

Universidade de São Paulo

João Alberto Baccarin Robles Tardelli

**Otimização de osciladores CMOS por
algoritmos meta-heurísticos**

Orientador:

Prof. Dr. João Navarro Soares Junior

São Carlos

2014

João Alberto Baccarin Robles Tardelli

Otimização de osciladores CMOS por algoritmos meta-heurísticos

Trabalho de Conclusão de Curso
apresentado à Escola de Engenharia de São
Carlos, da Universidade de São Paulo

Curso de Engenharia Elétrica com ênfase
em Eletrônica

ORIENTADOR: Prof. Dr. João Navarro Soares Junior

São Carlos
2014

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

T181o Tardelli, João Alberto Baccarin Robles
Otimização de osciladores CMOS por algoritmos
meta-heurísticos / João Alberto Baccarin Robles
Tardelli; orientador João Navarro Soares Junior. São
Carlos, 2014.

Monografia (Graduação em Engenharia Elétrica com
ênfase em Eletrônica) -- Escola de Engenharia de São
Carlos da Universidade de São Paulo, 2014.

1. algoritmos meta-heurísticos. 2. otimização. 3.
projeto. 4. circuitos CMOS. 5. osciladores. I. Título.

FOLHA DE APROVAÇÃO

Nome: João Alberto Baccarin Robles Tardelli

Título: "Otimização de Osciladores CMOS por algoritmos metaheurísticos"

Trabalho de Conclusão de Curso defendido e aprovado
em 30/06/2014,

com NOTA 8,7 (oito, sete), pela Comissão Julgadora:

Prof. Dr. João Navarro Soares Júnior - (Orientador - SEL/EESC/USP)

*Prof. Dr. José Roberto Boffino de Almeida Monteiro -
(SEL/EESC/USP)*

Prof. Dr. Maximilian Luppe - (SEL/EESC/USP)

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Associado Homero Schiabel

*Dedico este trabalho à minha família,
em especial, à minha querida mãe.*

Agradecimentos

Agradeço a minha família que me deu todo suporte necessário para eu enfrentar todos os desafios encontrados em minha jornada até aqui; em especial agradeço a minha mãe a quem sou eternamente grato.

Agradeço a todos os meus professores desde meu primeiro ciclo no Ensino Infantil até aos Mestres e Doutores que foram responsáveis pela minha formação acadêmica.

Agradeço, em especial, ao Professor Doutor João Navarro Soares Jr. responsável por minha orientação nesse trabalho e pelo incentivo a minha paixão pela área de microeletrônica.

Agradeço aos meus amigos e colegas que foram e são minha família nos momentos em que me encontro distante da mesma.

Agradeço ao CNPq pelo apoio econômico oferecido para esse trabalho e a Universidade de São Paulo como instituição por me possibilitar viver essa jornada de aprendizado.

*“Em alta frequências,
os amplificadores oscilam e os osciladores não.”*

Autor desconhecido.

*“In the high-frequency world,
amplifiers oscillate and oscillators don’t.”*

No original em inglês.

Resumo

No presente trabalho é feita a implementação de uma ferramenta para projeto de circuitos osciladores CMOS, através da utilização de algoritmos meta-heurísticos e simulações computacionais na plataforma Spice. Os algoritmos meta-heurísticos implementados em MatLab e aplicados foram os seguintes: Otimização por Enxame de Partículas (*particle swarm optimization*), Arrefecimento Simulado (*simulated annealing*) e Busca Padrão (*pattern search*). Testes foram realizados com osciladores em anel com três e cinco inversores, osciladores a cristal de 1,0 MHz e de 32768 Hz, além de osciladores controlados por tensão. A tecnologia aplicada foi a tecnologia CMOS 0,35 μm da AMS (Austria Micro Systems). No projeto foram analisados a frequência de oscilação, amplitude do sinal de saída, ruído de fase, potência consumida, e as distorções harmônicas. Um dos osciladores obtidos para 32768 Hz apresentou amplitude de saída superior a 2,0 V e potência consumida inferior a 3,0 μW para tensão de alimentação de 3,0 V. O circuito VCO (*voltage controlled oscillator*) projetado apresentou uma faixa de frequência de 0,8 GHz a 1,1GHz. Esses e outros resultados comprovaram a eficácia da utilização de algoritmos meta-heurísticos no projeto e na otimização de circuitos osciladores CMOS, porém estudos mais aprofundados para a criação de um método robusto para o projeto automatizado ainda são necessários.

Palavras Chaves: algoritmos meta-heurísticos, otimização, projeto, circuitos CMOS, osciladores.

Abstract

In this work, the implementation of a tool for design of CMOS oscillator circuits through metaheuristic algorithms and computational simulations on Spice platform is developed. The metaheuristic algorithms implemented in MATLAB and applied were: the particle swarm optimization, simulated annealing, and pattern search. Tests were performed in ring oscillators with three and five inverters, crystal oscillators of 1.0 MHz and 32768 Hz, and voltage-controlled oscillators. The applied technology was the AMS (Austria Micro Systems) CMOS 0.35 μm . In the design were analyzed the oscillation frequency, the output signal amplitude, the phase noise, the power consumption, and the harmonic distortions. One of the designs obtained for a 32768 Hz oscillator presented output amplitude greater than 2.0 V and power consumption less than 3.0 μW for a 3.0 V supply voltage. The VCO (voltage controlled oscillator) circuit designed obtained a frequency range from 0.8 GHz to 1.1 GHz. These and other results have confirmed the effectiveness of using metaheuristic algorithms in the design and optimization of CMOS oscillator circuits, however further studies to the creation of a robust method for the automated design are still needed.

Key-words: metaheuristics algorithm, optimization, project, CMOS circuits, oscillators.

Sumário

1	Introdução.....	1
1.1	Objetivos	3
2	Conceitos básicos.....	4
2.1	Transistores MOS	4
2.2	Circuitos osciladores	8
2.2.1	Circuitos osciladores controlados por tensão	13
2.2.2	Capacitâncias variáveis	14
2.2.3	Parâmetros de um oscilador	17
2.3	Algoritmos meta-heurísticos	19
2.3.1	Otimização por Enxame de Partículas	21
2.3.2	Arrefecimento Simulado.....	22
2.3.3	Busca Padrão	23
3	Metodologia	25
3.1	Plataforma de projeto	25
3.2	Ferramentas utilizadas	30
3.2.1	MatLab.....	30
3.2.2	HSpice RF	31
3.2	Circuitos projetados.....	33
3.2.1	Circuitos osciladores em Anel	33
3.2.2	Circuitos osciladores a cristal.....	35
3.2.3	Circuito oscilador controlado por tensão	39
3.3	Função objetivo	40
3.3.1	Frequência.....	41
3.3.2	Excursão do sinal de saída	42
3.3.3	Potência consumida.....	44
3.3.4	Ruído de fase	44
3.3.5	Distorção harmônica	45
3.3.6	Área	47
3.3.7	<i>Jitter</i>	47
3.4	Conjuntos de simulações	48
4	Resultados.....	52
4.1	Circuito oscilador com três inversores em anel.....	52
4.2	Circuito oscilador com cinco inversores em anel	54
4.3	Circuito oscilador com cristal de 1,0 MHz	57
4.4	Circuito oscilador com cristal de relógio (32768 Hz)	59
4.4.1	Refinamento da solução por Busca Padrão	61

4.4.2 Refinamento da solução por Arrefecimento Simulado.....	62
4.4.2 Análise do circuito final	62
4.5 Circuito oscilador controlado por tensão.....	64
4.6 Discussão.....	67
5 Conclusão.....	70
Referências	72
Apêndice A	75
Apêndice B	79
Apêndice C	87
Apêndice D.....	98

Índice de figuras

Figura 1: Corte horizontal de um transistor NMOS.....	6
Figura 2: Símbolos e convenções para transistores MOS.....	6
Figura 3: Diagrama de um sistema realimentado.....	10
Figura 4: Diagrama de blocos e função de frequência de um circuito VCO.....	13
Figura 5: Capacitor MOS em topologia livre (BUNCH, RAMAN; 2003, p.1327).....	16
Figura 6: Capacitor MOS na topologia com operação na região de inversão (BUNCH, RAMAN, 2003, p.1327).....	16
Figura 7: Capacitor MOS na topologia com operação na região de acumulação (BUNCH, RAMAN; 2003, p.1327).	17
Figura 8: Fluxograma de um algoritmo meta-heurístico genérico.....	20
Figura 9: Velocidade e posição de uma partícula.....	22
Figura 10: Interface Gráfica de dados.....	27
Figura 11: Interface Gráfica de execução.	27
Figura 12: Fluxograma da plataforma de projeto.....	28
Figura 13: Circuito oscilador em anel com três inversores. Estão indicados na figura os valores de W e L dos transistores.....	34
Figura 14: Circuito oscilador em anel com cinco inversores. Estão indicados na figura os valores de W e L dos transistores	35
Figura 15: Modelo de quatro elementos do cristal.....	36
Figura 16: Circuito oscilador a cristal. Estão indicados na figura os valores de W, L e M dos transistores	37
Figura 17: Circuito oscilador a cristal modificado. Estão indicados na figura os valores de W, L e M dos transistores	38
Figura 18: Circuito VCO utilizado. Estão indicados na figura os valores de W e L dos transistores.....	40
Figura 19: Sinal de saída do circuito oscilador com três inversores pelo tempo obtido pela análise SNOSC.....	54
Figura 20: Sinal de saída do circuito oscilador com cinco inversores pelo tempo obtido pela análise SNOSC.....	56
Figura 21: Sinal de saída do circuito com cristal de 1,0 MHz pelo tempo obtido pela análise SNOSC.....	59
Figura 22: Sinal de saída do circuito com cristal de 32768 Hz pelo tempo obtido pela análise SNOSC.....	63
Figura 23: Sinal de saída do circuito VCO pelo tempo obtido pela análise HBOSC.	67

Índice de tabelas

Tabela 1: Parâmetros elétricos dos transistores NMOS.....	33
Tabela 2: Parâmetros elétricos dos transistores PMOS.....	33
Tabela 3: Variáveis de ajuste do circuito oscilador em anel com três inversores.	35
Tabela 4: Variáveis de ajuste do circuito oscilador em anel com cinco inversores.	35
Tabela 5: Parâmetros dos modelos de quatro elementos dos cristais.....	36
Tabela 6: Variáveis de ajuste dos circuitos osciladores a cristal.	37
Tabela 7: Variáveis de ajuste do circuito oscilador controlado por tensão.....	39
Tabela 8: Pesos da função objetivo do circuito oscilador com três inversores em anel.	52
Tabela 9: Valores dos parâmetros obtidos no projeto do circuito oscilador com três inversores em anel.	53
Tabela 10: Valores dos critérios de desempenho do circuito oscilador com três inversores em anel projetado. Vmax e Vmin são a máxima e mínima tensão em out.	53
Tabela 11: Resultados das análises de transiente TRAN e balanço harmônico HBOSC. Vmax e Vmin são a máxima e mínima tensão em out.....	53
Tabela 12: Pesos da função objetivo do circuito oscilador com cinco inversores em anel.....	55
Tabela 13: Valores dos parâmetros obtidos no projeto do circuito oscilador com cinco inversores em anel.	55
Tabela 14: Valores dos critérios de desempenho do circuito oscilador com cinco inversores em anel projetado. Vmax e Vmin são a máxima e mínima tensão em out.	55
Tabela 15: Resultados das análises de transiente TRAN e balanço harmônico HBOSC. Vmax e Vmin são a máxima e mínima tensão em out.....	56
Tabela 16: Pesos da função objetivo do circuito oscilador com cristal de 1,0 MHz.	57
Tabela 17: Valores dos parâmetros obtidos no projeto do circuito oscilador com cristal de 1,0 MHz projetado.....	58
Tabela 18: Valores dos critérios de desempenho do circuito oscilador com cristal de 1,0 MHz projetado. Vmax e Vmin são a máxima e mínima tensão em out.	58
Tabela 19: Resultado das análises de transiente TRAN e balanço harmônico HBOSC. Vmax e Vmin são a máxima e mínima tensão em out.....	58
Tabela 20: Pesos da função objetivo do circuito oscilador com cristal de 32768 Hz. ..	60
Tabela 21: Valores dos parâmetros obtidos no projeto do circuito oscilador com cristal de 32768 Hz projetado. Vmax e Vmin são a máxima e mínima tensão em out.	60
Tabela 22: Valores dos critérios de desempenho do circuito oscilador com cristal de 32768 Hz projetado. Vmax e Vmin são a máxima e mínima tensão em out.	61

Tabela 23: Valores dos parâmetros obtidos pelo refinamento do circuito oscilador com cristal de 32768 Hz por Busca Padrão.	61
Tabela 24: Valores dos critérios de desempenho do circuito refinado por Busca Padrão. Vmax e Vmin são a máxima e mínima tensão em out.	61
Tabela 25: Valores dos parâmetros obtidos pelo refinamento do circuito oscilador com cristal de 32768 Hz por Arrefecimento Simulado. Vmax e Vmin são a máxima e mínima tensão em out.	62
Tabela 26: Valores dos critérios de desempenho do circuito refinado por Arrefecimento Simulado. Vmax e Vmin são a máxima e mínima tensão em out.	62
Tabela 27: Resultado das análises de transiente TRAN e balanço harmônico HOSC da rodada C. Vmax e Vmin são a máxima e mínima tensão em out.	63
Tabela 28: Pesos da função objetivo do circuito oscilador controlado por tensão.	65
Tabela 29: Valores dos parâmetros obtidos no projeto do circuito oscilador controlado por tensão projetado.	65
Tabela 30: Valores dos critérios de desempenho do circuito oscilador controlado por tensão projetado. Vmax e Vmin são a máxima e mínima diferença de tensão entre o e d.	66
Tabela 31: Resultado das análises de transiente TRAN. Vmax e Vmin são a máxima e mínima diferença de tensão entre o e d.	66
Tabela 32: Resultado da análise SNOSC. Vmax e Vmin são a máxima e mínima diferença de tensão entre o e d.	66
Tabela 33: Informações básicas do computador utilizado na execução dos algoritmos meta-heurísticos.	68

1 Introdução

A sociedade contemporânea está imersa em um mundo altamente tecnológico, resultado de décadas de avanços na microeletrônica desde a invenção do primeiro circuito integrado, onde vários dispositivos eletrônicos são fabricados em um mesmo substrato. A integração de circuitos foi inicialmente idealizada na década de 50 e, após dezenas de anos de desenvolvimento, hoje possibilita construção de processadores com bilhões de componentes em tecnologias com dimensões mínimas inferiores a 22 nm. A tecnologia de fabricação de circuitos eletrônicos predominante hoje é a tecnologia CMOS (Complementary Metal Oxide Semiconductor) e cerca de 95% dos circuitos integrados são implementados através dela (BAKER, 2008, p.7; SEMICONDUCTOR INDUSTRY ASSOCIATION - SIA, 2013),

O uso intensivo da tecnologia CMOS se deve a suas vantagens de produção e desempenho. Dentre elas se destacam: baixo custo de produção se comparada a outras tecnologias; facilidade de projeto de circuitos digitais; área reduzida do circuito devido à redução contínua das dimensões das tecnologias; e baixo consumo de potência devido à baixa dissipação de potência presente apenas no chaveamento dos transistores (RAZAVI, 2001, p.6; SASSI, 2013, p.21).

Na década de 80, diversos especialistas predisseram a extinção dos circuitos analógicos, devido ao aumento de desempenho dos algoritmos de processamento digital de sinais e o constante progresso na integração de circuitos digitais. Diversas funções que eram realizadas por circuitos analógicos passaram a ser realizadas por meio de circuitos digitais. Assim, na visão daqueles especialistas, com recursos de processamento digital de sinais suficientes, o processamento analógico não seria mais necessário (RAZAVI, 2001, p.1).

Apesar de hoje a maioria dos circuitos eletrônicos integrados possuírem caráter digital dominante, a importância da eletrônica analógica é incontestável. Isso se deve ao fato de que o mundo real possui caráter analógico. Dessa forma, na aquisição e saída de sinais para o mundo real é ainda necessária à utilização de circuitos eletrônicos analógicos. Mas a utilização desse tipo de circuitos não se limita a interface de sistemas com o mundo real; circuitos de inicialização, reguladores de tensões e correntes, circuitos de relógio (*clock*) entre outros, são analógicos. Segundo Razavi (2001, p.1), circuitos analógicos têm provado serem fundamentais para sistemas complexos e de alto desempenho.

Entre as aplicações em que circuitos eletrônicos analógicos são de grande importância estão as de telecomunicações, que fazem parte de uma área da engenharia

e da economia que movimenta recursos consideráveis. Nessas aplicações, circuitos osciladores possuem vital importância, já que seu desempenho influencia o desempenho dos sistemas como um todo. Em decorrência disso, o projeto e a otimização de circuitos osciladores é essencial para o perfeito funcionamento dos circuitos aplicados em telecomunicações.

A complexidade do projeto de circuitos eletrônicos depende das relações entre os parâmetros físicos e elétricos dos dispositivos e as características elétricas do circuito, assim como do número de dispositivos ativos, transistores, diodos, e passivos, resistores e capacitores, presentes. Os modelos matemáticos que descrevem o comportamento das grandezas elétricas dos transistores devem possuir suficiente simplicidade para serem utilizados manualmente em análises e sínteses de circuitos. Porém, quando maior for a simplificação dos modelos aplicados, menor será sua fidelidade ao comportamento real do transistor. Em consequência, equações mais ou menos complexas são necessárias.

Para a maior parte dos circuitos o número de equações que o descrevem é insuficiente para determinar uma solução única, fazendo com que a experiência do projetista seja um fator decisivo na escolha da solução aplicada. A complexidade das equações e sua possível indeterminação tornam os projetos de circuitos eletrônicos analógicos custosos, demorados e incertos.

Como alternativa ao projeto tradicional de circuitos eletrônicos analógicos, pesquisadores e projetistas tem alçado mão de métodos meta-heurísticos ou de programação geométrica para o projeto de circuitos de tal maneira que eles satisfaçam da melhor maneira possível às características desejadas.

O objetivo deste trabalho é montar uma plataforma de projetos utilizando algoritmos meta-heurísticos para o projeto de circuitos osciladores, deixando aberta ao usuário a escolha dos algoritmos, das especificações do projeto e da própria topologia que será utilizada.

Desenvolvida a plataforma de projetos, foi realizado o projeto de circuitos osciladores com frequência fixa em diferentes topologias – com três inversores em anel, com cinco inversores em anel, com cristal de 1,0 MHz, com cristal de 32768 Hz - e o projeto de um circuito oscilador controlado por tensão (VCO – *Voltage Controlled Oscillator*). As análises de transiente, de balanço harmônico e de *Shooting Newton* foram utilizadas para avaliar o desempenho dos circuitos nas simulações elétricas realizadas no *software* HSpice RF.

A inspiração desse trabalho é possibilitar a criação de uma plataforma de projeto autônoma de circuitos osciladores através da análise das simulações elétricas e dos algoritmos meta-heurísticos disponíveis para otimização, da escolha dos critérios de

desempenho para circuitos osciladores e da criação das funções avaliadoras para cada critério de desempenho.

Obteve-se o projeto de circuitos osciladores com inversores em anel com frequência de oscilação de 433 MHz e de 900 MHz. Os circuitos osciladores com cristais apresentaram baixo consumo de potência, em especial o circuito com cristal de relógio que apresentou potência de 2,95 μW . O circuito VCO projetado apresentou faixa de frequência de 0,8 GHz a 1,1 GHz.

1.1 Objetivos

O objetivo desse trabalho é otimizar cinco topologias de circuitos osciladores através de algoritmos meta-heurísticos e simulações elétricas. As topologias de circuitos osciladores otimizados são: com três inversores em anel, com cinco inversores em anel, com cristal de 1,0 MHz, com cristal de 32768 Hz, e com frequência controlada por tensão.

Para realizar as otimizações propostas, é proposto como objetivo desenvolver uma plataforma de projetos de circuitos eletrônicos analógicos osciladores com frequência fixa ou com frequência controlada por tensão que permita ao usuário:

- Escolher a topologia desejada para o circuito dentre as já citadas;
- Especificar o desempenho do circuito;
- Estabelecer os limites das dimensões dos elementos do circuito;
- Escolher o algoritmo meta-heurístico a ser utilizado na otimização;
- Configurar os pesos de uma função objetivo;
- Adicionar novas topologias.

Como objetivos específicos desse trabalho foram feitos: o projeto de osciladores com cristal de 32,768 kHz com potência abaixo de 10 μW ; e o projeto de um circuito oscilador controlado por tensão com faixa de frequência de 0,8 GHz a 1,1 GHz. A tecnologia aplicada foi a tecnologia CMOS 0.35 μm da AMS (*Austria Micro Systems*).

No Capítulo 2 serão apresentados os conceitos básicos utilizados neste trabalho acerca de algoritmos meta-heurísticos e de circuitos osciladores MOS. No Capítulo 3 serão apresentadas a metodologia e as ferramentas utilizadas e as desenvolvidas. No Capítulo 4 são descritos e discutidos os resultados obtidos nos projetos realizados. Por fim, no Capítulo 5 são apresentados a conclusão e os trabalhos futuros a serem realizados.

2 Conceitos básicos

2.1 Transistores MOS

Transistores são componentes eletrônicos utilizados como amplificadores de sinais elétricos ou interruptores eletroeletrônicos. Esse trabalho se baseou na tecnologia de transistores CMOS, ou seja, em uma tecnologia que permite a construção de transistores NMOS (transistores MOS com canal do tipo n) e PMOS (transistores MOS com canal do tipo p) em um mesmo circuito. A ferramenta desenvolvida, porém, suporta qualquer tecnologia desde que possua modelos de simulação para dispositivos suportados pelo simulador elétrico tipo SPICE (do inglês *Simulated Program with Integrated Circuits Emphasis*) utilizado, no caso deste trabalho o HSpice RF (SYNOPTIS, 2008a, 2008b, 2008c, 2008d, 2014).

Os conceitos básicos da tecnologia CMOS foram introduzidos por Weimer, *RCA*, em 1962 e por Wanlass, *Fairchild*, em 1963 (BAKER, 2008, p.6; WANLASS, 1967). A ideia de que um circuito poderia conter tanto transistores NMOS quanto transistores PMOS era, no entanto, muito nova para a imaturidade da tecnologia MOS (apenas em 1959, Atalla e Kahng, da *Bell Labs.*, conseguiram a operação de um transistor MOS (MELLIAR-SMITH, 1998)) e, devido à crescente popularidade dos transistores de junção bipolar (BJT, do inglês, *bipolar junction transistor*) a tecnologia CMOS não era vista como uma opção viável (BAKER, 2008, p.6).

O primeiro circuito CMOS comercial foi desenvolvido pela equipe liderada por Albert Medwin da *RCA* em 1968. No início, os circuitos CMOS eram de baixa potência, porém lentos para servirem como alternativa aos circuitos lógicos BJT que usavam a lógica digital TTL (do inglês, *Transistor-Transistor logic*). Durante a década de 70, fabricantes de relógios migraram para a tecnologia CMOS devido à importância do baixo consumo para maior duração de bateria e devido ao baixo nível de computação necessário para os relógios. Nesse período, a tecnologia CMOS começou também a ser aplicada no desenvolvimento de processadores. Atualmente, o mercado de circuitos integrados é predominantemente baseado na tecnologia CMOS, como já citado acima.

No estudo e projeto de circuitos integrados, há duas abordagens distintas utilizadas. Na primeira abordagem é feito o estudo do funcionamento interno dos dispositivos, realizado através da mecânica quântica, da física dos estados sólidos e da física de semicondutores, a modelagem dos dispositivos e finalmente o projeto dos circuitos. Na outra abordagem cada dispositivo semicondutor é tratado como uma caixa-

preta, cujo comportamento é descrito pelas correntes e tensões elétricas de seus terminais.

Segundo Behzad Razavi (2001, p.9), nenhuma dessas abordagens é ótima. Na primeira abordagem, o entendimento de toda a física dos dispositivos semicondutores é complexo e dificulta tanto o entendimento quanto o projeto dos circuitos integrados. Na segunda abordagem, efeitos de segunda ordem dos transistores normalmente são negligenciados no projeto dos circuitos integrados podendo acarretar o mau funcionamento do circuito projetado.

Atualmente na indústria eletrônica, o entendimento dos dispositivos semicondutores é essencial, não apenas no projeto de circuitos analógicos, mas, também, no projeto de circuitos digitais. Isso se deve ao fato de que os efeitos de segunda ordem dos transistores afetam diretamente seu desempenho, e esse impacto cresce conforme o nó de fabricação das tecnologias é reduzido (RAZAVI, 2001, p.9). Nó de fabricação é o comprimento mínimo da porta de um transistor na tecnologia e normalmente é utilizado como parâmetro de comparação entre tecnologias de fabricação CMOS.

Cabe tradicionalmente ao projetista decidir quais dos efeitos de segunda ordem serão negligenciados, mesmo não possuindo garantias de que estes efeitos não afetaram de fato o desempenho de seu circuito. Em busca de conhecer os efeitos de segunda ordem em um projeto tradicional de circuitos analógicos, após o equacionamento e dimensionamento dos dispositivos terem sido realizados, o circuito é testado em um simulador elétrico. É necessária, também, a análise do comportamento do circuito em face às variações dos parâmetros de fabricação.

Os transistores MOS são dispositivos que possuem quatro terminais: dreno (*drain*, em inglês), porta (*gate*, em inglês), fonte (*source*, em inglês) e o substrato ou corpo (*bulk*, em inglês). As regiões de dreno e fonte são fortemente dopadas em um substrato semiconductor dopado de forma oposta. A região intermediária entre o dreno e a fonte é chamada de canal. O tipo de portadores em maior concentração do canal determina o tipo de transistor MOS criado: canal p (PMOS), caso o canal possua maior concentração de portadores positivos (lacunas de elétrons); canal n (NMOS), caso o canal possua maior concentração de portadores negativos (elétrons).

A região do canal é isolada do terminal da porta por meio de uma camada de isolante, que foi durante muito tempo o óxido de silício, sobreposta por uma camada condutora, polisilício ou um metal. É importante notar que o dispositivo é simétrico em relação ao dreno e a fonte, ou seja, esses terminais são intercambiáveis. Na Figura 1, é exemplificado um transistor MOS de canal tipo n.

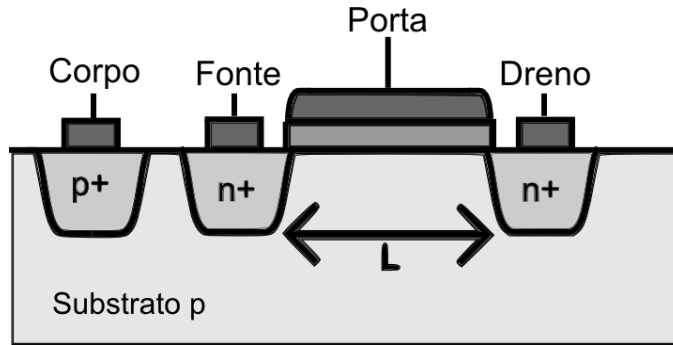


Figura 1: Corte horizontal de um transistor NMOS.

A dimensão da porta ao longo da direção fonte-dreno é normalmente chamada de comprimento, L (do inglês, *length*). A dimensão da porta perpendicular ao comprimento é chamada de largura, W (do inglês, *width*). Devido aos processos de fabricação, a distância efetiva entre o dreno e a fonte é ligeiramente menor que o valor de L . Neste documento, denotaremos como L o comprimento efetivo do canal.

O substrato possui grande influência no comportamento do transistor, sendo responsável por muitos dos efeitos de segunda ordem. Um transistor MOS deve possuir as junções p - n do dreno/fonte e substrato polarizadas reversamente em sua operação normal. Para que isso ocorra, é comum que o substrato de transistores NMOS esteja conectado a tensão elétrica mais negativa do circuito e o substrato dos transistores PMOS, a tensão mais positiva do circuito.

Na Figura 2, são mostrados os símbolos normalmente atribuídos aos transistores MOS. Em (a), observa-se a notação completa com indicação do corpo. Em (b), observa-se a notação com omissão do terminal de corpo, nesse caso o terminal de corpo está conectado ao terminal da fonte (S). Neste trabalho, são usadas as notações presentes nessa figura: o terminal da porta é denotado pela letra G; o terminal da fonte, por S; o terminal do dreno, por D; e o terminal do corpo é denotado por B.

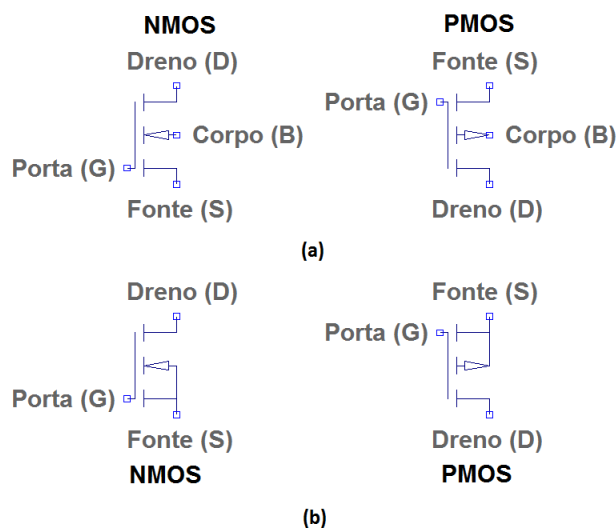


Figura 2: Símbolos e convenções para transistores MOS.

Na operação dos transistores MOS, é aplicada uma tensão elétrica entre os terminais da porta e da fonte (em que $V_{GS} > 0$ para transistores NMOS e $V_{GS} < 0$ para transistores PMOS). Essa tensão elétrica atrai os elétrons da região abaixo da porta no transistor NMOS – ou os repele, para transistores PMOS – podendo inverter o tipo do semiconductor presente nessa região e formando um canal de condução entre o dreno e a fonte. A magnitude da tensão $|V_{GS}|$ determina a região de operação do transistor MOS. As regiões de operação são três (SASSI, 2013, p.28):

- Fraca inversão: nessa região de operação, a tensão elétrica V_{GS} aplicada possui magnitude próxima à tensão elétrica de limiar de condução V_{TH} (*threshold voltage*, em inglês). É formado um canal de condução entre o dreno e a fonte que possui concentração de portadores menor que a concentração intrínseca de portadores do substrato, mas de tipo invertido. Essa região de operação é utilizada para circuitos com baixo consumo de potência como, por exemplo, em aplicações portáteis;
- Forte Inversão: nessa região de operação, a tensão elétrica V_{GS} aplicada é suficiente para formar um canal de condução com concentração de portadores superior ou igual à concentração de portadores intrínseca do substrato, mas de tipo invertido. A tensão $|V_{GS}|$ deve ser maior do que $|V_{TH}|$;
- Inversão moderada: essa é uma região de transição entre as regiões de fraca inversão e forte inversão, não possuindo limites claros e modelagem matemática imprecisa.

Nesse trabalho, não foi dada atenção a região de operação dos transistores MOS dos circuitos analisados. Nessa seção será apresentada apenas a região de operação de forte inversão para a descrição da operação dos transistores MOS em grandes sinais.

Uma característica importante do transistor MOS é que ele apenas permite a condução de corrente elétrica entre dreno e fonte caso a tensão elétrica entre a porta e a fonte tenha ultrapassado a tensão de limiar V_{TH} : $V_{GS} > V_{THN}$, para transistores NMOS; $V_{GS} < V_{THP}$, para transistores PMOS. Outra característica importante do transistor é que quando em condução sua corrente de dreno (I_D) depende da tensão V_{GS} aplicada.

A seguinte equação descreve a comportamento do transistor NMOS em relação aos seus parâmetros físicos e elétricos, em uma modelagem simplificada, considerando a fonte conectada ao corpo do transistor ($V_{SB} = 0$) e que o transistor opera na região triodo ($V_{DS} < V_{GS} - V_{THN}$):

$$I_{DN} = \mu_n \frac{W}{L} C_{OX} [(V_{GS} - V_{THN})V_{DS} - \frac{V_{DS}^2}{2}] \quad \text{onde } V_{DS} < V_{GS} - V_{THN}. \quad (1)$$

em que I_{DN} é a corrente de dreno, V_{THN} é a tensão de limiar, μ_n é a mobilidade de portadores no substrato e C_{OX} é capacitância do oxido da porta por unidade de área.

Nota-se que na região triodo, a corrente elétrica de dreno do transistor MOS depende também da tensão V_{DS} .

Quando a tensão $V_{DS} \geq (V_{GS} - V_{THN})$ o canal se estreita ao limite na proximidade do dreno, saturando o fluxo de corrente elétrica entre o dreno e a fonte. A corrente elétrica na região de saturação do transistor NMOS é descrita pela equação abaixo:

$$I_{DN} = \frac{1}{2} \mu_n C_{OX} \frac{W}{L} (V_{GS} - V_{THN})^2 (1 + \lambda \cdot V_{DS}) \quad \text{em que } V_{DS} \geq V_{GS} - V_{THN}. \quad (2)$$

O fator $(1 + \lambda \cdot V_{DS})$ presente na Equação 2 modela o efeito de modulação de comprimento de canal na região de saturação do transistor (SASSI, 2013, p. 30). A modulação de canal altera o comprimento efetivo do canal e é função da tensão aplicada entre o dreno e a fonte (V_{DS}). A equação que descreve o comportamento do transistor PMOS em relação aos seus parâmetros físicos e elétricos está a seguir e tem forma análoga à considerada para o transistor NMOS:

$$I_{DP} = -\mu_p \frac{W}{L} C_{OX} [(V_{GS} - V_{THP})V_{DS} - \frac{V_{DS}^2}{2}] \quad \text{em que } V_{SD} < V_{SG} - V_{THP}, \quad (3)$$

$$I_{DP} = -\frac{1}{2} \mu_p C_{OX} \frac{W}{L} (V_{GS} - V_{THP})^2 (1 + \lambda \cdot V_{DS}) \quad \text{em que } V_{SD} \geq V_{SG} - V_{THP}. \quad (4)$$

Nota-se pelas equações de ambos os transistores que, mesmo utilizando uma modelagem bastante simplificada, a complexidade do comportamento elétrico dos transistores MOS é alta. Essa complexidade justifica a utilização de alternativas ao dimensionamento manual dos transistores em projeto de circuitos analógicos. Além dos efeitos de primeira e segunda ordem modelados pelas equações acima, existem muitos outros fenômenos não modelados tais como: efeito da tensão de substrato, condução em tensão sublimiar, variação da mobilidade com as tensões, saturação da velocidade de portadores, capacitâncias do dispositivo, capacitâncias parasitas, correntes de fuga, etc.

Como já foi dito, além do equacionamento do circuito também simulações elétricas são aplicadas para analisar o comportamento primário e dos efeitos de segunda ordem dos transistores.

2.2 Circuitos osciladores

Circuitos osciladores são circuitos intencionalmente instáveis, que produzem sinais elétricos periódicos indefinidamente quando alimentados por tensões elétricas do

tipo contínua. Em diversos sistemas eletrônicos, é necessária a utilização de sinais periódicos para as mais diversas aplicações: temporização de circuitos lógicos, referência para recepção e transmissão de sinais, referências de tempo, geração de sons, etc. Para satisfazer tamanha gama de aplicações existe uma grande quantidade de topologias de circuitos osciladores com desempenhos variados.

A função de um circuito oscilador é gerar uma forma de onda com características, tais como frequência, amplitude, forma e ruído, bem definidas (S. Franco (2001)). Em alguns casos, essas características são projetadas para serem controladas externamente. Como exemplo de circuitos osciladores com controle externo de parâmetros, tem-se o circuito oscilador controlado por tensão (VCO), extensamente utilizado em circuitos Phase-Locked Loops (PLL).

Conforme Carter e Mancini (2009), existem duas classes de circuitos osciladores:

- Osciladores harmônicos: consistem de amplificadores com circuitos RC e/ou LC ressonantes ou cristais ressonantes que ajustam a frequência de oscilação do circuito. Circuitos osciladores harmônicos produzem oscilações quase senoidais devido ao fato de que seus componentes amplificadores operam na zona de linearidade;
- Osciladores de relaxação: são circuitos não lineares que produzem oscilações não senoidais, com formas que podem ser triangular, dente de serra, quadrada, pulso ou exponencial. Esses circuitos utilizam a carga e descarga de capacitores e indutores para produzir tais sinais. Assim, o período de oscilação é fator da constante de tempo de um bloco com capacitor ou indutor. As ondas possuem caráter não senoidal pelo fato de que um dos dispositivos eletrônicos do circuito opera na zona de não linearidade na maior parte do ciclo.

Esse trabalho irá se basear em osciladores harmônicos. Esses osciladores não utilizam sinais de entrada externos e possuem característica oscilatória obtida através da realimentação positiva, negativa ou combinações das mesmas.

As principais topologias de circuitos osciladores harmônicos baseiam-se em módulos passivos – que determinam a frequência de oscilação e constituem a realimentação necessária – e módulos amplificadores – responsáveis pela amplificação do sinal realimentado. Outras topologias, como os osciladores em anel, utilizam o atraso de propagação do sinal para gerar a oscilação desejada.

Os requisitos mínimos para a oscilação de um circuito realimentado são derivados com a estrutura canônica de realimentação utilizada em estudos de controle.

Na Figura 3, é mostrada a representação em diagrama de blocos da estrutura canônica de realimentação positiva ($B(s) < 0$) ou negativa ($B(s) > 0$).

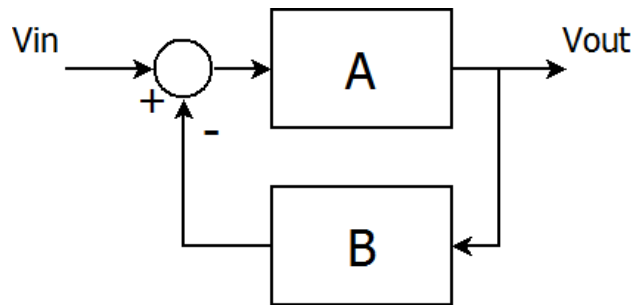


Figura 3: Diagrama de um sistema realimentado

A equação abaixo mostra a relação entre a entrada e a saída do sistema, considerando ambas como tensões elétricas.

$$\frac{V_{out}}{V_{in}}(s) = \frac{A(s)}{1+A(s) \cdot B(s)} \quad (5)$$

em que $A(s)$ é a função de transferência do bloco A.

Apesar de os circuitos osciladores harmônicos convencionais não terem sinal de entrada externo, eles utilizam frações da tensão de saída como entrada aplicada via uma rede de realimentação. Considerando que ganho da realimentação seja unitário, ou seja, $B(s) = 1$ na Equação 5, têm-se:

$$\frac{V_{out}}{V_{in}}(s) = \frac{A(s)}{1+A(s)} \quad (6)$$

A oscilação ocorre quando o sistema deixa de ser estável, o que acontece quando os polos do sistema estão localizados no semi-plano direito. Essa condição é alcançada quando o denominador da equação do sistema realimentado se torna igual à zero, ou seja, $1 + A(s) = 0$. Para tal condição ocorrer, $A(s)$ deve ser igual a -1 na frequência de oscilação desejada. Em outras palavras, se $A(j\omega_0) = -1$, o sistema terá ganho de malha fechada igual infinito na frequência ω_0 . Em consequência, o sistema irá amplificar os ruídos de seus próprios componentes e irá oscilar indefinidamente na frequência ω_0 . Pode-se ver que quando $A(j\omega_0) = -1$ o sinal na saída tem fase oposta a entrada. Esse sinal é então subtraído na entrada, causando a regeneração do sinal inicial e permitindo que a componente na frequência desejada cresça.

Assim, um circuito com realimentação negativa para oscilar na frequência ω_0 deve satisfazer as duas condições a seguir:

$$|A(j\omega_0)| = 1 \quad (7)$$

$$\angle A(j\omega_0) = 180^\circ. \quad (8)$$

Essas duas condições para que a oscilação ocorra são conhecidas como Critério de Barkhausen (RAZAVI, 2001, p.483; FRANCO, 2009). Para garantir a oscilação em presença de variações de temperatura e dos parâmetros de fabricação, tipicamente se escolhe um ganho de amplificação de ao menos duas ou três vezes o valor unitário (RAZAVI, 2001, p.483).

O Critério de Barkhausen determina que o sinal de saída deve ser defasado de 180° e ter amplitude igual ou maior que o sinal de entrada em A(s). Há duas maneiras de se obter essa condição: utilizando realimentação positiva ou utilizando realimentação negativa e defasagem de 180° do sinal de saída. Em condições reais e obedecendo ao critério de Barkhausen, a oscilação do sistema se origina pela amplificação de ruídos internos e/ou pelas tensões elétricas de inicialização. A estabilidade da oscilação provém da limitação da tensão de alimentação e da excursão máxima permitida pelos elementos amplificadores.

Há diversos parâmetros que devem ser considerados no projeto de um circuito oscilador entre eles se destacam: frequência de oscilação; potência consumida; ruído de fase, amplitude do sinal de saída, grau de estabilidade da frequência e tempo de início da oscilação.

Em um projeto, deve-se escolher a topologia adequada a aplicação desejada. Os diferentes tipos de topologias variam conforme: a configuração dos estágios de amplificação; o método de estabilização da frequência de oscilação; e o método de limitação do sinal de saída.

Um parâmetro importante para os osciladores harmônicos é o fator de qualidade Q da rede ressonante que forma o filtro seletor da malha de realimentação. Q é um parâmetro adimensional e, por definição, dado por (BOSTIAN, KRAUSS, RAAB, 1980; LEE, 1998):

$$Q \equiv 2\pi \times \frac{\text{máxima energia instantânea armazenada no circuito}}{\text{energia dissipada em um ciclo}}. \quad (9)$$

Em geral quanto maior o valor do Q, mais bem definida é a frequência de oscilação do circuito.

Os circuitos osciladores harmônicos podem ser classificados de acordo com o tipo de filtro seletor de frequência utilizado na malha de realimentação ou da utilização

de outro artifício para a seleção de frequência, como é o caso do circuito oscilador em anel. Os tipos possíveis são:

Osciladores RC

Em um circuito oscilador RC, o filtro seletivo é uma rede de resistores e capacitores. Esse tipo de circuito é comumente utilizado para sintetizar baixas frequências, por exemplo, na faixa de áudio. Circuitos osciladores RC comuns são o oscilador por defasagem de fase e o oscilador ponte de Wien.

Osciladores LC

Em um circuito oscilador LC, o filtro é um circuito tanque composto de um indutor e um capacitor conectados. Cargas fluem entre o capacitor e o indutor permitindo que o circuito tanque armazene energia enquanto oscila na frequência de ressonância. Há pequenas perdas no circuito tanque, mas o amplificador compensa essas perdas e fornece energia para que a oscilação se mantenha constante.

Circuitos osciladores LC são frequentemente utilizados para rádio frequências quando uma fonte de frequência sintonizável é necessária, tais como em geradores de sinais, sintonizadores de rádio transmissores e osciladores locais em rádio receptores. Circuitos osciladores LC típicos são os circuitos de Harley, Colpitts e Clapp.

Em circuitos osciladores LC, o ruído de fase é inversamente proporcional ao fator de qualidade Q de seu ressonador tanque (RAZAVI, 2012, p.501).

Osciladores a cristal

Circuitos osciladores a cristal utilizam um cristal piezoelétrico (comumente um cristal de quartzo) como elemento seletor de frequência. O cristal vibra mecanicamente como um ressonador e sua frequência de oscilação determina a frequência de oscilação do circuito. Cristais têm um alto fator de qualidade e uma melhor estabilidade com a temperatura que circuitos tanque. Assim, osciladores a cristal é o tipo de oscilador que possui a maior estabilidade de frequência. Eles são utilizados para estabilizarem a frequência de rádio transmissores e para gerar sinais de relógio em sistemas no geral.

Os circuitos osciladores a cristal geralmente utilizam as mesmas topologias que circuitos osciladores LC com o cristal substituindo o circuito tanque.

Dispositivos SAW (*surface acoustic wave*, em inglês) são outro tipo de ressonadores piezoelétricos usados em osciladores a cristal, podendo atingir frequências mais elevadas. Esses dispositivos são utilizados em aplicações que requerem uma alta frequência de referência, baixo consumo de potência e um sistema totalmente integrado (KARIM, NORDIN, ALAM, 2012, p.490).

Osciladores em anel

Um circuito oscilador em anel consiste de um determinado número de estágios de ganho em malha fechada (RAZAVI, 2001, p.484). Assim, esse tipo de circuito utiliza a defasagem de sinal provocada pelos estágios de ganho para satisfazer o Critério de Barkhausen (RAZAVI, 2012, p.507). No caso em que o circuito é composto por um anel de inversores, o Critério de Barkhausen será satisfeito quando um número ímpar de inversores for aplicado e a soma dos atrasos de propagação, atraso de subida mais atraso de descida, for igual ao período de oscilação (RAZAVI, 2001, p.490-491).

2.2.1 Circuitos osciladores controlados por tensão

Diversas aplicações exigem osciladores que sejam sintonizáveis, ou seja, que a frequência do sinal de saída seja uma função de um sinal de controle tomado como entrada do sistema. Normalmente, tanto o sinal de saída quanto o sinal de entrada são tensões elétricas (RAZAVI, 2001, p.510). Neste trabalho, os circuitos VCO's considerados tem tensão tanto no sinal de saída como no sinal de controle. Um oscilador controlado por tensão ideal é um circuito cuja frequência de saída é uma função linear do sinal de controle, descrito pela equação abaixo:

$$F_{out} = F_0 + K_{VCO}V_c \quad (10)$$

em que V_c é a tensão de controle, ω_0 é a frequência de oscilação quando $V_c = 0$ e K_{VCO} uma constante de sensibilidade do circuito.

Na Figura 4, o comportamento de um circuito VCO e sua representação em diagrama de blocos são mostrados. A faixa de frequência atingida pelo circuito ($F_2 - F_1$) é chamada de faixa de sintonização (*tuning range*, em inglês).

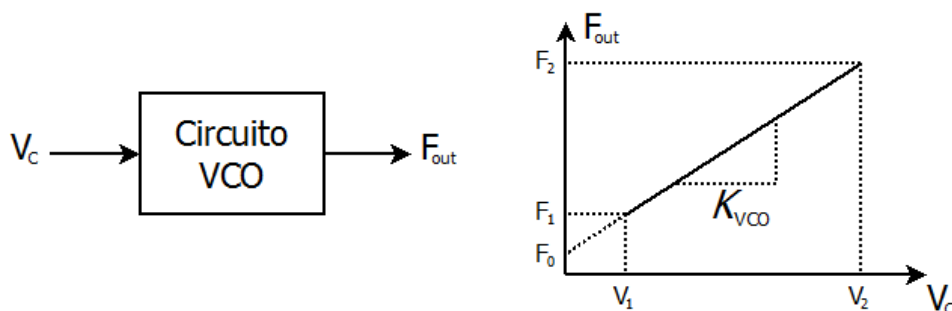


Figura 4: Diagrama de blocos e função de frequência de um circuito VCO.

As topologias utilizadas em circuitos VCO podem ser geradas a partir das topologias utilizadas em circuitos osciladores com frequência fixa, apenas tornando um dos elementos da rede ressonante variável com a tensão de controle. Comumente,

varactors de junção ou *varactors* MOS, são utilizados como elemento variável, como será explicado a seguir.

2.2.2 Capacitâncias variáveis

Usualmente a capacitância em circuitos VCO é a combinação de capacitâncias fixas e capacitâncias ajustáveis com tensão, os chamados *varactors*. Há um grande número de opções disponíveis de elementos com capacitância variável para os projetistas nas tecnologias CMOS (BUNCH, RAMAN; 2003, p.1325). Dentre elas destacam-se as duas opções a seguir:

- *Varactor* de junção *pn*. Utiliza a junção *pn*, formada por uma região *p+* e uma região *n* do substrato, quando a tecnologia de fabricação se basear um substrato *n*, ou o contrário quando for substrato *p*. Essa junção forma um diodo que é colocado reversamente polarizado. A capacitância da região de depleção é dependente da tensão reversa aplicada. Uma desvantagem dessa topologia é que com alta tensão de polarização, a junção *pn* pode entrar em condução devido a ruptura.
- Capacitores MOS. Esse tipo de capacitor opera de maneira similar a um capacitor de placas paralelas; no capacitor MOS, as placas paralelas são formadas pela polissilício de sua porta e seu canal (BUNCH, 2001, p.55). A capacitância entre a porta e os terminais da fonte e do dreno dessa estrutura varia não linearmente com a tensão elétrica aplicada na porta.

Neste trabalho, será utilizado como capacitância variável capacitores MOS. Um capacitor MOS possui três regiões distintas de operação que dependem das cargas elétricas presentes no dispositivo: acumulação, depleção e inversão.

A acumulação no capacitor MOS ocorre quando há o aumento de concentração dos portadores majoritários abaixo do terminal de porta. Isso acontece, nos transistores NMOS, quando $V_{GS} < 0$ (para PMOS, quando $V_{GS} > 0$). A depleção ocorre quando tanto elétrons como lacunas são afastadas da região abaixo da porta. A inversão, por sua vez, ocorre quando a quantidade de portadores minoritários supera a quantidade de portadores majoritários abaixo da porta. Isso acontece, nos transistores NMOS, quando $V_{GS} > V_{THN}$ (para PMOS, $V_{GS} < V_{THP}$). Portadores majoritários são os portadores com maior concentração intrínseca no semicondutor (lacunas para material tipo *p*, e elétrons para material tipo *n*). Em oposição, portadores minoritários são os portadores com menor concentração.

Realizando-se conexões adequadas nos terminais de um transistor MOS, as características de sua capacitância de porta podem ser alteradas e dessa maneira o capacitor MOS pode ser utilizado como dispositivo de sintonização em circuitos VCO. Diferentes variações na estrutura básica de um transistor MOS foram exploradas para implementar *varactors* com alto fator de qualidade (BUNCH, RAMAN; 2003, p.1326).

Normalmente, o substrato de um *wafer* de silício é do tipo p, produzindo transistores NMOS diretamente no substrato e transistores PMOS em poços do tipo n (*n-well*, no termo em inglês). Assim, é permitido ao projetista isolar cada transistor PMOS em seu próprio poço, o que pode ser utilizado de forma vantajosa em projetos de circuitos integrados analógicos. Também, a utilização de capacitores PMOS fabricados em poços do tipo n como *varactors* é vantajosa, pois permite que os dois terminais do capacitor, a porta e o corpo do transistor, possam ser ligados a quaisquer tensões. São capacitores que apresentam os terminais livres.

Considerando o capacitor PMOS com capacitância variável há três topologias possíveis para aplicação: topologia livre, topologia para trabalhar na região de inversão e topologia para trabalhar na região de acumulação. Em todas elas, um dos terminais do *varactor* é a porta do transistor; a diferença entre as topologias está em como os outros terminais do transistor PMOS são utilizados.

Na topologia livre, o dreno, a fonte e o corpo do transistor são interconectados formando um só terminal. Essa topologia possui a vantagem de permitir a variação da tensão de porta em uma gama maior de valores já que o *varactor* opera nas três regiões de operação do capacitor MOS. Na Figura 5, as conexões do *varactor* nessa topologia e o comportamento da capacitância normalizada em relação a V_{SG} são mostrados. Essa topologia se torna interessante no projeto de circuitos analógicos CMOS por algoritmos meta-heurísticos por apresentar uma maior flexibilidade nas tensões de controle do *varactor*. Por outro lado, o fato de que nessa topologia a curva valor da capacitância versus V_{SG} não ter comportamento monotonico é um empecilho no projeto automático do circuito. Adicionalmente, esse comportamento também tem reflexo na curva frequência do circuito VCO versus tensão de controle, tornando-a não linear e não monotônica.

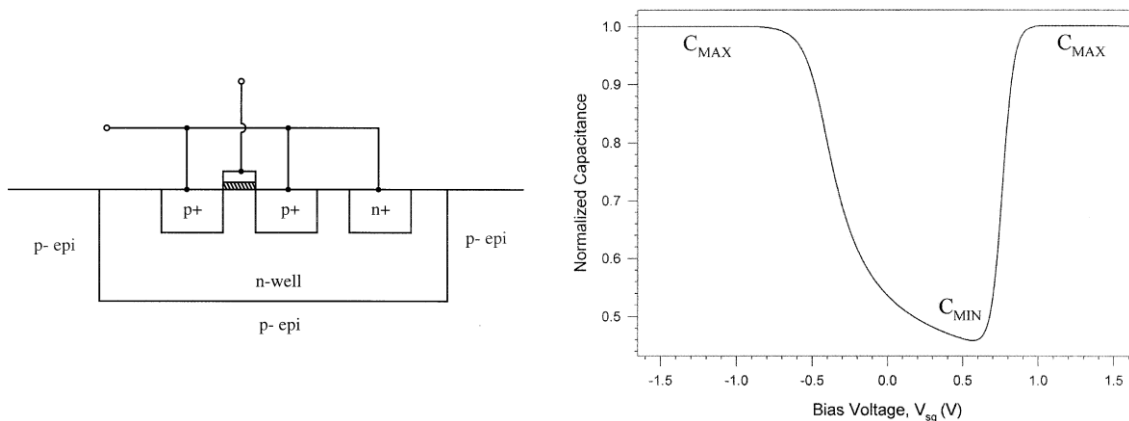


Figura 5: Capacitor MOS em topologia livre (BUNCH, RAMAN; 2003, p.1327).

Na topologia para operação na região de inversão, o dreno e a fonte são interconectados formando o outro terminal do *varactor* e o corpo é conectado a mais alta tensão do circuito. Nessa topologia o *varactor* opera na região de inversão, por causa da conexão do poço em potencial alto (BUNCH, RAMAN, 2003, p.1327). Na Figura 6, a conexão do *varactor* nessa topologia e o comportamento da capacitância normalizada em relação à V_{SG} são mostrados. Como observado no gráfico da capacitância, seu valor possui uma alta variação com V_{SG} , o que torna esta topologia inapropriada para o projeto de circuitos VCO, dado que sua utilização faz com que sensibilidade do circuito seja alta.

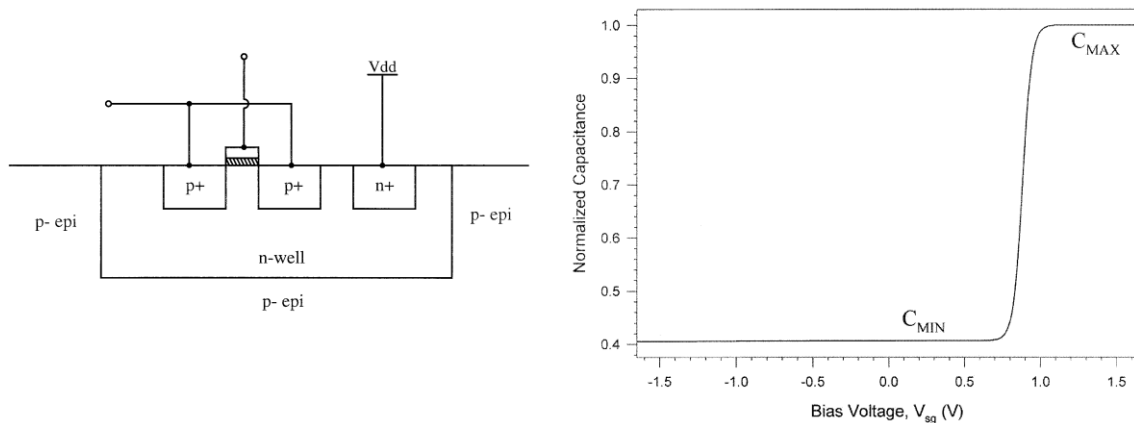


Figura 6: Capacitor MOS na topologia com operação na região de inversão (BUNCH, RAMAN, 2003, p.1327).

Na topologia para operação na região de acumulação, as regiões de fonte e dreno do transistor PMOS são substituídas por regiões dopadas com n+. Essas alterações garantem um bom contato no poço, impedem a injeção de portadores minoritários no canal e impedem, também, a inversão do canal. Na Figura 7, a conexão do *varactor* nessa topologia e o comportamento da capacitância normalizada em relação à V_{SG} são mostrados. Como visto pelo comportamento da capacitância, esse *varactor* possui uma menor sensibilidade da capacitância com relação à V_{SG} , o que é bom para

circuitos VCO. Por outro lado, a indisponibilidade de modelos de simulação impossibilita sua utilização neste trabalho.

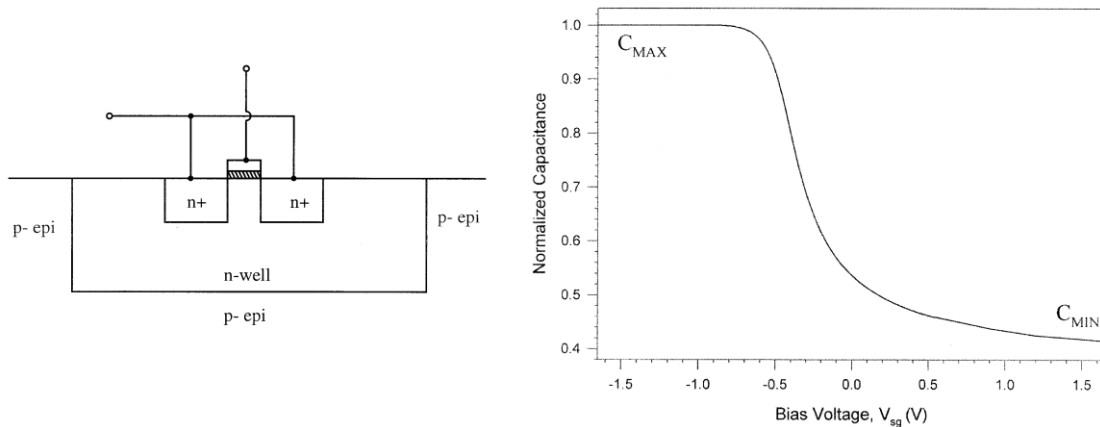


Figura 7: Capacitor MOS na topologia com operação na região de acumulação (BUNCH, RAMAN; 2003, p.1327).

2.2.3 Parâmetros de um oscilador

Um oscilador deve possuir parâmetros adequados para a aplicação em que é utilizado. A seguir são descritos alguns parâmetros importantes de circuitos osciladores e VCOs:

Frequência central

A frequência central é o valor de frequência do sinal de saída do circuito ou, para VCOs, a média da faixa de sintonização do circuito. Ela deve ser determinada visando à aplicação em que o circuito será utilizado. Segundo Razavi (2001, p.511), os circuitos osciladores nas tecnologias CMOS atuais alcançam frequência central de até 10 GHz.

Faixa de sintonização (VCOs)

É a faixa de frequências que um VCO pode atingir. Dois fatores influenciam na escolha da faixa de sintonização. O primeiro deles é a variação da frequência central com as variações de temperatura e dos parâmetros de fabricação. A frequência central de um circuito VCO pode variar até por um fator de dois nos extremos do processo e da temperatura, Razavi (2001, p.511). O segundo fator é o tamanho de faixa de sintonização requerida propriamente pela aplicação.

Em um circuito VCO é desejável ter uma faixa de sintonização com a maior largura possível, para garantir que a frequência central desejada sempre esteja na faixa sintonizável do circuito fabricado, não importando as variações de temperatura e dos parâmetros de fabricação.

Linearidade de sintonização (VCOs)

A linearidade da sintonização reflete a constância da sensibilidade do circuito VCO dentro de sua faixa de sintonização. Assim quanto mais constante for K_{VCO} , maior é a linearidade de sintonização do circuito VCO. Se K_{VCO} não for constante, a não linearidade de sintonização irá degradar o desempenho dos sistemas que fazem uso da frequência de saída do circuito VCO como sinal de controle.

Na maior parte das aplicações é desejável a minimização da variação de K_{VCO} ao longo da faixa de sintonização. Segundo (RAZAVI, 2001, p.511), os circuitos VCO tipicamente possuem uma maior sensibilidade ao redor da frequência central e uma menor sensibilidade nos extremos da faixa de sintonização.

Amplitude do sinal de saída

Indica a excursão do sinal de saída. É desejável atingir uma alta amplitude de oscilação do sinal de saída, fazendo com que a forma de onda seja menos sensível a ruídos. Esse aumento da amplitude, porém conflita com a dissipação de potência, com distorções do sinal e, em VCOs, com a faixa de sintonização.

Dissipação de potência

É o consumo do oscilador. Deseja-se uma menor quantidade de potência para a operação do circuito sem que prejudique o seu desempenho.

Rejeição a modo comum e a ruído na alimentação

Osciladores são sensíveis a ruídos que podem ser injetados pela linha de alimentação. Assim é desejável o projeto de osciladores com alta rejeição a modo comum, utilizando caminhos diferenciáveis, e com alta rejeição a ruídos presentes na tensão de alimentação.

Pureza do sinal de saída

Os circuitos osciladores, mesmo com tensão de controle constante em VCOs, apresentam forma de onda de saída não perfeitamente periódica. Os ruídos inerentes aos dispositivos do circuito, além do presente na fonte de alimentação, causam ruído de fase e *jitter* que devem ser limitados de acordo com requisitos determinados pela aplicação. A pureza do sinal de saída está relacionada também a distorção harmônica total. Essa medida considera o quanto de energia está disperso nas frequências harmônicas ao invés de ser empregado na frequência fundamental do sinal de saída. Esse parâmetro é definido como a relação da potência presente nas frequências harmônicas sobre a potência do sinal senoidal.

Outros fatores

Uma importante preocupação no projeto de circuitos VCO é a variação da frequência e da fase do sinal de saída como resultado de ruídos presentes na linha de controle. Para dada amplitude de ruído presente na linha de controle, o ruído que será apresentado na saída será proporcional por K_{VCO} , devido à Equação 10. Assim, quanto maior a faixa de sintonia do circuito, mais sensível é o VCO a ruídos no sinal de controle.

Como forma de minimizar o efeito do ruído na tensão de controle, a sensibilidade do circuito VCO deve ser minimizada. Porém, uma menor sensibilidade causa uma menor faixa de sintonização e corre-se o risco de, devido a variações de temperatura e dos parâmetros de fabricação, a faixa de sintonia não cobrir as frequências desejadas.

2.3 Algoritmos meta-heurísticos

Problemas de otimização são problemas de busca pela solução que minimiza (ou maximiza) o desempenho de um sistema em determinados aspectos. Na otimização de sistemas, cada possível solução é representada por um conjunto de valores de parâmetros do sistema dentro do espaço de busca do problema. Esse espaço de busca possui n dimensões, em que n é o número de parâmetros do sistema a ser otimizado.

Algoritmos meta-heurísticos são métodos que utilizam determinado grau de aleatoriedade para encontrar a solução ótima para um problema de otimização, ou uma tão boa quanto possível, segundo Sean Luke (2013, p.9). O mesmo autor descreve que a meta-heurística é aplicada em problemas descritos como: “você conhece um bom quando o vê” – “you know a good one when you see it”, no original. Ou seja, a meta-heurística é utilizada quando, previamente, se conhece pouco acerca do problema e de como resolvê-lo, porém caso se possa avaliar uma possível solução, pode-se determinar quão boa ela é.

O primeiro passo nesse tipo de algoritmo é a geração inicial de um conjunto de indivíduos, possíveis soluções, conjunto que pode ser obtido de forma randômica ou fornecido pelo usuário. Esses indivíduos são avaliados para alguns critérios de desempenho fornecidos pelo usuário. Após isso, os valores dos critérios de desempenho obtidos na avaliação são utilizados para o cálculo do desempenho geral do indivíduo, seu *score*. A determinação dos valores dos critérios de desempenho e o cálculo do desempenho geral é realizada por uma função objetivo, que será detalhada a frente.

Os valores de *score* de cada indivíduo são utilizados pelo algoritmo para a geração de novos indivíduos, feito por um método de busca. O processo gera

indivíduo(s) e avaliação se repete, permitindo o algoritmo testar várias possíveis soluções dentro do espaço de busca. Quando um critério de parada é atingido, a solução obtida é o melhor indivíduo analisado. Em alguns métodos, o último indivíduo analisado é considerado a solução mesmo que não seja o melhor indivíduo analisado. Na Figura 8, os passos principais de um algoritmo meta-heurístico são exemplificados.

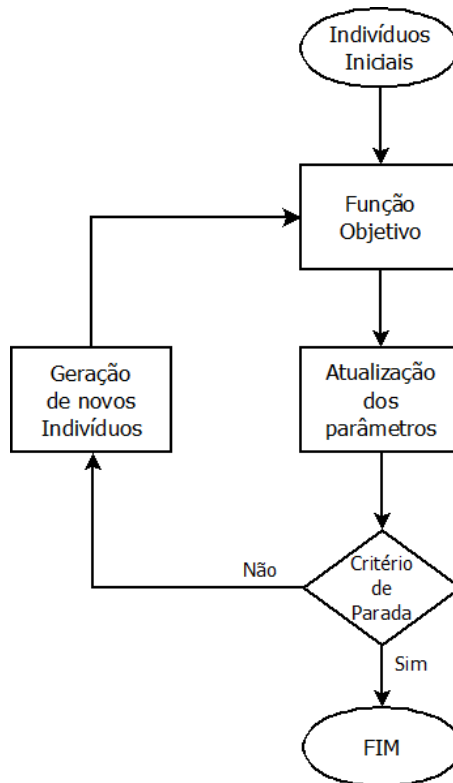


Figura 8: Fluxograma de um algoritmo meta-heurístico genérico.

Problemas de otimização, em que métodos tradicionais ou determinísticos de otimização são inviáveis por questões de complexidade e/ou de tempo, podem ser otimizados através da utilização de algoritmos meta-heurísticos.

O espaço de buscas deve ser limitado por restrições impostas pelo domínio do problema real, e a determinação dos limites deve ser realizado pelo usuário após a análise cuidadosa do problema. Dessa forma o método meta-heurístico é direcionado para um subespaço de busca em que as soluções são viáveis. Em nosso caso de projeto de circuitos CMOS, os parâmetros serão as características físicas dos transistores MOS e as características elétricas dos dispositivos elétricos passivos (resistores, capacitores e indutores) e ativos (fontes de tensão e correntes).

A função objetivo é um dos elementos mais importantes utilizadas nos algoritmos meta-heurísticos de otimização. Sua função é atribuir a cada solução testada um número real, o *score*, que indica sua qualidade. Para tal objetivo

extrai-se de cada indivíduo os valores dos critérios de desempenho que são utilizados no cálculo do *score*. No caso de nosso projeto de circuitos, tais valores são obtidos de simulações elétricas realizadas na plataforma HSPICE e que buscam refletir o desempenho real do circuito quando fabricado.

Neste trabalho, foram utilizados três algoritmos meta-heurísticos: Otimização por Enxame de Partículas, Arrefecimento Simulado e Busca Padrão.

2.3.1 Otimização por Enxame de Partículas

O método de otimização por Enxame de Partículas foi proposto por James Kennedy e Russel Eberhart (EBERHART, KENNEDY; 1995, p.1942-1948) nos anos 1990. Segundo Sean Luke (2013, p.55), esse método é uma técnica de otimização estocástica que é similar a algoritmos evolucionários em alguns pontos. Porém, difere desses em um importante quesito: não há seleção de indivíduos de nenhum tipo. A motivação para o algoritmo é o comportamento social de interação entre indivíduos (partículas) de um grupo (enxame).

A inspiração do método deriva do comportamento de bandos (enxame) de pássaros e cardumes de peixes (partículas) em busca de alimento em uma determinada região. Verifica-se que o comportamento do grupo é influenciado tanto pela experiência individual acumulada por cada indivíduo bem como pelo resultado da experiência acumulada pelo grupo (LUKE, 2013, p.55).

No algoritmo de otimização por Enxame de Partículas, cada solução, um ponto no espaço n-dimensional de busca, é uma partícula. A cada partícula está associada, além de sua coordenada (parâmetros) e o seu valor de *score*, uma velocidade que define a direção, sentido e intensidade do movimento da partícula.

A inicialização do enxame é randômica, ou seja, as partículas são geradas com posição e velocidade aleatórias. Pode ocorrer também do usuário fornecer, em algumas implementações, a posição inicial de uma das partículas. A cada iteração ocorre: o cálculo dos *scores* das partículas; a alteração da velocidade das partículas; e a determinação da nova posição das partículas devido a sua velocidade. A alteração das velocidades leva em conta os seguintes fatores: a posição com melhor *score* encontrado pela partícula (melhor local), a posição com melhor *score* encontrado pelo enxame (melhor global) e velocidade anterior da partícula. Na Figura 9, a posição da partícula e a obtenção da sua nova velocidade são exemplificadas.

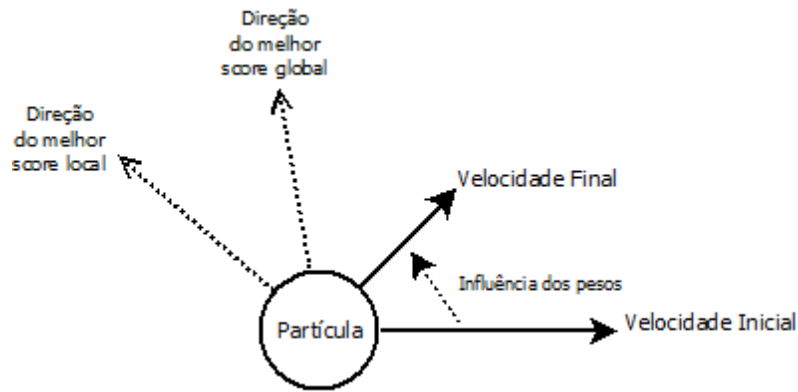


Figura 9: Velocidade e posição de uma partícula.

As sucessivas movimentações das partículas, com as velocidades das partículas apontando para o melhor posicionamento de cada partícula e do grupo, permitem que o enxame convirja para a melhor solução do sistema, ou ao menos para um mínimo local.

2.3.2 Arrefecimento Simulado

O método meta-heurístico denominado de Arrefecimento Simulado (*simulated annealing*, em inglês) é baseado no processo de aquecimento e resfriamento de metais (KIRKPATRICK, GELLAT, VECCHI, 1983, p.671-680). Nesse sistema, através do resfriamento controlado, as estruturas formadas por átomos de metais se estabilizam com poucas irregularidades e em um nível de energia mínimo. Segundo Luke (2013, p.25), esse método foi desenvolvido por vários pesquisadores na década de 80 e foi derivado do Algoritmo *Metropolis*.

O Arrefecimento Simulado é uma técnica que inicia a busca a partir de uma solução inicial, randômica ou fornecida previamente. Essa solução é chamada de solução corrente e seu *score* f_o é avaliado. Uma nova solução é gerada randomicamente na proximidade da solução corrente. A nova solução gerada é avaliada e seu *score* f_n é obtido. A função objetivo nesse método possui uma alusão à energia térmica do sistema. A nova solução é aceita como solução corrente ou não dependendo da diferença entre os *scores*, $\Delta = f_o - f_n$. Para um problema de minimização, tem-se:

- Caso $\Delta < 0$, a solução gerada é melhor que a solução corrente, ou seja há uma redução da energia do sistema. O método aceita a nova solução;
- E, se $\Delta \geq 0$, a solução gerada é pior ou igual a solução corrente, ou seja, não mudou a energia do sistema ou ocorreu um aumento. Neste caso, a nova solução será aceita com uma probabilidade que depende da temperatura do sistema, que é um parâmetro do método. Em temperaturas altas, a aceitação de

soluções piores é maior; em temperaturas menores a aceitação de soluções piores é menor.

A probabilidade de que a nova solução seja aceita quando houver aumento de energia do sistema, normalmente e neste trabalho, possui a seguinte distribuição (SASSI, 2013, p.61):

$$P(\Delta, T) = e^{\frac{-\Delta}{T}}, \quad (11)$$

em que, T é a temperatura e Δ é a diferença entre os scores.

A temperatura do sistema sofre uma redução continua seguindo uma função de atualização. Essa função pode ser linear, geométrica, logarítmica ou ter qualquer outro comportamento. Além disso, ela pode ser não-monotônica, permitindo o reaquecimento do sistema periodicamente. A cada etapa, uma nova solução é gerada nas proximidades da solução corrente e os processos de avaliação e de aceitação/descarte da solução são repetidos. O algoritmo pode ser encerrado quando a temperatura do sistema atingir um patamar previamente selecionado, quando o número máximo de iterações for atingido ou quando o tempo máximo de execução for atingido.

A aceitação de soluções piores que a solução corrente e a possibilidade de reaquecimento do sistema permitem que o algoritmo fuja de mínimos locais. A capacidade de fugir de mínimos locais é maior nas primeiras iterações; com o passar das iterações a temperatura do sistema diminui e o método passa a refinar a solução corrente. Havendo o reaquecimento, novamente ele pode fugir de mínimos locais.

A implementação do Arrefecimento Simulado utilizada nesse trabalho possui como função de atualização de temperatura uma função não monótona, permitindo o reaquecimento.

2.3.3 Busca Padrão

O método de Busca Padrão foi proposto por Hooke e Jeeves (1961), chamado inicialmente de busca direta (*direct search*, no original em inglês), Trata-se de um método de busca da solução ótima a partir de uma solução corrente. Segundo Hooke e Jeeves (1980), “busca direta é um método de força bruta não tendo nenhuma elegância matemática”.

Um indivíduo inicial é fornecido previamente pelo usuário ou gerado randomicamente pelo método e este será a solução corrente. Novos indivíduos são gerados simultaneamente em uma vizinhança do indivíduo inicial seguindo

um padrão de busca especificado e caracterizada por uma distância de busca. Todos os indivíduos então gerados são avaliados e os *scores* deles são calculados.

Os *scores* dos indivíduos são comparados com o *score* da solução corrente. Caso uma solução melhor seja encontrada, esta substitui a solução corrente e a distância de busca para a próxima iteração é aumentada por um fator; senão a solução corrente é mantida e a distância de busca é reduzida por um fator (SASSI, 2013, p. 62).

Ocorre, então, a repetição do algoritmo até que algum critério de parada seja alcançado. Os critérios de paradas utilizados no algoritmo utilizado nesse trabalho são:

- Distância de busca mínima para a geração de indivíduos vizinhos;
- Número máximo de iterações sem que haja melhoria do *score* da solução corrente;
- E, número máximo de iterações do algoritmo.

Nó método utilizado nesse trabalho, o padrão de busca e os fatores de expansão e de contração da distância de busca podem ser alterados.

3 Metodologia

3.1 Plataforma de projeto

Durante o projeto de um circuito oscilador o dimensionamento dos circuitos é um problema complexo, pois, existem diversos critérios a serem otimizados e que geralmente geram conflitos na escolha dos parâmetros do circuito. Trata-se, portanto, de uma área propícia à aplicação de algoritmos meta-heurísticos. Há duas maneiras de se aplicar algoritmos meta-heurísticos para o projeto de circuitos: utilizando equações que descrevem o circuito e seu desempenho (PEREIRA et al., 2013); ou, utilizando simulações para a extração dos valores dos critérios de desempenho (EBRAHIMI, NASEH, 2011; FILGUEIRAS, 2010; SASSI, 2013).

O trabalho de Pereira (2013), desenvolvido recentemente, apresentou uma comparação entre três algoritmos meta-heurísticos – Algoritmo Genético, Enxame de Partículas e Arrefecimento Simulado – para encontrar o dimensionamento de um circuito VCO com ressonador LC. A avaliação do circuito foi realizada exclusivamente por equacionamento e a solução obtida foi validada através de resultados de simulação no HSpice RF. O trabalho obteve êxito no projeto do circuito VCO.

Um trabalho semelhante foi realizado por Ebrahimi e Naseh (2011), que utilizou uma modificação do algoritmo genético como meta-heurística. A avaliação do circuito foi realizada por simulação de balanço harmônico do simulador HSpice RF. Nesse trabalho apenas a frequência de oscilação, a amplitude do sinal, a potência consumida e o ruído de fase do circuito foram analisados.

Como já foi mencionado, nosso trabalho buscou o desenvolvimento de uma plataforma para projeto de circuitos osciladores que pudesse ser utilizada nas mais diversas topologias e aplicações, ao contrário dos trabalhos citados que foram direcionados para alguns circuitos VCO's e que o número restrito de critérios analisados restringi sua utilização a aplicações específicas. Para alcançar nossos objetivos, nossa plataforma é baseada em simulações.

Na plataforma de projeto desenvolvida nesse trabalho, o usuário deve descrever o circuito que deseja otimizar em linguagem SPICE e tomar os seguintes cuidados:

- Todas as variáveis que o usuário deseja otimizar devem ser nomeadas com X seguido do número da variável, ou seja, X1, X2, X3, ..., XN;
- Incluir os nós *o* e *d* que devem ser nós presentes na rede ressonante;
- Nomear o nó de saída de *out* para circuitos com frequência de oscilação fixa, para circuitos VCO a tensão entre os nós *o* e *d* é considerada o sinal de saída;
- Acrescentar uma capacitância de carga no nó de saída com o valor do parâmetro Cl, caso se queira alterar esse valor pela plataforma de projeto;
- Incluir, caso o circuito seja do tipo VCO, uma fonte de tensão de controle com magnitude igual ao parâmetro VC. Esse parâmetro será alterado pela plataforma de projeto para avaliar o desempenho do VCO;
- Acrescentar um comando de condição inicial, caso esteja utilizando a análise SNOSE, como exemplo: `.IC o=0 d='VDD'`;
- A fonte de alimentação do circuito deve ser nomeada de V1 para o cálculo da potência consumida pelo circuito.

O circuito deve ser colocado em uma pasta criada de maneira correta nos diretórios do *programa*. O Apêndice A contém a descrição dos circuitos osciladores testados neste trabalho: osciladores em anel com três e cinco inversores; osciladores a cristal de 1,0 MHz e de 32,768 kHz; e oscilador controlado por tensão.

Devem ser colocados na interface gráfica de dados da plataforma de projeto, Figura 10, os valores dos critérios de desempenho que se deseja, (1), os limites de exploração dos parâmetros a serem otimizados e valores iniciais (2) e pesos aplicados na função objetivo (1). Nessa mesma interface gráfica devem ser selecionados os modelos dos transistores (3) a serem utilizados e o algoritmo meta-heurístico (4). Destaca-se, que o usuário pode escolher o circuito que será otimizado pela plataforma nessa interface (7). Em outra interface gráfica, a de execução apresentada na Figura 11, são colocados o número de otimizações que se deseja executar, (1), o número máximo de indivíduos testados por otimização, (2), e a semente para a geração de soluções iniciais (caso não seja colocado nenhum valor nesse campo, a solução inicial é extraída da interface de dados) (3).

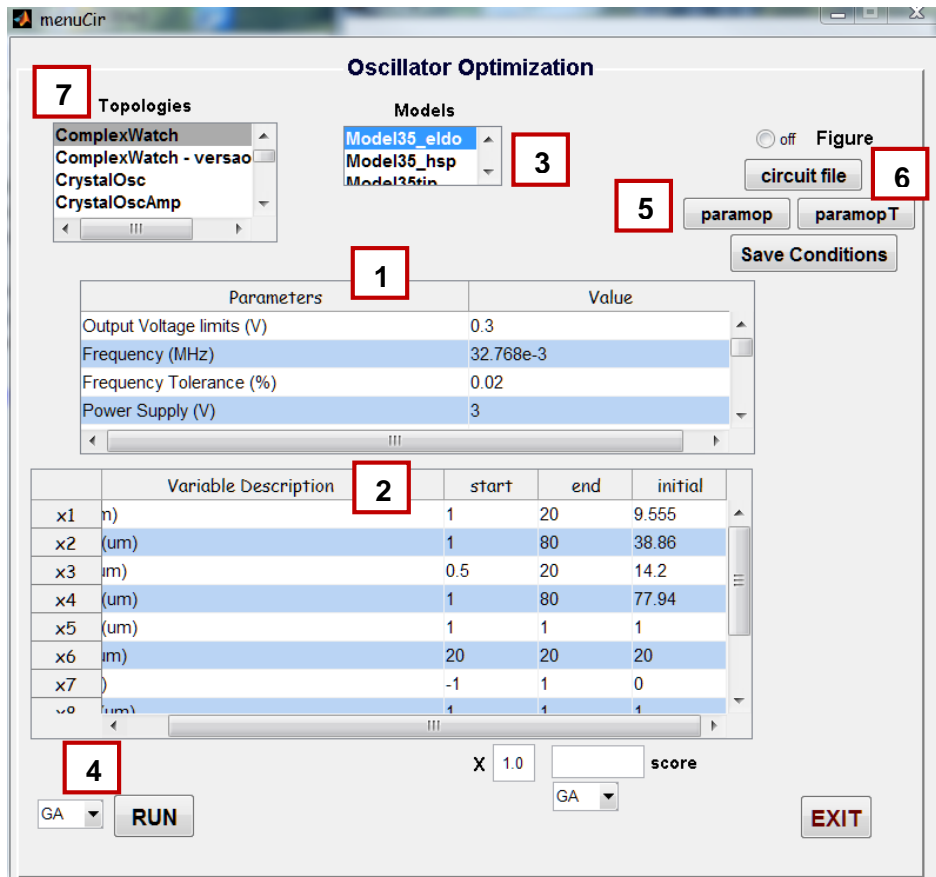


Figura 10: Interface Gráfica de dados.

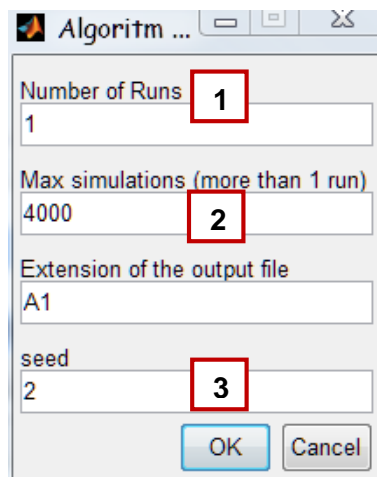


Figura 11: Interface Gráfica de execução.

Na Figura 12, o fluxograma dos processos executados pela plataforma de projeto durante a otimização do circuito é mostrado. No início de cada otimização, o algoritmo meta-heurístico irá gerar a população inicial de soluções, cada indivíduo será analisado através da **função objetivo** (*fitness.m* descrita nos Apêndices B e C). Conforme visto na Figura 12, a função objetivo é responsável por três tarefas: a geração do arquivo de simulação, a simulação elétrica do circuito e o cálculo do *score*. As medidas dos critérios de desempenho ocorrem dentro da simulação elétrica.

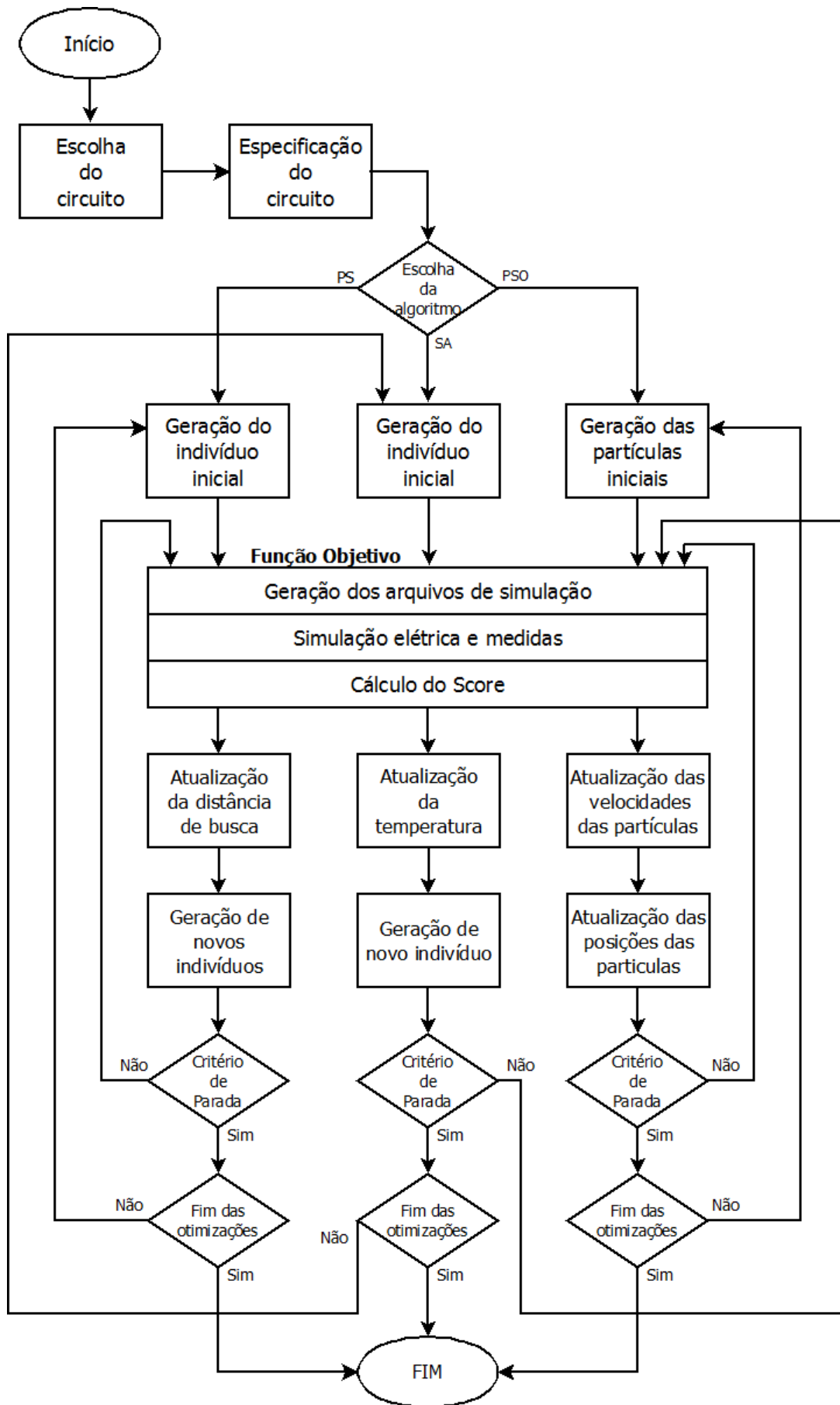


Figura 12: Fluxograma da plataforma de projeto.

A função *fitness.m* faz uso da função *param_mod.m*, também descrita nos Apêndices B e C, para gerar o arquivo de simulação. Esse arquivo contém: os parâmetros do circuito; os comandos das simulações e das medidas que serão realizadas; os comandos de inclusão da descrição do circuito feita pelo usuário; e o comando de inclusão do arquivo com os modelos escolhidos pelo usuário para os dispositivos elétricos e eletrônicos.

A simulação elétrica é chamada pela função *fitness.m* e após a simulação terminar, a função objetivo extrai as informações necessárias dos arquivos de saída da simulação. Com os valores dos critérios de desempenho obtidos, a função objetivo calcula as funções de avaliação e, portanto, obtém o *score* do circuito.

Após cada indivíduo da população ser analisado, o algoritmo metaheurístico utiliza os valores da função objetivo dos indivíduos para atualizar seus parâmetros internos: temperatura no Arrefecimento Simulado; distância de busca no algoritmo Busca Padrão; e as velocidades das partículas na Otimização por Enxame de Partículas.

Com bases nos novos valores de parâmetros internos, o algoritmo metaheurístico utilizado gera uma nova população de soluções que será analisada em seguida. No caso da Otimização por Enxame de Partículas, a nova população será as partículas do enxame em suas novas posições após a movimentação do enxame. Essa nova população será analisada pela função objetivo, repetindo os processos descritos acima.

A cada população, os critérios de parada dos algoritmos meta-heurísticos são analisados. Caso algum critério seja atingido, a otimização é finalizada, e é verificado se o número de otimizações desejadas foi realizado. Se o número de otimizações desejadas foi realizado, o projeto é finalizado; senão, inicia-se uma nova otimização.

Enquanto a otimização é executada, a plataforma fornece dois arquivos: um arquivo, chamado *paramop*, contendo os parâmetros e o desempenho do melhor circuito encontrado pela otimização corrente até o momento (que pode ser usado para acompanhar a evolução de cada iteração) e outro arquivo, chamado *paramopT*, contendo os parâmetros e o desempenho do melhor circuito encontrado por todas as otimizações executadas até o momento. Ao fim da otimização, o arquivo *paramopT* terá os parâmetros e o desempenho do melhor circuito, que é então o circuito projetado. Os arquivos *paramop* e *paramopT* podem ser acessados durante a execução através da interface gráfica de dados (Figura 10, (5) e (6)).

Como a convergência da otimização depende de diversos fatores - dentre eles a aleatoriedade - e como algoritmos meta-heurísticos não necessariamente encontram o mínimo local, o projeto de um circuito eletrônico por meio destes métodos deve

compreender diversas rodadas de otimizações e, quando possível, o refinamento dos resultados por um segundo tipo de algoritmo meta-heurístico.

Nesse trabalho, na maioria dos projetos de circuitos, foi utilizada uma rodada com três otimizações por Enxame de Partículas. No projeto do circuito oscilador com cristal de relógio, por sua vez, foram utilizadas três rodadas com duas otimizações por Enxame de Partículas cada. Os circuitos obtidos foram, posteriormente, refinados pelos métodos de Busca Padrão e de Arrefecimento Simulado. O melhor circuito obtido para cada projeto foi, ao fim, verificado através de simulações elétricas aplicando análises diferentes daquelas empregadas na otimização.

3.2 Ferramentas utilizadas

A plataforma de projeto de circuitos através de algoritmos meta-heurísticos desenvolvida neste trabalho utiliza diversas ferramentas de *software*. Elas serão descritas a seguir.

3.2.1 MatLab

A ferramenta MatLab é uma plataforma de desenvolvimento e interpretação de aplicativos de natureza técnica com linguagem de programação própria de alto nível (MATHWORKS, 2014a). A ferramenta MatLab foi escolhida como plataforma para nossa implementação por diversas razões, entre elas:

- a facilidade de programação;
- a facilidade no desenvolvimento de interfaces gráficas;
- a facilidade no desenvolvimento de interfaces para outras ferramentas (o simulador de circuitos elétricos, por exemplo);
- a utilização do *Toolbox Optimization Tool* do MatLab (MATHWORKS, 2014b) que já implementa vários algoritmos de otimização, entre eles versões dos algoritmos de Arrefecimento Simulado, do Busca Padrão e de Algoritmos Genéticos. Observa-se que o Algoritmo Genético não foi aplicado neste trabalho por ser muito demorado;
- a facilidade de encontrar algoritmos meta-heurísticos já implementados em linguagem MatLab.

O algoritmo Otimização por Enxame de Partículas foi implementado em código MatLab. Deve-se acrescentar que duas modificações foram realizadas no código do Arrefecimento Simulado do *Optimization Tool* para esta aplicação:

- o reaquecimento foi alterado para que fosse realizada toda vez que a temperatura atingisse 0,0001;
- foi acrescentado uma nova condição de parada para o algoritmo: o número de iterações realizadas não pode ultrapassar um valor máximo.

3.2.2 HSpice RF

O simulador de circuitos elétricos utilizado para a simulação e teste dos circuitos na plataforma de desenvolvimento foi o simulador HSpice RF, versão 2008. Segundo o *HSPICE User Guide: RF Analysis* (2008a, p.1), esse simulador possui um conjunto especial de análises e ferramentas que auxiliam no projeto de circuitos RF e de alta frequência. Segundo o mesmo guia, o HSpice RF foi construído a partir do HSpice convencional (2008b, 2008c), adicionando apenas novos métodos de análises.

O simulador HSpice RF foi escolhido neste trabalho pois é adequado para simulações de circuitos osciladores nas mais diversas frequências, inclusive em alta frequências. Ele permite, também, a utilização de diversos tipos de análises de oscilação no domínio do tempo e no domínio da frequência. Os algoritmos computacionais presentes no simulador permitem determinar a frequência de oscilação de um circuito oscilador, a amplitude de oscilação e o valor de outros critérios de desempenho do circuito.

Dentre os algoritmos e tipos de simulações presentes no *software* HSpice RF, os utilizados nesse trabalho foram:

Transient Analysis (TRAN)

A análise de transiente (TRAN) (SYNOPSIS, 2008b) soluciona as equações diferenciais, algébricas e não lineares que representam o circuito, buscando determinar o valor das grandezas elétricas do circuito em função do tempo.

A análise de transiente, porém, não permite a determinação do *jitter* e do ruído de fase dos sinais, assim como também não é a análise mais adequada para aplicar a osciladores com alto fator de qualidade.

Harmonic Balance for Oscillator Analysis (HBOSC)

A análise de balanço harmônico para osciladores (HBOSC) acrescenta a frequência fundamental de oscilação na lista de variáveis desconhecidas do circuito. Por possuir uma variável a mais, a fase de uma das variáveis desconhecidas (normalmente uma tensão nodal) é considerada como referência de fase com valor igual a zero

(SYNOPTSYS, 2008a, p.214-215). O circuito passa a possuir duas referências para o cálculo: o nó de referência de tensão e o nó de referência de fase.

O algoritmo segue uma técnica semelhante à descrita por Ngoya (NGOYA et al.; 1995) que utiliza uma fonte de tensão teste aplicada internamente no circuito para encontrar a frequência e a amplitude de oscilação do circuito. A frequência de oscilação é encontrada, ao analisar a resistência da fonte de teste, que é um curto circuito na frequência de oscilação do circuito e um circuito aberto caso contrário (SYNOPTSYS, 2008a, p.214-215).

Essa análise permite a obtenção da amplitude e da fase, tanto na frequência fundamental da oscilação quanto nos seus harmônicos, e do jitter dos sinais em qualquer nó do circuito e da potência consumida. O algoritmo é executado no domínio da frequência, porém, aplicando a transformada inversa de Fourier (TIF) aos resultados encontrados, a análise também retorna resultados no domínio do tempo.

Assim, a simulação HBOSC prove todas as informações necessárias para a análise de circuitos osciladores nesse trabalho. Destaca-se que o método HBOSC é o mais eficiente para analisar osciladores harmônicos (SYNOPTSYS, 2008, p.212). O algoritmo HBOSC pode ser inicializado com os resultados obtidos por uma análise de transiente através do comando HBINIT, essa opção não é aconselhada para circuitos osciladores em anel (SYNOPTSYS, 2008, p.219).

Oscillator Analysis Using Shooting Newton (SNOSC)

A análise de osciladores usando *Shooting Newton* (SNOSC) é mais adequada para circuitos que possuam as seguintes características (SYNOPTSYS, 2008a, p.222):

- Baixo fator de qualidade;
- Operam baseados em atrasos de propagação de sinais;
- Possuem comportamento fortemente não linear;
- Sinais de saída são lineares por partes ou com forma de onda quadrada.

Dessa forma, esse método de análise de oscilação é mais recomendado para circuitos osciladores em anel, principalmente se constituídos de inversores, e osciladores de relaxação. Esse algoritmo visa solucionar o estado estacionário do circuito no domínio do tempo, porém pode ser executado no domínio da frequência.

Como a análise por balanço harmônico, o objetivo dessa análise é encontrar a variável desconhecida adicional: período do sinal de oscilação do circuito. Assim, é realizada uma simulação transiente para obter os valores iniciais necessários para a resolução. O manual do HSpice RF (SYNOPTSYS, 2008a, p.223) recomenda que, para a análise de osciladores, o tempo de sinal analisado na simulação transiente seja maior

que um período da oscilação esperada. Com isso o circuito terá seus sinais estabilizados no transiente.

A análise utilizando Shooting Newton também possibilita a obtenção do ruído de fase e do jitter dos sinais nos nós do circuito, assim como a representação dos sinais elétricos nos domínios da frequência e do tempo.

3.2 Circuitos projetados

Os circuitos osciladores projetados nesse trabalho foram: circuitos osciladores em anel, circuitos osciladores a cristal e circuitos osciladores tanque controlados por tensão. Em todos os circuitos eletrônicos projetados nesse trabalho foi utilizada a tecnologia de fabricação CMOS 0,35 μ m da *Austria Micro System* (AMS 0,35 μ m) (2009a, 2009b), com modelos de simulação nível 49 para HSpice, BSIM3v3, com parâmetros típicos (SYNOPTISYS, 2008d). Os principais parâmetros para os transistores NMOS e PMOS nessa tecnologia são mostrados na Tabela 1 e Tabela 2.

Tabela 1: Parâmetros elétricos dos transistores NMOS.

Parâmetro	Mínimo	Típico	Máximo	Unidade
V_{TH} (canal longo)	0,36	0,56	0,56	V
V_{TH} (canal curto)	0,4	0,5	0,6	V
$K_{PN} = \mu_n C_{ox}$	150	170	190	$\mu A/V^2$

Tabela 2: Parâmetros elétricos dos transistores PMOS.

Parâmetro	Mínimo	Típico	Máximo	Unidade
V_{TH} (canal longo)	-0,58	-0,68	-0,78	V
V_{TH} (canal curto)	-0,55	-0,65	-0,75	V
$K_{PP} = \mu_p C_{ox}$	48	58	68	$\mu A/V^2$

3.2.1 Circuitos osciladores em Anel

Nessa categoria de circuitos, esse trabalho utilizou duas topologias: oscilador com três inversores CMOS em anel (Figura 13); e oscilador com cinco inversores CMOS em anel (Figura 14). Esses circuitos, apesar da simplicidade, não possuem as

características ideais para serem analisados pelo método de balanço de oscilação harmônica (HBOSC), mas possuem as características para serem analisados através do SNOSEC ou através da inicialização de transiente (HBINIT) dentro da análise HBOSC.

Neste trabalho esses circuitos foram utilizados, primeiro, para avaliar as versões preliminares de funções objetivo que utilizavam a análise de transitório (TRAN) na avaliação dos circuitos, e também para testar a eficiência da plataforma.

Como já foi visto anteriormente, esses circuitos utilizam o atraso de propagação para grandes sinais dos inversores para gerar a oscilação desejada. Para que o circuito possua *duty cycle* igual a 50%, a relação W/L do transistor PMOS deve ser igual K_{PN}/K_{PP} vezes a relação W/L do transistor NMOS. Para os parâmetros típicos de fabricação na tecnologia AMS 0,35 μ m seria $K_{PN}/K_{PP} = 2,59$. Os atrasos de propagação podem ser ajustados dimensionando-se as capacitâncias de carga dos inversores e a largura e o comprimento de canal dos transistores. A frequência é então inversamente proporcional ao número de inversores (N_I) no anel e aos atrasos de propagação:

$$f_o = \frac{1}{N_I(T_{PD} + T_{PS})} \quad (12)$$

em que T_{PD} é o atraso de propagação para descida e T_{PS} é o atraso de propagação para subida.

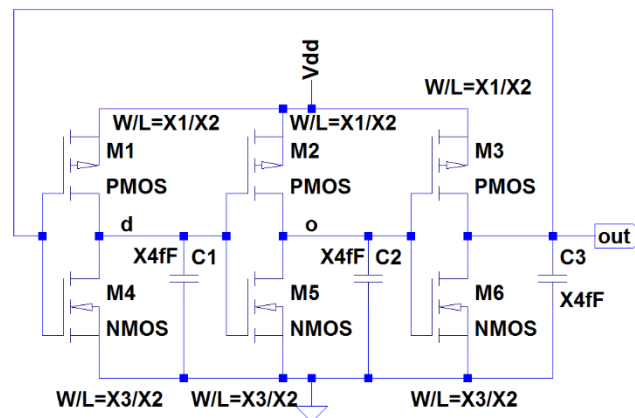


Figura 13: Circuito oscilador em anel com três inversores. Estão indicados na figura os valores de W e L dos transistores

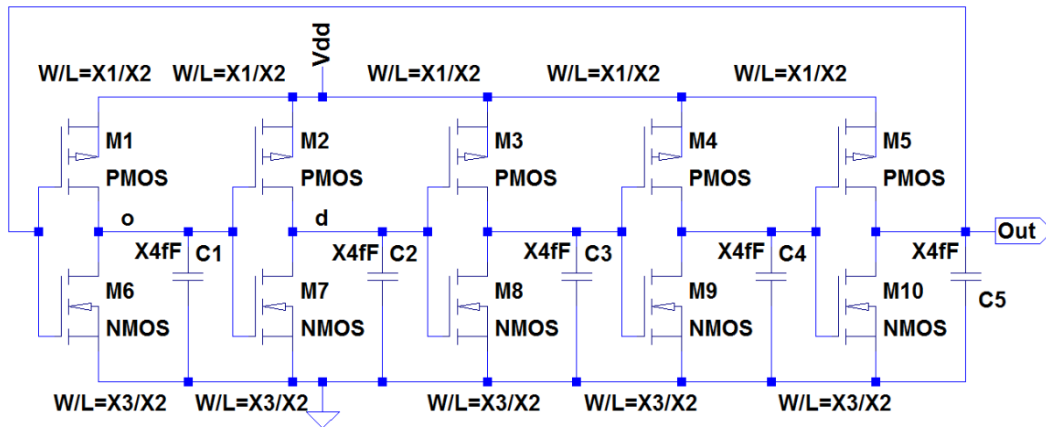


Figura 14: Circuito oscilador em anel com cinco inversores. Estão indicados na figura os valores de W e L dos transistores

No circuito da Figura 13, as variáveis de ajuste estão descritas na Tabela 3; no circuito da Figura 14, as variáveis de ajuste estão descritas na Tabela 4.

Tabela 3: Variáveis de ajuste do circuito oscilador em anel com três inversores.

Variável	Descrição	Dispositivo
X1	Largura do canal (μm)	M1, M2, M3
X2	Comprimento do canal (μm)	M1, M2, M3, M4, M5, M6
X3	Largura do canal (μm)	M4, M5, M6
X4	Capacitância (fF)	C1, C2, C3

Tabela 4: Variáveis de ajuste do circuito oscilador em anel com cinco inversores.

Variável	Descrição	Dispositivo
X1	Largura do canal (μm)	M1, M2, M3, M4, M5
X2	Comprimento do canal (μm)	M1, M2, M3, M4, M5, M6, M7, M8, M9, M10
X3	Largura do canal (μm)	M6, M7, M8, M9, M10
X4	Capacitância (fF)	C1, C2, C3, C4, C5

3.2.2 Circuitos osciladores a cristal

Os cristais ressonantes são representados nas simulações por modelos de quatro elementos passivos (SEDRA, SMITH; 2004, p.1183) como o mostrado na Figura 15. Foram utilizados modelos de dois cristais nos projetos: um cristal de 1,0 MHz; e um cristal de relógio (32768 Hz).

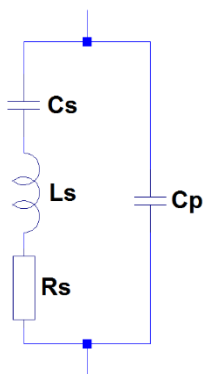


Figura 15: Modelo de quatro elementos do cristal

A frequência de oscilação ω_0 e o fator de qualidade de um cristal Q podem ser determinados pelas relações presentes na Equação 13:

$$\omega_0 = \frac{1}{\sqrt{L_S C_S}} \quad e \quad Q = \frac{1}{R_S} \sqrt{\frac{L_S}{C_S}} \quad (13)$$

Na Tabela 5, encontram-se os parâmetros do modelo de quatro elementos para os cristais utilizados nesse trabalho, junto com o fator de qualidade Q de cada cristal modelado:

Tabela 5: Parâmetros dos modelos de quatro elementos dos cristais.

Frequência (kHz)	Cp (pF)	Cs (fF)	Ls (H)	Rs (Ω)	Q
32,768	1,5	3	7863	30000	53965
1000	0,017	3,5	0,015	20	3273

Nessa categoria de circuitos foi utilizada uma topologia de oscilador com cargas ativas e dois estágios amplificadores. Na Figura 16, a topologia utilizada é mostrada. Nesse circuito, assim como no circuito da Figura 17, as variáveis de ajuste estão descritas na Tabela 6. É importante notar que: X_6 determina a corrente fornecida pela fonte de corrente I_1 , dada pela relação $I_1 = 10^{X_6} \mu A$; e os fatores M são a quantidade de transistores associados em paralelo do transistor em questão considerando a parte inteira da variável correspondente.

Tabela 6: Variáveis de ajuste dos circuitos osciladores a cristal.

Variável	Descrição	Dispositivo
X1	Comprimento do canal (μm)	M1, M2, M3
X2	Largura do canal (μm)	M1, M2, M3
X3	Comprimento do canal (μm)	MN1, MN2
X4	Largura do canal (μm)	MN1, MN2
X5	Resistência ($\text{k}\Omega$)	R1
X6	Fator de corrente elétrica ($10^{X6} \mu\text{A}$)	I1
X7	Fator M	M1
X8	Fator M	M3

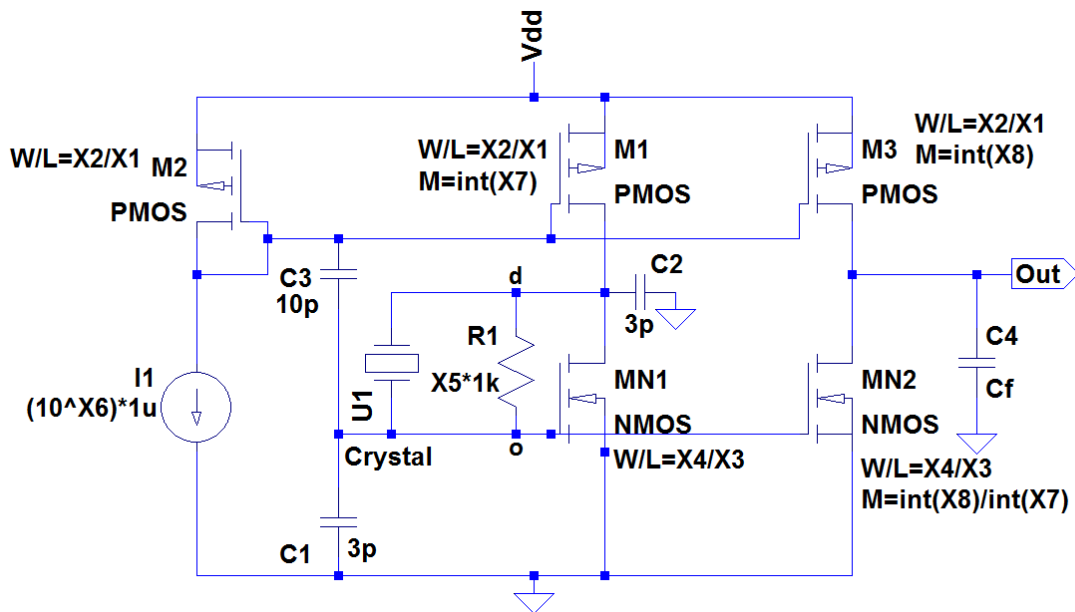


Figura 16: Circuito oscilador a cristal. Estão indicados na figura os valores de W, L e M dos transistores

Essa topologia foi baseada nos trabalhos apresentados por Werner Thommen (1999, p.146-149) e por Willy Sansens (2006, p.689). No trabalho de Thommen, essa topologia de circuito oscilador foi utilizada por ter duas características essenciais para o baixo consumo de potência: baixa tensão de alimentação e baixo consumo de corrente.

O núcleo central desse circuito é o primeiro estágio amplificador, transistores M_1 e M_{N1} , que, junto com o cristal, é responsável pela oscilação do circuito. Esse estágio está na configuração *push-pull* e é polarizado pelo espelho de corrente M_2 e M_1 e pelo resistor R_1 . O cristal faz a realimentação entre a entrada e a saída desse amplificador sendo C_1 e C_2 as capacitâncias dos terminais do cristal. O segundo estágio amplificador, transistores M_3 e M_{N2} , forma também um amplificador *push-pull*. Esse estágio é responsável pela amplificação do sinal da saída. O capacitor C_3 é responsável

por polarizar o circuito de tal maneira que os estágios amplificadores operam com características *push-pull*. Há, adicionalmente, no circuito, a fonte de corrente I_1 de polarização e a capacitância C_f de carga.

Observa-se na Figura 16 que as dimensões dos transistores são escolhidas de tal maneira que seja obedecida a relação

$$\frac{\frac{W_{P3}}{L_{P3}}}{\frac{W_{N2}}{L_{N2}}} = \frac{\frac{W_{P1}}{L_{P1}}}{\frac{W_{N1}}{L_{N1}}} \quad (14)$$

Essa relação de dimensões permite que os dois estágios de amplificador estejam polarizados na mesma situação e, assim, o segundo estágio é capaz de amplificar o sinal vindo do primeiro estágio. Esse circuito, por utilizar um cristal ressonante com alto Q, é um ótimo caso para se testar a plataforma de projeto desenvolvida.

Uma variação da topologia apresentada na Figura 16 é a topologia apresentada na Figura 17, que tem as mesmas variáveis de ajuste da topologia anterior. A principal diferença entre elas está no capacitor C3, Figura 16, ligando à porta do transistor MP1 a porta do transistor MN1, que foi retirado não permitindo que haja amplificação *push-pull* nos dois estágios amplificadores. A entrada do segundo estágio amplificador também foi alterada sendo ligada ao outro terminal do cristal. Essas alterações transformaram o primeiro estágio em um amplificador de fonte comum, aumentando a estabilidade do circuito, mas diminuindo o ganho.

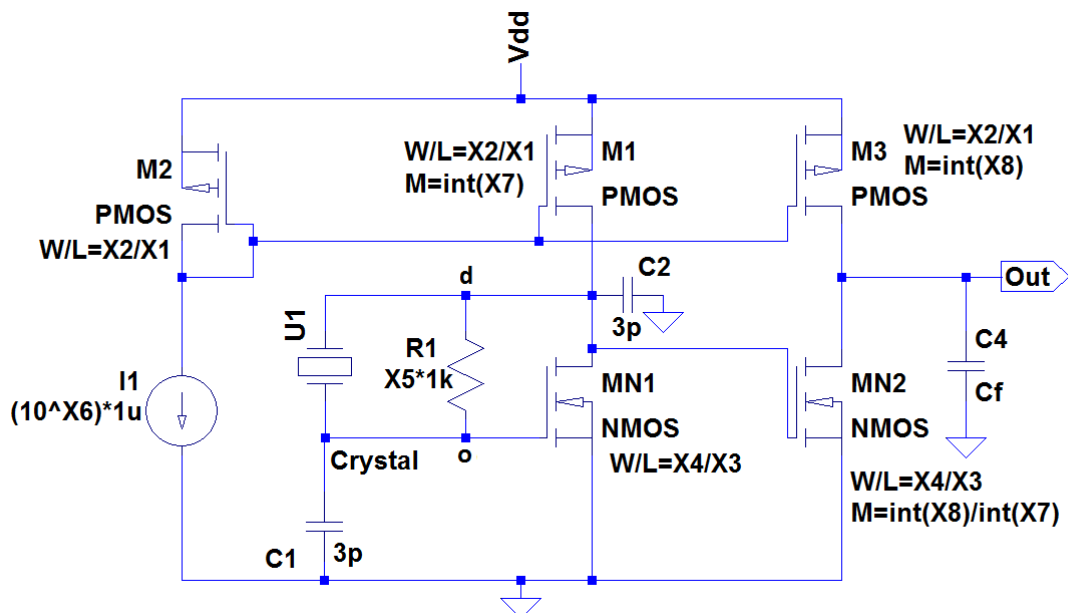


Figura 17: Circuito oscilador a cristal modificado. Estão indicados na figura os valores de W, L e M dos transistores

3.2.3 Circuito oscilador controlado por tensão

A topologia utilizada aqui, apresentada na Figura 18, foi baseada em um oscilador diferencial complementar com ressonador do tipo tanque sintonizável (BUNCH, RAMAN, 2003, p.1329; PEREIRA et al., 2013, p.2). O ressonador tanque sintoniza a frequência de oscilação e o ajuste da frequência é feito pela variação dos capacitores C_1 e C_2 , varactors, que dependem de tensão V_C . A amplificação da oscilação e a recuperação da perda de energia do sinal são realizadas pelos elementos ativos do circuito, que atuam como uma resistência negativa (PEREIRA et al., 2013, p.2).

Como no circuito apresentado por Bunch, foi utilizado mais de um *varactor* MOS em paralelo. Neste trabalho o número de *varactors* necessários para a oscilação na frequência requerida foi deixada como parâmetro de otimização. Outros fatores de diferenciação do circuito desse trabalho em relação ao de Bunch são a utilização de apenas um indutor ideal no circuito tanque, a não utilização de um circuito de polarização, ligando os transistores PMOS e NMOS diretamente a V_{DD} e terra, respectivamente, e a não utilização de um segundo estágio amplificador.

Neste circuito as variáveis de ajuste estão descritas na Tabela 7. É importante notar que: X5 é o comprimento do canal de cada capacitor MOS, em μm , que forma C1 e C2; X6 é a largura do canal de cada capacitor MOS em μm ; e X8, o fator M de paralelismo de cada capacitor MOS para forma C1 e C2.

Tabela 7: Variáveis de ajuste do circuito oscilador controlado por tensão.

Variável	Descrição	Dispositivo
X1	Comprimento do canal (μm)	M1, M2
X2	Largura do canal (μm)	M1, M2,
X3	Comprimento do canal (μm)	M3, M4
X4	Largura do canal (μm)	M3, M4
X5	Comprimento do canal (μm)	C1, C2
X6	Largura do canal (μm)	C1, C2
X7	Indutância (nH)	L1
X8	Fator M	C1, C2

As capacitâncias C1 e C2 foram implementadas com transistores PMOS.

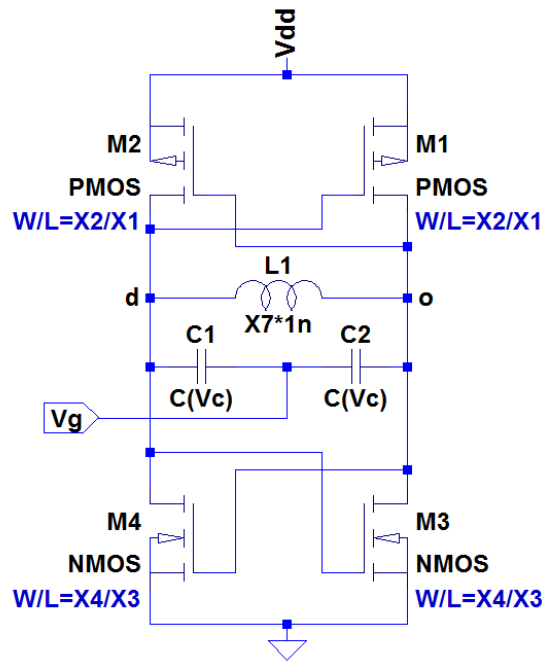


Figura 18: Circuito VCO utilizado. Estão indicados na figura os valores de W e L dos transistores

Na escolha do tipo de *varactor*, escolheu-se utilizar a topologia para operação na região de inversão comentada anteriormente, apesar da alta variação da capacitância para pequenas variações de tensão sobre seus terminais. Esse tipo de *varactor* é obtido através de um transistor PMOS comum não necessitando de um modelo específico. Testes com capacitores MOS em outras regiões de operação foram realizados, porém a otimização obteve sucesso apenas quando a implementação do capacitor MOS foi utilizada com operação na região de inversão.

3.3 Função objetivo

A função objetivo, como já foi dito, é a peça fundamental em um algoritmo meta-heurístico, pois é sua responsabilidade determinar para o algoritmo o desempenho geral do indivíduo através de uma pontuação. Nesse trabalho quanto menor o valor da função objetivo melhor é o desempenho geral do indivíduo. Logo, os algoritmos meta-heurísticos serão utilizados para minimizar o valor da função objetivo.

Considerou-se, neste trabalho, como desempenho geral uma medida ponderada do desempenho do indivíduo, os circuitos osciladores, em diversos critérios. O valor da função objetivo é assim calculado por meio dos pesos de ponderação (P_i) e dos valores de cada critério de desempenho que serão determinadas pelo que será chamado de funções de avaliação (Fa_i):

$$F_o = \sum_{i=1}^n P_i Fa_i. \quad (15)$$

O valor da função objetivo calculado dessa forma é chamado de valor agregado pois agrega, em um único número, o desempenho geral do indivíduo.

O grande desafio na utilização de algoritmos meta-heurísticos na otimização de sistemas é a escolha correta dos critérios de desempenho, das funções de avaliação e dos respectivos pesos de ponderação. Na otimização de circuitos, os critérios de desempenho devem ser capazes tanto de avaliar o correto funcionamento do circuito para o problema proposto quanto a sua performance nas mais diversas aplicações.

Na plataforma de projeto, o usuário possui a liberdade para alterar os pesos de ponderação da função objetivo. Além disso, ele tem à disposição diversos critérios de desempenho de forma que possa achar circuitos que satisfaçam as mais variadas aplicações.

A seguir são listados todos os critérios de desempenho e suas respectivas funções de avaliação. As diferenças entre as funções de avaliação para os circuitos osciladores de frequência fixa e os circuitos VCO serão detalhadas quando houver. Caso não sejam detalhadas, fica implícito que a função de avaliação é a mesma, porém, será utilizado em seu cálculo para um VCO o pior valor obtido do critério de desempenho nas simulações com a variação da tensão de controle.

3.3.1 Frequência

A frequência de oscilação do sinal de saída do circuito é o mais importante parâmetro de desempenho de um circuito oscilador. Assim, esse é o principal objeto de medida nas simulações a serem realizadas. A frequência de oscilação é obtida diretamente em simulações HBOSC e SNOSC.

Na simulação do tipo TRAN, a frequência de oscilação pode ser obtida, após um longo intervalo de simulação, através do inverso do período medido do sinal. O período, em uma simulação TRAN, é obtido pela diferença de tempo entre duas ocorrências consecutivas do valor médio do sinal, para o sinal subindo (ou descendo).

A função de avaliação para a frequência no oscilador simples é

$$Fa_1 = \begin{cases} \frac{|F_M - F_D|}{F_D}, & \text{se } \frac{|F_M - F_D|}{F_D} > T_F, \\ 0,0 & \text{se } \frac{|F_M - F_D|}{F_D} \leq T_T; \end{cases} \quad (16)$$

em que:

F_D = a frequência desejada fornecida pelo usuário;

F_M = a frequência medida;

T_F = o fator de tolerância para a frequência fornecido pelo usuário.

No VCO, a função de avaliação para a frequência será a somatória de várias componentes:

$$Fa_1 = C_F \sum_{i=1}^n fav_i, \quad (17)$$

em que:

$$fav_i = \begin{cases} \frac{|F_{Mi} - F_D|}{F_D}, & \text{se } \frac{|F_{Mi} - F_D|}{F_D} > T_F, \\ 0,0 & \text{se } \frac{|F_{Mi} - F_D|}{F_D} \leq T_F; \end{cases}$$

n = o número de tensões de controle em que o VCO é avaliado;

F_{Mi} = a frequência medida para tensão de controle V_i , e $1 \leq i \leq n$;

$$C_F = \begin{cases} \frac{|FF_M - FF_D|}{FF_D}, & \text{se } FF_M < FF_D - T_F, \\ 1,0 & \text{se } FF_M \geq FF_D - T_F; \end{cases}$$

FFD = a faixa de sintonia desejada;

FFM = a faixa de sintonia medida do VCO.

O corretor CF é utilizado para que circuitos, com faixa de sintonia menor que a faixa desejada, sejam penalizados.

Observe que para a frequência, caso os valores medidos estejam próximos do desejado, a função de avaliação é zerada.

3.3.2 Excursão do sinal de saída

A excursão do sinal de saída é importante em vários aspectos. Um circuito oscilador sempre faz parte de um sistema eletrônico mais complexo e sua saída servirá como entrada para outros blocos do sistema. Sua amplitude deverá, portanto, atingir níveis mínimos para servir aos outros blocos funcionarem. Adicionalmente, dado o comportamento matemático dos algoritmos de simulação, eles podem encontrar uma frequência de oscilação para o circuito, porém com amplitude do sinal de saída muito pequena (abaixo de miliVolts), o que não garante haver oscilação de fato. Assim o valor da tensão de saída será também importante para verificar se o oscilador realmente funciona.

A plataforma de projeto implementada possui comportamento variável dependendo da entrada especificada pelo usuário para a excursão do sinal. Caso sejam fornecidos dois valores de tensão elétrica, chamado caso 1, esses valores serão considerados como valores de pico da oscilação desejada (circuitos com valores de pico

acima do desejado não são penalizados). Caso seja fornecido apenas um valor, chamado caso 2, esse valor será interpretado como valor de excursão pico a pico mínima desejada.

As equações abaixo descrevem a função de avaliação para cada caso citado acima.

Caso 1:

$$Fa_2 = Vr_1 + Vr_2, \quad (18)$$

em que:

$$Vr_1 = \begin{cases} \frac{|V_{Mmax} - V_{Dmax}|}{V_{Dmax}}, & \text{se } V_{Mmax} < V_{Dmax}, \\ 0,0 & \text{se } V_{Mmax} \geq V_{Dmax}; \end{cases}$$

$$Vr_2 = \begin{cases} \frac{|V_{Mmin} - V_{Dmin}|}{V_{Dmin}}, & \text{se } V_{Mmin} > V_{Dmin}, \\ 0,0 & \text{se } V_{Mmin} \leq V_{Dmin}; \end{cases}$$

VDmax = o valor máximo do sinal desejado;

VDmin = o valor mínimo do sinal desejado;

VMmax = o valor máximo do sinal medido do oscilador;

VMmin = o valor mínimo do sinal medido do oscilador.

Caso 2:

$$Fa_2 = \begin{cases} \frac{|A_M - A_D|}{A_D}, & \text{se } A_M < A_D, \\ 0,0 & \text{se } A_M \geq A_D, \end{cases} \quad (19)$$

em que

$A_M = V_{Mmax} - V_{Mmin}$;

$A_D =$ o valor de amplitude desejada fornecido pelo usuário.

Observe que para a excursão, caso os valores medidos sejam superiores ao desejado, a função de avaliação é zerada.

3.3.3 Potência consumida

Diminuir a potência consumida é um objetivo em projetos de circuitos integrados que se torna cada vez mais importante. Também para circuitos osciladores esse é um parâmetro bastante relevante.

Normalmente, o peso de ponderação da potência na função objetivo é deixado baixo para que a potência seja minimizada apenas quando o oscilador já passou a oscilar da forma desejada.

A função de avaliação para a potência no oscilador é

$$Fa_3 = \frac{P_M}{P_{Ref}} \quad (20)$$

em que:

P_{Ref} = a potência de referência fornecida pelo usuário;

P_M = a potência consumida medida do oscilador.

Observe que para a potência a função de avaliação nunca é zerada. Dessa forma, o algoritmo de otimização procura sempre minimizar o consumo de potência.

3.3.4 Ruído de fase

Em muitas aplicações, principalmente para sistemas de telecomunicações, o ruído de fase é um dos principais critérios de desempenho. O ruído de fase quantifica as flutuações de amplitude e fase devidos às fontes de ruído externas e internas ao circuito (HAJIMIRI, LIMOTYRAKIS; 1999, p.790). Essas flutuações são tipicamente expressas em termo de sua densidade espectral de potência. Para a maioria dos circuitos osciladores, o ruído de fase ocasiona uma modulação de baixa frequência, que cria bandas laterais no espectro de frequência do oscilador em torno da frequência fundamental de oscilação (RAZAVI, 2012, p.537).

Na plataforma de projeto o usuário define n pontos de medida de ruído de fase: para cada um dos pontos o usuário dá a frequência de offset, em relação a frequência fundamental de oscilação, para a medida e o ruído de fase máximo aceito para esse ponto.

A função de avaliação para o ruído de fase no oscilador será a somatória de várias componentes:

$$Fa_4 = \sum_{i=1}^n fap_i, \quad (21)$$

em que:

$$fap_i = \begin{cases} \left| \frac{PN_{Mi} - PN_{Di}}{PN_{Di}} \right|, & \text{se } PN_{Mi} < PN_{Di}, \\ 0,0 & \text{se } PN_{Mi} \geq PN_{Di}; \end{cases}$$

n = o número de frequências de medida fn_i fornecidas pelo usuário;

PN_{Di} = o ruído de fase máximo aceito pelo usuário para a i -ésima frequência de medida fn_i ;

PN_{Mi} = o ruído de fase medido do oscilador para a i -ésima frequência de medida fn_i fornecida pelo usuário, e $1 \leq i \leq n$.

Observe que para o ruído de fase, caso os valores medidos sejam menores que os máximos aceitos, a função de avaliação é zerada.

3.3.5 Distorção harmônica

Em diversas aplicações é interessante garantir que o sinal elétrico de saída do oscilador possua a maior parte da potência concentrada na frequência fundamental de oscilação. A distorção harmônica total é a medida responsável por indicar quanto de potência do sinal periódico não se encontra na frequência fundamental.

A distorção harmônica total é dada pela expressão

$$THD = \frac{\sqrt{\sum_{k=2}^n V_k^2}}{\sqrt{\sum_{l=1}^n V_l^2}}, \quad (22)$$

em que:

n = o número de harmônicas consideradas;

V_1 = a amplitude da componente fundamental;

$V_2, V_3 \dots$ = a amplitude da segunda harmônica, da terceira, etc.

O número de harmônicas utilizadas no cálculo é fornecido pelo usuário, em um máximo de 25.

O THD pode também ser útil na verificação do funcionamento do circuito, dado que um THD elevado pode indicar um mau funcionamento ou falso positivo da simulação.

A função de avaliação para o TDH no oscilador é

$$Fa_5 = \begin{cases} \left| \frac{THD_M - THD_D}{THD_D} \right|, & \text{se } THD_M > THD_D, \\ 0,0 & \text{se } THD_M \leq THD_D, \end{cases} \quad (23)$$

em que:

THD_D = a distorção harmônica total máxima aceita pelo usuário;

THD_M = a distorção harmônica total medida do oscilador.

Observe que para o TDH, caso o valor medido seja menor que o máximo aceito, a função de avaliação é zerada.

Em alguns circuitos, o THD não é um fator importante, como no caso de osciladores de sinais quadrados onde THD é alto. Por outro lado, para esses osciladores é interessante minimizar as harmônicas pares para garantir uma maior qualidade do sinal e um *duty cycle* próximo de 50%. Em função disso criou-se um conceito que foi utilizado na plataforma de projeto: a distorção de harmônicas pares (DHP) dado, de forma análoga a THD, por

$$DHP = \frac{\sqrt{\sum_{k \text{ par}}^n V_k}}{\sqrt{\sum_{l=1}^n V_l}}. \quad (24)$$

É importante ressaltar que a plataforma de projeto deve atender ao projeto de variados circuitos osciladores, que podem gerar outras formas de ondas que não senoidais ou quadradas. Para isso é utilizado um fator de tolerância para o DHP para que o usuário possa criar uma faixa de DHP permitida.

A função de avaliação para o DHP no oscilador é

$$Fa_5 = \begin{cases} \left| \frac{DHP_M - DHP_D}{DHP_D} \right| & \text{se } DHP_M \geq DHP_D - T_{DHP} \text{ ou } DHP_M \leq DHP_D + T_{DHP}, \\ 0,0 & \text{se } DHP_D - T_{DHP} > DHP_M > DHP_D + T_{DHP} \end{cases} \quad (25)$$

em que:

DHP_D = a distorção harmônica par desejada pelo usuário;

DHP_M = a distorção harmônica par medida do oscilador;

T_{DHP} = a tolerância para a distorção harmônica par fornecida pelo usuário.

Observe que para o DHP, caso o valor medido esteja próximo ao desejado, a função de avaliação é zerada.

3.3.6 Área

A área de um circuito integrado depende das dimensões de transistores, resistores, capacitores e indutores, além do *layout* final que depende, por sua vez, da necessidade de casamentos entre componentes, da experiência do projetista e outros fatores. Porém, uma estimativa dessa área pode ser realizada através das dimensões dos dispositivos do circuito.

Na plataforma de projeto está aberto ao usuário a implementação de uma função, escrita em linguagem MatLab, para o cálculo da área estimada do circuito em função das dimensões dos dispositivos. A função de avaliação para a Área no oscilador é

$$Fa_6 = \left| \frac{Ar_e - Ar_{ref}}{Ar_{ref}} \right|, \quad (26)$$

em que:

Ar_{ref} = a área de referência fornecida pelo usuário;

Ar_e = a área estimada do oscilador.

Observe que para a área, de maneira semelhante a função de avaliação para a potência consumida, a função de avaliação não será zerada.

3.3.7 Jitter

O *jitter* é derivado do ruído de fase do circuito e representa a variabilidade do espaçamento entre as transições da oscilação (HAJIMIRI, LIMOTYRAKIS; 1999, p.790). Há diversas maneiras de medição de *jitter*, dentre elas se destacam: medida no tempo (*timing*), medida pela fase, medida em um período, *trackling*, *long-term*, e *jitter* ciclo-a-ciclo.

A razão para a inclusão do *jitter* como critério de desempenho, mesmo ele sendo consequência do ruído de fase, é facilitar a análise da influência do ruído de fase em osciladores que serão aplicados como *clock* em sistemas. Em razão disso, medida de *jitter* escolhida como critério de desempenho foi a medida no tempo.

A função de avaliação para o *jitter* no oscilador é

$$Fa_8 = \begin{cases} \left| \frac{J_M - J_D}{J_D} \right|, & \text{se } J_M > J_D, \\ 0,0 & \text{se } J_M \leq J_D, \end{cases} \quad (27)$$

em que:

J_M = o *jitter* máximo aceito pelo usuário;

J_D = o *jitter* medido do oscilador.

O *jitter* será incluído como critério de desempenho caso o usuário forneça a faixa de frequência em que o *jitter* deverá ser medido. Esse critério de desempenho está disponível apenas para a otimização de circuitos VCO.

Observe que para o *jitter*, caso o valor medido seja menor que o máximo aceito, a função de avaliação é zerada.

3.4 Conjuntos de simulações

Os circuitos gerados pelos algoritmos meta-heurísticos, que fornecem os parâmetros para os dispositivos, são simulados no HSpice RF. As simulações são as responsáveis por determinar os valores dos critérios de desempenho utilizados nas funções de avaliação. Logo, o conjunto de simulações executadas na função objetivo é de extrema importância.

O tipo de análise de simulação inicialmente utilizada para avaliar os circuitos osciladores na plataforma de otimização foi a análise de transiente (TRAN). Essa análise é amplamente utilizada em simulações de circuitos eletrônicos em programas tipo Spice e permite a obtenção de valores de diversos critérios de desempenho de circuitos eletrônicos. Porém, ao se tratar de circuitos osciladores, simulações de transitório apresentam diversas dificuldades.

A primeira dificuldade da análise TRAN aplicada a osciladores é a impossibilidade de se medir diretamente o ruído de fase. A segunda é a necessidade de se aplicar transformadas de Fourier para se obter magnitudes e fases das componentes fundamental e harmônicas dos sinais de saída, que são necessárias para cálculo das distorções harmônicas. A terceira dificuldade é que o tempo de execução, a convergência e a precisão dos resultados depende de parâmetros de simulação tais como: o tempo total analisado na simulação de transiente e o máximo passo de tempo permitido na simulação.

Uma dificuldade adicional relacionada com a análise TRAN está no fato de que a medida dos critérios de desempenho de um oscilador deve ser realizada quando ele atinge o regime permanente. No entanto, cada topologia de circuito oscilador - e até mesmo cada oscilador de determinada topologia - atinge o regime permanente num intervalo de tempo diferente. Isso cria complicações na escolha do tempo total de simulação e do máximo passo permitido. Essa escolha é mais difícil para osciladores com alto fator de qualidade, osciladores com cristal. Esses circuitos podem demorar muito para apresentar oscilações, além de apresentarem resultados muito sensíveis aos valores utilizados no máximo passo de análise.

As dificuldades apresentadas acima motivaram a utilização de análises mais apropriadas para circuitos osciladores, em especial, análises que pudessem determinar a frequência de oscilação corretamente. A utilização da análise de transiente para avaliação do circuito durante a otimização foi abandonada, e, conseqüentemente, o HSpice de uso geral foi substituído pelo HSpice RF. Essa versão, como já explicado, possui análises específicas para circuitos osciladores.

O segundo tipo de simulação aplicado no trabalho foi a análise de balanço harmônico (HBOSC), dada sua eficiência na simulação de osciladores com alta fator de qualidade como os osciladores a cristal. O tempo de execução com esta análise também se mostrou adequado para sua aplicação em otimizações por algoritmos meta-heurísticos. Porém, após algumas otimizações de teste do circuito oscilador a cristal, foi observado que alguns dos circuitos obtidos nas otimizações apresentavam oscilações pelos resultados do HBOSC, mas não oscilavam nem pela análise TRAN nem pela análise SNOSC. A análise de balanço harmônico ainda apresentou resultados que diferiam dependendo de parâmetros utilizados, tais como a tensão de *probe* inicial e o número de harmônicas utilizadas.

Tanto a obtenção de falsos positivos, ou seja, circuitos que não oscilam, mas são avaliados como osciladores, quanto a dependência dos resultados com certos parâmetros desencorajou a utilização do HBOSC na simulação dos circuitos osciladores a cristal gerados pelos algoritmos meta-heurísticos.

Apesar da análise SNOSC ser mais eficaz em circuitos com baixo fator de qualidade e com sinais não senoidais, ela também foi testada dado que as outras análises não apresentaram resultados satisfatórios. A análise SNOSC não apresentou falsos positivos para os circuitos osciladores a cristal.

A possibilidade de uma análise não convergir, seja qual for o tipo de análise, e, conseqüentemente, a plataforma recusar um circuito que na realidade oscila é o chamado falso negativo. Os falsos negativos não são tão maléficos a otimização quanto

os falsos positivos (circuito não possui caráter oscilatório, mas os resultados das simulações elétricas indicam que sim).

Os resultados obtidos com a análise SNOSC não dependeram de valores escolhidos para parâmetros da análise, em oposição ao ocorrido com a análise HBOSC nos circuitos osciladores a cristal. Observou-se, que quanto maior foi o período da simulação de transiente utilizada na inicialização da análise SNOSC, mais rápido o algoritmo convergiu e menos frequente foram as simulações demoradas durante a otimização.

A confiabilidade dos resultados da análise SNOSC, se comparada a outras análises, foi satisfatória, com tempos de execução adequados para sua aplicação à otimização por meio de meta-heurísticas.

Um aspecto importante para a análise do funcionamento e desempenho dos circuitos são as tensões de pico da oscilação do sinal de saída. A amplitude da oscilação, como já citado, é um dos parâmetros que comprova o funcionamento do circuito. Assim sua medida deve ser o mais fiel possível à amplitude de oscilação real. Uma das maneiras de se medi-la é obtendo a amplitude da componente fundamental de oscilação. Porém, em circuitos com sinal de saída não senoidal, essa medida irá diferir do valor real por não considerar as harmônicas.

A solução encontrada para obter as tensões de pico foi utilizar resposta no domínio do tempo do circuito, obtida a partir da transformada inversa de Fourier na análise HBOSC ou obtida diretamente na análise SNOSC, e então medir os valores máximo e mínimo do sinal de saída. Esse método apresenta desvantagens por desconsiderar o Efeito Gibbs na transformada de inversa de Fourier, porém apresentou resultados satisfatórios e próximos ao resultado fornecido pelas simulações de transitório.

As simulações que serão aplicadas aos circuitos variam de acordo com o tipo de circuito oscilador otimizado pela plataforma de projeto. Os parâmetros utilizados nas simulações variam conforme a especificação feita pelo usuário.

Na otimização de circuitos osciladores com frequência fixa, foi utilizada a simulação SNOSC para obter os valores dos critérios de desempenho do circuito. O circuito projetado final foi posteriormente analisado com TRAN e HBOSC para fins de verificação.

Na otimização de circuitos VCO, a análise SNOSC não apresentou resultados satisfatórios ao verificar a existência da oscilação, retornando ao final do projeto circuitos com amplitude de oscilação insatisfatória. Assim, nesse tipo de circuitos foi utilizada a simulação HBOSC para obter os valores dos critérios de desempenho do circuito. O

circuito projetado final foi posteriormente analisado com TRAN e SNOSC para fins de verificação.

Os parâmetros de desempenho obtidos na simulação SNOSC para circuitos com frequência fixa são:

1. Frequência de oscilação;
2. Tensão máxima do sinal;
3. Tensão mínima do sinal;
4. Potência consumida pela fonte de alimentação;
5. Amplitude das componentes fundamental e harmônica do sinal;
6. Ruído de fase.

Os parâmetros de desempenho obtidos na simulação HBOSC para circuitos VCO são:

1. Frequência de oscilação;
2. Tensão máxima do sinal;
3. Tensão mínima do sinal;
4. Potência consumida pela fonte de alimentação;
5. Amplitude das componentes fundamental e harmônica do sinal;
6. Ruído de fase;
7. *Jitter*.

Os comandos utilizados nas simulações estão no Apêndice D. A sintaxe dos comandos e parâmetros encontram-se explicados em (SYNOPSISYS, 2008a, 2008b, 2008c).

4 Resultados

Nesse capítulo serão apresentados os resultados de otimização realizados com osciladores em anel com três e cinco inversores, osciladores a cristal de 1,0 MHz e de 32768 Hz, além de um oscilador controlados por tensão.

4.1 Circuito oscilador com três inversores em anel

Foi realizado o projeto de um circuito oscilador com três inversores em anel como ilustrado na Figura 13. As especificações do oscilador desejado estão dadas abaixo:

- Frequência central: 900 MHz \pm 0,01%;
- Tensão de Saída: 1,0 V de amplitude;
- Tensão de Alimentação: 2,0 V;
- Potência Consumida Máxima: 1,0 mW;
- Distorção Harmônica Par: 10%;
- Distorção Harmônica Total: 50%;
- Ruído de Fase @9,0 kHz: -40 dB/Hz.

Os pesos aplicados na função objetivo estão indicados na Tabela 8.

Tabela 8: Pesos da função objetivo do circuito oscilador com três inversores em anel.

Critério de desempenho	Frequência	Tensão de Saída	Potência	Distorção par	Distorção Total	Ruído de Fase
Peso	10	100	0,1	0,1	1,0	0,1

Abaixo são apresentados detalhes sobre a otimização aplicada ao circuito:

- Método: PSO;
- Número de rodadas de otimização: uma;
- Número de otimizações por rodada: três;
- Máximo número de indivíduos analisados por otimização: 1500;
- Semente para a geração de números aleatórios: 1.

Na Tabela 9, são apresentados os valores dos parâmetros obtidos pela otimização junto com os limites das variáveis de ajustes especificados.

Tabela 9: Valores dos parâmetros obtidos no projeto do circuito oscilador com três inversores em anel.

Parâmetro	X1 (μm)	X2 (μm)	X3 (μm)	X4 (fF)
Valor min.	1,0	0,35	1,0	10
Valor	11,04	0,39	7,48	19,67
Valor max.	100	10,0	100	500

Na Tabela 10, são apresentados os valores dos critérios de desempenho do melhor circuito encontrado na otimização realizada pelo algoritmo meta-heurístico. Esse circuito apresentou score de 0,109.

Tabela 10: Valores dos critérios de desempenho do circuito oscilador com três inversores em anel projetado. V_{max} e V_{min} são a máxima e mínima tensão em out.

Critério de desempenho	Frequência (MHz)	V_{max} (V)	V_{min} (V)	Potência (mW)	DHP (%)	THD (%)	Ruído de fase (dB/Hz)
Valor	900	1,95	-0,02	1,09	9,33	15,3	- 47,4

Na Tabela 11, são apresentados os resultados obtidos nas análises TRAN e HBOSC para efeito de comparação com os resultados obtidos pela análise SNOSC.

Tabela 11: Resultados das análises de transiente TRAN e balanço harmônico HBOSC. V_{max} e V_{min} são a máxima e mínima tensão em out.

Critério de desempenho	Frequência (MHz)	V_{max} (V)	V_{min} (V)	Potência (mW)	Ruído de fase (dB/Hz)
TRAN	888,7	1,95	-0,03	1,09	S/M
HBOSC	900,3	1,95	-0,01	1,09	- 47,3

Os comandos para realização da simulação TRAN e medidas estão apresentados a seguir:

```
.tran 0.01n 1.1m 1m 10p
.MEASURE TRAN vmax MAX V(out) FROM=1m TO=1.10m
.MEASURE TRAN vmin MIN V(out) FROM=1m TO=1.10m
.MEASURE TRAN delta PARAM='VMAX-VMIN'
.MEASURE TRAN period TRIG V(out) val='delta/2' RISE = 500 TARG V(out) val='delta/2'
RISE=501
.MEASURE TRAN freq PARAM='1/period'
.MEASURE TRAN lavg AVG I(V1) FROM=1.00m TO=1.10m
.MEASURE TRAN Pot PARAM='lavg*VDD'
```

O comando para realização da simulação HBOSC está apresentado a seguir:

```
.HBOSC TONE=900000000 NHARMS=100 PROBENODE=o,d,'VDD/2'
```

Na Figura 19, o sinal de saída do circuito obtido pela análise SNOSC é apresentado.

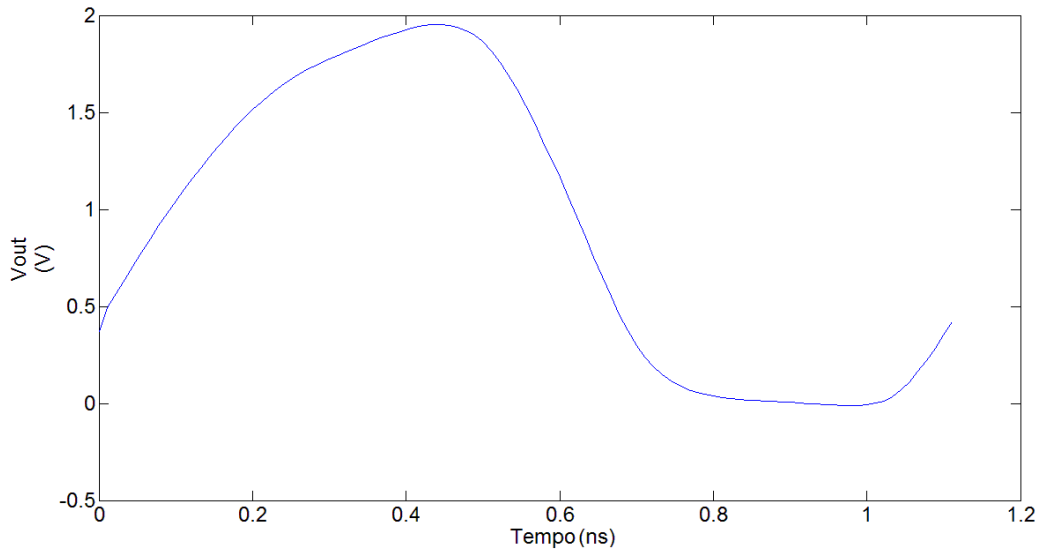


Figura 19: Sinal de saída do circuito oscilador com três inversores pelo tempo obtido pela análise SNOSC.

Observou-se que o circuito com oscilador três inversores projetado apresentou valores iguais ou muito próximos para os critérios de desempenho nas três análises de simulação (HBOSC, SNOSC e TRAN).

4.2 Circuito oscilador com cinco inversores em anel

Foi realizado o projeto de um circuito oscilador com cinco inversores em anel como ilustrado na Figura 14. As especificações do oscilador desejado estão dadas abaixo:

- Frequência: 433 MHz \pm 0,01%;
- Tensão de Saída: 1,0 V de amplitude;
- Tensão de Alimentação: 2,0 V;
- Potência Consumida Máxima: 1,0 mW;
- Distorção Harmônica Par: 10%;
- Distorção Harmônica Total: 50%;
- Ruído de Fase @4,33 kHz: -40 dB/Hz.

Os pesos aplicados na função objetivo estão indicados na Tabela 12.

Tabela 12: Pesos da função objetivo do circuito oscilador com cinco inversores em anel.

Critério de desempenho	Frequência	Tensão de Saída	Potência	Distorção par	Distorção Total	Ruído de Fase
Peso	10	100	0,1	0,1	1,0	0,1

Abaixo são apresentados detalhes sobre a otimização aplicada ao circuito:

- Método: PSO;
- Número de rodadas de otimização: uma;
- Número de otimizações por rodada: três;
- Máximo número de indivíduos analisados por otimização: 1500;
- Semente para a geração de números aleatórios: 1.

Na Tabela 13, são apresentados os valores dos parâmetros obtidos pela otimização junto com os limites das variáveis de ajustes especificados.

Tabela 13: Valores dos parâmetros obtidos no projeto do circuito oscilador com cinco inversores em anel.

Parâmetro	X1 (μm)	X2 (μm)	X3 (μm)	X4 (fF)
Valor min.	1,0	0,35	1,0	10
Valor	36,52	0,35	5,55	10,50
Valor Max.	100	40,0	100	600

Na Tabela 14, são apresentados os valores dos critérios de desempenho do melhor circuito encontrado na otimização realizada pelo algoritmo meta-heurístico. Esse circuito apresentou score de 0,189.

Tabela 14: Valores dos critérios de desempenho do circuito oscilador com cinco inversores em anel projetado. V_{max} e V_{min} são a máxima e mínima tensão em out.

Critério de desempenho	Frequência (MHz)	V_{max} (V)	V_{min} (V)	Potência (mW)	DHP (%)	THD (%)	Ruído de fase (dB/Hz)
Valor	433	2,11	-0,03	1,89	5,72	24,8	- 45,8

Na Tabela 15, são apresentados os resultados obtidos nas análises TRAN e HBOSC para efeito de comparação com os resultados obtidos pela análise SNOSC.

Tabela 15: Resultados das análises de transiente TRAN e balanço harmônico HBOSC. V_{max} e V_{min} são a máxima e mínima tensão em out.

Critério de desempenho	Frequência (MHz)	V_{max} (V)	V_{min} (V)	Potência (μW)	Ruído de fase (dB/Hz)
TRAN	425,7	2,11	-0,03	1,87	S/M
HBOSC	433,2	2,11	-0,03	1,89	-45,7 q

Os comandos para realização da simulação TRAN está apresentado a seguir, com os comandos de medida iguais aos já citados na seção anterior:

```
.tran 1n 1.2m 1m 1n UIC
```

O comando para realização da simulação HBOSC está apresentado a seguir:

```
.HBOSC TONE=433000000 NHARMS=100 PROBENODE=o,d,'VDD/2'
```

Na Figura 20, o sinal de saída do circuito obtido pela análise SNOSC é apresentado.

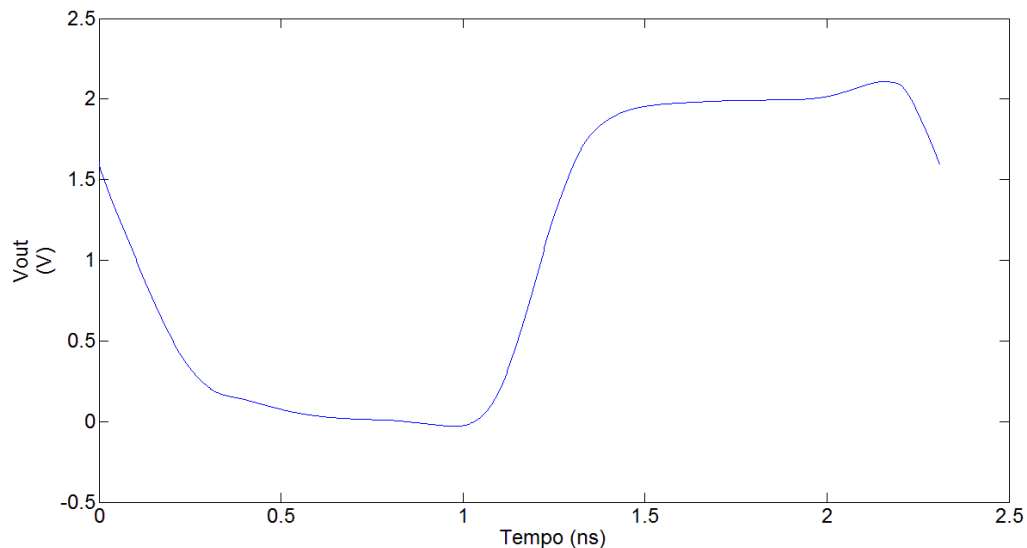


Figura 20: Sinal de saída do circuito oscilador com cinco inversores pelo tempo obtido pela análise SNOSC.

Observou-se que o circuito oscilador com cinco inversores projetado apresentou valores iguais ou muito próximos para os critérios de desempenho nas três análises de simulação (HBOSC, SNOSC e TRAN).

4.3 Circuito oscilador com cristal de 1,0 MHz

Foi realizado o projeto de um circuito oscilador com cristal de 1,0 MHz com topologia ilustrada na Figura 16. As especificações do oscilador desejado estão dadas abaixo:

- Frequência: 1,0 MHz \pm 0,1%;
- Tensão de Saída: 1,0 V de amplitude;
- Tensão de Alimentação: 3,0 V;
- Potência Consumida Máxima: 10 μ W;
- Distorção Harmônica Par: 10%;
- Distorção Harmônica Total: 50%;
- Ruído de Fase @10 kHz: -100 dB/Hz;
- Capacitor de carga (C_F): 50 fF.

Os pesos aplicados na função objetivo estão indicados na Tabela 16.

Tabela 16: Pesos da função objetivo do circuito oscilador com cristal de 1,0 MHz.

Critério de desempenho	Frequência	Tensão de Saída	Potência	Distorção par	Distorção Total	Ruído de Fase
Peso	10	100	0,1	0,1	1,0	0,1

Abaixo são apresentados detalhes sobre a otimização aplicada ao circuito:

- Método: PSO;
- Número de rodadas de otimização: uma;
- Número de otimizações por rodada: três;
- Máximo número de indivíduos analisados por otimização: 1500;
- Semente para a geração de números aleatórios: 1.

Na Tabela 17, são apresentados os valores dos parâmetros obtidos pela otimização junto com os limites das variáveis de ajustes especificados.

Tabela 17: Valores dos parâmetros obtidos no projeto do circuito oscilador com cristal de 1,0 MHz projetado.

Parâmetro	X1 (μm)	X2 (μm)	X3 (μm)	X4 (μm)
Valor min.	0,5	1,0	0,5	1,0
Valor	38,77	100	23,32	1,0
Valor max.	40	100	40	100
Parâmetro	X5 ($\text{k}\Omega$)	X6	X7	X8
Valor min.	10	-1,0	1,0	1,0
Valor	999,14	-0,30	6,94	2,85
Valor max.	1000	2,0	10	10

Na Tabela 18, são apresentados os valores dos critérios de desempenho do melhor circuito encontrado na otimização realizada pelo algoritmo meta-heurístico. Esse circuito apresentou score de 0,13.

Tabela 18: Valores dos critérios de desempenho do circuito oscilador com cristal de 1,0 MHz projetado. V_{max} e V_{min} são a máxima e mínima tensão em out.

Critério de desempenho	Frequência (MHz)	V_{max} (V)	V_{min} (V)	Potência (μW)	DHP (%)	THD (%)	Ruído de fase (dB/Hz)
Valor	1,0	1,35	0,33	13,03	6,88	7,11	- 107

Na Tabela 19, são apresentados os resultados obtidos nas análises TRAN e HBOSC para efeito de comparação com os resultados obtidos pela análise SNOSC.

Tabela 19: Resultado das análises de transiente TRAN e balanço harmônico HBOSC. V_{max} e V_{min} são a máxima e mínima tensão em out.

Critério de desempenho	Frequência (MHz)	V_{max} (V)	V_{min} (V)	Potência (μW)	Ruído de fase (dB/Hz)
TRAN	0,99	1,34	0,34	13,25	S/M
HBOSC	1,0	S/M	S/M	13,1	- 111,1

Os comandos para realização da simulação TRAN e medidas estão apresentados a seguir:

```
.tran 1n 110m 100m 1p
.MEASURE TRAN vmax MAX V(out) FROM=100m TO=110m
.MEASURE TRAN vmin MIN V(out) FROM=100m TO=110m
.MEASURE TRAN delta PARAM='VMAX-VMIN'
.MEASURE TRAN period TRIG V(out) val='delta/2' RISE = 500 TARG V(out) val='delta/2'
RISE=501
.MEASURE TRAN freq PARAM='1/period'
```



```
.MEASURE TRAN lavg AVG I(V1) FROM=100m TO=110m  
.MEASURE TRAN Pot PARAM='lavg*VDD'
```

O comando para realização da simulação HBOSC está apresentado a seguir:

```
.HBOSC TONE=1000000 NHARMS=100 PROBENODE=o,d,'VDD/2'
```

Na Figura 21, o sinal de saída do circuito obtido pela análise SNOSC é apresentado.

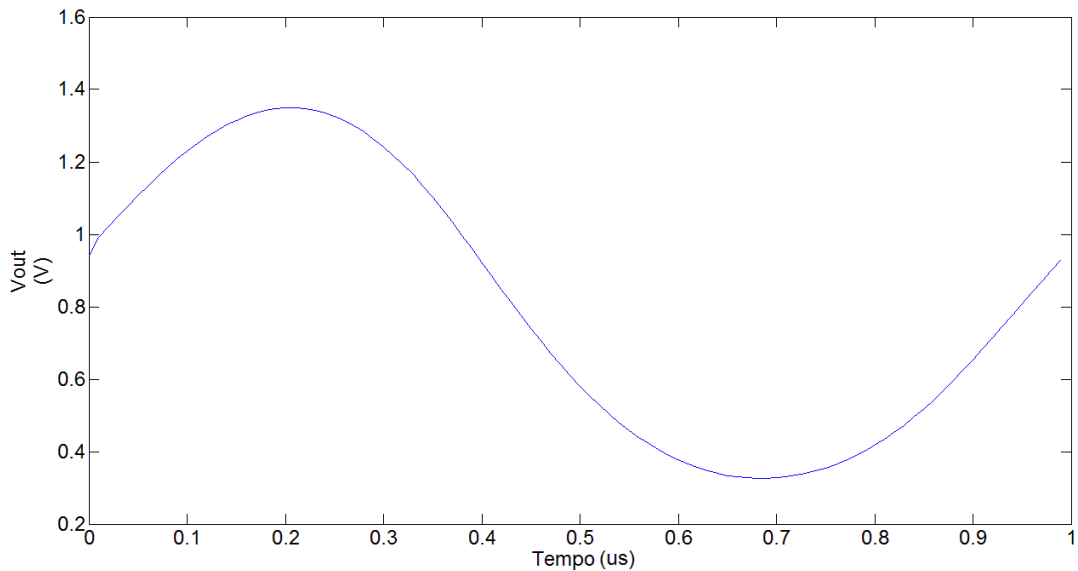


Figura 21: Sinal de saída do circuito com cristal de 1,0 MHz pelo tempo obtido pela análise SNOSC.

Observou-se que o circuito com oscilador a cristal de 1,0 MHz projetado apresentou valores iguais ou muito próximos para os critérios de desempenho nas três análises de simulação (HBOSC, SNOSC e TRAN). Porém, a simulação HBOSC, apesar de retornar um valor de frequência de oscilação igual as outras simulações, não retornou a amplitude de oscilação.

4.4 Circuito oscilador com cristal de relógio (32768 Hz)

Foi realizado o projeto de um circuito oscilador com cristal de 32768 Hz com topologia ilustrada na Figura 17. As especificações do oscilador desejado estão dadas abaixo:

- Frequência: 32768 Hz \pm 0,1%;
- Tensão de Saída: 1,0 V de amplitude;
- Tensão de Alimentação: 3,0 V;
- Potência Consumida Máxima: 10 μ W;
- Distorção Harmônica Par: 10%;
- Distorção Harmônica Total: 50%;

- Ruído de Fase @327,68 Hz: -80 dB/Hz
- Capacitor de carga (C_F): 50 fF.

Os pesos aplicados na função objetivo estão indicados na Tabela 20.

Tabela 20: Pesos da função objetivo do circuito oscilador com cristal de 32768 Hz.

Critério de desempenho	Frequência	Tensão de Saída	Potência	Distorção par	Distorção Total	Ruído de Fase
Peso	10	100	0,1	0,1	1,0	0,1

Abaixo são apresentados detalhes sobre a otimização aplicada ao circuito:

- Método: PSO;
- Número de rodadas de otimização: três;
- Número de otimizações por rodada: duas;
- Máximo número de indivíduos analisados por otimização: 1500;
- Semente para a geração de números aleatórios: 1; 10; 100.

Na Tabela 21, são apresentados os valores dos parâmetros obtidos pela otimização junto com os limites das variáveis de ajustes especificados.

Tabela 21: Valores dos parâmetros obtidos no projeto do circuito oscilador com cristal de 32768 Hz projetado. V_{max} e V_{min} são a máxima e mínima tensão em out.

Parâmetro	X1 (μm)	X2 (μm)	X3 (μm)	X4 (μm)
Rodada				
Valor min.	0,5	1,0	0,5	1,0
A	9,32	89,68	31,09	56,42
B	2,22	24,19	39,99	97,67
C	5,43	16,53	35,74	62,83
Valor max.	40	100	40	100
Parâmetro	X5 ($\text{k}\Omega$)	X6	X7	X8
Rodada				
Valor min.	10	-2,0	1,0	1,0
A	965,10	-0,51	3,25	8,94
B	619,54	-0,93	6,70	9,97
C	976,02	-1,29	9,85	10,0
Valor max.	1000	1,0	10	10

Na Tabela 22, são apresentados os valores dos critérios de desempenho do melhor circuito encontrado na otimização realizada pelo algoritmo meta-heurístico.

Tabela 22: Valores dos critérios de desempenho do circuito oscilador com cristal de 32768 Hz projetado. V_{max} e V_{min} são a máxima e mínima tensão em out.

Critério de desempenho	Score	Frequência (Hz)	V_{max} (V)	V_{min} (V)	Potência (μ W)	DHP (%)	THD (%)	Ruído de fase (dB/Hz)
Rodada								
A	0,112	32787	2,81	0,07	11,2	9,15	11,4	- 86,2
B	0,113	32814	1,28	0,09	5,82	8,11	8,98	- 59,1
C	0,05	32762	2,11	0,05	2,95	9,66	11,4	- 68,1

4.4.1 Refinamento da solução por Busca Padrão

A melhor solução encontrada na otimização do circuito oscilador com cristal de 32768 Hz – circuito encontrado pela rodada C na otimização por Enxame de Partículas - foi refinada através do algoritmo meta-heurístico Busca Padrão. A especificação, os limites dos parâmetros e os pesos da função objetivo foram mantidos como na Tabela 20 e na Tabela 21: Valores dos parâmetros obtidos no projeto do circuito oscilador com cristal de 32768 Hz projetado. V_{max} e V_{min} são a máxima e mínima tensão em out. **Erro! Fonte de referência não encontrada..** Após 597 indivíduos analisados o método convergiu para o circuito com os parâmetros descritos na Tabela 23 e com o desempenho descrito na Tabela 24. O circuito apresentou score de 0,157.

Tabela 23: Valores dos parâmetros obtidos pelo refinamento do circuito oscilador com cristal de 32768 Hz por Busca Padrão.

Parâmetro	X1 (μm)	X2 (μm)	X3 (μm)	X4 (μm)
Valor	4,98	15,41	35,29	70,02
Parâmetro	X5 ($k\Omega$)	X6	X7	X8
Valor	980,47	-1,30	9,75	10,0

Tabela 24: Valores dos critérios de desempenho do circuito refinado por Busca Padrão. V_{max} e V_{min} são a máxima e mínima tensão em out.

Critério de Desempenho	Frequência (Hz)	V_{max} (V)	V_{min} (V)	Potência (μ W)	DHP (%)	THD (%)	Ruído de fase (dB/Hz)
Valor	32793	2,7	0,04	2,93	11,2	13,4	- 81,1

4.4.2 Refinamento da solução por Arrefecimento Simulado

A melhor solução encontrada na otimização do circuito oscilador com cristal de 32768 Hz – circuito encontrado pela rodada C na otimização por Enxame de Partículas - foi refinada também através do algoritmo método meta-heurístico Arrefecimento Simulado. A especificação, os limites dos parâmetros e os pesos da função objetivo foram mantidos como na Tabela 20 e na Tabela 21. Após 1500 indivíduos analisados o método convergiu para o circuito com os parâmetros descritos na Tabela 25 e com o desempenho descrito na Tabela 26. O circuito apresentou score de 0,111.

Tabela 25: Valores dos parâmetros obtidos pelo refinamento do circuito oscilador com cristal de 32768 Hz por Arrefecimento Simulado. V_{max} e V_{min} são a máxima e mínima tensão em out.

Parâmetro	X1 (μm)	X2 (μm)	X3 (μm)	X4 (μm)
Valor	8,70	44,75	40,0	70,84
Parâmetro	X5 ($\text{k}\Omega$)	X6	X7	X8
Valor	963,24	-0,94	8,12	7,80

Tabela 26: Valores dos critérios de desempenho do circuito refinado por Arrefecimento Simulado. V_{max} e V_{min} são a máxima e mínima tensão em out.

Critério de desempenho	Frequência (Hz)	V_{max} (V)	V_{min} (V)	Potência (μW)	DHP (%)	THD (%)	Ruído de fase (dB/Hz)
Valor	32587	2,95	0,05	5,55	9,59	12,9	- 85,7

4.4.2 Análise do circuito final

O circuito encontrado através dos refinamentos pelos algoritmos de Arrefecimento Simulado e Busca Padrão não obtiveram melhoras em relação ao score do circuito. O circuito projetado final foi, portanto, o circuito da rodada C da otimização por Enxame de Partículas apresentado na Tabela 25. Na Tabela 27, são apresentados os resultados obtidos nas análises TRAN e HBOSC para efeito de comparação com os resultados obtidos pela análise SNOSE no caso deste circuito.

Tabela 27: Resultado das análises de transiente TRAN e balanço harmônico HBOSC da rodada C. Vmax e Vmin são a máxima e mínima tensão em out.

Critério de Desempenho	Frequência (Hz)	Vmax (V)	Vmin (V)	Potência (μW)	Ruído de fase (dB/Hz)
TRAN	30857	1,92	0,06	2,96	S/M
HBOSC	32768	0,52	0,52	2,96	S/M

Os comandos para realização da simulação TRAN e medidas estão apresentados a seguir:

```
.tran 1p 1200m 1000m 1p
.MEASURE TRAN vmax MAX V(out) FROM=1100m TO=1200m
.MEASURE TRAN vmin MIN V(out) FROM=1100m TO=1200m
.MEASURE TRAN delta PARAM='VMAX-VMIN'
.MEASURE TRAN period TRIG V(out) val='delta/2' RISE = 500 TARG V(out) val='delta/2'
RISE=501
.MEASURE TRAN freq PARAM='100/period'
.MEASURE TRAN Ton TRIG V(out) val='delta/2' RISE = 500 TARG V(out) val='delta/2' FALL=501
.MEASURE TRAN lavg AVG I(V1) FROM=1000m TO=1200m
.MEASURE TRAN Pot PARAM='lavg*VDD'
```

O comando para realização da simulação HBOSC está apresentado a seguir:

```
.HBOSC TONE=32768 NHARMS=100 PROBENODE=o,d,'VDD/2'
```

Na Figura 22, o sinal de saída obtido pela análise SNOOSC é apresentado.

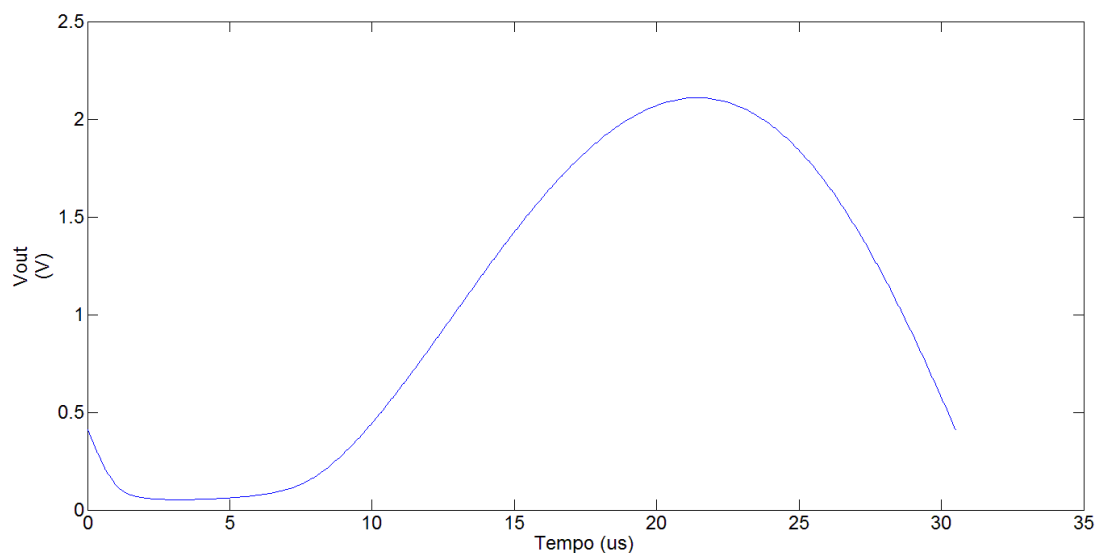


Figura 22: Sinal de saída do circuito com cristal de 32768 Hz pelo tempo obtido pela análise SNOOSC.

Observou-se que o circuito com cristal de 32768 Hz projetado apresentou valores semelhantes para a potência consumida nas três análises de simulação (HBOSC, SNOOSC e TRAN). A análise HBOSC não convergiu, e portanto, seus resultados não podem ser considerados. A análise de transiente apresentou a oscilação do sinal de

saída em uma frequência e amplitude diferentes da apresentada pela análise SNOSC. Porém, a análise de transiente demonstrou a existência da oscilação.

Nota-se que os refinamentos da solução encontrados pela otimização por Enxame de Partículas e pelo Arrefecimento Simulado tiveram uma piora em seu *score*. O aumento apresentado nos valores de *score* se deve: a um aumento do DHP no caso do refinamento por Busca Padrão; e a um afastamento na frequência de oscilação em relação a frequência desejada no caso do refinamento por Arrefecimento Simulado.

No primeiro caso, o circuito teve melhoras significativas em relação a outros critérios de desempenho, porém devido as especificações desejadas e aos pesos na função objetivo o circuito apresentou um *score* maior. Isso reflete a importância na escolha dos pesos da função objetivo para que o *score* reflita principalmente os principais critérios de desempenho desejados.

4.5 Circuito oscilador controlado por tensão

Foi realizado o projeto de um circuito oscilador controlado por tensão com topologia ilustrada na Figura 18. Nesse circuito, o sinal de saída foi considerado como a tensão elétrica entre os nós o e d , logo a tensão máxima de excursão, a tensão mínima de excursão, o THD e o DHP são relativos a tensão $V(o,d)$. As especificações do oscilador desejado estão dadas abaixo:

- Pontos de frequência: 800 MHz; 900 MHz; 1,0 GHz; 1,1 GHz;
- Tensões de Controle: 1,4 V; 1,5 V; 1,6 V; 1,7 V;
- Tolerância para a Frequência: $\pm 0,1\%$;
- Tensão de Saída: 1,0 V de amplitude;
- Tensão de Alimentação: 3,0 V;
- Potência Consumida Máxima: 500 μ W;
- Distorção Harmônica Par: 5%;
- Distorção Harmônica Total: 5%;
- Número de harmônicas: 10;
- Ruído de Fase @ 1 MHz: -100 dB/Hz;
- *Jitter* de 10 MHz até 20 MHz: 0,001 s.

Os pesos aplicados na função objetivo estão indicados na Tabela 28.

Tabela 28: Pesos da função objetivo do circuito oscilador controlado por tensão.

Critério de Desempenho	Frequência	Tensão de Saída	Potência	DHP	THD	Jitter	Ruído de Fase
Peso	10	100	0,1	0,1	1,0	0,1	0,1

Abaixo são apresentados detalhes sobre a otimização aplicada ao circuito:

- Método: PSO;
- Número de rodadas de otimizações: uma;
- Número de otimizações por rodada: três;
- Máximo número de indivíduos analisados por otimização: 1500;
- Semente para a geração de números aleatórios: 1.

Na Tabela 29, são apresentados os valores dos parâmetros obtidos pela otimização junto com os limites das variáveis de ajustes especificados.

Tabela 29: Valores dos parâmetros obtidos no projeto do circuito oscilador controlado por tensão projetado.

Parâmetro	X1 (µm)	X2 (µm)	X3 (µm)	X4 (µm)
Valor min.	0,5	1,0	0,5	1,0
Valor	4,57	8,48	0,51	15,59
Valor max.	50	200	50	100
Parâmetro	X5 (µm)	X6 (µm)	X7 (nH)	X8
Valor min.	10	1,0	5,0	1,0
Valor	22,76	20,54	12,06	4,91
Valor max.	40	100	30	5,0

Na Tabela 30, são apresentados os valores dos critérios de desempenho do melhor circuito encontrado na otimização realizada pelo algoritmo meta-heurístico. O score do circuito obtido foi de 0,8. É importante ressaltar que a potência consumida, o THD e o DHP foram obtidos apenas para a tensão de controle mais alta apenas.

Tabela 30: Valores dos critérios de desempenho do circuito oscilador controlado por tensão projetado. V_{max} e V_{min} são a máxima e mínima diferença de tensão entre o e d.

Critério de Desempenho	Frequência (GHz)	V_{max} (V)	V_{min} (V)	Potência (μ W)	DHP (%)	THD (%)	Jitter (ms)	Ruído de fase (dB/Hz)
Vc=1,4 V	1,10	2,15	-2,15	S/M	S/M	S/M	0,24	-122
Vc=1,5 V	1,00	1,78	-1,78	S/M	S/M	S/M	0,69	-113
Vc=1,6 V	0,89	1,41	-1,41	S/M	S/M	S/M	0,73	-113
Vc=1,7 V	0,80	1,05	-1,05	566	0	4,89	0,61	-114

Na Tabela 31, são apresentados os resultados obtidos na análise TRAN, e na Tabela 32, são apresentados os resultados obtidos na análise SNOSC para efeito de comparação com os resultados obtidos pela análise HBOSC, presentes na Tabela 30.

Tabela 31: Resultado das análises de transiente TRAN. V_{max} e V_{min} são a máxima e mínima diferença de tensão entre o e d.

Critério de Desempenho	Frequência (GHz)	V_{max} (V)	V_{min} (V)	Potência (μ W)
Vc=1,4 V	1,06	1,81	-1,81	666
Vc=1,5 V	0,96	1,51	-1,52	706
Vc=1,6 V	0,89	1,14	-1,14	628
Vc=1,7 V	0,75	0,98	-0,98	585

Tabela 32: Resultado da análise SNOSC. V_{max} e V_{min} são a máxima e mínima diferença de tensão entre o e d.

Critério de Desempenho	Frequência (GHz)	V_{max} (V)	V_{min} (V)	Potência (μ W)	Jitter (ms)	Ruído de fase (dB/Hz)
Vc=1,4 V	1,09	2,06	-2,06	S/M	0,26	-121
Vc=1,5 V	1,00	1,76	-1,76	S/M	0,49	-116
Vc=1,6 V	0,90	1,42	-1,42	S/M	0,66	-113
Vc=1,7 V	0,80	1,06	-1,06	574	0,70	-112

Os comandos para realização da simulação TRAN e medidas estão apresentados a seguir:

```
.tran 10p 400n 100n 10p sweep VC POI 4 1.4 1.5 1.6 1.7
.MEASURE TRAN vmax MAX V(o,d) FROM=100n TO=400n
.MEASURE TRAN vmin MIN V(o,d) FROM=100n TO=400n
.MEASURE TRAN delta PARAM='VMAX-VMIN'
.MEASURE TRAN period TRIG V(o,d) val='VMAX/2' RISE = 50 TARG V(o,d) val='VMAX/2'
RISE=51
.MEASURE TRAN freq PARAM='1/period'
.MEASURE TRAN Ton TRIG V(o,d) val='VMAX/2' RISE = 50 TARG V(o,d) val='VMAX/2'
FALL=51
```



```
.MEASURE TRAN Iavg AVG I(V1) FROM=100n TO=400n  
.MEASURE TRAN Pot PARAM='Iavg*VDD'
```

O comando para realização da simulação SNOSC está apresentado a seguir:

```
.SNOSC TRES=1e-13 PERIOD=1.1e-9 OSCNODE=d TRINIT=2.5e-7 sweep VC POI 4 1.40 1.50  
1.60 1.70
```

Na Figura 23, o sinal de saída do circuito obtido pela análise HBOSC é apresentado.

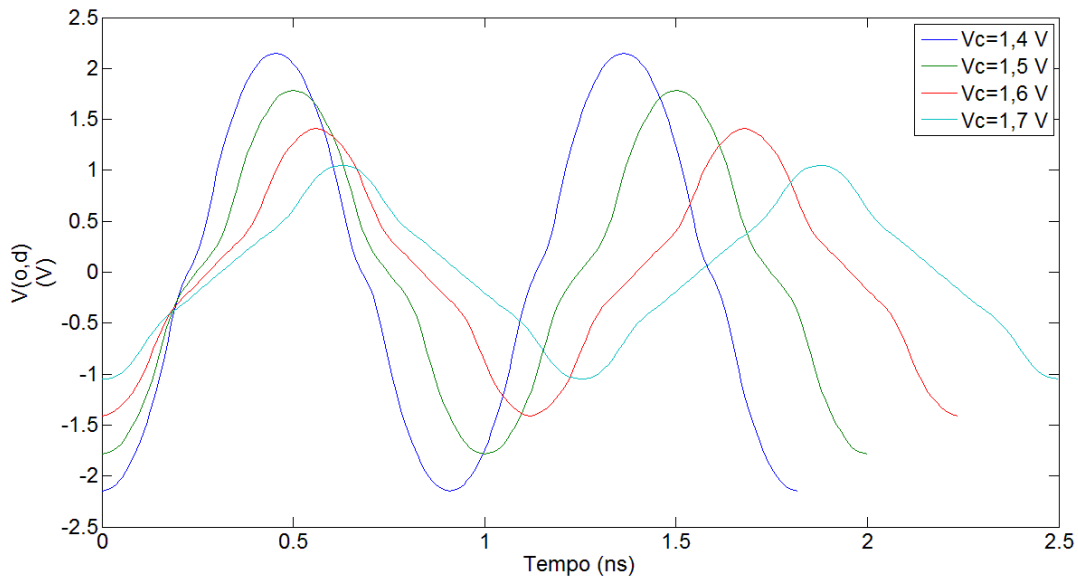


Figura 23: Sinal de saída do circuito VCO pelo tempo obtido pela análise HBOSC.

Observou-se que o circuito VCO projetado apresentou valores muito próximos para os critérios de desempenho obtidos pelas análises de simulação HBOSC e SNOSC. A análise de transiente apresentou valores condizentes com os resultados, porém com diferenças significativas na potência consumida e na amplitude de oscilação.

4.6 Discussão

As funções avaliadoras citadas em capítulos anteriores sofreram mudanças no decorrer do trabalho. As principais alterações ocorreram na nas funções de avaliação da excursão do sinal de saída e da frequência, no projeto de circuitos VCO's. Essas funções tiveram de ser alteradas para que circuitos com amplitude de oscilação baixas, ou com faixa de frequência baixa, fossem avaliados como não funcionais – com um alto valor de *score*. As antigas funções de avaliação (que mediam a diferença relativa do valor medido com o valor desejado) para esses dois parâmetros retornavam valores

baixos para desempenhos não satisfatórios - oscilações de milivolts, ou variações de frequência de poucos hertz.

Para se ter uma ideia do tempo necessário para se realizar as otimizações, pode-se considerar o circuito apresentado na Figura 13, otimizado em um computador com as características básicas apresentadas na Tabela 38 com o algoritmo otimização por Enxame de Partículas, e com as especificações apresentadas na Seção 4.1. Esta otimização dura, em média, seis horas. Da mesma maneira, a otimização do circuito da Figura 16 realizada na Seção 4.3 dura em média dezoito horas.

Tabela 33: Informações básicas do computador utilizado na execução dos algoritmos meta-heurísticos.

Processador	Intel® Core™ i7 3610QM 2,3GHz
Memória RAM	DR3 1600 MHz SDRAM, 2 x 4 Gb
Disco rígido	5400rpm 750Gb

Os circuitos obtidos para as topologias de circuitos osciladores com inversores em anel alcançaram as principais especificações desejadas, com exceção da potência consumida que a otimização não conseguiu reduzir a menos de 1,0 mW como o desejado.

O circuito oscilador com cristal de 1,0 MHz encontrado alcançou as principais especificações desejadas e a potência consumida do circuito foi de 13,03 μ W, valor próximo de 10 μ W que era o objetivo. A potência baixa apresentada mostra a eficiência dos circuitos apresentados por Thommen (1999, p.146-149) para circuitos osciladores de baixa potência. Porém, o estágio amplificador *push-pull* apresenta instabilidade elevada na tensão de saída e isso dificultou a convergência da otimização.

O circuito oscilador a cristal de 32768 Hz encontrado apresentou potência consumida de 2,95 μ W, cumprindo o objetivo desse trabalho de projetar um circuito oscilador para relógios com potência consumida abaixo de 10 μ W. O circuito apresentou essa potência com alimentação de 3,0 V e amplitude de oscilação de rail-to-rail. O refinamento da solução encontrado pelo método meta-heurístico Busca Padrão apresentou uma melhora em determinados critérios de desempenho tais como potência consumida e ruído de fase, porém obteve uma piora no DHP, ocasionando o descarte do circuito.

O refinamento do projeto do oscilador de 32768 Hz encontrado pelo método de Arrefecimento Simulado apresentou consumo de potência maior que o alcançado pelo projeto inicial, porém o ruído de fase foi melhorado. Isso ocorreu, pois os pesos da função objetivo valorizavam melhoras em potência e em ruído de fase igualmente. No

refinamento o método meta-heurístico percorreu o caminho da otimização do ruído de fase ao invés da potência. Esse fato reforça a importância da escolha dos pesos da função objetivo conforme a necessidade.

O circuito apresentado em Thommen (1999, pg.146-1499), em que se baseiam as topologias de cristal utilizadas nesse trabalho, apresentou alimentação de 1,2 V e 72nA de corrente consumida para a frequência de 32768 Hz. Porém, nesse artigo foram aplicadas técnicas de controle da corrente de polarização para obter sinais de baixa amplitude na saída. Esse circuito, também, utiliza estágios adicionais para amplificar a excursão do sinal de saída.

5 Conclusão

Neste trabalho, foi apresentado o desenvolvimento de uma plataforma de projeto de circuitos osciladores através de algoritmos meta-heurísticos. Os conhecimentos básicos sobre transistores MOS, circuitos osciladores e sobre os algoritmos meta-heurísticos utilizados foram apresentados, assim como todas as ferramentas utilizadas. O funcionamento da plataforma desenvolvida foi descrito, assim como o funcionamento das topologias de circuitos projetados. Diversos projetos foram realizados com sucesso pela plataforma de projeto. No Capítulo 4, alguns resultados foram apresentados. Uma breve discussão dos resultados e dos problemas existentes no desenvolvimento deste trabalho também foi dada.

A otimização de circuitos osciladores apresentou algumas dificuldades de convergência devido tanto aos algoritmos de otimização quanto, principalmente, aos diversos tipos de análises de simulação empregados. Essas dificuldades foram o grande desafio neste trabalho, exigindo modificações substanciais na plataforma no decorrer do desenvolvimento. As dificuldades foram apresentadas no Capítulo 4.

Em relação às funções de avaliação desenvolvidas, as soluções discutidas no Capítulo 3 levaram tempo para seu desenvolvimento, dado que muitos testes foram realizados para encontrar aquelas que melhores resultados apresentassem. Também os valores dos pesos foram cuidadosamente analisados. Esses fatos mostraram que a escolha das funções de avaliação e dos respectivos pesos é de extrema importância para a convergência dos métodos meta-heurísticos.

Os circuitos projetados pela ferramenta atingiram as especificações desejadas comprovando o sucesso do método meta-heurístico no dimensionamento de circuitos osciladores MOS, porém foram apresentadas discrepâncias entre os resultados das análises HBOSC, SNOSC e TRAN. Essas discrepâncias devem ser evitadas e para isso um estudo maior sobre as análises deve ser realizado. Caso necessário, mudanças nos critérios de desempenho e nas funções de avaliação devem ocorrer. Na maioria dos casos, porém, pelo menos duas das três análises apresentaram resultados semelhantes.

Os resultados obtidos das otimizações dos circuitos osciladores foram satisfatórios. As otimizações obtiveram circuitos com baixo consumo de potência nos circuitos oscilador a cristal e uma faixa de sintonia adequada para o circuito oscilador VCO com faixa central de 900 MHz. Quanto aos circuitos osciladores em anel, as principais especificações desejadas foram alcançadas, porém o consumo de potência foi elevado devido às características dessa topologia e à frequência de funcionamento.

Apesar dos resultados alcançados neste trabalho, ele não foi suficiente para concluir a implementação e validação de uma plataforma de projeto automático de circuitos osciladores genéricos por algoritmos meta-heurísticos, sendo ainda necessária a ação humana durante o projeto. Para tal, o estudo aprofundado das análises utilizadas nas simulações e sua validação são necessários, principalmente para osciladores com cristais de alto fator de qualidade.

Adicionalmente, testes mais amplos devem ser ainda aplicados ao projeto osciladores controlados por tensão (VCO), principalmente utilizando modelos mais completos e complexos para os *varactors* e indutores integrados. Ficou claro, também, que pode haver melhorias na forma com que as especificações de tensão de controle e pontos de frequência de referência são dadas. Essa questão interfere na linearidade do circuito e no espaço de busca do problema.

Conclui-se que os métodos escolhidos para o desenvolvimento de nossa plataforma se mostraram adequados para os objetivos da mesma. Os métodos meta-heurísticos se mostraram ferramentas eficazes e confiáveis no dimensionamento de circuitos eletrônicos osciladores. Nos piores casos, a melhor solução apresentada estava muito próxima da especificação. Apenas critérios de desempenho secundários não foram atingidos, conforme visto no projeto do circuito oscilador com cristal de relógio que não atingiu o ruído de fase mínimo requerido.

Os trabalhos futuros que devem ser realizados são:

- Acrescentar funções que extraíam modelos elétricos de indutores reais caracterizados por suas dimensões físicas.
- Acrescentar novas topologias de circuitos osciladores controlados por tensão;
- Acrescentar topologias com inicialização e controle de amplitude;
- Realizar o *Layout*, a integração e testes físicos de circuito osciladores projetados com a plataforma.

Referências

- AUSTRIAMICROSYSTEM. **0.35 μm CMOS C35 Design Rules**. Austria, 2003a.
- AUSTRIAMICROSYSTEM. **0.35 μm CMOS C35 Process Parameters**. Austria, 2003b.
- BAKER, R. Jacob. **CMOS Circuit Design, Layout, and Simulation**. Revised Second Edition, 2008.
- BOSTIAN, Charles W.; KRAUSS, Herbert L.; RAAB, Frederick H. **Solid State Radio Engineering**. USA: John Wiley & Sons, 1980.
- BUNCH, Ryan Lee. **Projeto de fontes de tensão de referência através de metaheurísticas**. 2001. Dissertação (Mestrado) – Faculty of the Virginia Polytechnic Institute and State University, São Carlos, 2013.
- BUNCH, Ryan Lee; RAMAN, Sanjay. Large-signal analysis of MOS varactors in CMOS-G m LC VCOs. **Solid-State Circuits, IEEE Journal of**, v. 38, n. 8, p. 1325-1332, 2003.
- FRANCO, Sergio. **Design with Operational amplifiers and analog integrated circuits**, Third Edition, 2001.
- EBERHART, R.C.; KENNEDY, J. **Particle swarm optimization** in *Proc. IEEE Int. Conf. Neural Networks*, Perth, Australia, Nov. 1995, pp. 1942–1948.
- EBRAHIMI, Emad; NASEH, Sasan. Investigating the performance of cross-coupled CMOS LC-VCOs using genetic algorithm. In: **Radioelektronika (RADIOELEKTRONIKA), 2011 21st International Conference**. IEEE, 2011. p. 1-4.
- FILGUEIRAS, I.F.R.S. **Otimização de circuitos CMOS por Algoritmo Genético**. São Carlos, 2010.
- HAJIMIRI, Ali; LIMOTYRAKIS, Sotirios; LEE, Thomas H. Jitter and phase noise in ring oscillators. **Solid-State Circuits, IEEE Journal of**, v. 34, n. 6, p. 790-804, 1999.
- HOOKE, Robert; JEEVES, T. A. “Direct Search” Solution of Numerical and Statistical Problems. **Journal of the ACM (JACM)**, v. 8, n. 2, p. 212-229, 1961.
- HOOKE, R.; JEEVES, T. A. Citation Classic-Direct Search Solution of Numerical and Statistical Problems. **CURRENT CONTENTS/ENGINEERING TECHNOLOGY & APPLIED SCIENCES**, n. 13, p. 14-14, 1980.
- KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. **Optimization by Simulated Annealing**. Science, New York, v. 220, n. 4598, p. 671-680, May 1983.

LEE, Thomas H. **The Design of CMOS Radio-Frequency Integrated Circuits**. Cambridge, United Kingdom: Cambridge University Press, 1998.

LUKE, Sean. **Essentials of Metaheuristics**, Second Edition, Online Version, 2013.

MATHWORKS. **MATLAB Language**, 2014a. Disponível em:
<http://www.mathworks.com/help/matlab/learn_matlab/product-description.html/>.
Acesso em: 13 de Jun 2014.

MATHWORKS. **Global Optimization Toolbox**, 2014b. Disponível em:
<<http://www.mathworks.com/help/optim/ug/product-description.html/>>. Acesso em: 13 de Jun 2014.

MANCINI, Ron. **Op Amps for Everyone – Design Guide**. Texas Instruments, 2002.

MELLIAR-SMITH, C.M. et al. *The transistor: an invention becomes a big business*. **Proceedings of the IEEE**, v.86, p.86-110, Jan. 1998.

NGOYA, R.; SUAREZ A.; SOMMET R.; QUERE R. “Steady State Analysis of Free or Forced Oscillators by Harmonic Balance and Stability Investigation of Periodic and Quasi-Periodic Regimes”. **International Journal of Microwave and Millimeter-Wave Computer-Aided Engineering**, Volume 5, Number 3, p. 210-223, 1995.

PEREIRA, Pedro et al. Metaheuristic algorithms comparison for the LC-Voltage controlled oscillators optimal design. In: **Modeling, Simulation and Applied Optimization (ICMSAO)**, 2013 5th International Conference on. IEEE, 2013. p. 1-6.

RAZAVI, Behzad. **Design of Analog CMOS Integrated Circuits**. First Edition, McGraw-Hill, 2001.

RAZAVI, Behzad. **RF Microelectronics**, 2nd Ed., Prentice Hall, 2012.

SANSENS, Willy M. **Analog Design Essentials**, First Edition, Springer, 2006.

SASSI, M. M. F. S. **Projeto de fontes de tensão de referência através de metaheurísticas**. 2013. Dissertação (Mestrado) - Escola de Engenharia de São Carlos, Universidade de São Carlos, São Carlos, 2013.

SEDRA, Adel S.; SMITH, Kenneth C. **Microeletronic Circuits**, 5th Ed., Oxford University Press, 2004.

SEMICONDUCTOR INDUSTRY ASSOCIATION (SIA). **The national technology roadmap for semiconductors**. San Jose, 2013 Disponível em <http://www.itrs.net/reports.html>.

SYNOPTSYS. **HSpice**, 2014. Disponível em: <
<http://www.synopsys.com/Tools/Verification/AMSVerification/CircuitSimulation/HSPICE/Pages/default.aspx>>. Acesso em 13 de Jun 2014.

SYNOPTSYS. **HSPICE User Guide: RF Analysis**, Version A-2008.3, 2008a.

SYNOPTSYS. **HSPICE Reference Manual: Commands and Control Options**, Version A-2008.3, 2008b.

SYNOPTSYS. **HSPICE Reference Manual: Elements and Device Models**, Version A-2008.3, 2008c

SYNOPTSYS. **HSPICE Reference Manual: MOSFET Models**, Version A-2008.3, 2008d

WANLASS, F. M. **Low Standby-Power Complementary Field Effect Transistor**. US Patent 3.356.858, 1967.

Apêndice A

Descrição SPICE do circuito oscilador com três inversores em anel:

```
*Circuit Oscillator Ring - 3
* Basic Circuit
* test version

.include param

*Circuit Description

M1 d out vd vd MODP W='X1*1u' L='X2*1u'
M2 o d vd vd MODP W='X1*1u' L='X2*1u'
M3 out o vd vd MODP W='X1*1u' L='X2*1u'

M4 d out 0 0 MODN W='X3*1u' L='X2*1u'
M5 o d 0 0 MODN W='X3*1u' L='X2*1u'
M6 out o 0 0 MODN W='X3*1u' L='X2*1u'

C1 d 0 C='X4*1f'
C2 o 0 C='X4*1f'
C3 out 0 C='X4*1f'

*Sources

V1 vd 0 DC 'VDD'

* Initial Condition

.IC out=0

.end
```

Descrição SPICE do circuito oscilador com cinco inversores em anel:

```
*Circuit Oscillator Ring - 5
* Basic Circuit
* Final version

.include param

*Circuit Description

M1 o out vd vd MODP W='X1*1u' L='X2*1u'
M2 d o vd vd MODP W='X1*1u' L='X2*1u'
M3 o3 d vd vd MODP W='X1*1u' L='X2*1u'
M4 o4 o3 vd vd MODP W='X1*1u' L='X2*1u'
M5 out o4 vd vd MODP W='X1*1u' L='X2*1u'

M6 o out 0 0 MODN W='X3*1u' L='X2*1u'
M7 d o 0 0 MODN W='X3*1u' L='X2*1u'
M8 o3 d 0 0 MODN W='X3*1u' L='X2*1u'
M9 o4 o3 0 0 MODN W='X3*1u' L='X2*1u'
M10 out o4 0 0 MODN W='X3*1u' L='X2*1u'
```

```

C1 o 0 C='X4*1f'
C2 d 0 C='X4*1f'
C3 o3 0 C='X4*1f'
C4 o4 0 C='X4*1f'
C5 out 0 C='X4*1f'

```

```
*Sources
```

```
V1 vd 0 DC 'VDD'
```

```
* Initial Condition
```

```
.IC out=0
```

```
.end
```

Descrição SPICE do circuito oscilador com cristal de 1MHz:

```
*Circuit Oscillator with Cristal 1 MHz - 1
```

```
* Basic Circuit
```

```
* Final Version
```

```
.include param
```

```
*Circuit Description
```

```
** PMOS
```

```
M1 vd b d vd MODP W='X2*1u' L='X1*1u' M='int(X7)'
```

```
M2 vd b b vd MODP W='X2*1u' L='X1*1u'
```

```
M3 vd b out vd MODP W='X2*1u' L='X1*1u' M='int(X8)'
```

```
** NMOS
```

```
MN1 d o 0 0 MODN W='X4*1u' L='X3*1u'
```

```
MN2 out o 0 0 MODN W='X4*1u' L='X3*1u' M='int(X8)/int(X7)'
```

```
** Resistor
```

```
R1 o d 'X5*1k'
```

```
* Capacitor
```

```
C1 o 0 3p
```

```
C2 d 0 3p
```

```
C3 b o 10p
```

```
Cf out 0 'Cl'
```

```
* Subcircuit
```

```
X1 o d crystal
```

```
** Crystal Model
```

```
** 1 MHz
```

```
.param Cp=0.017p Cs=3.5p Ls=15m Rs= 20
```

```
.subckt crystal NC1 NC2
```

```
Cp1 NC1 NC2 'Cp'  
Cs1 NC1 NC11 'Cs'  
Ls1 NC11 NC12 'Ls'  
Rs1 NC12 NC2 'Rs'
```

```
.ends
```

```
*Sources
```

```
V1 vd 0 DC 'VDD'  
I1 b 0 '(10^X6)*1u'
```

```
* Initial Condition
```

```
.IC o=0 d='VDD'
```

```
.end
```

Descrição SPICE do circuito oscilador com cristal de 32,678 kHz:

```
*Circuit Oscillator with Cristal 32 kHz - 1
```

```
* Basic Circuit
```

```
* Final Version
```

```
.include param
```

```
*Circuit Description
```

```
** PMOS
```

```
M1 vd b d vd MODP W='X2*1u' L='X1*1u' M='int(X7)'
```

```
M2 vd b b vd MODP W='X2*1u' L='X1*1u'
```

```
M3 vd b out vd MODP W='X2*1u' L='X1*1u' M='int(X8)'
```

```
** NMOS
```

```
MN1 d o 0 0 MODN W='X4*1u' L='X3*1u'
```

```
MN2 out d 0 0 MODN W='X4*1u' L='X3*1u' M='int(X8)/int(X7)'
```

```
** Resistor
```

```
R1 o d 'X5*1k'
```

```
* Capacitor
```

```
C1 o 0 3p
```

```
C2 d 0 3p
```

```
Cf out 0 'Cl'
```

```
* Subcircuit
```

```
X1 o d crystal
```

```
** Crystal Model
```

```
** 32KhZ
```

```
.param Cp=1.5p Cs=3f Ls=7863 Rs=30k
```

```

.subckt crystal NC1 NC2

Cp1 NC1 NC2 'Cp'
Cs1 NC1 NC11 'Cs'
Ls1 NC11 NC12 'Ls'
Rs1 NC12 NC2 'Rs'

.ends

*Sources

V1 vd 0 DC 'VDD'
I1 b 0 '(10^X6)*1u'

* Initial Condition

.IC o=0 d='VDD'

.end

```

Descrição SPICE do circuito oscilador controlado por tensão:

```

*Oscillator Circuit VCO Type A
* Basic Circuit
* test version

.include param

*Circuit Description

** PMOS

M1 vd d o vd MODP W='X2*1u' L='X1*1u'
M2 vd o d vd MODP W='X2*1u' L='X1*1u'

** NMOS

M3 o d 0 0 MODN W='X4*1u' L='X3*1u'
M4 d o 0 0 MODN W='X4*1u' L='X3*1u'

** MOS Varactor

MC2 vc o vc vd MODP W='X6*1u' L='X5*1u' M='int(X8)'
MC1 vc d vc vd MODP W='X6*1u' L='X5*1u' M='int(X8)'

* Inductor

L1 o d 'X7*1n'

*Sources

V1 vd 0 DC 'VDD'
VC vc 0 DC 'VC'

* Initial Condition

.IC o=0 d='VDD'

.end

```

Apêndice B

Função fitness.m (função objetivo) para circuitos osciladores com frequência fixa na linguagem MatLab:

```
function [sc, sci] = fitness(x)
% sc retorna o scores total e sci retorna os valores parciais
global slash;
global simuladorRF;
global circuito
global ParDados
global genesLB;
global Freqmin;
global Freqmax;
global THD;
global Vfonte;
global Vdmin; % Tensão desejada
global Vdmax;
global EH;
global EHT;
global PNoise;
global Foffset;
global pPot;
global AreaCir;
global pesos;
global Best
global BestRes;
global Certo;
global nharm;
global LoadC;
global modoind;
global modo;
global dif;
global cont;
global namext;
% leitura dos parâmetros
% Tensões desejadas mínimas e máximas ou só amplitude desejada
if length(str2num(ParDados{3,2})) == 2
    vet=str2num(ParDados{3,2});
    Vdmin=vet(1); Vdmax=vet(2);
elseif length(str2num(ParDados{3,2})) == 1
    Vdmax=str2num(ParDados{3,2});
    Vdmin=0.0;
else
    Vdmax=Vfonte;
    Vdmin=0;
end
% Frequências
if length(str2num(ParDados{1,2}))==1
    Freqmax=str2num(ParDados{1,2});
    Freqmin=0.0;
else
    error('Por favor, insira uma ou duas frequências desejadas para o projeto!');
end
```

```

% Tensão da Fonte
if length(str2num(ParDados{4,2}))==1
    Vfonte = str2num(ParDados{4,2});
else
    error('Por favor, insira uma tensão de alimentação para o projeto');
end
% Número de Harmônicas
if length(str2num(ParDados{6,2}))==1
    nharm=str2num(ParDados{6,2})+1;
else
    error('Por favor insira o número de harmônicas a serem analisadas');
end
% Potências
if length(str2num(ParDados{5,2}))==1
    pPot = str2num(ParDados{5, 2})*1e-6;
else
    error('Por favor insira uma potência de referência para o projeto');
end
% Harmônicas
if length(str2num(ParDados{7,2}))==1 && length(str2num(ParDados{8,2}))== 1
    EH = str2num(ParDados{7,2})/100;
    EHT = str2num(ParDados{8,2})/100;
else
    error('Por favor insira os parâmetros EH e EHT');
end
if length(str2num(ParDados{9,2}))==1
    THD= str2num(ParDados{9,2})/100;
else
    error('Por favor insira um parâmetro THD');
end
% Ruído de Fase
phaux= str2num(ParDados{10,2});
b=1;
if mod(length(phaux),2)==0
    PNoise=zeros(length(phaux)/2,1);
    Foffset=zeros(length(phaux)/2,1);
    for a=1:2:length(phaux)
        PNoise(b)=phaux(a+1);
        Foffset(b)=phaux(a);
        b=b+1;
    end
else
    error('Os parâmetros de Ruído de Fase e Frequências de offset devem ter o mesmo número de elementos!');
end
% Jitter
auxJit=0;
% Outros parâmetros
if length(str2num(ParDados{11,2}))<=1
    AreaCir = str2num(ParDados{11, 2});
else
    error('Por favor insira menos de dois valores para o parâmetro Área');
end
if length(str2num(ParDados{2,2}))==1
    TF= str2num(ParDados{2,2});
elseif length(str2num(ParDados{2,2}))==0

```

```

    TF=0.001;
else
    error('Por favor insira apenas um valor para o parâmetro TF');
end
% Capacitor de carga
if length(str2num(ParDados{12,2}))==1
    LoadC=str2num(ParDados{12,2});
else
    error('Por favor, informe o valor do Capacitor de Carga!');
end
% Pesos
if length(str2num(ParDados{13,2}))==7
    pesos = str2num(ParDados{13,2});
else
    error('7 pesos devem ser fornecidos');
end
% os parâmetros x (gerados) vão de 0 a 1; a partir deles são gerados os parâmetros xr para o
arquivo de simulação
%xr = x.*dif+genesLB;
j=1;
for i=1:length(genesLB)
    if (dif(i) ~= 0)
        xr(i)= (x(j)*dif(i)+genesLB(i));
        j=j+1;
    else xr(i)=genesLB(i);
    end;
end;
% Cálculo da área do circuito
cond = [circuito slash 'AreaCirMea.m'];
if exist(cond)
    eval(['cd ', circuito]);
    AreaMed = AreaCirMea(xr);
    eval('cd ..');
else AreaMed = 0;
end;
% arquivo param tem os comando para simulação
arq = fopen([circuito slash 'param'],'w');
param_mod(arq, xr);
fclose(arq);
disp _____
cont=cont +1;
fprintf('simulação = %d\n', cont);
if exist('circuito.msnnoi0', 'file')
    delete('circuito.msnnoi0');
end
if exist('circuito.msn0', 'file')
    delete('circuito.msn0');
end
if exist('circuito.msnf0', 'file')
    delete('circuito.msnf0');
end
% executa a simulação
[~, b] = system ([simuladorRF circuito slash 'circuito.sp']);
% [~, b] = system (['START /Realtime/wait/min C:\synopsys\Hspice_A-2008.03\BIN\hspicerf.exe
' circuito \circuito.sp']);
Certo=1;

```

```

% lê os resultados das simulações
try
    arq = fopen('circuito.msnnoi','r');
    MeasP=LeMeas(arq, 3);
    fclose(arq);
    if nharm>15
        auxmb0=4;
    else
        auxmb0=3;
    end
    arq = fopen('circuito.msnf0','r');
    MeasB=LeMeas(arq, auxmb0);
    fclose(arq);
    arq = fopen('circuito.msn0','r');
    MeasR=LeMeas(arq, 3);
    fclose(arq);
    na=4+nharm;
    nb=3+length(PNoise)+auxJit;
    if length(MeasB)~=na && length(MeasP)~=nb && length(MeasR) ~= 4
        Certo=0;
    end
catch
    Certo=0;
end
% Obtem os resultados das simulações
if Certo==1
    try
        Vauxmax=MeasR(1);
        Vauxmin=MeasR(2);
        Faux=MeasP(1);
        Paux=MeasB(1);
        Vout=MeasB(2:2+nharm);
        % Calcula THD, EHD
        EHaux=0;
        OHaux=0;
        % Indice 1 = DC, Indice 2 = Fo, Indice 3 = 2*Fo ...
        for a=3:1:length(Vout)
            if mod(a,2)==1
                EHaux=EHaux+Vout(a)^2;
            else
                OHaux=OHaux+Vout(a)^2;
            end
        end
        THDaux=sqrt(EHaux+OHaux)/sqrt(EHaux+OHaux+Vout(2)^2);
        EHaux=sqrt(EHaux)/sqrt(EHaux+OHaux+Vout(2)^2);
    catch
        Certo=0;
    end
    Pavg=sum(MeasP(2:end-2-auxJit))/(size(PNoise,2));
    MeasPN=MeasP(2:end-2-auxJit);
    if (Pavg===-300)
        Certo=0;
    end
end
% verifica se a simulação gerou todos resultados
if Certo==1

```



```

% calculate the circuit performance
% Tensão na saída
VO=0;
if (length(str2num(ParDados{3,2}))>1)
    P1=0;
    P2=0;
    if Vauxmax<Vdmax
        P1=abs((Vauxmax-Vdmax)/Vdmax);
    end
    if Vauxmin > Vdmin
        P2=abs((Vauxmin-Vdmin)/Vdmin);
    end
    VO=P1+P2;
    if (VO ~= 0)
        VO=VO+1/(Vauxmax - Vauxmin);
    end
else
    if (Vauxmax - Vauxmin) < Vdmax
        VO=abs(((Vauxmax - Vauxmin)-Vdmax)/Vdmax)+1/(Vauxmax - Vauxmin);
    end
end
% Frequência (FREQ)
TF1=1.0-(TF/100.0);
TF2=1.0+(TF/100.0);
FREQ=0;
if (Faux >= TF2*Freqmax*1e+6) || ( Faux <= TF1*Freqmax*1e+6)
    FREQ = abs((Faux - (Freqmax*1e+6))/(Freqmax*1e+6));
end
% PHASENOISE
PN=0;
MPaux=MeasPN;
for i=1:1:size(PNoise,2)
    if abs(PNoise(i))>abs(MPaux(i))
        PN=PN+(abs(abs(MPaux(i))-abs(PNoise(i)))/abs(PNoise(i)));
    end
end
% EHD
EvenH=0;
if ( EHaux <=(EH - EHT)) || (EHaux >= (EH+EHT))
    EvenH = abs((EHaux - EH)/EH);
end
% THD
TotalH=0;
if ( THDaux > (THD))
    TotalH = abs((THDaux - THD)/THD);
end
% Consumo de potência
FPot = abs(Paux)/pPot;
% Area
Farea=0;
if AreaMed~=0
    Farea=abs(AreaMed-AreaCir)/AreaCir;
end
%final score
% ['Pesos (FREQ, Output, Power, EH, THD, PN, Area) ['10 100 1 1 1 1 1']];

```

```

sc = (pesos(2)*VO + pesos(1)*FREQ + pesos(3)*FPot + pesos(4)*EvenH + pesos(5)*TotalH +
pesos(6)*PN + pesos(7)*Farea);
sci = [FREQ VO FPot PN EvenH TotalH Farea];
% salva a melhor soluçao
if(Best.score > sc)
    arq = fopen([circuito slash 'results' slash 'optimos.' modo namext], 'a+');
    fprintf(arq, '%d %.3g\n', cont, sc);
    fclose(arq);
    beep;
    Best.score = sc;
    fprintf('*>');
    arq = fopen([circuito slash 'paramop'], 'w');
    param_mod(arq, xr);
    fprintf(arq, '*Score=%.3g Freq= %.5g Hz (FREQ= %.3g %) VMAX= %.3gV VMIN= %.3gV
(VO= %.3g %) \n*Pot=%.2fuW (FPot= %.3g %) EHT= %.3g %% (EvenH= %.3g%%) THD=
%.3g %% (TotalH= %.3g%%) ', sc, Faux, FREQ*100.0, Vauxmax, Vauxmin, VO*100.0,
Paux*1e6, FPot*100.0, EHaux*100.0, EvenH, THDaux*100, TotalH);
    for i=1:1:size(PNoise,2)
        fprintf(arq, '*PhaseNoise%d= %.3g dB/Hz ', i, MeasPN(i));
    end
    fprintf(arq, '(PN= %.3g%%)\n\n', PN*100.0);
    fclose(arq);
    BestRes{modoind} = Best;
    if(Best.scoreT > sc)
        Best.scoreT = sc;
        Best.parameters = xr;
        arq = fopen([circuito slash 'paramopT'], 'w');
        param_mod(arq, xr);
        fprintf(arq, '*Score=%.3g Freq= %.5g Hz (FREQ= %.3g %) VMAX= %.3gV VMIN= %.3gV
(VO= %.3g %) \n*Pot=%.2fuW (FPot= %.3g %) EHT= %.3g %% (EvenH= %.3g%%) THD=
%.3g %% (TotalH= %.3g%%) ', sc, Faux, FREQ*100.0, Vauxmax, Vauxmin, VO*100.0,
Paux*1e6, FPot*100.0, EHaux*100.0, EvenH, THDaux*100, TotalH);
        for i=1:1:size(PNoise,2)
            fprintf(arq, '*PhaseNoise%d= %.3g dB/Hz ', i, MeasPN(i));
        end
        fprintf(arq, '(PN= %.3g%%)\n\n', PN*100.0);
        fclose(arq);
        BestRes{modoind} = Best;
    end;
end;
% Resultados de performance
fprintf('*Score=%.3g Freq= %.5g Hz (FREQ= %.3g %) VMAX= %.3gV VMIN= %.3gV
(VO= %.3g %) \n*Pot=%.2fuW (FPot= %.3g %) EHT= %.3g %% (EvenH= %.3g%%) THD=
%.3g %% (TotalH= %.3g%%) ', sc, Faux, FREQ*100.0, Vauxmax, Vauxmin, VO*100.0,
Paux*1e6, FPot*100.0, EHaux*100.0, EvenH, THDaux*100, TotalH);
for i=1:1:size(PNoise,2)
    fprintf('*PhaseNoise%d= %.3g dB/Hz ', i, MeasPN(i));
end
fprintf('(PN= %.3g%%)\n\n', PN*100.0);
% caso ocorra problemas na simulacao
else sc = inf('double');
sci = [ inf('double') inf('double') inf('double') inf('double') inf('double') inf('double')
inf('double')];
fprintf('Score=%.3g \n\n', sc);
end;
% parâmetros usados

```

```

for i = 1:length(xr)
    if (mod(i, 10) == 0)
        fprintf('\n');
    end
    fprintf('X%i= %1.2f ', i, xr(i));
end;
fprintf('\n');
end

```

Função param_mod.m em Matlab para circuitos osciladores com frequência fixa que é responsável por preparar o arquivo de entrada da simulação:

```

function param_mod(arq,x)
% função que recebe o arquivo de parâmetros e prepara o arquivo para simulação
global ParDados;
global modelo;
global Freqmin;
global Freqmax;
global slash;
global Vfonte;
global Foffset;
global nharm;
global LoadC;
% da include do modelo de simulação
fprintf(arq, '.include ../%s../%sModels%s%s\r\n', slash, slash, slash, modelo);
% coloca os parâmetros
fprintf(arq, '.Param ');
for i = 1:length(x)
    if (mod(i, 7) == 0) fprintf(arq, '\r\n.Param ');
    end
    fprintf(arq, 'X%i= %1.3f ', i, x(i));
end;
fprintf(arq, '\r\n');
fprintf(arq, '** ( ');
for i = 1:length(x)
    fprintf(arq, '%1.3f ', x(i));
end;
fprintf(arq, '\r\n');
    Freqmax=str2num(ParDados{1,2});
    Freqmin=0.0;
fprintf(arq, '.param VDD=%1.2f\r\n', Vfonte);
fprintf(arq, '.param Cl=%1.2f\r\n', LoadC);
    fprintf(arq, '.SNOSC TRES=%g PERIOD=%g OSCNODE=d TRINIT=%g UIC\r\n',
inv(Freqmax*1e6)/1000,inv(Freqmax*1e6), 500/(Freqmax*1e6));
fprintf(arq, '.MEASURE PHASENOISE freq1 PARAM="HERTZ[1]"\r\n');
fprintf(arq, '.MEASURE SNFD pot PARAM="P(V1)[0]"\r\n');
for a=0:1:min(nharm,25)
    fprintf(arq, '.MEASURE SNFD v%d PARAM="VM(out)[%d]"\r\n', a, a);
end
fprintf(arq, '.MEASURE SN zmin MIN V(out)\r\n');
fprintf(arq, '.MEASURE SN zmax MAX V(out)\r\n');
if length(Foffset)>1
    liminf=min(Foffset);
    limsup=max(Foffset);
    fprintf(arq, '.PHASENOISE V(out) LIN 100 %g %g\r\n', liminf*1e+3, limsup*1e+3);

```

```
for i=1:1:size(Foffset,2)
    fprintf(arq, '.MEASURE PHASENOISE phn%d FIND phnoise at=%g\r\n',i, Foffset(i)*1e+3);
end
elseif length(Foffset)==1
    liminf=Foffset/10;
    limsup=Foffset;
    fprintf(arq, '.PHASENOISE V(out) LIN 100 %g %g\r\n', liminf*1e+3, limsup*1e+3);
    fprintf(arq, '.MEASURE PHASENOISE phn%d FIND phnoise at=%g\r\n',1, Foffset*1e+3);
end
end
```

Apêndice C

Função fitness.m (função objetivo) para circuitos VCO na linguagem

MatLab:

```
function [sc, sci] = fitness(x)
% sc da o scores total e sci sao os valores parciais
global slash;
global simuladorRF;
global circuito
global ParDados
global genesLB;
global VCOMODO;
global Freqaux;
global Jitter;
global THD;
global Vfonte;
global Vdmin; % Tensão desejada
global Vdmax;
global Vcaux;
global Vcmax;
global EH;
global EHT;
global PNoise;
global Foffset;
global pPot;
global AreaCir;
global pesos;
global Best
global BestRes;
global Certo;
global Fjitter;
global nharm;
global LoadC;
global modoind;
global modo;
global dif;
global cont;
global namext;
% leitura dos parametros
% Tensões desejadas minimas e maximas ou só máxima
if length(str2num(ParDados{4,2})) == 2
    vet=str2num(ParDados{4,2});
    Vdmin=vet(1); Vdmax=vet(2);
elseif length(str2num(ParDados{4,2})) == 1
    Vdmax=str2num(ParDados{4,2});
    Vdmin=0.0;
else
    Vdmax=Vfonte;
    Vdmin=0;
end
% Frequências
if length(str2num(ParDados{1,2}))>=2
    Freqaux=str2num(ParDados{1,2});
```

```

VCOMODO=1;
elseif length(str2num(ParDados{1,2}))==1
    Freqaux=str2num(ParDados{1,2});
    VCOMODO=0;
else
    error('Por favor, insira uma ou mais frequências desejadas para o projeto!');
end
% Tensão da Fonte
if length(str2num(ParDados{5,2}))==1
    Vfonte = str2num(ParDados{5,2});
else
    error('Por favor, insira uma tensão de alimentação para o projeto');
end
% Tensão de controle
if length(str2num(ParDados{3,2}))>=2
    Vcaux=str2num(ParDados{3,2});
    Vcmax=max(Vcaux);
elseif length(str2num(ParDados{3,2}))==1
    Vcmax=str2num(ParDados{3,2});
end
if length(Vcaux)~=length(Freqaux)
    error('Por favor insira a mesma quantidade de tensões de controles e de frequências desejadas');
end
% Número de Harmônicas
if length(str2num(ParDados{7,2}))==1
    nharm=str2num(ParDados{7,2})+1;
else
    error('Por favor insira o número de harmônicas a serem analisadas');
end
% Potências
if length(str2num(ParDados{6,2}))==1
    pPot = str2num(ParDados{6, 2})*1e-6;
else
    error('Por favor insira uma potência de referência para o projeto');
end
% Harmônicas
if length(str2num(ParDados{8,2}))==1 && length(str2num(ParDados{9,2}))== 1
    EH = str2num(ParDados{8,2})/100;
    EHT = str2num(ParDados{9,2})/100;
else
    error('Por favor insira os parâmetros EH e EHT');
end
if length(str2num(ParDados{10,2}))==1
    THD= str2num(ParDados{10,2})/100;
else
    error('Por favor insira um parâmetro THD');
end
% Ruído de Fase
phaux= str2num(ParDados{11,2});
b=1;
if mod(length(phaux),2)==0
    Foffset=zeros(length(phaux)/2,1);
    PNoise=zeros(length(phaux)/2,1);
    for a=1:2:length(phaux)
        PNoise(b)=phaux(a+1);
    end
end

```

```

        Foffset(b)=phaux(a);
        b=b+1;
    end
else
    error('Os parâmetros de Ruído de Fase e Frequências de offset devem ter o mesmo número
de elementos!');
end
% Jitter
auxJit=0;
if length(str2num(ParDados{12,2}))==1
    Jitter = str2num(ParDados{12,2});
elseif length(str2num(ParDados{12,2}))>1
    error('Por favor insira um ou nenhum valor para o parâmetro Jitter');
end
if length(str2num(ParDados{12,2}))== 1 && length(str2num(ParDados{13,2}))==2
    Fjitter = str2num(ParDados{13,2});
    auxJit=1;
else
    error('Caso queira otimizar o jitter é necessário utilizar 2 frequências de range!');
end
% Outros parâmetros
if length(str2num(ParDados{14,2}))<=1
    AreaCir = str2num(ParDados{14, 2});
else
    error('Por favor insira menos de dois valores para o parâmetro Área');
end
if length(str2num(ParDados{2,2}))==1
    TF= str2num(ParDados{2,2});
elseif length(str2num(ParDados{2,2}))==0
    TF=0.1;
else
    error('Por favor insira apenas um valor para o parâmetro TF');
end
% Capacitor de carga
if length(str2num(ParDados{15,2}))==1
    LoadC=str2num(ParDados{15,2});
else
    error('Por favor, informe o valor do Capacitor de Carga!');
end
% Pesos
if length(str2num(ParDados{16,2}))==8
    pesos = str2num(ParDados{16,2});
else
    error('8 pesos devem ser fornecidos');
end
% os parâmetros x (gerados) vão de 0 a 1; a partir deles são gerados os parâmetros xr para o
arquivo de simulação
j=1;
for i=1:length(genesLB)
    if (dif(i) ~= 0)
        xr(i)= (x(j)*dif(i)+genesLB(i));
        j=j+1;
    else xr(i)=genesLB(i);
    end;
end;
% Calculo da área do circuito

```

```

cond = [circuito slash 'AreaCirMea.m'];
if exist(cond)
    eval(['cd ', circuito]);
    AreaMed = AreaCirMea(xr);
    eval('cd ..');
else AreaMed = 0;
end;
% arquivo param tem os comandos para simulacao
arq = fopen([circuito slash 'param'],'w');
param_mod(arq, xr);
fclose(arq);
disp _____
cont=cont +1;
fprintf('simulação = %d\n', cont);
if exist('circuito.mb0', 'file')
    delete('circuito.mb0');
end
if exist('circuito.mp0', 'file')
    delete('circuito.mp0');
end
if exist('circuito.mr0', 'file')
    delete('circuito.mr0');
end
% executa a simulacao
[~, b] = system ([simuladorRF circuito slash 'circuito.sp']);
certoVCO=1;
Certo=1;
% lê os resultados das simulações
try
    arq = fopen('circuito.mp0','r');
    MeasP=LeMeas(arq, 3);
    fclose(arq);
    if nharm>15
        auxmb0=4;
    else
        auxmb0=3;
    end
    arq = fopen('circuito.mb0','r');
    MeasB=LeMeas(arq, auxmb0);
    fclose(arq);
    arq = fopen('circuito.mr0','r');
    MeasR=LeMeas(arq, 3);
    fclose(arq);
    na=4+nharm;
    nb=3+length(PNoise)+auxJit;
    nc=4;
    if VCOMODO==1
        nb=length(Freqaux)*(nb+1);
        nc=5*length(Freqaux);
    end
    if length(MeasB)~=na && length(MeasP)~=nb && length(MeasR)~=nc
        Certo=0;
    end
catch
    certoVCO=0;
    Certo=0;

```



```

end
% obtêm os resultados das simulações
try
if Certo==1

    if(VCOMODO==0)
        Faux=MeasP(1);
        Pavg=sum(MeasP(2:end-2-auxJit))/(size(PNoise,2));

        if auxJit==1
            MeasJ=MeasP(end-2);
        end
        MeasPN=MeasP(2:end-2-auxJit);
        Vauxmax=MeasR(1);
        Vauxmin=MeasR(2);
        Amp=Vauxmax-Vauxmin;
    else
        Faux=zeros(length(Freqaux),1);
        raux=reshape(MeasR,length(MeasR)/length(Freqaux), length(Freqaux));
        Vauxmax=min(raux(:,2));
        Vauxmin=max(raux(:,3));
        Amp=min(raux(:,2)-raux(:,3));
        if auxJit==1
            MeasJ=zeros(length(Freqaux),1);
        end
        MeasPN=zeros(length(Freqaux),length(PNoise));
        mat=reshape(MeasP,length(MeasP)/length(Freqaux),length(Freqaux));
        for a=1:1:length(Freqaux)
            Faux(a)=mat(a,2);
            if auxJit==1
                MeasJ(a)=mat(a, end-2);
            end
            MeasPN(a, :)=mat(a, 3: end-2-auxJit);
        end
        Pavg=mean(mean(MeasPN));
    end
    if (Pavg== -300)
        Certo=0;
    end
    Paux=MeasB(1);
    Vout=MeasB(2:2+nharm);
    % Calcula THD, DHP and DHI
    EHaux=0;
    OHaux=0;
    % Indice 1 = DC, Indice 2 = Fo, Indice 3 = 2*Fo ...
    for a=3:1:length(Vout)
        if mod(a,2)==1
            EHaux=EHaux+Vout(a)^2;
        else
            OHaux=OHaux+Vout(a)^2;
        end
    end
    THDaux=sqrt(EHaux+OHaux)/sqrt(EHaux+OHaux+Vout(2)^2);
    EHaux=sqrt(EHaux)/sqrt(EHaux+OHaux+Vout(2)^2);
end
catch

```

```

Certo=0;
end
% verifica se a simulacao gerou todos resultados
if Certo==1 && certoVCO==1
% cálculo da performance do circuito
% Tensão na saída
VO=0;
if (length(str2num(ParDados{3,2}))>1)
P1=0;
P2=0;
if Vauxmax<Vdmax
P1=abs((Vauxmax-Vdmax)/Vdmax);
end
if Vauxmin > Vdmin
P2=abs((Vauxmin-Vdmin)/Vdmin);
end
VO=P1+P2;
if (VO ~= 0)
VO=VO+1/(Amp);
end
else
if Amp < Vdmax
VO=abs((Amp-Vdmax)/Vdmax)+1/Amp;
end
end
% Frequência (FREQ)
TF1=1.0-(TF/100.0);
TF2=1.0+(TF/100.0);
FREQ=0;
if VCOMODO==0
if (Faux >= TF2*Freqmax*1e+6) || ( Faux <= TF1*Freqmax*1e+6)
FREQ = abs((Faux - (Freqmax*1e+6))/(Freqmax*1e+6));
end
else
FREQ=0;
for a=1:1:length(Faux)
if (Faux(a) >= TF2*Freqaux(a)*1e+6) || (Faux(a) <= TF1*Freqaux(a)*1e+6)
FREQ=abs(Faux(a)-(Freqaux(a)*1e+6))/(Freqaux(a)*1e+6)+FREQ;
end
end
perc=(max(Faux)-min(Faux))/(max(Freqaux)*1e+6-min(Freqaux)*1e+6)+10*eps;
if perc>TF1 && perc<1
perc=1;
end
FREQ=FREQ/perc;
end
% JITTER
Jl=0;
if auxJit && min(MeasJ) >= Jitter
Jl=abs(Jitter-min(MeasJ))/Jitter;
end
% PHASENOISE
PN=0;
MPaux=MeasPN;
if VCOMODO

```

```

    MPaux=max(MeasPN);
end
for i=1:1:length(PNoise)
    if abs(PNoise(i))>abs(MPaux(i))
        PN=PN+(abs(abs(MPaux(i))-abs(PNoise(i)))/abs(PNoise(i)));
    end
end
% DHP
EvenH=0;
if ( EHaux <=(EH - EHT)) || (EHaux >= (EH+EHT))
    EvenH = abs((EHaux - EH)/EH);
end
% THD
TotalH=0;
if ( THDaux > (THD))
    TotalH = abs((THDaux - THD)/THD);
end
% Potência consumida
FPot = abs(Paux)/pPot;
% Area
Farea=0;
if AreaMed~=0
    Farea=abs(AreaMed-AreaCir)/AreaCir;
end
% score final
sc = (pesos(2)*VO + pesos(1)*FREQ + pesos(3)*FPot + pesos(4)*EvenH + pesos(5)*TotalH +
pesos(6)*PN+ pesos(7)*JI + pesos(8)*Farea);
sci = [FREQ VO FPot PN EvenH TotalH JI Farea];
% Salva a melhor solução
if(Best.score > sc)
    arq = fopen(['circuito slash 'results' slash 'optimos.' modo namext'],'a+');
    fprintf(arq,'%d %.3g\n', cont, sc);
    fclose(arq);
    beep;
    Best.score = sc;
    fprintf('*>');
    arq = fopen(['circuito slash 'paramop'],'w');
    param_mod(arq,xr);
    if(VCOMODO==0)
        fprintf(arq,'*Score=%.3g Freq= %.5g Hz (FREQ= %.3g %) VMAX= %.3gV VMIN=
%.3gV (VO= %.3g %)\n*Pot=%.2fuW (FPot= %.3g %) EHT= %.3g %% (EvenH=
%.3g%%) THD= %.3g %% (TotalH= %.3g%%) ', sc, Faux, FREQ*100.0, Vauxmax, Vauxmin,
VO*100.0, Paux*1e6, FPot*100.0, EHaux*100.0, EvenH, THDaux*100, TotalH);
        if auxJit
            fprintf(arq,'Jitter= %.3g db/Hz (JI= %.3g %)\n', MeasJ, JI);
        end
        for i=1:1:size(PNoise,2)
            fprintf(arq,'*PhaseNoise%d= %.3g dB/Hz ', i, MeasPN(i));
        end
        fprintf(arq,'(PN= %.3g%%)\n\n', PN*100.0);
    else
        fprintf(arq,'*Score=%.3g Freq (V1)= %.5g Hz (V2) = %.5g Hz (FREQ= %.3g %) VMAX=
%.3gV VMIN= %.3gV (VO= %.3g %)\n*Pot=%.2fuW (FPot= %.3g %) EHT= %.3g %%
(EvenH= %.3g%%) THD= %.3g %% (TotalH= %.3g%%) ', sc, min(Faux), max(Faux),
FREQ*100.0, Vauxmax, Vauxmin, VO*100.0, Paux*1e6, FPot*100.0, EHaux*100.0, EvenH,
THDaux*100, TotalH);
    end
end

```

```

    if auxJit
        fprintf(arq,'Jitter (V1)= %.3g db/Hz (V2)= %.3g db/Hz (JI= %.3g %%)\n', min(MeasJ),
max(MeasJ), JI);
    end
    for i=1:1:size(PNoise,2)
        fprintf(arq,'*PhaseNoise(%d)(V1)= %.3g dB/Hz (V2)= %.3g dB/Hz ', i, min(MeasPN(:,i)),
max(MeasPN(:,i)));
    end
    fprintf(arq,'(PN= %.3g%%)\n\n', PN*100.0);
end
fclose(arq);
BestRes{modoind} = Best;
if(Best.scoreT > sc)
    Best.scoreT = sc;
    Best.parameters = xr;
    arq = fopen([circuito slash 'paramopT'],'w');
    param_mod(arq,xr);
    if(VCOMODO==0)
        fprintf(arq,'*Score=%.3g Freq= %.5g Hz (FREQ= %.3g %) VMAX= %.3gV VMIN=
%.3gV (VO= %.3g %)\n*Pot=%.2fuW (FPot= %.3g %) EHT= %.3g %% (EvenH=
%.3g%%) THD= %.3g %% (TotalH= %.3g%%) ', sc, Faux, FREQ*100.0, Vauxmax, Vauxmin,
VO*100.0, Paux*1e6, FPot*100.0, EHaux*100.0, EvenH, THDaux*100, TotalH);
        if auxJit
            fprintf(arq,'Jitter= %.3g db/Hz (JI= %.3g %%)\n', MeasJ, JI);
        end
        for i=1:1:size(PNoise,2)
            fprintf(arq,'*PhaseNoise%d= %.3g dB/Hz ', i, MeasPN(i));
        end
        fprintf(arq,'(PN= %.3g%%)\n\n', PN*100.0);
    else
        fprintf(arq,'*Score=%.3g Freq (V1)= %.5g Hz (V2) = %.5g Hz (FREQ= %.3g %)
VMAX= %.3gV VMIN= %.3gV (VO= %.3g %)\n*Pot=%.2fuW (FPot= %.3g %) EHT= %.3g
%% (EvenH= %.3g%%) THD= %.3g %% (TotalH= %.3g%%) ', sc, min(Faux), max(Faux),
FREQ*100.0, Vauxmax, Vauxmin, VO*100.0, Paux*1e6, FPot*100.0, EHaux*100.0, EvenH,
THDaux*100, TotalH);
        if auxJit
            fprintf(arq,'Jitter (V1)= %.3g db/Hz (V2)= %.3g db/Hz (JI= %.3g %%)\n', min(MeasJ),
max(MeasJ), JI);
        end
        for i=1:1:size(PNoise,2)
            fprintf(arq,'*PhaseNoise(%d)(V1)= %.3g dB/Hz (V2)= %.3g dB/Hz ', i,
min(MeasPN(:,i)), max(MeasPN(:,i)));
        end
        fprintf(arq,'(PN= %.3g%%)\n\n', PN*100.0);
    end
    fclose(arq);
    BestRes{modoind} = Best;
end;
end;
% resultados de performance
if(VCOMODO==0)
    fprintf('*Score=%.3g Freq= %.5g Hz (FREQ= %.3g %) VMAX= %.3gV VMIN= %.3gV
(VO= %.3g %)\n*Pot=%.2fuW (FPot= %.3g %) EHT= %.3g %% (EvenH= %.3g%%) THD=
%.3g %% (TotalH= %.3g%%) ', sc, Faux, FREQ*100.0, Vauxmax, Vauxmin, VO*100.0,
Paux*1e6, FPot*100.0, EHaux*100.0, EvenH, THDaux*100, TotalH);
    if auxJit

```

```

    fprintf('Jitter= %.3g db/Hz (JI= %.3g %%)\n', MeasJ, JI);
end
for i=1:size(PNoise,2)
    fprintf('*PhaseNoise%d= %.3g dB/Hz ', i, MeasPN(i));
end
fprintf('(PN= %.3g%%)\n\n', PN*100.0);
else
    fprintf('*Score=%.3g Freq (V1)= %.5g Hz (V2)= %.5g Hz (FREQ= %.3g %) VMAX=
%.3gV VMIN= %.3gV (VO= %.3g %)\n*Pot=%.2fuW (FPot= %.3g %) EHT= %.3g %%
(EvenH= %.3g%%) THD= %.3g %% (TotalH= %.3g%%) ', sc, min(Faux), max(Faux),
FREQ*100.0, Vauxmax, Vauxmin, VO*100.0, Paux*1e6, FPot*100.0, EHaux*100.0, EvenH,
THDaux*100, TotalH);
    if auxJit
        fprintf('Jitter (V1)= %.3g db/Hz (V2)= %.3g db/Hz (JI= %.3g %)\n', min(MeasJ),
max(MeasJ(2)), JI);
    end
    for i=1:size(PNoise,2)
        fprintf('*PhaseNoise(%d)(V1)= %.3g dB/Hz (V2)= %.3g dB/Hz ', i, min(MeasPN(:,i)),
max(MeasPN(:,i)));
    end
    fprintf('(PN= %.3g%%)\n\n', PN*100.0);
end
% caso haja problemas na simulacao
else sc = inf('double');
    sci = [ inf('double') inf('double') inf('double') inf('double') inf('double') inf('double')
inf('double') inf('double')];
    fprintf('Score=%.3g \n\n',sc);
end;
% imprime os parâmetros utilizados
for i = 1:length(xr)
    if (mod(i, 10) == 0)
        fprintf('\n');
    end
    fprintf('X%i= %1.2f ', i, xr(i));
end;
fprintf('\n');
end

```

Função param_mod.m na linguagem Matlab para circuitos VCO que é responsável por preparar o arquivo de entrada da simulação:

```

function param_mod(arq,x)
% função que recebe o arquivo de parâmetros e os parâmetros propriamente ditos
global ParDados;
global modelo;
global Freqaux;
global Vcaux;
global Vcmax;
global slash;
global Vfonte;
global Foffset;
global Fjitter;
global nharm;
global LoadC;

```

```

% inclui o arquivo do modelo de simulação
fprintf(arq, '.include ../%s../%sModels%s%\r\n', slash, slash, slash, modelo);
% acrescenta os parâmetros no arquivo de simulação
fprintf(arq, '.Param ');
for i = 1:length(x)
    if (mod(i, 7) == 0) fprintf(arq, '\r\n.Param ');
    end
    fprintf(arq, 'X%i= %1.3f ', i, x(i));
end;
fprintf(arq, '\r\n');
fprintf(arq, '** ( ');
for i = 1:length(x)
    fprintf(arq, '%1.3f ', x(i));
end;
fprintf(arq, ')\r\n');
if (length(str2num(ParDados{1,2}))>1)
    vet=str2num(ParDados{1,2});
    Freqmin=vet(1); Freqmax=vet(2);
    VCOMODO=1;
else
    Freqmax=str2num(ParDados{1,2});
    Freqmin=0.0;
    VCOMODO=0;
end
fprintf(arq, '.param VDD=%1.2f\r\n', Vfonte);
fprintf(arq, '.param Cl=%1.2f\r\n', LoadC);
if (VCOMODO == 0)
    fprintf(arq, '.HBOSC TONE=100 NHARMS=%d PROBENODE=d,o,"VDD/2"\r\n',
Freqmax*1e6);
else
    fprintf(arq, '.param VC=%1.2f\r\n', Vcmax);
    fprintf(arq, '.HBOSC TONE=%g NHARMS=%d PROBENODE=d,o,"VDD/2"
FSPTS= 2, %g, %g sweep VC POI %d', (max(Freqaux)+min(Freqaux))*0.5e6, nharm,
min(Freqaux)*1e6, max(Freqaux)*1e6, length(Vcaux));
    for a=1:1:length(Vcaux)
        fprintf(arq, '%1.2f', Vcaux(a));
    end
    fprintf(arq, '\r\n');
end
fprintf(arq, '.MEASURE PHASENOISE freq1 PARAM="HERTZ[1]"\r\n');
fprintf(arq, '.MEASURE HBOSC pot PARAM="P(V1)[0]"\r\n');
for a=0:1:min(nharm,25)
    fprintf(arq, '.MEASURE HBOSC v%d PARAM="VM(o,d)[%d]"\r\n', a, a);
end
fprintf(arq, '.MEASURE HBTR zmin MIN V(o,d)\r\n');
fprintf(arq, '.MEASURE HBTR zmax MAX V(o,d)\r\n');
if length(Foffset)>=1 && length(Fjitter)==2
    liminf=min(min(Foffset), min(Fjitter));
    limsup=max(max(Foffset), max(Fjitter));
% % disp(Foffset)
% % disp(Fjitter)

```

```

fprintf(arq, '.PHASENOISE V(o,d) LIN 100 %g %g\r\n', liminf*1e+3, limsup*1e+3);
for i=1:1:size(Foffset,2)
    fprintf(arq, '.MEASURE PHASENOISE phn%d FIND phnoise at=%g\r\n',i,
Foffset(i)*1e+3);
    end
    fprintf(arq, '.MEASURE PHASENOISE zjitter RMSJITTER phnoise from %g to %g
units =rad\r\n', min(Fjitter)*1e+3, max(Fjitter)*1e+3);
elseif length(Foffset)==1 && length(Fjitter)~=2
    limsup=Foffset;
    liminf=Foffset/10;
    fprintf(arq, '.PHASENOISE V(o,d) LIN 100 %g %g\r\n', liminf*1e+3, limsup*1e+3);
    for i=1:1:size(Foffset,2)
        fprintf(arq, '.MEASURE PHASENOISE phn%d FIND phnoise at=%g\r\n',i,
Foffset(i)*1e+3);
    end
elseif length(Fjitter)==2
    liminf=min(Fjitter);
    limsup=max(Fjitter);
    fprintf(arq, '.PHASENOISE V(o,d) LIN 100 %g %g\r\n', liminf*1e+3, limsup*1e+3);
    fprintf(arq, '.MEASURE PHASENOISE zjitter RMSJITTER phnoise from %g to %g
units =rad\r\n', min(Fjitter)*1e+3, max(Fjitter)*1e+3);
end
end
end

```

Apêndice D

Os comandos de simulação SNOSC e HBOSC estão exemplificados abaixo (quando o circuito oscilador tiver frequência fixa, o comando *sweep* é omitido):

```
.SNOSC TRES=Tstep PERIOD=Tperiod OSCNODE=o TRINIT=Tmax SWEEP VC POI Np Vp1 ... Vpn
```

```
.HBOSC TONE=Fo NHARMS=N PROBENODE=d,o,'VDD/2' SWEEP VC POI Np Vp1 ... Vpn
```

A análise de ruído de fase é feita através do comando a seguir (todas as frequências são tomadas em relação a frequência fundamental de oscilação encontrada pelo algoritmo):

```
.PHASENOISE V(out) LIN 100 F1 F2
```

As medidas da análise de ruído de fase e *jitter* são feitas com os seguintes comandos:

```
.MEASURE PHASENOISE freq1 PARAM='HERTZ[1]': Medida de frequência
```

```
.MEASURE PHASENOISE phni FIND phnoise at=Fi Medida de ruído de fase
```

```
.MEASURE PHASENOISE jitter RMSJITTER from=Fj1 to Fj2 Medida de jitter
```

Observa-se que a frequência de oscilação pode ser obtida por outros comandos, mas apenas as medidas nas análises de ruído de fase e nos domínios do tempo são fornecidas para cada tensão de controle analisada no comando *sweep*.

Quando a análise SNOSC é empregada, as medidas são obtidas com:

```
.MEASURE SNFD pot PARAM='P(V1)[0]' Medida da potência
```

```
.MEASURE SNFD vi PARAM='VM(out)[i]' Medida das harmônicas
```

```
.MEASURE SN vmax MAX V(out) Medida da máxima tensão
```

```
.MEASURE SN vmin MIN V(out) Medida da mínima tensão
```

É importante ressaltar que: v_i representa a magnitude da i -ésima harmônica desejada; *MEASURE SNFD* fornece valores no domínio da frequência; e *MEASURE SN* fornece valores no domínio do tempo.

Quando a análise HBOSC é empregada, as medidas são obtidas com:

```
.MEASURE HBOSC pot PARAM='P(V1)[0]' Medida da potência
```

```
.MEASURE HBOSC vi PARAM='VM(out)[i]' Medida das harmônicas
```

```
.MEASURE HBTR vmax MAX V(out) Medida da máxima tensão
```

```
.MEASURE HBTR vmin MIN V(out) Medida da mínima tensão
```

É importante ressaltar que: v_i representa a magnitude da i -ésima harmônica desejada; *MEASURE HBOSC* fornece valores no domínio da frequência; e *MEASURE HBTR* fornece valores no domínio do tempo.

A sintaxe completa de todos os comandos utilizados está descrita nos manuais do HSpice RF (SYNOPSYS, 2008a, 2008b, 2008c, 2008d).