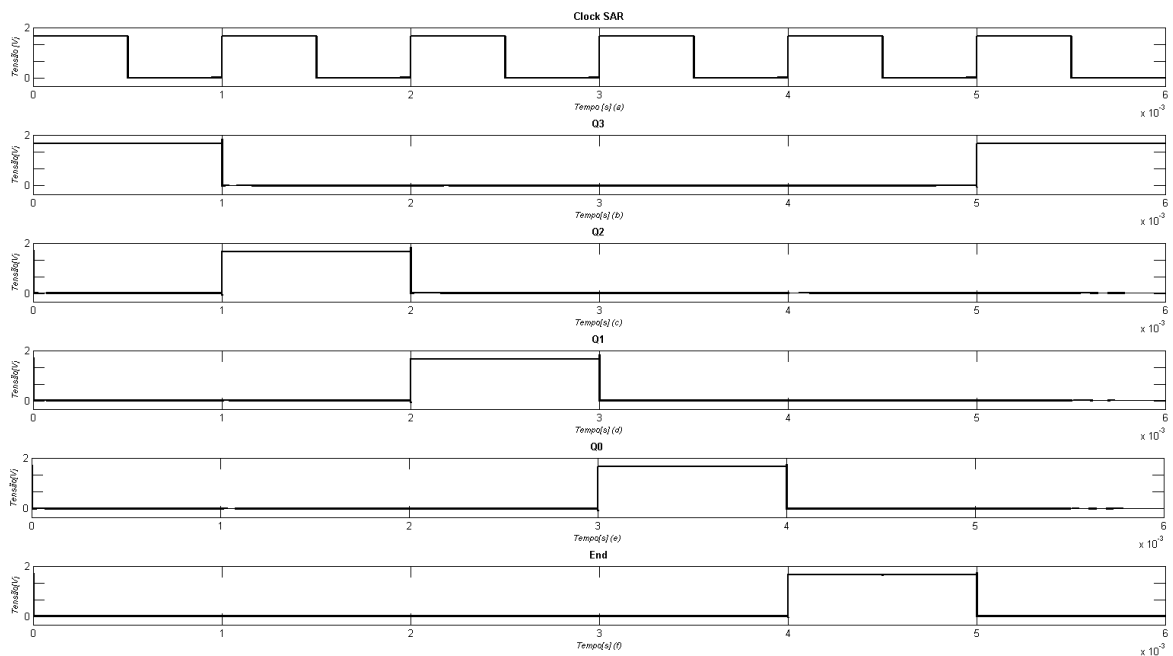


Figura 11: Operação do Registrador de Aproximação Sucessiva com a entrada SEL em nível lógico baixo.

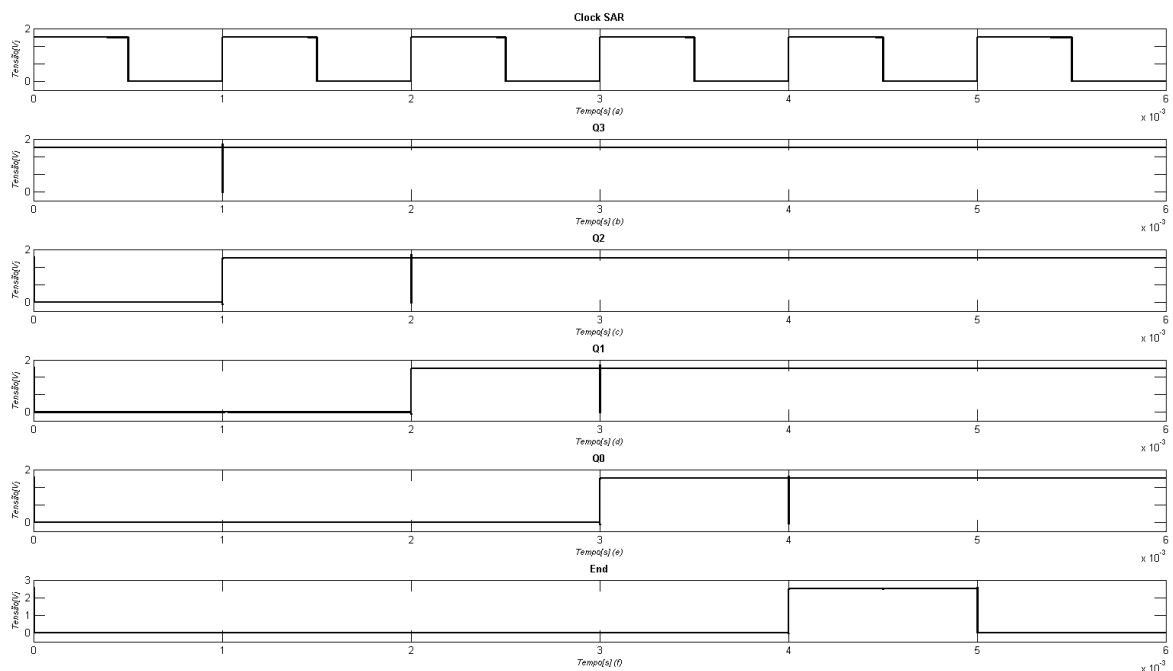


Figura 12: Operação do Registrador de Aproximação Sucessiva com a entrada SEL em nível lógico alto.

3.4 Modulador PWM Digital

Como anteriormente explicado, a etapa de conversão Digital-Analógico presente em um conversor A/D com aproximação sucessiva convencional foi substituída por um circuito digital capaz de variar a largura de seu sinal de saída de acordo com o a palavra digital em sua entrada. Uma etapa de filtragem passiva é posteriormente utilizada para recuperar o valor médio do trem de pulsos gerado pelo modulador PWM.

A Figura 13 ilustra a lógica utilizada no circuito modulador PWM que é composta de um contador, de um registrador e de um comparador. Enquanto a contagem do contador não alcança o valor presente no registrador, a saída do comparador envia nível lógico alto. Quando a contagem atinge o valor presente no registrador a saída do comparador apresenta nível lógico baixo.

A Figura 14 ilustra o circuito evidenciando seus sub circuitos. Podem ser notados os registradores, o contador síncrono implementado com flip-flops do tipo T, a lógica de reset do circuito bem como a lógica de comparação, responsável por gerar o sinal PWM em sua saída.

Assim que o circuito é iniciado por um pulso de *reset* (RST), o contador é preenchido com a palavra digital 0000 e os registradores recebem a palavra digital presente na entrada do modulador PWM. A cada pulso de *clock* (CLK) o contador sofre um incremento. A

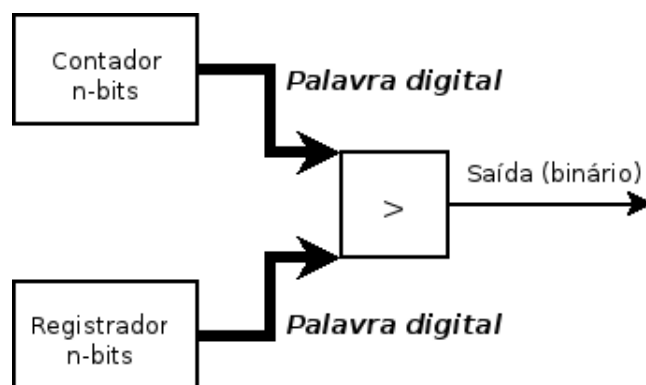


Figura 13: Lógica do circuito modulador PWM.

lógica de comparação garante que a saída (PWM) permaneça em nível lógico alto enquanto o contador não alcançar o valor contido nos registradores. Assim que isso acontece, o sinal de saída recebe nível lógico baixo e a contagem prossegue até que o contador seja reiniciado.

A lógica de *reset* assegura que toda vez que a contagem termine, ocorra um pre-set no flip-flop que produz o sinal de saída, colocando a saída em nível lógico alto. Nesse mesmo instante os registradores são atualizados com a palavra digital que está presente na entrada do modulador PWM. É necessário um flip-flop D na lógica de *reset* para que haja um sincronismo do sinal de saída da porta AND com o sinal de *clock*, uma vez que os sinais de entrada da porta AND chegam em instantes diferentes.

É importante notar que os registradores fazem com que o circuito modulador PWM opere de maneira independente do registrador de aproximação sucessiva, uma vez que seus valores são atualizados de acordo com o sinal gerado pela lógica de *reset* e não quando ocorre uma alteração no valor do registrador de aproximação sucessiva.

Outra característica importante desse circuito é trabalhar com uma frequência de *clock* muito superior à frequência de *clock* do registrador de aproximação sucessiva, caracterizando assim uma espécie de sobreamostragem do valor contido no mesmo. Essa característica será melhor explicada no equacionamento do sistema. A Tabela 3 resume os sinais do modulador de largura de pulso e suas funções, enquanto as Figuras 15 e 16 ilustram o sinal PWM e a tensão de quantização correspondente aos códigos digitais 0001 e 1000.

Observando-se as Figuras 15 e 16, nota-se que o circuito responde da maneira esperada, ou seja, um aumento na palavra digital de entrada se reflete em um aumento na largura do pulso do sinal PWM de saída do circuito.

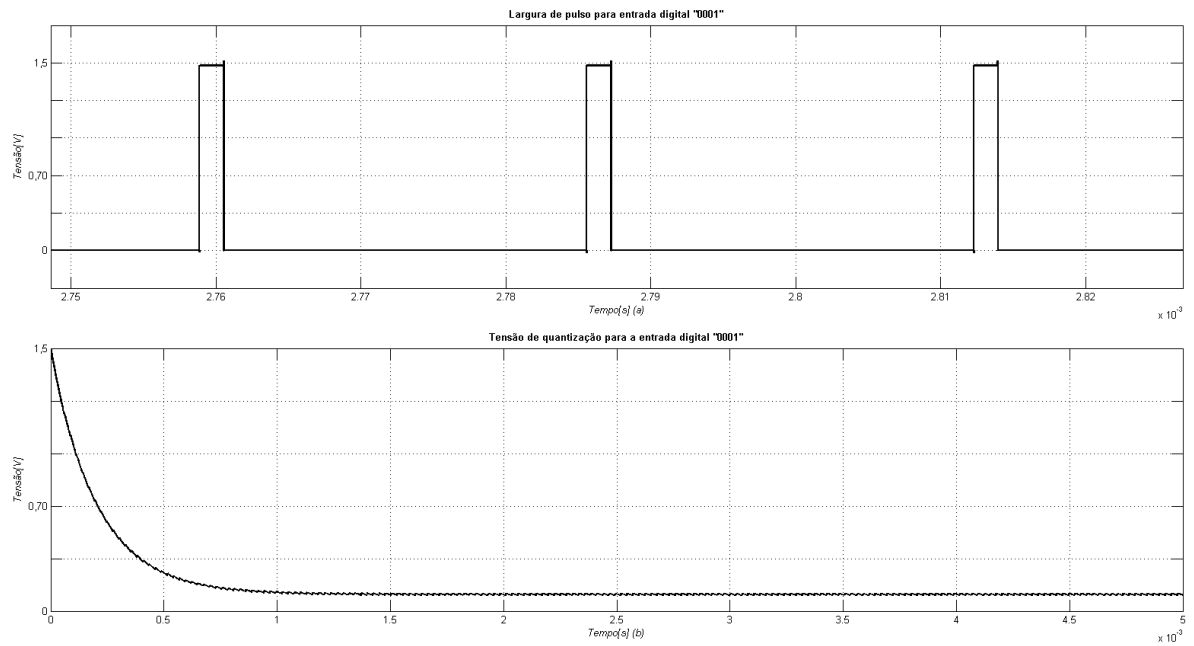


Figura 15: Sinal PWM para uma entrada digital 0001 e a tensão de quantização após a filtragem.

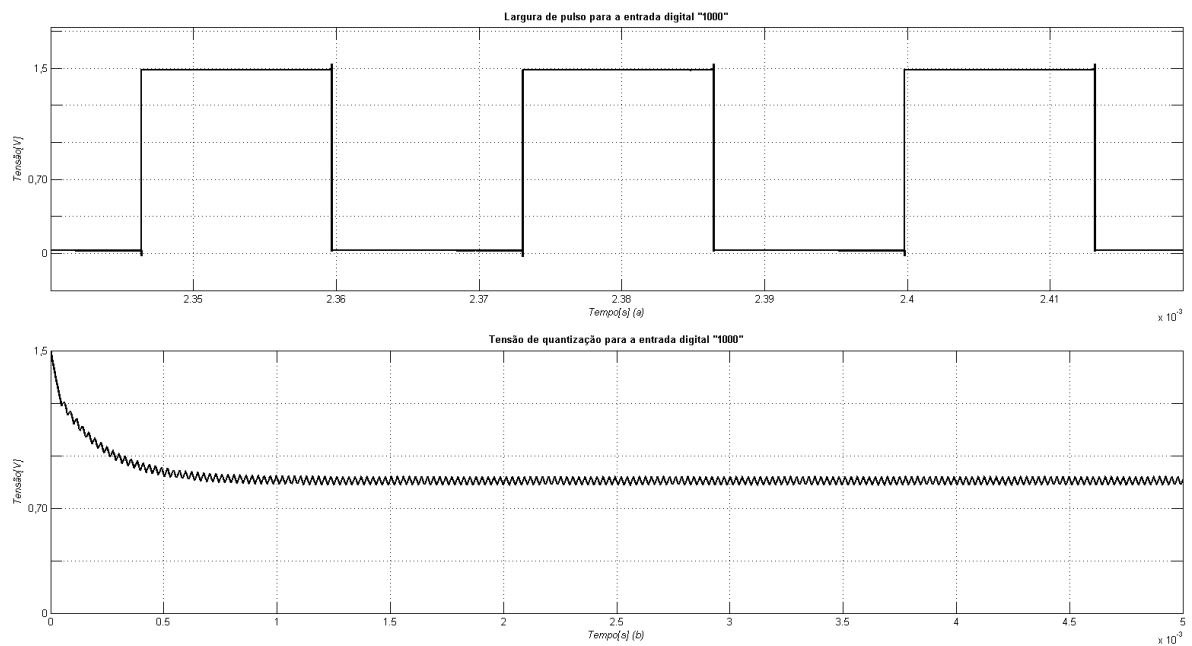


Figura 16: Sinal PWM para uma entrada digital 1000 e a tensão de quantização após a filtragem.

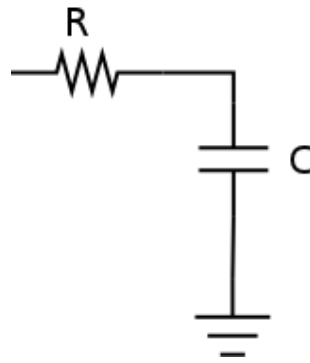


Figura 17: Esquemático do filtro passivo de primeira ordem.

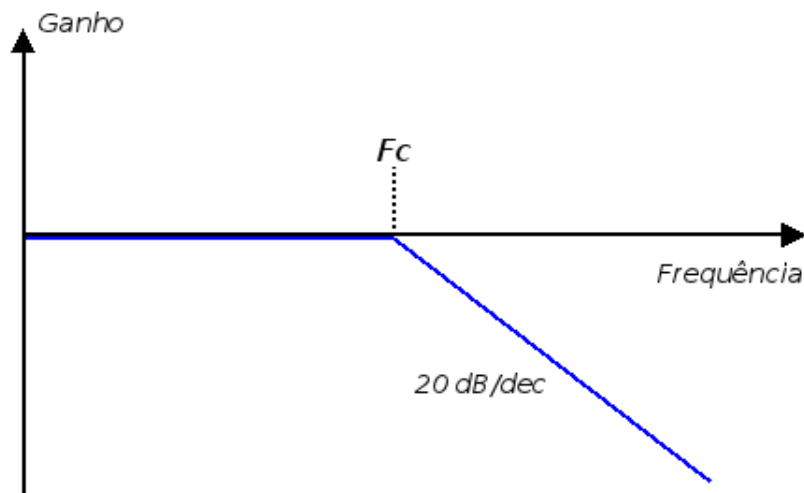


Figura 18: Resposta do filtro de primeira ordem.

3.5 Filtro de primeira ordem

A modelagem do filtro de primeira ordem foi realizada no Simulink através do bloco *transfer function*. Entretanto para a implementação foi utilizado um filtro resistivo-capacitivo simples, como mostrado na Figura 17. A Figura 18 ilustra a curva característica do filtro, onde F_c é sua frequência de corte.

A ideia principal é projetar o filtro para que se tenha uma determinada atenuação em uma dada frequência de operação do sinal a ser filtrado.

3.6 Comparador

No estágio comparador, foi utilizada uma arquitetura de baixo consumo[10], apresentada na Figura 19. Nota-se a necessidade de dois sinais de *clock* defasados, garantindo que as chaves S_1 , S_2 e S_3 não sejam fechadas no mesmo instante.

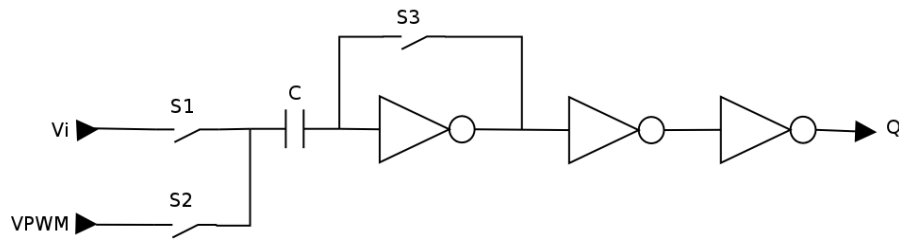


Figura 19: Circuito do estágio comparador.

No instante ϕ_1 a chave S_2 está aberta e as chaves S_1 e S_3 são fechadas, fazendo com que o capacitor C seja carregado com $V_{in} - V_t$, onde V_t é a tensão de threshold do circuito inversor. Dessa maneira, qualquer variação da tensão em um instante ϕ_2 será entendido pelo circuito inversor de saída. Dessa maneira, no instante ϕ_2 as chaves S_1 e S_3 estão abertas e a chave S_2 é fechada, fazendo com que o sinal produzido pelo modulador PWM seja aplicado na entrada do circuito comparador. O capacitor C garante que o circuito inversor esteja sempre no limiar de operação, assim, o circuito comparador toma uma decisão rapidamente.

As chaves S_1 , S_2 e S_3 foram substituídas por chaves de estado sólido baseadas em um transistor nMOS. O sinal de clock é ligado ao gate desses transistores enquanto seu substrato é ligado à referência. Ocorrendo um nível lógico alto no sinal de clock, os transistores são fechados e os sinais presentes nas fontes dos mesmos são transmitidos. Ocorrendo um nível lógico baixo no sinal de clock, os transistores são abertos e consequentemente os sinais presentes em suas fontes não são transmitidos.

Entretanto, os sinais a serem transmitidos são da mesma ordem de grandeza do sinal de clock de controle. Isso faz com que as chaves nMOS não conduzam corretamente os sinais de entrada do comparador em toda sua faixa de excursão, pois para tanto, é necessário uma tensão V_{GS} maior que a tensão de threshold V_t . Uma vez que os sinais de entrada do comparador estão ligados diretamente na fonte dos transistores nMOS, a tensão da fonte pode variar de zero até a tensão máxima do sinal de clock V_G , podendo produzir valores de V_{GS} inferiores à tensão de threshold V_t . Para solucionar esses problema foi considerada uma estrutura de booster [11] capaz de dobrar a tensão do colck aplicada ao gate dos transistores nMOS, garantindo assim sua operação correta em toda a faixa de excursão dos sinais de entrada do comparador. As Figuras 20 e 21 ilustram o circuito dobrador de tensão e seu funcionamento, respectivamente.

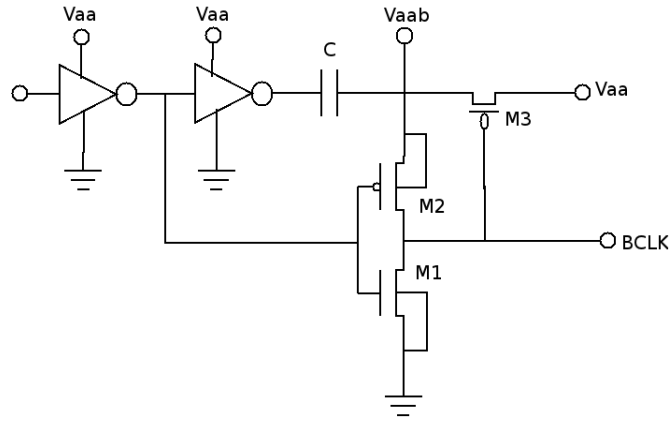


Figura 20: Circuito dobrador de tensão.

A tensão aumentada é mostrada na equação (3.2).

$$V_{aab} = \left(\frac{2 \times C + C_L}{C + C_L} - a \right) \times V_{aa} = B \times V_{aa} \quad (3.2)$$

onde:

C_L é a capacitância de carga,

a é a taxa de descarga em porcentagem de V_{aa} observada em $BCLK$ e V_{aab} devido a uma corrente de carga DC (I_L),

B é o fator de aumento da tensão.

O valor do capacitor C pode ser determinado através da equação (3.3).

$$C = \frac{t \times I_L}{a \times V_{aa}} \quad (3.3)$$

onde:

t é o tempo em que o sinal de clock permanece em nível lógico alto.

A saída $BCLK$ é conectada diretamente ao gate do transistor nMOS enquanto seu substrato é conectado à referência. A Figura 22 ilustra o funcionamento do circuito comparador. A Figura 22 (b) mostra um sinal de referência V_{in} constante ao longo do tempo e um sinal a ser comparado V_{PWM} . Durante os $2ms$ iniciais, o sinal a ser comparado é maior que o sinal de referência e nos $2ms$ finais, o sinal a ser comparado é inferior ao sinal de referência. A Figura 22 (c) mostra a saída do circuito, nota-se que nos primeiros semi-ciclos do sinal de Clock a saída está com o valor da tensão de threshold do inversor enquanto nos semi-ciclos finais a saída possui o valor da comparação entre os sinais.

Nota-se ainda que nos primeiros $2ms$ a comparação resulta em nível lógico baixo e nos últimos $2ms$ a comparação resulta em nível lógico alto, conforme esperado.

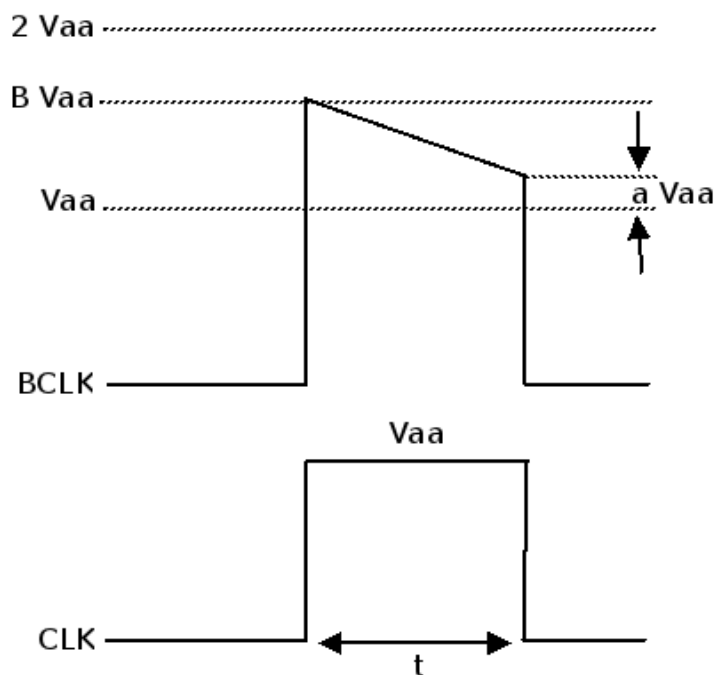


Figura 21: Sinais do circuito dobrador de tensão.

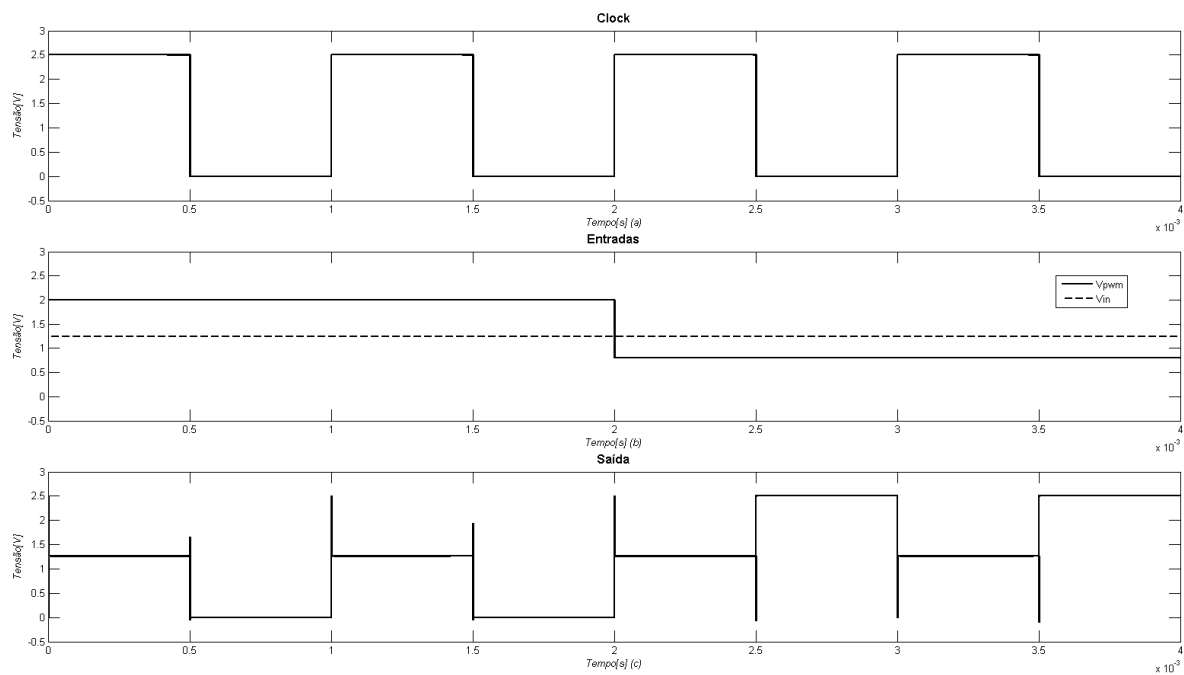


Figura 22: Funcionamento do circuito comparador.

3.7 Equacionamento

Nas seções anteriores foi mostrada a modelagem funcional de cada estágio do sistema proposto, entretanto faltam determinar as características do filtro de primeira ordem e das frequências de *clock* das etapas digitais, bem como definir as bases que regem o funcionamento do sistema completo.

Um sinal é limitado em banda se não contiver componentes em frequências maiores que o limite da banda B . O teorema da amostragem garante que amostras discretas espaçadas uniformemente são uma representação completa do sinal quando sua largura de banda é menor do que a metade da taxa de amostragem.

Considerando-se $x(t)$ um sinal contínuo ao longo do tempo, e considerando-se ainda que $X(f)$ seja sua transformada de Fourier, tem-se.

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-i2\pi ft} dt \quad (3.4)$$

O sinal $x(t)$ será limitado em banda, se $X(f) = 0$ para qualquer $|f| > B$. A condição suficiente para uma exata recuperação do sinal à partir das amostras em uma taxa de amostragem uniforme é dada pela Equação (3.5).

$$f_s \geq 2B \quad (3.5)$$

onde:

f_s representa a taxa de amostragem;

B representa a banda do sinal a ser amostrado.

Sabendo que a conversão deve ser finalizada em um período inferior ou igual a $T \leq \frac{1}{f_s}$, ou seja, todos os N bits devem estar determinados antes que seja realizada uma nova amostragem. A Equação (3.6) define a frequência de trabalho do comparador e do registrador de aproximação sucessiva.

$$f_{SAR} \geq f_s \times N \quad (3.6)$$

onde:

N representa o número de bits do registrador de deslocamento em anel do modelo do registrador de aproximação sucessiva, uma vez que o bit responsável por indicar o final da conversão também deve ser considerado, como já visto anteriormente na etapa de modelagem do registrador de aproximação sucessiva.

Uma vez visto como se determinar a frequência de operação do comparador e do registrador de aproximação sucessiva, deve-se então caracterizar a etapa de filtragem. O sistema proposto utiliza um filtro passivo de primeira ordem, cuja frequência de corte é dada pela Equação (3.7).

$$f_c = \frac{1}{2\pi\tau} \quad (3.7)$$

onde:

f_c representa a frequência de corte;

τ representa a constante de tempo do filtro.

Sabendo-se que um sinal leva aproximadamente cinco constantes de tempo para sua acomodação, a Equação (3.7) pode ser alterada para (3.8).

$$f_c = \frac{1}{2\pi 5\tau} \quad (3.8)$$

Analisando o diagrama de blocos da Figura 7, nota-se que o filtro deve ser capaz de responder mais rapidamente ou no pior caso ao mesmo tempo em que o registrador de aproximação sucessiva determina cada bit da conversão. Assim, a Equação (3.9) determina a máxima constante de tempo permitida para a etapa de filtragem. Assim;

$$\begin{aligned} f_{SAR} &\leq f_c \\ f_{SAR} &\leq \frac{1}{2\pi 5\tau} \\ \tau &\leq \frac{1}{2\pi 5f_{SAR}} \end{aligned} \quad (3.9)$$

Com o filtro já caracterizado, falta apenas determinar a frequência de operação do modulador PWM. Para tanto é necessário primeiramente caracterizar o sinal modulado em PWM antes e depois da etapa de filtragem.

Do Capítulo 2, sabe-se que o sinal PWM pode ser representado pela Equação (3.10).

$$F_{PWM}(t) = A_0 + \sum_{n=1}^{\infty} A_n \cos\left(\frac{2n\pi t}{T}\right) \quad (3.10)$$

Também é sabido que as componentes harmônicas do sinal PWM são dadas segundo a Equação (3.11).

$$g_n(t) = A_n \cos\left(\frac{2n\pi t}{T}\right), n = (0, 1, 2, \dots) \quad (3.11)$$

Dessa maneira, a energia contida em cada componente harmônica de uma função cossenoidal pode ser determinada lembrando-se que a mesma é proporcional ao quadrado da função ($g_n^2(t)$). A energia máxima ocorre quando $\frac{\partial}{\partial p} g_n^2(t) = 0$. Assim:

$$\begin{aligned} \frac{\partial}{\partial p} g_n^2(t) &= \frac{\partial}{\partial p} (A_n^2 \cos^2\left(\frac{2n\pi t}{T}\right)) \\ &= \cos^2\left(\frac{2n\pi t}{T}\right) \frac{\partial}{\partial p} (A_n^2) \\ &= \cos^2\left(\frac{2n\pi t}{T}\right) 2A_n \frac{\partial}{\partial p} (A_n) = 0 \end{aligned} \quad (3.12)$$

O termo cossenoidal é invariante com o ciclo de trabalho p , assim ele não é necessário na igualdade anterior. É importante notar também que o termo $2A_n$ vale zero apenas quando o termo p possui os valores zero ou um. Esse caso representa a energia mínima contida nas componentes harmônicas, onde o sinal PWM possui os valores constantes zero (quando $p = 0$), e k (quando $p = 1$). Dessa maneira, a solução para a energia máxima da Equação (3.12) é obtida igualando-se o termo $\frac{\partial}{\partial p} A_n$ a zero, como descrito na Equação (3.13).

$$\begin{aligned} \frac{\partial}{\partial p} A_n &= \frac{\partial}{\partial p} \left(\frac{1}{n\pi} [\text{sen}(n\pi p) - \text{sen}(2n\pi(1 - \frac{p}{2}))] \right) \\ &= \cos(n\pi p) + \cos(2n\pi(1 - \frac{p}{2})) \\ &= \cos(n\pi p) + \cos(2n\pi - n\pi p) \\ &= \cos(n\pi p) + \cos(2n\pi) \cdot \cos(n\pi p) + \text{sen}(2n\pi) \cdot \text{sen}(n\pi p) = 0 \end{aligned} \quad (3.13)$$

Pode-se observar que o termo $\cos(2n\pi)$ possui valor unitário para qualquer valor de n , e também que o termo $\text{sen}(2n\pi)$ possui valor nulo para qualquer valor de n . Dessa maneira, a Equação (3.13) pode ser reescrita na forma da Equação (3.14)

$$\frac{\partial}{\partial p} A_n = 2\cos(n\pi p) = 0 \quad (3.14)$$

A Equação (3.14) mostra que a energia máxima em cada harmônica é obtida com diferentes ciclos de trabalho. Tomando como exemplo a primeira harmônica ($n = 1$), o valor de p deve ser $p = \frac{1}{2}$ para que o termo cossenoidal tenha valor unitário. Para a quarta harmônica ($n = 4$), a situação de máximo ocorre para dois valores de ciclo de trabalho

$$(p = \frac{1}{4} \text{ e } p = \frac{3}{4}).$$

Sabendo-se que não existe um filtro ideal, após a filtragem, os sinais não são completamente eliminados, ocorrendo apenas sua atenuação. Dessa maneira se faz necessário determinar corretamente sua atenuação para que o sistema funcione, pois essa atenuação está diretamente ligada à amplitude do ripple presente no sinal filtrado.

Como apresentado anteriormente, a componente da primeira harmônica é a que carrega maior quantidade de energia. Dessa maneira, é razoável caracterizar o filtro e também a frequência do sinal PWM para que tal componente seja corretamente atenuada.

Assim, considerando-se a primeira harmônica, tem-se $n = 1$ e considerando-se ainda a condição de maior energia $p = \frac{1}{2}$, isolando-se o termo $A_n \cos(\frac{2n\pi t}{T})$ da Equação (3.10), é obtida a expressão da amplitude máxima do ripple do sinal dada pela primeira componente harmônica, mostrada na Equação (3.15). A Figura 23 mostra as características do sinal filtrado, onde h_1 representa a amplitude do ripple do sinal.

$$h_1 = \frac{2k}{\pi} \cos(\frac{2n\pi t}{T}) \quad (3.15)$$

É importante notar que o termo cossenoidal faz com que a amplitude máxima da primeira componente harmônica varie no intervalo de $-\frac{2k}{\pi} \leq \frac{2k}{\pi}$. A Equação (3.16) mostra a variação máxima de pico a pico da primeira componente harmônica. Assim:

$$h_{1_{pp}} = \frac{2k}{\pi} - (-\frac{2k}{\pi}) = \frac{4k}{\pi} \quad (3.16)$$

Conforme mostrado no Capítulo 2, o sinal filtrado possui níveis determinados para a sua amplitude. A condição para o funcionamento correto é que o ripple presente em um nível não se sobreponha ao ripple de nenhum dos níveis adjacentes, conforme mostrado na Figura 23. Assim, a Equação (3.17) mostra a atenuação mínima necessária para satisfazer essa condição.

$$\begin{aligned} -h_{1_{pp}} A &\leq \frac{k}{2^{N-1}} \\ -\frac{4k}{\pi} A &\leq \frac{k}{2^{N-1}} \\ A &\geq \frac{\pi}{2^{N+1}} \\ A_{dB} &\geq 20 \log(\frac{\pi}{2^{N+1}}) \end{aligned} \quad (3.17)$$

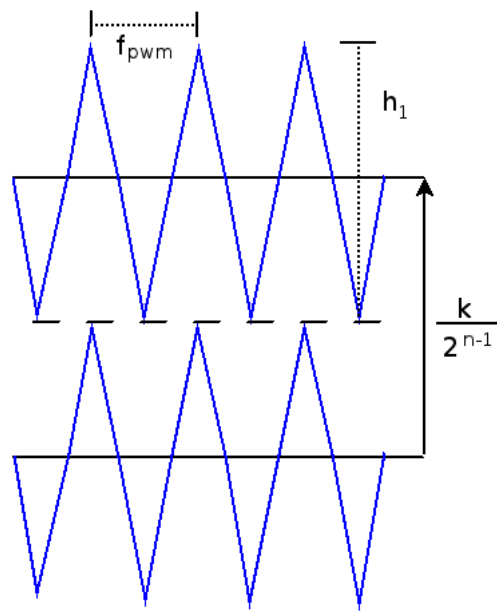


Figura 23: Caracterização do ripple do sinal filtrado.

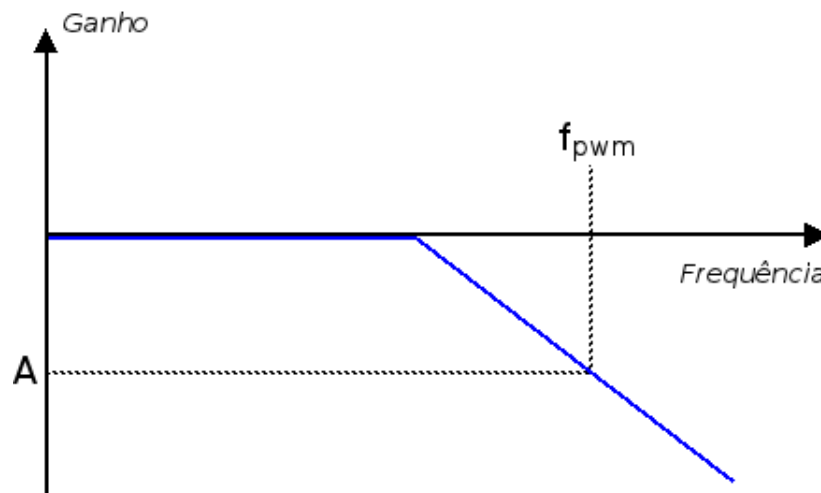


Figura 24: Determinação da frequência do sinal PWM em função da atenuação.

Como pode ser notado, a Equação (3.17) traz a atenuação mínima em decibéis. Assim, a maneira mais simples de se determinar a frequência de operação do sinal PWM é traçar a função de transferência do filtro em um diagrama de Bode e verificar a frequência em que a atenuação mínima é conseguida, como mostrado na Figura 24. A Equação (3.18) ilustra a função de transferência de um filtro de primeira ordem em função de sua constante de tempo.

$$T(s) = \frac{1}{\tau s + 1} \quad (3.18)$$

A etapa final da modelagem do sistema é determinar a frequência de operação do

modulador PWM. Sabendo-se a frequência de operação do sinal PWM e verificando que o modulador foi baseado em um contador síncrono, é fácil perceber que o mesmo deve operar com uma frequência $2^{N-1}f_{pwm}$, como mostrado na Equação (3.19).

$$f_{clk} = 2^{N-1}f_{pwm} \quad (3.19)$$

3.8 Validação do modelo

Para validar o equacionamento proposto, foi projetado um conversor A/D de quatro bits, alimentado com uma tensão de 1,5V e trabalhando com uma taxa de amostragem de 200 Hz.

3.8.1 Dados

$N = 5$ (4 bits de resolução e 1 bit para indicar o final da conversão)

$$V_{REF} = k = 1,5V$$

$$f_s = 200S/s$$

3.8.2 Cálculos

Utilizando-se a Equação (3.6) pode-se determinar a frequência de clock do registrador de aproximação sucessiva e do comparador inversor, como mostrado na Equação (3.20).

$$\begin{aligned} f_{SAR} &\geq f_s N \\ &\geq 200 \times 5 \\ &\geq 1,0kHz \end{aligned} \quad (3.20)$$

O tempo de resposta do filtro pode ser encontrado utilizando-se a Equação (3.9). Assim:

$$\begin{aligned} \tau &\leq \frac{1}{2\pi 5 f_{SAR}} \\ &\leq \frac{1}{2\pi 51000} \\ &\leq 3,19 \times 10^{-5}s \end{aligned} \quad (3.21)$$

A atenuação necessária pode ser obtida através da Equação (3.17) e seu valor é dado

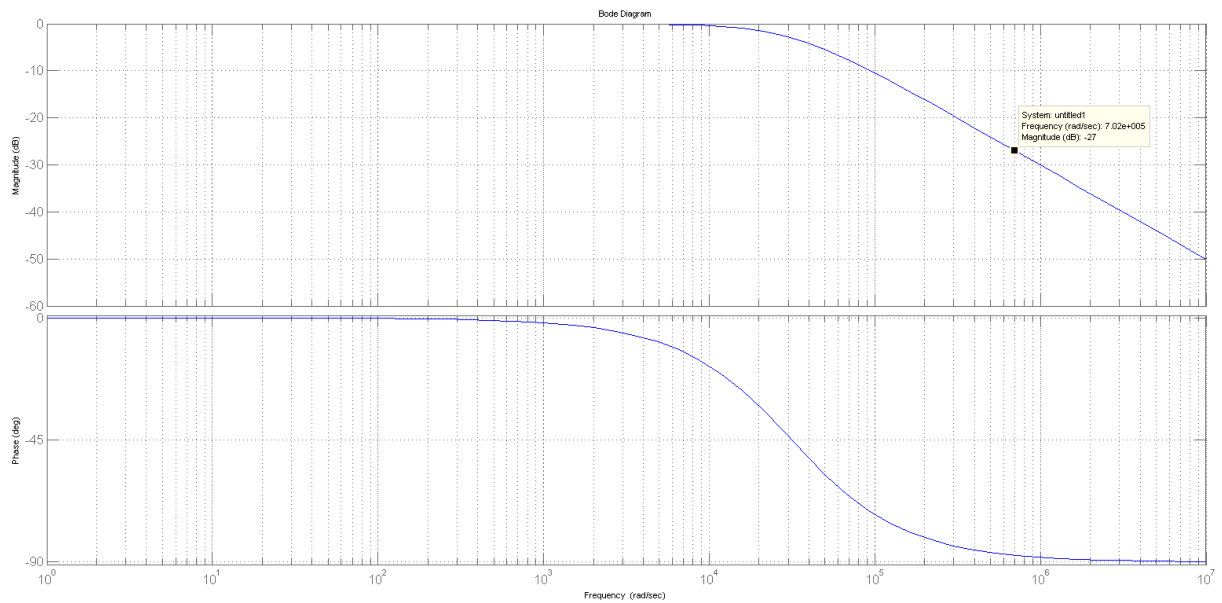


Figura 25: Diagrama de Bode do filtro projetado.

por:

$$\begin{aligned}
 A_{dB} &\geq 20 \log\left(\frac{\pi}{2^{N+1}}\right) \\
 &\geq 20 \log\left(\frac{\pi}{2^6}\right) \\
 &\geq 26,18 dB.
 \end{aligned} \tag{3.22}$$

Assim, é razoável escolher a atenuação maior que $27dB$.

Para finalizar o projeto do conversor em questão, se faz necessário o auxílio do diagrama de Bode do filtro já caracterizado, dado pela Figura 25

Nota-se que para obter-se uma atenuação de $27dB$ é necessário que o sinal PWM trabalhe em uma frequência de $f_{PWM} \geq 113,5kHz$. A frequência de clock do modulador PWM pode ser determinada através da Equação (3.19). Desta forma, tem-se:

$$\begin{aligned}
 f_{pwm}^{clk} &= 2^{N-1} f_{pwm} \\
 &= 2^4 \times 113,5 \\
 &= 1,82MHz
 \end{aligned} \tag{3.23}$$

Assim, é razoável escolher uma frequência de $2MHz$ para o clock do modulador PWM.

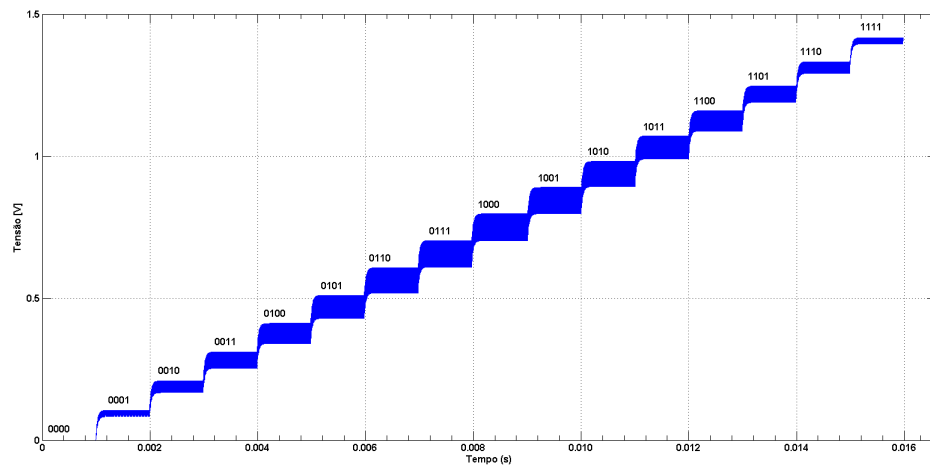


Figura 26: Resposta do conversor D/A para os dezesseis códigos possíveis.

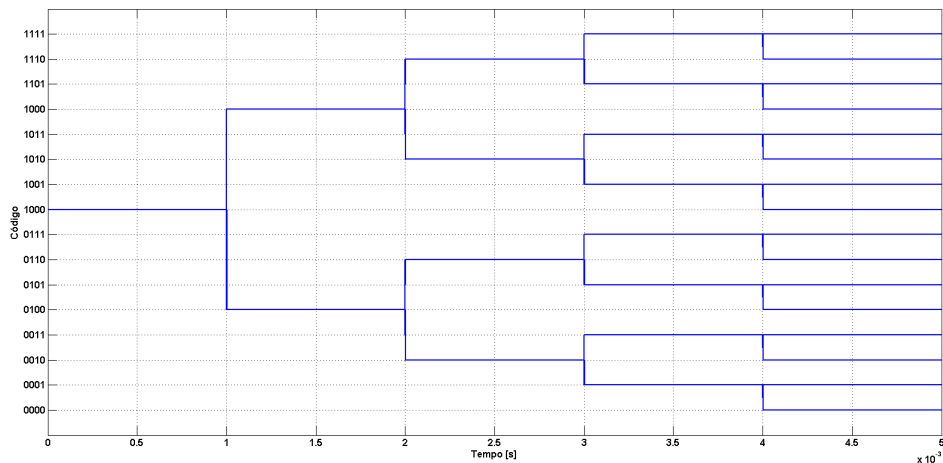


Figura 27: Funcionamento do conversor A/D com registrador de aproximação sucessiva

3.8.3 Simulações

As simulações mostradas nas figuras 26 e 27 foram realizadas com o auxílio da ferramenta Simulink e dos modelos digitais em VHDL ilustrados no Apêndice A. A Figura 26 mostra a resposta do conversor D/A baseado no modelo digital do modulador PWM em conjunto com o filtro especificado. Nota-se que o filtro responde conforme esperado e que os níveis de ripple dos níveis de tensão adjacentes não se sobrepõem.

A Figura 27 ilustra o funcionamento do conversor A/D. Nota-se que o registrador de aproximação sucessiva realiza a conversão em quatro ciclos de clock e no quinto ciclo os dados estão disponíveis para serem lidos. Nota-se ainda que não há códigos perdidos no conversor modelado.

3.8.4 Funcionamento da arquitetura

A seguir será ilustrado o funcionamento do conversor A/D baseado na arquitetura validada de 4 bits. Para tanto, será examinada passo-a-passo a conversão da tensão $0,4V$ em uma palavra digital de quatro bits, mostrada na Figura 28.

Após um pulso de reset, o registrador de aproximação sucessiva é carregado com o valor 1000 e o bit que indica o final da conversão (EOC) é carregado com o valor 0. Durante o primeiro ciclo do sinal de clock sar, o bit mais significativo é avaliado. A palavra presente na saída do registrador de aproximação sucessiva é atualizada na entrada do gerador PWM em todo instante em que seu contador atinge a condição de reset. Assim, o gerador PWM produz um sinal quadrado com ciclo de trabalho de 50%, que após a filtragem entrega uma tensão de $0,75V$ na entrada do comparador. Como esse valor é superior ao valor de $0,40V$ presente na outra entrada do comparador, o mesmo produz o valor 0 como resultado da avaliação do bit mais significativo.

Durante o segundo ciclo do sinal de clock o registrador de aproximação sucessiva avalia o segundo bit mais significativo já levando em conta o valor do bit mais significativo anteriormente determinado, nesse caso, possui o valor 0100 e o sinal EOC ainda permanece com valor 0. Essa palavra digital produz um sinal PWM com ciclo de trabalho de 25 % capaz de entregar uma tensão de $0,325V$ ao comparador. Como essa tensão é inferior à tensão presente em sua outra entrada, o comparador produz 1 como resposta ao bit em avaliação.

No terceiro ciclo de clock o registrador SAR possui o valor 0110 e o sinal EOC continua em 0. A palavra digital produz um sinal PWM de ciclo de trabalho 37,5 % capaz de gerar uma tensão de $0,563V$ na entrada do comparador após a etapa de filtragem. Assim, o comparador responde com o valor 0, indicando que a tensão do sinal PWM é superior à tensão presente na outra entrada.

No quarto ciclo de clock, o valor do ultimo bit é avaliado. O registrador SAR possui o valor 0101 e o sinal EOC continua em 0. A palavra digital na entrada do gerador PWM corresponde a um sinal PWM com ciclo de trabalho de 31,25 %, que resulta em uma tensão de $0,469V$ na entrada do comparador. Como esse valor é superior ao valor presente na outra entrada, o comparador conclui a avaliação do ultimo bit com o valor 0.

No ultimo ciclo de clock, o registrador SAR recebe a resposta do ultimo bit avaliado pelo comparador, possuindo o valor da conversão 0100 e o sinal EOC recebe o valor 1 indicando o final da conversão.

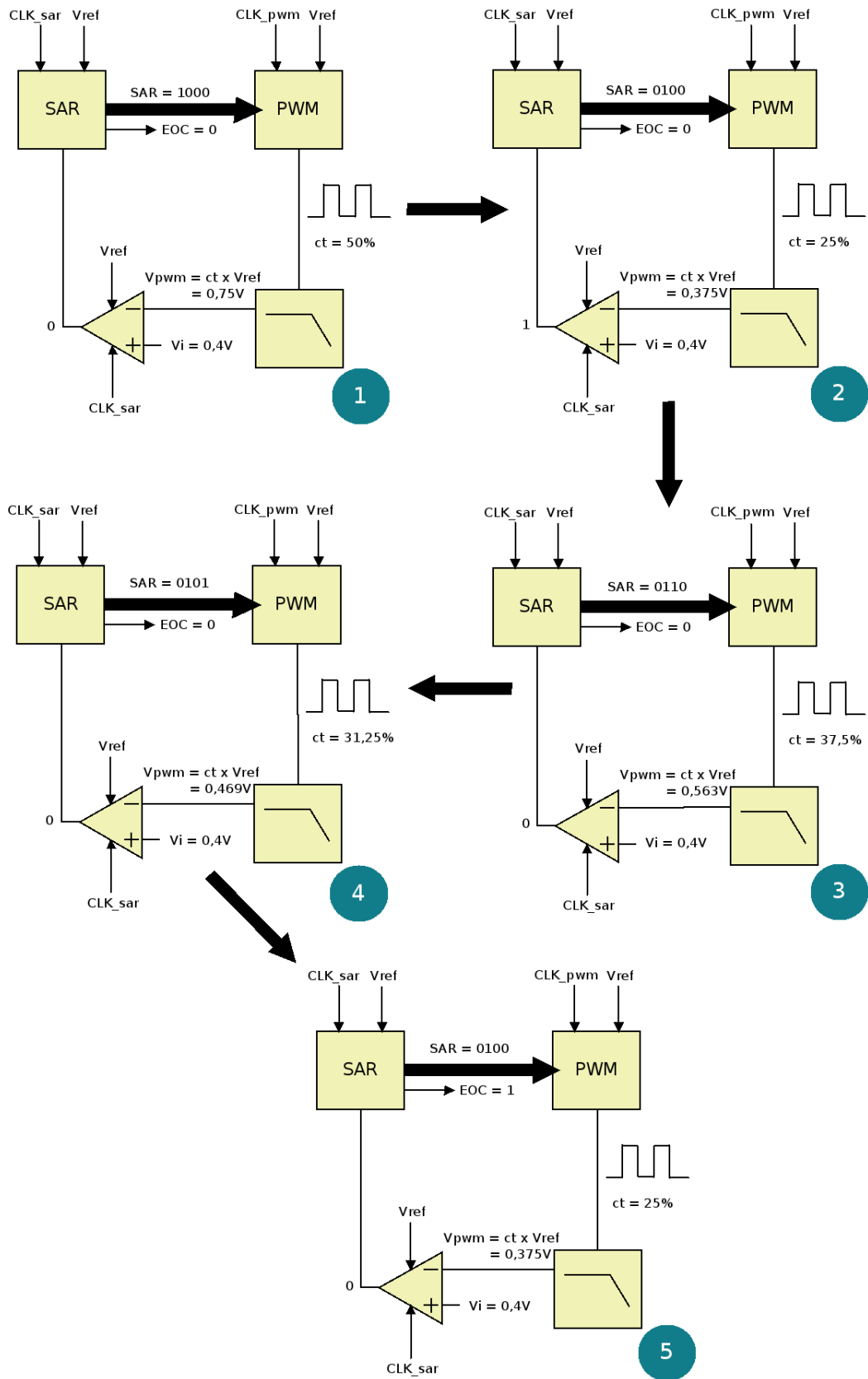


Figura 28: Etapas da conversão A/D.

4 *Layout e simulações*

Esse capítulo tem o objetivo de mostrar o layout da arquitetura proposta, bem como caracterizar as simulações efetuadas para validar o circuito integrado. Sua caracterização é muito importante, pois apenas quando o método de teste é bem definido, pode-se determinar se as características do circuito conversor, mostradas pelos resultados, são válidas ou não para determinada aplicação [3].

Os parâmetros críticos do conversor são determinados de acordo com o seu tipo de aplicação [12]. Dessa maneira, serão analisados os parâmetros estáticos INL, DNL e os parâmetros dinâmicos ENOB e SNR.

4.1 **Layout**

A seguir são ilustrados os layouts dos estágios mostrados no diagrama de blocos do circuito proposto, ilustrado na figura 7 do terceiro capítulo. Os layouts foram desenvolvidos utilizando a ferramenta CADENCE Virtuoso seguindo os conceitos abordados em [13], [14] e [15].

No layout do comparador, vale salientar a utilização de um arranjo de capacitores unitários de aproximadamente $100fF$, como mostrado na figura 29.

Foram ligados três capacitores unitários em paralelo para formar cada capacitor de $300fF$ dos dois circuitos dobradores de tensão utilizados nos sinais de clock. O restante dos capacitores foram ligados em paralelo para totalizar a capacitância de $1,4pF$ na entrada do circuito comparador.

Esse tipo de matriz de capacitores unitários tem a função de minimizar o descasamento de capacitâncias causado por variações no processo de fabricação.

Outra boa prática de layout que pode ser observada é a blindagem dos circuitos com uma camada de metal 3 que possui a finalidade de minimizar os efeitos dos ruídos.

Nos circuitos digitais ilustrados nas figuras 30 e 31, as trilhas que carregam os sinais de clock foram uniformemente distribuídas de maneira que os sinais cheguem ao mesmo tempo nas portas lógicas. As trilhas de alimentação têm maior espessura para suportar a condição de máxima corrente.

A figura 32 ilustra o layout completo do conversor A/D que ocupa uma área de $0,749\text{mm}^2$, incluindo o circuito de reset utilizado para realizar as medidas de INL, DNL e ENOB. Pode-se notar a presença de anéis de guarda circundando o circuito analógico que é mais sensível a ruídos provenientes do substrato.

Também pode ser observado que o substrato e o terra foram conectados internamente e são exteriorizados através de uma única saída. Dessa maneira o substrato experimenta as mesmas variações de tensão presentes na referência, minimizando o ruído no dispositivo.

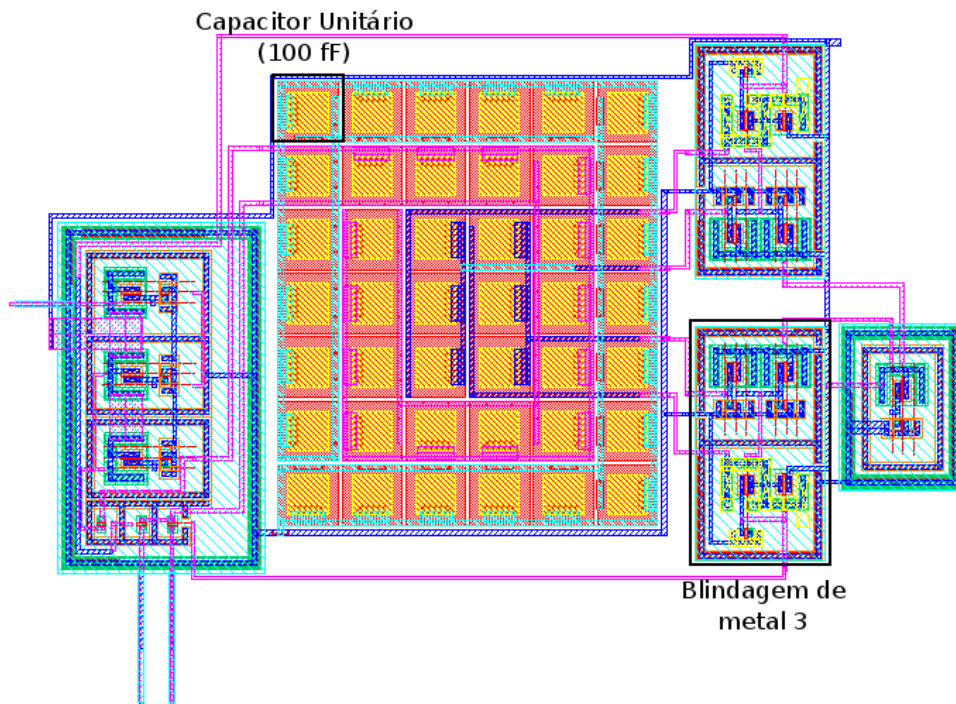


Figura 29: Layout do estágio comparador.

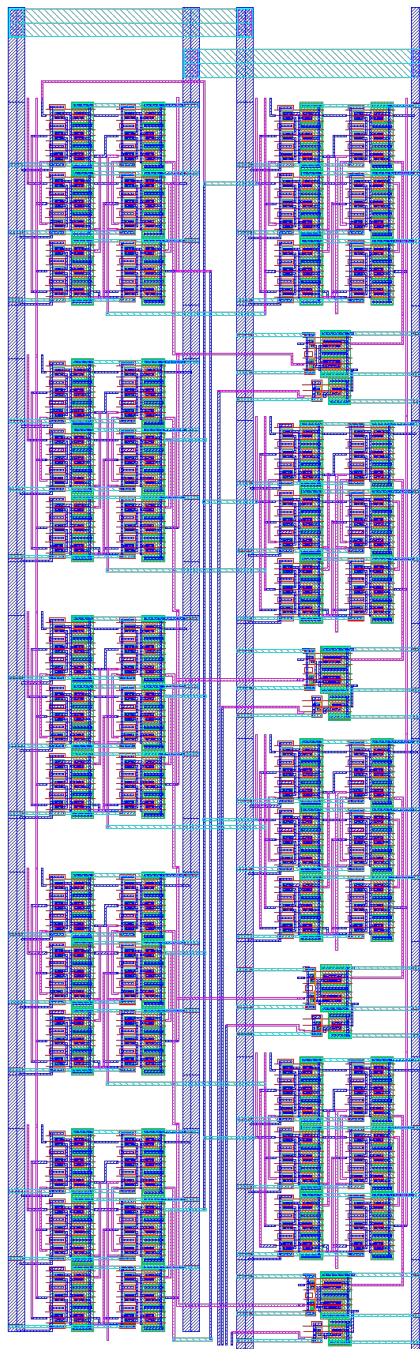


Figura 30: Layout do estágio de aproximação sucessiva.

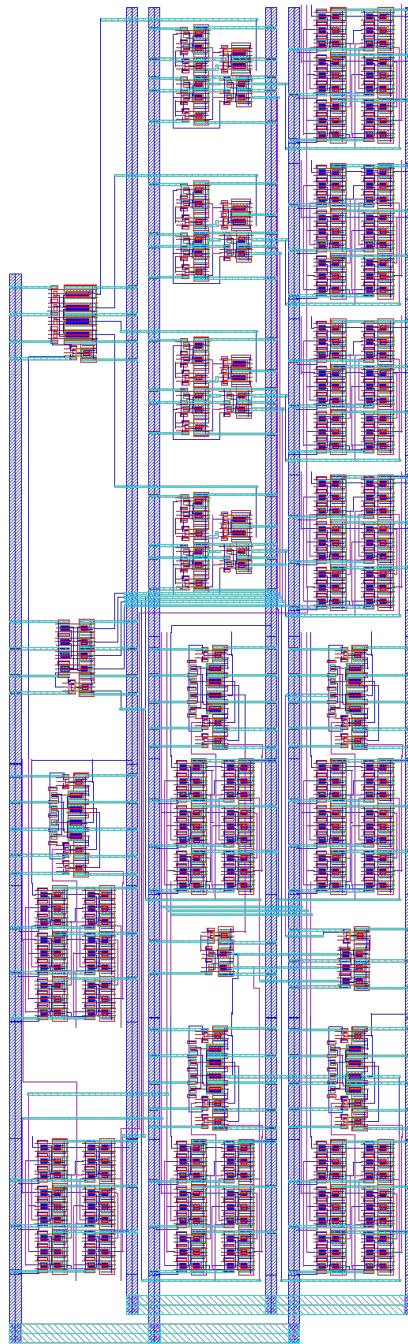


Figura 31: Layout do estágio modulador PWM.

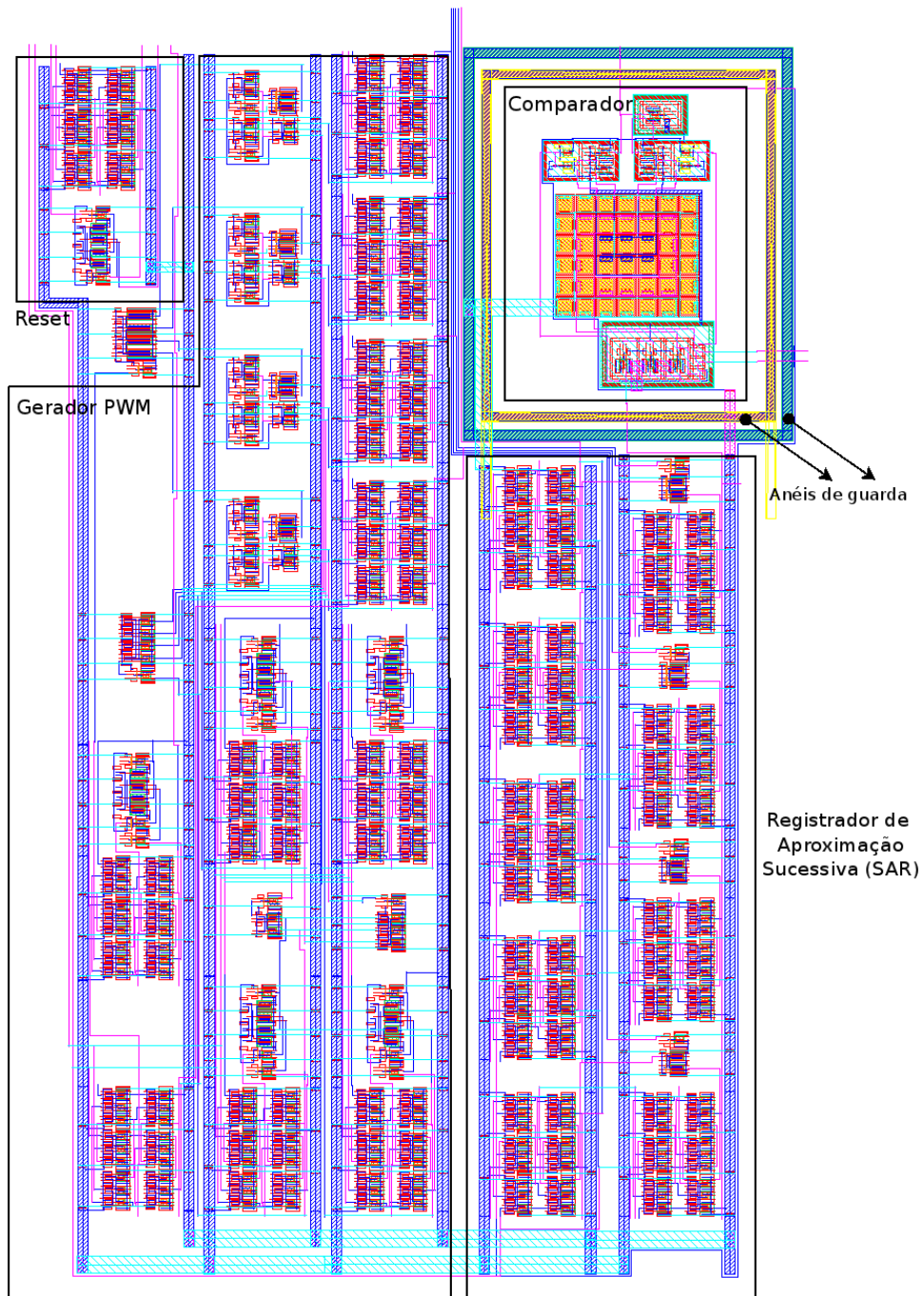


Figura 32: Layout completo do conversor.

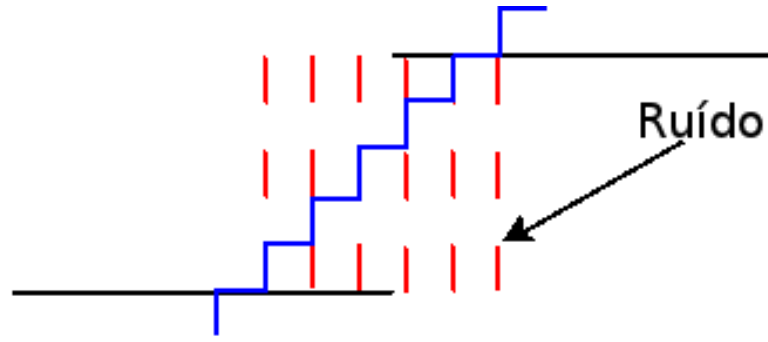


Figura 33: Entrada analógica amostrada através da região de transição.

4.2 Parâmetros estáticos

Os resultados das simulações de pós-layout dos parâmetros estáticos são realizados através da aplicação de uma série de níveis D.C. na entrada do conversor e do monitoramento da conversão desses níveis.

Para a determinação dos parâmetros estáticos do conversor, foi utilizada a metodologia de teste por rampa [3]. Nessa metodologia, uma rampa lenta que varia de $0V$ até V_{REF} é aplicada na entrada do conversor de maneira que todos os possíveis códigos possam ser obtidos em sua saída. O nível de transição é determinado quando o conversor retorna um novo código durante 50% do tempo. Para se determinar a localização dos níveis de transição, a metodologia utiliza a definição probabilística dos níveis de transição.

A Figura 33 mostra o que acontece quando uma rampa lenta é amostrada na região de transição. Devido ao ruído presente nessa região, a cada incremento da rampa ocorre um aumento na probabilidade de conversão de um novo código. A Figura 34 ilustra a função de probabilidade de ocorrência de um novo código.

Fica claro que a localização da transição entre os códigos é obtida quando a contagem do novo código atinge 50 % do tempo.

Para essa simulação, foram estipuladas 6 amostras para cada nível de transição. Sabendo-se que o conversor projetado possui $2^4 = 16$ níveis e que a arquitetura proposta necessita de $5ms$ para converter uma amostra, a Equação (4.1) mostra o tempo necessário para realizar a conversão de todas as amostras.

$$t_c = n \times A \times \left(\frac{1}{f_s}\right) = 16 \times 6 \times \left(\frac{1}{200}\right) = 480ms \quad (4.1)$$

onde

t_c representa o tempo de conversão;

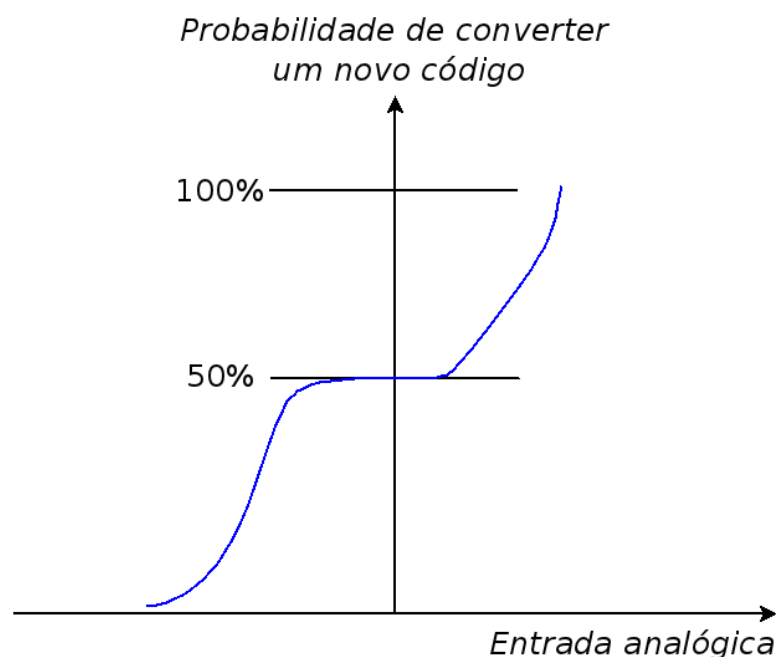


Figura 34: Curva de probabilidade de conversão de um novo código.

n representa o número de níveis de transição;

A representa o número de amostras por nível de transição;

f_s representa a taxa de amostragem do conversor proposto.

Entretanto, para que a conversão ocorra de maneira contínua, é necessário um circuito de *reset* que seja capaz de identificar o final de uma amostragem e informar ao conversor quando iniciar a próxima amostragem. Sabendo-se que esse circuito necessita de um ciclo de *clock* para realizar essa função e que entra em funcionamento à partir do final da primeira amostra, deve-se considerar na rampa de entrada o tempo inserido pelo circuito de *reset*, conforme mostrado na Equação (4.2).

$$t_r = \frac{1}{f_{clock}} \times (n - 1) = 0,001 \times 15 = 15ms \quad (4.2)$$

onde

t_r representa o tempo introduzido pelo circuito de *reset*;

T_{clock} representa o período do sinal de *clock*;

n representa o número de níveis de transição.

Assim, o período de subida da rampa de entrada é dado pela soma de t_c com t_r , como mostrado na Equação (4.3).

$$T_{rampa} = t_c + t_r = 480ms + 15ms = 495ms \quad (4.3)$$

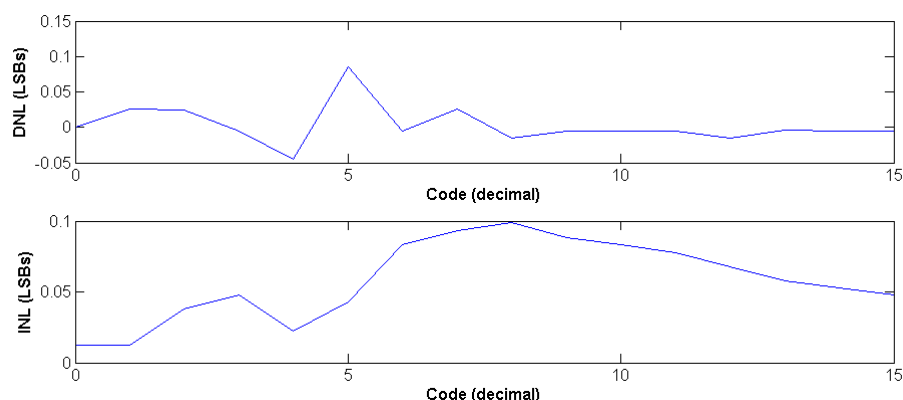


Figura 35: Visualização dos erros DNL e INL do conversor.

A Tabela 4 mostra os resultados obtidos para os erros de DNL e INL. O tratamento dos dados foi realizado com base nos conceitos definidos no segundo capítulo. A coluna DNL mostra que houve tanto degraus maiores quanto menores em relação ao tamanho ideal, sendo que as variações ocorreram entre $-0,045LSB$ e $+0,086LSB$. A coluna INL mostra que os pontos médios de todos os códigos encontram-se ligeiramente deslocados acima de seus valores ideais, sendo o deslocamento máximo de $0,098LSB$. A Tabela 4 nos permite verificar a precisão obtida pela arquitetura proposta, uma vez que os valores de DNL e INL são inferiores ao ruído de quantização teórico de $0,5LSB$. A Figura 35 ilustra de maneira gráfica os dados apresentados na tabela 4.

Tabela 4: Resultado da simulação com uma entrada em rampa lenta.

| Decimal | V_i ideal | V_i simulado | DNL | INL |
|---------|-------------|----------------|----------|---------|
| 0 | 0,00000 | 0,00000 | 0,00000 | 0,01263 |
| 1 | 0,15625 | 0,18150 | 0,02525 | 0,01263 |
| 2 | 0,31250 | 0,36290 | 0,02515 | 0,03783 |
| 3 | 0,46875 | 0,51410 | -0,00505 | 0,04788 |
| 4 | 0,62500 | 0,62500 | -0,04535 | 0,02268 |
| 5 | 0,78125 | 0,86690 | 0,08565 | 0,04283 |
| 6 | 0,93750 | 1,01800 | -0,00515 | 0,08308 |
| 7 | 1,09375 | 1,20000 | 0,02575 | 0,09338 |
| 8 | 1,25000 | 1,34100 | -0,01525 | 0,09863 |
| 9 | 1,40625 | 1,49200 | -0,00525 | 0,08838 |
| 10 | 1,56250 | 1,64300 | -0,00525 | 0,08312 |
| 11 | 1,71875 | 1,79400 | -0,00525 | 0,07788 |
| 12 | 1,87500 | 1,93500 | -0,01525 | 0,06763 |
| 13 | 2,03125 | 2,08700 | -0,00425 | 0,05788 |
| 14 | 2,18750 | 2,23800 | -0,00525 | 0,05313 |
| 15 | 2,34375 | 2,38900 | -0,00525 | 0,04788 |

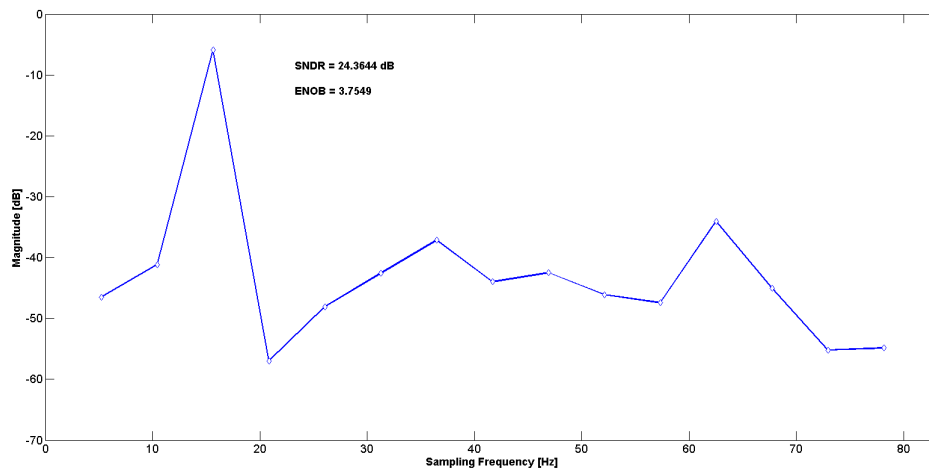


Figura 36: Resultado da FFT para 32 amostras.

4.3 Parâmetros dinâmicos

Os testes dos parâmetros dinâmicos são realizados aplicando-se sinais de entrada periódicos que tenham uma forma de onda parecida com as formas de onda envolvidas na aplicação em que o circuito conversor será utilizado. Para as simulações dos parâmetros dinâmicos, foi utilizada a metodologia da Transformada Rápida de Fourier (FFT) [3]. Esse tipo de teste avalia diretamente a capacidade do circuito conversor em converter um sinal analógico conhecido.

Dessa maneira, considerando-se os dados apresentados na Sessão 2.1.9 do Capítulo 2, pode-se projetar os parâmetros utilizados na simulação. Levando em conta a taxa de amostragem de 166,67 Hz, devido ao circuito de reset, pode-se aplicar um sinal senoidal de 15,15 Hz de frequência e de 2,40V na entrada do circuito, de maneira a se conseguir 32 amostras. A Figura 36 ilustra o resultado da transformada rápida de Fourier para os 32 pontos.

A relação sinal ruído (SNDR) alcançou 24,36dB, ficando muito próxima do limite teórico de 25,84 dB para uma resolução de 4bits, resultando em 3,75 bits efetivos. Esse resultado comprova o alto desempenho obtido pela arquitetura proposta.

4.4 Análise comparativa

Com a finalidade de compreender melhor os resultados obtidos, a Tabela 5 mostra uma comparação entre o trabalho proposto e duas outras arquiteturas de conversão A/D. Em

[16], é proposto um conversor de 9 bits que utiliza uma arquitetura SAR para a conversão dos 6 bits menos significativos em conjunto com uma arquitetura Flash para converter os 3 bits mais significativos. Em [17], é descrita uma arquitetura que utiliza conversores SAR intercalados, com rede capacitiva no estágio de conversão digital-analógico.

Tabela 5: Quadro comparativo.

| | [16] | [17] | Esse trabalho |
|---------------------------|--------------|------------------|----------------|
| Tensão de Alimentação (V) | 1,3 | 1,8 | 2,5 |
| Tecnologia (nm) | 90 | 18 | 500 |
| Arquitetura | Flash+SAR | SAR intercalados | SAR |
| DNL (LSB) | -0,48 / 0,35 | -0,70 / 0,20 | -0,045 / 0,086 |
| INL (LSB) | -0,48 / 0,44 | -1,00 / 0,20 | 0,00 / 0,099 |
| Resolução (bit) | 9 | 4 | 4 |
| SNDR (dB) | - | 23,9 | 24,36 |
| Potência (mW) | 2,2 | 23,3 | 0,016 |

Analisando-se o dados conclui-se que a arquitetura proposta é mais precisa devido aos seus valores inferiores de DNL e INL, respectivamente 0,086 e 0,099. Outra característica é o menor consumo de potência, apenas $16\mu W$, mesmo quando comparado com arquiteturas alimentadas com uma tensão inferior.

Também ficou comprovado a que a eliminação dos erros de descasamento do estágio DAC resulta em um aumento do número efetivo de bits, traduzido pelo aumento do valor da relação sinal ruído (SNDR) quando comparado com [17], que utiliza uma rede capacitiva sujeita aos erros de descasamento em seu estágio DAC.

Assim, a Tabela 6 resume as características da arquitetura proposta. Os baixos valores de DNL e INL (inferiores a 0,1LSB) e o baixo consumo de potência ($16\mu W$) obtidos pela arquitetura proposta vão de encontro as necessidades das aplicações de baixa potência que necessitam de precisão, como as biomédicas.

Tabela 6: Característica da arquitetura simulada.

| | |
|---------------------------------|-----------|
| Tecnologia | 0,5 um |
| Tensão de Alimentação | 2,5V |
| Maxima Frequência de Amostragem | 200 Hz |
| ENOB(@166,67 Hz) | 3,7549-b |
| DNL(max) | 0,086 LSB |
| INL(max) | 0,099 LSB |
| Consumo de Potência | 16 uW |

5 *Conclusão e trabalhos futuros*

Esse capítulo finaliza o trabalho detalhando as conclusões obtidas a partir dos resultados das simulações realizadas e sugere possíveis temas para estudos futuros.

5.1 Conclusão

Como apresentado no Capítulo 3, foi modelada e projetada uma arquitetura de conversão A/D de 4 – *bits* que minimiza os erros de DNL e INL através da substituição do estágio DAC convencional, sujeito à descasamentos introduzidos pelo processo de fabricação, por um modulador PWM digital e um filtro passa baixas de primeira ordem.

A presente arquitetura utiliza um circuito comparador inversor quase totalmente digital de complexidade reduzida. Essas modificações reduzem a complexidade do circuito total e sendo aproximações digitais, contribuem para minimizar seu consumo de potência.

Para realizar a validação da mesma, seu layout foi desenvolvido utilizando-se o processo CMOS de $0.5\mu m$, ocupando uma área de silício de $0,749mm^2$.

Os resultados das simulações mostram que a arquitetura desenvolvida apresenta grande precisão, pois os erros de DNL e INL são da ordem $0,086LSB$ e $0,099LSB$, respectivamente, muito abaixo dos valores obtidos por outras arquiteturas.

As características dinâmicas também se mostraram bastante satisfatórias, obtendo-se valores de $24,36dB$ de SNDR e 3,75 bits efetivos. O consumo de potência da ordem de $16\mu W$ também se mostrou bastante inferior aos valores obtidos por outras arquiteturas, mesmo quando alimentadas com tensões inferiores. Essa característica é adequada a aplicações de baixa potência.

A validação da arquitetura proposta foi feita utilizando-se uma resolução de 4 bits devido às razões discutidas na sessão 3.1 do Capítulo 3. Entretanto pode-se construir conversores de maior resolução realizando-se alterações nos circuitos digitais do Registra-

dor de Aproximação Sucessiva e do Modulador PWM digital, adicionando o hardware necessário em cada estágio. Outra alternativa seria construir conversores do tipo pipeline utilizando-se a arquitetura de 4 bits já desenvolvida.

Enfim, as simulações apresentadas no capítulo anterior comprovam a melhoria da precisão do conversor A/D, como pretendido e também evidenciam o seu baixo consumo de potência.

5.2 Trabalhos futuros

A seguir são apresentadas algumas ideias que complementam a arquitetura proposta e sugerem aplicações para a mesma.

- Desenvolvimento de uma topologia de filtro que minimize a influencia do processo de fabricação.
- Desenvolvimento de um circuito de *sample-and-hold* específico para essa arquitetura de conversão A/D.
- Integrar o circuito conversor em um sistema completo de aquisição de dados biomédicos.
- Utilização da arquitetura proposta em conversores mais eficiente, como os assíncrono ou os AIC (Conversores Análogo-Informação).

Referências

- [1] ZOU, X. et al. A 1-v 450-nw fully integrated programmable biomedical sensor interface chip. *Solid-State Circuits, IEEE Journal of*, v. 44, n. 4, p. 1067 –1077, april 2009. ISSN 0018-9200.
- [2] NG, K.; CHAN, P. A cmos analog front-end ic for portable eeg/ecg monitoring applications. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, v. 52, n. 11, p. 2335 – 2347, nov. 2005. ISSN 1549-8328.
- [3] HOESCHELE, D. F. J. *Analog-to-Digital and Digital-to-Analog Conversion Techniques*. 2nd. ed. [S.l.]: John Wiley e Sons, 1994.
- [4] ALLEN, P. E.; R., H. D. *CMOS Analog Circuits Design*. [S.l.]: Oxford University Press, 2002.
- [5] HILDEBRAND, F. *Advanced Calculus for Applications*. 2nd edition. ed. [S.l.]: Prentice-Hall Inc., 1976.
- [6] BOGART, T. F. J. *Introduction to digital Circuits*. [S.l.]: Lake Forest: Mcgraw hill International Editions, 1992.
- [7] OPPENHEIM A. V.; WILLISKY, A. S.; YOUNG, I. T. *Signals and systems*. [S.l.]: Prentice-Hall, 1983.
- [8] LIN, Y.-Z. et al. A 9-bit 150-ms/s 1.53-mw subranged sar adc in 90-nm cmos. In: *VLSI Circuits (VLSIC), 2010 IEEE Symposium on*. [S.l.: s.n.], 2010. p. 243 –244.
- [9] ALTER, D. M. *Using PWM Output as a Digital-to-Analog Converter on a TMS320F280x Digital Signal Controller*. [S.l.], 2088.
- [10] MIKKOLA, E. et al. Set tolerant cmos comparator. *Nuclear Science, IEEE Transactions on*, v. 51, n. 6, p. 3609 – 3614, dec. 2004. ISSN 0018-9499.
- [11] AY, S. U. A sub-1 volt 10-bit supply boosted sar adc design in standard cmos. *Analog Integr. Circuits Signal Process.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 66, p. 213–221, February 2011. ISSN 0925-1030. Disponível em: <<http://dx.doi.org/10.1007/s10470-010-9515-3>>.
- [12] IEEE Standard for Terminology and Test Methods for Analog-To-Digital Converters. *IEEE Std 1241-2000*, p. i, 2001.
- [13] WESTE, N. *Principles of CMOS VLSI design*. [S.l.]: Addison-Wesley, 1985.
- [14] HASTINGS, A. *The art of analog layout*. [S.l.]: Prentice-Hall, Inc., 2001.
- [15] RAZAVI, B. *Design of analog CMOS integrated circuits*. [S.l.]: McGraw Hill, 2001.

-
- [16] LIN, Y.-Z. et al. A 9-bit 150-ms/s 1.53-mw subranged sar adc in 90-nm cmos. In: *VLSI Circuits (VLSIC), 2010 IEEE Symposium on*. [S.l.: s.n.], 2010. p. 243 –244.
- [17] TALEKAR, S. et al. A low power 700msps 4bit time interleaved sar adc in 0.18um cmos. In: *TENCON 2009 - 2009 IEEE Region 10 Conference*. [S.l.: s.n.], 2009. p. 1 –5.

APÊNDICE A – Modelos Digitais

Esse apêndice apresenta a modelagem em linguagem de descrição de hardware (VHDL) dos elementos digitais tratados no capítulo 3. São apresentados os modelos dos estágios Registrador de Aproximação Sucessiva (SAR) e Modulador de largura de pulso (PWM).

A.1 Registrador de Aproximação Sucessiva-SAR

```

1  library ieee ;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_arith.all;
4  entity sar is
5      port (clk_sar: in std_logic;
6            rst_sar: in std_logic;
7            sel_sar: in std_logic;
8            out_sar: out std_logic_vector (3 downto 0);
9            end_sar: out std_logic);
10 end sar;
11 architecture rtl of sar is
12 begin
13     reg: process (clk_sar, rst_sar)
14         variable aux: std_logic_vector(4 downto 0);
15         variable cnt: std_logic_vector(4 downto 0);
16         begin
17             if rst_sar = '1' then
18                 cnt:= "10000";
19                 aux:= "10000";
20                 end_sar <= '0';
21                 out_sar <= "1000";
22             elsif clk_sar'event and clk_sar='1' then
23                 if sel_sar = '0' then
24                     aux:= aux and not cnt;
25                 end if;
26                 cnt:= '0' & cnt(4 downto 1);
27                 aux:= aux or cnt;
28             end if;
29             out_sar <= aux(4 downto 1);
30             end_sar <= not(cnt(4) or cnt(3) or cnt(2) or cnt(1) or cnt(0));
31         end process;
32     end rtl;

```

A.2 Modulador PWM

Para um melhor entendimento o código do modulador PWM foi dividido em quatro partes. As estruturas *Registrador*, *Contador* e *Comparador* foram modeladas separadamente. O modulador PWM foi então descrito utilizando essas estruturas como componentes. Os componentes foram então interligados utilizando-se sinais internos auxiliares e a instrução *port map*.

A.2.1 Registrador

```
1 library ieee;
2 use ieee.std_logic_1164.all;
3
4 entity reg is
5 port(load_reg: in std_logic;
6       in_reg: in std_logic_vector( 3 downto 0);
7       out_reg: out std_logic_vector (3 downto 0);
8       rst_reg: in std_logic);
9 end reg;
10
11 architecture rtl of reg is
12 signal data: std_logic_vector (3 downto 0);
13 begin
14     reg: process(rst_reg , load_reg)
15     begin
16         if rst_reg = '1' then
17             data <= in_reg;
18             out_reg <= data;
19         elsif (load_reg='1') then
20             data <= in_reg;--carregar dados.
21         end if;
22         out_reg <= data;
23     end process;
24 end rtl;
```

A.2.2 Contador

```
1 library ieee ;
2 use ieee.std_logic_1164.all;
3 use ieee.std_logic_arith.all;
4
5 entity cnt is
6     port (clk_cnt: in STD_LOGIC;
7           rst_cnt: in STD_LOGIC;
8           out_cnt: out STD_LOGIC_VECTOR (3 downto 0);
9           end_cnt: out std_logic);
10 end cnt;
11
12 architecture estructura of cnt is
13 begin
14     contador: process (clk_cnt, rst_cnt) — contador de 4 bits ,
15     variable valor, qnext: STD_LOGIC_VECTOR(3 downto 0);
16     begin
17         valor:= "0001";
18         if (rst_cnt = '1') then —Reset assíncrono
19             qnext:= "0000";
20             end_cnt <= '0';
21             out_cnt <= "0000";
22         elsif (clk_cnt'event and clk_cnt = '1') then
23             qnext:= unsigned(qnext) + unsigned(valor); —Incrementa de 1
24             if (qnext="0000") then
25                 end_cnt <= '1';
26             end if;
27         end if;
28         out_cnt <= qnext;
29     end process;
30 end;
```

A.2.3 Comparador

```
1 library ieee;
2 use ieee.std_logic_1164.all;
3 use ieee.std_logic_arith.all;
4
5 entity cmp is
6 port(out_cmp: out std_logic:= '1';--Inicia a saída em '1'
7       in1_cmp: in std_logic_vector (3 downto 0);
8       in2_cmp: in std_logic_vector (3 downto 0);
9       clk_cmp: in std_logic;
10      rst_cmp: in std_logic);
11 end cmp;
12
13 architecture simple of cmp is
14 begin
15 compare: process (clk_cmp, rst_cmp)
16 variable in1_hold: std_logic_vector (3 downto 0);
17 variable in2_hold: std_logic_vector (3 downto 0);
18 begin
19     in1_hold:= in1_cmp; --Passar o valor da entrada 1
20                    --para o sinal 1 correspondente
21     in2_hold:= in2_cmp; --Passar o valro da entrada 2
22                    --para o sinal 2 correspondente
23     if in1_cmp = "0000" then
24         out_cmp <= '0';
25     elsif rst_cmp = '1' then
26         in1_hold:= in1_cmp;
27         in2_hold:= in2_cmp;
28     elsif clk_cmp'event and clk_cmp = '1' then
29         if unsigned(in1_hold) > unsigned(in2_hold) then
30             out_cmp <= '1'; --Mantém a saída em '1'
31             --enquanto a entrada 1
32             out_cmp <= '0'; --for maior que a 2
33         end if;
34     end if;
35 end process;
36 end simple;
```

A.2.4 Modulador PWM completo

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3  entity DPWM is
4  port (in_pwm: in std_logic_vector (3 downto 0);
5        rst_pwm: in std_logic;
6        clk_pwm: in std_logic;
7        load_pwm: in std_logic;
8        out_pwm: out std_logic);
9  end DPWM;
10 architecture blk of DPWM is
11  —DECLARAÇÃO DOS SINAIS INTERNOS
12  signal s_out_cnt: std_logic_vector (3 downto 0);
13  signal s_out_reg: std_logic_vector (3 downto 0);
14  signal s_end_cnt: std_logic;
15  signal s_rst_cnt: std_logic;
16  —DECLARAÇÃO DOS COMPONENTES UTILIZADOS
17  —Comparador de 4 bits
18  component cmp port (out_cmp: out std_logic;
19                    in1_cmp: in std_logic_vector (3 downto 0);
20                    in2_cmp: in std_logic_vector (3 downto 0);
21                    clk_cmp: in std_logic;
22                    rst_cmp: in std_logic);
23  end component;
24  —Contador de 4 bits
25  component cnt port (clk_cnt: in std_logic;
26                    rst_cnt: in std_logic;
27                    out_cnt: out std_logic_vector (3 downto 0);
28                    end_cnt: out std_logic);
29  end component;
30  —Registrador de 4 bits com sinal de load
31  component reg port (load_reg: in std_logic;
32                    in_reg: in std_logic_vector ( 3 downto 0);
33                    out_reg: out std_logic_vector (3 downto 0);
34                    rst_reg: in std_logic);
35  end component;
36  begin
37  —ATRIBUIÇÃO DOS SINAIS INTERNOS
38  s_rst_cnt <= rst_pwm or s_end_cnt;
39  —MAPEAMENTO DOS COMPONENTES / SINAIS
40  CMP1: cmp port map (out_pwm, s_out_reg, s_out_cnt, clk_pwm, s_rst_cnt);
41  CNT1: cnt port map (clk_pwm, s_rst_cnt, s_out_cnt, s_end_cnt);
42  REG1: reg port map (s_end_cnt, in_pwm, s_out_reg, rst_pwm);
43  end blk;

```

APÊNDICE B – Publicações

O presente trabalho teve como resultado as publicações listadas abaixo.

B.1 Artigos

A Low Power ADC Converter Based on PWM Technique, apresentado e publicado no Latin Display 2011, ocorrido nos dias 26 a 31 de agosto de 2011.

A Low Power Successive Approximation A/D Converter based on PWM Technique, a ser apresentado e publicado no LASCAS 2012 durante os dias 29 de fevereiro e 2 de março de 2012.

A SAR A/D Converter using PWM Technique, aceito para apresentação em forma de painel no 28th ICM, a ser realizado nos dias 13 a 16 de maio de 2012.

B.2 Capítulo de Livro

InTech - Biomedical Engineering.