



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Campus de Ilha Solteira

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**“Desenvolvimento de um Sintetizador de Frequência de Baixo
Custo em Tecnologia CMOS”**

VLADEMIR DE JESUS SILVA OLIVEIRA

Orientador: Prof. Dr. Nobuo Oki

Tese apresentada à Faculdade de
Engenharia - UNESP – Campus de Ilha
Solteira, para obtenção do título de
Doutor em Engenharia Elétrica.

Área de Conhecimento: Automação.

Ilha Solteira – SP
novembro/2009

FICHA CATALOGRÁFICA

Elaborada pela Seção Técnica de Aquisição e Tratamento da Informação
Serviço Técnico de Biblioteca e Documentação da UNESP - Ilha Solteira.

O48d	<p>Oliveira, Vlademir de Jesus Silva. Desenvolvimento de um sintetizador de frequência de baixo custo em tecnologia CMOS / Vlademir de Jesus Silva Oliveira. -- Ilha Solteira : [s.n.], 2009. 163 f. : il.</p> <p>Tese (doutorado) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira. Área de conhecimento: Automação, 2009</p> <p>Orientador: Nobuo Oki Bibliografia: p. 144-150</p> <p>1.Sintetizador de frequência. 2.CMOS. 3.Circuitos integrados. 4. PLL.</p>
------	---

CERTIFICADO DE APROVAÇÃO

TÍTULO: Desenvolvimento de um sintetizador de frequência de baixo custo em tecnologia CMOS

AUTOR: VLADEMIR DE JESUS SILVA OLIVEIRA
ORIENTADOR: Prof. Dr. NOBUO OKI

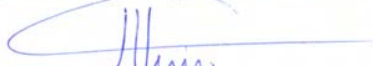
Aprovado como parte das exigências para obtenção do Título de DOUTOR em ENGENHARIA ELÉTRICA, Área: AUTOMAÇÃO, pela Comissão Examinadora:



Prof. Dr. NOBUO OKI
Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira



Profa. Dra. SUELY CUNHA AMARO MANTOVANI
Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira



Prof. Dr. JOZUE VIEIRA FILHO
Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira



Prof. Dr. MARCELO ARTURO JARA PEREZ
Brazil Semiconductor Technology Center / Departamento de Engenharia de Desenvolvimento e Integração de SoC



Prof. Dr. PAULO AUGUSTO DAL FABBRO
Centro de Pesquisa Renato Archer / Centro de Tecnologia da Informação

Data da realização: 25 de novembro de 2009.

À minha filha Veronika.

AGRADECIMENTOS

À todos os meus amigos e colegas, que me apoiaram moralmente e materialmente na realização desse trabalho.

A minha esposa, por respeitar as minhas necessidades.

À todos os professores que ajudaram na minha formação e em especial ao meu orientador prof. Dr. Nobuo Oki.

Agradeço ao CNPq e à FAPESP pela ajuda financeira concedida.

RESUMO

Nesta tese, propõe-se um sintetizador de frequência baseado em *phase locked loops* (PLL) usando uma arquitetura que utiliza um *dual-path loop filter*, constituído de componentes passivos e um integrador digital. A proposta é empregar técnicas digitais, para reduzir o custo da implementação do sintetizador de frequência, e flexibilizar o projeto do *loop filter*, para possibilitar que a arquitetura opere em uma faixa de frequência larga de operação e com redução de tons espúrios. O *loop filter* digital é constituído de um contador crescente/ decrescente cujo clock é proveniente da amostragem da diferença de fase de entrada. As técnicas digitais aplicadas ao *loop filter* se baseiam em alterações da operação do contador, em tempos pré-estabelecidos, os quais são controlados digitalmente. Essas técnicas possibilitam reduzir o tempo de estabelecimento do PLL ao mesmo tempo em que problemas de estabilidade são resolvidos. No desenvolvimento da técnica de *dual-path* foi realizado o estudo de sua estabilidade, primeiramente, considerando a aproximação do PLL para um sistema linear e depois usando controle digital. Nesse estudo foram deduzidas as equações do sistema, no domínio contínuo e discreto, tanto para o projeto da estabilidade, quanto para descrever o comportamento do PLL. A metodologia *top-down* é usada no projeto do circuito integrado. As simulações em nível de sistema são usadas, primeiramente, para as criações das técnicas e posteriormente para a verificação do seu comportamento, usando modelos calibrados com os blocos projetados em nível de transistor. O circuito integrado é proposto para ser aplicado em identificação por rádio frequência (RFID) na banda de UHF (*Ultra High Frequency*), usando *multi-standard*, e deve operar na faixa de 850 MHz a 1010 MHz. O sintetizador de frequência foi projetado na tecnologia CMOS 0.35 μm da AMS alimentado em 2 V. Em simulações, o circuito obteve 300 μs de tempo de estabelecimento e 140 Hz de resolução. O sintetizador de frequência proposto possui baixa complexidade e apresentou uma supressão de ruído da referência 45,6 dB melhor que a arquitetura convencional.

Palavras Chaves – Sintetizador de frequência, CMOS, circuitos analógicos e mistos, circuito integrado, PLL, técnicas digitais, RF.

ABSTRACT

In this thesis, a frequency synthesizers phase locked loops (PLL) based with an architecture that uses a dual-path loop filter consisting of passive components and a digital integrator are proposed. The objective is to employ digital techniques to reduce the implementation cost and get loop filter design flexibility to enable the architecture to have a large tuning range operation and spurious reduction. The digital loop filter is based in an up/down counter where the phase difference is sampled to generate the clock of the counter. The techniques applied in the digital path are based in digitally controlled changes in the counter operation in predefined time points. These techniques provide PLL settling time reductions whiling the stability issues are solved. The stability study of the proposed dual path has been developed. First the linear system approximation for the PLL has been assumed and then employing digital control. The continuous and discrete time equations of architecture were derived in that study applied to stability design as well as to describe the architecture behavior. The top-down methodology has been applied to the integrated circuit design. In the beginning, the system level simulations are used for the techniques creation and then the behavioral models that were calibrated with transistor level blocks are simulated. The application of the circuit is proposed to Radio Frequency Identification (RFID) using UHF (Ultra High Frequency) band for multi-standards application and will operate in range of 850 MHz to 1010 MHz. The proposed frequency synthesizer has been designed in the AMS 0.35 μm CMOS technology with 2V power supply. A 300 μs of settling time and 140 Hz of resolution was obtained in simulations. The proposed frequency synthesizer have low complexity and shown a reference noise suppression about 45.6 dB better than the conventional architecture.

Index Terms – Frequency synthesizer, CMOS, analog and mixed-signal circuits, integrated circuits, PLL, digital techniques, RF.

LISTA DE ILUSTRAÇÕES

Fig. 2.1. Sistema genérico de um transceptor usando <i>Duplexer Filter</i> .	21
Fig. 2.2. Efeito do ruído de fase na translação do sinal (<i>down-conversion</i>).	23
Fig. 2.3. Efeito dos tons espúrios na translação do sinal (<i>down-conversion</i>).	24
Fig. 3.1. Espectro do ruído de fase em um oscilador	27
Fig. 3.2. Início da oscilação em um oscilador tanque LC [43].	29
Fig. 3.3. Início da oscilação em um oscilador em anel diferencial [43].	29
Fig. 3.4. Diagrama de fase de um oscilador [45].	30
Fig. 3.5. Sistema em malha fechada de um oscilador linear [45].	31
Fig. 3.6. Formato do espectro do ruído ($1/f^2$)	32
Fig. 3.7. Ruído de fase previsto por Leeson \times linear	34
Fig. 3.8. Um oscilador LC excitado por um impulso de corrente.	35
Fig. 3.9. Resposta à entrada impulso [47].	36
Fig. 3.10. Exemplos de ISF para (a) oscilador LC e (b) oscilador em anel [47].	37
Fig. 3.11. Diagrama de blocos para a modelagem do sistema [47].	38
Fig. 3.12. Conversão da função de fase para o espectro do ruído de fase.	39
Fig. 3.13. Formato do espectro do ruído de fase proveniente das harmônicas de ω_0 [47].	40
Fig. 3.14. Representação de um oscilador perturbado saindo de sua órbita.	43
Fig. 4.1. Phase-locked loop (PLL) usando <i>charge-pump</i> , modelado como um sistema linear representado pelo ganho de cada bloco.	47
Fig. 4.2. <i>Loop filter</i> usando componentes passivos.	48
Fig. 4.3. Análise do comportamento do sistema com a variação do amortecimento (a) no domínio do tempo (b) no domínio da frequência.	51
Fig. 4.4. Modelo simplificado das fontes de ruído de fase no sistema PLL.	52
Fig. 4.5. Função de transferência típica para o ruído dentro da banda (<i>in-band noise</i>) [2].	53
Fig. 4.6. Função de transferência típica para o ruído do VCO [2].	53
Fig. 4.7. Determinação do ponto ótimo para a largura de banda do PLL em termos de ruído de fase de saída [2].	54
Fig. 5.1. Arquitetura do PLL proposto; laço analógico: PFD digital, <i>charge-pump</i> e componentes RC; laço digital: PFD digital, amostragem da diferença de fase, contador crescente / decrescente e conversor D/A.	56
Fig. 5.2. Operação do <i>dual-path loop filter</i> usando componentes passivos [6].	56
Fig. 5.3. Lugar das raízes da solução 1 para 2 pólos na origem e um zero alocado em 0.7 rad/s.	59
Fig. 5.4. Resultados no MATLAB para solução 1 em malha aberta, margem de fase e diagrama de Bode.	59
Fig. 5.5. Esquemático no Simulink da solução 1 para $b = 0.1$.	60
Fig. 5.6. Resposta ao degrau unitário.	60
Fig. 5.7. Lugar das raízes da solução 2 para 1 pólo na origem, um pólo em -4 rad/s e um zero alocado em $-0,25$ rad/s.	61
Fig. 5.8. Resultados no MATLAB para solução 2 em malha aberta, margem de fase e diagrama de Bode.	61
Fig. 5.9. Esquemático no Simulink da solução 2 para $a = 0,25$ e $b = 4$ (a) com componentes passivos (b) usando um laço digital.	62
Fig. 5.10. Resposta ao degrau para o <i>loop filter</i> (a) contínuo (b) usando um laço discreto com período de amostragem de 0.1 s.	62
Fig. 5.11. O PLL simulado no Simulink usando um laço digital $TK_2/(z-1)$, como na solução 2.	63
Fig. 5.12. PFD usado no laço analógico.	63

Fig. 5.13. PFD digital e um contador crescente/decrecente utilizado como <i>loop filter</i> digital (Delay1 = T).	64
Fig. 5.14. O acumulador discreto empregando um contador crescente/decrecente e um conversor D/A. O número de bits do contador depende do projeto do <i>loop filter</i>	64
Fig. 5.15. Bloco que gera pulsos para amostrar a diferença de fase.	65
Fig. 5.16. Modo utilizado na amostragem da diferença de fase.	66
Fig. 5.17. Bloco que gera pulsos para amostrar a diferença de fase modificado	66
Fig. 5.18. Representação dos laços que compõem a tensão de controle do VCO.	69
Fig. 5.19. Técnica aplicada ao <i>loop filter</i> digital para controle do ganho.	71
Fig. 5.20. Resultado do cálculo no MATLAB da margem de fase e ω_c (a) usando um divisor por 8 (b) com um divisor por 16.	73
Fig. 5.21. Resposta a um degrau de 5.1 MHz step, $\omega_p = 2\pi \times 200$ kHz ou $\omega_{pi} = i2\pi \times 50$ kHz e N = 1, o amortecimento é aumentado por 61/16 (a) laço analógico (b) laço digital (c) tensão de controle.	74
Fig. 5.22. O trecho final da simulação de transiente para um degrau de 5.1 MHz mostrando o erro de fase em 150 μ s de transitório (a) a versão analógica da simulação usando o chaveamento (c) para a arquitetura proposta.	74
Fig. 5.23. O resultado de simulação apresentado na Fig. 5.22, com ênfase na redução da ondulação obtida pelo aumento do amortecimento quando o PLL está quase em “regime permanente”.	75
Fig. 5.24. Simulação de <i>settling time</i> semelhante a da Fig. 5.21, para um tuning range de 40MHz \times 4 (devido a uma seleção direta de canal de 2 bits) de 832 a 992 (MHz), o qual atende a alguns padrões de RFID, como o europeu 865 MHz a 868 MHz e o americano 902 MHz a 928 MHz.	76
Fig. 5.25. Simulação apresentada na Fig. 5.24 com chaveamento em 135 μ s.	76
Fig. 5.26. Simulação apresentada na Fig. 5.24 com chaveamento em 185 μ s.	77
Fig. 5.27. Diagrama em blocos do PLL proposto (a) diagrama atual (b) diagrama equivalente usado na determinação da equivalência discreta de primeira ordem.	78
Fig. 5.28. Lugar das raízes e diagrama de Bode do PLL híbrido para $K_2/2$ após o chaveamento.	81
Fig. 5.29. Posição dos pólos em malha fechada para $K_2/2$	82
Fig. 5.30. Resposta ao degrau para $K_2/2$	82
Fig. 5.31. Lugar das raízes e diagrama de Bode do PLL híbrido para $K_2/(2^8)$ após o chaveamento.	83
Fig. 5.32. Posição dos pólos em malha fechada para $K_2/(2^8)$	83
Fig. 5.33. Resposta ao degrau para $K_2/(2^8)$	84
Fig. 5.34. Simulação apresentada na Fig. 5.21 demonstrando um menor erro de regime depois do chaveamento, comparado a resultados prévios (Fig. 2.15 e Fig. 2.16)	84
Fig. 5.35. Simulação de ruído no simulink para a topologia de PLL proposto.	85
Fig. 5.36. Simulação de ruído no Simulink para 3 estados do <i>loop filter</i> com N = 16 (1) antes do chaveamento com $\omega_c = 2\pi \times 50$ kHz (2) depois do chaveamento com $\omega_c = 2\pi \times 12.5$ kHz (3) depois do chaveamento com $\omega_c \cong 2\pi \times 12.5$ kHz e $k_2 = k_2/4$	86
Fig. 5.37. Simulação de ruído no Simulink para 3 estados do <i>loop filter</i> com N = 2412 (1) antes do chaveamento com $\omega_c = 2\pi \times 50$ kHz (2) depois do chaveamento com $\omega_c = 2\pi \times 12.5$ kHz (3) depois do chaveamento com $\omega_c \cong 2\pi \times 12.5$ kHz e $k_2 = k_2/4$	87
Fig. 6.1. Metodologia usada no projeto da arquitetura proposta.	90
Fig. 6.2. Estrutura usada para simulação em nível de sistema da arquitetura <i>dual-path</i> proposta utilizando modelos comportamentais da biblioteca da Mentor Graphics.	93

Fig. 6.3. Resultado da simulação de tempo de estabelecimento para um degrau de 40 MHz (a) para o laço digital, tensão de controle e laço analógico (b) mostrando em detalhe a amplitude da tensão de controle.	94
Fig. 6.4. Representação em blocos do esquemático em nível de transistor para o sintetizador, usando a arquitetura <i>dual-path</i> proposta.	96
Fig. 6.5. Detector de fase e frequência empregado.	98
Fig. 6.6. (a) <i>Charge pump</i> convencional (b) Topologia de <i>charge pump</i> utilizada.....	99
Fig. 6.7. <i>Charge Pump</i>	100
Fig. 6.8. Resultados da simulação de córner para a tensão de saída e <i>charge-sharing</i> (a) circuito original (b) topologia proposta.	101
Fig. 6.9. Topologia de VCO proposto para implementação.	103
Fig. 6.10. Curva da capacitância de porta do transistor em função da tensão de substrato....	105
Fig. 6.11. Três bits da matriz de capacitores controlada pelo <i>loop filter</i> digital	105
Fig. 6.12. Um dos blocos usados na matriz de capacitores contendo quatro bits e um capacitor em série.....	106
Fig. 6.13. MOSFETs usados na seleção direta de canal.....	106
Fig. 6.14. Simulação paramétrica da frequência do oscilador em função da tensão de controle. As curvas representam as quatro curvas possíveis usando seleção direta de canal.....	107
Fig. 6.15. Ruído de fase do VCO na frequência central.....	108
Fig. 6.16. Esquemático do VCO modificado em forma modular para facilitar o processo de simulação pós-leiaute	109
Fig. 6.17. Circuito para seleção direta de canal, proposto durante o projeto pós-leiaute.....	110
Fig. 6.18. Ruído de fase do VCO na frequência central, após a extração de parasitas	110
Fig. 6.19. Leiaute do VCO híbrido proposto contendo indutores, transistor de cauda, resistor, capacitor de filtro, MOSFET varactor da tensão de controle, banco de capacitores, transistores cruzados e roteamento; As dimensões x e y desse leiaute são $632 \times 543,7$ ($\mu\text{m} \times \mu\text{m}$).	111
Fig. 6.20. Representação de circuitos CML (a) usando chaves (b) usando um circuito simplificado.	112
Fig. 6.21. Circuito que divide a frequência por dois usando <i>latches</i> tipo D de alta frequência.	115
Fig. 6.22. <i>Latch</i> tipo D digital usado no circuito divisor por dois.....	115
Fig. 6.23. Ruído de fase da saída do divisor simulado usando simulação de transiente com (a) para o melhor resultado alcançado usando o VCO com transistor de cauda NMOS (b) para o VCO usando transistor de cauda PMOS e o divisor da Fig. 6.9.....	116
Fig. 6.24. Circuito utilizado no processo de extração para gerar o macromodelo do conjunto VCO e prescaler.....	117
Fig. 6.25. Resultado da extração do circuito para uma variação da tensão de controle de 0 a 2V com passos de 0.1V (a) ruído de fase do conjunto para um <i>offset</i> de 1MHz (b) amplitude das 15 harmônicas simuladas.	118
Fig. 6.26. Esquemático do Divisor Programável usando a técnica <i>Pulse Swallow</i>	119
Fig. 6.27. <i>Main Counter</i>	120
Fig. 6.28. <i>Swallow Counter</i>	121
Fig. 6.29. <i>Latch</i> tipo D usando SCL (<i>Source Coupled Logic</i>) usado no circuito divisor por dois do prescaler.	122
Fig. 6.30. Circuito de polarização do <i>latch</i> usando SCL proposto.....	122
Fig. 6.31. Resultado de simulação de Monte Carlo, com 100 execuções, para o divisor de frequências da Fig. 6.29.....	123
Fig. 6.32. Resultado da simulação de transiente, usando a extração de parasitas do leiaute, para o divisor de frequências da Fig. 6.29.....	123

Fig. 6.33. Leiaute do divisor de frequências por 2 usando células SCL	124
Fig. 6.34. <i>Prescaler</i>	125
Fig. 6.35. As duas possibilidades de comportamento da saída do segundo estágio.	126
Fig. 6.36. Multiplexador.	126
Fig. 6.37. Circuito usado na detecção da seqüência.	127
Fig. 6.38. Formas de onda na detecção da seqüência (a) saída (b) seqüência desejada (c) seqüência não desejada.	127
Fig. 6.39. Leiaute do prescaler	128
Fig. 6.40. Detector de fase com saída amostrada.	129
Fig. 6.41. Bloco que gera pulsos no <i>clock</i> do contador em tecnologia CMOS.	129
Fig. 6.42. 1 bit do contador crescente/decrescente usando somente células standard.	131
Fig. 6.43. Bits menos significativos da saída do <i>loop filter</i> digital.	132
Fig. 6.44. Esquemático do <i>loop filter</i> digital usado em simulações para testar sua funcionalidade.	132
Fig. 6.45. Simulação para verificar o funcionamento do loop filter digital na detecção da diferença de fase.	133
Fig. 6.46. Comportamento do contador quando a diferença de fase é detectada (a) bit menos significativo no momento em que se inicia a contagem (b) bit menos significativo no momento em que se finaliza a contagem.	134
Fig. 6.47. Leiaute do circuito de PFD (<i>Phase Frequency Detector</i>) amostrado	134
Fig. 6.48. Leiaute do contador crescente/decrescente usando as técnicas digitais propostas.	134
Fig. 6.49. Esquemático do <i>loop filter</i> digital usando modelos comportamentais utilizado nas simulações de ruído e tempo de estabelecimento.	136
Fig. 6.50. Esquemático de teste do PLL convencional usando os modelos comportamentais que foram calibrados com os blocos em nível de transistor.	136
Fig. 6.51. Esquemático de teste da PLL proposto usando os modelos comportamentais que foram calibrados com os blocos em nível de transistor.	137
Fig. 6.52. Resultado de simulação de ruído de fase, obtido através de simulação transiente, usando o macromodelo em PPV do VCO proposto.	138
Fig. 6.53. Resultado de simulação de ruído de fase obtida através de simulação transiente usando PPV (a) para o PLL convencional (b) para o PLL usando dual-path	139
Fig. 6.54. Resultado de simulação de <i>injection locking</i> com um sinal de 1.8GHz-1mA injetado, obtida através de simulação transiente usando PPV (a) para o PLL convencional (b) para o PLL usando <i>dual-path</i>	140
Fig. 6.55. Resultado de simulação de ruído de fase para uma injeção de jitter de entrada de 0,1 ps (ciclo-para-ciclo) obtida através de simulação transiente usando PPV (a) para o PLL convencional (b) para o PLL usando dual-path	141

SUMÁRIO

Capítulo I.....	14
1. Introdução.....	14
1.1. Motivação.....	14
1.1. Revisão da Literatura.....	16
1.2. Arquitetura Proposta.....	18
1.3. Organização desta Tese.....	19
Capítulo II.....	20
2. Sistemas Sintetizadores de Freqüências.....	20
2.1. Aplicação do Sintetizador de Freqüência em Sistemas de Comunicação Digital.....	20
2.2. Especificações dos Padrões de Comunicação.....	21
2.2.1. Banda de Sintonia (<i>Tuning Range</i>).....	22
2.2.2. Tempo de Estabelecimento (<i>Settling Time</i>).....	22
2.3. Identificação das Especificações.....	22
Capítulo III.....	25
3. Osciladores Controlados e Teoria de Ruído de Fase.....	25
3.1. Ruído de Fase em Osciladores Controlados.....	26
3.1.1. Modelagem de Osciladores Controlados.....	27
3.1.2. Introdução a Teoria de Ruído de Fase.....	30
3.1.3. Teoria de Ruído de Fase Linear e Variante no Tempo.....	35
Capítulo IV.....	46
4. Projeto de Sintetizadores de Freqüências Baseados em PLL.....	46
4.1. Tipos de PLL.....	46
4.2. Dinâmica de Sistemas PLL.....	47
4.3. Análise de Ruído de Fase e Espúrios em Sistemas PLL.....	51
Capítulo V.....	55
5. Sintetizador de Freqüência com <i>Dual-Path</i> e Técnicas Híbridas Analógicas e Digitais Aplicadas ao <i>Loop-Filter</i>	55
5.1. Descrição da Arquitetura.....	55
5.2. <i>Dual Path Loop Filter</i>	56
5.3. Blocos Usados nas Simulações.....	63
5.4. Projeto do <i>Loop Filter</i> Híbrido usando Emulação.....	64
5.4.1. Amostragem da Diferença de Fase.....	65
5.4.2. Equacionamento do <i>Loop Filter</i> Híbrido Analógico e Digital.....	68
5.4.3. Técnica para Redução de Tons Espúrios.....	70
5.4.4. Projeto da Estabilidade.....	71
5.4.5. Comparação dos Resultados.....	75
5.5. Abordagem Usando Controle Digital.....	77
5.5.1. Projeto do <i>Loop Filter</i> Usando Controle Digital.....	80
5.6. Simulação de Ruído de Fase no Simulink.....	85
Capítulo VI.....	88
6. Metodologia de Projeto Utilizada e Implementação da Arquitetura Proposta.....	88
6.1. Metodologia de Projeto do Circuito Integrado.....	89
6.2. Projeto do PLL e Simulações Usando Modelos Comportamentais.....	93
6.3. Projeto dos Blocos.....	95
6.3.1. Projeto do Laço Analógico.....	97
6.3.2. Projeto do VCO e Blocos Divisores de Alta Freqüência.....	102
6.3.3. Projeto do Divisor Programável.....	118

6.3.4. Projeto do Loop Filter Digital Utilizando Ferramenta de EDA	128
6.4. Projeto do PLL e Considerações de Desempenho	135
Capítulo VII.....	143
7. Conclusão	143
Referências.....	146

Capítulo I

1. Introdução

Sistemas de comunicação sem fio (*wireless*) têm sido alvo de muitas pesquisas, desde meados da década de 1990. A demanda por esses dispositivos impulsionou novas tecnologias para osciladores, novas arquiteturas de transceptores e soluções para a viabilização e otimização dessas propostas. Com o surgimento de novos processos de fabricação e aplicações, as pesquisas puderam ser direcionadas para encontrar soluções para cada aplicação e para cada necessidade específica do mercado. Na maioria dos sistemas de comunicação, para atingir as especificações exigidas em cada aplicação, são necessárias soluções diferentes para o oscilador local. Por isso, um dos blocos que mais influencia a aplicação de um sistema é o sintetizador de frequência, que atua como o oscilador local. O sintetizador de frequência é o bloco responsável pela geração da frequência dos canais da portadora de sistemas de comunicação. Devido às exigências de custo, dissipação, integração e tensão de alimentação o sintetizador de frequência ainda é considerado um dos gargalos dos projetos de sistemas *wireless* [1].

1.1. Motivação

Nos últimos anos, tem havido um enorme crescimento no desenvolvimento de sistemas de comunicação que utilizam modulação digital, especialmente nas áreas de telefonia celular, comunicação via satélite e sistemas *wireless* LAN (*Local Area Network*). Com o surgimento de padrões de comunicação 3G, RFID (*Radio Frequency Identification*) e *ultra wideband*, é provável que essa tendência se mantenha. A gama de topologias disponíveis atualmente não é suficiente para atender, com alto desempenho e boa relação custo-benefício, a toda a demanda dos

sistemas usados no mercado de consumo. Considerando que há muitos sistemas em uso sem padronização deve-se prever que as pesquisas nessa área devam perdurar por muitos anos.

Devido à competitividade desse mercado, reduzir o custo da implementação do circuito integrado é um dos pontos principais do projeto de sistemas de rádio frequência (RF) [2]. Essa tendência da indústria refletiu-se nas pesquisas acadêmicas e na busca pela adequação das tecnologias usadas em RF. A transição entre o uso de tecnologias onerosas, como o Arseneto de Gálio (GaAs), até o uso de CMOS digital para realizar rádios com mais de 5 GHz durou décadas, a custo de pesquisas massivas [3]. As tecnologias mais utilizadas nos transceptores atualmente são GaAs, Silício-Germânio (SiGe) e CMOS. O parâmetro chave quando se refere ao uso de tecnologias de fabricação é a frequência de transição (f_T) ou frequência de ganho unitário do transistor. Como exemplo, uma das principais vantagens do transistor bipolar em relação ao MOSFET vem do fato de sua f_T ser maior, dentre outras vantagens. O processo SiGe tem sido muito utilizado em circuitos de RF para aplicações em frequências na ordem de GHz nos últimos anos e devido à combinação de características importantes, como redução de ruído, baixa dissipação de potência, maior f_T , dentre outras, os processos BiCMOS em SiGe têm sido muito empregados na indústria até os dias atuais. As duas principais vantagens da integração de circuitos de comunicação usando tecnologia CMOS são o custo e a compatibilidade com o *baseband* digital. Essas duas vantagens justificaram a criação de processos CMOS não-convencionais com a inclusão de novos materiais, o escalonamento das tecnologias e principalmente o desenvolvimento de novas arquiteturas, o que possibilitou que a integração do *RF front-end* usando tecnologia CMOS se tornasse atrativa [1] [3]. A própria tendência da indústria de circuitos integrados de aumentar a integração utilizando tecnologia CMOS digital estabeleceu essa idéia. A redução das dimensões mínimas do processo CMOS (escalonamento) tem possibilitado que propostas de sistemas sintetizadores de frequência usando apenas técnicas digitais sejam viáveis. Nos novos processos de fabricação, lançados no início desse século, a resolução dos sinais digitais no tempo é maior que a resolução da tensão de um sinal analógico [4]. Arquiteturas de sintetizadores conhecidas como *all-digital* têm se popularizado atualmente [1]. Porém, mesmo nessas técnicas alguns componentes passivos ainda perduram [5]. Pode-se imaginar que quase todos os blocos do transceptor, exceto o LNA, sejam

implementados digitalmente daqui a alguns anos [3], mas é difícil prever qual a técnica será usada nos sistemas de comunicação no futuro. Na próxima subseção, realiza-se uma breve revisão do estado da arte que se encontram as pesquisas de sintetizadores de frequência, que é o tema desse estudo.

1.1. Revisão da Literatura

Sintetizadores de frequência baseados em PLL (*Phase-Locked Loop*) ainda são os sistemas mais utilizados como oscilador local. Esses sintetizadores são divididos em dois grandes grupos, de acordo com o método de divisão da frequência utilizado. O primeiro grupo é o dos sintetizadores de frequência por divisão inteira (*integer-N*), que devem ter razão de divisão alta e frequência de referência muito baixa, e que mesmo assim possuem baixa resolução. Sua velocidade de chaveamento do canal (*lock time*) é mais lenta e os tons espúrios aparecerem em frequências de *offset* da portadora iguais à frequência de referência. Além disso, o ruído de fase, causado pela divisão de frequência é proporcionalmente aumentado [6]. A alternativa para contornar estes problemas é empregar o segundo grupo, denominado de sintetizadores de frequência por divisão fracionária (*fractional-N*), já que estes possibilitam maiores resoluções de frequências de saída e maior facilidade na escolha da frequência de referência [7]. Esses sintetizadores permitem frequências de referência maiores, sem comprometer a resolução da frequência de saída, que pode ser fracionada em valores muito pequenos. Sua grande desvantagem é a geração de tons digitais espúrios, chamados de espúrios fracionados [8]. Podem-se suprimir os espúrios com uma banda pequena no *loop filter*, entretanto, geralmente opta-se por transformar os tons espúrios em ruído de fase, como no caso em que se utiliza VCO's (*Voltage-Controlled Oscillators*) com multi-fase e randomiza-se o *jitter* por interpolação [9], [10]. Esse método possibilita o aumento da largura de banda do *loop filter*, o que permite um *lock time* mais rápido. Recentemente, muitas técnicas de redução de tons espúrios são combinadas para melhorar os seus desempenhos [11], [12], geralmente utilizando modulação $\Sigma\Delta$, que possui a vantagem de não inserir ruído *flicker* ($1/f^2$). Essas abordagens visam reduzir o erro de quantização transformado em ruído de fase. Uma modelagem interessante da influência do ruído de fase introduzida pela modulação $\Sigma\Delta$, no

domínio-z discreto, foi proposta em [13]. Esse modelo utiliza uma arquitetura, com implementação digital, mais geral e comumente usada, a MASH (*Multi-Stage Noise Shaping*) [6]. Algumas propostas que incorporam processamento digital e, até mesmo, topologias de transceptor totalmente digitais têm sido apresentadas para reduzir área e custo [14], [15], [16]. Os sintetizadores chamados de *all-digital* têm ganhado espaço, apesar da complexidade do seu projeto [12]. Os sintetizadores *Direct Digital Synthesis* (DDS) têm alta resolução e rapidez, mas operam em frequências limitadas (<GHz), consomem muita potência e área. Algumas técnicas usam a combinação destes com os sintetizadores baseados em PLL para compensar essas desvantagens.

Dentre os tipos de sintetizadores citados, cada um tem vantagens em relação às especificações dos padrões de comunicação, como o máximo tempo de chaveamento do canal (*settling time*), largura da frequência de operação (*tuning range*) e resolução, ou em relação ao custo benefício, como área, robustez e simplicidade. Na proposição de um sintetizador de frequência devem ser considerados esses fatores.

Alguns parâmetros têm especificações conflitantes, como ruído de fase e dissipação de potência. Da mesma forma, algumas técnicas têm sido propostas para melhorar o compromisso entre *settling time* e largura de banda do *loop filter* [17], *settling time* e redução de espúrios [18] ou *tuning range* e capacitâncias do *loop filter* [19]. A tabela I mostra alguns desses compromissos. Tomando as especificações citadas como sendo as principais especificações, seria interessante que uma técnica pudesse englobar todas essas citadas com baixo custo e dissipação de potência.

Tabela I – Compromissos do Projeto de Sintetizadores de Frequência

ω_c	ζ	Resultados
↑	↓	Rapidez no tempo de estabelecimento
↓	↑	Melhor estabilidade
↓	–	Supressão de tons espúrios
–	↑	Redução do overshoot
↓	↑	Supressão de ruído de entrada
↑	–	Menor capacitância do filtro

1.2. Arquitetura Proposta

O enfoque principal da arquitetura proposta é aumentar o desempenho da topologia padrão de sintetizador de frequência baseado em PLL e melhorar a relação custo-benefício do projeto. A abordagem digital é interessante para manter um circuito compacto e rápido [20] [21]. Porém, os *all-digital* PLLs não provêm uma boa resolução sem que o custo da implementação seja comprometido. Um sintetizador com um projeto misto, sendo as técnicas analógicas usadas para aumentar a resolução e as digitais para reduzir o tempo de chaveamento dos canais e fazê-lo compacto, foi proposto em [22]. Nessa proposta foi usada a técnica de interpolação com atraso (*delay-interpolation*) para "eliminar" componentes espúrios, mas não se alcançou uma resolução de frequência boa por causa da ênfase nas técnicas digitais. Um acumulador simples como o proposto em [23] [24] pode ser combinado com um laço analógico para obter um circuito compacto e ainda mantendo-se baixa complexidade no projeto. Implicações diretas, desse tipo de arquitetura, são a geração de tons espúrios e problemas de estabilidade. Por isso, um dos compromissos priorizados nesse trabalho, foi estudar técnicas para realizar a supressão de tons espúrios e o aumento do desempenho do sistema de uma forma geral.

Nessa tese, técnicas de *dual-path* foram aplicadas para flexibilizar o compromisso entre a largura de banda do filtro e a área. Explorando essa proposta obtiveram-se técnicas de redução de espúrios e redução do tempo de estabelecimento. Algumas técnicas, encontradas na literatura, podem ser empregadas para redução da área do *loop filter* [25], [18], mas normalmente utilizam dispositivos ativos, como amplificadores operacionais, o que aumenta substancialmente o ruído na tensão de controle [26]. Técnicas semelhantes para redução de espúrios e tempo de estabelecimento poderiam ser usadas, usando chaveamento de capacitores [27], donde se conclui que o uso de técnicas digitais é justificado pela redução no custo do projeto. A complexidade desse recurso, no entanto, deve requerer que a arquitetura proposta seja apenas empregada em alguns tipos de aplicações. Um dos objetivos dessa tese é ampliar as possibilidades de aplicação desse tipo de arquitetura, desde que ela seja robusta e sem a adição de técnicas muito complexas.

Para validar a arquitetura propõe-se que a implementação da topologia desenvolvida seja aplicada em identificação por rádio frequência (RFID) na banda de UHF (*Ultra High Frequency*). Neste tipo de aplicação deseja-se um custo baixo, um circuito compacto e com baixa complexidade [28]. Seguindo a tendência das publicações atuais, o sintetizador é do tipo *multi-standard*, o que necessita de uma faixa larga de frequência do sintetizador. A topologia proposta deve operar na faixa de 850 MHz a 1010 MHz, obtendo uma banda de 160 MHz. Essa faixa de frequência atende ao padrão da Coréia do Sul, de 908 MHz a 914 MHz e o do Japão, de 950 MHz a 956 MHz, como muitos outros padrões asiáticos [29] [30]. A tecnologia usada no projeto é a CMOS 0.35 μm da AMS compatível com o processo CMOS padrão.

1.3. Organização desta Tese

Nesse capítulo foi dada uma introdução aos tópicos principais abordados na tese, bem como, a motivação para a realização da tese e o embasamento para a sua proposição. No capítulo II são abordadas a aplicação dos sintetizadores de frequências em sistemas *wireless* e a determinação das especificações a partir dos padrões de comunicação. A teoria de ruído de fase aplicada a osciladores controlados é introduzida no capítulo III. No capítulo IV é dada uma introdução ao projeto de sintetizadores de frequências baseados em PLL e algumas equações importantes para o projeto da arquitetura são deduzidas. A arquitetura proposta e o seu desenvolvimento são descritos no capítulo V. No capítulo VI são apresentadas a metodologia de projeto de circuito integrado empregada e a descrição do projeto dos blocos. No capítulo VII as conclusões são apresentadas, bem como sugestões de trabalhos futuros.

Capítulo II

2. Sistemas Sintetizadores de Freqüências

Nesse capítulo são abordados os conceitos fundamentais usados no projeto de sintetizadores de freqüências. Algumas definições envolvidas no projeto de sistemas de RF relacionadas à sintetizadores de freqüência são dadas. As aplicações mais comuns serão apresentadas e a teoria envolvida para determinação de suas especificações serão discutidas.

2.1. Aplicação do Sintetizador de Freqüência em Sistemas de Comunicação Digital

Na Fig. 2.1, um transceptor genérico é representado contendo apenas os blocos de processamento do sinal da portadora [31]. Qualquer que seja a modulação digital empregada, o sinal de RF (portadora) deve ser transladado para uma freqüência inferior e vice-versa, no caso do transmissor através do *up-conversion mixer* e no receptor do *down-conversion mixer*. Essa freqüência pode ser a freqüência intermediária ou a da banda base (*baseband*).

Os blocos que compõem o transceptor são determinados pelo tipo de modulação e acesso utilizados. O dispositivo capaz de gerar as freqüências dos canais, necessária para acessar o sinal transmitido ou transmiti-lo, é chamado sintetizador de freqüência.

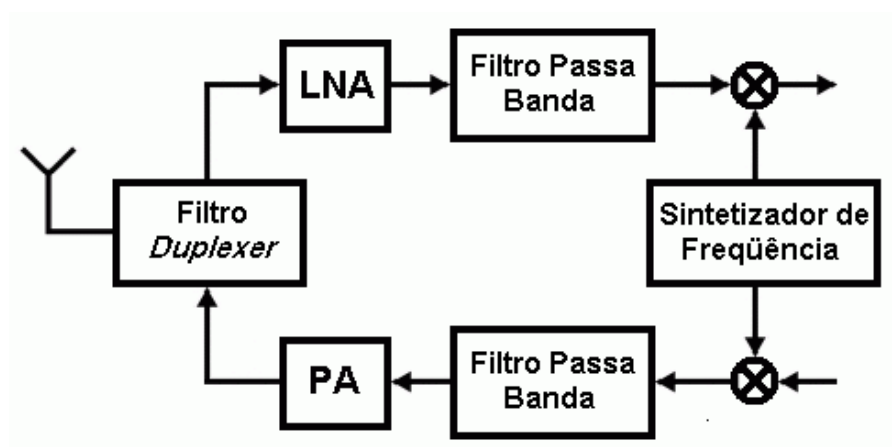


Fig. 2.1. Sistema genérico de um transceptor usando *Duplexer Filter*.

2.2. Especificações dos Padrões de Comunicação

Na tabela I estão descritas as principais especificações dos padrões de comunicação *wireless* mais utilizados atualmente. O projeto de um sintetizador depende da interpretação desses valores e da determinação de especificações do projeto do sintetizador a partir daqueles. A seguir, serão descritos alguns parâmetros importantes no projeto de um sintetizador de frequência.

Tabela I – Principais padrões *Wireless*

	Bluetooth	802.11a	802.11b	802.11g
Taxa de transmissão	1 Mbps	54 Mbps	11 Mbps	54 Mbps
Sensibilidade	-70 dBm	-82 dBm	-76 dBm	-76 dBm
Frame Error Rate	10^{-3} (BER)	10^{-5}	8×10^{-2}	8×10^{-2}
Banda (MHz)	2400–2479	5180–5805	2412–2472	2412–2472
Channel Spacing	1 MHz	20 MHz	5 MHz	20 MHz
Exatidão	± 75 kHz	± 20 ppm	± 25 ppm	± 25 ppm
Tempo de estabelecimento	< 259 μ s	224 μ s	224 μ s	224 μ s
Interferência	+ 40 dB at 3 MHz	+ 32 dB at 40 MHz	+ 35 dB at 25 MHz	+ 35 dB at 25 MHz

2.2.1. Banda de Sintonia (*Tuning Range*)

Os padrões de comunicação sem fio utilizam uma banda de frequência pré-definida, por exemplo, a de 2.4 GHz para o Bluetooth. A faixa de frequência de saída do VCO, dentro dessa banda, é chamada de banda de sintonia (*tuning range*). Alguns fatores também determinam a banda de sintonia além do especificado no padrão ou padrões. Em sintetizadores baseados em PLL, normalmente a faixa de frequência do VCO é projetada para ser maior que o necessário. Devido a variações no processo, a faixa de frequência do VCO pode desviar de mais de 10 %, por isso uma margem de segurança será imprescindível para o PLL alcançar as frequências desejadas [26].

2.2.2. Tempo de Estabelecimento (*Settling Time*)

Os sintetizadores de frequências usam uma frequência de referência para gerar as frequências seletoras dos canais. A diferença entre a frequência do canal desejada e a frequência de saída do sintetizador de frequência é a especificação denominada exatidão (*accuracy*). A dinâmica do sistema sintetizador de frequência torna o chaveamento dos canais não-instantâneo. O tempo de estabelecimento (*settling time*) é definido como o tempo despendido para o sintetizador variar por toda a faixa de frequência, o qual significa o máximo tempo necessário para chaveamento dos canais.

2.3. Identificação das Especificações

Os ruídos de fase e tons espúrios (*spectral purity*) não são especificados explicitamente na maioria dos padrões de comunicação [26]. O método geralmente empregado extrai informações das especificações de interferência e BER (*Bit Error Rate*) do padrão para determinar o ruído de fase e tons espúrios tolerados. A seguir será descrito um método para o cálculo desses valores.

Usando todos os valores em dB nos cálculos, a mínima pureza espectral tolerada é determinada subtraindo-se a interferência apresentada no sinal da banda base pelo mínimo *SNR* (*Signal-to-Noise Ratio*) requerido. Na Fig. 2.2 estão

ilustrados os espectros envolvidos antes e após o bloco *down-conversion mixer* e o efeito do ruído de fase no sinal da banda base. Sendo LO (*Local Oscillator*) a frequência de saída do sintetizador, a especificação de ruído de fase pode ser deduzida como a seguir.

Quando o sinal (P_{Sig}) é transladado para DC ou IF (*Intermediate Frequency*) pelo sinal do LO (P_{LO}), a Interferência (P_{Int}) também é transladada para DC ou IF pelo ruído de fase (P_N) sendo acrescentado ao sinal de interesse. Desde que o ruído de fase é um processo aleatório, dado em dBc/Hz, a sua largura de banda (P_{BW}) é adicionada para o cálculo da potência total. A relação sinal ruído (SNR) do sinal da banda básica é a diferença entre a potência dos dois, e ele deve ser maior do que o SNR mínimo necessário para satisfazer os requerimentos de BER do receptor.

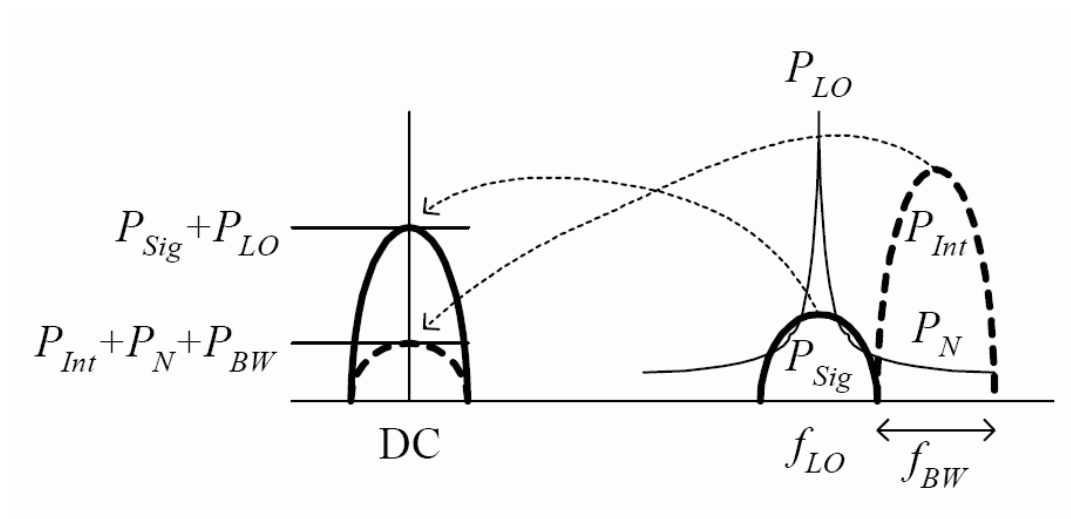


Fig. 2.2. Efeito do ruído de fase na translação do sinal (*down-conversion*).

$$SNR = (P_{Sig} + P_{LO}) - (P_{Int} + P_N + P_{BW}) > SNR_{min} \quad (2.1)$$

e rearranjando para os valores desejados,

$$P_N - P_{LO} < (P_{Sig} - P_{Int}) - P_{BW} - SNR_{min} \quad (2.2)$$

sendo o ruído de fase = $P_N - P_{LO}$.

O ruído de fase pode ser especificado, portanto, através da interferência máxima admitida em relação à potência do sinal, dada em dB, a largura de banda do canal e o SNR_{min} tolerado para o mesmo *offset* de frequência da interferência. Como exemplo, da tabela I, o Bluetooth pode ser especificado resolvendo a equação (2.2), como

$$\text{Ruído de fase} = (-40 \text{ a } 3\text{MHz}) - 60 - 18 = -118 \text{ dBc a } 3\text{MHz da portadora,}$$

sendo a largura de banda do canal de 1 MHz, e portanto $P_{BW} = \log_{10}(10^6) = 60 \text{ dB}$, o mínimo SNR para um BER de 10^{-3} foi determinado como 18 dB [26].

A especificação de tom espúrio é semelhante, sem a necessidade de adicionar a largura de banda do canal, pois os tons espúrios se apresentam em apenas uma frequência. A Fig. 2.3 ilustra o efeito da distorção do oscilador local na conversão do sinal. O máximo espúrio tolerado pode ser encontrado como

$$P_{Sp} - P_{LO} < P_{Sig} - P_{Int} - SNR_{min}$$

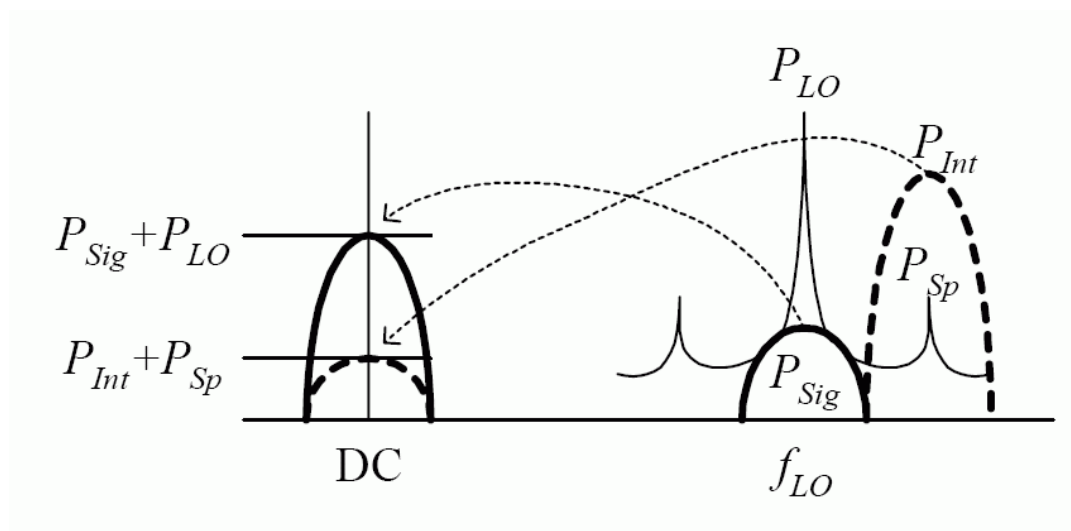


Fig. 2.3. Efeito dos tons espúrios na translação do sinal (*down-conversion*).

Mais detalhes sobre a especificação de sintetizadores de frequências podem ser encontrados em [26].

Capítulo III

3. Osciladores Controlados e Teoria de Ruído de Fase

Osciladores controlados são blocos largamente usados em circuitos de comunicação e de instrumentação, tais como *phase-locked loop* (PLL), moduladores, demoduladores, sintetizadores de frequência e em circuitos de recuperação de base de tempo [32]. Na grande maioria das aplicações de osciladores controlados, estes são controlados por tensão, no entanto há a possibilidade de uso de osciladores controlados por corrente. Um fator decisivo para a escolha entre CCO (*Current-Controlled Oscillator*) e VCO (*Voltage-Controlled Oscillator*) é o tipo de oscilador a empregar. Osciladores controlados com tanque LC têm algumas limitações frente aos osciladores em anel. Geralmente torna-se necessário a sua fabricação usando-se processos especiais (*nonstandard process*), pela necessidade de indutores e, às vezes, varactors. Além disso, ele tem uma faixa de ajuste estreita de frequência e maior área [33]. Entretanto, muitas pesquisas foram dedicadas para resolver essas limitações, devido ao seu fator de qualidade Q ser alto, o que tende a reduzir o consumo de potência [34], seu bom desempenho em frequências altas e menor ruído de fase [35]. Em contrapartida, os osciladores em anel possuem uma faixa de frequência de operação mais larga, custo baixo e maior facilidade de integração. Os osciladores em anel podem ser controlados por corrente ou tensão. Os dois tipos de osciladores citados podem apresentar características diferenciais e a saída em quadratura.

As principais barreiras para a aplicação de osciladores controlados em anel são a velocidade e o desempenho em termos de ruído de fase. Muitas pesquisas são voltadas para aumentar a velocidade dos osciladores em anel [36], [37]. Um grande avanço nessa busca foi possível a partir de propostas de osciladores com multi-defasamento [38], [39]. Desde então, vários circuitos têm sido propostos, com ruído de fase baixo e frequências altas. Nesse contexto, um trabalho foi proposto,

utilizando um CCO e um conversor V-I para aplicação em sistemas com GPS totalmente integrado [40]. Apesar de, nessa aplicação não haver muita exigência (normalizada) sobre o ruído de fase, o sintetizador alcançou -95 dBc/Hz em 1 MHz de *offset*.

3.1. Ruído de Fase em Osciladores Controlados

Aplicações que requerem precisão no sincronismo de fase tornam as pesquisas sobre osciladores e *clocks* com ruído baixo muito atraentes e com grau de exigência alto. O ruído de fase (*phase noise*) é caracterizado por variações randômicas na frequência e ou na fase da forma de onda de saída de osciladores, sendo quantificado pela magnitude do ruído de fase ou *jitter*. O *jitter* pode ser interpretado como a variação da fase de um sinal periódico no tempo [41]. Exemplo de aplicações onde o ruído de fase é um fator limitante para osciladores e *clocks* são os receptores, *local area networks (LANs)*, *disk driver systems* e microprocessadores [42].

Em sistemas *wireless*, o sintetizador de frequência é normalmente implementado usando um PLL, sendo o ruído do oscilador controlado uma das principais contribuições para o ruído de fase do PLL. Um oscilador controlado com ruído de fase baixo é essencial para implementações de sintetizadores de frequência com alto desempenho [43].

O ruído de fase tem sido alvo de inúmeras pesquisas nas últimas décadas, buscando um modelo adequado para explicar seu comportamento. Este tópico, ainda é motivo de alguns trabalhos devido ao lançamento de propostas de implementação e novas tecnologias de fabricação [42], [44]. Nessa seção será apresentada uma introdução à teoria de ruído de fase, restringindo-se às pesquisas realizadas na década de 1990 [45], [46].

3.1.1. Modelagem de Osciladores Controlados

Uma definição usual de ruído de fase é descrita no domínio da frequência, em relação a uma frequência de oscilação ω_0 , geralmente tida como a frequência da portadora em sistemas de comunicação. O espectro ideal e com ruído de fase são representados na Fig. 3.1. O ruído de fase é quantificado pela potência do ruído por unidade de largura de banda em uma frequência $\Delta\omega$ de *offset*, dividido pela potência da portadora. Geralmente a unidade de ruído de fase é dBc/Hz, dBc especifica a portadora como referência, portanto se, por exemplo, o ruído presente na saída do oscilador é -115 dB em relação à potência da portadora em uma frequência de 100 kHz deslocada da frequência ω_0 , então o ruído de fase é

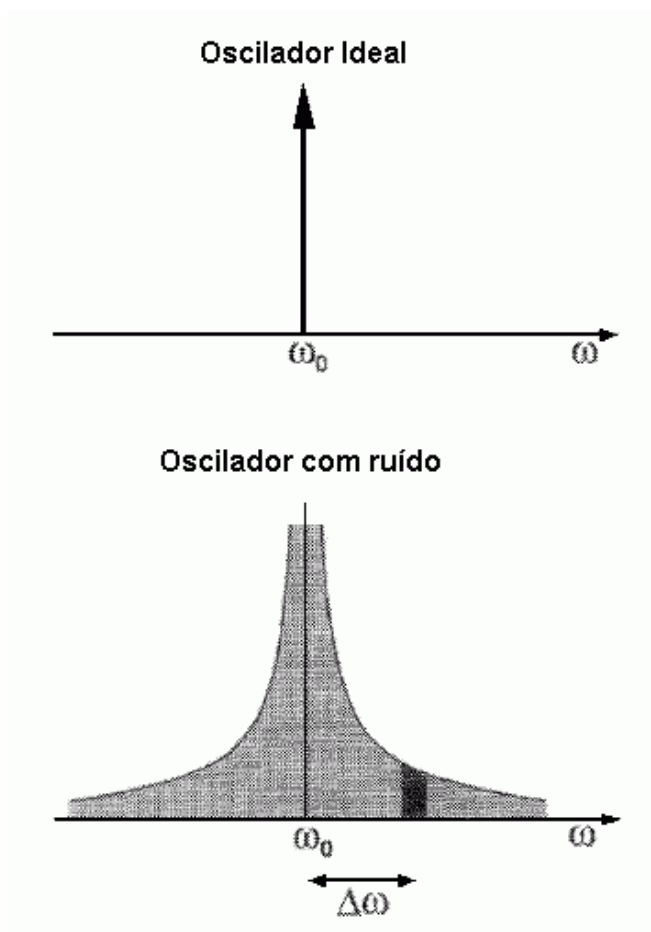


Fig. 3.1. Espectro do ruído de fase em um oscilador

de -115 dBc/Hz em um *offset* de 100 kHz. Pode-se observar que quanto maior o módulo desse parâmetro melhor é a relação sinal-ruído de fase.

Existem mais de uma forma de modelar os osciladores para análise de ruído, por exemplo, o modelo linear e invariante no tempo (LIT) e o linear e variante no tempo (LVT). Apesar do modelo LIT obter uma boa descrição qualitativa do comportamento em relação aos parâmetros, como o fator de qualidade Q, potência dissipada, ou a frequência de oscilação, o modelo LVT providencia maior precisão na determinação do ruído em relação às medições, a qual é exigida nos projetos de sistemas de comunicação modernos. Recentemente, modelos que consideram o oscilador como um sistema não-linear foram apresentados [42], [25], mas estes não serão abordados nessa resenha introdutória.

Um oscilador apresenta não-linearidades devido ao controle de amplitude da forma de onda gerada na saída do sistema. No início da oscilação, há mais pólos do lado direito do plano s . Para que a amplitude da onda pare de crescer é necessário que esses pólos transladem para o eixo imaginário. Algum fator deve limitar a amplitude do sinal, introduzindo não-linearidades à função de transferência das células de atraso. A análise a pequenos sinais é utilizada para obter a frequência de oscilação inicial e posteriormente a frequência de oscilação é reduzida e definida através dos atrasos das células inversoras. As Fig. 3.2 e Fig. 3.3 ilustram o comportamento do mecanismo de limitação da amplitude, para o oscilador LC e em anel respectivamente. Nesse caso a amplitude é limitada pela corrente de cauda I_{SS} desses osciladores.

Osciladores controlados passam por mudanças de frequência constantes, entretanto, após alguns ciclos onde se mantém a mesma forma de onda, pode-se assumir um modelo linear para o sistema do oscilador sem muita perda de generalização.

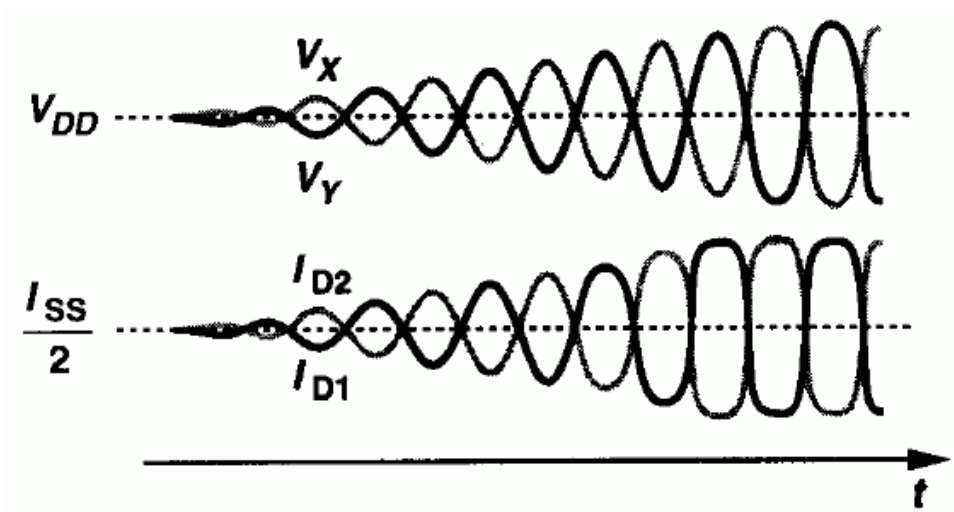


Fig. 3.2. Início da oscilação em um oscilador tanque LC [43]

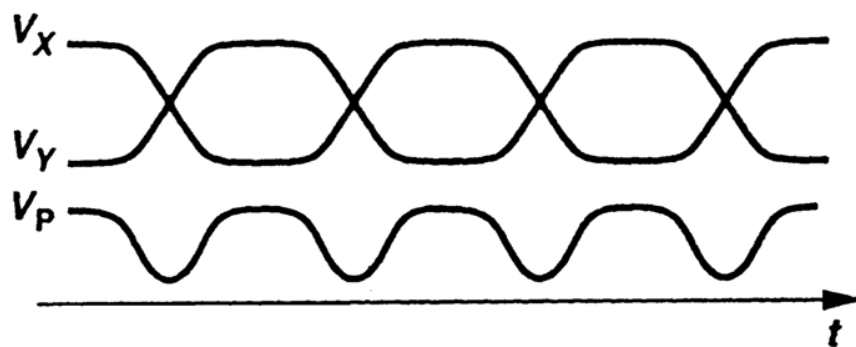


Fig. 3.3. Início da oscilação em um oscilador em anel diferencial [43]

Apesar de parecer um modelo mais conciso, o modelo LIT foi preterido e uma teoria para modelamento do ruído de fase para um sistema linear variante no tempo foi desenvolvida [46], [47]. A observação de alguns fenômenos que não poderiam ser realizados em sistemas LIT reforça a hipótese da variância no tempo. A análise que segue desenvolve algumas definições importantes para um oscilador idealizado e depois para o modelo de sistema LVT de um oscilador, onde ficará clara a justificativa para uma abordagem variante no tempo.

3.1.2. Introdução a Teoria de Ruído de Fase

A definição mais comum para o fator de qualidade Q é $Q = \frac{\omega_0}{BW}$, sendo ω_0 a frequência da portadora e BW a largura de banda em -3dB . Essa equação pode ser aplicada para filtros e circuitos RLC, mas uma definição mais geral é dada por 2π vezes a razão entre a energia armazenada e a energia dissipada por ciclo, essa pode ser medida aplicando uma entrada degrau e observando o decaimento das oscilações [45]. No entanto, essa definição não pode ser aplicada se o circuito não apresentar um comportamento oscilatório. Outra definição para o fator de qualidade de circuitos oscilatórios é expressada por $Q = \frac{\omega_0}{2} \frac{d\Phi}{d\omega}$, sendo Φ a fase do oscilador (ver Fig. 3.4). Pode-se entender o seu significado lembrando que a defasagem total da malha de um oscilador é constante e igual a 360° , sendo assim, na prática, a derivada reflete um fator de qualidade muito alto, se a defasagem deslocar-se de 360° a mudança na frequência é muito pequena. Embora essa definição expresse muito bem o comportamento de circuitos tanques LC, não se pode aplicá-la a circuitos não osciladores, em que a defasagem total é zero para qualquer frequência e, portanto, Q seria igual a zero.

Pode-se definir uma expressão mais geral para o fator de qualidade Q , admitindo-se cada fonte de ruído como uma entrada para um sistema realimentado. Deve-se ressaltar que nessas definições de Q , os osciladores são

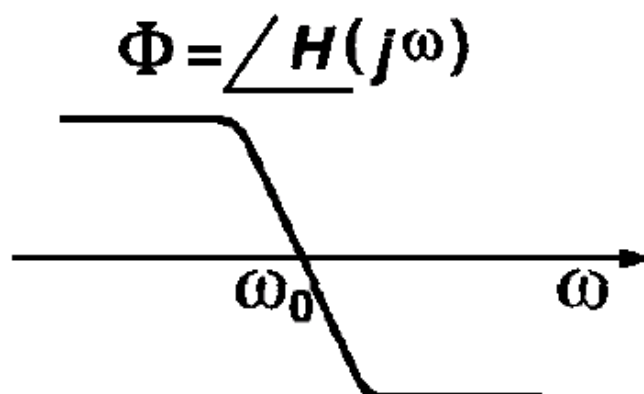


Fig. 3.4. Diagrama de fase de um oscilador [45]

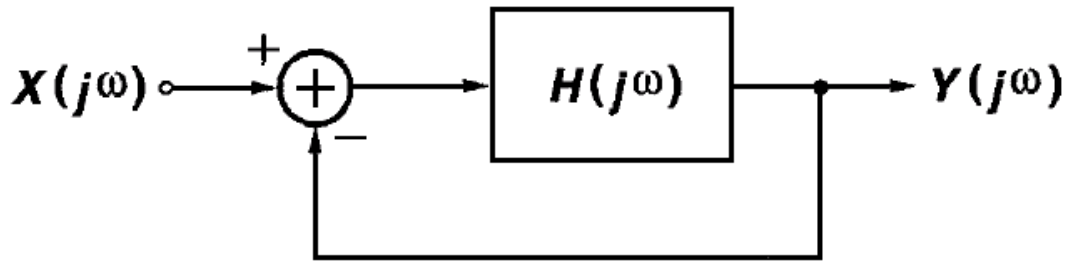


Fig. 3.5. Sistema em malha fechada de um oscilador linear [45]

considerados sistemas LIT. Apesar da análise quantitativa ser enfatizada nesse trabalho, o resultado de análises qualitativas são importantes para a otimização dos osciladores.

Um sistema, como o da Fig. 3.5, pode ser usado para modelar a função transferência para as fontes de ruído de um oscilador, considerando que a frequência de ressonância seja $\omega = \omega_0$ e que

$$\frac{Y}{X}(j\omega) = \frac{H(j\omega)}{1+H(j\omega)} \quad (3.1)$$

Então, se $\frac{Y}{X}(j\omega)$ tende ao infinito, $H(j\omega_0) = -1$, condizente com o princípio de Barkhausen. Para frequências próximas da portadora, $\omega = \omega_0 + \Delta\omega$ e a função transferência de malha aberta, pode ser aproximada como

$$H(j\omega) \approx H(j\omega_0) + \Delta\omega \frac{dH}{d\omega} \quad (3.2)$$

E a função de transferência da entrada para a saída do ruído se torna

$$\frac{Y}{X}[j(\omega_0 + \Delta\omega)] = \frac{H(j\omega_0) + \Delta\omega \frac{dH}{d\omega}}{1 + H(j\omega_0) + \Delta\omega \frac{dH}{d\omega}} \quad (3.3)$$

Desde que $H(j\omega_0) = -1$, e considerando que para casos práticos $|\Delta\omega dH/d\omega| \ll 1$ [45], então (3) pode ser aproximada por

$$\frac{Y}{X} [j(\omega_0 + \Delta\omega)] \approx \frac{-1}{\Delta\omega \frac{dH}{d\omega}} \quad (3.4)$$

A Fig. 3.6 ilustra como a densidade espectral de potência do ruído será moldada pela função de transferência deduzida para (3.4). Nesse exemplo, o ruído térmico é a fonte de ruído e a função transferência típica para a densidade espectral de potência é dada por

$$\left| \frac{Y}{X} [j(\omega_0 + \Delta\omega)] \right|^2 = \frac{1}{(\Delta\omega)^2 \left| \frac{dH}{d\omega} \right|^2} \quad (3.5)$$

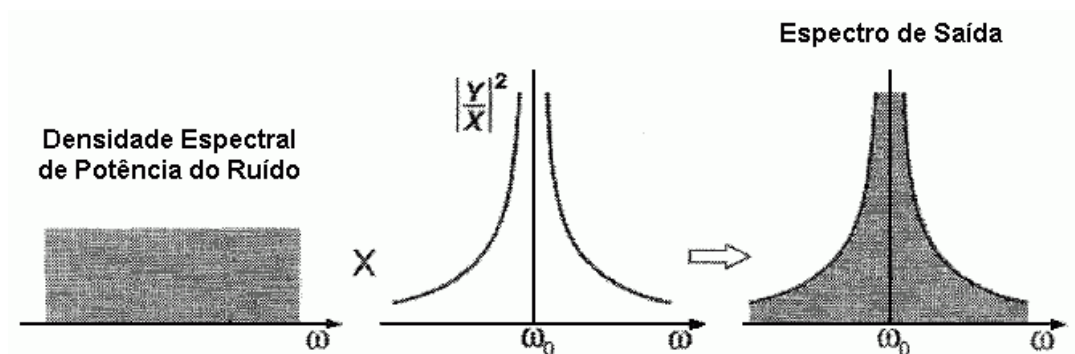


Fig. 3.6. Formato do espectro do ruído ($1/f^2$)

Em notação fasorial pode-se escrever $H(j\omega) = A(\omega)\exp(j\Phi)$, e derivando essa expressão obtém-se:

$$\frac{dH}{d\omega} = \left(\frac{dA}{d\omega} + jA \frac{d\Phi}{d\omega} \right) \exp(j\Phi). \quad (3.6)$$

Sendo $\omega \approx \omega_0$ e $A \approx 1$, (3.5) pode ser reescrita como

$$\left| \frac{Y}{X} [j(\omega_0 + \Delta\omega)] \right|^2 = \frac{1}{(\Delta\omega)^2 \left[\left(\frac{dA}{d\omega} \right)^2 + \left(\frac{d\Phi}{d\omega} \right)^2 \right]}, \quad (3.7)$$

lembrando que o módulo de $\exp(j\Phi)$ é igual a um.

Nesse ponto pode-se introduzir uma equação para o fator de qualidade Q , que é o objetivo dessa dedução. Dessa forma, define-se:

$$Q \equiv \frac{\omega_0}{2} \sqrt{\left(\frac{dA}{d\omega} \right)^2 + \left(\frac{d\Phi}{d\omega} \right)^2}. \quad (3.8)$$

De (3.7) e (3.8) pode-se deduzir que

$$\left| \frac{Y}{X} [j(\omega_0 + \Delta\omega)] \right|^2 = \frac{1}{4Q^2} \left(\frac{\omega_0}{\Delta\omega} \right)^2. \quad (3.9)$$

Equações semelhantes a (3.9) podem ser encontradas em alguns trabalhos para osciladores específicos ou genéricos [48]. Ela é verdadeira para sistemas LIT e geral porque pode ser aplicada na maioria dos osciladores conhecidos [45].

Considerando uma análise para um oscilador RLC ideal, o qual a única fonte de ruído seja proveniente de ruído térmico da resistência, então a densidade espectral média de ruído é [41]

$$\frac{i_n^2}{\Delta f} = \frac{4kT}{R} \quad (3.10)$$

Usando (3.9) e (3.10) e multiplicando por R^2 para obter a saída em tensão, pode-se deduzir que

$$\frac{v_n^2}{\Delta f} = 4kTR \left(\frac{\omega_0}{2Q\Delta\omega} \right)^2. \quad (3.11)$$

Segundo o teorema da equi-partição da termodinâmica, metade do ruído é de amplitude e metade é de fase. Desde que o mecanismo de limite de amplitude dos osciladores introduza metade do ruído total, pode-se aplicar a definição de ruído de fase e determiná-lo como

$$L\{\Delta\omega\} = 10 \log \left[\frac{2kT}{P_{\text{sin}}} \left(\frac{\omega_0}{2Q\Delta\omega} \right)^2 \right], \quad (3.12)$$

sendo P_{sin} a potência do sinal da portadora.

Destes resultados verifica-se que o ruído de fase decai em uma proporção de $1 / (\Delta\omega)^2$. Na prática, as simplificações admitidas implicam que o ruído de fase real, tenha um comportamento menos satisfatório. Um modelo mais realístico atribui fatores empíricos para representar efeitos observados, isso modela o espectro como apresentado na Fig. 3.7. Esse modelo inclui o ruído $1/f$ ou *flicker*

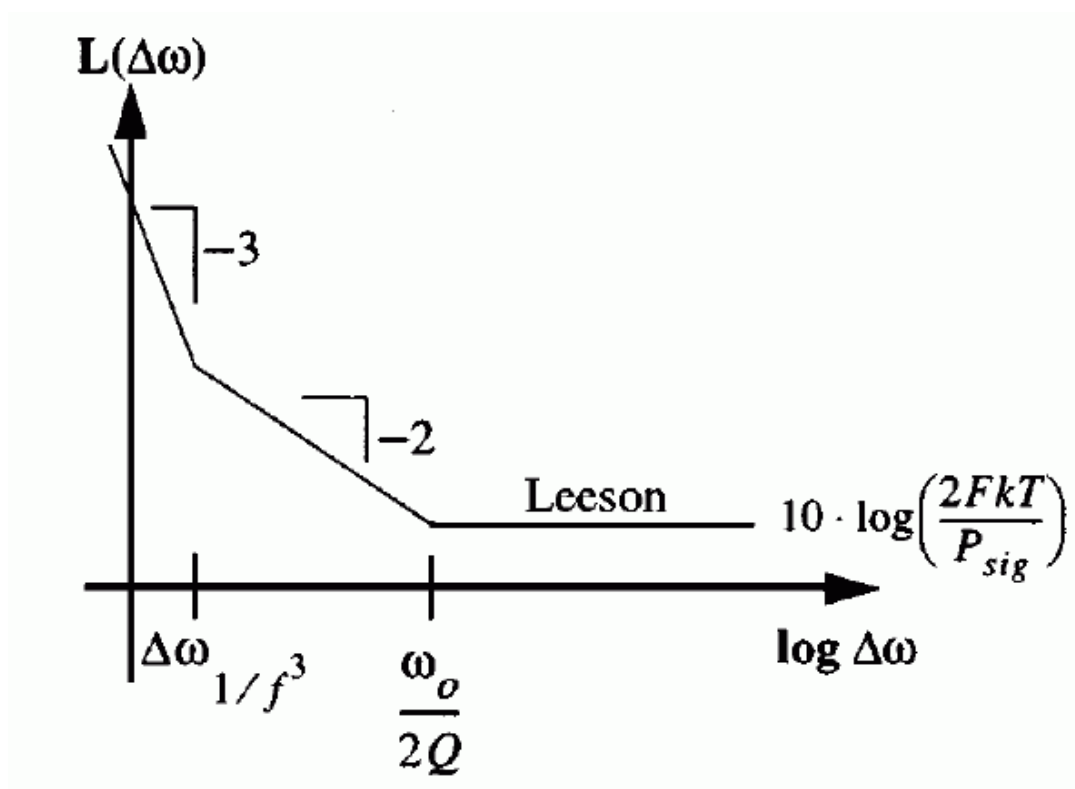


Fig. 3.7. Ruído de fase predito por Leeson \times linear

noise para pequenos *offsets* de freqüência, uma magnitude constante para freqüências superiores a $\omega_0/2Q$ causada por *buffers* externos e uma depreciação comum a todo espectro quantificada pelo fator empírico F . As imperfeições previstas por Leeson são descritas pela seguinte equação [48]:

$$L\{\Delta\omega\} = 10 \log \left\{ \frac{2FkT}{P_{sin}} \left[1 + \left(\frac{\omega_0}{2Q\Delta\omega} \right)^2 \right] \left(1 + \frac{\Delta\omega_{1/f^3}}{|\Delta\omega|} \right) \right\}. \quad (3.13)$$

Nem mesmo a equação (3.13) pode prever quantitativamente o ruído de fase. Essa resolução é muito satisfatoriamente determinada na análise que será realizada para sistemas lineares variantes no tempo, e mais precisa, em estudos recentes que consideram não-linearidades na análise [42], [25].

3.1.3. Teoria de Ruído de Fase Linear e Variante no Tempo

Pode-se mostrar o princípio da variância no tempo, observando o comportamento do sistema oscilatório para pulsos de corrente injetados em instantes diferentes, como os vistos na Fig. 3.8. Na Fig. 3.9(a) o impulso acontece junto com a amplitude máxima da oscilação, o resultado é que a amplitude máxima aumenta por uma quantidade $\Delta V = \Delta Q / C$, mas a fase não se altera.

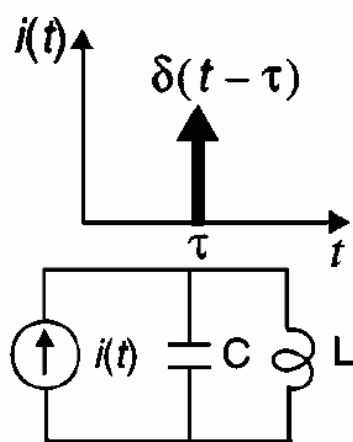


Fig. 3.8. Um oscilador LC excitado por um impulso de corrente.

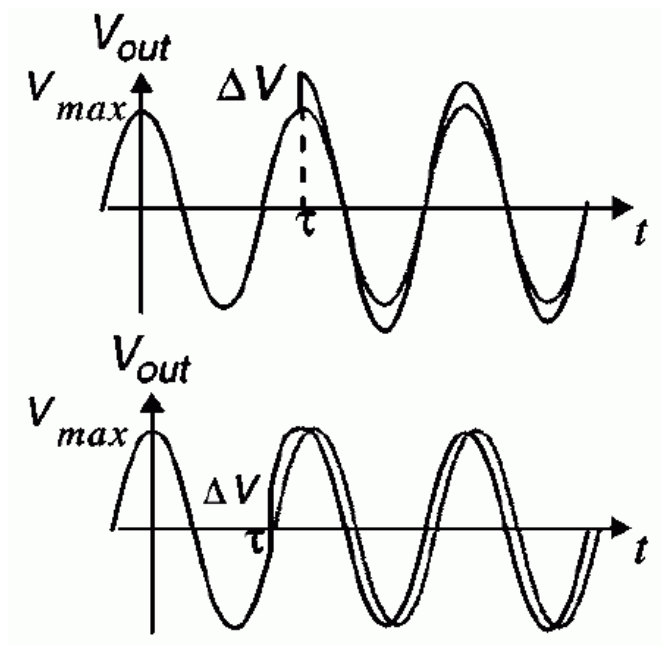


Fig. 3.9. Resposta à entrada impulso [47].

Na Fig. 3.9(b) o impulso é aplicado no cruzamento com o zero. Pode-se concluir que a aplicação de pulsos em diferentes pontos do ciclo afeta a amplitude e a fase da oscilação. Essa característica não pode ser realizada por sistemas sem memória, isto é, sistemas lineares invariantes no tempo.

A resposta ao impulso produz um degrau na fase da oscilação, que pode ser determinada admitindo uma função de sensibilidade ao impulso (ISF) $\Gamma(\omega t)$ [47]. A dedução é mostrada a seguir:

$$h_{\phi}(t, \tau) = \frac{\Gamma(\omega_0 \tau)}{q_{m\acute{a}x}} u(t - \tau), \quad (3.14)$$

Sendo $u(t - \tau)$ a função degrau unitária e $q_{m\acute{a}x}$ a carga máxima deslocada pelo capacitor. A função $\Gamma(x)$ codifica informações sobre a variação na fase do oscilador, sendo uma função periódica com a mesma frequência da oscilação. Quando a amplitude da oscilação for máxima, $\Gamma(x)$ é zero e assumindo seu valor máximo quando o sinal oscilante passa por zero. As representações de funções

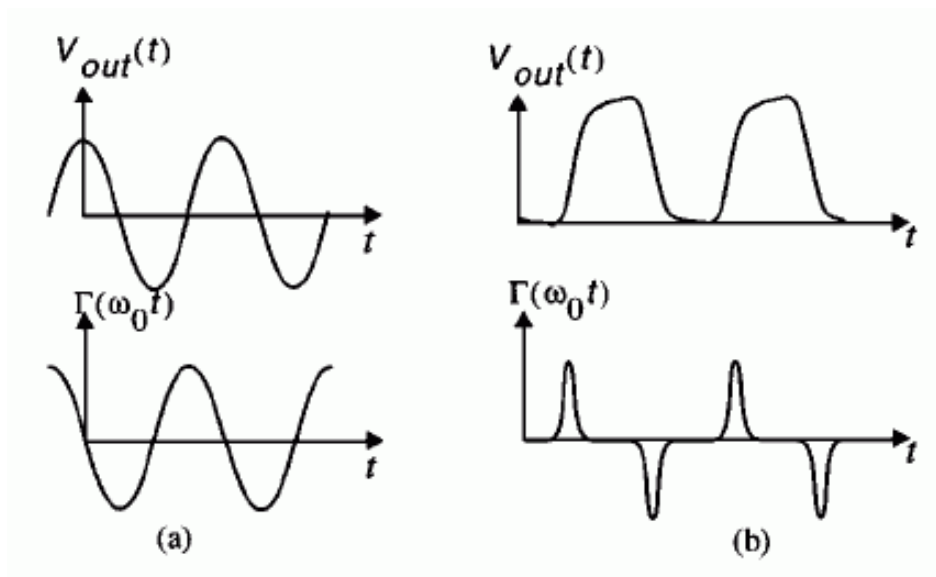


Fig. 3.10. Exemplos de ISF para (a) oscilador LC e (b) oscilador em anel [47].

desse tipo são mostradas na Fig. 3.10. Em geral $\Gamma(x)$ é determinado por simulação, mas há formas analíticas para aproximá-lo [49].

Determinada a função $\Gamma(x)$, pode-se encontrar a fase, através da integral de superposição. Desde que o sistema seja linear

$$\phi(t) = \int_{-\infty}^{\infty} h_{\phi}(t, \tau) i(\tau) d\tau = \frac{1}{q_{\text{máx}}} \int_{-\infty}^{\infty} \Gamma(\omega_0 \tau) i(\tau) d\tau. \quad (3.15)$$

Sendo $\Gamma(x)$ periódica, pode-se expressá-la pela série de Fourier como

$$\Gamma(\omega_0 \tau) = \frac{c_0}{2} + \sum_{n=1}^{\infty} c_n \cos(n\omega_0 \tau + \theta_n). \quad (3.16)$$

O ângulo de fase θ_n será desprezado nas análises que seguem, por não haver relação entre θ_n com o ruído de fase. Substituindo-se (3.16) em (3.15) tem-se

$$\phi(t) = \frac{1}{q_{\text{máx}}} \left[\frac{c_0}{2} \int_{-\infty}^{\infty} i(\tau) d\tau + \sum_{n=1}^{\infty} c_n \int_{-\infty}^{\infty} i(\tau) \cos(n\omega_0 \tau) d\tau \right]. \quad (3.17)$$

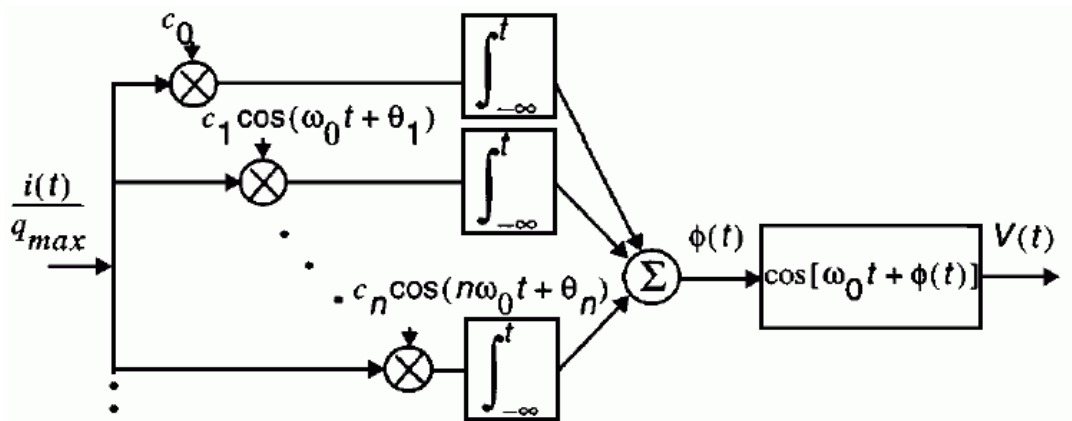


Fig. 3.11. Diagrama de blocos para a modelagem do sistema [47].

A equação (3.17) pode ser representada em diagrama de blocos, como visto na Fig. 3.11. Pode-se verificar que as operações realizadas no sistema da Fig. 3.11 são lineares, condizentes com o modelo procurado.

Considerando um sinal de corrente senoidal injetado no sistema, assim como

$$i(t) = I_m \cos[(m\omega_0 + \Delta\omega)t], \quad (3.18)$$

sendo $\Delta\omega \ll \omega_0$ e m um múltiplo inteiro. Inserindo (3.18) em (3.17) pode-se verificar que são nulas as integrais para $n \neq m$. Resolvendo, obtém-se

$$\phi(t) \approx \frac{I_m c_m \sin(\Delta\omega t)}{2q_{\max} \Delta\omega}. \quad (3.19)$$

O espectro de $\phi(t)$ é ilustrado pela Fig. 3.12 (a) e consiste de dois impulsos simétricos nas frequências $n\omega_0 + \Delta\omega$ [46]. Este resultado é importante para entender a evolução do ruído em um oscilador. Entretanto, a definição de ruído de fase pelo espectro da tensão de saída é mais interessante para ser quantificado, porque pode ser comparado com resultados de medição. Utilizando a modulação em fase de $\phi(t)$ para $v(t)$ vista na Fig. 3.11, dada pela equação de conversão $v_{out}(t) = \cos[\omega_0 t + \phi(t)]$, pode-se determinar o ruído de fase de um sinal de corrente injetado, sendo assim obtém-se [46]

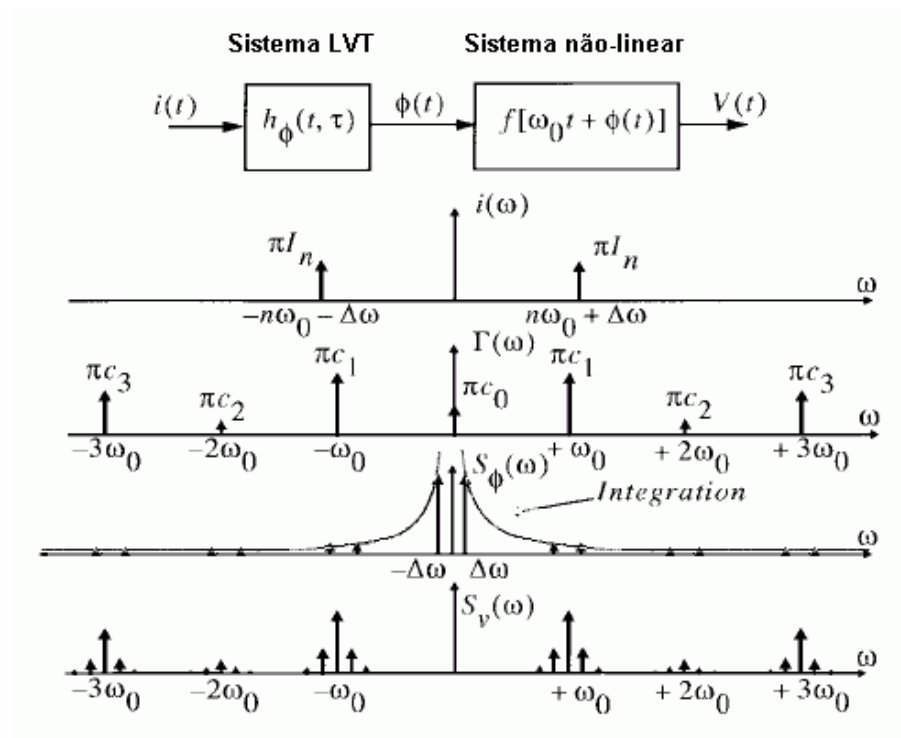


Fig. 3.12. Conversão da função de fase para o espectro do ruído de fase.

$$P_{SBC}(\Delta\omega) = 10 \log \left(\frac{I_m c_m}{4q_{m\acute{a}x} \Delta\omega} \right)^2. \quad (3.20)$$

E esse resultado pode ser estendido para o ruído branco como

$$P_{SBC}(\Delta\omega) = 10 \log \left(\frac{\overline{i_n^2} \sum_{m=0}^{\infty} c_m^2}{4q_{m\acute{a}x}^2 \Delta\omega^2} \right). \quad (3.21)$$

A Fig. 3.13 ilustra que o ruído de fase é composto pelas componentes c_m das faixas de ruído branco em frequências harmônicas de ω_0 . O ruído toma a forma de $1/f^2$ para as componentes c_m do ruído branco e $1/f^3$ para a componente c_0 pela influência do ruído $1/f$.

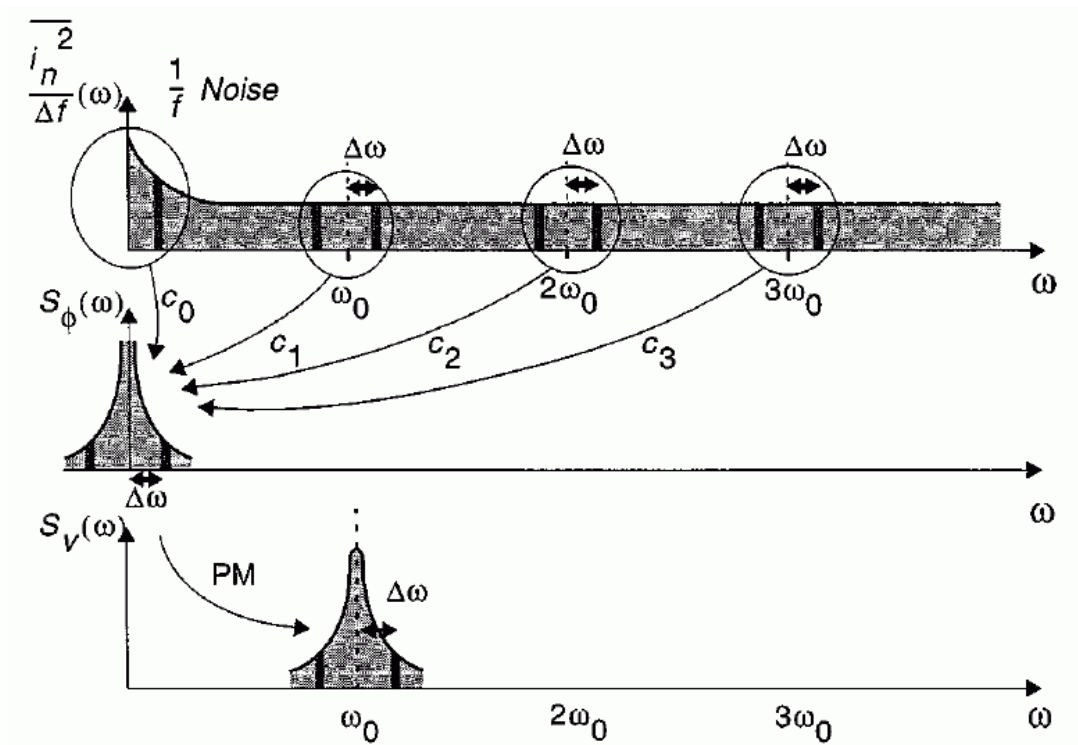


Fig. 3.13. Formato do espectro do ruído de fase proveniente das harmônicas de ω_0 [47].

Torna-se claro que minimizando os componentes (c_m), o ruído de fase será reduzido. Para entender essa conclusão quantitativamente, pode-se utilizar o teorema de Parseval para encontrar o valor rms de ISF, como se mostra em seguida

$$\sum_{n=0}^{\infty} c_m^2 = \frac{1}{\pi} \int_0^{2\pi} |\Gamma(x)|^2 dx = 2\Gamma_{rms}^2 \quad (3.22)$$

e o ruído de fase agora é expresso por

$$P_{SBC}(\Delta\omega) = 10 \log \left(\frac{\overline{i_n^2} \Gamma_{rms}^2}{\Delta f \cdot 2q_{máx}^2 \Delta\omega^2} \right) \quad (3.23)$$

Nesse ponto, podem-se apontar alguns fatores que destacam a importância da análise de ruído variante no tempo. Uma análise importante em sistemas *wireless*, devido à distância entre os canais ser muito estreita, é o modelamento do ruído junto à portadora. O modelo LIT não prediz precisamente o comportamento do ruído na região $1/f$ (ver Fig. 3.13(b) e Fig. 3.13(c)). Portanto, assumindo

$$\overline{i_{n,1/f}^2} = \overline{i_n^2} \frac{\omega_{1/f}}{\Delta\omega} \quad (3.24)$$

sendo $\omega_{1/f}$ a frequência de canto do ruído $1/f$.

Pode-se mostrar que [47]

$$\Delta\omega_{1/f^3} = \omega_{1/f} \frac{c_0^2}{4\Gamma_{rms}^2} = \omega_{1/f} \left(\frac{\Gamma_{dc}}{\Gamma_{rms}} \right)^2 \quad (3.25)$$

sendo $\Delta\omega_{1/f^3}$ a banda de frequência do ruído de fase $1/f^3$ do ruído $1/f$.

Outra análise importante se refere às fontes de ruídos que não são estacionárias, como o ruído branco de dreno. Elas exigem uma modelagem para ruído ciclo-estacionário [45]. Desde que seja considerada uma fonte de ruído, sendo o produto do ruído branco por uma função periódica, então

$$i_n(t) = i_{n0}(t) \cdot \alpha(\omega_0 t). \quad (3.26)$$

e substituindo (3.26) em (3.15) pode-se escrever

$$\Gamma_{eff}(x) = \Gamma(x) \cdot \alpha(x). \quad (3.27)$$

3.1.4. Macromodelo de Oscilador usando Perturbation Projection Vector (PPV)

Osciladores são sistemas não-lineares, visto que um limitador de ganho se faz necessário para manter a amplitude de oscilação. Os dois principais métodos para modelar essas características são ISF e PPV (*Perturbation Projection Vector*). A seguir são dados os conceitos principais de PPV [50].

Pode-se definir um oscilador sendo perturbado, pela seguinte equação:

$$\dot{x} + f(x) = Bb(t) \quad (3.28)$$

sendo $b(t)$ a perturbação aplicada, $x(t)$ um vetor composto pelas variáveis de estado do oscilador e B uma constante. A Fig. 3.14 representa o sistema perturbado. E para pequenas perturbações, pode-se linearizar (3.28) em torno de sua órbita não-perturbada, como

$$\dot{w}(t) \approx - \left. \frac{\partial f(x)}{\partial x} \right|_{x_s(t)} w(t) + Bb(t) = A(t)w(t) + Bb(t) \quad (3.29)$$

sendo $w(t)$ o desvio causado pela perturbação e $x_s(t)$ a solução de regime permanente para o oscilador não-perturbado. A equação (3.29) é um sistema linear periódico e variante no tempo. Esse sistema pode ser resolvido usando a teoria de Floquet, de modo a obter uma matriz de estados, como a seguir [50]

$$\Phi(t, \tau) = U(t) \exp(D(t - \tau)) V(\tau) \quad (3.30)$$

sendo $U(t)$ e $V(t)$, matrizes não-singulares que satisfazem a condição $v_i^T(t) u_j(t) = \delta_{ij}$, sendo $D = \text{diag}[\mu_1, \dots, \mu_n]$, e μ_i os expoentes de Floquet. A equação (3.30) pode ser, então, resolvida por

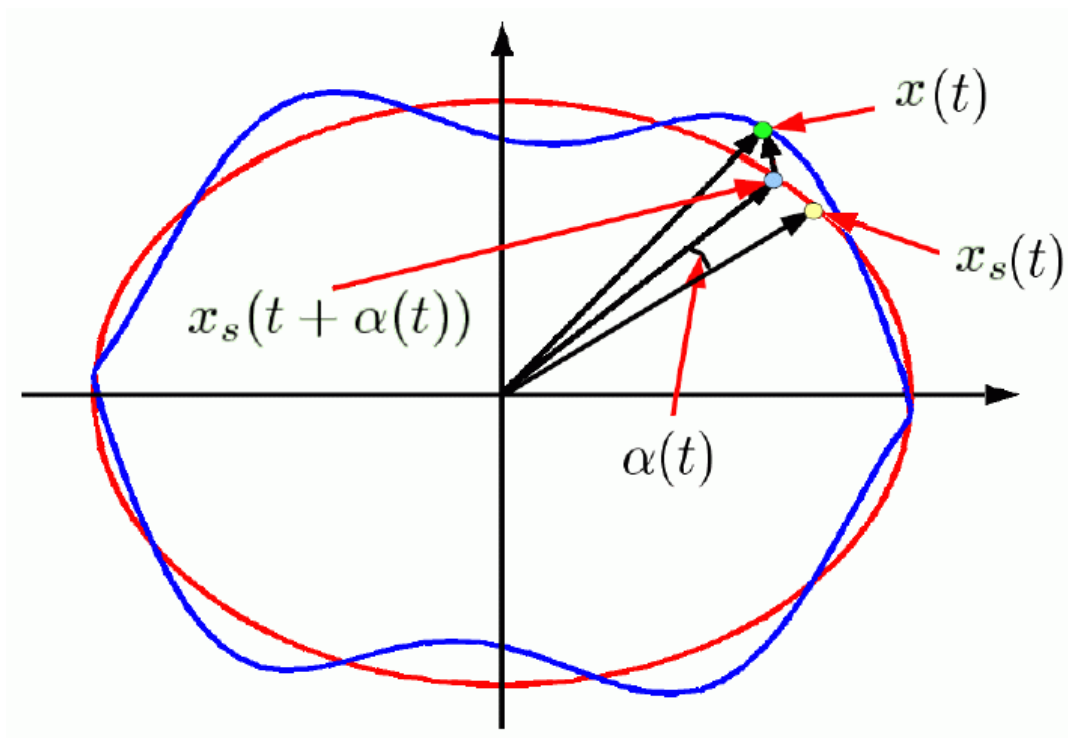


Fig. 3.14. Representação de um oscilador perturbado saindo de sua órbita.

$$w(t) = \sum_{i=1}^n u_i(t) \int_0^1 \exp(\mu_i(t-\tau)) v_i^T(\tau) Bb(\tau) d\tau \quad (3.31)$$

Essa solução não é consistente com a condição de que $w(t)$ seria válido apenas para pequenas perturbações. A solução para isso, pode ser dada dividindo a expressão $Bb(t)$ em duas partes, assim

$$\dot{x} + f(x) = b_1(t) + \bar{b}(t) \quad (3.32)$$

sendo

$$b_1(t) = v_1^T(t + \alpha(t)) Bb(t) u_1(t + \alpha(t)) \quad (3.33)$$

e

$$\bar{b}(t) = \sum_{i=2}^n v_i^T(t + \alpha(t)) Bb(t) u_i(t + \alpha(t)) \quad (3.34)$$

Quando foram caracterizados, foi observado que $b_1(t)$ apenas induziu desvio de fase em $x_s(t)$ e $\bar{b}(t)$ desvio no orbital. De fato, se a solução para a equação $\dot{x} + f(x) = b_1(t)$ for dada por

$$x_p(t) = x_s(t + \alpha(t)), \quad (3.35)$$

então $\alpha(t)$ é o desvio na fase causado pela perturbação $b_1(t)$. Pode ser mostrado que $\alpha(t)$ é dado pela equação a seguir [50]

$$\dot{\alpha}(t) = v_1^T(t + \alpha(t)) \cdot Bb(t) \quad (3.36)$$

O vetor periódico $v_1(t)$ ou PPV é um vetor com período igual à $x_s(t)$ que descreve a sensibilidade da fase do oscilador a perturbações. Então, se o PPV de um oscilador é conhecido, pode-se determinar o seu desvio de fase devido a perturbações, resolvendo-se a equação não-linear (3.36). Como a dimensão de $\alpha(t)$ está em tempo, precisa-se multiplicar o resultado pela frequência de oscilação [51]. Existem alguns métodos disponíveis na literatura para determinar o PPV através do SPICE. Na ferramenta de EDA (*Electronic Design Automation*) da Cadence, esse método é utilizado para gerar um macromodelo para modelar a fase do VCO. O macromodelo é gerado de uma tabela baseada na teoria de PPV apresentada. No processo, simulações de PSS e PNOISE são utilizadas para determinar essa tabela.

A decomposição da equação (3.32) em duas partes permite que o sistema seja resolvido para $\bar{b}(t)$, sem a restrição de pequenas perturbações aplicada a $b_1(t)$. Então, definindo-se o sistema oscilador como

$$\dot{x} + f(x) = Bb(t), \quad y(t) = C^T x(t), \quad (3.37)$$

com solução igual a

$$x(t) = x_p(t) + o(t), \quad (3.38)$$

sendo $o(t)$ o desvio do orbital causado pela perturbação $\bar{b}(t)$, então (3.37) se torna

$$\dot{x}_p(t) + \dot{o}(t) + f(x_p(t) + o(t)) = b_1(t) + \bar{b}(t). \quad (3.39)$$

Linearizando-se (3.39) em torno de $x_p(t)$, a equação de $o(t)$ fica

$$\dot{o}(t) \approx - \frac{\partial f}{\partial x} \Big|_{x_s(t + \alpha(t))} o(t) + \bar{b}(t), \quad (3.40)$$

sendo que $A(t)$ representa o desvio de amplitude causado pela perturbação $\bar{b}(t)$.

Em [50], o sistema foi resolvido para obter um macromodelo da saída do oscilador, com a condição de que $\dot{\alpha}(t) = v_1^T(t) b_1(t) \ll 1$, resolvendo-se a equação para $A(t)$ como

$$\hat{\dot{a}}(t) = \bar{D} \hat{a}(t) + \bar{V}(t) B b(t), \quad A(t) = \hat{A}(t) = C^T \bar{U}(t) \hat{a}(t) \quad (3.41)$$

Então, um macromodelo para a saída do oscilador pode ser encontrado através da equação a seguir

$$y(t) = C^T x_s(t) + \hat{A}(t). \quad (3.42)$$

Capítulo IV

4. Projeto de Sintetizadores de Freqüências Baseados em PLL

Nesse capítulo são abordados os tópicos principais a serem avaliados no projeto de um PLL. Os fundamentos apresentados nesse capítulo são as bases para o desenvolvimento da arquitetura proposta, a qual será apresentada no capítulo 5. As equações dinâmicas do sistema são deduzidas para que possam ser empregadas posteriormente usando controle clássico. O PLL foi analisado em termos das fontes de ruído no sistema e suas influências no ruído de fase total de saída são determinadas.

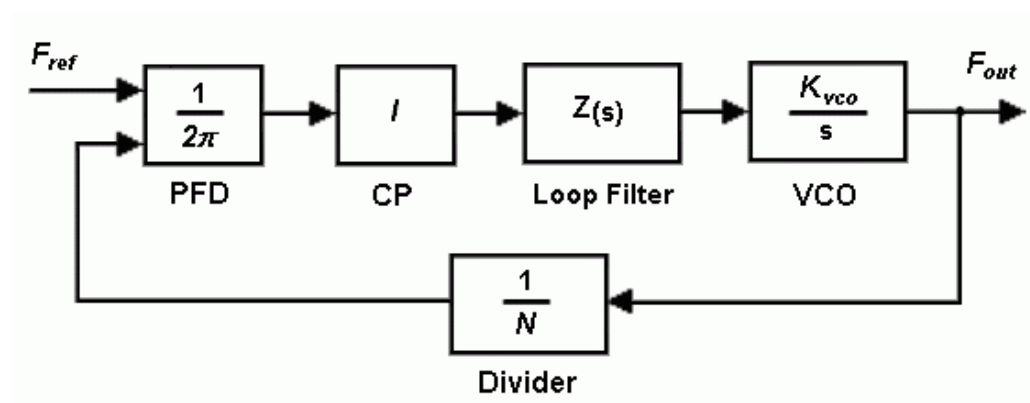
4.1. Tipos de PLL

Os circuitos PLLs podem ser classificados pelo tipo de técnica utilizada no projeto do seu sistema. A nomenclatura segue a ordem cronológica que as topologias foram criadas e por isso deve-se ter cuidado ao ser interpretada. Os tipos principais de PLL estão resumidos na tabela 4.1 [57]. Os PLLs ainda são divididos pela sua dinâmica, mais especificamente o número de pólos na origem do *loop filter*, que são designados: PLL tipo 1, o que não possui pólo na origem e tipo II, o que possui 1 pólo na origem. O PLL abordado nesse capítulo é o PLL mais empregado comumente, classificado por PLL digital por usar um PFD (*Phase Frequency Detector*) digital.

Tabela 4.1 – Tipos de PLL

Tipo de PLL	Detector de Fase	Loop Filter	Oscilador Controlado
PLL linear	Multiplicador analógico	Filtro RC	Tensão
PLL digital	PFD digital	Filtro RC	Tensão
<i>All digital PLL</i>	PFD digital	Filtro digital	Controlado digitalmente
<i>Software PLL</i>	<i>Software PFD</i>	<i>Software filter</i>	<i>Software oscillator</i>

4.2. Dinâmica de Sistemas PLL

Fig. 4.1. Phase-locked loop (PLL) usando *charge-pump*, modelado como um sistema linear representado pelo ganho de cada bloco.

Na Fig. 4.1 é mostrado um PLL tipo II usando *charge-pump* (CP). Quando o PLL é assumido como um sistema linear considera-se que o erro de fase causado pelos blocos PFD e *charge-pump* sejam desprezíveis. Esses blocos são digitais e teriam um modelo mais preciso no domínio-z. Os limites para essa aproximação são estabelecidos na equação 4.1 [26], [52]. No desenvolvimento da arquitetura proposta nessa tese será usado um modelo no domínio-z, devido ao *loop filter digital* proposto.

$$\omega_c < \frac{\omega_{REF}}{\pi \left(1 + \frac{\pi \omega_z}{\omega_{REF}} \right)} \quad (4.1)$$

$$\text{sendo } \omega_c \approx \omega_n^2 / \omega_z \quad (4.2)$$

A seguir é apresentada uma análise do *loop filter* usando componentes passivos. Essa configuração é muito utilizada nas configurações convencionais de PLL (Fig. 4.2). Existem outras configurações de *loop filter* que obtêm maior atenuação na transimpedância em altas frequências (a frequência depende do valor do pólo atribuído) e conseqüentemente maior redução de tons espúrios, porém exigem mais componentes passivos [53].

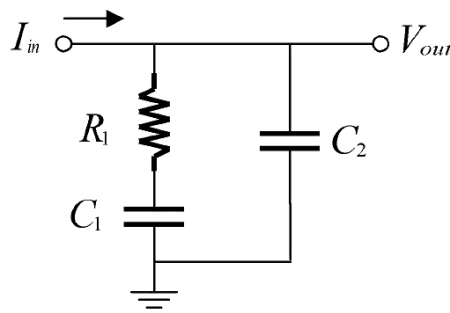


Fig. 4.2. *Loop filter* usando componentes passivos.

A tensão de saída (tensão de controle do VCO) dividida por a corrente vinda do *charge-pump* pode ser deduzida como:

$$Z_{eq} = R_1 + \frac{1}{sC_1} // \frac{1}{sC_2} = \frac{(sR_1C_1 + 1)}{\frac{sR_1C_1 + 1}{sC_1} + \frac{1}{sC_2}}$$

$$Z_{eq} = \frac{(sR_1C_1 + 1)}{(C_1 + C_2) \left(\frac{sR_1C_1C_2}{C_1 + C_2} + 1 \right) s} \quad (4.3)$$

$$\text{sendo } \omega_z = \frac{1}{R_1C_1} \text{ e } \omega_p = \frac{(C_1 + C_2)}{R_1C_1C_2}$$

$$\text{NB: Se } C_1 \gg C_2 \Rightarrow \omega_p = \frac{1}{R_1C_2}$$

ou

$$Z_{eq} = \frac{\left(s + \frac{1}{R_1 C_1}\right)}{C_2 \left(s + \frac{C_1 + C_2}{R_1 C_1 C_2}\right) s}. \quad (4.4)$$

Também se pode deduzir da equação do pólo que:

$$\begin{aligned} C_2(1 - R_1 C_1 \omega_p) &= -C_1 \Rightarrow C_2 = \frac{C_1}{(R_1 C_1 \omega_p - 1)} \\ \Rightarrow \frac{C_1}{C_2} &= R_1 C_1 \omega_p - 1 = \frac{\omega_p}{\omega_z} - 1 \end{aligned} \quad (4.5)$$

A função de transferência de malha aberta do PLL pode ser escrita como

$$H_{open}(s) = \frac{K_D K_{VCO} \left(1 + \frac{s}{\omega_z}\right)}{\left(1 + \frac{s}{\omega_p}\right) s^2}, \quad (4.6)$$

sendo $K_D = \frac{I}{2\pi C_1 N}$

A equação (4.6) é importante para estudar a estabilidade do sistema para os parâmetros de projeto disponíveis. Os parâmetros são, portanto, a corrente do *charge-pump* (I), o ganho do VCO (K_{VCO}), os componentes do *loop filter* e a razão de divisão do *prescaler* (N). Sendo que o K_{VCO} e os componentes do *loop filter* também influem no ruído de fase de saída e N é definido pela arquitetura, então a corrente do *charge-pump* fica sendo um dos poucos parâmetros que podem flexibilizar o projeto.

Análises importantes também podem ser obtidas da função transferência de malha fechada, que pode ser obtida por $H_{closed}(s) = \frac{H_{open}(s)}{1 + H_{open}(s)}$, resultando em:

$$H_{closed}(s) = \frac{1 + \frac{s}{\omega_z}}{1 + \frac{s}{\omega_z} + \frac{s^2}{K_D K_{VCO}} + \frac{s^3}{\omega_p K_D K_{VCO}}} \quad (4.7)$$

Se for assumido que $\omega_p \gg \omega_n$, sendo $\omega_n^2 = K_D K_{VCO}$, então o sistema passa a ser de segunda ordem e as aproximações clássicas podem ser aplicadas. A equação da função de transferência de malha fechada se torna:

$$H_{closed}(s) \approx \frac{1 + \frac{s}{\omega_z}}{1 + \frac{s}{\omega_z} + \frac{s^2}{K_D K_{VCO}}} \quad (4.8)$$

O tempo de estabelecimento (*settling time*) pode ser deduzido a partir de (4.8), através da resposta ao degrau do sistema. A seguir são mostradas as equações do tempo de estabelecimento para as três condições do transitório, dadas pelo amortecimento ζ [53].

$$\left\{ \begin{array}{l} t_s = \frac{1}{\zeta \omega_n} \ln \frac{\Delta f}{\alpha f_0 \sqrt{1 - \zeta^2}} \quad \text{para } \zeta < 1 \\ t_s = \frac{1}{\zeta \omega_n} \ln \frac{\Delta f}{\alpha f_0} \quad \text{para } \zeta = 1 \\ t_s = \frac{1}{(\zeta - \sqrt{\zeta^2 - 1}) \omega_n} \ln \frac{\Delta f (\sqrt{\zeta^2 - 1} + \zeta)}{2 \alpha f_0 \sqrt{\zeta^2 - 1}} \quad \text{para } \zeta > 1 \end{array} \right. \quad (4.9)$$

sendo f_0 a frequência de início do transitório, Δf a variação de frequência do canal, que no caso de t_s é o *tuning range*, α a exatidão desejada e o amortecimento é dado por

$$\zeta = \frac{\omega_n}{2\omega_z}. \quad (4.10)$$

A margem de fase do sistema é [53]

$$\phi_m = \text{tag}^{-1}\left(\frac{\omega_c}{\omega_z}\right) - \text{tag}^{-1}\left(\frac{\omega_c}{\omega_p}\right) \quad (4.11)$$

Nota-se, a partir das equações (4.1) e (4.9) que quanto maior a largura de banda ω_c menor é o tempo de estabelecimento. Já o amortecimento, só influencia expressivamente o aumento do tempo de estabelecimento quando super-amortecido e permanecendo quase constante quando sub-amortecido.

Uma comparação das três condições é ilustrada na Fig. 4.3. Observa-se que o estado sub-amortecido (*underdamped*) causa resultados indesejados de *overshoot* e na resposta em frequência, conclui-se que $\zeta = 1$ é o valor mais apropriado para se usar em projetos.

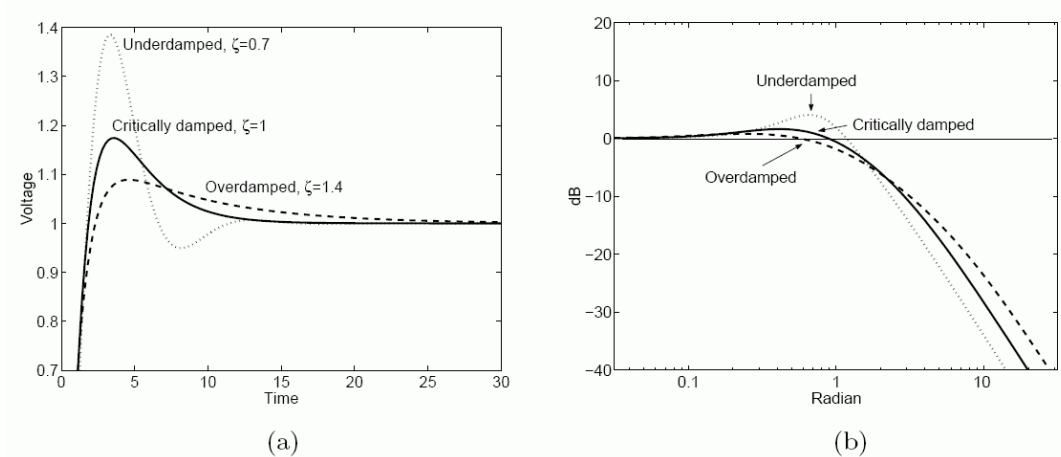


Fig. 4.3. Análise do comportamento do sistema com a variação do amortecimento (a) no domínio do tempo (b) no domínio da frequência.

4.3. Análise de Ruído de Fase e Espúrios em Sistemas PLL

A análise a seguir tem o objetivo de representar a influência das fontes de ruído de fase do PLL no ruído de fase total de saída [54], [2].

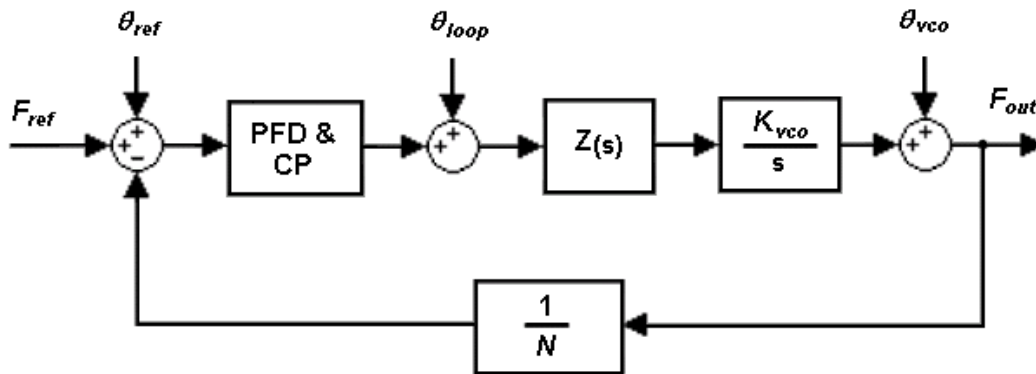


Fig. 4.4. Modelo simplificado das fontes de ruído de fase no sistema PLL.

Na Fig. 4.4 estão apresentadas as fontes de ruído. Para simplificar a análise define-se $G(s)$ como o ganho sem realimentação e $O(s)$ o ganho em malha aberta, como dado abaixo:

$$G(s) = \frac{K_D K_{VCO} Z_{eq}(s)}{s} \quad (4.12)$$

$$O(s) = \frac{K_D K_{VCO} Z_{eq}(s)}{Ns} \quad (4.13)$$

Sendo θ_o o ruído de fase de saída, podem-se deduzir as equações para a função de transferência das fontes de ruído, como a seguir:

$$\frac{\theta_o}{\theta_{ref}} = \frac{\frac{K_D K_{VCO} Z_{eq}(s)}{s}}{1 + \frac{K_D K_{VCO} Z_{eq}(s)}{Ns}} = N \frac{O(s)}{1 + O(s)} \quad (4.14)$$

$$\frac{\theta_o}{\theta_{loop}} = \frac{\frac{K_{VCO} Z_{eq}(s)}{s}}{1 + \frac{K_D K_{VCO} Z_{eq}(s)}{Ns}} = \frac{N}{K_D} \frac{O(s)}{1 + O(s)} \quad (4.15)$$

$$\frac{\theta_o}{\theta_{vco}} = \frac{1}{1 + \frac{K_D K_{VCO} Z_{eq}(s)}{Ns}} = \frac{1}{1 + O(s)} \quad (4.16)$$

Todas as fontes de ruído, exceto o ruído de fase do VCO, são definidas como ruído dentro da banda (*in-band noise*), e têm semelhanças na resposta em frequência, como se pode notar nas equações (4.14) e (4.15). Sendo $H(s) = 1 / N$ e lembrando que ω_c é a frequência de ganho unitário do sistema, pode-se representar a função de transferência do ruído de fase do sistema como ilustrado nas Fig. 4.5 e Fig. 4.6.

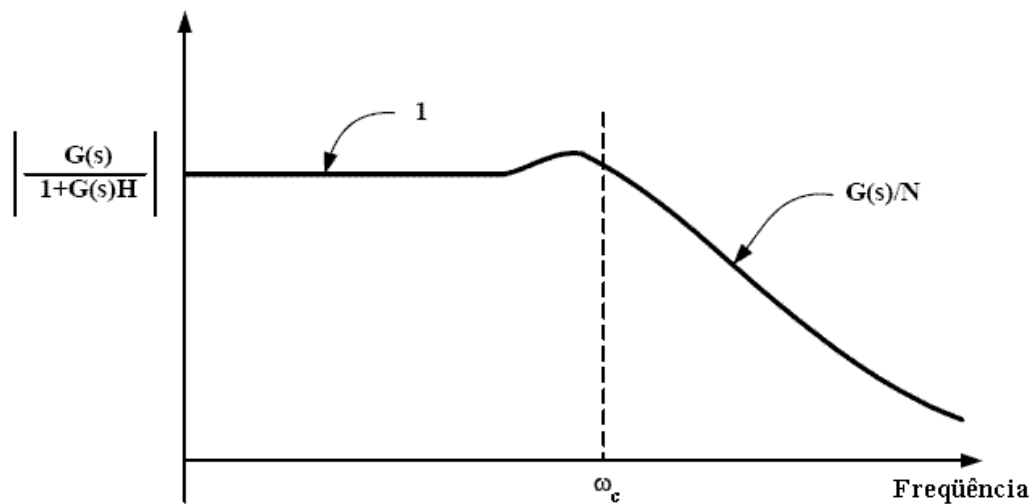


Fig. 4.5. Função de transferência típica para o ruído dentro da banda (*in-band noise*) [2].

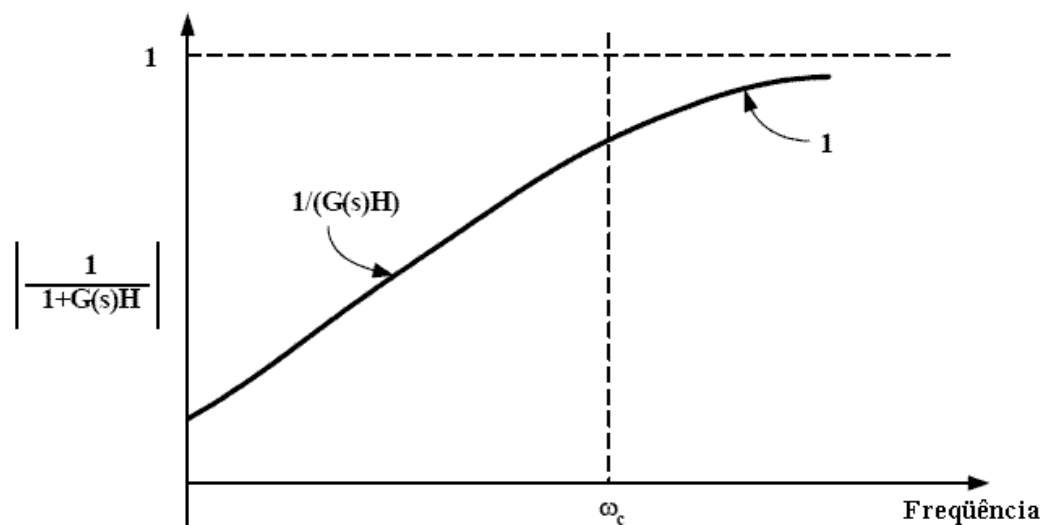


Fig. 4.6. Função de transferência típica para o ruído do VCO [2].

Por fim, para resumir as influências dos ruídos mais importantes e definir um ponto ótimo para a largura de banda do *loop filter*, as fontes de ruído foram representadas na figura 4.7 em relação à saída do sistema. O ponto ótimo definido é o cruzamento do gráfico do ruído de fase do VCO para o do ruído da frequência de referência referida à saída, sendo que F_{ref} / R é a menor frequência encontrada na entrada.

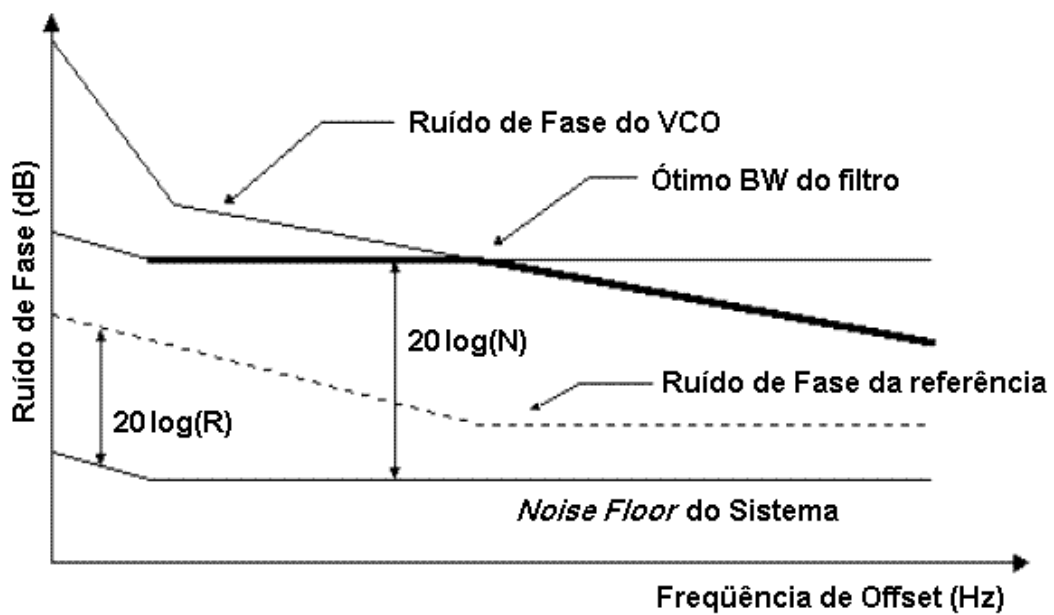


Fig. 4.7. Determinação do ponto ótimo para a largura de banda do PLL em termos de ruído de fase de saída [2].

Capítulo V

5. Sintetizador de Freqüência com *Dual-Path* e Técnicas Híbridas Analógicas e Digitais Aplicadas ao *Loop-Filter*

Nesse capítulo é descrita a arquitetura do PLL proposto e é apresentado o desenvolvimento teórico da nova topologia. A estabilidade do sistema é analisada para definir a topologia e para projetar os parâmetros dos blocos do PLL. Será visto que a aplicação de controle digital viabiliza uma metodologia de projeto para a arquitetura proposta.

O acumulador utilizado e os demais blocos digitais empregados no projeto apresentam baixa complexidade. O sintetizador desenvolvido nesse trabalho é proposto para ter custo baixo e pouca complexidade.

5.1. Descrição da Arquitetura

A arquitetura de PLL proposta utiliza técnicas de *dual-path* para realizar um PLL cuja tensão de controle do VCO seja mista, composta de uma tensão de controle analógica e um barramento digital. O laço digital não possui *charge-pump*, e em seu lugar, a saída do PFD (*Phase Frequency Detector*) digital é amostrada e injetada em um contador crescente/decrescente. Neste contador digital são aplicadas todas as técnicas necessárias para o funcionamento da topologia e para aumentar a seu desempenho em relação às convencionais. O barramento pode ser aplicado ao VCO sem a necessidade de um conversor D/A propriamente dito, como no caso dos osciladores digitalmente controlados (DCO) [55]. A Fig. 5.1 ilustra a topologia proposta.

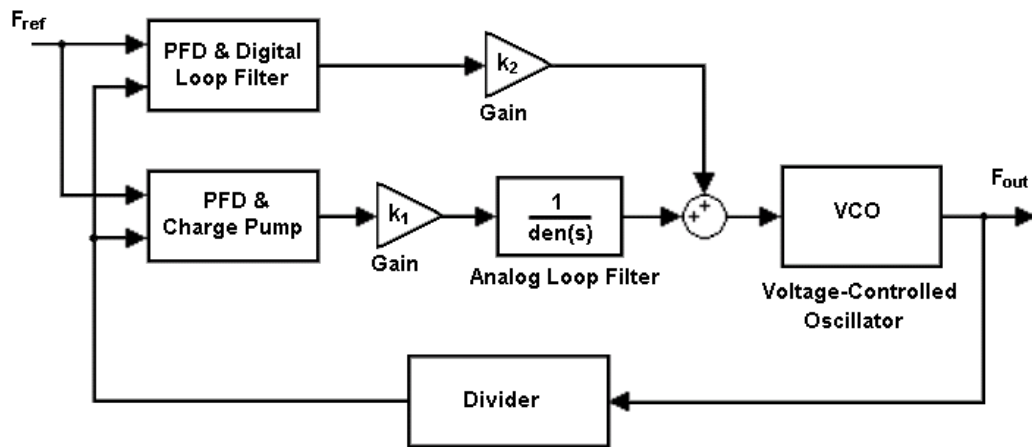


Fig. 5.1. Arquitetura do PLL proposto; laço analógico: PFD digital, *charge-pump* e componentes RC; laço digital: PFD digital, amostragem da diferença de fase, contador crescente / decrescente e conversor D/A.

5.2. Dual Path Loop Filter

A topologia representada na Fig. 5.2 é utilizada para resolver problemas em projetos que precisam de capacitores grandes no *loop filter* [6].

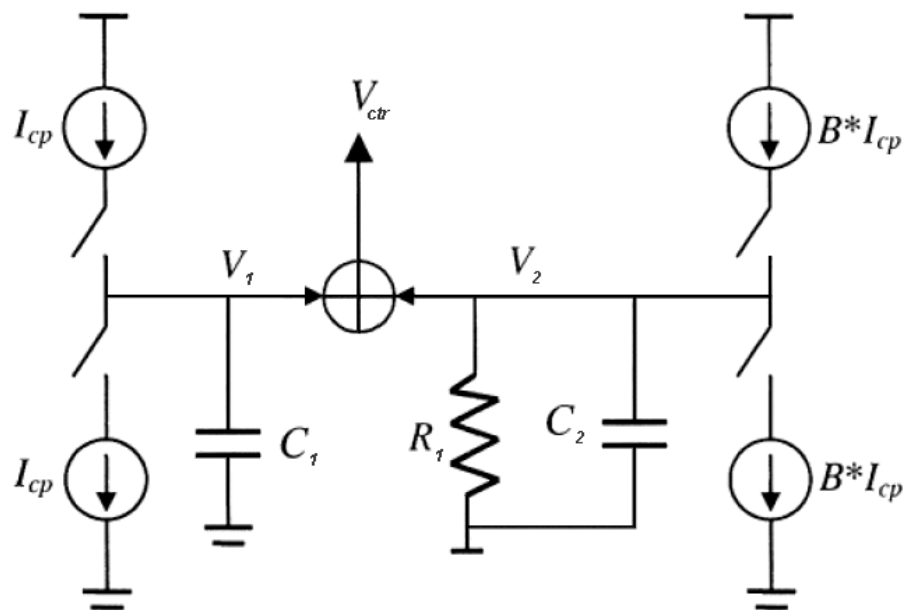


Fig. 5.2. Operação do *dual-path loop filter* usando componentes passivos [6].

O circuito pode ser equacionado como a seguir:

$$V_{ctr} = V_1 + V_2 = \frac{I_{cp}}{sC_1} + \frac{BI_{cp}R_1}{1 + sR_1C_2} = \frac{I_{cp} [1 + sR_1(B C_1 + C_2)]}{sC_1(1 + sR_1C_2)} \quad (5.1)$$

Verifica-se que o zero dessa topologia é dado pela equação abaixo:

$$\omega_z = \frac{1}{R_1(B C_1 + C_2)} \quad (5.2)$$

Sendo que o pólo em malha aberta é o mesmo que no circuito de *loop filter* convencional (ver capítulo 4), o escalonamento da corrente do *charge-pump* é equivalente a reduzir o capacitor C_1 na mesma proporção, como se verifica na equação (5.2).

Apesar de se obter uma grande vantagem nessa redução, muitas desvantagens são ocasionadas. O escalonamento da corrente resulta em corrente baixa no laço de integração, o que aumenta proporcionalmente o ruído. Outro problema é o aumento da ondulação (*ripple*) na tensão de controle, causado pela queda na tensão do laço RC. Um dos objetivos de se usar técnicas digitais no laço de integração é resolver esses problemas.

5.2.1. Estudo da Estabilidade

Algumas topologias foram avaliadas, em termos de estabilidade, antes de se definir a topologia atual para o uso das técnicas de *dual-path*. Determinaram-se três soluções, que foram analisadas em tempo contínuo. Deve ser considerado que o integrador analógico $\frac{K_2}{s}$ será substituído por um integrador digital no decorrer do desenvolvimento da arquitetura.

Soluções para o *Loop Filter*

- Solução 1

$$\frac{K_2}{s} + K_1 \frac{(s+a)}{s} = K_1 \frac{\left(s + \frac{K_2+a}{K_1} \right)}{s}$$

$\varpi_z = \frac{K_2+a}{K_1}$, PLL de 2ª ordem; rápido, mas não adequado para supressão de tons espúrios.

- Solução 2

$$\frac{K_2}{s} + \frac{K_1}{(s+a)} = (K_1 + K_2) \frac{\left(s + \frac{K_2 a}{K_1 + K_2} \right)}{s(s+b)}$$

$\varpi_z = \frac{K_2 a}{K_1 + K_2}$, um PLL de 3ª ordem com um pólo extra para supressão de tons espúrios.

- Solução 3

$$\frac{K_2}{s} + K_1 \frac{(s+a)}{(s+b)} = \frac{K_1 s^2 + (K_1 a + K_2) s + K_2 b}{s(s+b)}$$

um PLL de 3ª ordem com um zero extra; está entre PLL de 2ª e 3ª ordem.

O PLL foi simulado no domínio da fase para um ω_c de aproximadamente 1 rad/s. Alguns programas de MATLAB são usados para projetar o loop antes das simulações no Simulink (ver apêndices A1 e A2). Fig. 5.3 e Fig. 5.4 mostram o lugar das raízes, margem de fase, ω_c e o diagrama de Bode para a solução 1. A área pontilhada representa $t_s = 1s$ e um *overshoot* de 5 %. As simulações no Simulink são apresentadas nas Fig. 5.5 e Fig. 5.6.

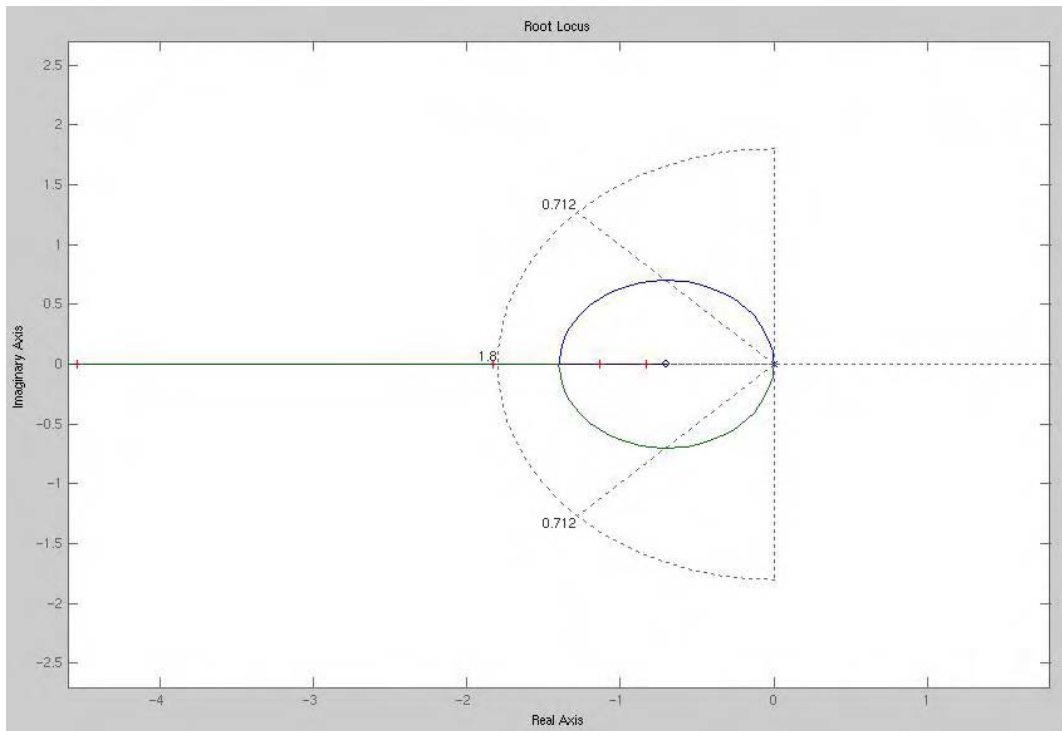


Fig. 5.3. Lugar das raízes da solução 1 para 2 pólos na origem e um zero alocado em 0.7 rad/s.

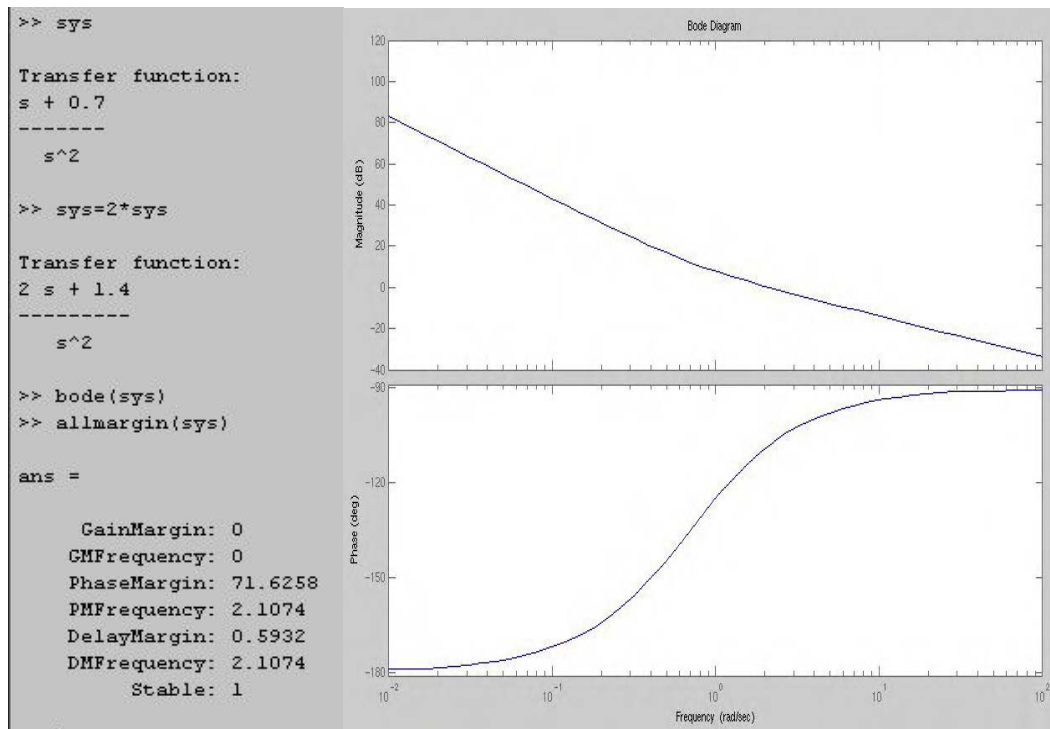


Fig. 5.4. Resultados no MATLAB para solução 1 em malha aberta, margem de fase e diagrama de Bode.

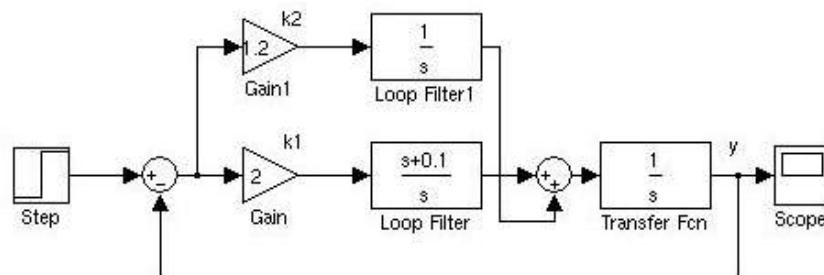


Fig. 5.5. Esquemático no Simulink da solução 1 para $b = 0.1$.

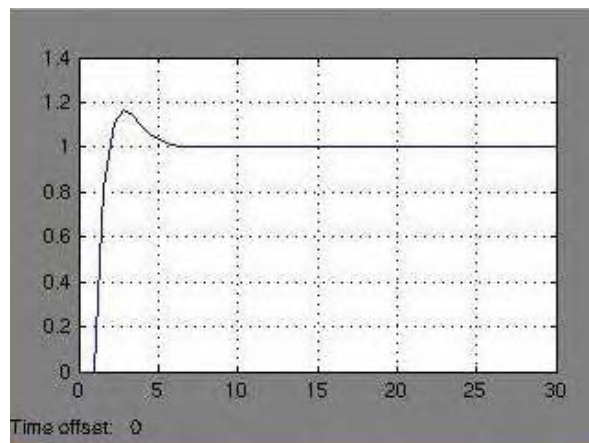


Fig. 5.6. Resposta ao degrau unitário.

As mesmas análises foram realizadas para solução 2 e estão ilustradas nas Fig. 5.7 à Fig. 5.10. Nessas análises foram considerados a margem de fase, resposta em frequência, sendo que houve maior ênfase na frequência de ganho unitário e no sobre-sinal da resposta em malha fechada. Após realizar a análise de estabilidade para as três soluções conclui-se que a solução 2, além de ser mais adequada para supressão de tons espúrios, também proporciona mais escalonamento dos dispositivos passivos quando for aplicada a técnica digital ao laço integrador.

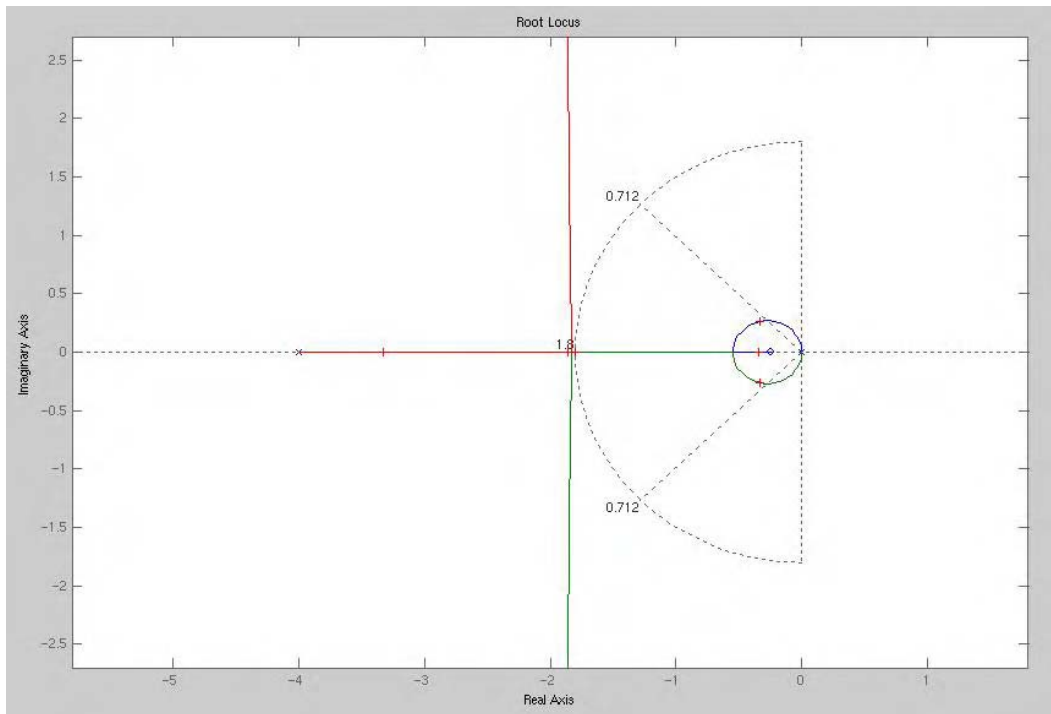


Fig. 5.7. Lugar das raízes da solução 2 para 1 pólo na origem, um pólo em -4 rad/s e um zero alocado em $-0,25$ rad/s.

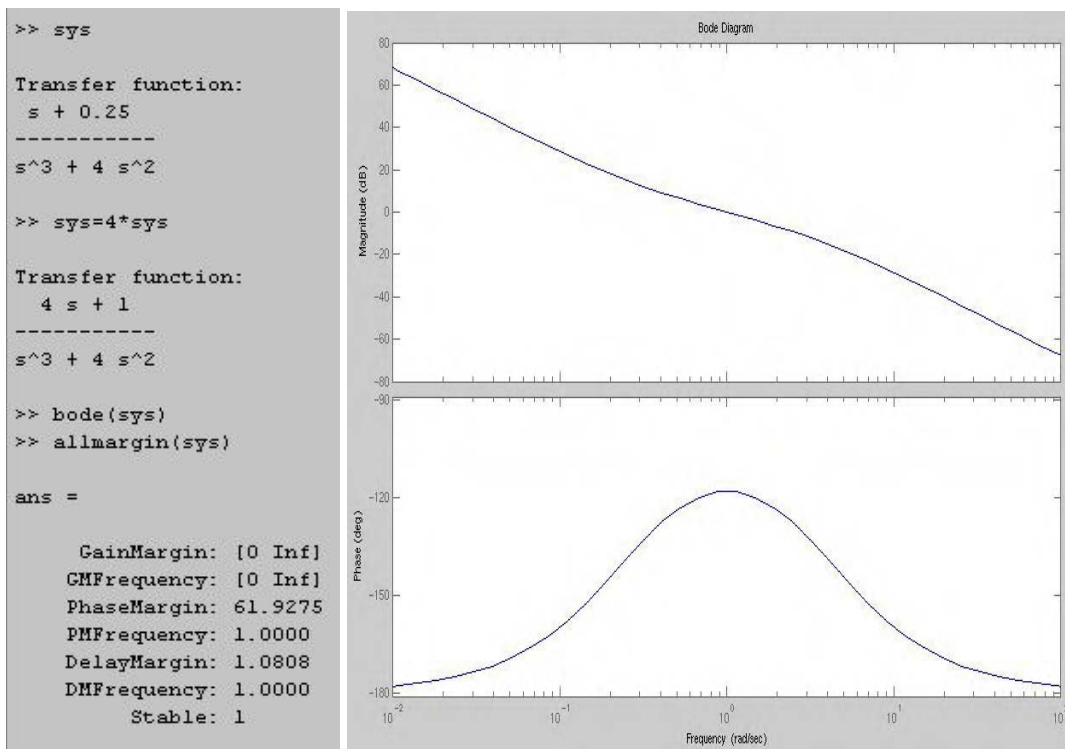


Fig. 5.8. Resultados no MATLAB para solução 2 em malha aberta, margem de fase e diagrama de Bode.

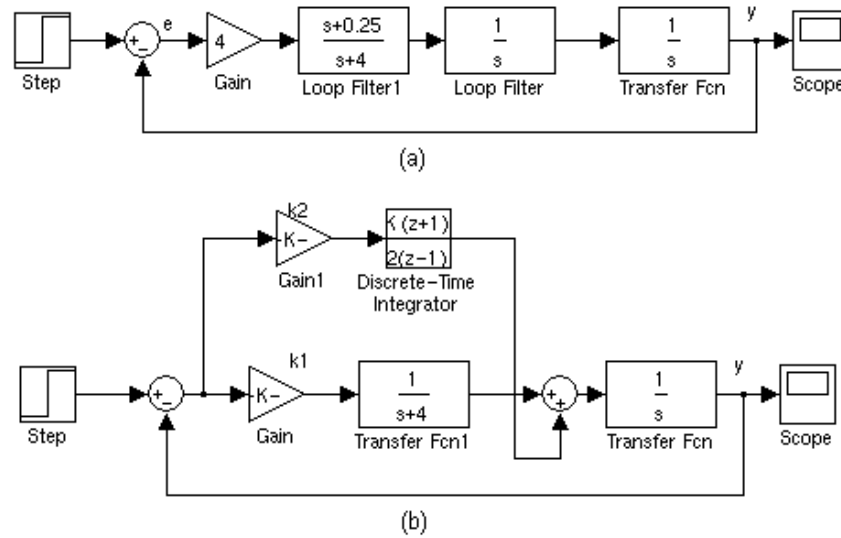


Fig. 5.9. Esquemático no Simulink da solução 2 para $a = 0,25$ e $b = 4$ (a) com componentes passivos (b) usando um laço digital.

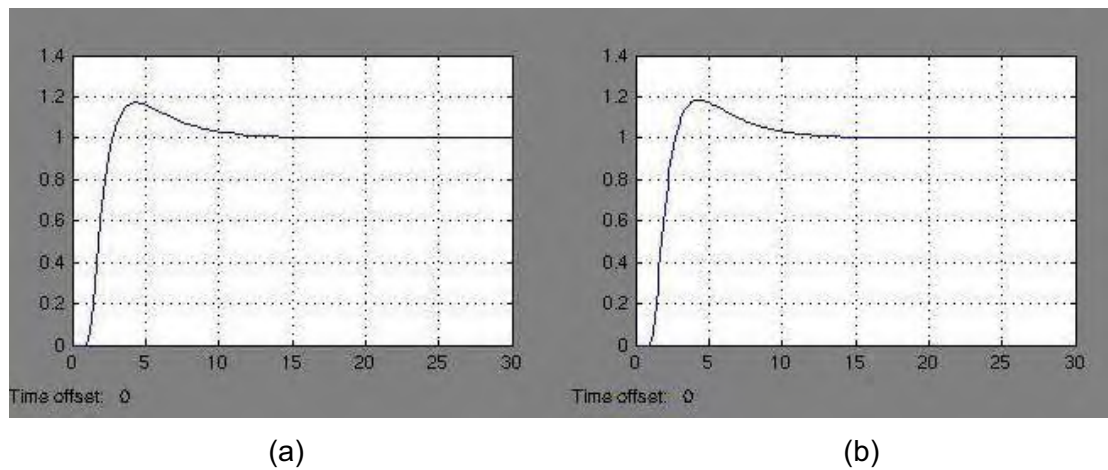


Fig. 5.10. Resposta ao degrau para o *loop filter* (a) contínuo (b) usando um laço discreto com período de amostragem de 0.1 s.

5.3. Blocos Usados nas Simulações

Apesar da modelagem que foi empregada no detector de fase (PFD) não representar precisamente os atrasos e não-linearidades dos dispositivos os resultados de simulação de transitório obtidos dessa forma permitem uma análise teórica suficiente para prever o comportamento do sistema em regime transiente. Os blocos que foram usados no Simulink estão apresentados abaixo.

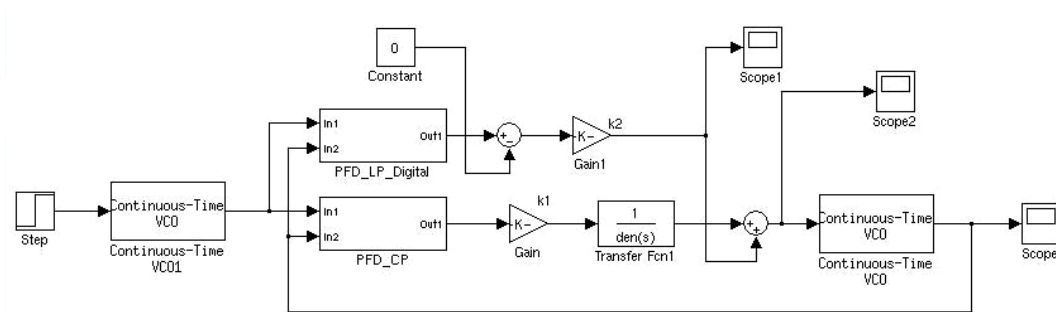


Fig. 5.11. O PLL simulado no Simulink usando um laço digital $TK_2/(z-1)$, como na solução 2.

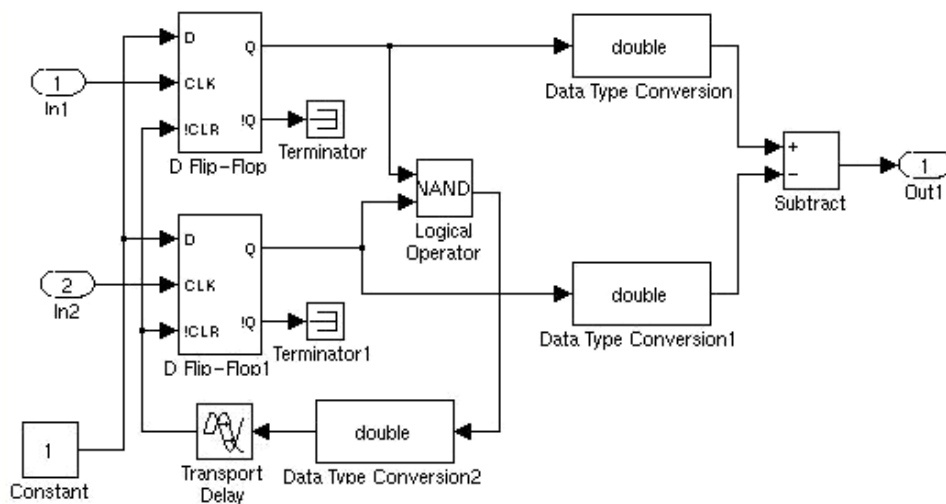


Fig. 5.12. PFD usado no laço analógico.

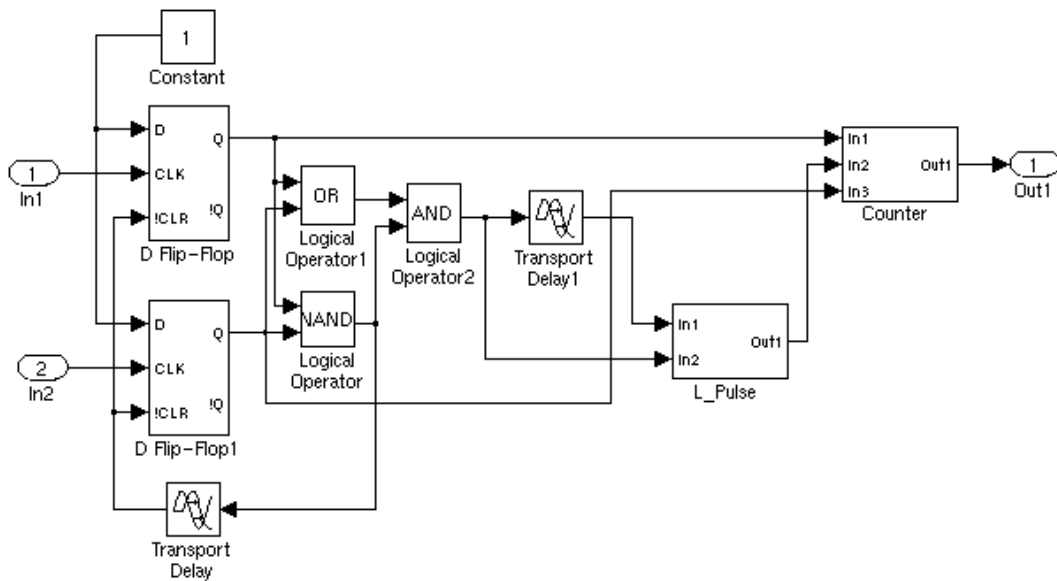


Fig. 5.13. PFD digital e um contador crescente/decrecente utilizado como *loop filter* digital (Delay1 = T).

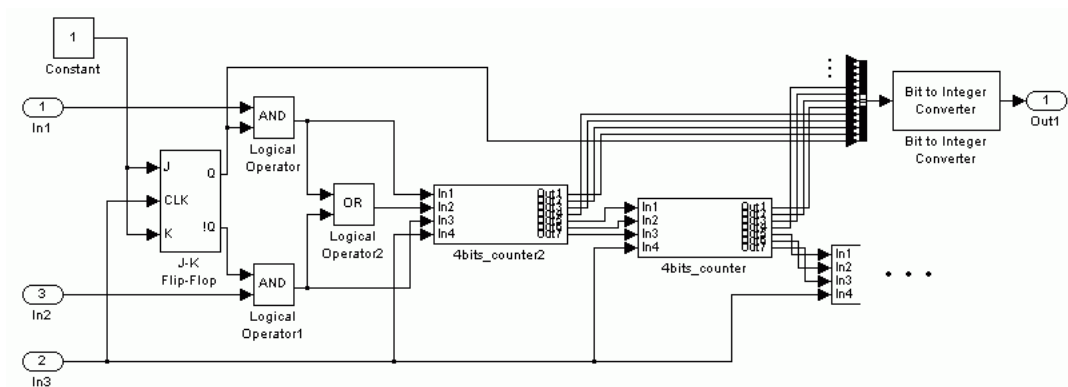


Fig. 5.14. O acumulador discreto empregando um contador crescente/decrecente e um conversor D/A. O número de bits do contador depende do projeto do *loop filter*.

5.4. Projeto do *Loop Filter* Híbrido usando Emulação

Como o projeto do *loop filter* não utiliza técnicas de controle digital, mas sim, a emulação de um integrador analógico, o projeto do PFD utiliza a amostragem da defasagem de entrada em uma frequência algumas vezes maior que a de entrada. Caso fosse usado algum tipo de algoritmo no processamento digital do *loop filter*,

não haveria a necessidade da frequência de amostragem ser alta. Todavia, o objetivo nessa arquitetura é de reduzir a área e a complexidade do projeto, obtendo uma metodologia de projeto usando um circuito relativamente simples. O circuito da Fig. 5.15 é um exemplo de circuito que gera pulsos na entrada do contador. Quanto maior a frequência de amostragem, maior deverá ser o número de bits do contador para representar um mesmo ganho.

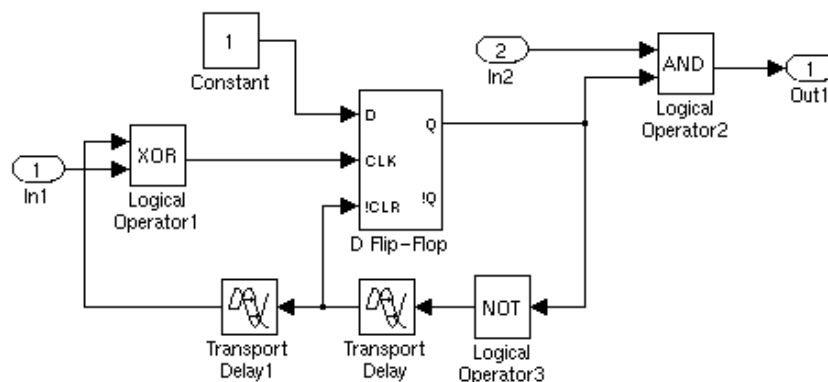


Fig. 5.15. Bloco que gera pulsos para amostrar a diferença de fase.

O número de bits do contador pode ser encontrado através da equação do valor máximo do contador, como a seguir.

$$Co_{m\acute{a}x} = \text{tuning range} / (K_{VCO} \times K_{dig}), \quad (5.3)$$

sendo $Co_{m\acute{a}x}$ o valor máximo da saída do contador e K_{dig} é o ganho do laço digital.

Por exemplo, se $\Delta f = 5.1$ MHz, $K'_2 = 0.25/4$, $T = 7.5 \times 10^{-10}$, $N = 1$ e $\omega_n = 2\pi \times 25$ kHz, o $Co_{m\acute{a}x}$ é 93572.4519, o que significa 17 bits.

5.4.1. Amostragem da Diferença de Fase

Foi verificado, que na amostragem da diferença de fase, o primeiro pulso só deveria ser tomado após um período de amostragem completo (Fig. 5.16), deste modo, somando-se o valor correspondente da defasagem amostrada.

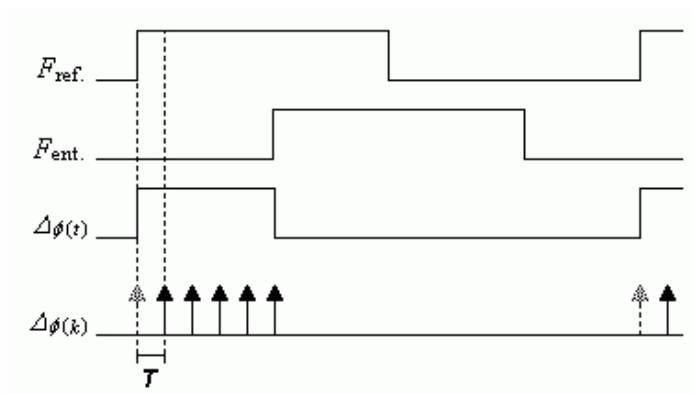


Fig. 5.16. Modo utilizado na amostragem da diferença de fase.

Além disso, quando esse critério não é considerado, e um pulso é gerado no momento em que a defasagem é detectada, um dado errado pode ser tomado. O circuito gerador de pulsos corrigido é mostrado na Fig. 5.17. Essa modelagem demonstrou uma redução na ondulação do sinal de controle do VCO quando os dois modelos (Fig. 5.15 e Fig. 5.17) foram comparados em simulações. No capítulo VI verifica-se que efeitos conhecidos, para o PFD e *charge-pump*, podem ser correlacionados ao *loop filter* digital pelos atrasos usados nas células propostas.

Verifica-se que quando a defasagem é menor que T , a precisão na detecção começa a cair proporcionalmente. Isso provoca efeitos indesejáveis, como *ripple* e tons espúrios. Dessa característica poderia derivar uma técnica onde se aumenta a definição do acumulador (integrador digital usando o contador crescente/decrescente) gradativamente no momento em que o PLL está quase

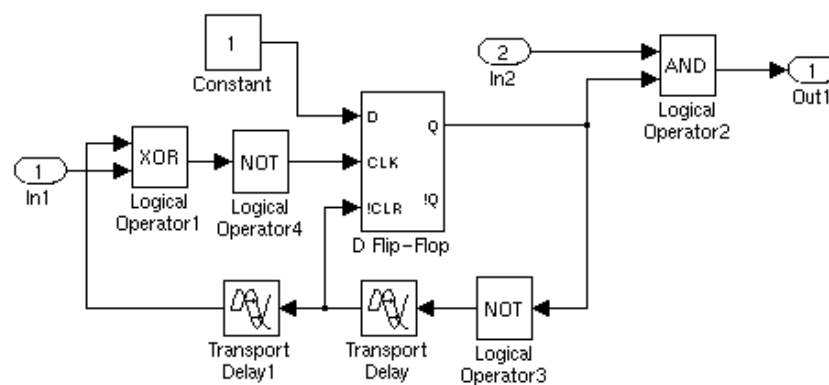


Fig. 5.17. Bloco que gera pulsos para amostrar a diferença de fase modificado

sincronizando com o objetivo de reduzir o erro de fase até a precisão necessária, porém isso aumentaria a complexidade e área do projeto. Essa possibilidade foi levantada no início dos estudos, mas foi descartada, pois exigiria um algoritmo para controlar automaticamente o sistema.

A função de transferência do laço digital pode ser deduzida pelos passos a seguir:

A equação de diferenças do contador pode ser escrita como

$$y[K] = x[K] + y[K - 1] \quad (5.4)$$

e a transformada Z dessa equação pode ser usada para determinar a sua função de transferência e, portanto

$$Y(z) = X(z) + Y(z)z^{-1} \Rightarrow Y(z)(1 - z^{-1}) = X(z)$$

$$\frac{Y(z)}{X(z)} = \frac{z}{(z - 1)} \quad (5.5)$$

Definindo-se $\Delta\phi = \phi_0 u[K]$ e 2π como um período da frequência de referência, a tensão correspondente a defasagem em função do tempo é dada como a seguir.

$$V_{out}[KT] = K_{dig} \frac{\phi_0}{2\pi} KT u[K] \quad (5.6)$$

e a resposta ao impulso do sistema e sua função de transferência ficam

$$h[KT] = K_{dig} \frac{T}{2\pi} u[K] \quad (5.7)$$

$$\frac{V_{out}}{\Delta\phi_0}(z) = H(z) = \frac{K_{dig} T}{2\pi} \frac{z}{(z - 1)} \quad (5.8)$$

Usualmente, o projeto do controlador digital usando emulação é obtido através de três etapas [58]: primeiro se projeta o controlador no domínio contínuo

usando o plano-s, o projeto usando controle linear é mais conhecido e tradicionalmente usado pelos projetistas, sendo esse o motivo do uso da emulação. Depois, usando a transformada bilinear, dada por $z = e^{sT}$, transfere-se os pólos e zeros do controlador para o plano-z discreto. Em uma última etapa, o ganho do controlador discreto é determinado, concluindo-se o processo. Devido ao fato de que nesse método os conversores ADC e DAC são desconsiderados, o processo só atinge precisão se T for suficientemente pequeno para que não ocorra discrepância entre o controlador contínuo e o discreto.

No projeto do PLL proposto, utiliza-se um integrador contínuo para emular o integrador da equação (5.5) e depois o ganho encontrado, usando controle linear para atingir as especificações do sistema, deve ser passado para o acumulador digital.

5.4.2. Equacionamento do *Loop Filter* Híbrido Analógico e Digital

O *loop filter* analógico+digital foi equacionado, considerando a aproximação para um sistema contínuo [52]. A equação que relaciona os laços analógico e digital pode ser deduzida como abaixo.

Se a parte analógica do loop filter for definida como mostrado na Fig. 5.18, sua função de transferência será

$$Z_{eq} = \frac{1}{C_2 \left(s + \frac{1}{R_1 C_2} \right)}. \quad (5.9)$$

O acumulador pode ser aproximado como $\frac{TK_2}{(z-1)} \Leftrightarrow \frac{K_2}{s}$ (considerando que um conversor D/A foi empregado). Fazendo $\frac{K_2}{s} = \frac{K'_2}{sC_2}$ para relacionar os laços analógico e digital na equação, o loop filter analógico + digital pode ser deduzido como

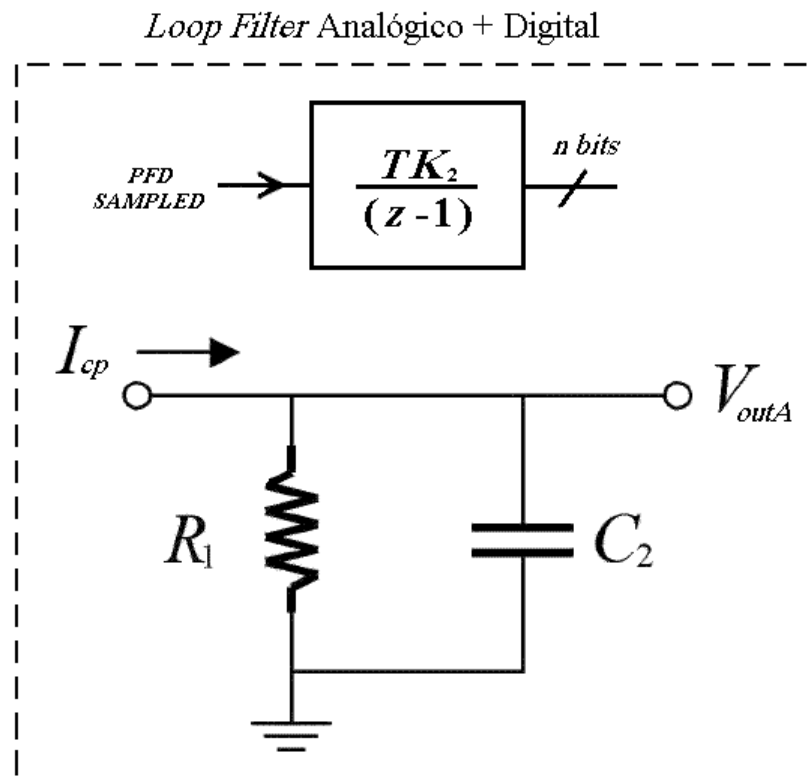


Fig. 5.18. Representação dos laços que compõem a tensão de controle do VCO.

$$\frac{K'_1}{C_2 \left(s + \frac{1}{R_1 C_2} \right)} + \frac{K'_2}{s C_2} = (K'_1 + K'_2) \frac{\left(s + \frac{K'_2}{R_1 C_2 (K'_1 + K'_2)} \right)}{s C_2 \left(s + \frac{1}{R_1 C_2} \right)} \quad (5.10)$$

Se $C_1/C_2 = K'_2$ e $(K'_1 + K'_2) = 1$ a equação (5.10) se torna exatamente como a deduzida para o *loop filter* convencional do PLL de 3ª ordem, deduzida no capítulo 4 e repetida abaixo

$$Z_{eq} = \frac{\left(s + \frac{1}{R_1 C_1} \right)}{s C_2 \left(s + \frac{1}{R_1 C_2} \right)}. \quad (4.4)$$

Comparando as duas equações, obtém-se: $\frac{K_2}{s} \approx \frac{1}{sC_1}$. Esse resultado significa que o capacitor C_1 foi trocado por um acumulador digital.

Deve-se observar que essa conclusão só é válida para $\zeta = 1$ se, e somente se, $K'_1 = 3,75/4$ e $K'_2 = 0,25/4$, pois $C_2/C_1 = K'_2 = 1/16$. Entretanto, esse resultado foi baseado em equações aproximadas, e, a rigor, existem outros fatores a se considerar no cálculo.

Conclui-se que o valor do ganho do laço digital pode ser encontrado a partir de (4.6), como:

$$K_{dig} = 2\pi NK'_2 T \frac{C_1}{C_2} \frac{\sigma_n^2}{K_{VCO}} = \frac{K'_2 T I_{cp}}{C_2} = \frac{T I_{cp}}{C_1} \quad (5.4)$$

Essa equação leva em conta que um sistema superamortecido tenha sido usado.

Grosseiramente, o aumento do capacitor em um *loop filter* analógico, por exemplo C_1 , é proporcional ao aumento da área, enquanto que o *dual-path loop filter* necessita adicionar 1 bit/flip-flop para cada divisão por 2, ou seja, a relação entre o aumento de flip-flops e a divisão é \log_2 (relação de divisão). Assim, a topologia apresenta uma vantagem quando a supressão de espúrios é requerida ou o aumento do *tuning range*.

5.4.3. Técnica para Redução de Tons Espúrios

No PLL proposto foi empregada uma técnica para realizar a supressão de tons espúrios ao mesmo tempo em que se minimiza o tempo de estabelecimento. Nessa técnica, o *loop filter* digital possui no mínimo dois estados de operação que determinam a largura de banda do *loop filter* (ω_b). Quando o PLL está quase sincronizado, a banda é reduzida para obter supressão de espúrios e quando a frequência é comutada, a banda passa a ser mais larga para que o tempo de estabelecimento seja mais rápido.

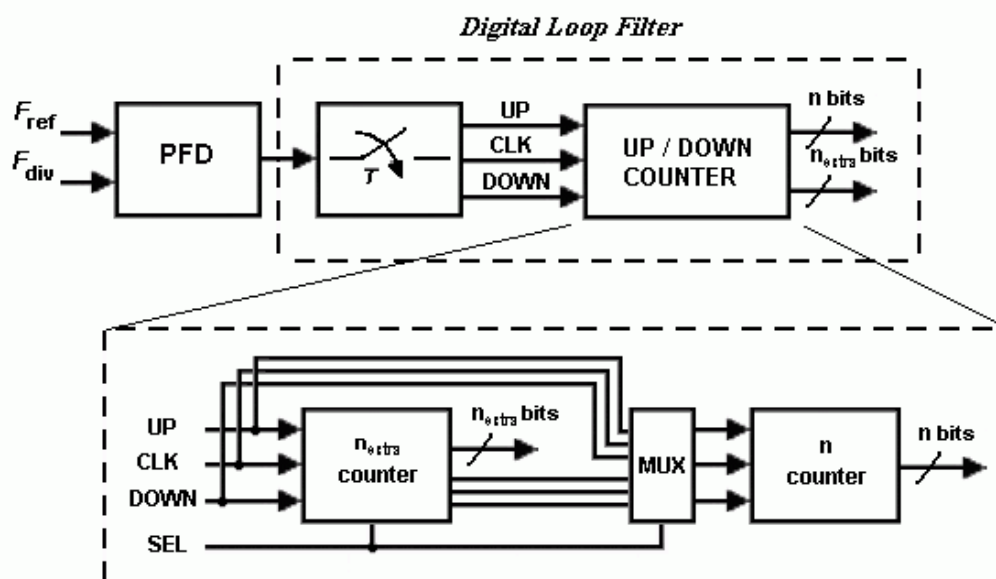


Fig. 5.19. Técnica aplicada ao loop filter digital para controle do ganho.

A largura de banda do *loop filter* pode ser alterada manipulando-se o ganho do laço digital. Uma solução para aplicar este tipo de técnica ao laço digital é aplicar chaves (multiplexadores) para acrescentar bits menos significativos ao contador. Como resultado, o contador original tem a sua frequência de *clock* de entrada dividida, cada bit acrescentado divide o ganho por dois. A Fig. 5.19 ilustra esta técnica.

A adição de bits extras ao conversor D/A durante o transiente pode ser realizada somando-se os bits extras com peso reduzido, isto é, se 4 bits fossem acrescentados, por exemplo, os pesos seriam 2^{-1} , 2^{-2} , 2^{-3} e 2^{-4} .

5.4.4. Projeto da Estabilidade

Para garantir que a arquitetura atinja a exatidão necessária para atender a especificação do padrão de comunicação, pode-se controlar a resolução do sistema através da técnica citada. Por exemplo, se a exatidão precisa ser de 30 ppm, e o *tuning range* é de 40 MHz, a resolução será de 1200 Hz. Para um K_{VCO} de 100 MHz/V a resolução da tensão de controle é de 12 μ V, então, se

$T = 7,5 \times 10^{-10}$, $C_1 = 250 \text{ pF}$ e $I_{cp} = 60 \mu\text{A}$, o ganho do laço digital é dado por

$$K_{dig} = \frac{TI}{C_1} = 1.8 \times 10^{-4}.$$

Dividindo-se a resolução de tensão pelo ganho para encontrar o valor de cada bit, verifica-se que o valor encontrado, que no caso desse exemplo, é de $6,67 \times 10^{-2}$, é menor que a unidade, ou seja, seria necessário um bit com peso de 2^{-4} para ser compatível com a resolução desejada.

A ondulação na tensão de controle causada pela baixa resolução empregada no projeto é um desafio para o desenvolvimento teórico e a implementação da arquitetura. O *dual-path loop filter* proposto usa o detector de fase digital convencional com amostragem da defasagem detectada. Verificou-se, em simulações de transiente, que aumentar o amortecimento quando o PLL está sincronizado, ao mesmo tempo em que a largura de banda do sistema é feita mais estreita, reduziu a ondulação na saída. Com a técnica empregada, o aumento do amortecimento não prejudica o tempo de estabelecimento, como explicado anteriormente. O amortecimento (ζ) pode ser aumentado aplicando (5.10) e as equações:

$$\zeta = \omega_n / 2\omega_z \quad (5.11)$$

$$\omega_c \cong \omega_n^2 / \omega_z. \quad (5.12)$$

Pode-se proceder aumentando-se a relação entre ω_z e ω_p , pois se K_2' é reduzido, então ω_z também diminui. Ao mesmo tempo, se K_1' for reduzido, então obtém-se também aumento do amortecimento e da estabilidade. As mudanças são mostradas na tabela I.

Tabela I

Parâmetros	Durante o transitório	Quase sincronizado
ω_c	$2\pi \times 50 \text{ kHz}$	$2\pi \times 12.5 \text{ kHz}$
ω_n	$2\pi \times 25 \text{ kHz}$	$2\pi \times 6.25 \text{ kHz}$
ω_p	$2\pi \times 200 \text{ kHz}$	$2\pi \times 50 \text{ kHz}$

Se no momento do chaveamento K_2' for dividido por 8 e ω_h (o ganho) por 4, o zero sofrerá uma variação de $\omega_z = 2\pi \times 12.5$ kHz para $\omega_z = 2\pi \times 1.6129$ kHz. A ondulação é reduzida, mas não o suficiente para uma precisão de 20 ppm. Por isso, K_2' foi dividido por 16 para obter uma variação de $\omega_z = 2\pi \times 12.5$ kHz para $\omega_z = 2\pi \times 0.8197$ kHz. Nesse caso ω_c não pode ser reduzido, mas o fator de 4 foi considerado suficiente.

Nas Fig. 5.20 (a) e (b) são mostrados os resultados calculados no MATLAB para a margem de fase e ω_c para os casos em que K_2' é dividido por 8 e 16 respectivamente.

<pre>>> 15*{(2*pi*6.25e3)^2}*0.96875 ans = 2.2409e+10 >> sys Transfer function: s + 1.013e04 ----- s^3 + 3.142e05 s^2 >> allmargin(sys*2.2409e+10) ans = GainMargin: [0 Inf] GMFrequency: [0 Inf] PhaseMargin: 69.1821 PMFrequency: 7.0326e+04 DelayMargin: 1.7169e-05 DMFrequency: 7.0326e+04 Stable: 1</pre>	<pre>>> 15*{(2*pi*6.25e3)^2}*0.953125 ans = 2.2048e+10 >> sys Transfer function: s + 5150 ----- s^3 + 3.142e05 s^2 >> allmargin(sys*2.2048e+10) ans = GainMargin: [0 Inf] GMFrequency: [0 Inf] PhaseMargin: 73.3723 PMFrequency: 6.8751e+04 DelayMargin: 1.8627e-05 DMFrequency: 6.8751e+04 Stable: 1</pre>
--	---

Fig. 5.20. Resultado do cálculo no MATLAB da margem de fase e ω_c (a) usando um divisor por 8 (b) com um divisor por 16.

Os resultados da simulação de transitório para um degrau de 5.1 MHz usando a técnica de mudar a largura de banda do sistema *dual-path* são apresentados na Fig. 5.21. A simulação apresenta resultado similar ao encontrado em uma versão inteiramente analógica do projeto. Na Fig. 5.22 os trechos finais das simulações dessas duas topologias são comparados.

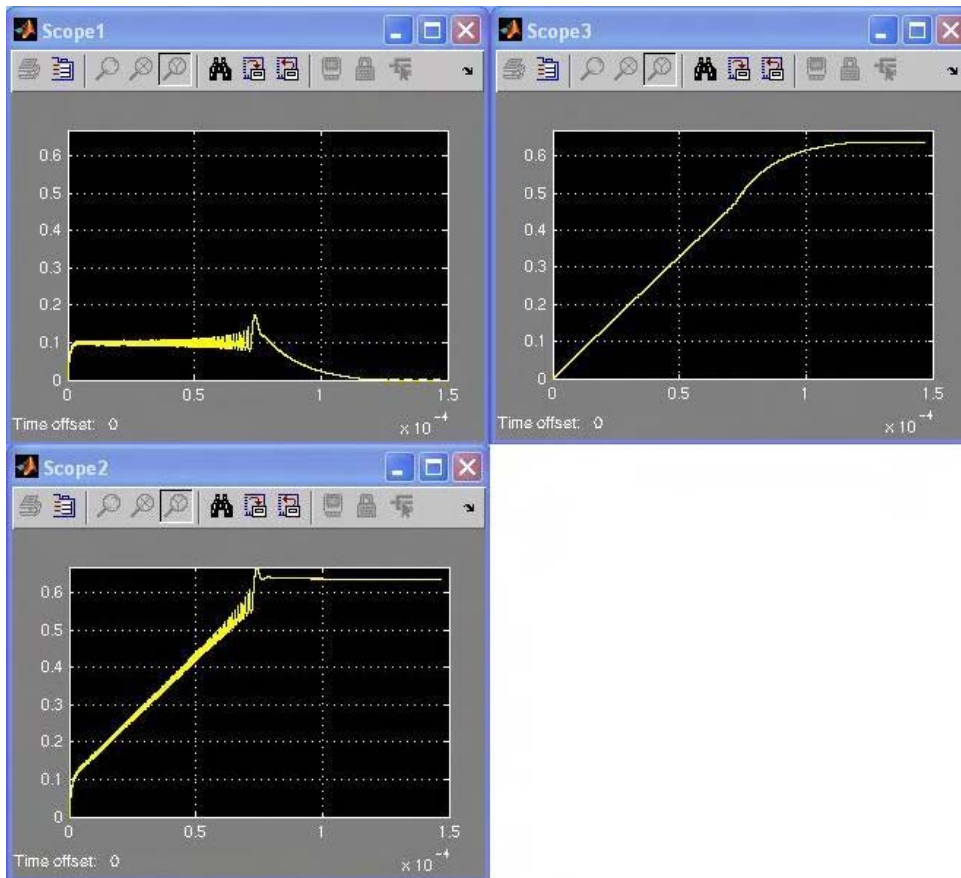


Fig. 5.21. Resposta a um degrau de 5.1 MHz step, $\omega_p = 2\pi \times 200$ kHz ou $\omega_p = 2\pi \times 50$ kHz e $N = 1$, o amortecimento é aumentado por $61/16$ (a) laço analógico (b) laço digital (c) tensão de controle.

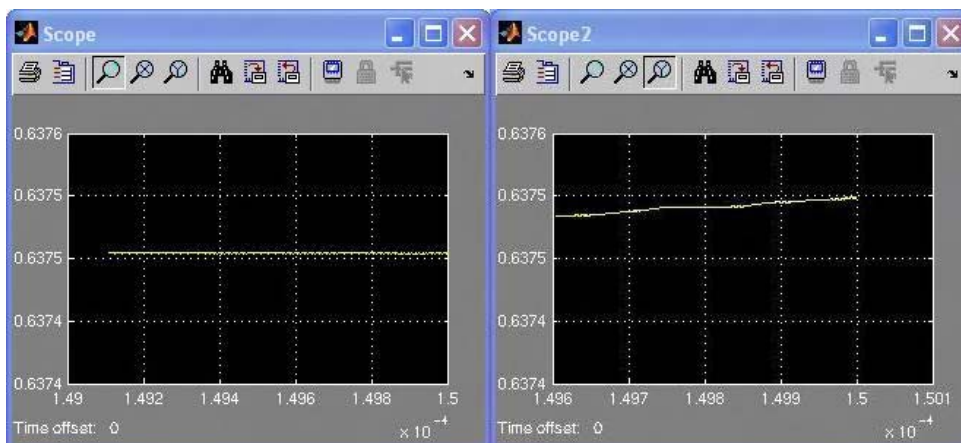


Fig. 5.22. O trecho final da simulação de transiente para um degrau de 5.1 MHz mostrando o erro de fase em $150 \mu\text{s}$ de transitório (a) a versão analógica da simulação usando o chaveamento (c) para a arquitetura proposta.

5.4.5. Comparação dos Resultados

Verificou-se que o melhor instante para se realizar o chaveamento do *loop filter* é quando o PLL está quase sincronizado, antes disso o aumento na estabilidade é desnecessário e aumenta o tempo de estabelecimento. Porém resta determinar qual a alteração no desempenho para cada instante escolhido. Na simulação mostrada na Fig. 5.23 o degrau foi de 8 MHz. Considerou-se que o regime permanente foi atingido em 90 μ s logo após o chaveamento em 85 μ s.

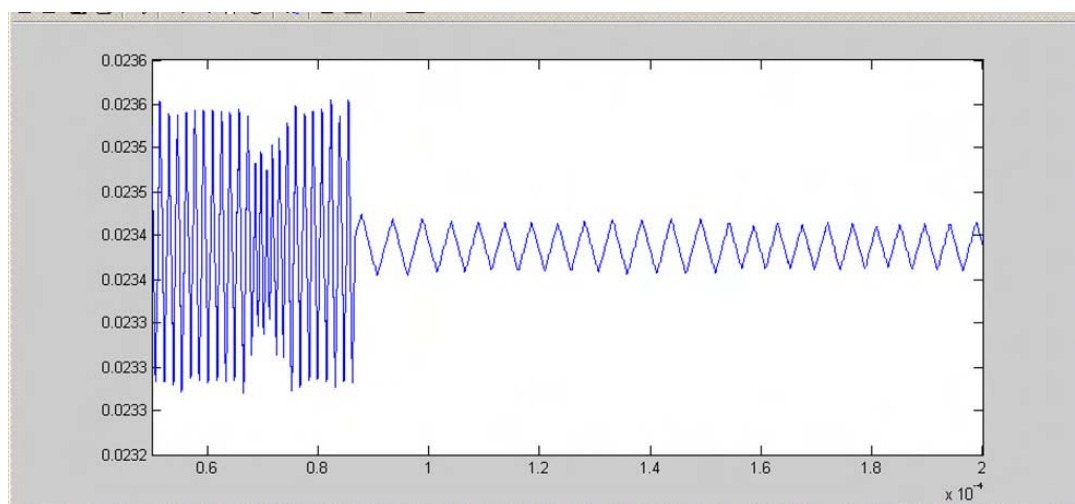


Fig. 5.23. O resultado de simulação apresentado na Fig. 5.22, com ênfase na redução da ondulação obtida pelo aumento do amortecimento quando o PLL está quase em “regime permanente”.

Para simular o tempo de estabelecimento da arquitetura em termos de especificação, foi utilizado o *tuning range* proposto para implementação nessa tese. Portanto, o $\Delta f = 40$ MHz e ω_c é chaveada de $2\pi \times 50$ kHz para $2\pi \times 3.125$ kHz. O resultado da simulação é apresentado na Fig. 5.24.

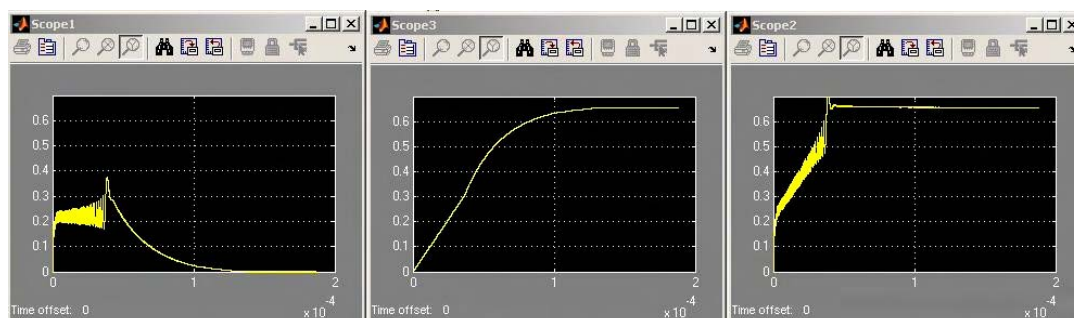


Fig. 5.24. Simulação de *settling time* semelhante a da Fig. 5.21, para um tuning range de $40\text{MHz} \times 4$ (devido a uma seleção direta de canal de 2 bits) de 832 a 992 (MHz), o qual atende a alguns padrões de RFID, como o europeu 865 MHz a 868 MHz e o americano 902 MHz a 928 MHz.

Analisando os resultados das Fig. 5.25 e Fig. 5.26, observa-se que após $150\ \mu\text{s}$ o chaveamento faz-se necessário, bem como acelera alcançar o estabelecimento. Uma estratégia adequada para essa arquitetura seria programar o controle de seleção dos canais para que antes do *hopping* o chaveamento aconteça em um tempo pré-estabelecido, por exemplo $150\ \mu\text{s}$.

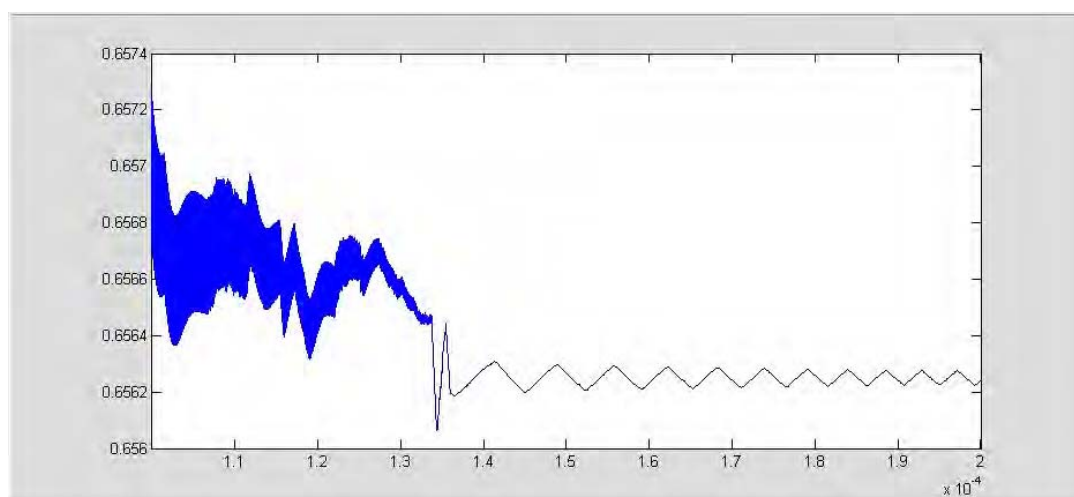


Fig. 5.25. Simulação apresentada na Fig. 5.24 com chaveamento em $135\ \mu\text{s}$.

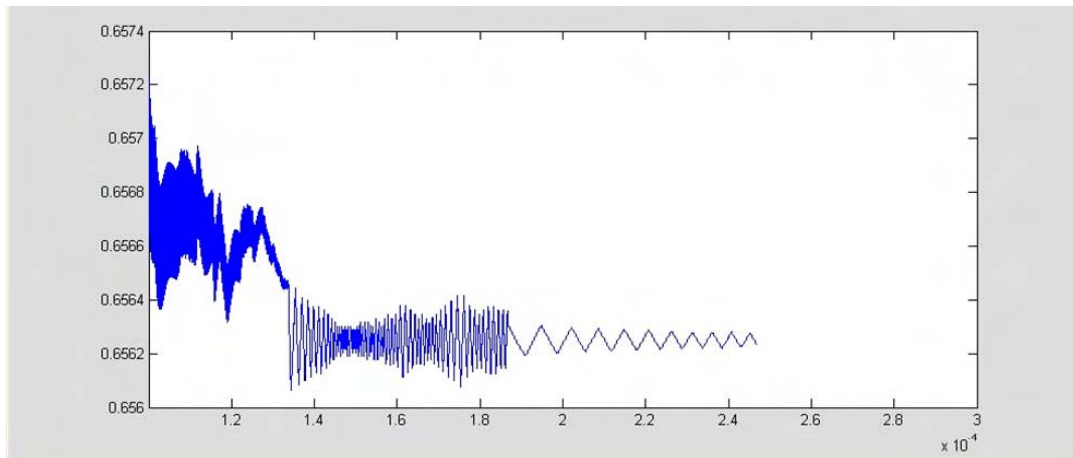


Fig. 5.26. Simulação apresentada na Fig. 5.24 com chaveamento em 185us.

5.5. Abordagem Usando Controle Digital

No caso da arquitetura proposta o controlador do sistema de controle digital, só será variado pelo laço analógico e pelo ganho K_2 . O projeto de controle digital não terá flexibilidade total. Mesmo assim, o modelo no domínio-z permitirá que o projeto seja realizado com mais precisão e segurança do que quando usado a emulação. O equacionamento do circuito no domínio-z foi obtido calculando-se a equivalência discreta de primeira ordem do VCO através de duas etapas. Na Fig. 5.27(a) foi aplicada a propriedade distributiva para separar os laços analógico e digital, obtendo-se o diagrama da Fig. 5.27(b). Depois é calculada a equivalência discreta do VCO e multiplicada ao laço digital. Por fim, é calculada a equivalência do laço analógico formado. A função de transferência discreta de malha aberta é determinada pela soma dos dois laços. A dedução da equação característica do sistema é feita a seguir:

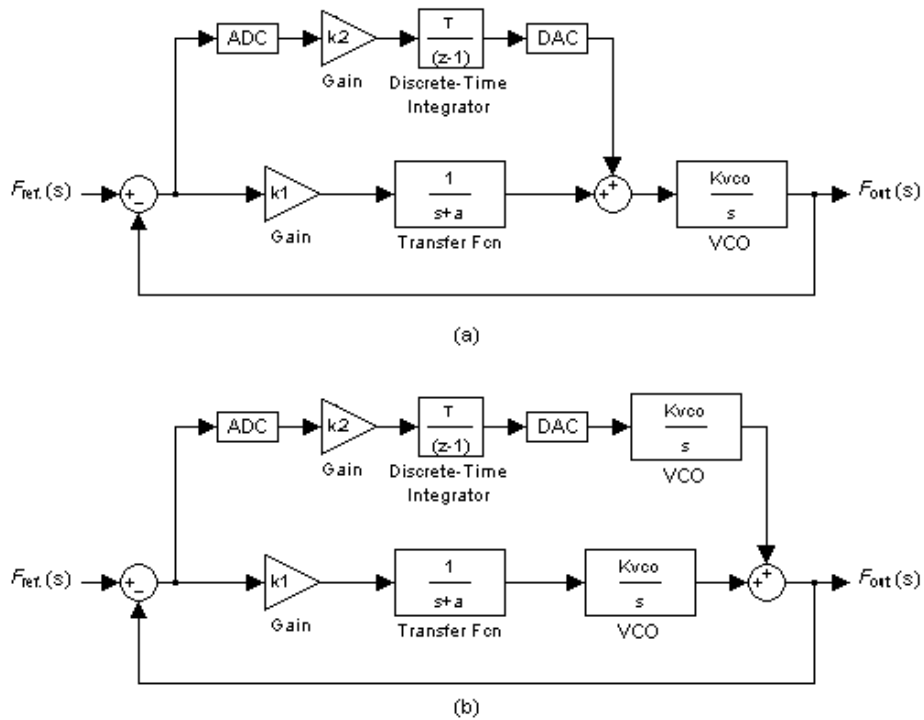


Fig. 5.27. Diagrama em blocos do PLL proposto (a) diagrama atual (b) diagrama equivalente usado na determinação da equivalência discreta de primeira ordem.

Se a equivalência discreta do VCO e do laço analógico for respectivamente [58]:

$$\frac{TK_{vco}}{(z-1)} \quad (5.13)$$

$$e \quad \frac{K_1 K_{vco} [(aT - 1 + e^{-aT})z + (1 - e^{-aT} - aTe^{-aT})]}{a^2(z-1)(z - e^{-aT})} \quad (5.14)$$

sendo T o período de amostragem, $a = \omega_p$ e a equivalência discreta igual a [58]

$$(1 - z^{-1})L^{-1} \left\{ \frac{G(s)}{s} \right\}_{t=KT} \quad (5.15)$$

Então, somando-se os dois laços obtém-se

$$F(z) = \frac{K_1 K_{VCO} [(aT - 1 + e^{-aT})z + (1 - e^{-aT} - aTe^{-aT})]}{a^2 (z-1)(z - e^{-aT})} + \frac{K_2 K_{VCO} T^2}{(z-1)^2}$$

ou

$$F(z) = \frac{\frac{K_1 K_{VCO}}{a^2} (z-1) [(aT - 1 + e^{-aT})z + (1 - e^{-aT} - aTe^{-aT})] + T^2 K_2 K_{VCO} (z - e^{-aT})}{z^3 - (2 + e^{-aT})z^2 + (1 + 2e^{-aT})z - e^{-aT}}. \quad (5.16)$$

Os parâmetros do sistema podem ser encontrados através de sua função de transferência em (5.16). Usualmente, em controle digital, as especificações do sistema são extraídas do lugar das raízes no plano-z discreto e passadas para variáveis no domínio-s contínuo. Do denominador da equação de malha aberta pode-se determinar os pólos discretos.

Da equação (5.16), $p_1 = e^{-aT}$ e $p_{2,3} =$ raiz de $\{(z - 1)^2\}$, que são valores próximos à unidade para $\zeta \cong 1$.

Usando a transformada bilinear, isto é,

$$z = e^{sT}, \quad (5.17)$$

conclui-se que $\omega_p = a$, como já era esperado.

O numerador da equação de malha aberta pode ser escrito como a seguir.

$$a_1 z^2 + (a_2 - a_1 + K_2 T^2)z - (a_2 + K_2 T^2 e^{-aT}) \quad (5.18)$$

sendo $a_1 = \frac{K_1}{a^2} (aT - 1 + e^{-aT})$ e $a_2 = \frac{K_1}{a^2} (1 - e^{-aT} - aTe^{-aT})$.

Usando Báscara, pode-se deduzir que os zeros serão dados como abaixo.

$$\Delta = (a_2 + a_1 + K_2 T^2)^2 - 4a_1 K_2 T^2 (1 - e^{-aT}) \quad \text{e} \quad z_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2a_1},$$

sendo que $b = a_2 - a_1 + K_2 T^2$ e Δ pode ser aproximado usando polinômio de Taylor como $\sqrt{x_0 + x} \cong \left(\sqrt{x_0} + \frac{x}{2\sqrt{x_0}} - \frac{x^2}{8\sqrt{x_0^3}} + \frac{x^3}{16\sqrt{x_0^5}} \right)$, sendo que $x_0 = (a_2 + a_1 + K_2 T^2)^2$ e $x = -4a_1 K_2 T^2 (1 - e^{-aT})$.

Os zeros podem ser aproximados usando-se dois termos do polinômio, como a seguir:

$$z_1 = 1 - \frac{K_2 T^2 (1 - e^{-aT})}{(a_1 + a_2 + K_2 T^2)} \quad \text{e} \quad z_2 = \frac{K_2 T^2 (1 - e^{-aT})}{(a_1 + a_2 + K_2 T^2)} - \frac{a_2 - K_2 T^2}{a_1} \quad (5.19)$$

O zero no domínio contínuo é equivalente ao zero positivo no domínio discreto. Usando a transformada bilinear, obtém-se a equação abaixo:

$$\omega_z = \frac{\ln(z_1)}{T} \quad (5.20)$$

Sendo assim, a margem de fase pode ser calculada pela equação (4.11) como:

$$\phi_m = \text{tag}^{-1} \left(\frac{\omega_c}{\omega_z} \right) - \text{tag}^{-1} \left(\frac{\omega_c}{\omega_p} \right) \quad (4.11)$$

O cálculo dos parâmetros no domínio contínuo permite que o mesmo método de projeto, que se usaria no PLL convencional, seja aplicado ao PLL híbrido. Em termos práticos, isso auxilia bastante, porém, somente se consegue uma precisão suficiente no projeto das especificações de tempo de subida, *overshoot* e tempo de estabelecimento, se usado o lugar das raízes da planta no domínio-z discreto.

5.5.1. Projeto do *Loop Filter* Usando Controle Digital

O objetivo é que nos dois estados, antes e depois do chaveamento do *loop filter*, o PLL permaneça estável. O MATLAB foi utilizado para projetar a estabilidade

dessa arquitetura usando controle digital. Ao se utilizar o MATLAB deve-se tomar cuidado na determinação da função de transferência de malha aberta do sistema devido a aproximações do software. A equação (5.12) foi obtida por cálculo manual e tem sido utilizada, nessa tese, para traçar o lugar das raízes conseguindo uma boa precisão nos cálculos (ver apêndice A4).

O programa do apêndice A4 foi utilizado para projetar a estabilidade do sistema. O programa calcula a equação característica através de (5.10) e abre o rltool a partir desta. O rltool é uma ferramenta versátil para projetos de controle e, nesse estudo, tem sido usada principalmente para o traçado do lugar das raízes, determinação de ω_c , margem de fase e erro de regime. Através do lugar das raízes puderam-se projetar as especificações do PLL.

As figuras Fig. 5.28 à Fig. 5.33 mostram os resultados finais da análise feita no rltool. Os ganhos dos blocos encontrados por essas análises serão posteriormente usados como base para projetar os blocos em nível de transistor.

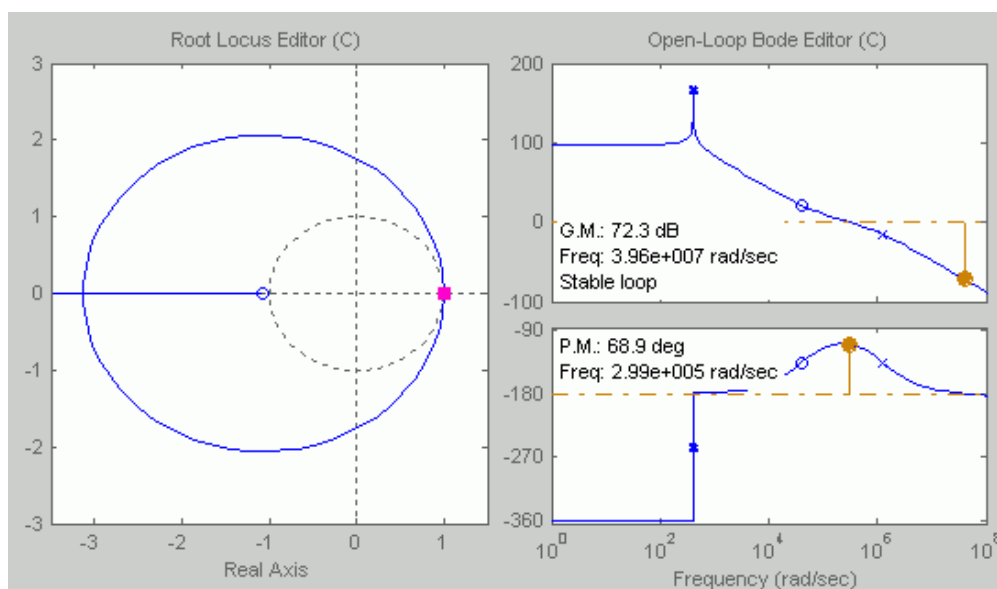


Fig. 5.28. Lugar das raízes e diagrama de Bode do PLL híbrido para $K_2/2$ após o chaveamento.

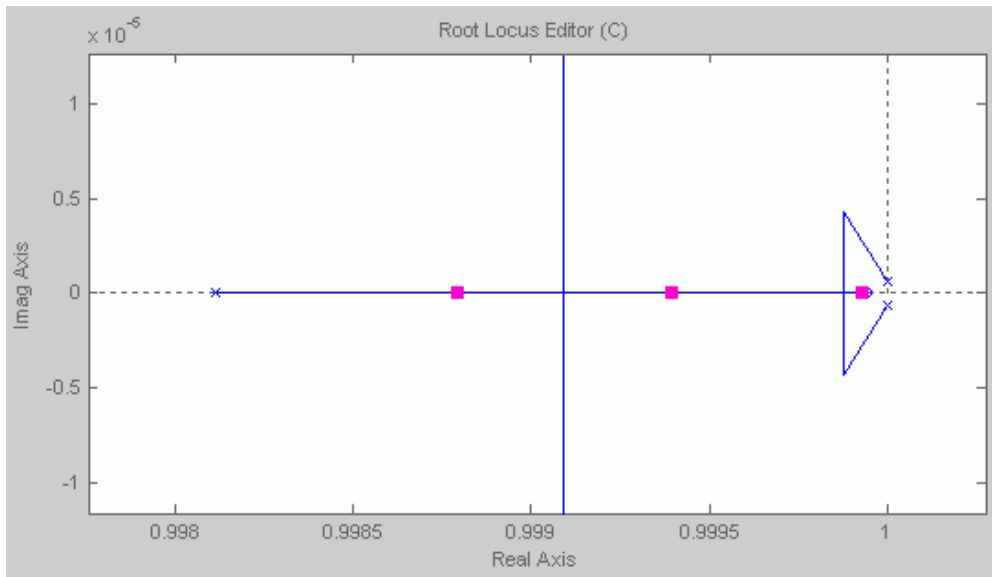


Fig. 5.29. Posição dos pólos em malha fechada para $K_2/2$.

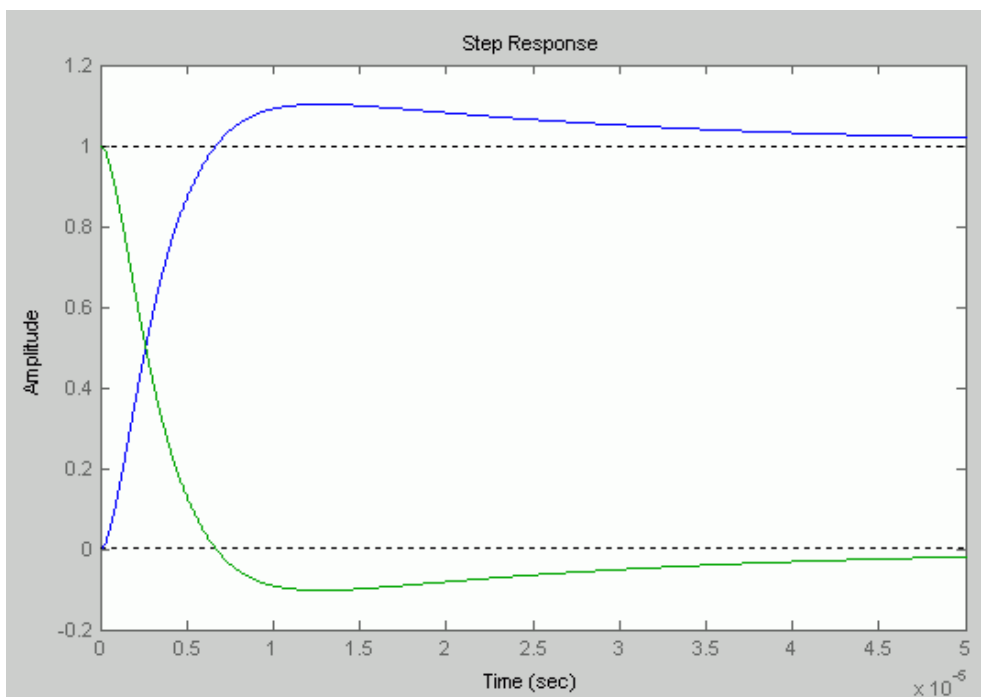


Fig. 5.30. Resposta ao degrau para $K_2/2$.

Através da análise realizada mostra-se como o chaveamento do ganho K_2 altera a dinâmica do sistema. Comparando as Fig. 5.29 e Fig. 5.32, verifica-se que a posição do zero altera a posição dos pólos em malha fechada do sistema, isso

implica em estar ou não dentro do lugar geométrico que atende às especificações requeridas. Quanto mais próximo o zero discreto está de 1, verticalmente, maior é o amortecimento do sistema.

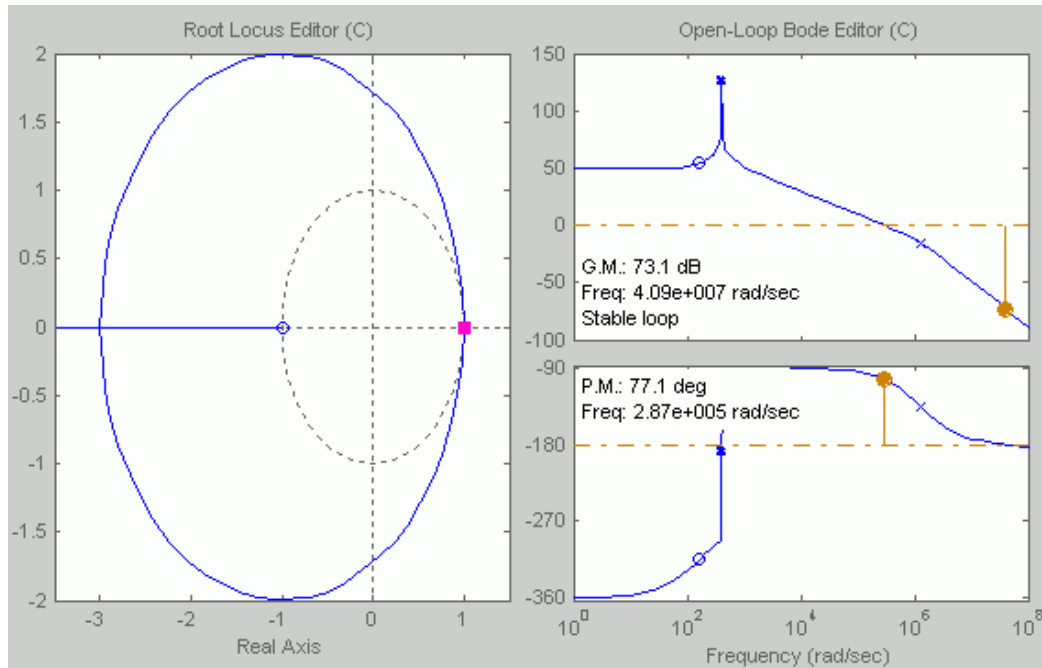


Fig. 5.31. Lugar das raízes e diagrama de Bode do PLL híbrido para $K_2/(2^8)$ após o chaveamento.

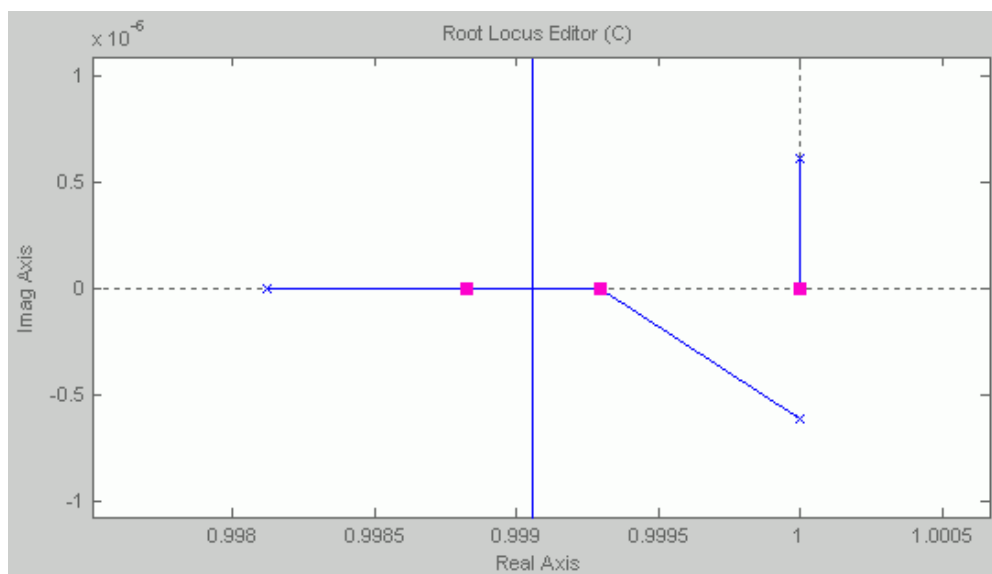


Fig. 5.32. Posição dos pólos em malha fechada para $K_2/(2^8)$.

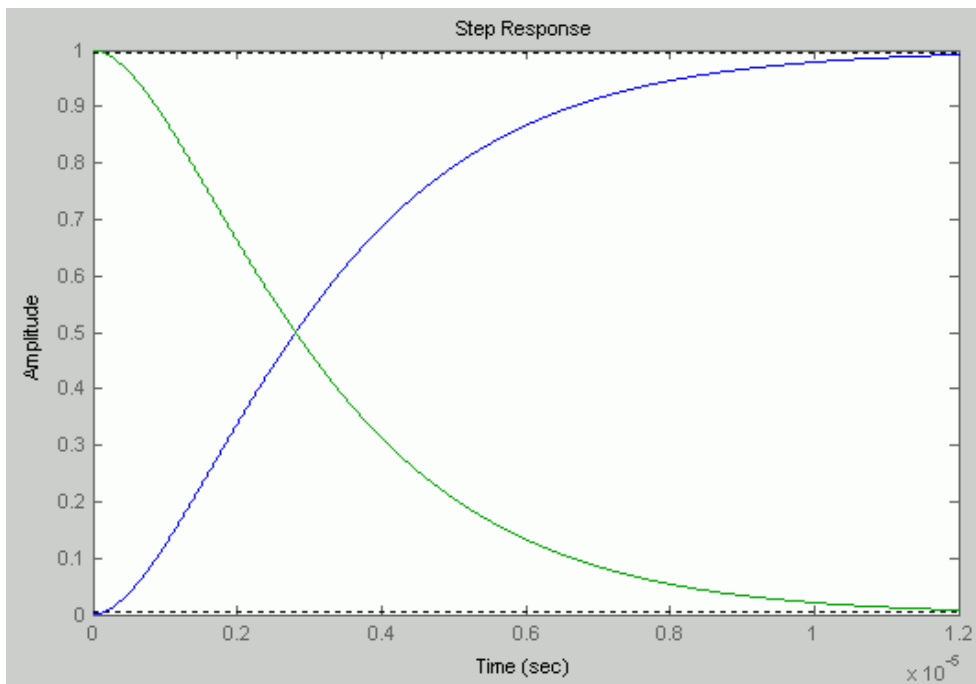


Fig. 5.33. Resposta ao degrau para $K_2/(2^8)$.

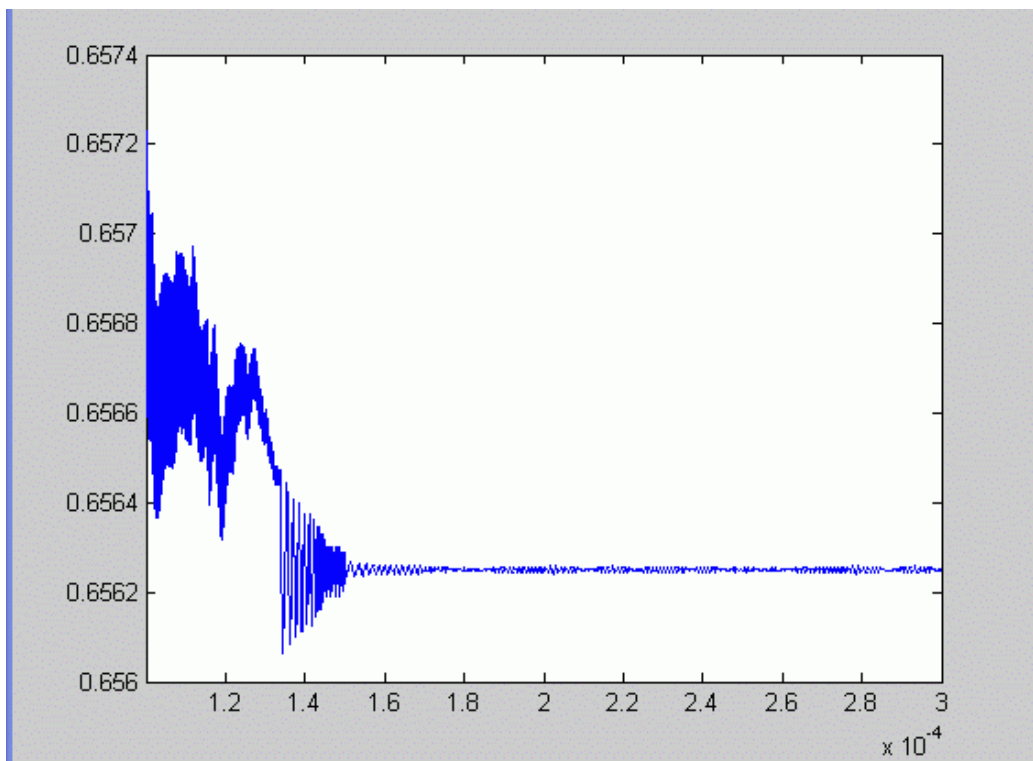


Fig. 5.34. Simulação apresentada na Fig. 5.21 demonstrando um menor erro de regime depois do chaveamento, comparado a resultados prévios (Fig. 2.15 e Fig. 2.16)

No exemplo apresentado foi usada $f_s = 7.5e-10$, antes e depois do chaveamento e K_2 foi dividido por 2^8 . Apesar de se obter estabilidade nos dois estados, não é possível provar que o PLL será estável durante o chaveamento. Para isso, as simulações de transiente no Simulink podem demonstrar qual é o comportamento, como apresentado na Fig. 5.34. Esse resultado mostra que a ondulação na tensão de controle do VCO foi bem reduzida e que a especificação de exatidão (*accuracy*) deve ser atendida.

5.6. Simulação de Ruído de Fase no Simulink

O Simulink foi usado para simular a influência do ruído no PLL proposto utilizando ruído injetado [56]. O PLL é modelado como um sistema linear, como mostrado na Fig. 5.35.

Usando os conceitos da seção 4.2 o PLL foi analisado em termos de ruído.

Várias simulações foram feitas para analisar os cálculos teóricos do projeto do PLL. Nessas simulações foi analisada a influência do ruído de entrada, interno e do

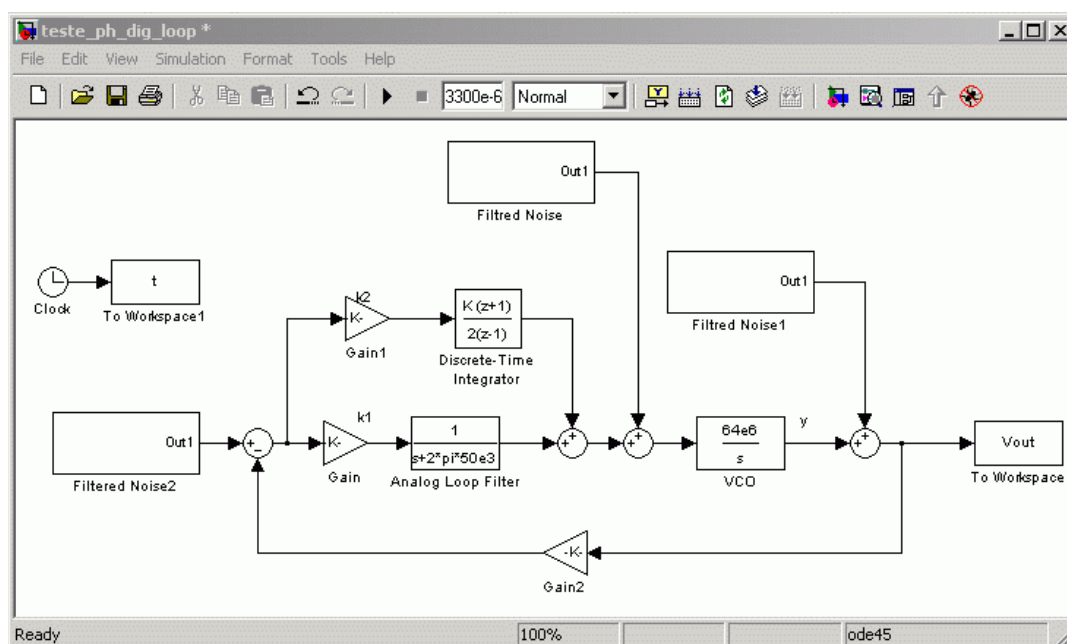


Fig. 5.35. Simulação de ruído no simulink para a topologia de PLL proposto.

VCO para algumas situações do sistema e de configuração do PLL. O PLL teve $N = 16$ e 2412 e a largura de banda foi variada conforme a técnica empregada de chaveamento do *loop filter*. Nas Fig. 5.36 e Fig. 5.37 são mostradas as comparações do ruído de fase de saída do PLL para 3 casos distintos, para dois valores de N diferentes.

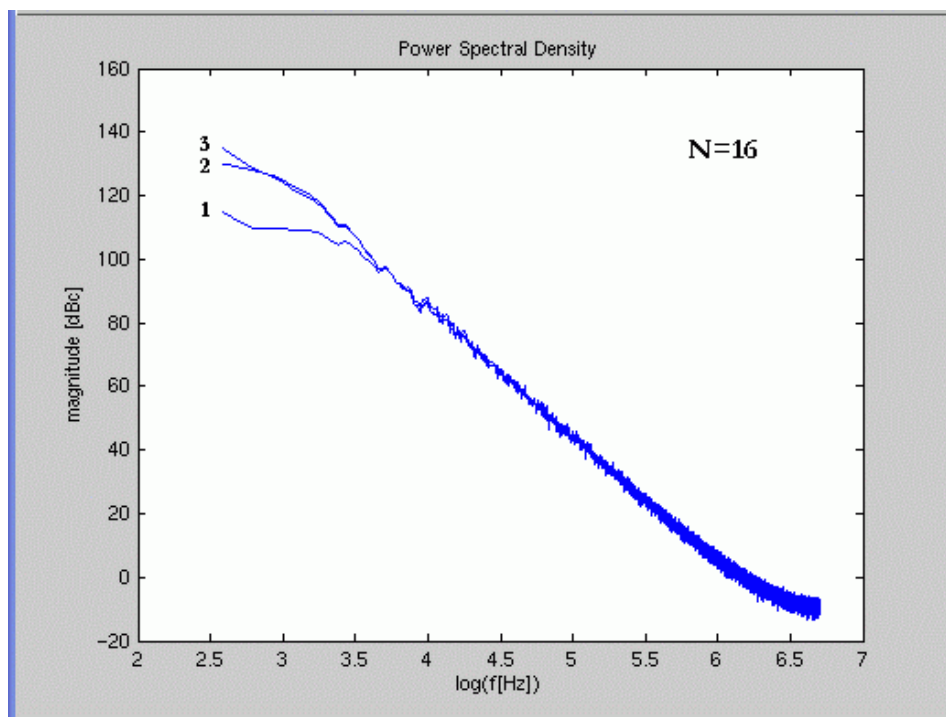


Fig. 5.36. Simulação de ruído no Simulink para 3 estados do *loop filter* com $N = 16$ (1) antes do chaveamento com $\omega_c = 2\pi \times 50$ kHz (2) depois do chaveamento com $\omega_c = 2\pi \times 12.5$ kHz (3) depois do chaveamento com $\omega_c \cong 2\pi \times 12.5$ kHz e $k_2 = k_2/4$.

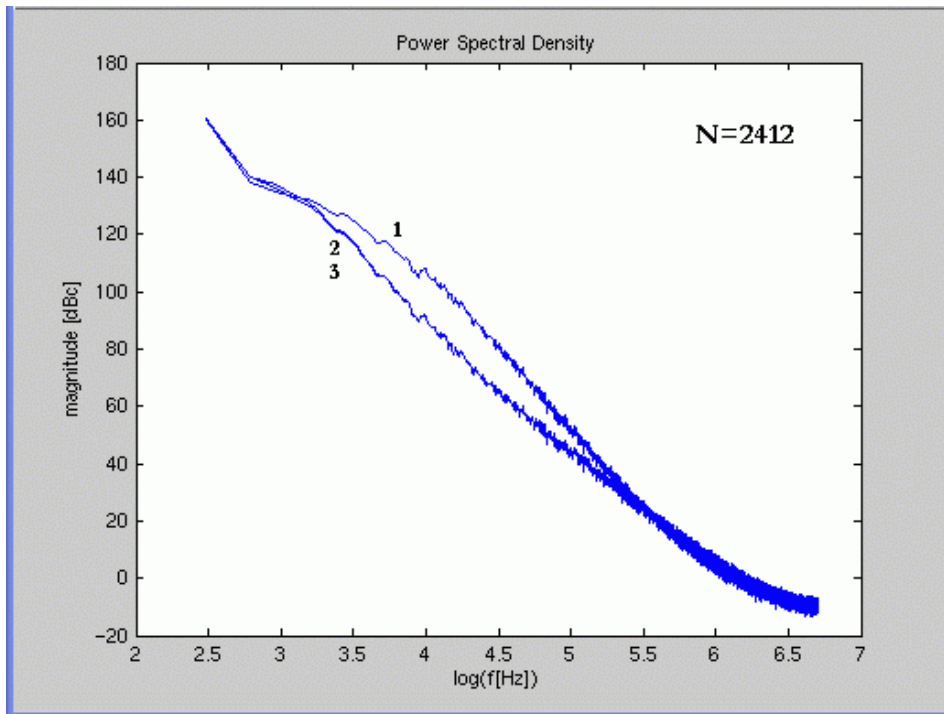


Fig. 5.37. Simulação de ruído no Simulink para 3 estados do *loop filter* com $N = 2412$ (1) antes do chaveamento com $\omega_c = 2\pi \times 50$ kHz (2) depois do chaveamento com $\omega_c = 2\pi \times 12.5$ kHz (3) depois do chaveamento com $\omega_c \cong 2\pi \times 12.5$ kHz e $k_2 = k_2/4$.

Capítulo VI

6. Metodologia de Projeto Utilizada e Implementação da Arquitetura Proposta

Nesse capítulo será descrita a metodologia empregada no projeto da arquitetura em circuito integrado, bem como o projeto da arquitetura e seus blocos, desde a sua concepção até o leiaute da mesma. A topologia que foi desenvolvida no Simulink foi estruturada com células em nível de transistor para ser implementada usando o processo CMOS 0.35 μm da AMS com 2V de alimentação. Essa tecnologia é compatível com CMOS padrão. Os blocos são compostos de células da biblioteca de células digitais padrão da AMS ou por blocos criados a partir de transistores. A metodologia *top-down* de projeto de circuito integrado (CI) tem sido empregada utilizando-se simulações em nível de sistema e de transistor. As principais ferramentas utilizadas nesse trabalho foram MATLAB, Mentor Graphics e Cadence. Na fase final do projeto, a metodologia da Cadence para projeto de PLL considerando ruído foi aplicada.

As ferramentas de EDA (*Electronic Design Automation*) são utilizadas para as simulações em nível de transistor analógicas ou mistas, de transiente, resposta em frequência, estatística (Monte Carlo), regime permanente (*Steady-State*), etc, nos projetos dos blocos que compõem o sistema. No projeto analógico, em nível de transistor, as ferramentas de EDA são empregadas para o projeto de circuitos usando esquemático, leiaute dos circuitos, verificações como, *Design Rule Checking* (DRC) ou *Layout Versus Schematic* (LVS), extração de parasitas e simulação pós-leiaute. Deve-se salientar que as ferramentas de EDA também permitem simulações em nível de sistema com auxílio de bibliotecas ou macromodelos construídos pelo

usuário. Desse modo, simulações que despendem tempo e espaço em disco, como a simulação de transiente, por exemplo, podem ser aplicadas nesse nível, com maior agilidade, reduzindo o tempo de projeto necessário.

O sintetizador foi proposto para atender a alguns padrões asiáticos de RFID na frequência de UHF sendo, portanto, uma arquitetura *multi-standard*. Por isso, o sintetizador foi especificado com largura de canal de 1 MHz, operando na faixa de 850 MHz a 1010 MHz, obtendo uma banda de 160 MHz. Ruído de fase abaixo de -100 dBc/Hz, tempo de estabelecimento menor que 200 μ s e exatidão de 30 ppm também são especificações desse projeto.

Na seção 6.1, a metodologia usada no projeto do CI é descrita. Na seção 6.2, o projeto da aplicação proposta para a arquitetura é executado. O projeto dos blocos é desenvolvido na seção 6.3. Essa descrição é dividida em quatro partes, o laço analógico, o VCO híbrido e o primeiro divisor de frequência, o divisor programável e o laço digital. Na seção 6.4 o projeto do PLL é desenvolvido e todos os demais resultados apresentados.

6.1. Metodologia de Projeto do Circuito Integrado

No projeto, foram utilizadas ferramentas de EDA, o MATLAB e o Simulink para o desenvolvimento da arquitetura. A Fig. 6.1 ilustra a metodologia empregada.

Após a primeira fase de projeto em nível de sistema, usando MATLAB e Simulink, iniciou-se o projeto do PLL com modelos comportamentais usando a ferramenta Mentor Graphics, que até então deveria ser usada até o fim do projeto. Apesar disso, quando se disponibilizou a ferramenta Cadence, para estações com muito maior capacidade computacional, o projeto deu continuidade nessa ferramenta. Abaixo serão descritos os passos seguidos para realização do projeto.

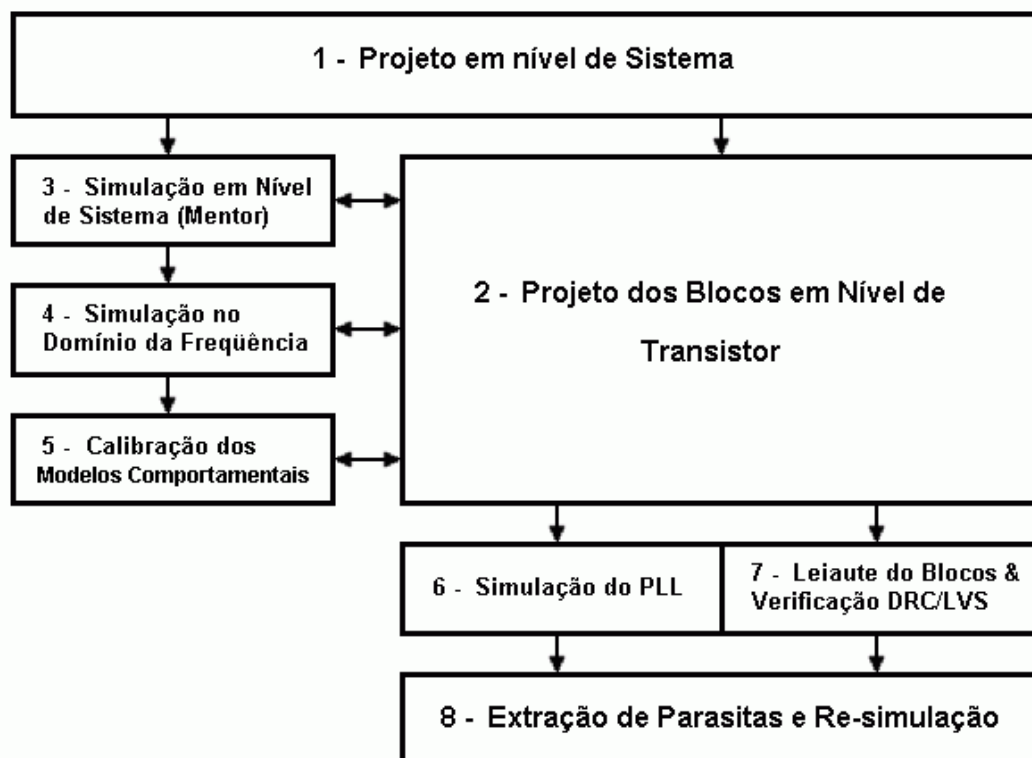


Fig. 6.1. Metodologia usada no projeto da arquitetura proposta.

1 – Projeto da Arquitetura em Nível de Sistema: O MATLAB foi utilizado para projetar a estabilidade e determinar as especificações dos projetos dos blocos, a partir do que foi determinado para o desenvolvimento da arquitetura. Alguns programas foram criados para projetar a estabilidade, sendo que muitos deles executam a ferramenta rtool do MATLAB. O Simulink foi utilizado para simulação da arquitetura e seu desenvolvimento usando os blocos que foram criados para modelar a arquitetura em termos de circuito.

2 – Projeto dos Blocos: Os blocos em nível de transistor são projetados usando as ferramentas de EDA. Todas as simulações necessárias para satisfazer as especificações requeridas, como frequência de operação, ruído de fase e dissipação de potência devem ser realizadas no projeto dos blocos, bem como as células devem ser otimizadas nos parâmetros que dizem respeito à junção dos blocos, como atrasos da célula, acoplamento CC, amplitude e impedância de saída. Os projetos dos blocos e do sistema devem ser executados em paralelo para que as

especificações obtidas em nível de sistema sejam obtidas no projeto dos blocos e o desempenho desses blocos seja avaliado em nível de sistema. Além disso, visando verificar o efeito da junção dos blocos, algumas células são simuladas em conjunto e posteriormente modelos comportamentais são criados a partir dessa junção. Os fatores que determinam que duas ou mais células sejam simuladas em conjunto ficarão mais claros no decorrer desse capítulo. Esses fatores dependem da frequência de operação e pré-requisitos das células. O VCO e o *prescaler* são exemplos desse método. O VCO e o *prescaler* são os blocos que operam com as maiores frequências do PLL e definem a frequência de saída. No fluxo utilizado, o VCO e o divisor de frequências são modelados pelo mesmo bloco, utilizando PPV (*Perturbation Projection Vector*). Esse bloco, pertence a uma das bibliotecas de PLL da Cadence (*pllMMLib*) e tem a vantagem de não impor aos demais blocos a frequência de saída [59].

3 – Simulação em Nível de Sistema (Mentor): O sistema é projetado usando blocos em Verilog-A e desse modo reduzindo bastante o tempo gasto em simulações no domínio do tempo e de um modo geral agilizando o projeto. Esses modelos incluem um detalhamento dos parâmetros em nível de dispositivo, permitindo uma estimativa prévia do projeto em nível de transistor. Foram utilizados blocos da biblioteca COMMLIB da Mentor Graphics, equivalentes aos usados com a ferramenta Spectre HDL da Cadence. Posteriormente, os modelos comportamentais serão calibrados com os blocos em nível de transistor que estão sendo projetados paralelamente para atender às especificações do sistema.

4 – Simulação no Domínio da Frequência: O PLL deve ser simulado para verificar as especificações de ruído de fase e espúrios do sistema. Essas simulações podem utilizar modelos comportamentais que descrevam a função transferência dos blocos e a inserção de algumas estimativas para o ruído de entrada, da tensão de controle e do VCO. O método usado para obter a simulação pode variar, como por exemplo, usar a simulação CA junto com ruído, ou quando não disponível, usar a simulação transiente para posteriormente, aplicar a FFT (*Fast Fourier Transform*). A resposta do sistema ao ruído é determinada verificando-se a influencia de cada fonte de ruído e cada bloco em malha fechada para o ruído de saída. A análise para tons espúrios é idêntica.

5 – Calibração dos Modelos Comportamentais: As ferramentas de EDA possuem formas de calibrar os modelos comportamentais automaticamente. A calibração também pode ser realizada manualmente, tomando-se o cuidado de incluir todos os parâmetros importantes para o bloco em específico. O fluxo da Cadence para projeto de PLL considerando ruído (SpectreRF Noise-Aware PLL Flow) foi aplicado para obter os modelos. Nesse fluxo, alguns blocos de circuito são simulados em nível de transistor usando simulação de regime permanente e de ruído. Durante a simulação a ferramenta extrai automaticamente o modelo que posteriormente será utilizado em simulações de transiente. Esse modelo pode ser em Verilog-A ou macromodelo, ambos incluirão o ruído extraído. Quando o circuito é modelado por PPV, como por exemplo, o VCO, o comportamento do circuito em relação ao ruído é especialmente preciso, incluindo simulações de efeitos não-lineares como, *injection locking* e *injection pulling*.

6 – Simulação do PLL: Após a calibração dos modelos, esses blocos são simulados junto com a parte digital do projeto (*digital loop filter*). Um cuidado especial deve ser tomado para os blocos do laço digital, pois muitos blocos, por serem novos, não podem ser calibrados automaticamente com a inclusão de ruído. Esses blocos podem ser calibrados manualmente ou por outra ferramenta da Cadence. A simulação do PLL é realizada no domínio do tempo, basicamente por simulação de transiente, outros tipos de simulação podem não convergir, bem como são menos precisas para efeitos não-lineares. Como o oscilador é um circuito autônomo e os outros blocos são estimulados (*driven circuits*) pela frequência de referência, não é recomendável usar simulação de regime permanente. A simulação de transiente pode ser usada para verificar as especificações analisando a forma de onda, como tempo de estabelecimento e exatidão, quanto pode ser usada para analisar o ruído do sistema.

7 – Leiaute do Circuito e Verificação DRC / LVS: Por conter circuitos de RF, o leiaute do circuito é realizado observando-se as características do projeto na presença de sinais de RF, como fontes de ruído por acoplamento ou formadas em resistências parasitas. Além disso, as técnicas usadas em circuitos analógicos também são empregadas, como simetria, casamento de pares diferenciais, bem

como, técnicas para evitar acoplamento de sinais resultantes do chaveamento digital.

8 – Extração de Parasitas e Re-Simulação: A extração dos componentes parasitas (capacitância, resistência) do leiaute é realizada e o circuito é verificado novamente através de re-simulação. Dependendo do resultado, o projeto pode ser reformulado.

6.2. Projeto do PLL e Simulações Usando Modelos Comportamentais

Na Fig. 6.2 está ilustrado o esquemático usado para simulação da arquitetura *dual-path* proposta. O laço digital apresenta dois pontos de controle nesse teste, SEL1 e SEL2. O PLL pode ser projetado usando os conceitos descritos no capítulo 4, considerando-se três fatores principais: o padrão definido, a topologia

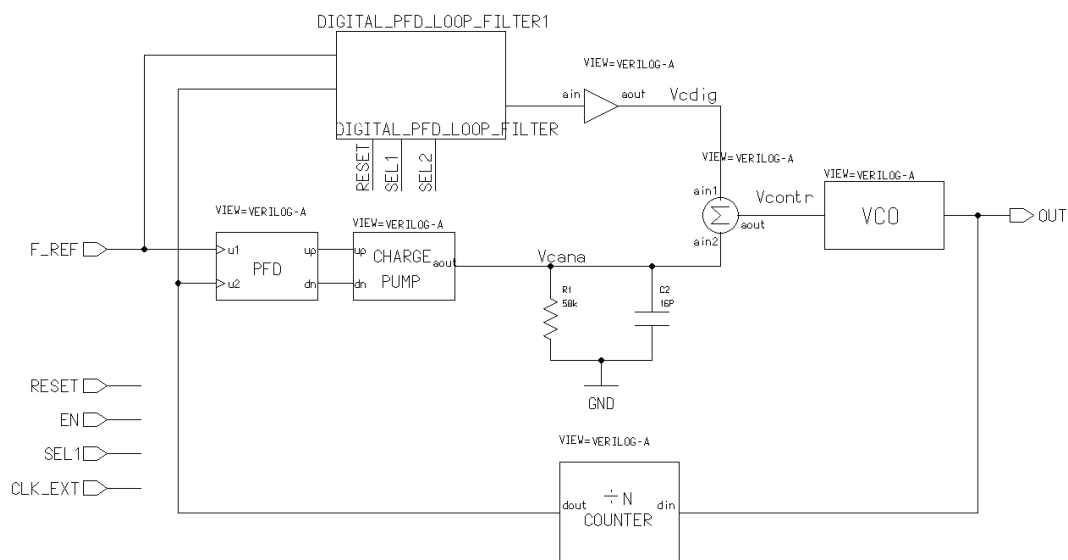


Fig. 6.2. Estrutura usada para simulação em nível de sistema da arquitetura *dual-path* proposta utilizando modelos comportamentais da biblioteca da Mentor Graphics.

desenvolvida no capítulo 5 e simulações usando modelos comportamentais. Nessa primeira fase do projeto, as simulações têm como objetivo analisar a resposta transitória para uma determinada largura de banda (ω_c) e verificar as especificações de tempo de estabelecimento e exatidão.

Portanto, considerando-se $\zeta = 1$, $\omega_n = 2\pi \times 25$ kHz, $\omega_p = 2\pi \times 200$ kHz e $\omega_z = 2\pi \times 12,5$ kHz. Para $C_2 = 16$ pF, obtém-se através das equações 4.4 e 4.5:

$$R = \frac{1}{\omega_p C_2} = 50k\Omega \text{ e } C_1 = \frac{1}{\omega_z R} = 255pF$$

Assumindo-se $F_{ref} = 1$ MHz, $N = 900$ e um $K_{VCO} = 2\pi \times 100$ MHz/V, pode-se determinar a corrente do *charge pump* como deduzido abaixo:

$$K_D = \frac{I}{2\pi C_1 N} = \frac{\omega_n^2}{K_{VCO}} = 40 \Rightarrow I = 57 \mu A$$

O resultado da simulação de transiente usando o *testbench* da Fig. 6.2 está mostrado na Fig. 6.3(a). Nessas simulações foi obtido um tempo de estabelecimento de 300 us, considerando que se deseja uma exatidão de 30 ppm.

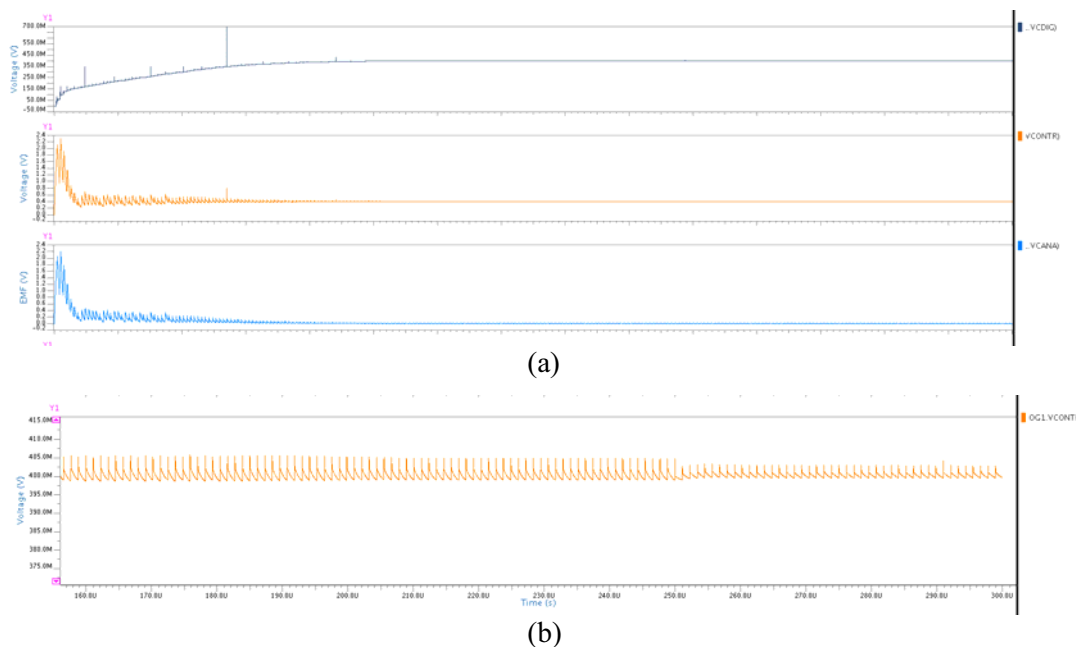


Fig. 6.3. Resultado da simulação de tempo de estabelecimento para um degrau de 40 MHz (a) para o laço digital, tensão de controle e laço analógico (b) mostrando em detalhe a amplitude da tensão de controle.

Na Fig.6.3 (b) está mostrado o resultado de simulação na situação em que foram utilizados dois tipos de chaveamento. O primeiro chaveamento altera o ganho do laço digital, através da técnica desenvolvida no capítulo 5, com o objetivo de corrigir problemas na dinâmica do sintetizador proposto. No segundo chaveamento, uma chave usando transistor aumenta o capacitor do laço analógico em um tempo pré-definido, com o objetivo de reduzir o *ripple* do laço analógico na tensão de controle, como verificado na simulação.

Para analisar a potencialidade da arquitetura podem-se testar possibilidades de mudança no projeto. Por exemplo, se houvesse a necessidade de aumentar a corrente do *charge pump* para reduzir a inserção de ruído de fase através desse bloco, pode-se por exemplo, reduzir o K_{VCO} proporcionalmente, o que também reduziria o ruído na tensão de controle. Dessa forma K_D permanece 40 e também $\zeta = 1$, restando apenas corrigir o valor de K_2 para a mudança no K_{VCO} . No caso do projeto proposto, o *tuning range* é de 4x40 MHz ou 40 MHz para cada bit de seleção direta de canal, o que resulta em 0.4 V de variação na tensão de controle. Portanto, se a corrente do *charge pump* fosse dobrada ou quadruplicada e o K_{VCO} reduzido proporcionalmente resultaria em um aumento drástico no capacitor C_1 . Porém, para a arquitetura proposta um fator de 2 significa acrescentar 1 bit ao contador e um fator de 4 acrescentar 2 bits. Atualmente, com a popularização de osciladores digitalmente controlados, esse aumento não seria considerado nem mesmo expressivo, em termos complexidade e tampouco de área.

6.3. Projeto dos Blocos

A implementação do sintetizador proposto usa a estrutura em blocos ilustrada na Fig. 6.4. A arquitetura *dual-path* proposta utiliza dois caminhos para o *loop filter*. Isso flexibiliza o projeto do PLL, desde que a metodologia de projeto, desenvolvida no capítulo 5, seja usada. No fluxo utilizado, cada conjunto de blocos, que representa um bloco em nível de sistema, deve ser projetado de modo a atender a especificação determinada no projeto em nível de sistema. O sintetizador proposto pode ser dividido por quatro grandes conjuntos, o laço analógico, o laço digital, o VCO híbrido junto com o primeiro bloco divisor de frequência e o divisor

programável. Além disso, esses conjuntos são usados para gerar blocos em nível comportamental, calibrados com esses conjuntos de blocos. O projeto nesse nível é utilizado para verificar e otimizar vários quesitos do projeto do sintetizador de frequência, que exigiriam muito recurso computacional, se os blocos fossem simulados em nível de transistor. Após esse passo, mesmo a simulação do PLL em nível de transistor usa uma ferramenta específica para particionar o circuito e reduzir o tempo de simulação. Nas subseções que seguem, são descritos os projetos dos blocos, são dadas informações sobre a calibração dos modelos comportamentais e do leiaute e re-simulação dos blocos. Na próxima subseção, esses dados são utilizados como bloco no projeto do PLL.

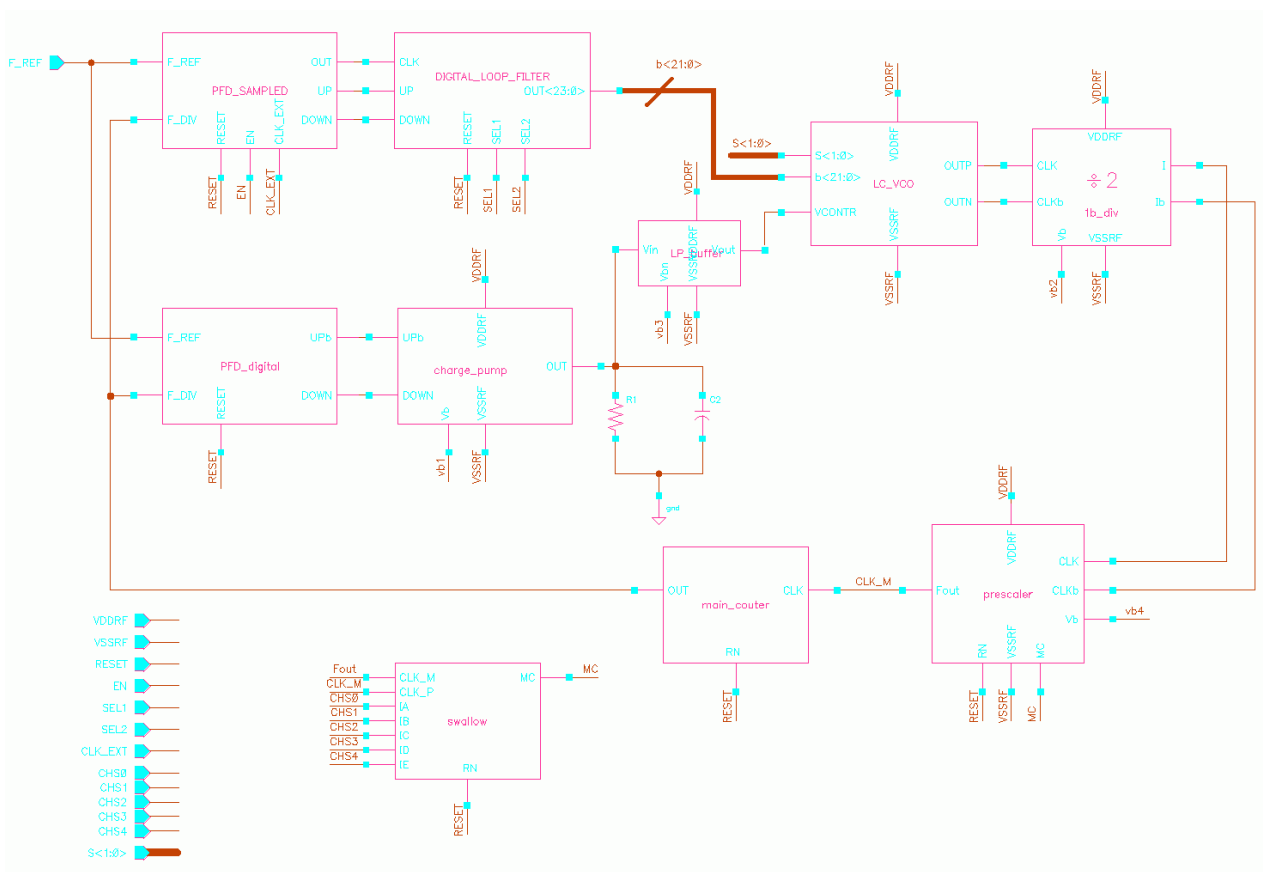


Fig. 6.4. Representação em blocos do esquemático em nível de transistor para o sintetizador, usando a arquitetura *dual-path* proposta.

6.3.1. Projeto do Laço Analógico

O laço analógico é composto por um detector de fase e frequência (PFD) digital, um *charge pump*, os componentes passivos do *loop filter*, que definem o pólo em malha aberta e um *buffer* de *loop filter*. Esses blocos têm características divergentes, em termos de implementação. O PFD é puramente digital, o *charge pump*, apesar de ser chaveado por sinais da saída do PFD, é uma célula analógica, os componentes do filtro devem ser isolados no leiaute e o *buffer* é uma célula analógica que tem a finalidade de isolar o sinal de RF vindo do VCO. Por isso, deve-se tomar cuidados especiais no leiaute, em todas essas células para evitar efeitos de acoplamento. Nas subseções seguintes descreve-se cada célula separadamente e é apresentada a análise da performance do conjunto na geração da tensão de controle do laço analógico.

6.3.1.1. Detector de Fase ou Frequência Digital

A função do detector de fase é produzir um sinal que determine a diferença de fase entre dois sinais. O detector de fase ou frequência digital, que é a topologia escolhida para essa função nessa tese, é um circuito largamente usado nessa aplicação atualmente e opera em conjunto com o bloco *charge pump*. A Fig. 6.5 mostra a representação do PFD usando blocos digitais standard [31]. A estrutura foi adaptada para gerar os sinais necessários para o *charge pump* utilizado, por isso uma técnica usando portas lógicas XOR foi empregada para inverter a saída UP e manter o mesmo atraso da porta na saída DOWN [6]. A redução do efeito *dead zone* foi analisado em simulações e o atraso das portas foi considerado suficiente para os casos estudados.

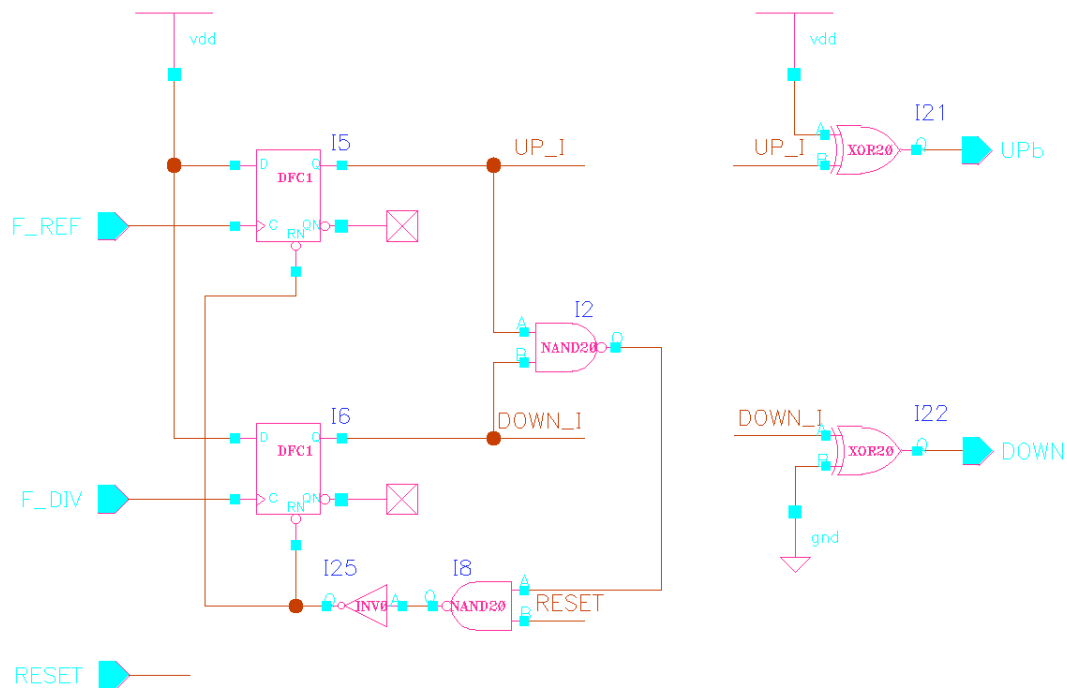


Fig. 6.5. Detector de fase e frequência empregado.

6.3.1.2. Charge Pump

A topologia de *charge pump* empregada foi baseada no circuito ilustrado na Fig. 6.6 [60]. A principal vantagem dessa topologia em relação à convencional é a redução dos *glitches* de corrente nas capacitâncias parasitas. Em um *charge pump* implementado como na Fig. 6.6(a) a carga do *loop filter* é compartilhada com a carga das capacitâncias parasitas de dreno-fonte. Que é o efeito conhecido como *charge-sharing*. No circuito da Fig. 6.6(b), no instante do chaveamento, as capacitâncias são carregadas ou descarregadas pelos transistores M_{10} e M_{13} antes que a corrente de M_{11} e M_{12} circule, resultando na redução esperada. As capacitâncias parasitas e as chaves possuem constantes de tempo bem menores que as do filtro e por isso são mais rápidas. Além disso, os capacitores ligados às portas de M_{11} e M_{12} e as chaves M_{14} e M_{15} ajudam a reduzir os *glitches*. Os capacitores oferecem um caminho CA para o terra, e para as chaves, baixa impedância no desligamento das correntes.

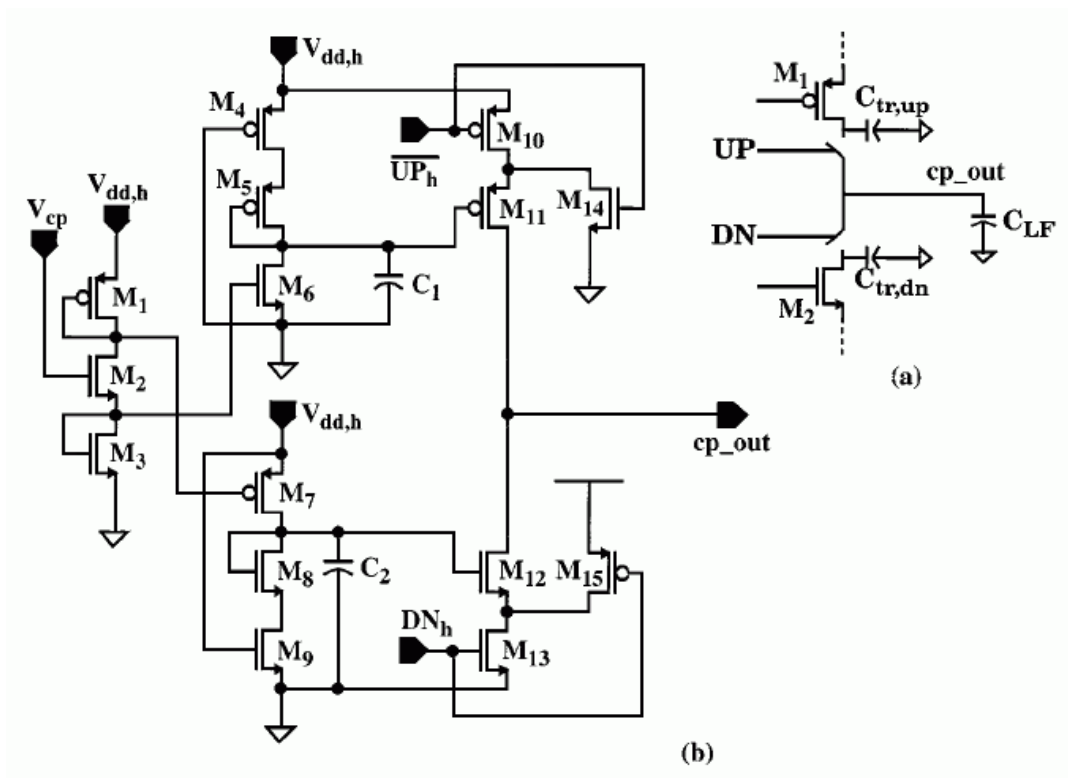


Fig. 6.6. (a) *Charge pump* convencional (b) Topologia de *charge pump* utilizada.

Esta topologia foi proposta para uma alimentação de 3V e teve que ser adaptada para operar em 2V. A topologia proposta para ser empregada no projeto é apresentada na Fig. 6.7. A fonte de corrente usada utiliza a tensão de um *band gap* da biblioteca da AMS. Para reduzir o erro no espelhamento das correntes espelhos de corrente *high-swing* são utilizados em todos os espelhos antes da saída. Essa medida também reduz o descasamento das correntes do *charge pump*. O descasamento das correntes do *charge pump* causa uma injeção dessa diferença de corrente a cada período da frequência de referência. Em regime permanente, quando o PLL estiver estável, esse descasamento se torna um *offset* na fase do sistema. Para compensar o descasamento, a frequência de saída apresenta sinais espúrios com a frequência da referência [53]. Esse efeito é conhecido como *charge-injection*.

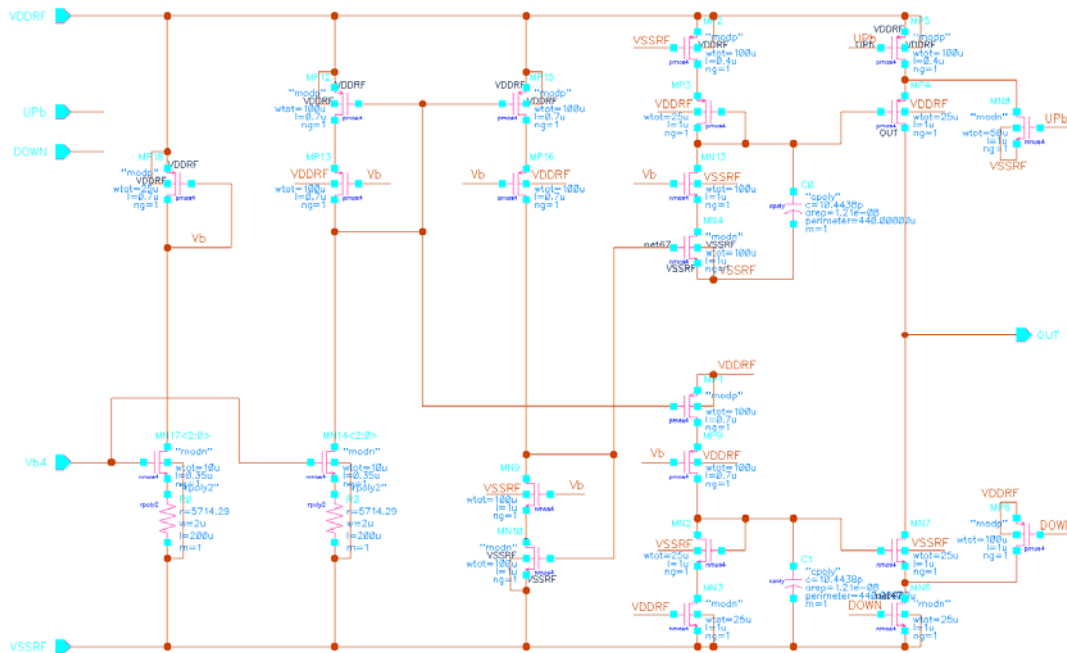


Fig. 6.7. Charge Pump

Simulações de córner foram executadas com o objetivo de prever o comportamento da tensão de saída e medir o *charge-sharing*, para uma comparação do circuito original com o proposto. O resultado das simulações é apresentado na Fig. 6.8. A variação com o córner é sensivelmente reduzida, exceto no córner 6, que combina o pior caso de rapidez (*ws*) com a temperatura de 85° C.

O descasamento das correntes do *charge pump* foi analisado através da média de amostras para UP e DOWN. O resultado foi 69.37uA e 69.39uA.

O ruído do *charge pump* é determinado pela equação a seguir

$$\hat{i}_{cp}^2 = 2\alpha_{cp} \cdot \hat{i}_n^2 \quad (6.1)$$

sendo \hat{i}_{cp} o ruído do *charge pump*, α_{cp} a fração de tempo que as correntes estão ligadas e \hat{i}_n o ruído total nas correntes. O tempo que as chaves ficam ligadas corresponde aos pulsos gerados pelo PFD para eliminar a *dead-zone*. Portanto, existe um compromisso entre eliminar a *dead-zone* e o ruído do *charge pump*.

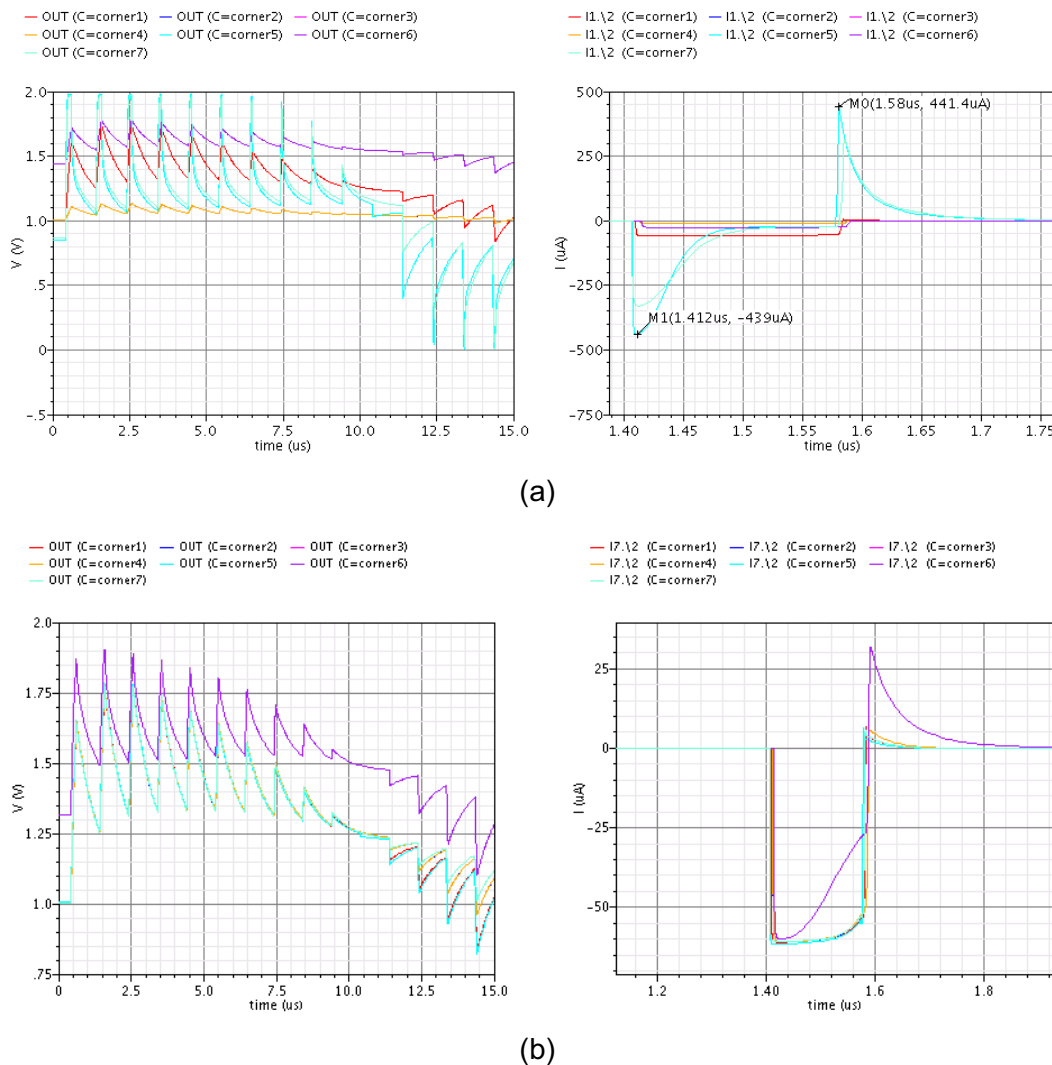


Fig. 6.8. Resultados da simulação de córner para a tensão de saída e *charge-sharing* (a) circuito original (b) topologia proposta.

Através da metodologia usada, os blocos PFD e *charge pump* são calibrados automaticamente em um modelo em Verilog-A com todos os efeitos mencionados, isto é, descasamento das correntes, corrente de *offset*, atrasos do PFD e do *charge pump* e ruído gerado. Com esse modelo é possível realizar simulações de transiente do PLL e determinar a influência dos blocos. O código em Verilog-A, do conjunto PFD e *charge pump* está no Apêndice B1.

6.3.2. Projeto do VCO e Blocos Divisores de Alta Frequência

O VCO é um dos gargalos do projeto, sendo um dos blocos mais importantes para o bom funcionamento do mesmo. O projeto do VCO define a frequência de saída, a faixa de frequência de operação do PLL, o K_{VCO} e uma parte do ruído de fase da saída. Além disso, o VCO e o *prescaler* são os estágios de maior frequência do PLL, e por isso, muitos dos pontos críticos, em termos de projeto de bloco, estão nessa junção.

A frequência especificada para atender aos padrões da proposta *multi-standard*, na banda de 900 MHz, é baixa para um VCO LC tanque. Normalmente, os indutores tornam o VCO muito grande para determinadas frequências e a solução adotada é dobrar a frequência e depois dividi-la por 2. Além disso, o divisor por 2 proporciona isolamento entre a frequência de saída e a frequência de saída do VCO, o que evita efeitos indesejáveis de performance, como *injection locking* ou *injection pulling* [31]. Nesse projeto não será utilizado *buffer* de RF, por ser considerado fora do escopo.

6.3.2.1. VCO Híbrido

Em princípio, foi pretendido que o projeto do VCO se baseasse em duas topologias encontradas na literatura [61] [62]. Atualmente, pode-se encarar o VCO como uma adaptação da topologia de VCO com técnica de filtragem, com dimensão mais compacta. O circuito do VCO é apresentado na Fig. 6.9. Em [62] foi verificado que a segunda harmônica do ruído gerado pelo circuito de polarização pode ser filtrada no transistor de cauda, esse ruído contribui significativamente para o ruído de fase total do VCO.

Antes de se definir a topologia escolhida, foi cogitada uma topologia baseada em transistor de cauda NMOS, devido ao fato que isso maximiza o *tuning range* do VCO (devido ao maior g_m). Uma desvantagem dessa abordagem é que o transistor NMOS gera mais ruído $1/f$, e apesar disso, o *tuning range* tem prioridade no projeto. Porém, o que definitivamente decidiu que o projeto do VCO usaria uma topologia baseada em um transistor de cauda PMOS foi o acoplamento DC

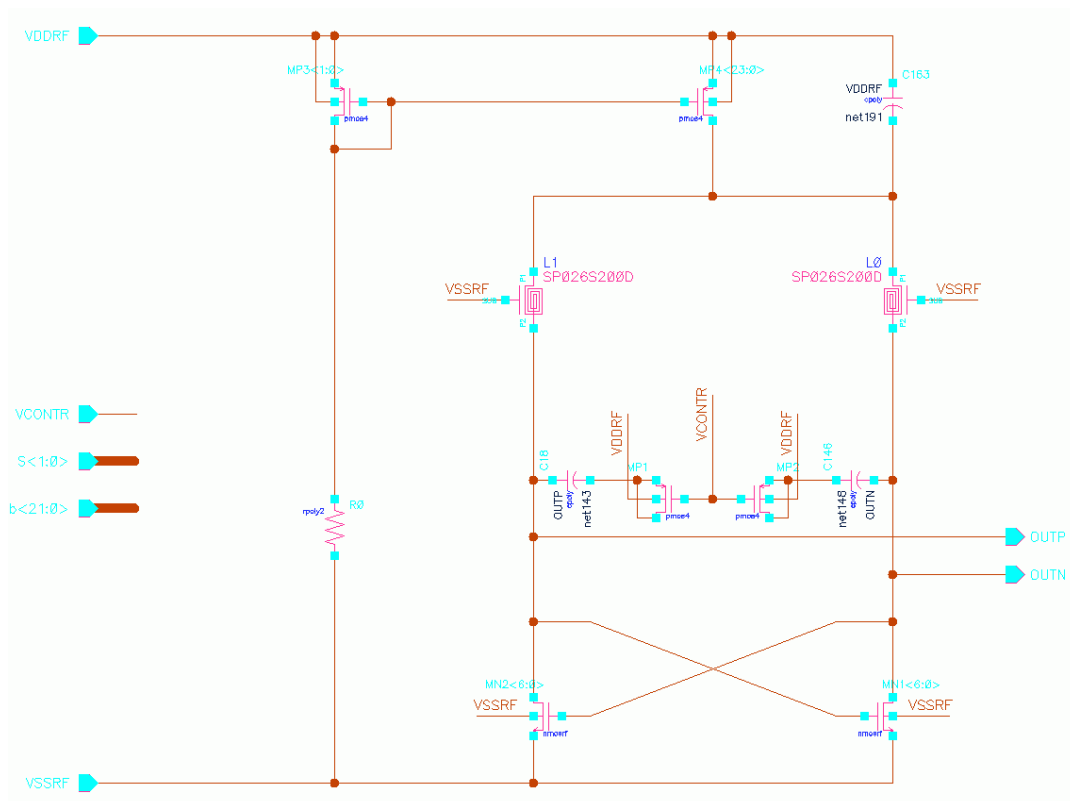


Fig. 6.9. Topologia de VCO proposto para implementação.

entre o VCO e o primeiro divisor de frequência. Essa característica será analisada, mais adiante, nessa seção.

Na metodologia de projeto empregada, o VCO é dividido em duas partes: O tanque – que é a maior parte da rede RLC; e o par diferencial cruzado – que é representado por uma resistência negativa que garante a oscilação.

As seguintes características foram consideradas no projeto do VCO:

- $\omega = \frac{1}{\sqrt{L_p C}}$ e $R_p = Q \times \omega_p \times L_p$, sendo que C é a capacitância equivalente consistindo principalmente da capacitância dos varactores.
- O critério para oscilação (Barkhausen) desse oscilador é que a resistência negativa compense R_p (a perda do tanque) portanto o ganho do par cruzado deve ser tal que $R_p > 1 / g_m$.

- O ruído de fase depende fortemente da relação sinal ruído (SNR) entre a amplitude da tensão de saída e do ruído. A equação 3.12 foi usada para prever o ruído.

A performance do VCO é governada principalmente pelos componentes passivos. No VCO proposto utiliza-se indutores, capacitores e MOSFET varactores. Esses componentes serão descritos a seguir.

O indutor usado no VCO é da biblioteca de indutores de 4 metais da AMS tecnologia c35b4c3. A Tabela 2 mostra os principais parâmetros dos indutores, sendo que a frequência de ressonância de todos eles é superior a 6 GHz. O Indutor usado está sinalizado em vermelho, donde se pode verificar que $Q \leq 4.7$. Um fator de qualidade baixo é uma desvantagem, pois degrada o ruído de fase, porém, pretende-se que o PLL tenha uma faixa de operação de frequência larga e, nesse caso, um fator de qualidade alto poderia dificultar o projeto.

Tabela II

Indutor	Indutancia [nH]			Qmax	Q	
	L _s	@2.4GHz	@5.0GHz		@2.4GHz	@5.0GHz
SP014S300D	1.37	1.34	1.38	6.8@3.8GHz	6.1	6.2
SP020S180D	2.01	1.99	2.04	5.8@5.0GHz	4.5	5.7
SP026S200D	2.60	2.64	2.83	5.7@3.8GHz	4.7	5.3
SP037S180D	3.77	3.87	4.30	5.1@4.3GHz	3.9	4.9
SP047S180D	4.78	5.01	5.81	4.8@3.8GHz	4.1	4.1
SP050S155D	5.52	5.62	6.47	4.4@3.8GHz	3.4	4.1
SP090S155D	9.15	9.98	12.7	3.5@3.3GHz	3.3	2.5

Foram usados dois tipos de MOSFET varactores:

- *Accumulation/Depletion only varactor*
- *Inversion only varactor* (com o substrato no V_{DD}).

Basicamente, o VCO possui três formas de controlar a frequência através de varactores. A tensão de controle, a seleção direta dos canais e os bits do *loop filter* digital. No gráfico da Fig. 6.10 é mostrada a curva da capacitância do transistor em

função da tensão. A vantagem de usar *inversion only varactor* é que a capacitância só depende de V_{DD} e da tensão da porta. Usando *accumulation/depletion only varactor* a variação da capacitância é maior. Ambos possuem vantagens, e por isso, a escolha do tipo de MOSFET varactor dependeu do circuito ao qual se aplicaria melhor. Nas Fig. 6.11 à Fig. 6.13 são mostrados os bancos de capacitores usados do projeto do VCO. Nos varactores que estão em série com um capacitor foi usado *inversion only* e nos varactores ligados à tensão de saída foram usados *accumulation/depletion only*.

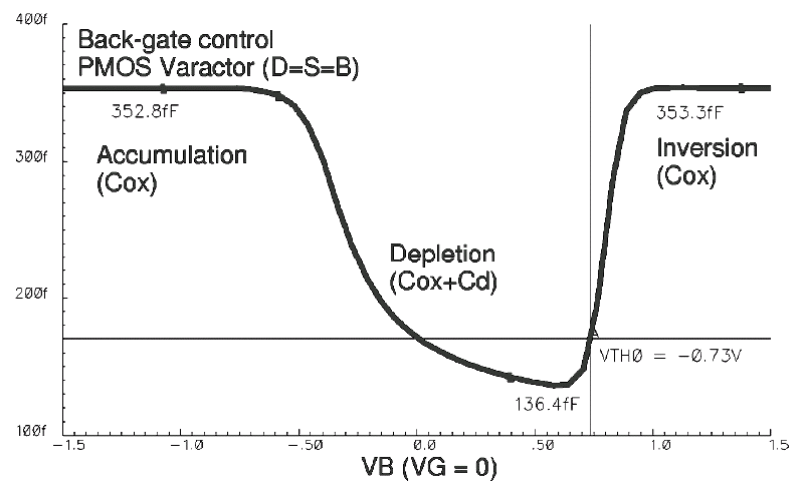


Fig. 6.10. Curva da capacitância de porta do transistor em função da tensão de substrato.

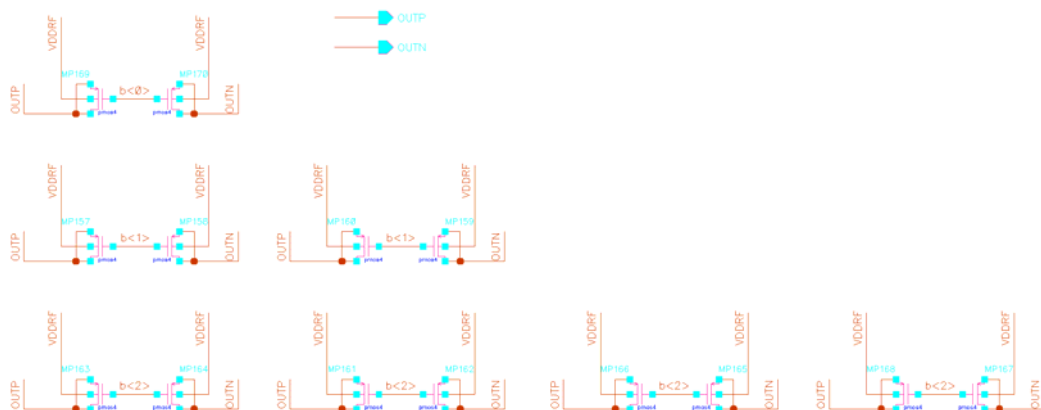


Fig. 6.11. Três bits da matriz de capacitores controlada pelo *loop filter* digital

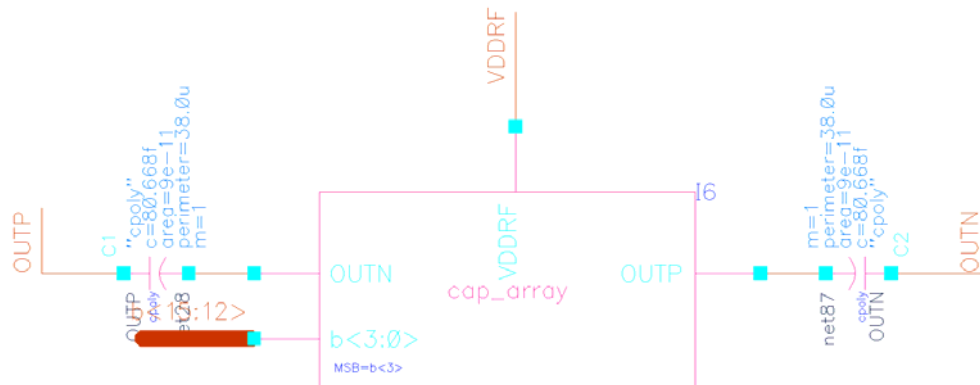


Fig. 6.12. Um dos blocos usados na matriz de capacitores contendo quatro bits e um capacitor em série.

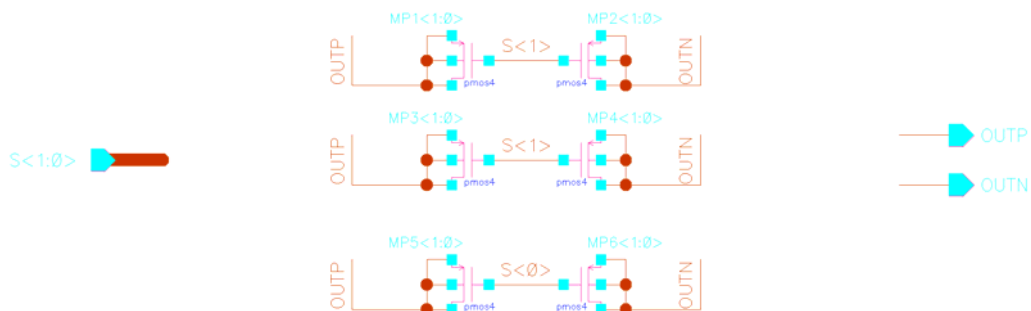


Fig. 6.13. MOSFETs usados na seleção direta de canal.

O projeto do VCO híbrido tem como especificação um K_{VCO} de 200 MHz/V, *tuning range* de 4X80 MHz, com seleção direta de canais e uma frequência de centro de 1824 MHz. Isso possibilita que as especificações previstas anteriormente para a banda de 900 MHz sejam alcançadas na saída. A saída do PLL é obtida por um bloco que divide a frequência de saída do VCO por 2. A Fig. 6.15 mostra as curvas para o K_{VCO} projetado e a Fig. 6.16 o ruído de fase na frequência central. A tabela III apresenta as dimensões do projeto.

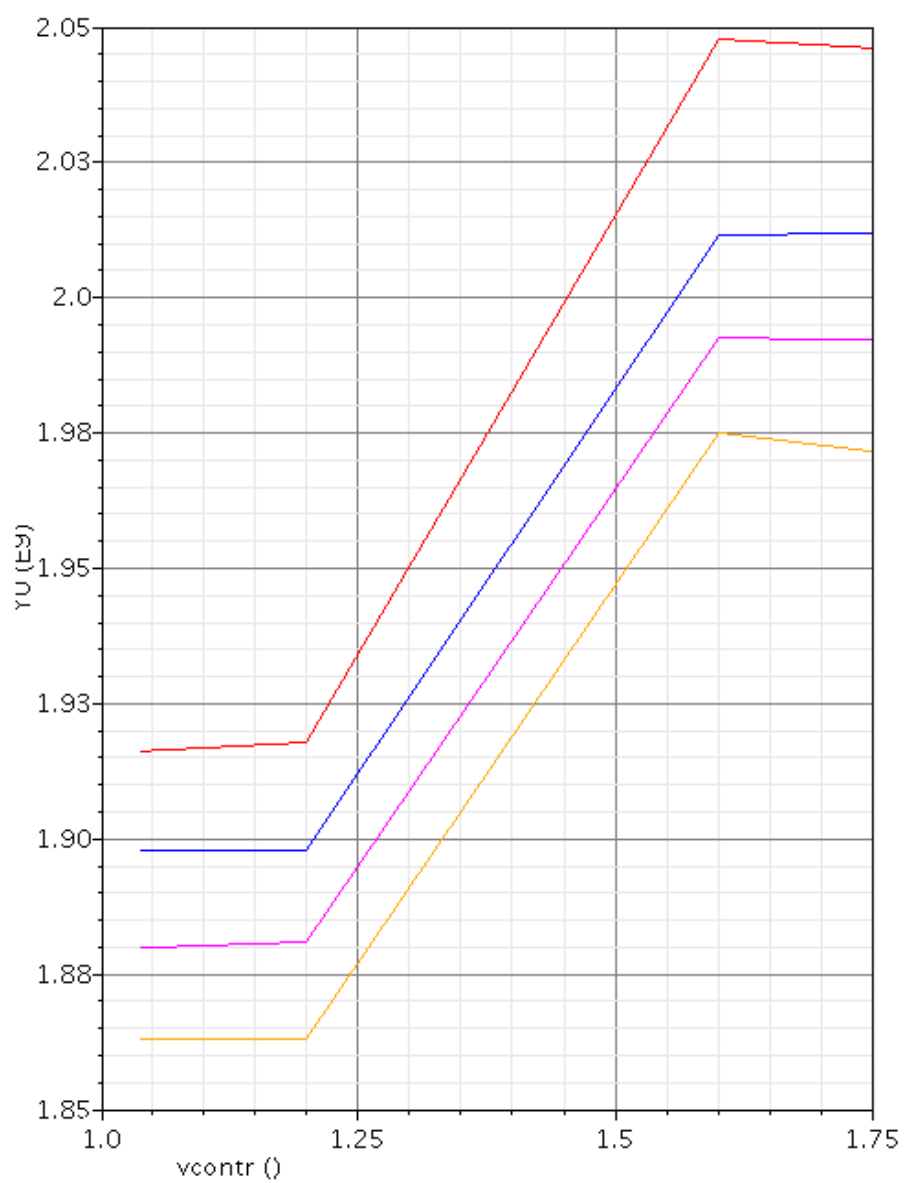


Fig. 6.14. Simulação paramétrica da frequência do oscilador em função da tensão de controle. As curvas representam as quatro curvas possíveis usando seleção direta de canal.

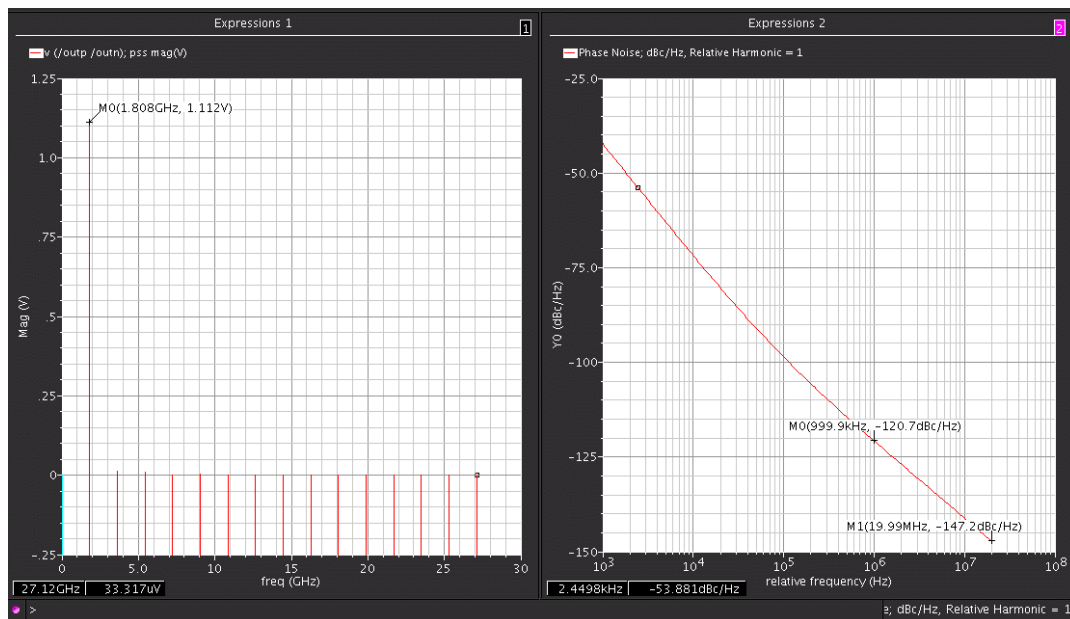


Fig. 6.15. Ruído de fase do VCO na frequência central

Tabela III

Componente	Dimensão
M ₁ , M ₂	70/0,35 (μm/μm)
M ₃	50/0,5 (μm/μm)
M ₄	600/0,5 (μm/μm)
R ₁	4kΩ
C ₁	6,8pF

O leiaute do VCO híbrido foi dividido em vários blocos para possibilitar que a extração de cada bloco individual seja testada em separado e, dessa forma, permitindo identificar quais os blocos têm sua operação alterada devido à adição de parasitas. O esquemático do VCO usado no leiaute é apresentado na Fig. 6.16. Esse método facilitou o processo de re-simulação da extração de parasitas do leiaute do VCO.

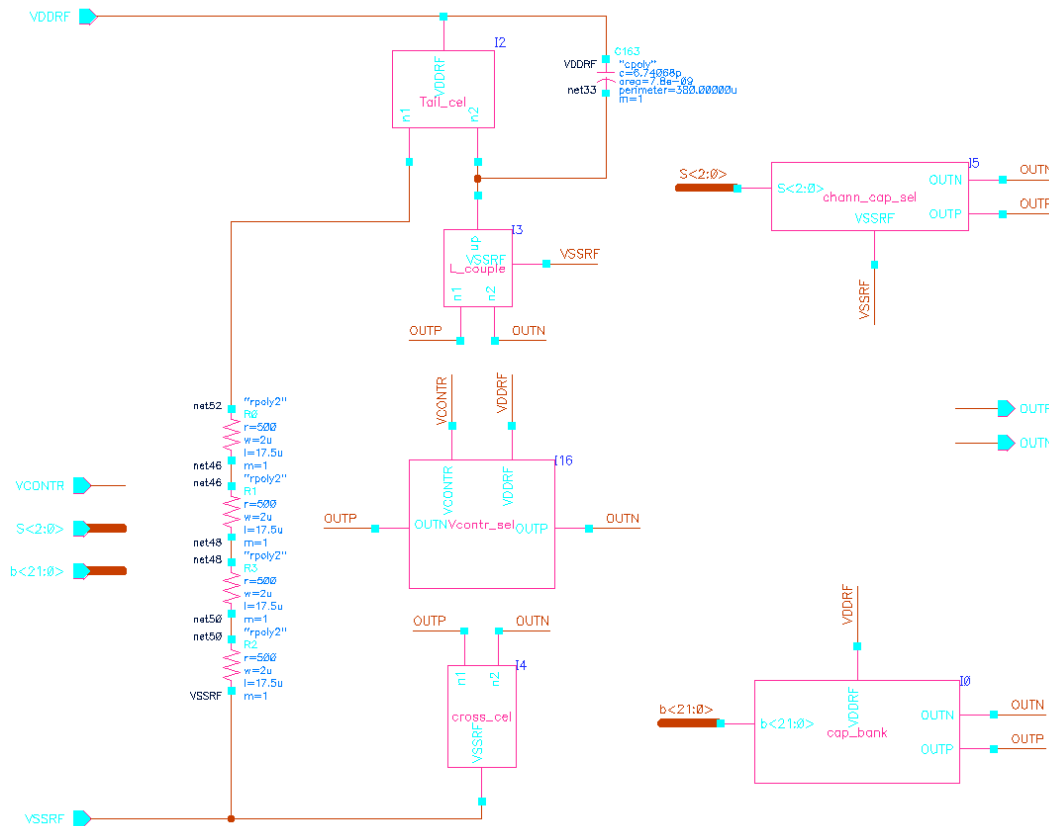


Fig. 6.16. Esquemático do VCO modificado em forma modular para facilitar o processo de simulação pós-leiaute

Após o processo de re-projeto do VCO híbrido, o VCO atingiu as especificações desejadas, com algumas alterações. O leiaute foi reeditado para reduzir capacitâncias e resistências parasitas em pontos verificados e o capacitor em paralelo com o transistor de cauda foi ajustado para atingir a frequência desejada. Dentre pequenas alterações, a alteração mais drástica ocorrida foi a mudança de topologia do circuito do banco de capacitores que provê a seleção direta de canal e, por isso, o circuito da Fig. 6.13 foi alterado para o da Fig. 6.17. Esse circuito também é muito encontrado na literatura, para essa função, e foi crucial para a operação do VCO proposto.

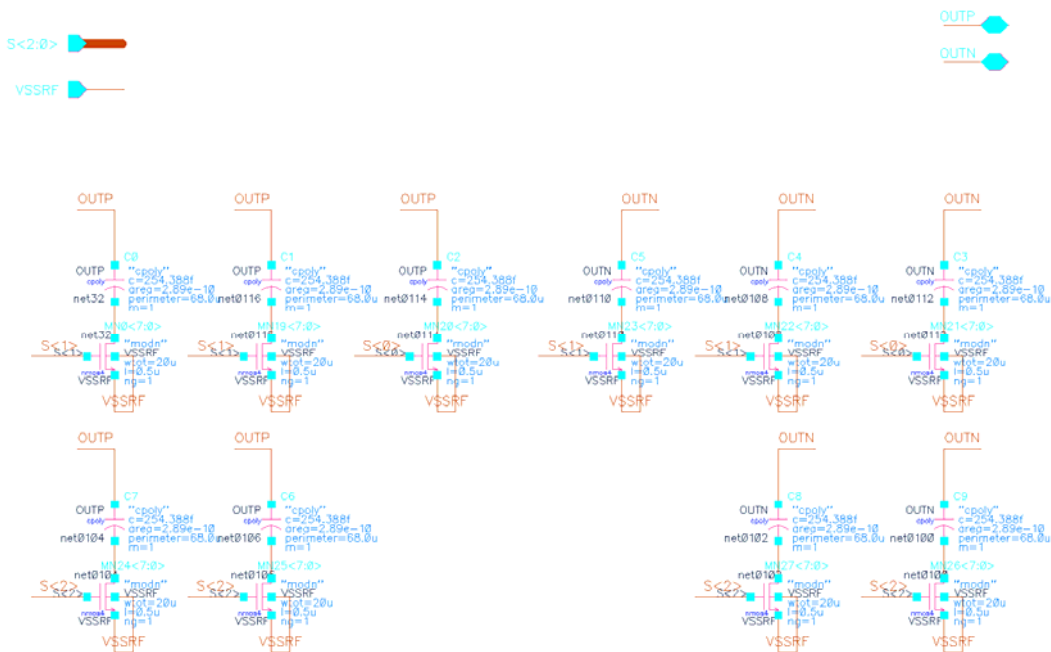


Fig. 6.17. Circuito para seleção direta de canal, proposto durante o projeto pós-leiute.

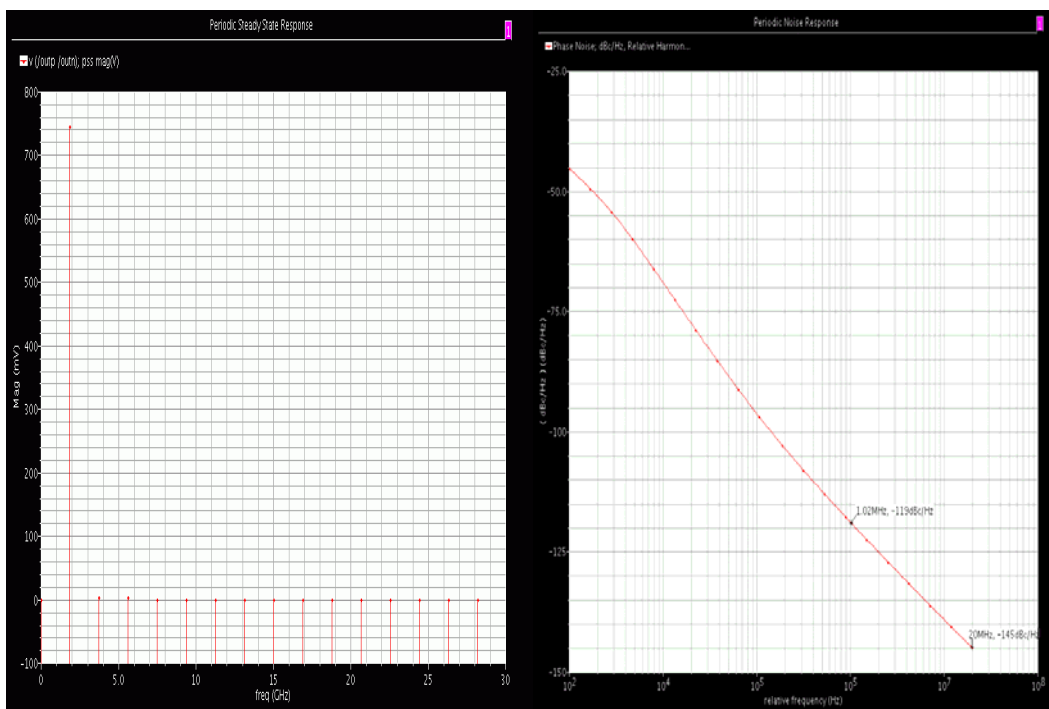


Fig. 6.18. Ruído de fase do VCO na frequência central, após a extração de parasitas

O resultado de simulação de ruído de fase do VCO após a extração de parasitas, para a frequência central, está mostrado na Fig. 6.18. O leiaute do VCO é ilustrado na Fig. 6.19, sendo que as dimensões x e y desse leiaute são $632 \times 543,7$ ($\mu\text{m} \times \mu\text{m}$). O espaço vazio dentro dessa área deverá ser preenchido com os demais blocos do sintetizador.

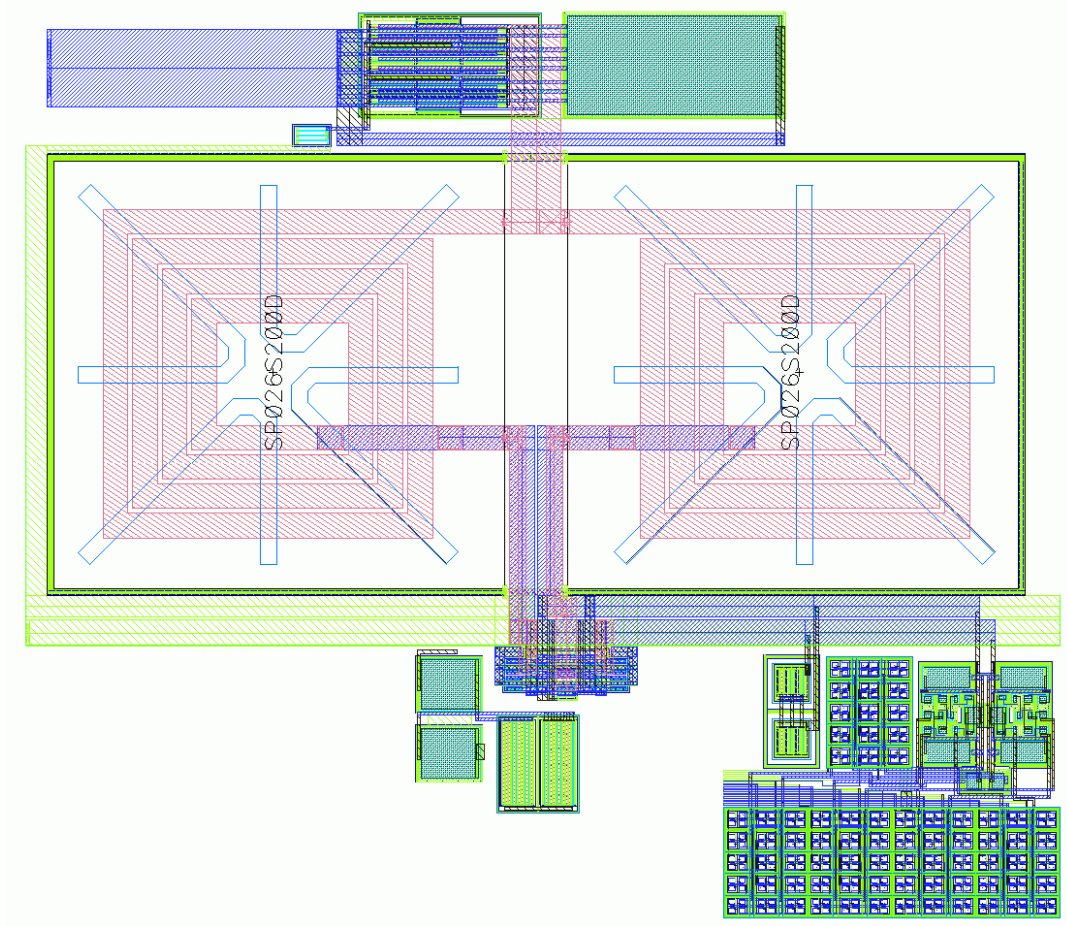


Fig. 6.19. Leiaute do VCO híbrido proposto contendo indutores, transistor de cauda, resistor, capacitor de filtro, MOSFET varactor da tensão de controle, banco de capacitores, transistores cruzados e roteamento; As dimensões x e y desse leiaute são $632 \times 543,7$ ($\mu\text{m} \times \mu\text{m}$).

6.3.2.2. Primeiro Estágio Divisor de Frequência

A frequência de saída do PLL é obtida por um circuito divisor por 2, ligado a saída do VCO, que é o primeiro estágio divisor de frequências do sintetizador. Como já foi comentado, esse estágio é crucial para o PLL, pois fornece a frequência de saída. Circuitos lógicos que operam com mais de duas centenas de MHz devem utilizar algum tipo de lógica dinâmica, caso contrário, uma porta lógica CMOS padrão pode nem mesmo operar com constantes de tempo tão baixas. Existem circuitos adequados para tal função, os quais obtêm atrasos de porta bem baixos e processam o sinal analogicamente, ou seja, os níveis lógicos são dados diferencialmente. O circuito divisor utilizado emprega SCL (*Source Coupled Logic*) também conhecido como CML (*Current Mode Logic*).

CML/SCL reduz o ruído de chaveamento [63] que é intrínseco nos circuitos CMOS padrão. Os circuitos CML são geralmente diferenciais e possuem corrente quiescente constante para reduzir o transitório causado pelo chaveamento. A Fig. 6.20 mostra que esse tipo de circuito opera por um direcionamento da corrente controlada pela tensão do par diferencial.

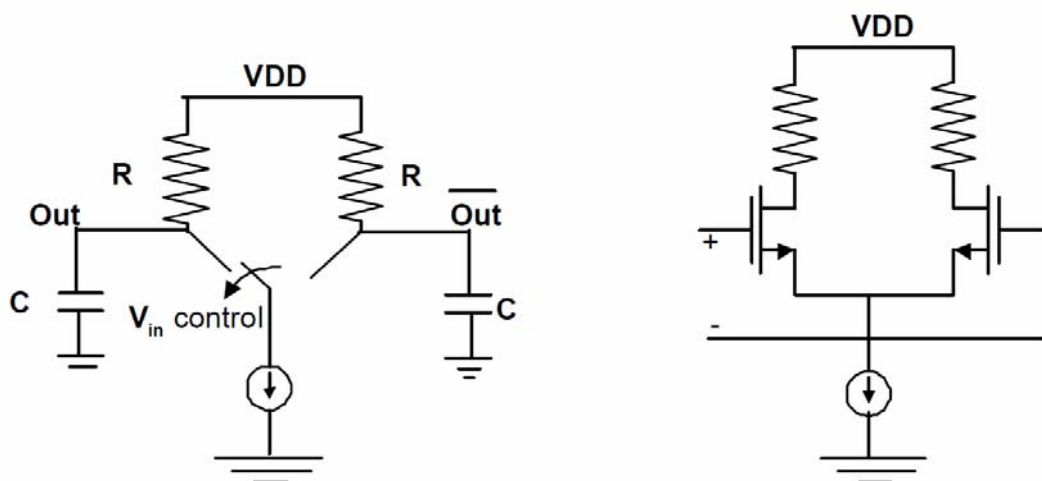


Fig. 6.20. Representação de circuitos CML (a) usando chaves (b) usando um circuito simplificado.

Diversas comparações entre circuitos CMOS padrão e CML foram realizadas na literatura [64]. Supondo-se N portas lógicas idênticas em cascata, todas com uma capacitância de carga C em cada nó de saída, o atraso de propagação total (D) do conjunto de portas será proporcional a:

$$D_{CMOS} = \frac{N \times C \times V_{dd}}{0.5k \times (V_{dd} - V_t)^\alpha} \quad (6.2)$$

onde k e α são parâmetros dependentes das dimensões dos transistores e do processo. Assumindo que a frequência de *clock* de circuitos CMOS padrão é igual ao inverso do atraso de propagação, a dissipação dinâmica da potência (*dynamic power dissipation*), e os produtos *power-delay* (PD) e *energy-delay* (ED), são dados por:

$$P_{CMOS} = N \times C \times V_{dd}^2 \times \frac{1}{D_{CMOS}} \quad (6.3)$$

$$PD_{CMOS} = N \times C \times V_{dd}^2 \quad (6.4)$$

$$ED_{CMOS} = N^2 \times 2 \times \frac{C^2 \times V_{dd}^2}{k \times (V_{dd} - V_t)^\alpha} \quad (6.5)$$

De um modo geral, o objetivo de um projeto digital é de otimizar o produto (ED). Pode-se deduzir que a tensão de fonte ótima para minimizar o produto ED para o CMOS é

$$V_{dd} = \frac{2V_t}{3 - \alpha} \quad (6.6)$$

Já as equações para circuitos inversores em cascata usando lógica CML/SCL são dadas por

$$D_{CML} = NRC = \frac{N \times C \times \Delta V}{I} \quad (6.7)$$

$$P_{CML} = N \times I \times V_{dd} \quad (6.8)$$

$$PD_{CML} = NIV_{dd} \times \frac{NC\Delta V}{I} = N^2 \times C \times \Delta V \times V_{dd} \quad (6.9)$$

$$ED_{CML} = N^2 CV_{dd} (\Delta V) \times \frac{NC\Delta V}{I} = \frac{N^3 C^2 V_{dd} \Delta V^2}{I} \quad (6.10)$$

sendo ΔV a variação de saída.

Os resultados acima indicam que os circuitos usando CML podem ser otimizados reduzindo-se a tensão de alimentação ou a excursão do sinal de tensão, e aumentando a corrente da cauda.

As velocidades mais elevadas atingidas pelos circuitos CML podem ser atribuídas a dois aspectos principais: 1) os transistores não necessitam ser completamente desligados, como no caso das células digitais CMOS *padrão*, e 2) a excursão mais baixa na tensão pode carregar/descarregar a capacitância equivalente do nó de saída mais rapidamente. Em contrapartida, a dissipação estática baixa de circuitos digitais CMOS convencionais não se mantém, bem como, a dissipação dinâmica pode também ser maior devido às frequências altas de operação desses circuitos.

A topologia mostrada na Fig. 6.21 tem sido bastante utilizada em circuitos divisores de alta frequência [65]. O *latch* usando lógica CML está ilustrado na Fig. 6.22 [63]. A corrente do transistor de cauda é de 400 μA .

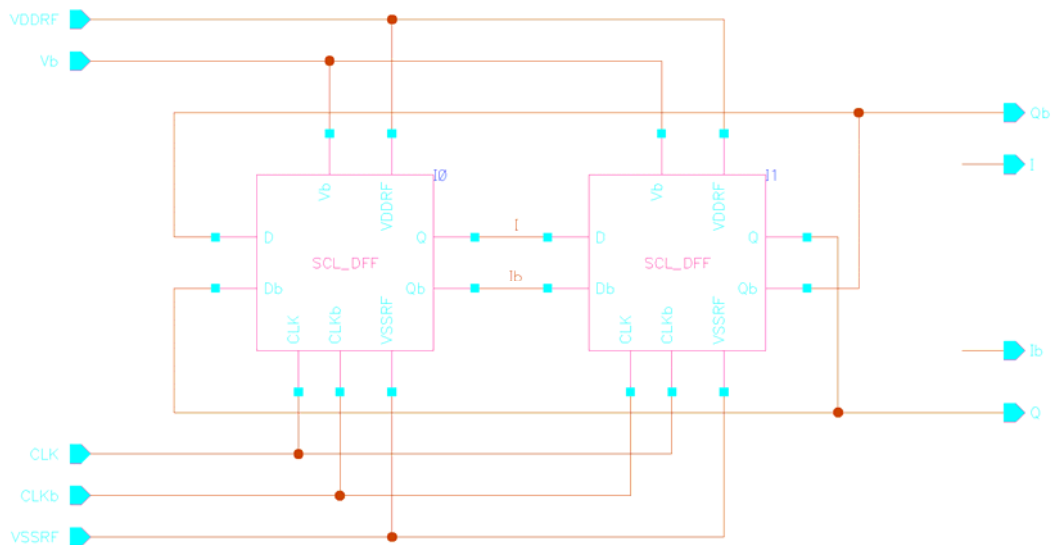


Fig. 6.21. Circuito que divide a frequência por dois usando *latches* tipo D de alta frequência.

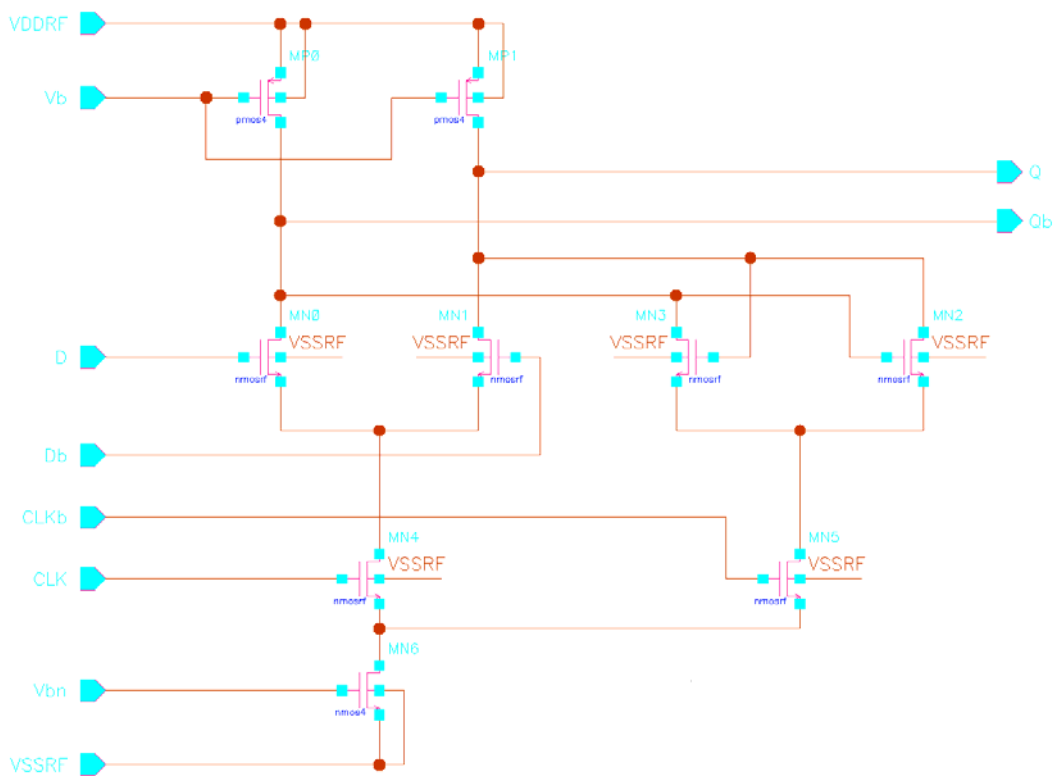
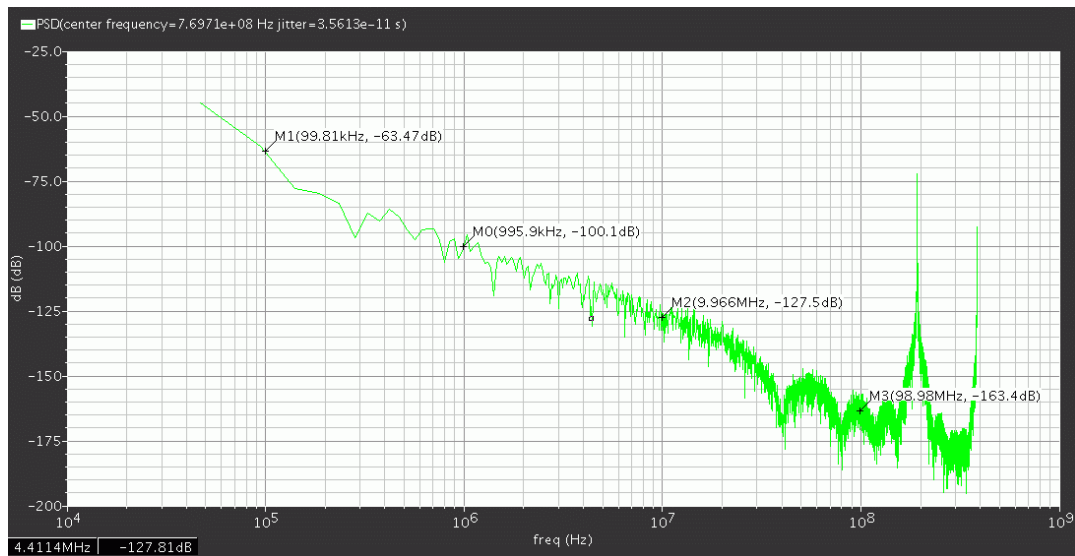
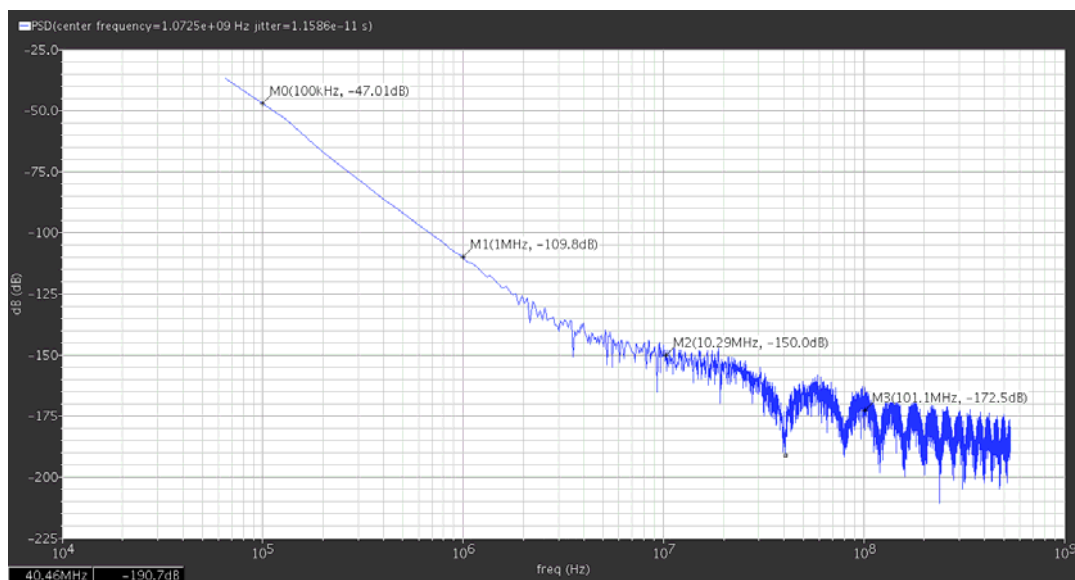


Fig. 6.22. *Latch* tipo D digital usado no circuito divisor por dois.

Verificou-se que o acoplamento CC entre os circuitos das Fig. 6.9 e Fig. 6.22 influi no ruído de fase na saída do divisor. Algumas topologias foram testadas e o circuito da Fig. 6.19 alcançou o maior SNR simulado. O ruído de fase do conjunto foi simulado usando simulação transiente, pois simulações de regime permanente não convergiram até que o acoplamento adequado dos blocos foi obtido.



(a)



(b)

Fig. 6.23. Ruído de fase da saída do divisor simulado usando simulação de transiente com (a) para o melhor resultado alcançado usando o VCO com transistor de cauda NMOS (b) para o VCO usando transistor de cauda PMOS e o divisor da Fig. 6.9.

Simulações comparativas com tempo total de $5 \mu\text{s}$ foram usadas, e no final do projeto, com tempo de simulação de $100 \mu\text{s}$ foi realizada, quando também uma simulação de regime permanente pôde ser realizada. O resultado da simulação de ruído transiente, obtida através desse método, é apresentado na Fig. 6.23.

O procedimento para calibração do conjunto formado pelo VCO e *prescaler* consiste de dois passos principais, a preparação do circuito de teste, contendo o VCO e o primeiro bloco divisor, e a simulação para geração do macromodelo usado nas simulações do PLL. O circuito utilizado está mostrado na Fig. 6.24, sendo que o circuito foi preparado para permitir que simulações de injeção de sinal sejam realizadas posteriormente. O resultado da simulação de regime permanente e ruído, usado no processo de extração dos circuitos em nível de transistor, é mostrado na Fig. 6.25. O resultado gera uma tabela de valores que é automaticamente convertido em um macromodelo pela ferramenta. Os resultados de simulação de transiente alcançaram, em termos de ruído, resultados compatíveis aos encontrados com o circuito em nível de transistor.

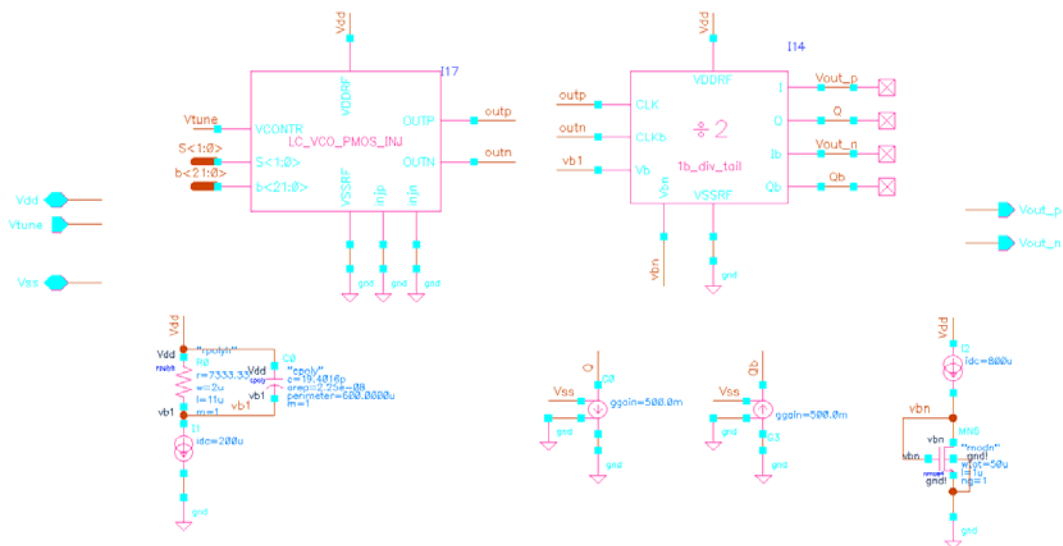


Fig. 6.24. Circuito utilizado no processo de extração para gerar o macromodelo do conjunto VCO e prescaler.

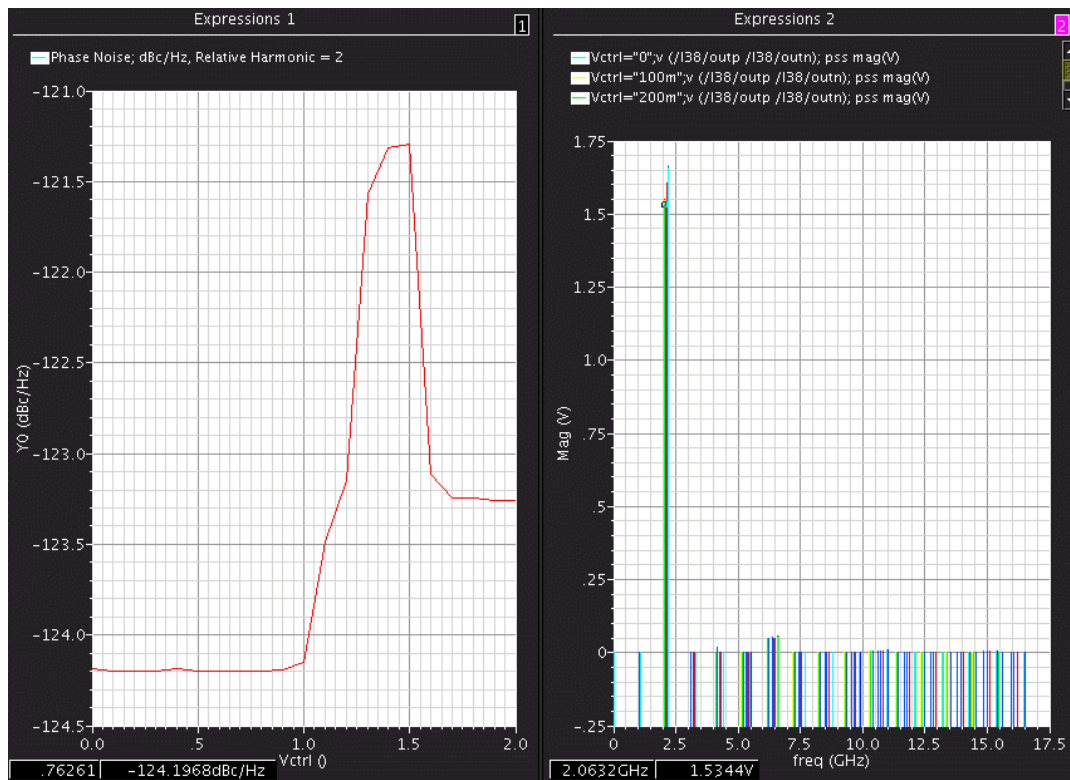


Fig. 6.25. Resultado da extração do circuito para uma variação da tensão de controle de 0 a 2V com passos de 0.1V (a) ruído de fase do conjunto para um *offset* de 1MHz (b) amplitude das 15 harmônicas simuladas.

6.3.3. Projeto do Divisor Programável

A arquitetura de PLL *Integer-N* necessita de muito menos complexidade em relação ao projeto de sistemas PLL *Fractional-N*, essa complexidade é devido ao uso de divisão fracionária. Podem-se comparar os divisores programáveis para os dois tipos de arquiteturas, observando que os estágios divisores são iguais (*prescaler & main counter*) nos dois tipos de circuito, mas, como se verá mais adiante, o *pulse swallow* é bem mais simples do que qualquer técnica usada para fracionar a frequência, normalmente empregada nos PLLs do tipo *Fractional-N*. Esses circuitos são necessários para lidar com os sinais espúrios gerados no processo e além de mais complexos ocupam uma área relativamente grande. Entretanto, comparado com o *Fractional-N* esse tipo de projeto tem desvantagens na

supressão de ruído, por ser baseado nos múltiplos da frequência de referência, além de comprometer a resolução das frequências dos canais.

A técnica mais utilizada para esse fim tem sido *pulse swallow*, devido a sua simplicidade e eficiência. A topologia utilizada de divisor programável está mostrada na Fig. 6.26 [66]. Apesar de conter blocos com pontos críticos de projeto, a operação do *pulse swallow* é relativamente simples. Existem dois contadores decrescentes, *main counter*, com razão de divisão fixa de P e *swallow counter* com razão de divisão S controlada externamente, sendo que $S < P$. O *prescaler* é um divisor de frequência com razão $M / M+1$, correspondente ao sinal MC (0/1) que é a saída do *swallow counter*. No início da contagem o prescaler está dividindo com razão $M+1$ e portanto o *clock* do *main counter* é $f_{in}/(M+1)$. Quando a contagem do *swallow counter* chega a zero, MC assume o valor 0 e o *main counter* passa a dividir f_{in}/M até que o *main counter* atinja zero e o sistema se reinicialize. Como resultado, o *prescaler* divide a frequência de entrada S vezes por $(M+1)$ e $(P-S)$ vezes por M e, portanto, a razão de divisão total será $N = S(M+1) + (P-S)M = S + MP$.

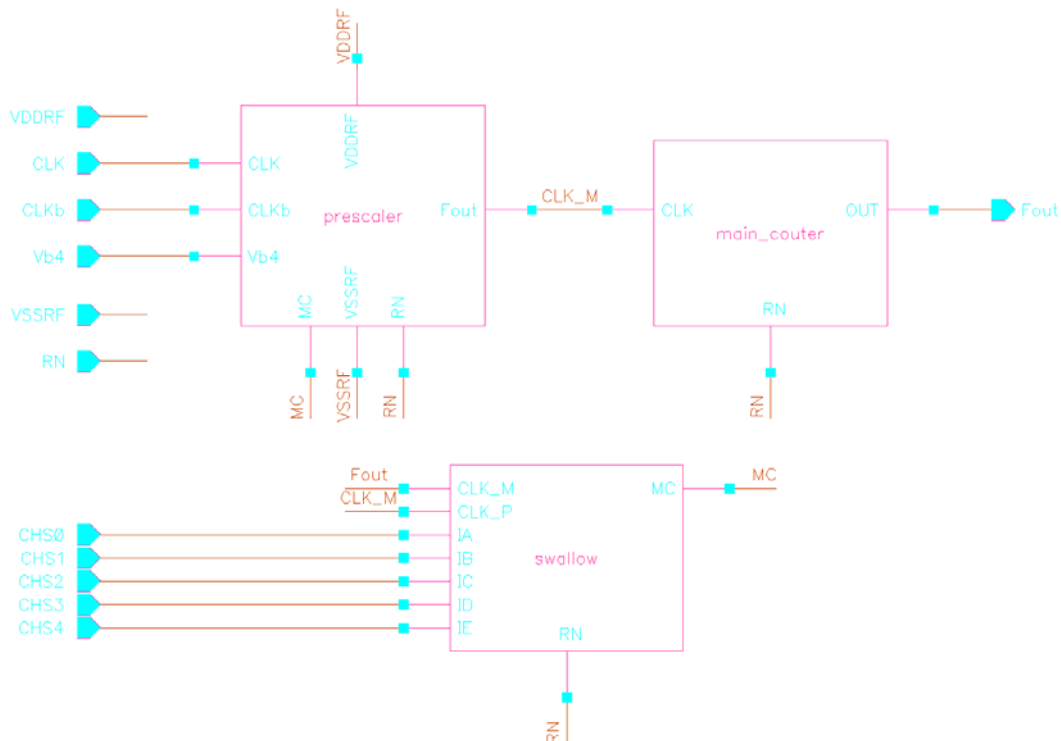


Fig. 6.26. Esquemático do Divisor Programável usando a técnica *Pulse Swallow*.

No projeto do divisor programável dessa tese foi utilizado $M = 32$, $P = 30$ e o S de 0 a 32. Nas subseções seguintes, é descrito o projeto de cada bloco usado no divisor programável.

6.3.3.1. Main Counter e Swallow Counter

Os dois contadores têm os projetos semelhantes, são ambos de 5 bits e decrescentes. As Fig. 6.27 e Fig. 6.28 mostram os circuitos dos contadores usando células digitais padrão. Os contadores possuem *reset* assíncrono e uma lógica recorrente para contadores baseados em flip-flop tipo D, que possui um *reset* síncrono.

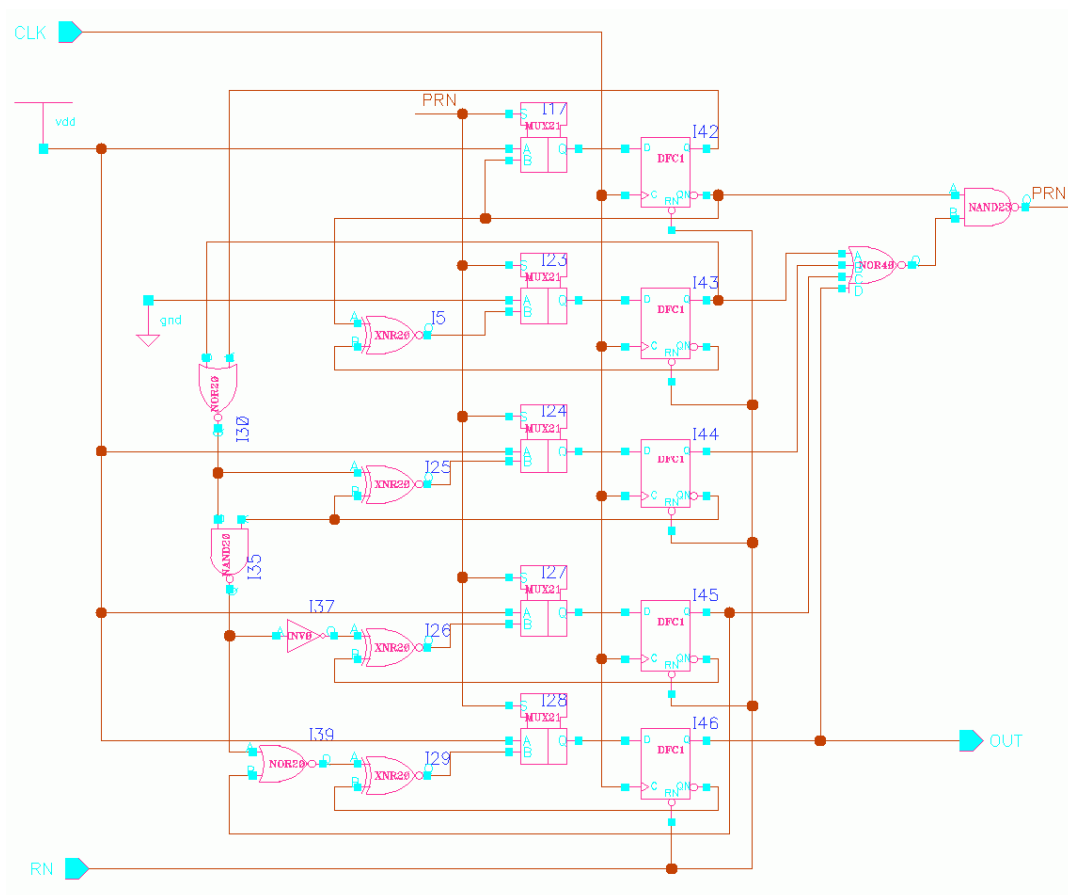


Fig. 6.27. Main Counter.

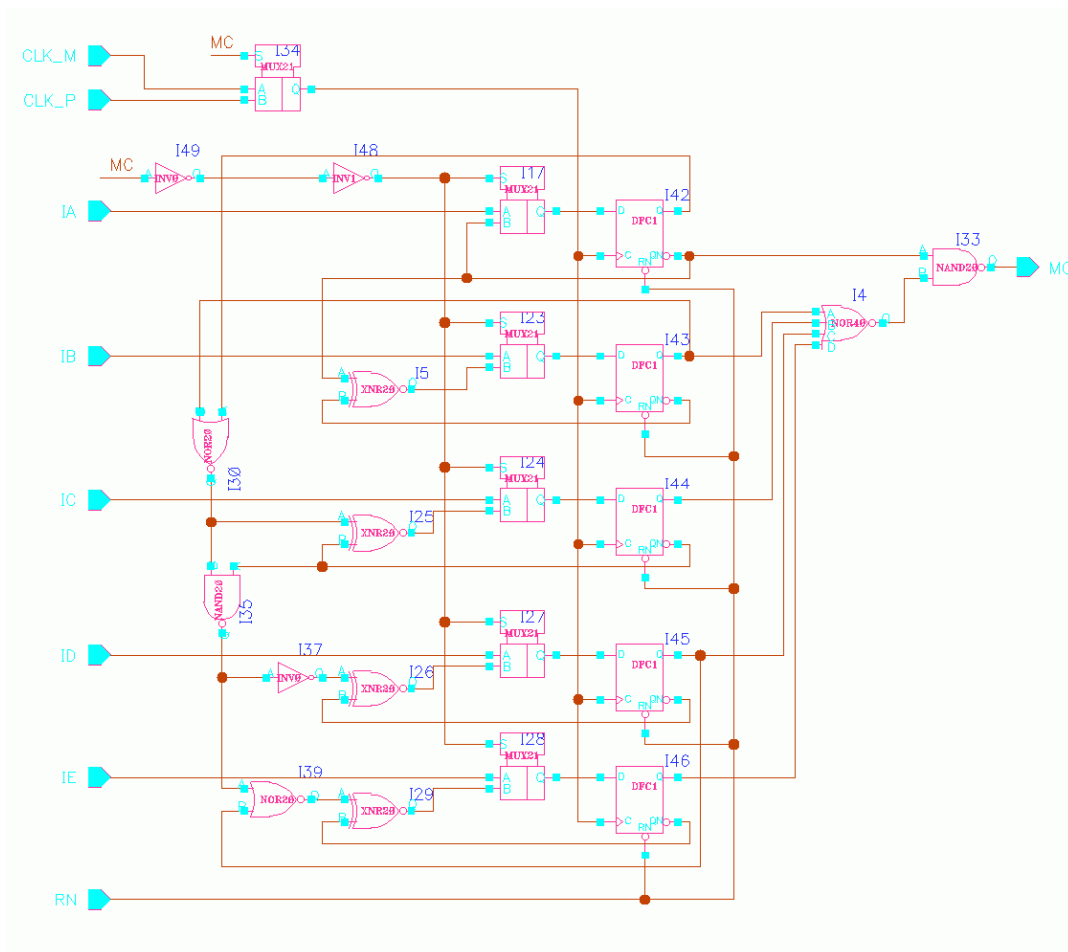


Fig. 6.28. Swallow Counter.

6.3.3.2. Divisor de Frequência Usado no Prescaler

O circuito da Fig. 6.29 é usado no *prescaler* para divisão de frequência de modo assíncrono. O divisor por dois utiliza a configuração da Fig. 6.21, assim como explicado anteriormente. O circuito de polarização, usando uma fonte de corrente baseada em *band gap*, está mostrado na Fig. 6.30. Os pontos principais verificados no projeto desse bloco foram a distorção harmônica e erros por variação de processo.

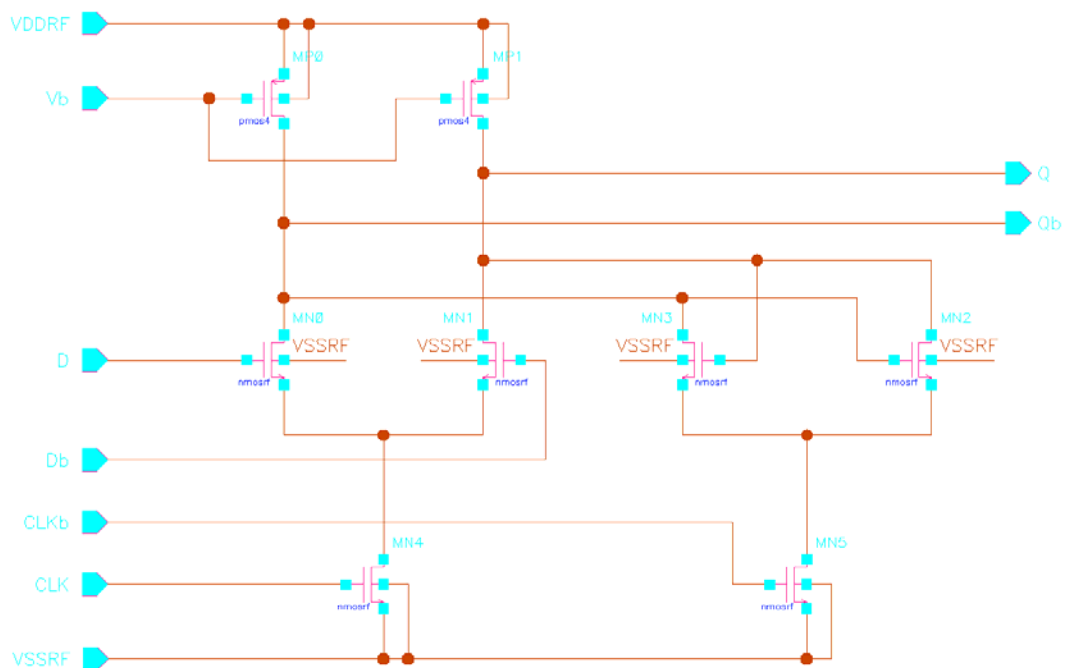


Fig. 6.29. *Latch* tipo D usando SCL (*Source Coupled Logic*) usado no circuito divisor por dois do prescaler.

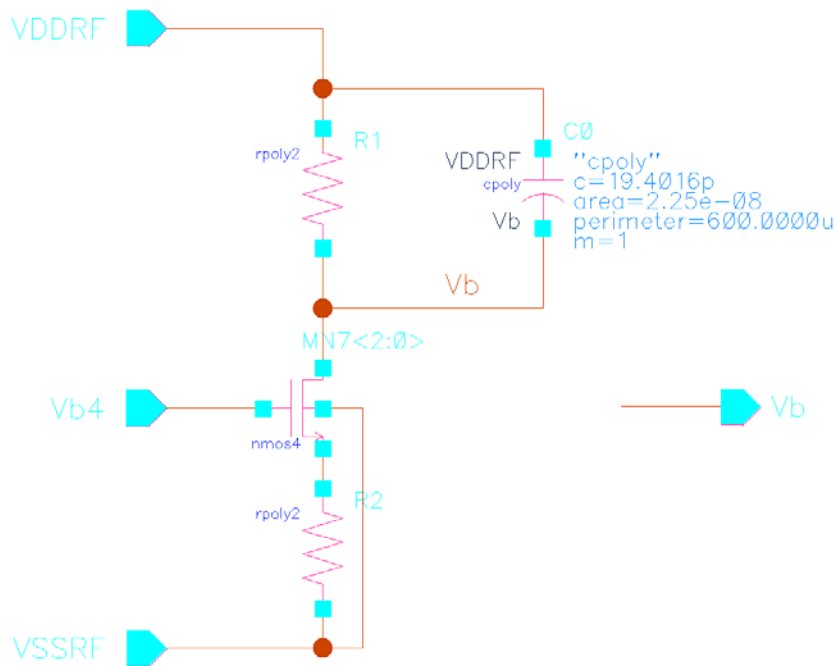


Fig. 6.30. Circuito de polarização do *latch* usando SCL proposto

A Fig. 6.31 mostra o resultado de simulação de Monte Carlo para 100 execuções. O teste foi realizado usando duas ondas quadradas defasadas de 180° na entrada do divisor, e a saída esperada é a divisão por 2.

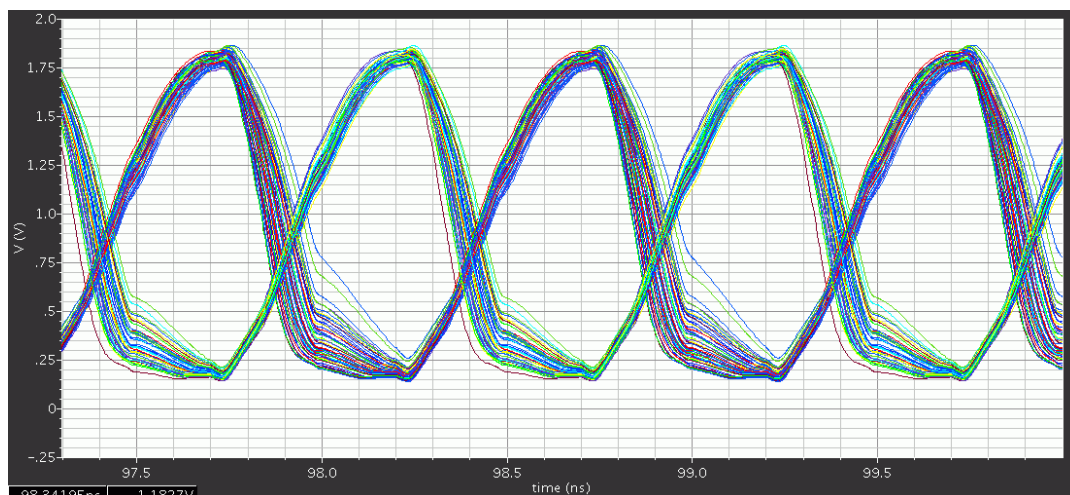


Fig. 6.31. Resultado de simulação de Monte Carlo, com 100 execuções, para o divisor de frequências da Fig. 6.29.

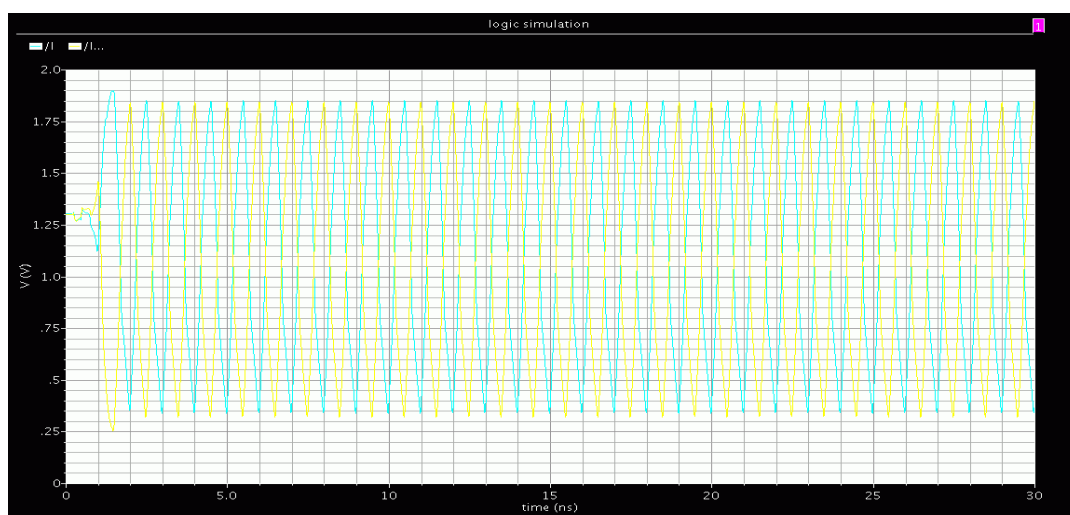


Fig. 6.32. Resultado da simulação de transiente, usando a extração de parasitas do leiaute, para o divisor de frequências da Fig. 6.29.

Na Fig. 6.32 é mostrado o resultado de simulação pós-leiaute do circuito, usando o mesmo teste citado. O leiaute do divisor por 2, usando células SCL está ilustrado na Fig. 6.33. Os pares diferenciais foram casados no leiaute, porém de modo simplificado. Nesse leiaute, a maior parte do roteamento foi em M3, para reduzir as capacitâncias parasitas geradas pelos metais. Também se utilizou de metais de largura de no mínimo 2 μm . Além de reduzir as resistências das ligações, deve-se reduzir a resistência para a corrente de polarização do circuito, devido aos picos de corrente provocados no chaveamento, apesar da corrente desses circuitos serem limitadas, podem ocorrer quedas de tensão indesejadas durante a operação do circuito.

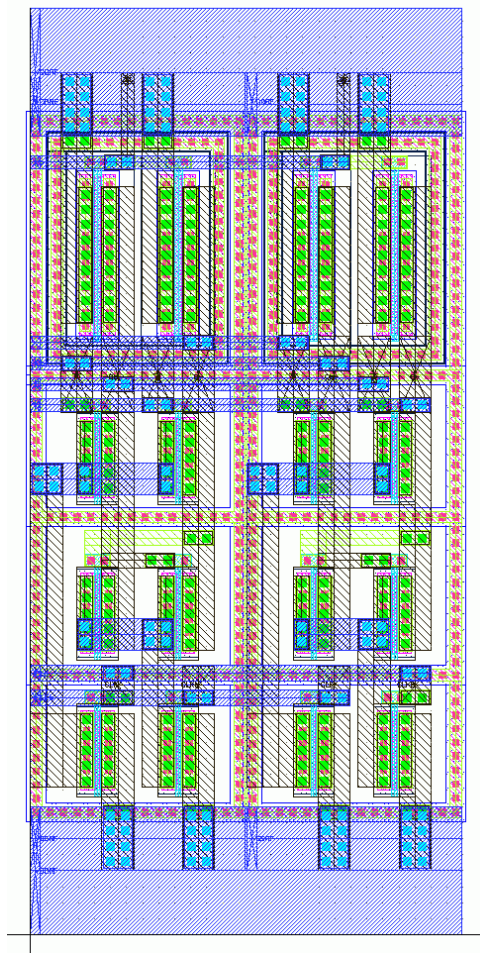


Fig. 6.33. Leiaute do divisor de frequências por 2 usando células SCL

6.3.3.3. Prescaler

A topologia utilizada no projeto do prescaler está ilustrada na Fig. 6.34 [67], [6], [53]. Diferentemente de outras topologias que utilizam contador síncrono nos primeiros estágios para gerar uma razão de divisão $M/M+1$, essa técnica divide a frequência por 4 de modo assíncrono, nos primeiros estágios. Esse fato reduz expressivamente a dissipação dinâmica de potência, sendo que apenas o primeiro estágio opera na frequência de entrada. Apesar disso, existem alguns detalhes que aumentam, relativamente, a complexidade desse projeto. O funcionamento completo do circuito será descrito a seguir. Após a frequência de entrada ser dividida duas vezes por 2, o segundo estágio apresenta 8 saídas defasadas de 45° . Cada divisor por 2 possui 4 saídas defasadas de 90° e as saídas dos dois divisores estão defasadas de 45° , por isso, as saídas devem estar intercaladas. Se $MC = 0$, então a saída será uma das possibilidades da Fig. 6.35 dividida por 8. Se $MC = 1$, a saída do multiplexador alterará pela seqüência $p7 \rightarrow p6 \rightarrow p5 \rightarrow p4 \rightarrow p3 \rightarrow p2 \rightarrow p1 \rightarrow p0$. Como resultado, o *clock* do divisor por 8 terá um pulso a mais após um ciclo da saída e o *prescaler* teria dividido o sinal por 5.

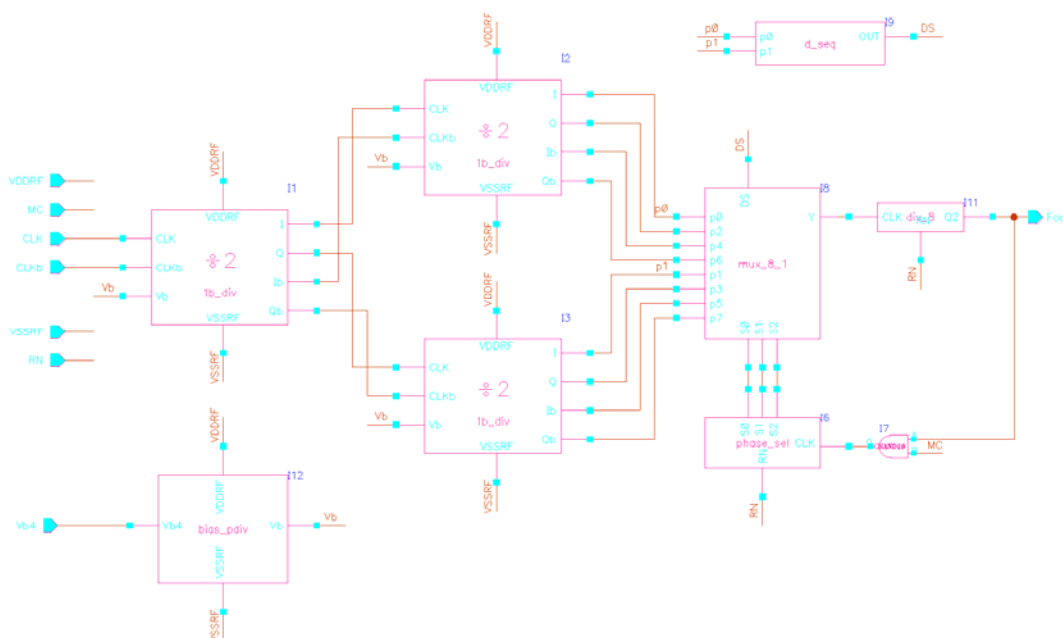


Fig. 6.34. Prescaler.

Se as saídas tiverem o comportamento da Fig. 6.35(a), o *prescaler* funcionará adequadamente, se não (Fig. 6.35(b)), o *prescaler* poderá apresentar *glitches* na divisão. Felizmente, só há duas possibilidades para as fases de saída, isso possibilita a correção da seqüência das fases. A seqüência pode ser corrigida se o multiplexador admitir duas possibilidades para a seqüência. A Fig. 6.36 mostra o circuito do multiplexador utilizado. O multiplexador admite duas possibilidades de seqüência. Se for detectada que a seqüência não é a desejada, a seqüência tomada é mudada para $p3 \rightarrow p6 \rightarrow p1 \rightarrow p4 \rightarrow p7 \rightarrow p2 \rightarrow p5 \rightarrow p0$, o que resulta nos sinais adequados para a seqüência.

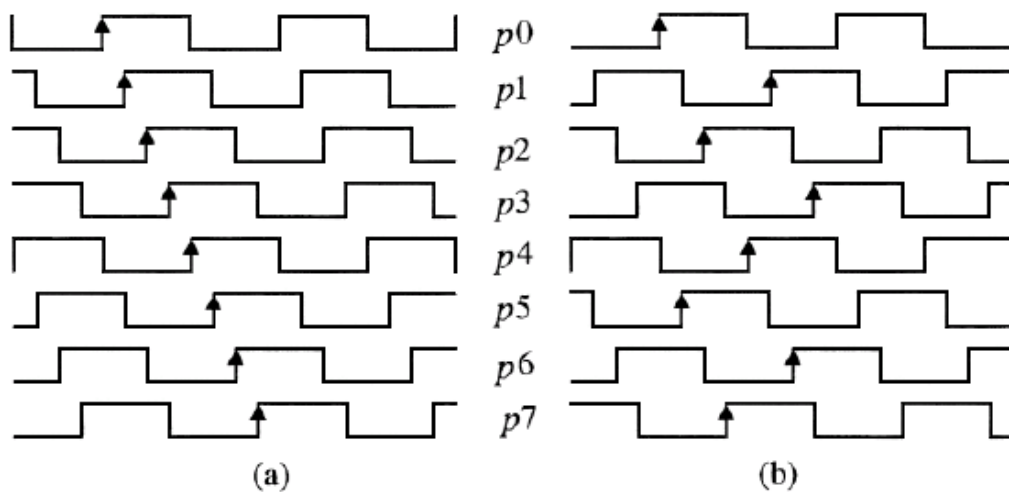


Fig. 6.35. As duas possibilidades de comportamento da saída do segundo estágio.

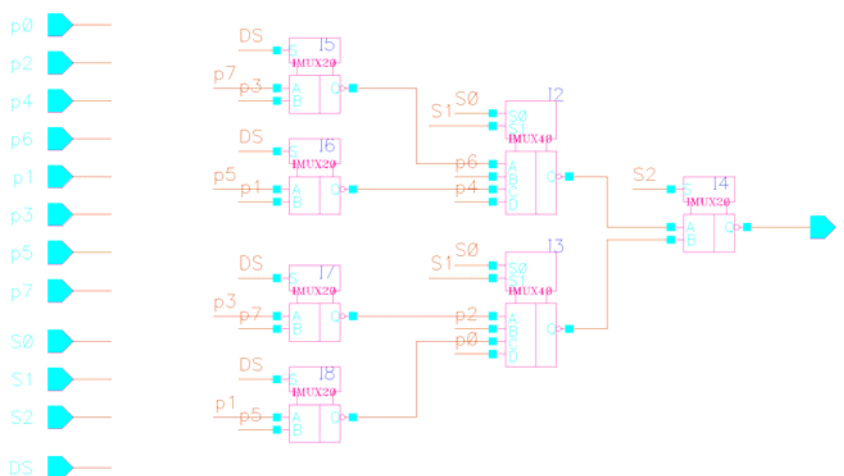


Fig. 6.36. Multiplexador.

O circuito da Fig. 6.37 é usado para detectar a seqüência de entrada [53]. Esse circuito gera um *spike* na detecção, por isso um inversor especialmente projetado com transistores com comprimento de $2\mu\text{m}$ é usado para suprimir esse sinal. Na Fig. 6.38 é mostrada a simulação desse bloco na detecção, a saída acusa a seqüência indesejada com nível lógico 1.

O *prescaler* apresentou bons resultados nas simulações do contador principal. O leiaute do *prescaler* está ilustrado na Fig. 6.39, as dimensões x e y desse leiaute são 123×146 ($\mu\text{m} \times \mu\text{m}$). Uma medida, tomada foi de dobrar os anéis de guarda do leiaute para evitar acoplamentos de sinais gerados nesses blocos para o substrato.

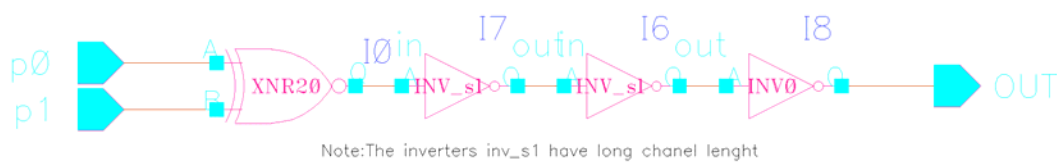


Fig. 6.37. Circuito usado na detecção da seqüência.

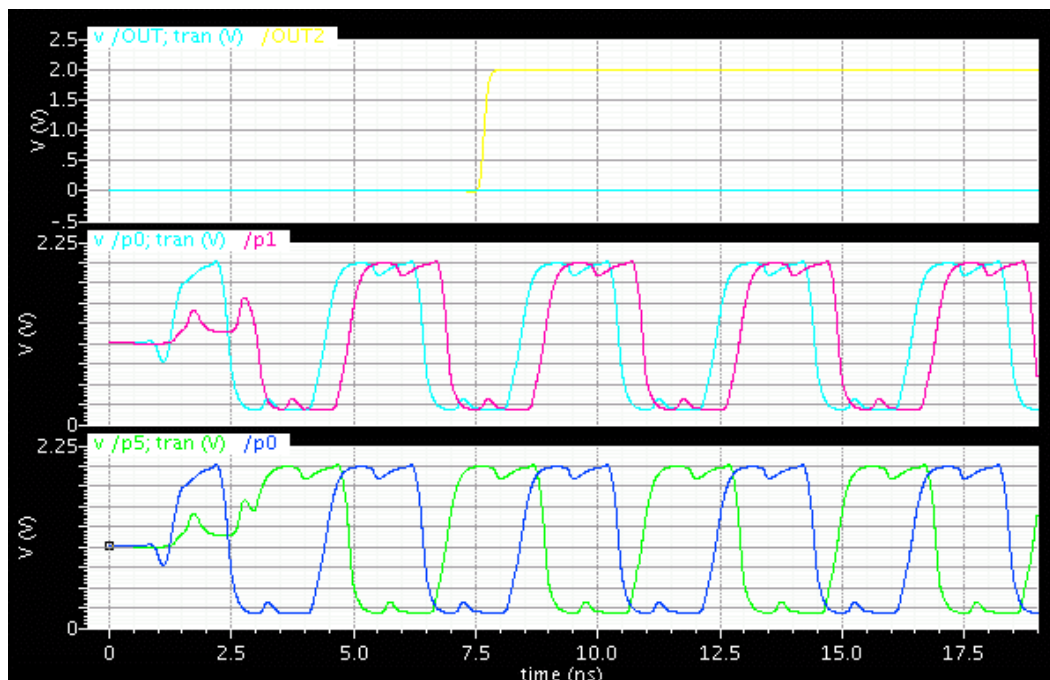


Fig. 6.38. Formas de onda na detecção da seqüência (a) saída (b) seqüência desejada (c) seqüência não desejada.

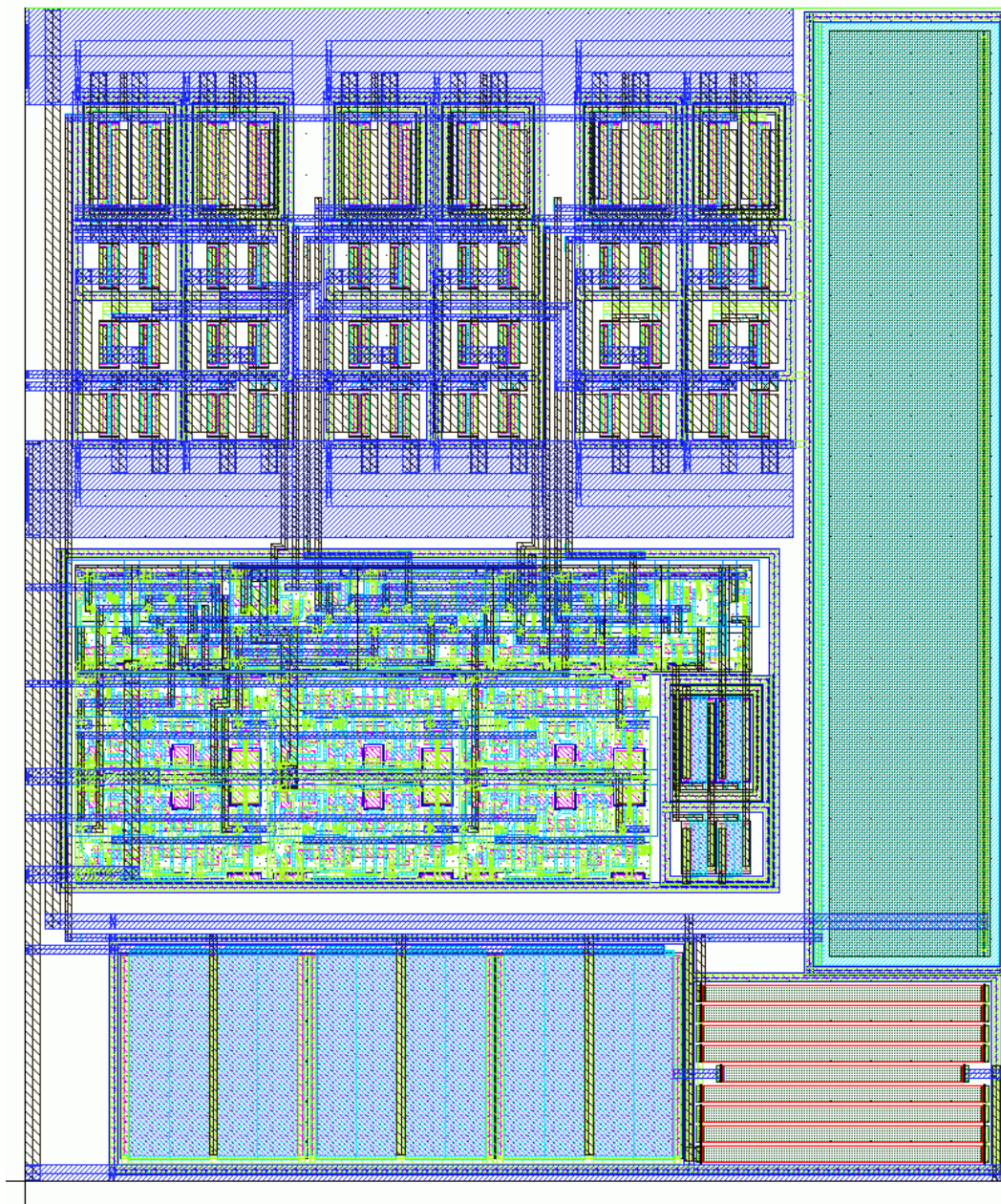


Fig. 6.39. Leiaute do prescaler

6.3.4. Projeto do Loop Filter Digital Utilizando Ferramenta de EDA

Na Fig. 6.40 é mostrado o bloco que tem a função de detector de fase com saída amostrada. O bloco foi desenvolvido para oferecer os mesmos parâmetros que

o bloco proposto em nível de sistema (Fig. 5.12). O circuito do detector de fase com saída amostrada é dividido em 3 estágios. No estágio de entrada utiliza-se um circuito semelhante ao circuito detector de fase convencional (Fig. 6.5), que fornece os sinais UP e DOWN diretamente para a saída do circuito. A lógica proposta é aplicada nesses sinais, porém as células foram rearranjadas para proporcionar os atrasos necessários. No estágio de saída, o bloco L_PULSE (Fig. 6.41) gera a diferença de fase amostrada, que é usado como sinal de *clock* para o contador crescente/decrecente.

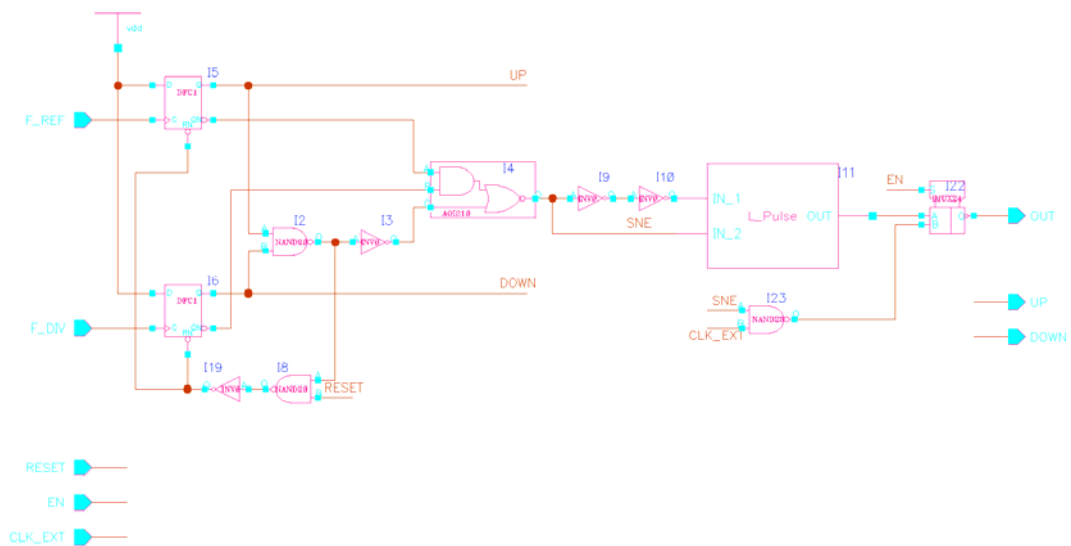


Fig. 6.40. Detector de fase com saída amostrada.

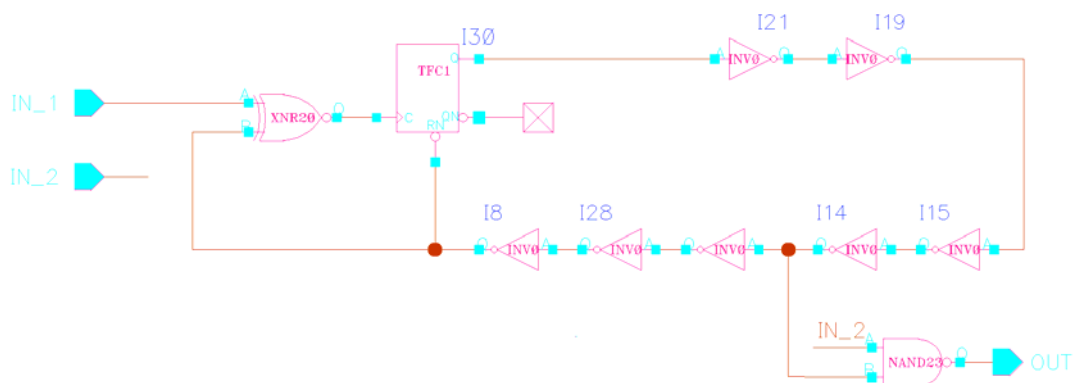


Fig. 6.41. Bloco que gera pulsos no *clock* do contador em tecnologia CMOS.

O circuito que amostra a diferença de fase que foi apresentado no capítulo 5 está mostrado na Fig. 6.41 usando lógica CMOS. Devido às discrepâncias entre os atrasos das células em nível de transistor e de portas, o circuito foi rearranjado. Esse circuito funciona como um oscilador em anel disparado pelo flip-flop e a lógica da porta XNOR. Se a entrada IN_1 está em nível lógico 0, a porta funciona como inversor. Se em 1, a porta XNOR funciona como buffer, assim o oscilador fica travado enquanto não houver diferença de fase detectada e oscila durante o período de diferença de fase detectado.

A saída do contador armazena o estado do *loop filter* digital e a frequência de amostragem do bloco L_PULSE determina a resolução da frequência de saída do sintetizador. A implementação dos blocos lógicos, cuja velocidade de chaveamento está na ordem de GHz, pode ser realizada utilizando lógica SCL, e nesse caso, os blocos seriam projetados analogicamente. Os flip-flops, junto com as demais portas lógicas que compõem o bloco contador de 1bit, seriam implementados usando resistores e transistores. A tecnologia CMOS AMS 0.35 μm possui apenas dois tipos de resistores, com resistividade relativamente baixa e coeficientes de temperatura de bom desempenho. Resistores desse tipo ocupam mais área no chip. Existem processos comerciais que disponibilizam resistores com menor área, o qual seria uma melhor opção para a caracterização da arquitetura proposta. Sendo que o objetivo do projeto é obter uma arquitetura com área reduzida, a implementação usando uma tecnologia como essa seria totalmente indesejável.

A solução para esse transtorno foi encontrada re-projetando a estabilidade da arquitetura. Para que o *loop filter* pudesse ser implementado usando apenas blocos digitais padrões, o período de amostragem foi aumentado de $7,5 \times 10^{-10}\text{s}$ para $3 \times 10^{-9}\text{s}$. A estabilidade foi atingida apenas aumentando o fator de divisão de K_2 . Foram usados dois chaveamentos nesse projeto, resultando em 8 bits acrescentados, o que significa dividir o ganho digital por 2^8 e uma resolução de 140 Hz. O rtool foi utilizado nesse projeto, através do método descrito no capítulo 5.

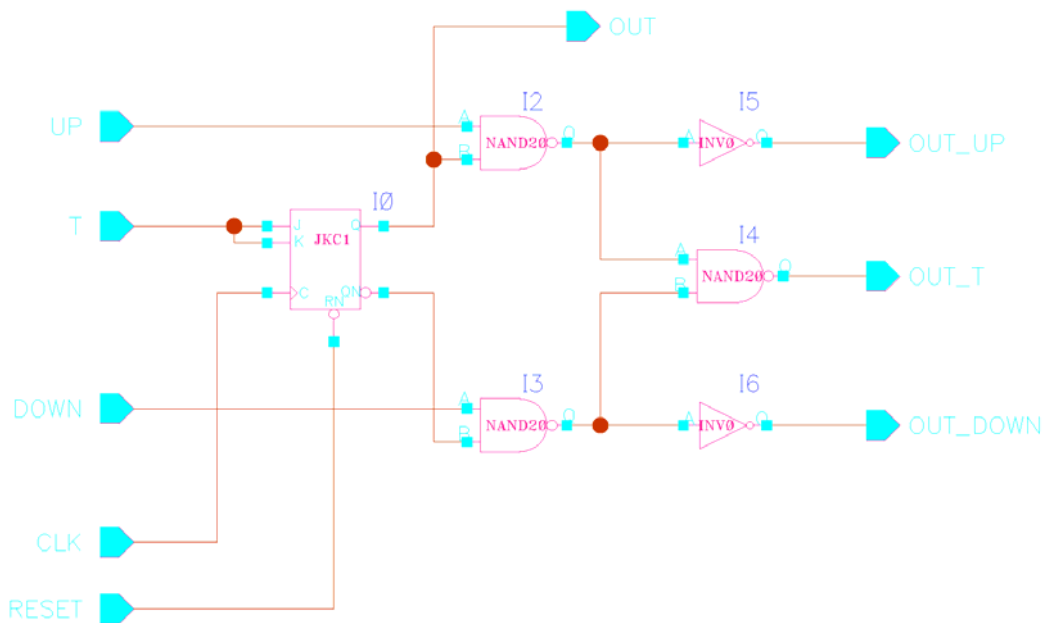


Fig. 6.42. 1 bit do contador crescente/decrecente usando somente células standard.

Na Fig. 6.42 está ilustrado o bloco que gera um bit de contagem para o contador. O contador utiliza uma topologia de contador síncrono usando flip-flop tipo T e possui *reset* assíncrono. O contador opera normalmente até que se atinja um instante pré-definido no projeto do sintetizador, nesse instante acontece o chaveamento, onde se incorporam alguns bits menos significativos ao contador. As saídas do contador, que já estavam habilitadas, permanecem com o mesmo peso de antes do chaveamento. As saídas introduzidas, são somadas com peso reduzido à matriz de varactores do VCO híbrido. Como resultado, a cada bit incluído o ganho do *loop filter* é dividido por dois, pois um contador é colocado em cascata com o contador que iniciou a contagem. Esse método está ilustrado na Fig. 6.43.

O circuito do *loop filter* digital, mostrado na Fig. 6.44, foi testado em simulações em nível de transistor para verificar a sua funcionalidade. Um conversor D/A foi utilizado para gerar a saída do circuito. Na Fig. 6.45 é apresentado o resultado da simulação para verificar o funcionamento do *loop filter* digital na amostragem da diferença de fase e como integrador digital. Na Fig. 6.46 os detalhes da operação do circuito são apresentados, mostrando que não há detecção errada da diferença de fase.

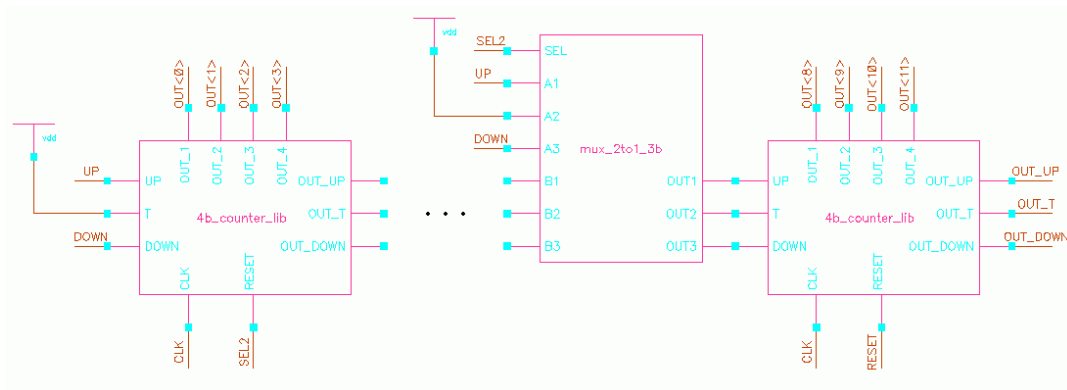


Fig. 6.43. Bits menos significativos da saída do *loop filter* digital.

Os leiautes dos circuitos do *loop filter* digital estão mostrados nas figuras Fig. 6.47 e Fig. 6.48. Esses leiautes apenas usam células da biblioteca de células digitais da AMS e roteamento. A área total desse leiaute é de aproximadamente 340×96 ($\mu\text{m} \times \mu\text{m}$).

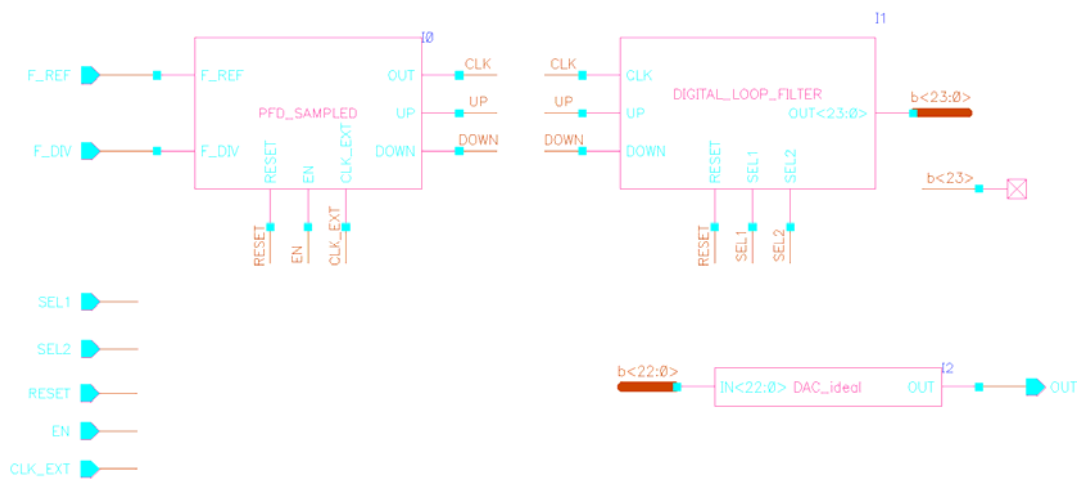


Fig. 6.44. Esquemático do *loop filter* digital usado em simulações para testar sua funcionalidade.

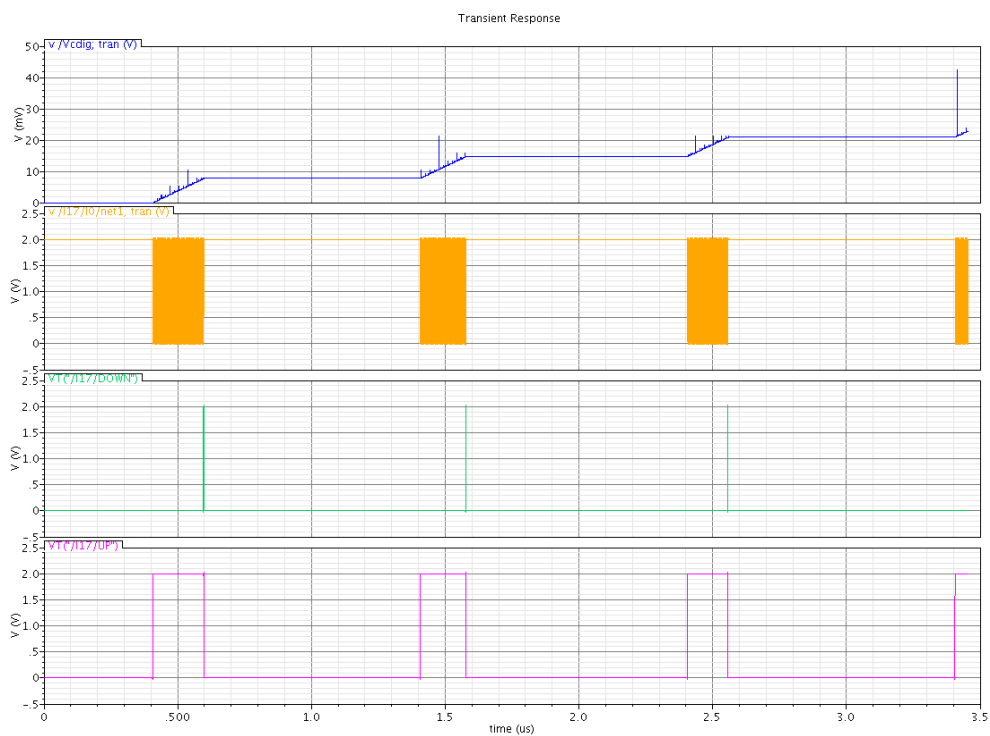
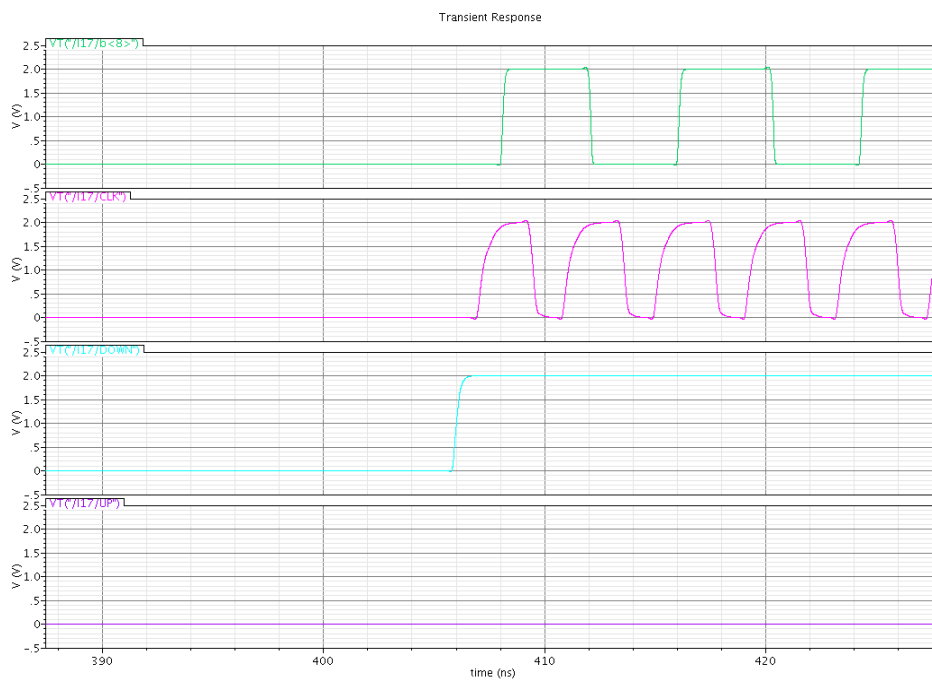
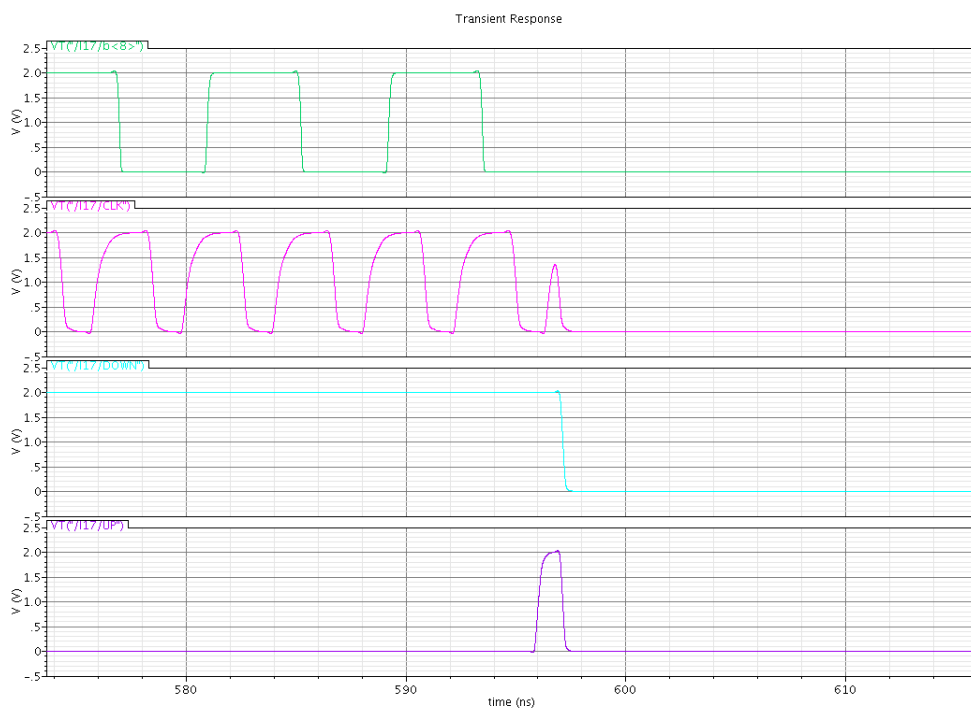


Fig. 6.45. Simulação para verificar o funcionamento do loop filter digital na detecção da diferença de fase.



(a)



(b)

Fig. 6.46. Comportamento do contador quando a diferença de fase é detectada (a) bit menos significativo no momento em que se inicia a contagem (b) bit menos significativo no momento em que se finaliza a contagem.

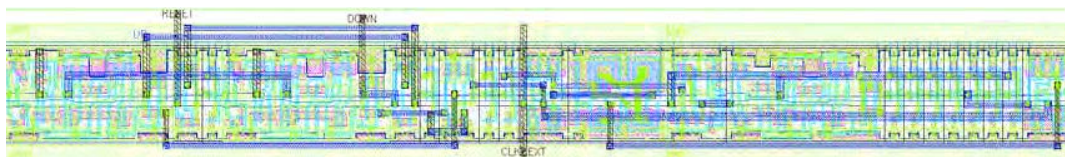


Fig. 6.47. Leiaute do circuito de PFD (*Phase Frequency Detector*) amostrado

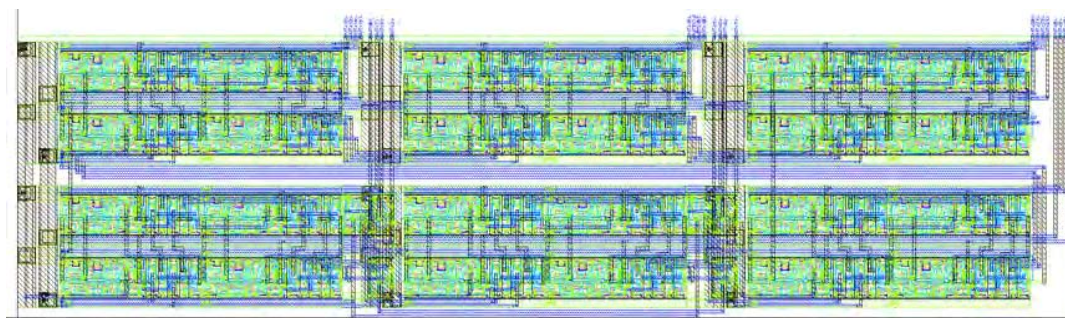


Fig. 6.48. Leiaute do contador crescente/decrescente usando as técnicas digitais propostas

6.4. Projeto do PLL e Considerações de Desempenho

Nessa seção, descreve-se o projeto do PLL usando os blocos em nível de transistor, bem como o seu leiaute e re-simulação. Com o objetivo de verificar as especificações definidas para o sintetizador, simulações em nível de sistema e transistor foram realizadas e são descritas nessa seção. Os ajustes necessários podem ser conseguidos através dos parâmetros do filtro, K_{VCO} , corrente do *charge pump* e ganho do laço digital. Nesse ponto do projeto, onde os blocos já foram projetados, pretende-se que apenas ajustes pequenos sejam ministrados, e mesmo pequenos, é interessante que não atinjam parâmetros tão críticos como o K_{VCO} . Ajustes como, valores de componentes passivos, são bem vindos como solução para desvios da especificação ou problemas encontrados na arquitetura.

6.4.1. Modelos Comportamentais Usados nas Simulações

Os modelos comportamentais calibrados com os blocos em nível de transistor, incluem modelos em Verilog-A e macromodelos calibrados automaticamente e modelos calibrados manualmente, por análise de resultados de simulação. Por exemplo, as células lógicas usadas no bloco PFD_SAMPLED são portas lógicas descritas em Verilog-A calibrados com os atrasos e tempos de transição das células digitais da biblioteca do fabricante. O código em Verilog-A está no apêndice B. A Fig. 6.49 mostra os blocos usados para modelar o *loop filter* digital nas simulações de ruído de fase e tempo de estabelecimento. Um PLL com *loop filter* convencional foi simulado em paralelo com a topologia proposta, para comparar os resultados obtidos com os dois tipos de circuito. O objetivo é verificar as vantagens e desvantagens de se usar a arquitetura proposta, sendo que o intuito dessa proposta é realizar todas as funções de um PLL convencional. As figuras Fig. 6.50 e Fig. 6.51 mostram os esquemáticos de teste, usando os modelos comportamentais que foram calibrados como os blocos em nível de transistor, respectivamente, para a topologia convencional e a topologia proposta de PLL.

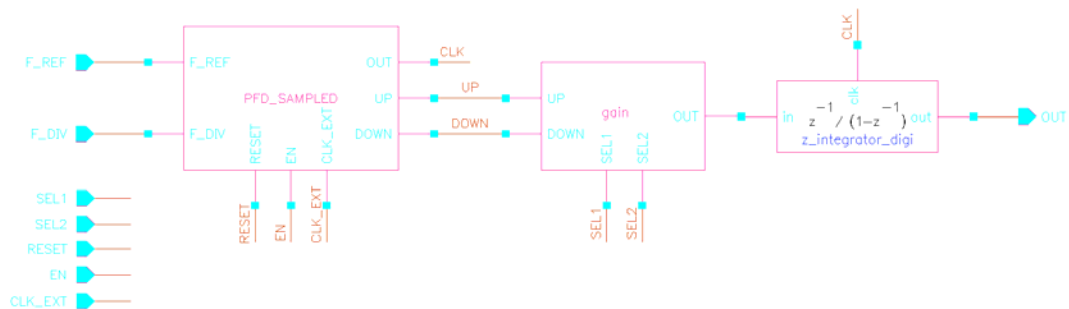


Fig. 6.49. Esquemático do *loop filter* digital usando modelos comportamentais utilizado nas simulações de ruído e tempo de estabelecimento.

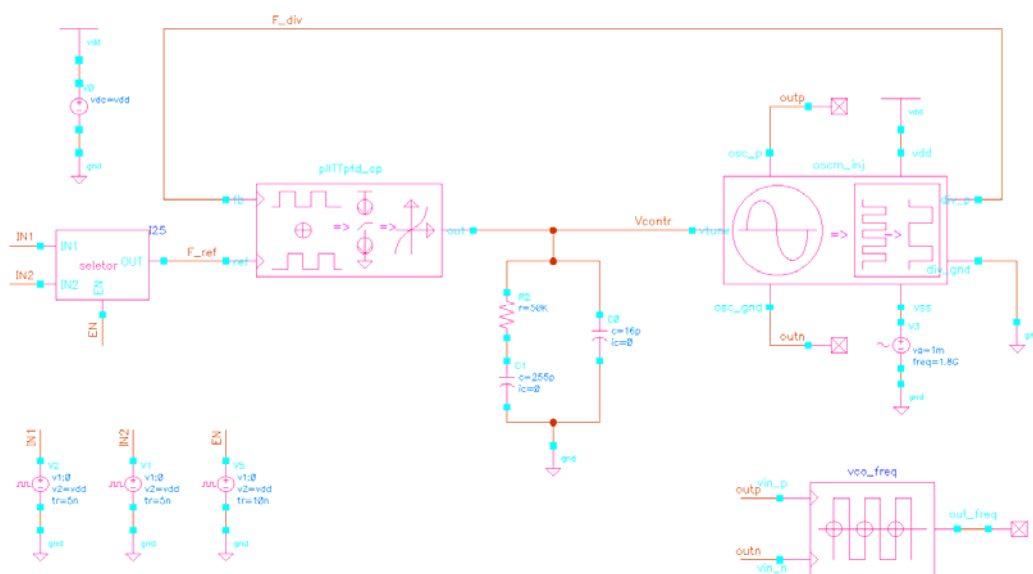


Fig. 6.50. Esquemático de teste do PLL convencional usando os modelos comportamentais que foram calibrados com os blocos em nível de transistor.

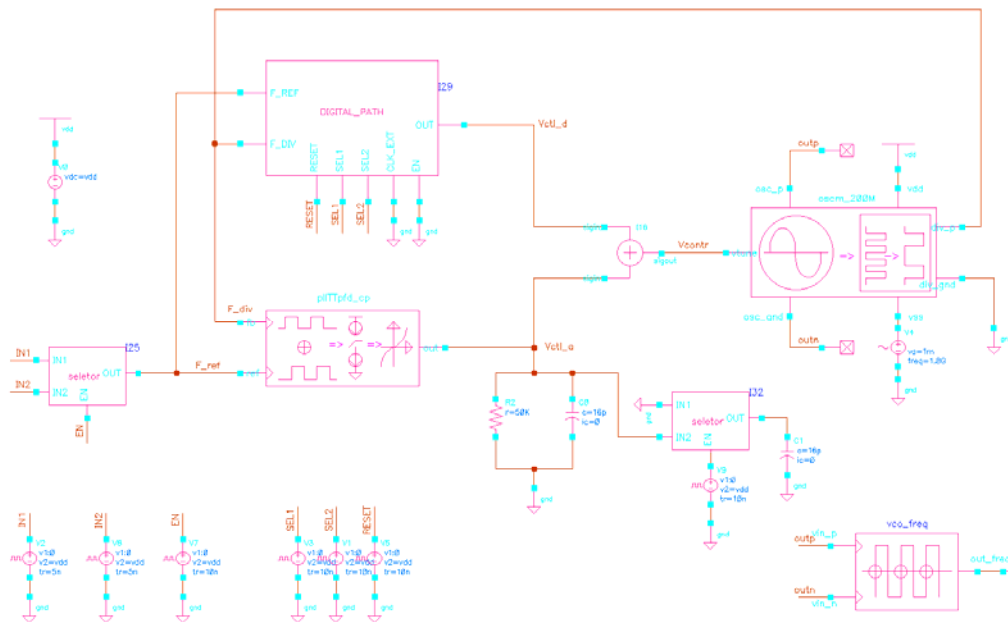


Fig. 6.51. Esquemático de teste da PLL proposto usando os modelos comportamentais que foram calibrados com os blocos em nível de transistor.

6.4.2. Análises Comparativas de Ruído

O resultado de simulação de ruído de fase, para o VCO calibrado, está mostrado na Fig. 6.52. Esse resultado serve como base para a análise do ruído do PLL, considerando a ação da dinâmica do sistema no ruído de fase. Como já foi comentado, os resultados com PPV mostraram-se compatíveis com os resultados em nível de transistor.

Na análise das simulações, não foram encontradas discrepâncias entre o circuito do PLL convencional e o proposto, como pode ser observado na Fig. 6.53. O PLL proposto apresenta *jitter* baixo e supressão de tons espúrios, assim como previsto. Até mesmo, os resultados com o *dual-path* apresentaram um ruído de fase ligeiramente menor que os obtidos da topologia convencional.

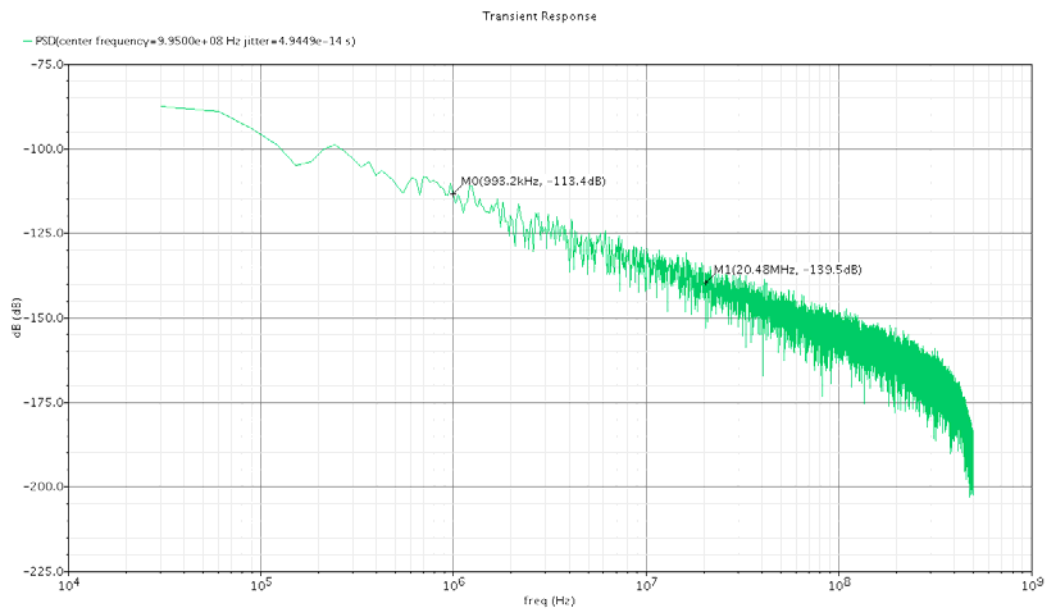
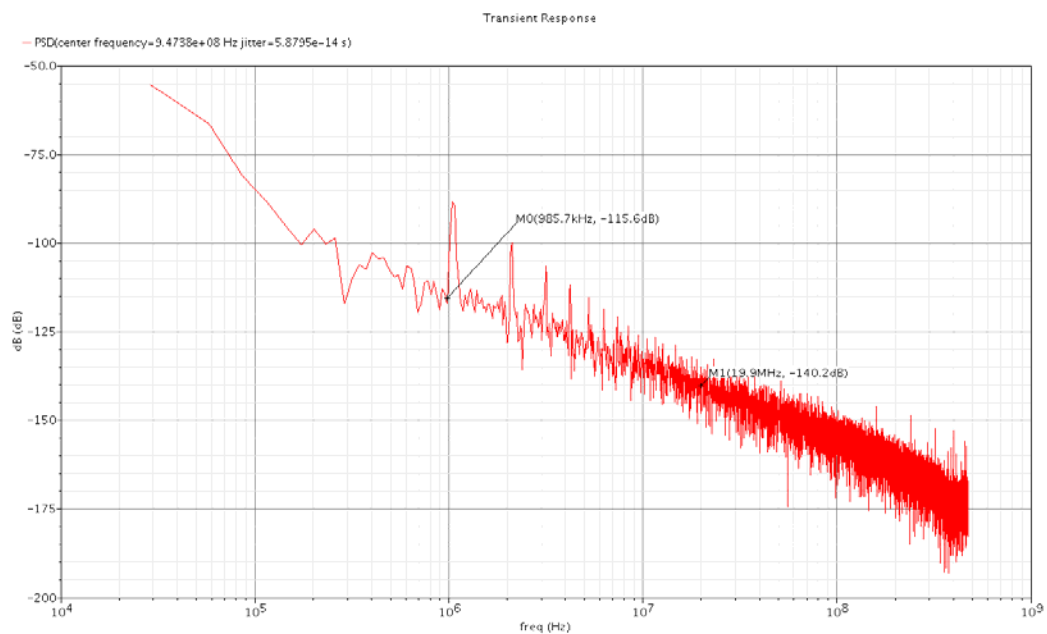
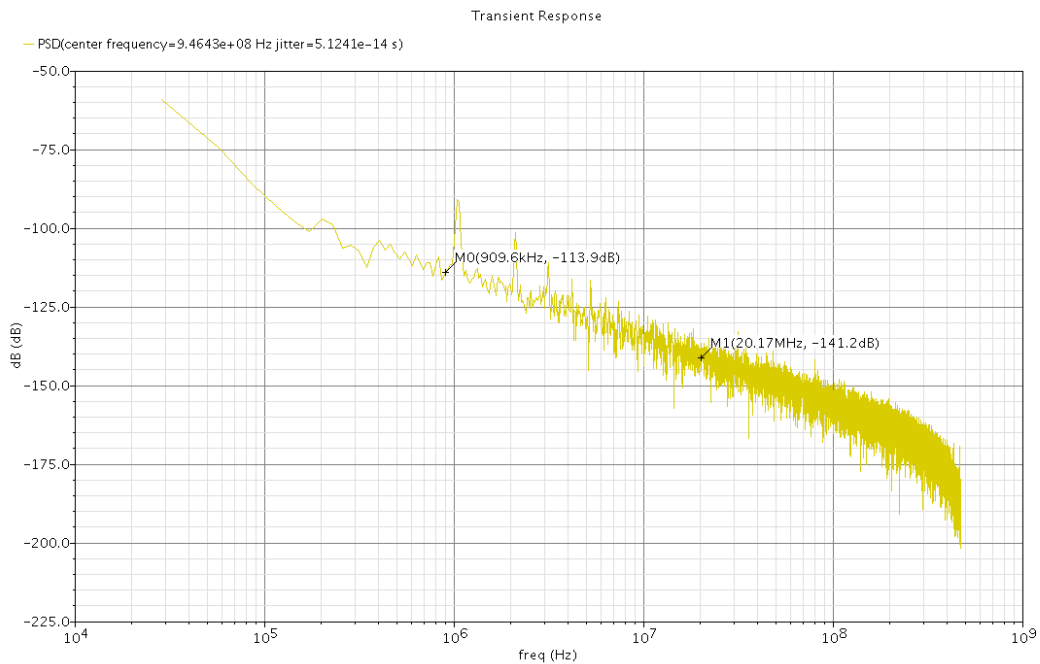


Fig. 6.52. Resultado de simulação de ruído de fase, obtido através de simulação transiente, usando o macromodelo em PPV do VCO proposto.



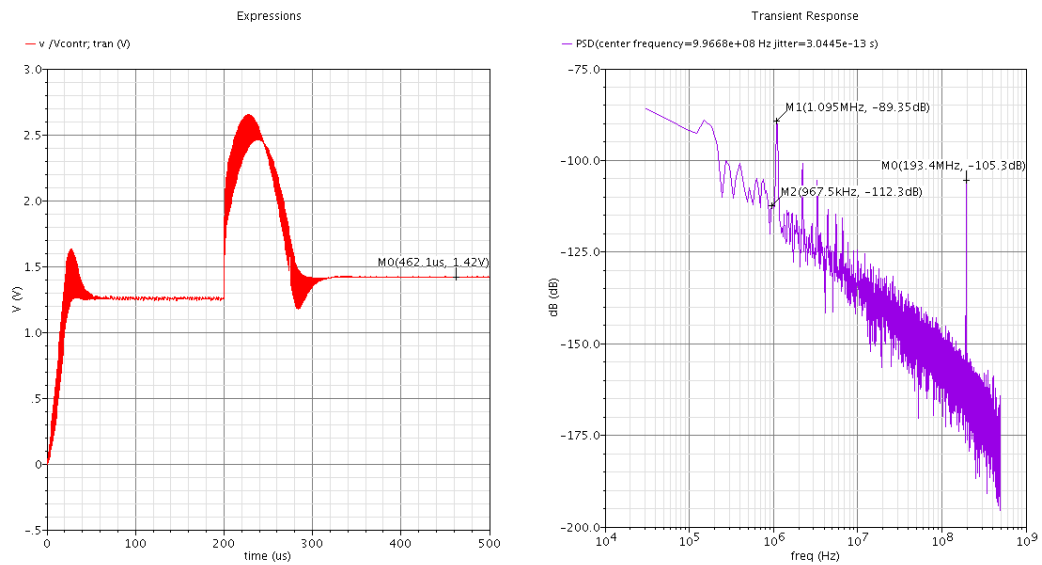
(a)



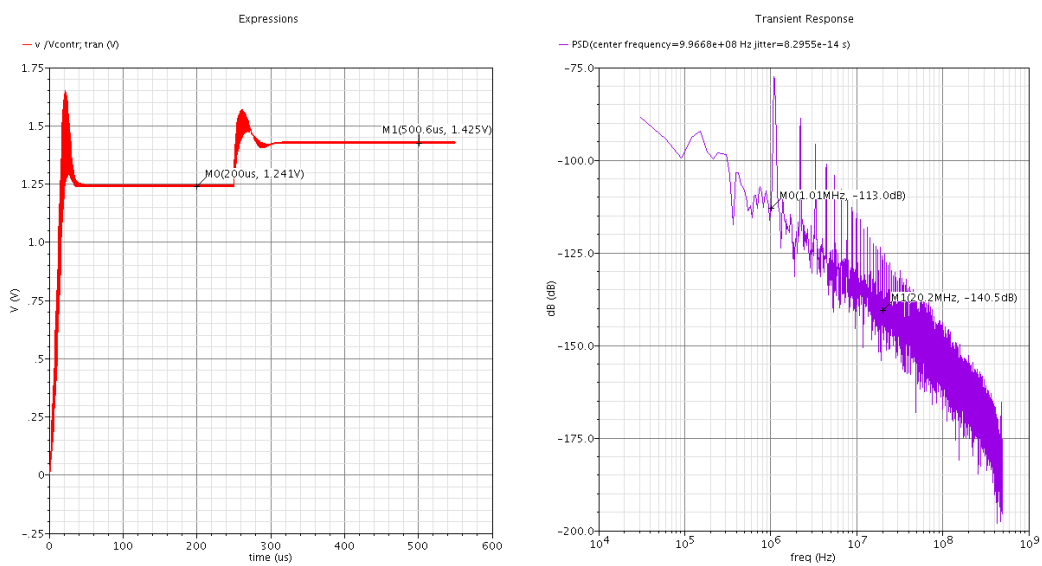
(b)

Fig. 6.53. Resultado de simulação de ruído de fase obtida através de simulação transiente usando PPV (a) para o PLL convencional (b) para o PLL usando dual-path .

Simulações usando injeção de sinal foram realizadas para verificar a influência de efeitos como *injection locking* no funcionamento da topologia *dual path*. Devido a limitações no modelo realizado usando PPV, ocorreram tons espúrios no espectro do ruído. Após o uso do bloco *z_integrator_dig*, da Fig. 6.49, os resultados passaram a apresentar tons espúrios não previstos e, por isso, o mais prudente é desprezar essas ocorrências. O resultado da comparação dos resultados é que o sinal injetado é completamente suprimido na arquitetura proposta, mas não é suprimido na arquitetura convencional, como pode ser visto na Fig. 6.54. Verifica-se também, que a dinâmica do PLL usando dual-path, obtém um menor tempo de estabelecimento, devido às técnicas empregadas.



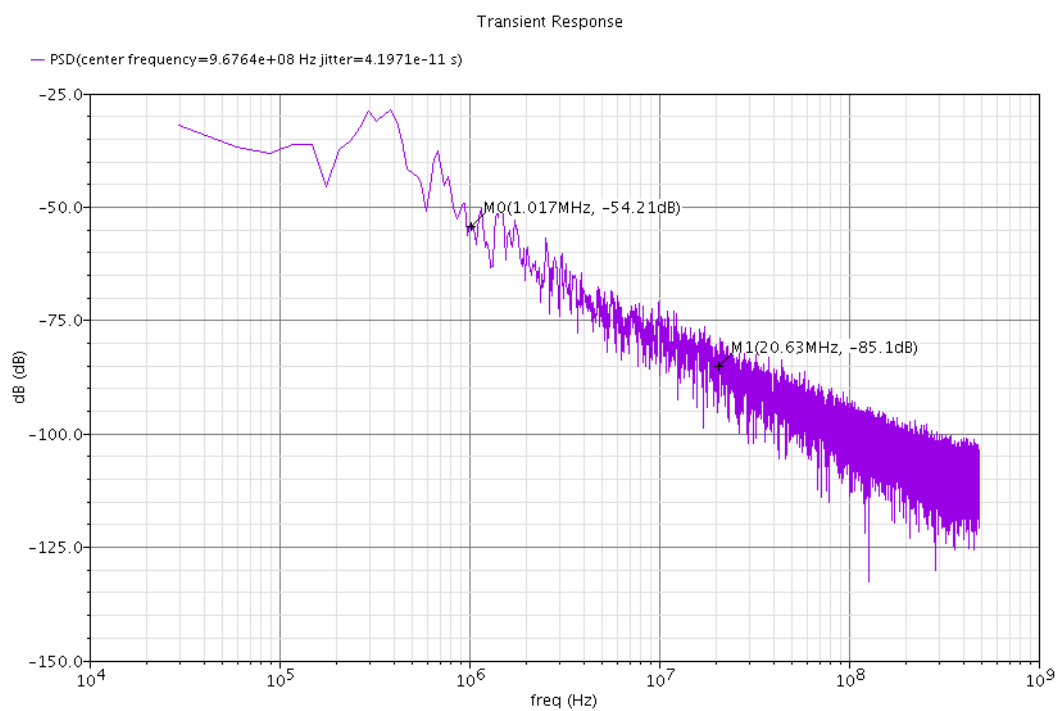
(a)



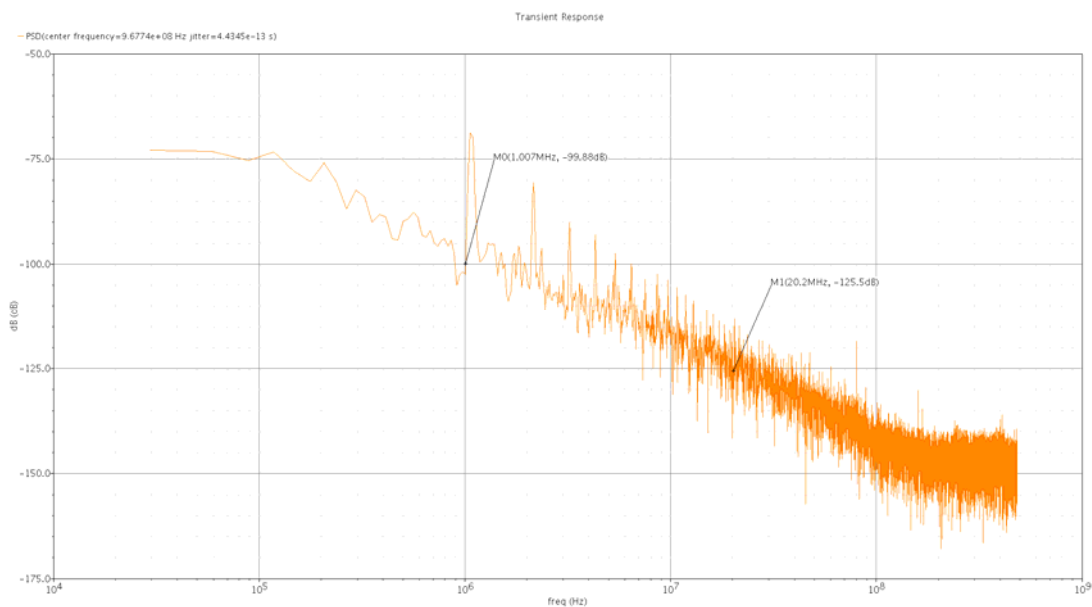
(b)

Fig. 6.54. Resultado de simulação de *injection locking* com um sinal de 1.8GHz-1mA injetado, obtida através de simulação transiente usando PPV (a) para o PLL convencional (b) para o PLL usando *dual-path*.

Na Fig. 6.55 é mostrado o resultado da simulação de supressão de ruído de entrada para um jitter da referência de 0,1 ps (ciclo-para-ciclo). Um jitter relativamente elevado foi usado propositalmente para facilitar as comparações.



(a)



(b)

Fig. 6.55. Resultado de simulação de ruído de fase para uma injeção de jitter de entrada de 0,1 ps (ciclo-para-ciclo) obtida através de simulação transiente usando PPV (a) para o PLL convencional (b) para o PLL usando dual-path .

Analisando os resultados verifica-se que o circuito PLL convencional não é eficiente na supressão de ruído da referência, enquanto que o PLL usando *dual-path* obtém uma degradação muito inferior. A supressão de ruído da referência é 45,6 dB melhor no PLL proposto.

Pretende-se ainda simular a topologia apresentada na Fig. 6.4, em nível de transistor, para verificar as especificações de exatidão e tempo de estabelecimento. Essas simulações, bem como as simulações pós-leiaute do sintetizador exigem uma ferramenta adequada, como o Ultrasim. Devido às restrições no modelo realizado usando PPV, simulações como *hopping* dos canais, onde o divisor programável deve ser usado, e exatidão não podem ser validadas sem utilizar simulações em nível de transistor. Foram realizadas simulações em nível de transistor para o PLL proposto, onde se pôde verificar o funcionamento do VCO híbrido, porém devido à indisponibilidade de licença da ferramenta Cadence, durante a fase final do doutorado, os resultados de tempo de estabelecimento e exatidão em nível de transistor não constam do texto da tese. Muito provavelmente em publicações posteriores, serão encontrados esses resultados e todos os quesitos analisados em nível de sistema poderão ser verificados.

Capítulo VII

7. Conclusão

A arquitetura *dual-path* desenvolvida mostrou-se uma boa alternativa para aplicações de baixo custo atingindo bom desempenho para atender às especificações de sistemas sintetizadores de frequência. Destacam-se duas vantagens principais da técnica *dual-path* empregada, em relação à topologia convencional de *loop-filter*. A primeira é que o laço digital promove uma redução significativa na área do filtro, como foi provado. A outra é que a inserção do laço digital permite que técnicas digitais sejam aplicadas para alterar a dinâmica do PLL enquanto o circuito está sincronizando, possibilitando a otimização desse processo de uma forma simplificada.

O uso de um acumulador digital no *loop filter* impôs várias restrições ao projeto do sintetizador. Os principais problemas observados estão relacionados com a resolução da frequência de saída, distorção e estabilidade. Algumas das alternativas avaliadas para atingir a resolução requerida, foram utilizar lógica CML para aumentar a frequência de amostragem do acumulador ou que circuitos mais complexos sejam utilizados no processamento digital do *loop filter*. Esses métodos não estariam de acordo com as metas desejadas. A técnica proposta, além de mais simplificada, pois se utilizam apenas células digitais padrão, propiciou que também se aumentasse o desempenho do sintetizador de frequências no processo. O projeto do *loop filter* digital desenvolvido, segue dois passos. No primeiro, a resolução necessária é definida como o menor valor obtido pelo *loop filter* digital, ou seja, o ganho para o bit menos significativo do contador. No segundo utiliza-se da técnica de chaveamento do ganho do laço digital para reduzir o ganho para esse valor. Esse procedimento reduz o tempo de estabelecimento do sintetizador de frequências e ainda atinge a resolução necessária, como foi mostrado nas simulações em nível de sistema. As técnicas para redução de tons espúrios, desenvolvidas para esse *loop*

filter, utilizam as alterações que essa técnica faz na frequência de ganho unitário do loop filter. Para tanto, foi desenvolvida uma metodologia de projeto da estabilidade do sintetizador proposto. A estabilidade é projetada usando emulação do controle linear e controle digital. O projeto usando emulação permite que controle linear clássico seja utilizado. Com esse método foi deduzida a equação que compara a arquitetura proposta com a convencional. O resultado mostra que a arquitetura oferece uma forma de reduzir a área de projetos que precisam de supressão de tons espúrios ou faixa de frequência larga porque a área não necessita ser aumentada proporcionalmente à supressão, mas sim em uma escala logarítmica. O projeto da estabilidade usando controle digital permite mais precisão no projeto das especificações do sistema. Além disso, foram deduzidas as equações dos pólos e zeros no domínio-s contínuo a partir da equação no domínio-z discreto. Esses dados possibilitam que o projetista utilize o método convencional para projetar o PLL proposto. O projeto da estabilidade do sintetizador proposto foi verificado nas simulações em nível de sistema na ferramenta da Mentor Graphics.

Um dos objetivos requeridos para a arquitetura proposta foi que a adição de blocos digitais não representasse um aumento substantivo na complexidade do projeto do PLL, pois essa é justamente uma das desvantagens dos projetos digitais. Na topologia desenvolvida, um circuito digital simples realiza todo o processo explicado. Os chaveamentos devem ser realizados em tempos pré-definidos no projeto do sintetizador de frequência. O *baseband* digital deve ser o responsável pelo controle desse processo.

No fluxo utilizado, cada grupo de blocos foi simulado previamente em nível de transistor e depois esses grupos foram calibrados para as simulações de sistema. O loop filter digital obteve bons resultados nas simulações em nível de transistor, ocupando uma área total de aproximadamente 340×96 ($\mu\text{m} \times \mu\text{m}$). O prescaler também obteve bons resultados, ocupando uma área de 123×146 ($\mu\text{m} \times \mu\text{m}$). O VCO híbrido atingiu as especificações do sistema nas simulações pós-leiaute. As dimensões x e y do VCO são $632 \times 543,7$ ($\mu\text{m} \times \mu\text{m}$) e estima-se que os demais blocos devem preencher a área vazia. Os circuitos calibrados foram simulados em termos de ruído, donde se concluiu que a arquitetura proposta obtém a supressão de espúrios desejada.

O objetivo da arquitetura de produzir uma topologia mais propícia para processo CMOS padrão foi alcançado. Considerando que a área que seria ocupada

por apenas o capacitor necessário para desempenhar a mesma função do circuito proposto é maior que $632 \times 543,7$ ($\mu\text{m} \times \mu\text{m}$), conclui-se que o objetivo de obter uma arquitetura de baixo custo foi alcançado. Essa arquitetura pode ser utilizada para integrar outros sistemas sintetizadores de frequência mais sofisticados. O *loop filter* digital proposto pode ser usado, em trabalhos futuros, como bloco de uma arquitetura multi-path, no qual poderia se empregar controle moderno ao invés de controle digital clássico, como no caso da arquitetura atual.

Referências

- [1] STASZEWSKI, R.B.; BALSARA, P.T. **All-digital frequency synthesizer in deep-submicron CMOS**. New Jersey: John Wiley & Sons, 2006.261 p.
- [2] Son, H.W. **A fully integrated fractional-N frequency synthesizer for wireless communications**. 2004. 126 f. Tese (Doctor of Philosophy in Electrical Engineering) - Georgia Institute of Technology, 2004. Disponível em: <<http://smartech.gatech.edu/handle/1853/5254>>. Acesso em: 27 nov. 2009.
- [3] ABIDI, A.A. RF CMOS comes of age. **IEEE Journal of Solid-State Circuits**, New York, v.39, n.4, p. 549-561, Apr. 2004.
- [4] STASZEWSKI, R.B. et al. All-digital TX frequency synthesizer and discrete-time receiver for Bluetooth radio in 130-nm CMOS. **IEEE Journal of Solid-State Circuits**, New York, v.39, n.12, p. 2278-2291, Dec. 2004.
- [5] MOON, H.; HAN, J. et al. A 0.13- μm CMOS multi-band WCDMA/HSDPA receiver adopting silicon area reducing techniques. In: IEEE RADIO FREQUENCY INTEGRATED CIRCUITS SYMPOSIUM, 2009, Boston. **Proceedings...** Boston: [S. n.], 2009.p. 17-20.
- [6] SHU, K.; SÁNCHEZ-SINENCIO, E. **CMOS PLL synthesizers: analysis and design**. New York: The Kluwer International Series in Engineering and Computer Science, 2005.215 p.
- [7] AGNELLI, F. et al. Wireless multi-standard terminals: system analysis and design of a reconfigurable RF front-end. **IEEE Circuits and Systems Magazine**, New York, v.6, n.1, p. 38–59, First Quarter 2006.
- [8] APPEL, G. Fractional N synthesizers: analyzing fractional N synthesizers and their ability to reduce phase noise, improve loop speed and reduce reference spur levels. **RF Design magazine | RF Signal Processing**. [S. l.], [S. n.] p. 34–50, Nov. 2000.
- [9] HENG, C.-H.; SONG, B.-S. A 1.8-GHz CMOS fractional-N frequency synthesizer with randomized multiphase VCO. **IEEE Journal of Solid-State Circuits**, New York, v.38, n.6, p. 848–854, Jun. 2003.

- [10] ZARKESHVARI, F.; NOEL, P.; KWASNIEWSKI, T. PLL-based fractional-N frequency synthesizers. In: FIFTH INTERNATIONAL WORKSHOP ON SYSTEM-ON-CHIP FOR REAL-TIME APPLICATIONS, 2005, Alberta, **Proceedings...** Alberta: [S. n.], 2005.p. 85-91.
- [11] YANG, Y.-C.; YU, S.-A.; WANG, T.; LU, S.-S. A dual-mode truly modular programmable fractional divider based on a 1/1.5 divider cell. **IEEE Microwave and Wireless Components Letters**, New York, v.15, n.11, p. 754-756, Nov. 2005.
- [12] MENINGER, S.E.; PERROTT, M.H. A 1-MHZ bandwidth 3.6-GHz 0.18- μm CMOS fractional-N synthesizer utilizing a hybrid PFD/DAC structure for reduced broadband phase noise. **IEEE Journal of Solid-State Circuits**, New York, v.41, n.4, p. 966-980, Apr. 2006.
- [13] PERROTT, M.H.; TROTT, M.D.; SODINI, C.G. A modeling approach for Σ - Δ fractional-N frequency synthesizers allowing straightforward noise analysis. **IEEE Journal of Solid-State Circuits**, New York, v.37, n. 8, p. 1028-1038, Aug. 2002.
- [14] MUNKER, C.; KLEPSE, B.-U. et al. Digital RF CMOS transceivers for GPRS and EDGE. In: IEEE RADIO FREQUENCY INTEGRATED CIRCUITS SYMPOSIUM, 2005, Long Beach, **Proceedings...** Long Beach: [S. n.], 2005.p. 265-268.
- [15] SCHEIBLHOFER, S.; SCHUSTER, S.; STELZER, A. Signal model and linearization for nonlinear chirps in FMCW radar SAW-ID tag request **IEEE Transactions on Microwave Theory And Techniques**, New York, v.54, n.4, p. 1477-1483, Apr. 2006.
- [16] STASZEWSKI, R.B.; LEIPOLD, D.; Balsara, P. T. Direct frequency modulation of an ADPLL for bluetooth/GSM with injection pulling elimination. **IEEE Transactions on Circuits and Systems-II: Express Briefs**, New York, v.52, n.6, p. 339-343, Jun. 2005.
- [17] LO, C.-W.; LUONG, H.C. A 1.5-V 900-MHz monolithic CMOS fast-switching frequency synthesizer for wireless applications. **IEEE Journal of Solid-State Circuits**, New York, v.37, n.4, p. 459-470, Apr. 2002.
- [18] LEE, T.-C.; RAZAVI, B. A stabilization technique for phase-locked frequency synthesizers. **IEEE Journal of Solid-State Circuits**, New York, v.38, n.6, p. 888-894, Jun. 2003.

- [19] HERZEL, F.; FISCHER, G.; GUSTAT, H. An integrated CMOS RF synthesizer for 802.11a wireless LAN. **IEEE Journal of Solid-State Circuits**, New York, v.38, n.10, p. 1767-1770, Oct. 2003.
- [20] CHIANG, J.-S.; CHEN, K.-Y. The design of an all-digital phase-locked loop with small DCO hardware and fast phase lock. **IEEE Transactions on Circuits and Systems-II**, New York, v.46, n.7, p. 945–950, Jul. 1999.
- [21] STASZEWSKI, R.B. et al. All-digital PLL and transmitter for mobile phones. **IEEE Journal of Solid-State Circuits**, New York, v.40, n.12, p. 2469-2482, Dec. 2005.
- [22] CHIUEH, T.-D.; YANG, J.-B.; WU, J.-S. Design and implementation of a low-voltage fast-switching mixed-signal-controlled frequency synthesizer. **IEEE Transactions on Circuits and Systems-II**, New York, v.48, n.10, p. 961–971, Oct. 2001.
- [23] HSU, T.-Y. et al. Design of a wide-band frequency synthesizer based on TDC and DVC techniques. **IEEE Journal of Solid-State Circuits**, New York, v.37, n.10, p. 1244-1255, Oct. 2002.
- [24] TEMPORITI, E.; WELTIN-WU, C.; BALDI, D. et al. A 3 GHz fractional all-digital PLL with a 1.8 MHz bandwidth implementing spur reduction techniques. **IEEE Journal of Solid-State Circuits**, New York, v.44, n.3, p. 824-834, Mar. 2009.
- [25] SHU, K.; SÁNCHEZ-SINENCIO, E.; SILVA-MARTINEZ, J. et al. A 2.4-GHz monolithic fractional-N frequency synthesizer with robust phase-switching prescaler and loop capacitance multiplier. **IEEE Journal of Solid-State Circuits**, New York, v.38, n.6, p. 866-874, Jun. 2003.
- [26] MOON, S.T.; VALERO-LÓPEZ, A.Y.; SÁNCHEZ-SINENCIO, E. Fully integrated frequency synthesizers: A tutorial. **International Journal of High Speed Electronics and Systems**. [S. l.] v.15, n.2, p. 353-376, 2005.
- [27] ZHANG, B.; ALLEN, P.E.; HUARD, J.M. A fast switching PLL frequency synthesizer with an on-chip passive discrete-time loop filter in 0.25- μm CMOS. **IEEE Journal of Solid-State Circuits**, New York, v.38, n.6, p. 855-865, Jun. 2003.

- [28] ROBERT, M. What is RFID? **RFID Journal**. [S. l.] Disponível em: <<http://www.rfidjournal.com/>>. Acesso em 4 nov. de 2009.
- [29] KIM, S.C.; PARK, H.; LEE, D. et al. A novel design of an UHF RFID reader antenna for PDA. In: ASIA-PACIFIC MICROWAVE CONFERENCE, 2006, Yokohama, **Proceedings...** Yokohama: [S. n.], 2006.
- [30] GLIDDEN, R.; SCHROTER, J. Bringing long-range UHF RFID tags into main stream supply chain application. **RF-Design**. [S. l.] [S. n.], 2005.
- [31] RAZAVI, B. **RF Microelectronics**. New Jersey: Prentice-Hall, 1998.
- [32] GIERKINK, S.L.J.; VAN TUIJ, E. A coupled sawtooth oscillator combining low jitter with high control linearity. **IEEE Journal of Solid-State Circuits**, New York, v.37, n.6, p. 702–710, Jun. 2002.
- [33] LEUNG, B.H.; MCLEISH, D. Investigation of phase noise of ring oscillators with time-varying current and noise sources by time-scaling thermal noise. **IEEE Trans. Circuits Syst. I: Analog and Digital Signal Processing**, New York, v.51, n.10, p. 1926–1939, Oct. 2004.
- [34] RUFFIEUX, D. et al. A 1.2-mW RDS receiver for portable applications. **IEEE Journal of Solid-State Circuits**, New York, v.39, n.7, p. 995–1005, Jul. 2004.
- [35] JUNG, B.; HARJANI, R. High-frequency LC VCO design using capacitive degeneration. **IEEE Journal of Solid-State Circuits**, New York, v.39, n.12, p. 2359–2370, Dec. 2004.
- [36] CHEN, H.; GEIGER, R. Maximizing the oscillation frequency of CMOS VCOs. In: IEEE MIDWEST SYMPOSIUM ON CIRCUITS AND SYSTEMS, 43, 2000, Lansing, **Proceedings...** Lansing: [S. n.], 2000.p. 1248 – 1251.
- [37] JEONG, D.-Y.; CHAI, S.-H.; SONG, W.-C. et al. CMOS current controlled oscillator using Multiple-Feedback-Loop ring architecture. In: ISSCC DIG. TECH. PAPERS, 1997, San Francisco, **Proceedings...** San Francisco: [S. n.], 1997.p. 386–387.

- [38] MANEATIS, J.; HOROWITZ, M. Precise delay generation using coupled transistors. **IEEE Journal of Solid-State Circuits**, New York, v.28, n.12, p. 1273–1282, Dec. 1993.
- [39] LEE, S.-J.; KIM, B.; LEE, K. A novel high-speed ring oscillator for multiphase clock generation using negative Skewed-Delay scheme. **IEEE Journal of Solid-State Circuits**, New York, v.32, n.2, p. 289–291, Feb. 1997.
- [40] MONTAGNA, G. et al. A 35-mW 3.6-mm² fully integrated 0.18- μ m CMOS GPS radio. **IEEE Journal of Solid-State Circuits**, New York, v.38, n.7, p. 1163–1171, Jul. 2003.
- [41] RAZAVI, B. **Design of analog CMOS integrated circuits**. New York: McGraw-Hill, 2000.
- [42] LEUNG, B. H. A novel model on phase noise of ring oscillator based on last passage time. **IEEE Trans. Circuits Syst. I: Analog and Digital Signal Processing**, New York, v.51, n.3, p. 471–482, Mar. 2004.
- [43] PARK, C. H.; KIM, B. A low-noise, 900-MHz VCO in 0.6- μ m CMOS. **IEEE Journal of Solid-State Circuits**, New York, v.34, n.5, p. 586–591, May 1999.
- [44] MAGIEROWSKI, S. K.; ZUKOTYNSKI, S. CMOS LC-oscillator phase-noise analysis using nonlinear models. **IEEE Trans. Circuits Syst. I: Analog and Digital Signal Processing**, New York, v.51, n.4, p. 664–667, Apr 2004.
- [45] RAZAVI, B. A study of phase noise in CMOS oscillators. **IEEE Journal of Solid-State Circuits**, New York, v.31, p. 331–343, Mar. 1996.
- [46] HAJIMIRI, A.; LEE, T.H. A general theory of phase noise in electrical oscillators. **IEEE Journal of Solid-State Circuits**, New York, v.33, n.2, p. 179–194, Feb. 1998.
- [47] LEE, T.H.; HAJIMIRI, A. Oscillator phase noise: A tutorial. **IEEE Journal of Solid-State Circuits**, New York, v.35, n.3, p. 326–336, Mar. 2000.
- [48] LEESON, D.B. A simple model of feedback oscillator noise spectrum. **Proc. IEEE**, New York, v.54, p. 329–330, Feb. 1966.

- [49] KAERTNER, F.X. Determination of the correlation spectrum of oscillators with low noise. **IEEE Trans. Microwave Theory Tech.**, New York, v.37, n.1, p. 90–101, Jan. 1989.
- [50] LAI, X.; ROYCHOWDHURY, J. Automated oscillator macromodelling techniques for capturing amplitude variations and injection locking. In: IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER AIDED DESIGN, ICCAD-2004, San Jose, **Proceedings...** San Jose: [S. n.], 2004, p. 687-694.
- [51] VYTYAZ, I.; CARNES, J.; WU, T. et al. noise tolerant oscillator design using perturbation projection vector analysis. In: IEEE CUSTOM INTEGRATED CIRCUITS CONFERENCE, CICC 2008, San Jose. **Proceedings...** San Jose: [S. n.], 2008.p. 695-698.
- [52] GARDNER, F.M. Charge-pump phase-locked loops. **IEEE Transaction on Communication**, New York, v.28, p. 1849–1858, Nov. 1980.
- [53] VALERO-LOPEZ, A.Y. **Design of frequency synthesizers for short range wireless transceivers**. May 2004. 235 f. Tese (Doctor of Philosophy) - Texas A&M University, College Station, 2004. Disponível em: <[http : / /handle .tamu .edu /1969 .1 /108](http://handle.tamu.edu/1969.1/108)>. Acesso em: 2 dez. 2009.
- [54] CRANINCKX, J.; STEYAERT, M. A fully integrated CMOS DCS-1800 frequency synthesizer. **IEEE Journal of Solid-State Circuit**, New York, v.33, n.12, Dec. 1998.
- [55] STASZEWSKI, R.B.; HUNG, C.-M.; LEIPOLD, D. et al. A first multigigahertz digitally controlled oscillator for wireless applications. **IEEE Transactions on Microwave Theory and Techniques**, New York, v.51, n. 11, p. 2154-2164, Nov. 2003.
- [56] HE, J.; YANG, J.S.; KIM, Y. et al. System-level time-domain behavioral modeling for a mobile WIMAX transceiver. In: IEEE INTERNATIONAL BEHAVIORAL MODELING AND SIMULATION WORKSHOP, 2006, San Jose, **Proceedings...** San Jose: Set. 2006, p. 138-143.
- [57] Gardner, F.M. **Phaselock Techniques**, 2nd ed. New York: John-Wiley & Sons, 1979.
- [58] FRANKLIN, G.F.; POWELL, J.D.; WORKMAN, M.L. **Digital control of dynamic systems**. Massachusetts: Addison-Wesley Publishing Company, 1992.

- [59] MULLEN, B. Simplifying PLL design. **EE Times-India**. Feb. 2008. Disponível em: <<http://www.eetindia.co.in>>. Acesso em: 02 dez. 2009.
- [60] HUNG, C.-M.; O, K.K. A fully integrated 1.5-V 5.5-GHz CMOS phase-locked loop. **IEEE Journal of Solid-State Circuits**, New York, v.37, n.4, p. 521-525, Apr. 2002.
- [61] DO, M.A.; ZHAO, R.Y.; YEO, K.S. et al. New wideband/dualband CMOS LC voltage-controlled oscillator. **IEE Proc.-Circuits Devices Syst.**, New York, v.150, n.5, p. 453-459, Oct. 2003.
- [62] HEGAZI, E.; SJÖLAND, H.; ABIDI, A.A. A filtering technique to lower LC oscillator phase noise. **IEEE Journal of Solid-State Circuits**, New York, v.36, n.12, p. 1921-1930, Dec. 2001.
- [63] YANG, W.-R.; CAO, J.-L.; RAN, F. et al. A high-speed divide-by-32/33 frequency divider in 0.25 μ m CMOS technology. In: ICMMT 2004: 4TH INTERNATIONAL CONFERENCE ON MICROWAVE AND MILLIMETER WAVE TECHNOLOGY, 2004, Hardcover. **Proceedings...** Hardcover: [S. n.], 2004.p. 554-557.
- [64] DESIKACHARI, R. **High-speed CMOS dual-modulus prescalers for frequency synthesis**. Oct. 2003. 63 f. Dissertação (Master of Science) - Oregon State University, Corvallis, 2003. Disponível em: <web.engr.oregonstate.edu/~moon>. Acesso em: 02 dez. 2009.
- [65] LEVANTINO, S. et al. Phase noise in digital frequency dividers. **IEEE Journal of Solid-State Circuits**, New York, v.36, n.12, p. 1921-1930, Dec. 2001.
- [66] ARGÜELLO, A. M. G.; SOARES JÚNIOR, J. N. Implementation of a programmable high speed divider for a 2.4 GHz CMOS Integer-N frequency synthesizer. In: WORKSHOP IBERCHIP, 10, 2004, Cartagena de Índias. **Memórias ...** Cartagena de Índias: 2004, p. 1-8.
- [67] CRANINCKX, J.; STEYAERT, M. A 1.75-GHz/3-V dual-modulus divide-by-128/129 prescaler in 0.7- μ m CMOS. **IEEE J. Solid-State Circuits**, New York, v.31, p. 890-897, Jul 1996.

Apêndice A – Programas Usados para Projeto de Estabilidade

A1 – Programa usado para calcular as regiões das especificações do projeto de controle no lugar das raízes

```
po=input('Digite a porcentagem de overshoot desejada : ');
te=input('Digite o tempo de estabelecimento desejado : ');

a=log(po/100);
A=pi^2+2*a^2;

B=(pi-sqrt(A))/a    % B é o coeficiente de amortecimento.

Wn=1.8/te          % r determina o lugar geométrico no plano pra o tempo de
estabelecimento
```

A2 – Programa usado para traçar o lugar das raízes e projetar o controlador

```
num=input('Digite o numerador da planta continua : ');
den=input('Digite o denominador da planta continua : ');
n2=input('Digite os zeros do controlador continuo : ');
d2=input('Digite os polos do controlador continuo : ');

n2=poly(n2);
d2=poly(d2);          % nesse ponto os polos e zeros sao colocados na
forma de numerador e denominador do
                    % controlador, para um k de 0 a +infinito

num=conv(num,n2);
den=conv(den,d2);    % aqui é determinada a equação
caracteristica do sistema de malha aberta

sys=tf(num,den);
rlocus(sys);        % o root locus é traçado
axis([p_max-Wn Wn (-1.5*Wn) (1.5*Wn)])
sgrid(B,Wn)

[k1,polo]=rlocfind(num,den)
[k2,polo]=rlocfind(num,den) % o usuario do programa pode determinar a
magnitude de k em um ponto do root locus.
                    nesse caso, pode-se escolher um Kmin e um Kmax
k=(k1+k2)/2;

sys_cl=feedback(k*sys,1)
```

A3 – Programa que determina a equivalência discreta de um subsistema

```
function [nz,dz]=tfc2d(num,den,ts);
%
% Determina a equivalência Discreta do subsistema
%
%   u(kT) - D/A - G(s) - A/D - y(kT)
%
%   Sendo ts o período de amostragem
%
%   Y(z)/U(z) = H(z) = nz(z)/dz(z)
%
[a, b, c, d] = tf2ss(num,den);
[ad,bd] = c2d(a, b, ts);
[nz,dz] = ss2tf(ad, bd, c, d, 1);
```

A4 – Programa que usa a equação deduzida para a função transferência de malha aberta (5.16)

```
ts=input('Digite o periodo de amostragem : ');
kvco=input('Digite o numerador da planta continua : ');
d3=input('Digite os polos do controlador contínuo : ');
d2=input('Digite os polos do controlador discreto : ');%rootz(z-1)=1

k1=(3.75/4)*16*((2*pi*25e3)^2)/(kvco);
k2=15e-10*16*((2*pi*25e3)^2)*(0.25/4)/(kvco)/2/(2^8);

a=2*pi*d3;      %polo do laço analogico
d2=poly(d2);    % d2=[1 -1]

n1=ts*kvco;d1=[1 -1];

num1=conv(n1,k2);
den1=conv(d1,d2);

num=conv(kvco,k1);

a1=a*ts-1+exp((-a)*ts);a2=1-exp((-a)*ts)-a*ts*exp((-a)*ts);
num2=num*[a1 a2]/(a^2);
b1=[1 -1]; b2=[1 -exp((-a)*ts)];

den=[1 -(2+exp((-a)*ts)) (1+2*exp((-a)*ts)) -exp((-a)*ts)];

num3=conv(num2,[1 -1]);
num1=conv(num1,b2);

num=num3+[0 num1];

sys=tf(num,den,ts);

rltool(sys);          % o root locus é tracado
```

Apêndice B – Modelos Comportamentais Usados em Simulações

B1 – Código em Verilog-A do conjunto PFD e *charge pump* que foi calibrado automaticamente usando o Cadence

```
// (C) 2007 Cadence Design Systems, Inc. All rights reserved worldwide.
`include "discipline.h"
`include "constants.h"

module pllTTPfd_cp(out, ref, fb);
input ref, fb;
output out;
electrical ref, fb, out;
parameter real Imax=3.1248e-05;
parameter real Imis=0.0000e+00;
parameter real Ioffset=0.0000e+00;
parameter real refdelay=5.6474e-09;
parameter real fbdelay=7.2574e-09;
parameter real uptr=1.3527e-09;
parameter real downtr=2.2370e-09;
parameter real vtrans=1.0000e+00;
parameter integer dir=1 from [-1:1] exclude 0;
parameter jitter=0.0000e+00;

integer seed;
integer up_state, down_state;
real ref_dt, fb_dt, maxdt;
real ref_td, fb_td, time_reset;
real time_ref_cross, time_fb_cross;
real Iup, Idown, Iupmax, Idownmax, maxtr;
real Jee, tt;

analog begin
    @(initial_step) begin
        seed=911;
        up_state=0;
        down_state=0;
        Iupmax=-Imax*(1+0.5*Imis);
        Idownmax=Imax*(1-0.5*Imis);
        maxtr=max(uptr,downtr);
        Jee=jitter;
        tt=0.1e-12;
    end
end
```

```

@(cross(V(ref) - Vtrans, dir, tt )) begin
    up_state=1;
    ref_dt=Jee*$rdist_normal(seed,0,1);
    ref_td=refdelay+ref_dt;
    if (ref_td<0) ref_td=0;
    time_ref_cross=$abstime;
end

@(cross(V(fb) - Vtrans, dir, tt )) begin
    down_state=1;
    fb_dt=Jee*$rdist_normal(seed,0,1);
    fb_td=fbdelay+fb_dt;
    if (fb_td<0) fb_td=0;
    time_fb_cross=$abstime;
end

if (up_state==1 && down_state==1) begin
    maxdt=max(abs(ref_dt),abs(fb_dt));
    if (time_fb_cross >= time_ref_cross) time_reset=
time_fb_cross+maxtr+maxdt;
    if (time_fb_cross < time_ref_cross) time_reset=
time_ref_cross+maxtr+maxdt;
end

@(timer(time_reset)) begin
    if (up_state==1 && down_state==1) begin
        ref_td=refdelay;
        fb_td=fbdelay;
        up_state=0;
        down_state=0;
    end
end

Iup=transition(up_state*Iupmax,ref_td,uptr,uptr,tt);

Idown=transition(down_state*Idownmax,fb_td,downtr,downtr,tt);
I(out) <+ (Iup+Idown+Ioffset+1e-09*v(out));

end
endmodule

```

B2 – Código em Verilog-A usado para desempenhar a função de converter um barramento digital em tensão

```
// Verilog-A for synt_blocks, DAC_ideal, veriloga

`include "constants.vams"
`include "disciplines.vams"

module DAC_ideal(IN, OUT);

    parameter integer bits = 22;
    parameter real thr=0.5;
    integer a[0:bits-1];
    real b[0:bits-1];
    integer i;

    input [21:0] IN;
    output OUT;
    electrical [21:0] IN;
    electrical OUT;

analog begin
    @(initial_step) begin
        for(i=0;i<=bits-1;i=i+1)begin
            b[i]=pow(2,i)*(1.4476e-6);
        end
    end
end

    if(V(IN[0]) > 0.5) a[0]=1; else a[0]=0;
    if(V(IN[1]) > 0.5) a[1]=1; else a[1]=0;
    if(V(IN[2]) > 0.5) a[2]=1; else a[2]=0;
    if(V(IN[3]) > 0.5) a[3]=1; else a[3]=0;
    if(V(IN[4]) > 0.5) a[4]=1; else a[4]=0;
    if(V(IN[5]) > 0.5) a[5]=1; else a[5]=0;
    if(V(IN[6]) > 0.5) a[6]=1; else a[6]=0;
    if(V(IN[7]) > 0.5) a[7]=1; else a[7]=0;
    if(V(IN[8]) > 0.5) a[8]=1; else a[8]=0;
    if(V(IN[9]) > 0.5) a[9]=1; else a[9]=0;
    if(V(IN[10]) > 0.5) a[10]=1; else a[10]=0;
    if(V(IN[11]) > 0.5) a[11]=1; else a[11]=0;
    if(V(IN[12]) > 0.5) a[12]=1; else a[12]=0;
    if(V(IN[13]) > 0.5) a[13]=1; else a[13]=0;
    if(V(IN[14]) > 0.5) a[14]=1; else a[14]=0;
    if(V(IN[15]) > 0.5) a[15]=1; else a[15]=0;
    if(V(IN[16]) > 0.5) a[16]=1; else a[16]=0;
```

```

    if(V(IN[17]) > 0.5) a[17]=1; else a[17]=0;
    if(V(IN[18]) > 0.5) a[18]=1; else a[18]=0;
    if(V(IN[19]) > 0.5) a[19]=1; else a[19]=0;
    if(V(IN[20]) > 0.5) a[20]=1; else a[20]=0;
    if(V(IN[21]) > 0.5) a[21]=1; else a[21]=0;

V(OUT)<+(a[0]*b[0]+a[1]*b[1]+a[2]*b[2]+a[3]*b[3]+a[4]*b[4]+a[5]*b[5]+a[6]*b
[6]+a[7]*b[7]+a[8]*b[8]+a[9]*b[9]+a[10]*b[10]+a[11]*b[11]+a[12]*b[12]+a[13]
*b[13]+a[14]*b[14]+a[15]*b[15]+a[16]*b[16]+a[17]*b[17]+a[18]*b[18]+a[19]*b[
19]+a[20]*b[20]+a[21]*b[21]);

end
endmodule

```

B3 – Código em Verilog-A das chaves utilizadas nos circuitos analógicos

```

// verilogA for synt_blocks, seletor, veriloga

`include "constants.vams"
`include "disciplines.vams"

module seletor (IN1, IN2, EN, OUT);
parameter real vtrans = 1;
inout IN1, IN2, OUT;
electrical IN1, IN2, EN, OUT;

analog begin

    if(V(EN) > vtrans) V(OUT) <+ V(IN2);
    else V(OUT) <+ V(IN1);
end

endmodule

```

B4 – Código em Verilog-A do somador de sinais

```

`include "discipline.h"
`include "constants.h"

module adder(sigin1, sigin2, sigout);
input sigin1, sigin2;

```

```

output sigout;
electrical sigin1, sigin2, sigout;
parameter real k1 = 1;
parameter real k2 = 1;

    analog
        v(sigout) <+ k1*v(sigin1) + k2*v(sigin2);

endmodule

```

B5 – Código em Verilog-A da função Z-1

```

// VerilogA for wlanRficLib, z_inv, veriloga

`include "constants.vams"
`include "disciplines.vams"

module z_inv_digi(in, out,clk);
input in, clk;
output out;
electrical in, out, clk;

parameter real vtrans=1;
parameter real tr=10p; //rise,fall time
parameter real tt=0.1p; //time tolerance
parameter real td=10p; //delay time
parameter real init_outval=0; //initial output value

real outval;

analog begin

@(initial_step) begin
    outval=init_outval;
end

@ (cross(v(clk)-vtrans, +1, tt)) outval=v(in);

v(out) <+ transition(outval,td,tr,tr,tt);
end
endmodule

```

B1 – Código em Verilog-A das portas lógicas calibradas manualmente com as células em nível de transistor da biblioteca da AMS

```

`include "discipline.h"
`include "constants.h"

module not_gate(vin, vout);
input vin;
output vout;
electrical vin, vout;
parameter real vlogic_high = 2;
parameter real vlogic_low = 0;
parameter real vtrans = 1;
parameter real tdel = 10p from [0:inf);
parameter real trise = 50p from (0:inf);
parameter real tfall = 50p from (0:inf);

    real vout_val;
    integer logic_in;

    analog begin

        @ ( initial_step ) begin
            if (vlogic_high < vlogic_low) begin
                $display("Range specification error. vlogic_high = (%E) less
than vlogic_low = (%E).\n", vlogic_high, vlogic_low );
                $finish;
            end
            if (vtrans > vlogic_high || vtrans < vlogic_low) begin
                $display("Inconsistent $threshold specification w/logic
family.\n");
            end
        end

        logic_in = v(vin) > vtrans;

        @ (cross(v(vin) - vtrans, 1)) logic_in = 1;
        @ (cross(v(vin) - vtrans, -1)) logic_in = 0;

        //
        // define the logic function.
        //
        vout_val = !(logic_in) ? vlogic_high : vlogic_low;

        v(vout) <+ transition( vout_val, tdel, trise, tfall);

```



```

    end
endmodule

//-----

`include "discipline.h"
`include "constants.h"

module nand_gate(vin1, vin2, vout);
input vin1, vin2;
output vout;
electrical vin1, vin2, vout;
parameter real vlogic_high = 2;
parameter real vlogic_low = 0;
parameter real vtrans = 1;
parameter real tdel = 50p from [0:inf);
parameter real trise = 200p from (0:inf);
parameter real tfall = 50p from (0:inf);

    real vout_val;
    integer logic1, logic2;

    analog begin

        @ ( initial_step ) begin
            if (vlogic_high < vlogic_low) begin
                $display("Range specification error.  vlogic_high = (%E) less
than vlogic_low = (%E).\n", vlogic_high, vlogic_low );
                $finish;
            end
            if (vtrans > vlogic_high || vtrans < vlogic_low) begin
                $display("Inconsistent $threshold specification w/logic
family.\n");
            end
        end

        logic1 = v(vin1) > vtrans;
        logic2 = v(vin2) > vtrans;

        @ (cross(v(vin1) - vtrans, 1)) logic1 = 1;
        @ (cross(v(vin1) - vtrans, -1)) logic1 = 0;

        @ (cross(v(vin2) - vtrans, 1)) logic2 = 1;
        @ (cross(v(vin2) - vtrans, -1)) logic2 = 0;
    end
endmodule

```

```

//
// define the logic function.
//
vout_val = !(logic1 && logic2) ? vlogic_high : vlogic_low;

V(vout) <+ transition( vout_val, tdel, trise, tfall);
end
endmodule

//-----

`include "discipline.h"
`include "constants.h"

module nor_gate(vin1, vin2, vout);
input vin1, vin2;
output vout;
electrical vin1, vin2, vout;
parameter real vlogic_high = 2;
parameter real vlogic_low = 0;
parameter real vtrans = 1;
parameter real tdel = 50p from [0:inf);
parameter real trise = 200p from (0:inf);
parameter real tfall = 50p from (0:inf);

real vout_val;
integer logic1, logic2;

analog begin

    @ ( initial_step ) begin
        if (vlogic_high < vlogic_low) begin
            $display("Range specification error. vlogic_high = (%E) less
than vlogic_low = (%E).\n", vlogic_high, vlogic_low );
            $finish;
        end
        if (vtrans > vlogic_high || vtrans < vlogic_low) begin
            $display("Inconsistent $threshold specification w/logic
family.\n");
        end
    end

    logic1 = V(vin1) > vtrans;
    logic2 = V(vin2) > vtrans;

    @ (cross(V(vin1) - vtrans, 1)) logic1 = 1;

```

```

    @ (cross(V(vin1) - vtrans, -1)) logic1 = 0;

    @ (cross(V(vin2) - vtrans, 1)) logic2 = 1;
    @ (cross(V(vin2) - vtrans, -1)) logic2 = 0;

    //
    // define the logic function.
    //
    vout_val = (!(logic1 || logic2)) ? vlogic_high : vlogic_low;

    V(vout) <+ transition( vout_val, tdel, trise, tfall);
end
endmodule

//-----

`include "discipline.h"
`include "constants.h"

module xnor_gate(vin1, vin2, vout);
input vin1, vin2;
output vout;
electrical vin1, vin2, vout;
parameter real vlogic_high = 2;
parameter real vlogic_low = 0;
parameter real vtrans = 1;
parameter real tdel = 50p from [0:inf);
parameter real trise = 200p from (0:inf);
parameter real tfall = 50p from (0:inf);

    real vout_val;
    integer logic1, logic2;

    analog begin
        @ ( initial_step ) begin
            if (vlogic_high < vlogic_low) begin
                $display("Range specification error.  vlogic_high = (%E) less
than vlogic_low = (%E).\n", vlogic_high, vlogic_low );
                $finish;
            end
            if (vtrans > vlogic_high || vtrans < vlogic_low) begin
                $display("Inconsistent $threshold specification w/logic
family.\n");
            end
        end
    end
end

```

```

    logic1 = V(vin1) > vtrans;
    logic2 = V(vin2) > vtrans;

    @ (cross(V(vin1) - vtrans, 1)) logic1 = 1;
    @ (cross(V(vin1) - vtrans, -1)) logic1 = 0;

    @ (cross(V(vin2) - vtrans, 1)) logic2 = 1;
    @ (cross(V(vin2) - vtrans, -1)) logic2 = 0;

    //
    // define the logic function.
    //
    vout_val = (!(logic1 ^ logic2)) ? vlogic_high : vlogic_low;

    V(vout) <+ transition( vout_val, tdel, trise, tfall);
end
endmodule

//-----

`include "discipline.h"
`include "constants.h"

module jk_clk_ff(vin_j, vin_k, vclk, vout_q, vout_qbar, vrn);
input vin_j, vin_k, vclk, vrn;
output vout_q, vout_qbar;
electrical vin_j, vin_k, vclk, vout_q, vout_qbar, vrn;
parameter real vlogic_high = 2;
parameter real vlogic_low = 0;
parameter real vtrans_clk = 1;
parameter real vtrans = 1;
parameter real tdel = 650p from [0:inf);
parameter real trise = 50p from (0:inf);
parameter real tfall = 200p from (0:inf);

integer q;

analog begin
    @ (cross( V(vclk) - vtrans_clk, +1 )) begin
        if (V(vin_j) > vtrans && V(vin_k) > vtrans) begin
            q = !q;
        end else if (V(vin_j) > vtrans && V(vin_k) < vtrans) begin
            q = 1;
        end else if (V(vin_j) < vtrans && V(vin_k) > vtrans) begin

```

