



**Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Curso de Engenharia Eletrônica**

**MODELAGEM DE UM TRANSCEPTOR ZIGBEE
UTILIZANDO A LINGUAGEM VERILOG-AMS**

**Autor: Regina Silva Campos
Orientador: Wellington Avelino do Amaral**

**Brasília, DF
2014**



REGINA SILVA CAMPOS

**TÍTULO: MODELAGEM DE UM TRANSCEPTOR ZIGBEE UTILIZANDO A
LINGUAGEM VERILOG-AMS**

Monografia submetida ao curso de graduação em Engenharia Eletrônica da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica.

Orientador: Prof. Dr. Wellington A. D. Amaral

**Brasília, DF
2014**

CIP – Catalogação Internacional da Publicação*

Campos, Regina Silva.

Modelagem de uma transceptor Zigbee / Regina Silva
Campos. Brasília: UnB, 2013. 103 p. : il. ; 29,5 cm.

Monografia (Graduação) – Universidade de Brasília
Faculdade do Gama, Brasília, 2014. Orientação: Regina Silva
Campos.

1. Transceptor. 2. Radiofrequência. 3. Protocolo Zigbee I.
Avelino do Amaral, Wellington.
. II. Dr.

CDU Classificação



MODELAGEM DE TRANSEPTOR ZIGBEE UTILIZANDO LINGUAGEM VERILOG-AMS

Regina Silva Campos

Monografia submetida como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica da Faculdade UnB Gama - FGA, da Universidade de Brasília, em 16/06/2014 apresentada e aprovada pela banca examinadora abaixo assinada:

Prof. Dr.: Wellington Avelino do Amaral, UnB/ FGA
Orientador

Prof. Dr.: Gilmar Silva Beserra, UnB/ FGA
Membro Convidado

Prof. Dr.: Sandro Augusto Pavlik Haddad, UnB/ FGA
Membro Convidado

Brasília, DF
2014

Esse trabalho é dedicado a minha família maravilhosa, que está sempre me apoiando de forma incondicional nos bons e maus momentos.

AGRADECIMENTOS

Agradeço ao meu professor orientador Wellington A. D. Amaral pela oportunidade de desenvolver este trabalho nesta área encantadora, pela orientação e compartilhamento de conhecimento.

Um agradecimento em especial a minha família, que sempre esteve ao meu lado nesta difícil jornada me guiando com apoio, amor, carinho, compreensão, força e amizade.

Agradeço também a todos meus amigos e colegas nos momentos de alegria, cumplicidade e diversão e a companhia nas longas noites de estudo.

RESUMO

O projeto descrito neste documento apresenta a proposta de modelagem de um transceptor Zigbee (IEEE 802.15.4) utilizando uma linguagem de descrição de *hardware* em alto nível Verilog-AMS. A ferramenta a ser utilizada será o *software* Cadence sob a metodologia de projetos *Top-Down*. A modelagem do sistema completo possibilita a co-simulação entre os blocos internos que compõem o mesmo, permitindo assim a análise comportamental, a identificação de características e funcionalidades, e a geração de modelos executáveis. Este sistema irá compor uma biblioteca com modelos de protocolos tradicionais em alto nível com o objetivo de dar suporte ao avanço das pesquisas na área de microeletrônica na Universidade de Brasília e na captação de projetos comerciais. Na primeira etapa do projeto foi desenvolvida a modelagem do Amplificador de Baixo ruído (LNA) em nível de transistor utilizando como ferramenta o *software Advanced Design System* (ADS) para validar a tecnologia adotada (XFAB 0,35 μ m) e para ser usado no apoio às simulações mistas na validação funcional do sistema transceptor. O projeto deste bloco, em especial, segue uma metodologia particular (*Bottom-Up*) orientada sob especificações reais utilizadas atualmente pela indústria eletrônica. Na segunda parte do projeto foi implementada a modelagem funcional dos principais blocos da seção RF de um transceptor. O pacote de ferramentas Cadence foi utilizado para descrição em nível de programação, compilação, simulação e validação das cadeias de transmissão e recepção.

Palavras-chave: Transceptor, radiofrequência, protocolo Zigbee, Verilog-AMS, *Top-Down*, Amplificador de Baixo Ruído.

ABSTRACT

The project described in this paper shows the proposed model for a Zigbee transceiver (IEEE 802.15.4) using a hardware description language at Verilog-AMS high level. The tool that will be used is the software Cadence under the Top-Down design methodology. The modeling of the entire system enables the co-simulation of internal blocks that compose the same, thereby allowing the behavioral analysis, the features and functionality, and the generation of executable models. This system will assemble a library of traditional models of high-level protocols in order to support the advancement of research in the area of microelectronics at the University of Brasilia and in attracting commercial projects. In the first stage of the project was developed to model the Low Noise Amplifier (LNA) at the transistor level using the Advanced Design System (ADS) software to validate the technology adopted (XFAB 0,35 μm) and to be used in supporting mixed simulations in the functional validation of the transceiver system. The design of this block, in particular, follows a particular methodology (Bottom-Up) driven under actual specifications currently used by the electronics industry. In the second part of the project functional modeling of the main blocks of a RF transceiver section was implemented. The Cadence tools package was used for description-level programming, compilation, simulation and validation of chains of transmission and reception.

Keywords: Transceiver, radio-frequency, Zigbee protocol, Verilog-AMS, Top-down, Low Noise Amplifier.

LISTA DE ILUSTRAÇÕES

1.	Componentes de um nó sensor	9
2.	Espaço de operação para padrões 802 WLAN e WPAN	13
3.	Topologias de rede	14
4.	Diferentes camadas do protocolo Zigbee	15
5.	Seleção de canais de comunicação para um ambiente na presença do IEEE 802.11b - <i>Wi-Fi</i> e IEEE 802.15.4 –Zigbee.....	16
6.	Fluxo das etapas de alto nível da síntese lógica.....	18
7.	Arquitetura Verilog-MAS.....	20
8.	Módulo em Verilog_AMS que descreve o comportamento de uma chave.....	23
9.	Arquitetura típica de uma etapa transceptora de um sistema Zigbee.....	27
10.	Arquitetura geral de um LNA com estágios de casamento de impedância.....	29
11.	Diagrama Mixer.....	29
12.	Operação na frequência efetuada pelo <i>Mixer</i>	30
13.	Níveis de abstração	36
14.	Ciclo de um projeto de CI com metodologia <i>top-down</i>	42
15.	Bloco receptor com vários estágios	47
16.	Compromisso dos parâmetros no projeto de um LNA	48
17.	Amplificadores MOS de estagio simples (a) porta comum (b) dreno comum (c) fonte comum	49
18.	Casamento impedância de entrada (a) casamento resistivo (b) casamento por realimentação (c) estagio porta comum	50
19.	Arquitetura degenerativa indutiva por fonte: (a) circuito (b) circuito com indutor na entrada (c) modelo simplificado para cálculo de Z_{in}	53
20.	Estabilização por (a) neutralização e (b) cascode	54
21.	Topologia adotada para o LNA	55
22.	Esquemático do LNA implementado	56

23.	Declaração das variáveis LNA	64
24.	Módulo LNA	65
25.	Definição de bibliotecas e diretiva de tempo	67
26.	Declaração das variáveis do Mixer	67
27.	Módulo do Mixer	68
28.	PGA e dois bits	69
29.	Declaração dos pinos e variáveis locais PGA	71
30.	Controle de ganho do PGA	71
31.	Modulo principal do PGA	72
32.	Parâmetros do VCO	74
33.	Módulo principal do VCO	74
34.	Parâmetros reais do PA	76
35.	Modelo funcional PA	77
36.	Modelo funcional do Filtro	78
37.	Função no matlab para filtro	79
38.	Testbench LNA	80
39.	Simulação LNA	81
40.	Testbench do mixer	82
41.	Simulação Mixer.....	83
42.	Testbench PGA	84
43.	Simulação PGA	85
44.	Simulação PGA com todos os estágios de ganho	86
45.	Testbench VCO	87
46.	Simulação VCO para mínima e máxima tensão de controle	88
47.	Variação da frequência devido à variação do sinal de controle	89
48.	Testbench do PA	90
49.	Simulação PA	90

50.	Testbench filtro passa-baixas	91
51.	Simulação Filtro passa baixa	92
52.	Simulação do test-bench off-chip.....	93
53.	Casamento feito na entrada	94
54.	Casamento feito na saída	95
55.	Carta de Smith	96
56.	Ganho do amplificador	97
57.	Representação da figura de ruído	97
58.	Componentes do símbolo de um sistema de recepção	98
59.	Testbench de um sistema de recepção	99
60.	Resposta entrada da seção e saída do LNA	100
61.	Resposta das saídas do LNA, VCO e Mixer	101
62.	Resposta da saída mixer e da saída filtro	102
63.	Resposta da saída filtro e da saída PGA	102
64.	Componentes do símbolo de um sistema transceptor	104
65.	<i>Testbench</i> de um sistema de transmissão	104
66.	Resposta da entrada da seção e saída do filtro	105
67.	Resposta do VCO	106
68.	Resposta do Mixer	107
69.	Resposta saída Mixer e saída PGA	108
70.	Resposta saída da cadeia de transmissão	109
71.	Esquemático elétrico topologia Sallen-Key	110
72.	Função no Matlab para filtro de ordem 2	110
73.	Esquemático elétrico do filtro no software Cadence	111
74.	Resultado da simulação mista do sistema transceptor completo	112

LISTA DE TABELAS

1.	Comparação de tecnologias sem fios	12
2.	Especificações do projeto	56
3.	Parâmetros obtidos nos cálculos	61
4.	Especificações para o LNA	63
5.	Pinos do LNA	64
6.	Especificações para o Mixer	66
7.	Pinos do Mixer	66
8.	Especificações para o PGA	70
9.	Pinos do PGA	70
10.	Especificações para o VCO	73
11.	Pinos do VCO	73
12.	Especificações para o PA	76
13.	Pinos do PA	76
14.	Especificações para o Filtro	78
15.	Pinos do filtro	78
16.	Bits de controle x ganho	86

SUMÁRIO

1 INTRODUÇÃO	ix
1.1 CONTEXTUALIZAÇÃO	1
1.2 OBJETIVOS	3
1.3 MOTIVAÇÃO	3
1.4 APRESENTAÇÃO DO MANUSCRITO	5
2 REVISÃO BIBLIOGRAFICA	7
2.1 REDES DE SENSORES SEM FIO	7
2.1.1 INTRODUÇÃO	7
2.1.2 PADRÃO ZIGBEE	10
2.1.3 CARACTERÍSTICAS DO PROTOCOLO ZIGBEE	13
2.2 LINGUAGEM DE DESCRIÇÃO DE HARDWARE	16
2.2.1 VISÃO GERAL	16
2.2.2 VERILOG-AMS	19
2.2.3 CARACTERÍSTICAS E APLICAÇÕES DE VERILOG-AMS	21
2.3 TRANSCEPTOR RF	25
2.3.1 CARACTERÍSTICAS	25
2.3.2 COMPONENTES BÁSICOS	26
2.3.3 LNA	28
2.3.4 MIXER	29
2.3.5 OSCILADOR LOCAL	31
2.3.6 CONVERSOR A/D E D/A	32
2.3.7 AMPLIFICADOR DE POTÊNCIA (PA)	32
2.3.8 FILTROS	33
3 METODOLOGIA DE PROJETOS	35
3.1 ABORDAGEM DA METODOLOGIA A SER UTILIZADA	35
3.2 METODOLOGIA TOP-DOWN	39
3.3 METODOLOGIA DA MODELAGEM DO TRANSCEPTOR	45
4 PROJETO	46
4.1 ARQUITETURA DO AMPLIFICADOR DE BAIXO RUÍDO (LNA)	46
4.1.1 TOPOLOGIA LNA	48
4.1.2 PROJETO ELÉTRICO LNA	55
4.2 PROJETO FUNCIONAL DO LNA	62
4.2.1 ESPECIFICAÇÕES	63
4.2.2 MODELAGEM	63
4.3 PROJETO FUNCIONAL DO MIXER	65
4.3.1 ESPECIFICAÇÕES	66
4.3.2 MODELAGEM	66
4.4 PROJETO FUNCIONAL DO PGA	68
4.4.1 ESPECIFICAÇÕES	70

4.4.2	MODELAGEM	70
4.5	DESCRIÇÃO FUNCIONAL DO VCO	72
4.5.1	ESPECIFICAÇÕES	73
4.5.2	MODELAGEM	74
4.6	DESCRIÇÃO FUNCIONAL DO PA	75
4.6.1	ESPECIFICAÇÕES	76
4.6.2	MODELAGEM	76
4.7	DESCRIÇÃO FUNCIONAL DO FILTRO PASSA-BAIXA	77
4.7.1	ESPECIFICAÇÕES	78
4.7.2	MODELAGEM	78
5	SIMULAÇÕES E ANÁLISE DOS RESULTADOS	80
5.1	VALIDAÇÃO BLOCOS INDIVIDUAIS	80
5.1.1	SIMULAÇÃO LNA	80
5.1.2	SIMULAÇÃO MIXER	82
5.1.3	SIMULAÇÃO PGA	83
5.1.4	SIMULAÇÃO VCO	87
5.1.5	SIMULAÇÃO PA	89
5.1.6	SIMULAÇÃO FILTRO	91
5.2	VALIDAÇÃO PROJETO ELÉTRICO LNA	93
5.3	VALIDAÇÃO CADEIA DE RECEPÇÃO	97
5.4	VALIDAÇÃO CADEIA DE TRANSMISSÃO	103
5.5	SIMULAÇÃO MISTA	109
6	CONCLUSÕES	113
6.1	CONCLUSÃO	113
6.2	PROPOSTAS PARA TRABALHOS FUTUROS	115
	REFERENCIAS BIBLIOGRAFICAS	116
	ANEXO A	119
	ANEXO B	120

1. INTRODUÇÃO

Esta seção tem por objetivo explicitar o contexto em que foi concebida a proposta de modelagem do transceptor Zigbee, definir os objetivos e a motivação deste trabalho, bem como mostrar como o texto está estruturado.

1.1. Contextualização

Foi observada nos últimos anos uma grande expansão do mercado de dispositivos eletrônicos de áudio, vídeo, comunicação sem fio, entre outros, nas mais variadas áreas de aplicação. Com circuitos cada vez mais complexos, a indústria microeletrônica se deparou com a necessidade de melhora das etapas de fluxo de projeto a fim de que os circuitos eletrônicos pudessem ser melhor projetados e simulados antes dos chips serem confeccionados. O intuito da melhora dos mecanismos de simulação do circuito se dá com os objetivos tanto de reduzir o tempo entre modelagem e confecção, quanto de minimizar as taxas de falhas e custos, aumentando assim a qualidade e confiabilidade dos dispositivos.

Ferramentas computacionais foram desenvolvidas para dar suporte aos mecanismos de projetos suprimindo suas necessidades na parte de modelagem, desenvolvimento e simulação de componentes eletrônicos. Os simuladores abordam técnicas matemáticas para emular processos ou operações reais com modelos computacionais que sejam capazes de conciliarem as limitações computacionais à realidade. Com a capacidade de formalizar a descrição de sistemas eletrônicos interligando as informações do comportamento, das estruturas, das conexões e das características geométricas, surgiram em meados dos anos 1980, de fato, as linguagens de descrição de hardware (HDL – *Hardware Description Language*). Conforme as necessidades citadas acima foram se sofisticando e tornando cada vez mais complexas, devido ao nível tecnológico que nos encontramos, foi elaborada a linguagem Verilog-AMS que é capaz de simular blocos analógicos e digitais juntos em um mesmo sistema.

Em paralelo, ocorria também a evolução dos dispositivos de comunicação sem fio. A comunicação *wireless* hoje presente no cotidiano de todos, se

expandiu de maneira a necessitar de padrões para garantir a transferência dos mais variados tipos de dados entre os também variados dispositivos de variados fabricantes. Neste contexto, surgiu um dos mais recentes padrões de comunicação, o padrão Zigbee (Zigbee Alliance, 2013) criado com o objetivo de suprir a necessidade de um padrão específico para sistemas de monitoramento e sensoriamento. Os principais requisitos destes sistemas são: baixa transferência de dados, otimização do consumo de energia para prolongar a vida útil de baterias, baixa latência, elevado número de dispositivos por rede e baixa complexidade dos nós. Neste sentido, o padrão Zigbee junto com o padrão IEEE 802.15.4, visam à uniformização da rede, assegurando a confiabilidade e segurança na comunicação a uma frequência de 2.4GHz. As redes Zigbee são comumente utilizadas em automação residencial, sistemas de segurança, sistemas de aquisição de dados, monitoramento florestal, monitoramento de poços de petróleo, monitoramento e controle industrial, e muito mais.

É particularmente adequado para aplicações de biotelemetria, bem como para monitoramento de pacientes em casa. Parâmetros vitais do paciente, como pressão arterial e frequência cardíaca podem ser medidos por meio de dispositivos portáteis. O paciente usa um dispositivo Zigbee que interage com um sensor que reúne as informações, sendo em seguida transmitidos sem fios para um servidor local onde a análise inicial é realizada. Outro exemplo de uma aplicação ZigBee é monitorar a saúde estrutural de grandes construções em grande escala. Nesta aplicação, vários ZigBee habilitando sensores sem fio, como acelerômetros, podem ser instalados em um edifício e todos esses sensores podem formar uma única rede sem fio para coletar as informações que serão utilizadas para avaliar a saúde estrutural da construção e detectar sinais de possíveis danos. Depois de um terremoto, por exemplo, um edifício pode exigir, antes de ser reaberto ao público, uma análise estrutural e os dados recolhidos pelos sensores poderiam ajudar ainda mais a diagnosticar e reduzir o custo de inspeções (Kanna, 2011).

Neste sentido, um sistema de comunicação sem fio destinado ao monitoramento e sensoriamento possui um sistema composto por vários

estágios. Porém, a unidade de interesse do projeto é apenas a unidade de comunicação composta pelos módulos de recepção e transmissão de dados. Ela será modelada em linguagem de alto nível, obtendo-se também os modelos dos sub-módulos do sistema, através da análise funcional dos mesmos.

1.2. OBJETIVOS

Este projeto tem o intuito de realizar a modelagem de um sistema Zigbee (IEEE 802.15.4), utilizando linguagem de alto nível. Este sistema irá compor uma biblioteca de modelos de sistemas comumente utilizados em rádio frequência. Tal biblioteca auxiliará no desenvolvimento de pesquisas na área de microeletrônica e na captação de projetos comerciais. A linguagem de descrição de hardware utilizada será o verilog-AMS, por apoiar a modelagem e simulação de sistemas analógicos e de sinais mistos, podendo auxiliar na especificação de sistemas tanto no estágio inicial do projeto quanto nos estágios finais de simulação de circuitos em nível de transistores. Esta linguagem foi escolhida, também, por permitir sua inclusão na metodologia de projeto do tipo *Top-Down* e por sua total compatibilidade com o fluxo de projeto das ferramentas Cadence.

O projeto inicia-se com o estudo e implementação dos blocos individuais que compõem a cadeia de recepção e a cadeia de transmissão do transceptor, e somente posteriormente segue-se para a implementação do Amplificador de Baixo Ruído (pertencente ao sistema de recepção do transceptor) em nível de transistores utilizando o software *Advanced Design System* (ADS). O projeto deste bloco irá validar tanto o uso da tecnologia adotada para baixo nível (XFAB 0.35 μ m), quanto à metodologia *Top-Down*. Desta maneira, ele será reutilizado no decorrer do projeto nas simulações mistas do sistema completo junto com os demais blocos modelados em alto nível no software *Virtuoso* da Cadence.

1.3. MOTIVAÇÃO

Este trabalho está inserido no âmbito de um projeto atualmente em desenvolvimento na Universidade de Brasília Campus Gama, que visa à

criação de uma biblioteca de alto nível com modelos de protocolos tradicionais tais como Zigbee, Wi-Fi, Bluetooth e assim por diante. A meta principal é dar suporte para o avanço das pesquisas na área de microeletrônica tanto na graduação quanto pós-graduação, e para a prospecção de projetos comerciais, uma vez que um portfólio de modelos pode ser visto como um diferencial na busca de incentivos e investimentos.

Fazer com que a modelagem de um sistema faça parte de um projeto tem como vantagem a extração de informações pertinentes e necessárias a partir de simulações, bem como a possibilidade de estimação de parâmetros de interesse. Isso faz com que o sistema seja caracterizado de forma mais completa, ou seja, a forma com que o modelo é desenvolvido e os componentes resultantes deste permitem saber como o sistema funciona de forma mais minuciosa, simulando-os, poupando tempo e dinheiro (Nascimento, 2010).

A modelagem de sistemas ou blocos oferece uma visão abstrata do projeto em um determinado momento, o que representa certos aspectos da realidade escondendo outros que não são relevantes ou ainda não conhecidos. Abstração é uma técnica poderosa para a concepção de sistemas complexos. Com ela a complexidade é tratada da forma que se interesse para a sua análise, omitindo aspectos desnecessários em determinados momentos, para depois serem trabalhados de forma adequada e em conjunto com o bloco completo. Diferentes quantidades de dados correspondem a diferentes níveis de abstração. Assim, modelos de projetos em cada nível de abstração fornecem a base para aplicação de análise, síntese e verificação técnicas (Nascimento, 2010).

A automação e sensoriamento é um ramo que cresce muito nos dias atuais. Visando tornar os diversos sistemas existentes nas diversas áreas em sistemas inteligentes que poderão ser controlados à distância, consumir menos energia e desempenhar melhor suas funções, vários projetos utilizando redes sem fio vêm sendo desenvolvidos ao redor do mundo. O protocolo Zigbee foi especialmente desenvolvido para se adequar às necessidades destas redes: é

capaz de comportar milhares de nós, oferecendo uma comunicação segura e de baixo consumo. Neste sentido, este trabalho de conclusão de curso irá auxiliar o desenvolvimento de sistemas de comunicação de forma muito mais eficiente, rápida e econômica.

1.4. APRESENTAÇÃO DO MANUSCRITO

Esta dissertação está organizada em cinco capítulos. No capítulo 1 apresentou-se a proposta do trabalho, detalhada nos itens contextualização, motivação e objetivos.

O capítulo 2 é composto por uma revisão bibliográfica que aborda temas que, quando correlacionados, são de fundamental importância para o melhor entendimento do projeto. Primeiramente são apresentadas as tecnologias de comunicação sem fio, dando ênfase às redes dedicadas à baixa transferência de dados e baixo consumo, denominadas redes *Zigbee*. Será feita a análise da arquitetura destas redes, bem como dos dispositivos e modos de operação das mesmas. Posteriormente são abordadas as linguagens de descrição de hardware dando destaque ao Verilog-AMS, por permitir a modelagem de sistemas analógicos e digitais juntos. Por último, será feita a descrição do sistema transceptor RF e de seus blocos internos.

No capítulo 3, é apresentada a metodologia de desenvolvimento do projeto como um todo, que deverá seguir os requisitos da metodologia *top-down*, que é baseada no fluxo de projeto de maior abstração com baixo detalhamento para o de menor abstração com maior detalhamento, por ser compatível com a linguagem Verilog-AMS e com a ferramenta Cadence.

No capítulo 4, aborda-se os conceitos e as características do bloco LNA, onde é feita a análise das topologias existentes e a escolha da que será implementada no projeto. Neste mesmo capítulo é feito o cálculo dos parâmetros do circuito, utilizando especificações da tecnologia adotada. Posteriormente, é feita a modelagem funcional dos principais blocos da seção RF em Verilog-AMS.

O capítulo 5 apresenta e discute os resultados obtidos a partir de simulações realizadas. Primeiramente, será feita a análise do LNA em baixo nível utilizando os parâmetros calculados na seção anterior. Em seguida, será feita a simulação dos blocos em alto nível individualmente, seguido das cadeias de recepção e transmissão e, por fim, do sistema transceptor completo.

Por fim, no capítulo 6, são feitas as conclusões do trabalho como um todo, através da síntese de tudo que foi apresentado e as propostas para a continuidade do projeto.

2. REVISÃO BIBLIOGRÁFICA

Este capítulo trata da abordagem de alguns temas referentes aos elementos necessários para o entendimento básico teórico relacionado ao tema proposto, facilitando as relações entre o problema e o conhecimento existente do mesmo. Os tópicos a serem abordados são: as redes de sensores sem fio com padrão Zigbee/ IEEE 802.15.4, linguagem de descrição de hardware Verilog-AMS e os conceitos e características de transceptores RF.

2.1. REDES DE SENSORES SEM FIO

Esta seção introduz conceitos sobre as tecnologias de comunicação sem fios existentes atualmente, destacando a tecnologia voltada para o sensoriamento sem fio, Zigbee. Serão mencionadas as dificuldades e problemas que culminaram na sua criação, salientando suas características e vantagens perante as demais. É feito um estudo da forma como ocorre o tráfego de informação no sistema de comunicação regido pelas redes Zigbee na frequência de 2.4GHz, em que se revisa os tipos de dispositivos no que diz respeito às funções desempenhadas e aos modos de operação. Finaliza-se com a arquitetura protocolar Zigbee/ IEEE 802.15.4 mediante a análise de suas camadas.

2.1.1. Introdução

Nos últimos anos as tecnologias de comunicação sem fio (wireless) vêm se expandindo muito, graças à sua maturação e estabilidade, tornando-se de fundamental importância para a evolução das tradicionais redes de comunicação cabeadas. Com a conquista de características como confiabilidade e praticidade de instalação, aliadas à redução do preço de componentes, o resultado foi um crescente investimento neste setor, impulsionado pelo aumento da demanda deste tipo de rede, em substituição das convencionais redes.

O desenvolvimento e avanço da tecnologia wireless permitiram o surgimento de várias alternativas e protocolos para viabilizar uma melhor

comunicação tanto na aquisição de dados como no controle e monitoramento para diferentes aplicações embora, a princípio, tenha se enfatizado a transmissão de voz e imagem com altas taxas de transferência. Em uma rede de âmbito maior e mais simples, como por exemplo, a industrial, onde se encontra aplicações com sensores (de temperatura, de umidade, gases, etc) e dispositivos de controle (chaves, relés, etc), ter uma largura de banda elevada para funcionarem não era visto como uma característica necessária. Em contrapartida, a baixa latência, a otimização do consumo energético para preservar a vida útil das baterias, a capacidade de suportar implementações de redes com elevada densidade de dispositivos e a baixa complexidade dos nós eram as principais características neste tipo de aplicação. Assim, o mercado exigia cada vez mais padrões de redes sem fio para diferentes aplicações com diferentes características.

Uma importante área de aplicação das redes *wireless* surgiu da necessidade de sensoriamento, monitoramento e controle de processos, culminando na concepção das redes de sensoriamento sem fio. No geral, o objetivo principal dessas redes é a extração de informação do meio que as rodeiam, tais como: temperatura, umidade, pressão, aceleração, vento, luminosidade, radiação, campos magnéticos e etc. Estas informações são processadas e encaminhadas, na maioria das vezes, por radiofrequência, sendo posteriormente disponibilizadas numa plataforma informática.

O tráfego de informações numa aplicação de monitoramento é feito por meio de nós-sensores constituintes das redes, no geral, formados por módulos de detecção (sensores), módulo de processamento, módulo de comunicação e uma fonte de alimentação, usualmente uma bateria com dimensões reduzidas. Para casos mais complexos, ele ainda pode conter uma unidade de localização, uma unidade de geração de energia e uma unidade de movimento, conforme a Figura 1. Estes nós cooperam entre si, de modo que os dados recolhidos da variável do ambiente chegam ao nó central coordenador que, estando ligado a um computador, resulta no processamento da informação de modo adequado. Os nós sensores, além de interagirem com o ambiente e se

comunicarem com nós vizinhos, também executam um pré-processamento dos dados coletados (Sohraby et al. 2007).

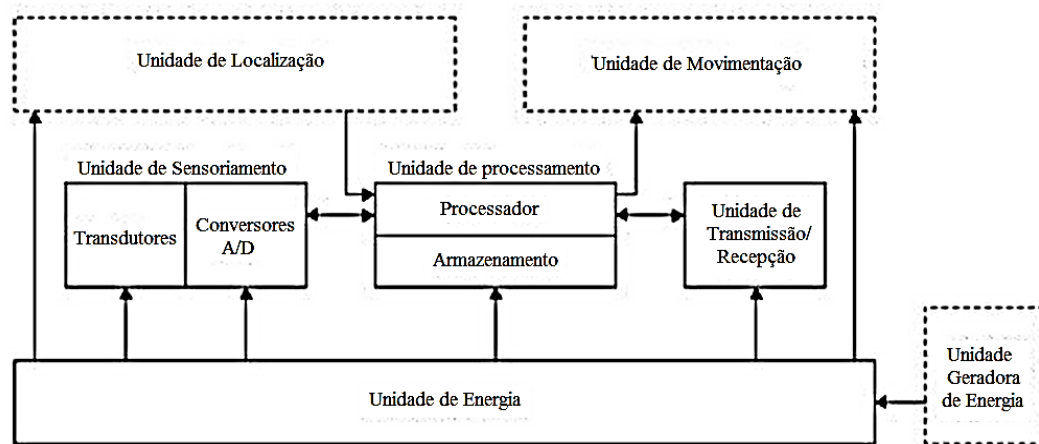


Figura 1: Componentes de um nó-sensor (Tavares, 2002).

A figura 1 mostra os componentes de um nó sensor comum e a interligação entre eles. Dentre todos estes componentes, o trabalho estará focado em apenas um, mais especificamente na unidade de transmissão/recepção dos dados ou transceptor, a qual é responsável pela comunicação do sistema.

As redes de sensores sem fio apresentam características bastante diferentes das demais, como apenas transmissão e recepção de dados, em oposição, por exemplo, às redes pessoais, que necessitam de áudio e vídeo. As áreas de aplicação deste tipo de rede são: militar, industrial, saúde, monitoramento doméstico e monitoramento ambiental.

As soluções wireless disponíveis são *Bluetooth*, *Wi-Fi*, *Wi-max*, as Redes de Sensoriamento Sem Fio, dentre outras. Assim, devido às vantagens que se proporciona, o protocolo Zigbee é utilizado pelas redes de sensores, uma vez que foi criado essencialmente para se adequar às necessidades inerentes destas com intuito de efetuar comunicação sem fio confiável, com baixo consumo energético e baixas taxas de transmissão.

2.1.2. Padrão Zigbee

Um conjunto de protocolos de comunicação é definido pelo padrão Zigbee, especificamente, para redes sem fio de curto alcance e baixa taxa de tráfego de dados. Por volta de 1999, as redes do tipo Zigbee começaram a ser projetadas. Naquela época, todas as aplicações eram desenvolvidas sob a visão *Wi-Fi* e *Bluetooth*. Entretanto, estas começaram a se tornar inviáveis, não respondendo a necessidades específicas tais como gerenciamento de energia, ineficiência de largura de banda, complexidades de protocolos, difícil interoperabilidade de sistemas, ou seja, dificuldades de relacionar os diferentes sistemas proprietários utilizados, etc.

No que diz respeito a redes *wireless Wi-Fi* o objetivo era desenvolver um protocolo seguro com acesso à internet para dispositivos fixos e móveis com elevada taxa de transmissão de dados. Já em relação a redes *Bluetooth*, almejava-se uma rede um pouco menor, com limitação de dispositivos, distância e taxa de transmissão. Assim, caso as aplicações para redes de sensores utilizassem o padrão *Wi-Fi*, estas seriam demasiadamente mais complexas e caras ou extremamente limitadas, caso utilizassem o padrão *Bluetooth*.

Neste sentido, como até então não havia um protocolo que atendesse às necessidades específicas de sensores e dispositivos de controle, os vários fabricantes destes equipamentos adotavam soluções próprias, o que resultava em graves problemas, dentre eles a dificuldade de troca e utilização de dados entre dispositivos de diferentes fabricantes com diferentes configurações.

Em 1997 surgiu a Zigbee Alliance, da associação, inicialmente, de oito empresas de tecnologia e atualmente em conjunto com membros do IEEE (*Institute of Electrical and Electronics Engineers*) e financiada por mais de 150 empresas, com o intuito de criar um padrão para redes sem fio que possua as características de ser um sistema *wireless* altamente confiável e seguro, de ter consumo muito baixo de energia, baixa taxa de transmissão para aplicações de monitoramento e controle, alta relação custo benefício e que seja um padrão de comunicação global aberto para o uso de qualquer pessoa ou instituição

interessada. Trata-se de um padrão voltado para sistemas que utilizem bateria, com baixa taxa de tráfego de dados, baixo custo, baixa complexidade e baixas potências de transmissão (Evangelista, 2010).

O grupo de trabalho IEEE 802 são os responsáveis por definirem as normas de uma rede. Estes classificam quatro grupos de redes de comunicação: WPAN, WLA, WMAN e WWAN (Midorikama, 2001).

WPAN – *Wireless Personal Area Network*: abrangem tecnologias de redes sem fio com pequeno alcance, entre 10 e 100 metros. É projetada para baixo custo e baixa transferência. Está normalmente associada ao Bluetooth, com aplicações de periféricos de PC e telemóveis, PDA's, e ao Zigbee, com aplicações de controle remoto, sensores e dispositivos alimentados por bateria.

WLAN – *Wireless Local Area Network*: se destina a tecnologias sem fio com alcance de 100 e 300 metros, trata-se de uma extensão ou alternativa a redes de cabeamento convencional. Esta, normalmente associada a redes Wi-Fi instaladas em lugares públicos, aeroportos, universidades.

WMAN: *Wireless Metropolitan Area Network*: Possui o alcance em torno de 6km, correspondente a acessos de banda larga em redes de âmbito metropolitana.

WWAN: *Wireless Wide Area Network*: trata-se da rede com amplo alcance, referente a serviços de telecomunicações (voz e dados) em longa distância de transmissão.

Assim, os principais padrões sem fio são *Wi-Fi*, *Bluetooth* e *Zigbee*. O *Wi-Fi*, baseado sobre normas da família IEEE 802.11, é aplicado a redes com altas taxas de transferência, acesso a internet e a conteúdos multimídia, apresentando complexo suporte a pilha protocolar e também alto consumo energético, principalmente em estado de *standby*. Já o Bluetooth e o Zigbee apresentam baixas taxas de transferência, porém com aplicações distintas. O Bluetooth foi projetado para apresentar média taxa de transferência de dados, com limitada quantidade de dispositivos conectados com a remoção de quaisquer cabos existentes na utilização eletrônicos. Já o protocolo Zigbee apresenta um consumo bem mais reduzido quando comparados aos demais,

especialmente nos períodos inativos, sendo nesta fase cerca de 100 vezes inferior ao Bluetooth. Devido à grande economia de energia associada a aceitáveis taxa de transferência e alcance, e a menor complexidade da pilha protocolar, este exige recursos mais simples nos dispositivos que a utilizarem, o que o torna interessante para módulos de sensores. A tabela 1 apresenta uma comparação entre o *Bluetooth* e o *Zigbee*.

Especificação	Camada física	Débito	Duração das baterias	Recusos	Nós	Alcance
Bluetooth	802.15.1	1Mbps	1 a 7 dias	≈250KB	7	1 a 10m
Zigbee	802.15.4	250Kbps	100 a 1000 dias	4 a 32KB	65535	100m

Tabela 1: Comparação de tecnologias sem fios (Malafaya, *cut all*).

O Zigbee opera em três bandas de radio conhecidas denominadas ISM (*Industrial, Scientifical and Medical*), que não requerem licença para uso. Ele atua globalmente na frequência 2.4GHz (nos Estados Unidos a 915MHz e na Europa a 868MHz), sendo que seu espectro é dividido com outras tecnologias como a *Wi-Fi*. O Zigbee possui máxima taxa de transmissão de 250Kbps com 16 canais disponíveis (Estados Unidos com 40Kbps e 10 canais, e Europa com 20Kbps e 1 canal), o que atende bem as aplicações pretendidas. Se tratando de uma comunicação por rádio frequência (RF) um fator importante que merece atenção na transmissão por RF é com relação às perdas de sinal, que variam conforme a distância entre os nós e possíveis obstáculos. Outros fatores com bastante relevância é a sensibilidade do receptor, o ganho e a eficiência da antena (Braga, 2010).

A Figura 2 ilustra o posicionamento dos padrões mediante o mercado de tecnologia wireless, através da relação entre consumo energético e taxa de transmissão.

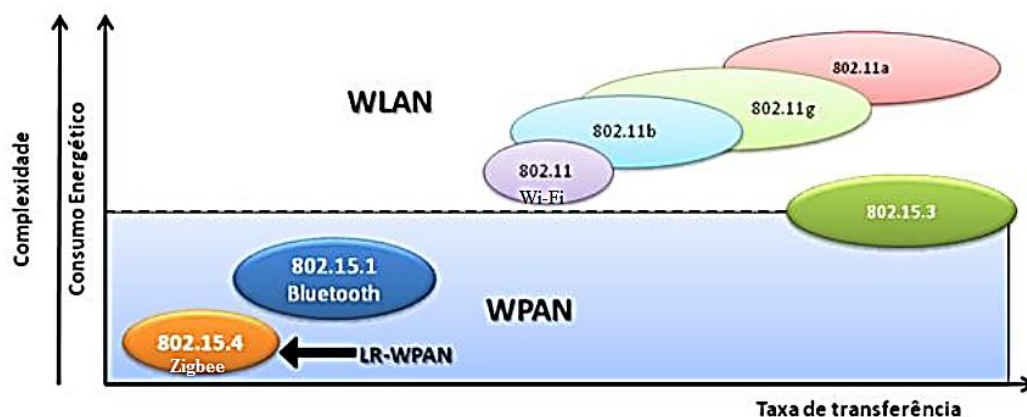


Figura 2: Espaço de operação para padrões 802 WLAN e WPAN. (Braga, 2010).

2.1.3. Características do protocolo Zigbee

De acordo com Ahamed (2005), o Zigbee é um novo padrão com uma vasta área de aplicação que vai desde o controle industrial, telemetria à automação de residências, ele possui determinadas características que o tornam completamente diferente das demais, justificando sua criação.

Baixo consumo de energia (*low power*), taxa de dados de até 250kbps e tamanho de hardware reduzido.

O Zigbee pode operar em dois modos: *beacon* e *non-beacon*. Trata-se de um recurso que pode ser usado pelo coordenador para operar a rede com o uso de uma estrutura chamada *superframe*, que nada mais é que um sinal.

Possui capacidade de suportar uma grande densidade de nós por rede. Sendo um máximo de 65535 dispositivos por cada Zigbee coordenador, enquanto no *Bluetooth* são 8 e no *Wi-Fi* são 30 (Silva, 2007).

Apresenta baixo tempo de ligação á rede, quando comparado aos outros protocolos, maior rapidez na transação do modo *standby* a ativo, e baixa latência.

Contem dois estados de operação sendo o ativo (para envio ou recepção de dados) e o *sleep*, não precisando assim se preocupar com o modo mais adequado (Silva, 2007).

Proporciona segurança e confiabilidade, pois possui recursos de encriptação com a implementação do padrão AES (*Advanced Encryption Standard*) de 128 bits.

Uma rede Zigbee apresenta suporte para duas classes de dispositivos físicos, os quais estão associados à norma IEE 802.15.4, sendo que ambas podem estar presentes numa mesma rede. O *Full Function Device* (FFD) é caracterizado por funcionar como coordenador da rede tendo acesso a todos os outros dispositivos. Já o *Reduced Function Device* (RFD), é caracterizado por poder se comunicar com um coordenador de rede, sendo limitado a uma configuração estrela, não podendo ser coordenador (Silva, 2007).

Esses dois dispositivos anteriores correspondem a três tipos de dispositivos lógicos que concretizam a rede de comunicação, sendo eles: coordenador, roteadores ou (*router*) e os dispositivos terminais (*end-devices*).

De acordo com Farahani (2008), a rede permite flexibilidade suportando diferentes topologias. As aplicações que utilizam como fundamento o IEEE 802.15.4 apresentam rearranjos a fim de que transmissões e recepções de mensagens de dados possam ser feitas. Estas topologias, Figura 3, são baseadas nos três dispositivos lógicos já citados, formando as seguintes topologias: estrela, árvore ou malha.

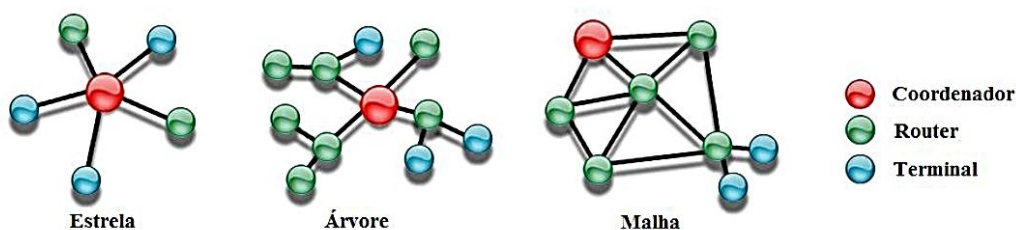


Figura 3: Topologias de rede

Apresentam também uma pilha protocolar de implementação simplificada (sendo menor que um quarto da do Bluetooth), com interfaces de baixo custo (*low cost*).

A pilha protocolar Zigbee, figura 4, é formada parte por camadas definidas pela Norma IEEE, compondo as camadas inferiores formada pela camada

física (PHY) e de acesso ao meio (MAC) e parte por camadas sob ótica do protocolo Zigbee, compondo as camadas superiores sendo formada pela camada de rede (NWK) e camada de aplicação (APL).



Figura 4: Diferentes camadas do protocolo Zigbee (Silva, 2008).

A camada física opera no Brasil na faixa de frequência de 2.4GHz, conforme já foi dito, onde são definidos 16 canais separados por 5MHz dos outros adjacentes. Estes canais presentes no protocolo 802.15.4 não são porções do espectro de radiofrequência. A Figura 5 mostra uma representação desses canais. Há duas observações interessantes. Primeira, os canais são half-duplex, ou seja, o dispositivo escuta o meio ou transmite informação, nunca as duas coisas ao mesmo tempo. Segunda, o dispositivo só pode acessar um canal por vez, ou seja, quando o dispositivo estiver conectado a um canal ele não será capaz de perceber nada do que ocorre nos outros 15 canais (IEEE, 2006).

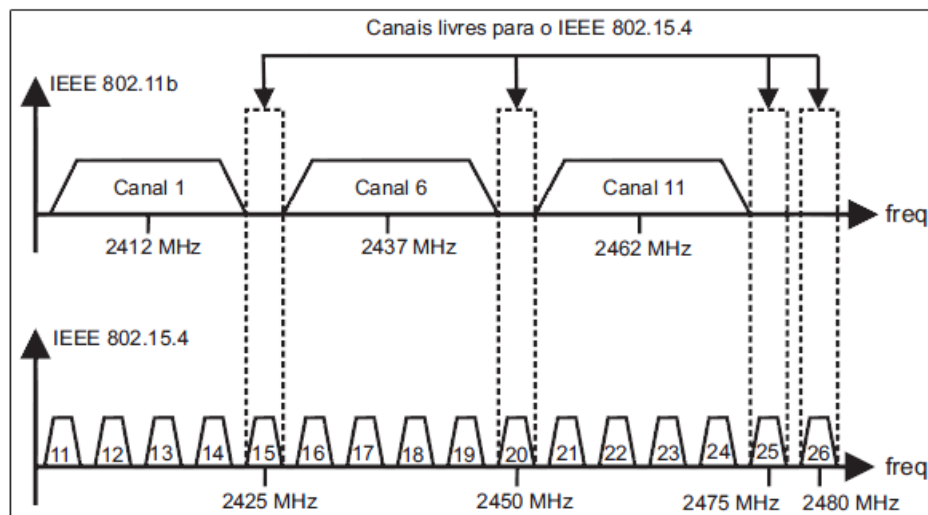


Figura 5: Seleção de canais de comunicação para um ambiente na presença do IEEE 802.11b -*Wi-Fi* e IEEE 802.15.4 –Zigbee. (IEEE, 2006)

A forma de organização dos canais permite a coexistência do padrão IEEE 802.15.4 (Zigbee) com outros padrões, conforme mostra a Figura 5. É possível observar que apesar do IEEE 802.11b (*Wi-Fi*) estar utilizando uma grande faixa de frequência disponível, os canais 15, 20, 25, 26 do IEEE 802.15.4 (Zigbee) ainda podem ser utilizados (IEEE,2006).

2.2. LINGUAGEM DE DESCRIÇÃO DE HARDWARE

Este texto tem como objetivo apresentar de forma bem sucinta as linguagens de descrição de hardware, se aprofundando na linguagem Verilog-AMS. Ressalta-se os benefícios de se implementar projetos mais complexos com esta abordagem, a qual permite uma verificação e descrição de sistemas mistos e a descrição dos mais variados comportamentos. Ela promove um melhor desempenho do projeto com relação ao tempo de verificação e a capacidade de adicionar modelos programáveis.

2.2.1. Visão Geral

As linguagens de programação tradicionais geralmente apenas descrevem algoritmos de procedimentos, o que nada mais é que uma sequência de operações executadas de forma série para os dados na memória ou em elementos periféricos. Diferentemente das linguagens de descrição de

hardware (HDL's), que são utilizadas para projetar *hardware*, bem como para descrever funcionalidades e aplicações. Ou seja, trata-se de linguagem com suporte semântico e sintático para modelagem do comportamento temporal e estrutura espacial do hardware, já que os sistemas de hardware possuem muitos componentes individuais que são interligados para operar simultaneamente.

Uma linguagem de descrição de hardware, especificamente na eletrônica, é uma linguagem referente a computador, especificação ou modelagem para uma caracterização formal e projeto de circuitos eletrônicos, como a lógica digital. Uma HDL é capaz de descrever o funcionamento do circuito, a concepção e organização do mesmo, bem como é capaz, através de simulações, de testar e visualizar seu funcionamento. Trata-se de padrões de expressões baseados em texto, da estrutura espacial, temporal e comportamental de sistemas eletrônicos capazes de representar diretamente equações booleanas, tabelas da verdade e operações complexas como operações aritméticas (Pêcheux, 2005).

Do elevado crescimento tecnológico vivenciado e da presença cada vez maior de produtos eletrônicos no cotidiano das pessoas, verificou-se a necessidade de maior integração dos circuitos eletrônicos (CI's) e maior agilidade no processo de fabricação dos mesmos com uso de metodologias e ferramentas de projeto. Assim a estratégia foi a utilização de modelos compactos padronizados compatíveis com a maioria dos simuladores comerciais, por meio de bibliotecas padrões que permitem a atualização e distribuição à medida que alterações ocorram.

Somente a partir de 1980, com a substituição das linguagens tradicionais como "C" pelas HDL's, é que estas necessidades começaram a ser supridas. Existem nos dias atuais, duas HDL's padronizadas disponíveis: o Verilog e o VHDL, cada qual com suas características específicas. Por se tratarem de padrões já aprovados e publicados pelo IEEE (Instituto dos Engenheiros Elétricos e Eletrônicos), existem inúmeras ferramentas disponíveis no mercado. Assim, para o desenvolvimento de circuitos, podem ser utilizados diferentes

sistemas de desenvolvimento de diferentes fabricantes, já que o produto final das ferramentas deve apresentar características padronizadas (Aisola, 2003).

Em termos gerais, há duas aplicações principais das linguagens de descrição de hardware: a síntese e simulação de projetos eletrônicos. A simulação é feita através de uma ferramenta de automação de projetos eletrônicos (EDA) em que estímulos são aplicados a um modelo executável, descrito em HDL, como sinais de entrada para prever como serão os sinais de saída, ou seja, como ele responderá. Esta aplicação permite entender o comportamento de sistemas complexos sem a necessidade da construção física do circuito. A síntese refere-se ao processo real de implementação do hardware (Figura 6). É definida como a capacidade de traduzir a descrição de uma linguagem de programação de nível mais abstrato, num hardware de nível mais refinado visando uma maior proximidade à implementação automática, ou seja, realiza transformação entre níveis de especificação do sistema até a realização física deste, introduzindo detalhes estruturais e geométricos. Está relacionada com a tecnologia de construção deste sistema que por sua vez é formada por bibliotecas específicas (Moraes, 2008).

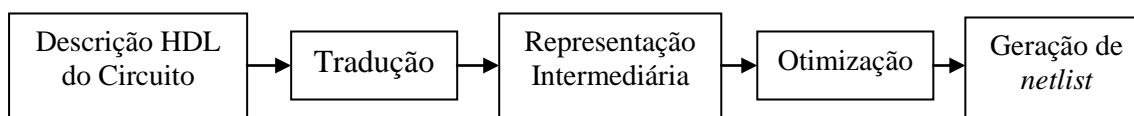


Figura 6: Fluxo das etapas de alto nível da síntese lógica. (Nascimento, 2010)

A eletrônica pode ser diferenciada por seus tipos de sinais sendo eles analógicos, digitais ou mistos. As HDLs mais tradicionais são o VHDL e o Verilog, entretanto estas são utilizadas apenas para modelagem de circuitos digitais. Para a descrição de *hardware* de sinais analógicos ou mistos, considerados mais complexos que os digitais, foram feitas expansões destas linguagens como: VHDL-AMS e Verilog-AMS. Apesar de possuírem características diferentes, elas são ditas concorrentes por atuarem nos mesmos circuitos e na busca dos mesmos resultados, uma vez que ambas são novas linguagens comportamentais e estruturais e também possuem novos

mecanismos de simulação. Entretanto, apenas o Verilog-AMS será foco de estudo deste projeto (Kundert, 2004).

2.2.2. Verilog Ams

Numa visão geral, Midorikawa (2001) entende que:

O Verilog oferece ao projetista os meios para descrever um sistema digital em vários níveis de abstração, e também suporta ferramentas de projeto para síntese lógica. Projetistas de hardware podem expressar suas ideias com construções comportamentais, deixando os detalhes para fases posteriores do projeto. Uma representação abstrata pode ser usada para explorar alternativas arquiteturais através de simulações e para detectar restrições de projeto antes do projeto detalhado. Com o detalhamento do projeto, são criadas descrições com construções comportamentais (MIDORIKAWA, 2001, p. 3).

Logo, o Verilog-AMS (*Analog and Mixed Signal*) é uma linguagem de programação resultada de um esforço da IEEE para estender a linguagem de descrição de hardware Verilog no intuito de apoiar a modelação e simulação de sistemas analógicos e de sinais mistos. Ela permite que projetistas de ambos os sistemas e de circuitos integrados sejam capazes de criar e utilizar os módulos (blocos de hardware com entradas e saídas) que encapsulam as descrições comportamentais de alto nível, bem como descrições estruturais dos sistemas e componentes. O comportamento de cada módulo pode ser descrito matematicamente em termos de suas portas e parâmetros externos aplicados no mesmo, descrevendo o que o módulo faz. Já a descrição estrutural de cada componente pode ser feita em termos de sub-componentes mais simples interligados (Aisola, 2003, v. 2.1, p. 1-1, tradução nossa).

Historicamente, o *VERIfying LOGic* (Verilog) é considerado a primeira linguagem de descrição de hardware, criada por Philip Moorby e Prabhu Goel entre 1983 e 1984 na empresa *Gateway Design*. Em 1990 a Gateway foi comprada pela empresa *Cadence Design Systems*, que a transformou de uma linguagem proprietária para uma de domínio público com a criação da *Open Verilog International* (OVI). Atualmente, o Verilog é um padrão IEEE com algumas extensões feitas, devido à necessidade de melhorias no decorrer dos

anos. O primeiro a ser criado foi o padrão IEEE 1364-1995 em 1995 correspondente ao domínio digital denominado Verilog-HDL. Em 2001 com o padrão IEEE 1364-2001 padronizou a extensão analógica da linguagem conhecida como Verilog-A. E em 2005, com o padrão IEEE-2005, foi feita a última e atual versão com algumas melhorias e inclusão de sinais analógicos e mistos, denominada Verilog-AMS (Moraes, 2008).

Assim como o próprio nome já sugere, “*Mixed Signal*”, a linguagem Verilog-AMS é proveniente da junção de duas outras linguagens: Verilog- HDL e Verilog-A, conforme a Figura 7. Verilog-HDL com domínio puramente digital permite a descrição apenas de componentes digitais. O mesmo acontece com o Verilog-A, que por ser puramente analógico, permite a descrição apenas de componentes analógicos. Por fim, a linguagem Verilog-AMS formada pelas anteriores e adicionada de algumas funcionalidades a mais, permite a descrição de componentes de sinais mistos, oferecendo assim meios de especificar sistemas eletrônicos em vários níveis de abstração. Logo, com estas três linguagens citadas, formam-se a família de linguagem Verilog.

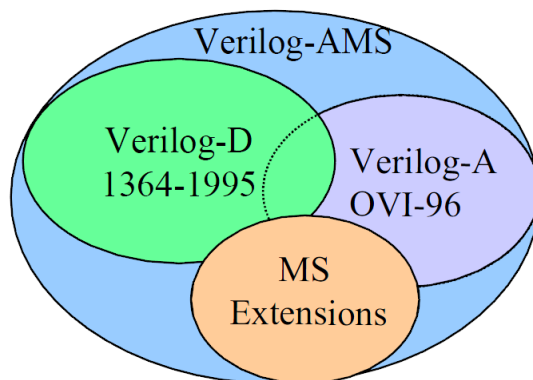


Figura 7: Arquitetura Verilog-AMS. (Accellera, 2003)

Como a linguagem pode especificar sistemas tanto no estágio inicial do projeto quanto nos estágios finais de simulação do circuito, a confecção de projetos com sinais analógicos, digitais e mistos se tornam mais simples devido à capacidade de simplificar da linguagem e disponibilidade em um único simulador.

A padronização da linguagem Verilog-AMS foi feita na intenção de que os desenvolvedores tanto dos sistemas quanto dos circuitos integrados pudessem

utilizar ou implementar módulos descritos em alto nível, tornando possível a simulação dos sistemas analógicos e digitais, a fim de que a funcionalidade de um sistema como um todo pudesse ser verificada.

Com isto, segundo kundert (2004), será mais fácil fornecer um único fluxo de projeto no desenvolvimento de sistemas com sinais mistos, uma vez que o padrão apresenta uma única linguagem e um único simulador que pode ser compartilhado pelos designers analógicos e digitais e pelos designers de blocos ou sistemas, tornando mais simples o trabalho conjunto entre estes profissionais. O autor também reconhece que a principal utilização do Verilog-AMS é para fins de verificação.

2.2.3. Características e Aplicações de Verilog-AMS

A linguagem Verilog-AMS apresenta inúmeras vantagens ao ser utilizada desde sua facilidade por possuir raiz na linguagem de programação C a sua capacidade de simplificar projetos ditos complexos.

Embasando-se nas teorias de Moraes (2008) apud Grabinski e Wakerly e nas teorias de Kundert (2004), esta linguagem apresenta como características o possibilidade dos projetos poderem ser decompostos hierarquicamente, bem como dos componentes de um modelo poderem ser projetados. Aumentando, assim, a capacidade de adicionar modelos a simuladores, por meio de uma grande variedade de características que podem ser usadas para descrever de forma eficiente os mais variados modelos como: componentes básicos (resistores, capacitores etc), componentes semicondutores (TBJ, Mosfets), blocos funcionais e dispositivos (sensores, transdutores).

Esta linguagem também apresenta a capacidade de criar *test benches*, que nada mais é que um substituto do circuito constituído por componentes idealizados. Ele opera todas as funções do circuito ideal com a finalidade de promover teste de forma a proporcionar um ambiente no qual os circuitos possam operar adequadamente, além de possuir a capacidade de descrever os mais variados comportamentos: linear, não linear, integro-diferencial, etc.

Com relação ao tempo de simulação, a linguagem promove sua aceleração, através da substituição de partes não críticas por modelos comportamentais ao invés de um modelo em nível de transistor. Esta, também possibilita a verificação de sistemas de sinais mistos, permitindo a descrição de circuitos digitais e analógicos mais adequados para cada tipo.

Uma característica bastante relevante é a capacidade dos módulos poderem ser construídos em uma HDL e ser invocados por outra e a facilidade para a adequação de processos de projeto *top-down*. Este último é ideal para projeto de sistemas grandes e complexos, permitindo o projeto e verificação primeiro da arquitetura do sistema em um nível mais abstrato para apenas posteriormente iniciar o projeto mais detalhado dos blocos individuais que compõem o sistema conforme a Figura 8.

A estrutura da linguagem permite com que o projetista defina o fluxo dos sinais, o que implica em formulações mais estáveis e robustas. Há também a possibilidade de execução automática da interface entre modelos analógicos e digitais.

A metodologia de projeto utilizada no decorrer deste trabalho de conclusão de curso será do tipo *Top-Down*, na qual o desempenho individual dos blocos, necessário para satisfazer os requisitos globais de desempenho do sistema, será cuidadosamente estudado e compreendido antes de seu desenvolvimento em nível de transistor.

Os códigos descritos em Verilog-AMS seguem um padrão definido, onde no corpo principal encontra-se a definição dos parâmetros de escala de tempo, instância de bibliotecas e a descrição do sistema. Neste último, a seção analógica encontra-se separada da digital. A figura 8 exemplifica a estrutura básica de programação em Verilog-AMS de um módulo que descreve o comportamento de uma chave com entradas analógicas e digitais.

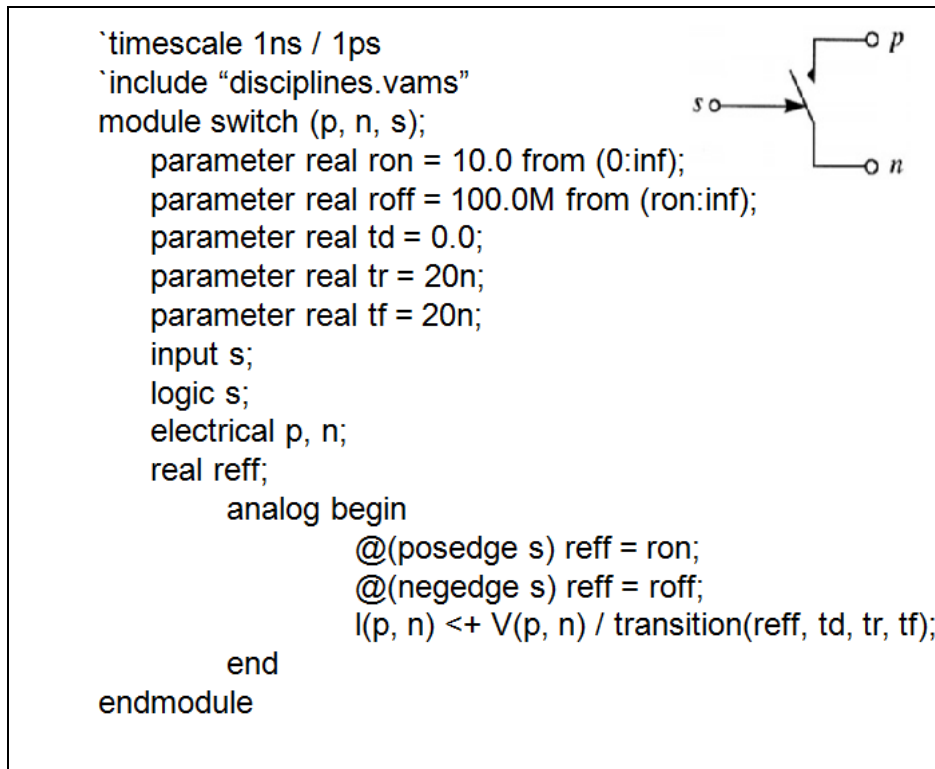


Figura 8: Módulo em Verilog_AMS que descreve o comportamento de uma chave (Kundert, 2004).

A construção Verilog básica é o *module*. Os sinais de entrada e saída de um módulo (sua interface) são designados usando *ports* (pinos, portas). Cada sinal de um *port* pode ser declarado como *input*, *output* ou *inout*. A seguir são apresentados alguns elementos importantes da sintaxe de módulos AMS de acordo com (Kundert, 2004).

- *discipline*: Verilog_AMS é uma linguagem que suporta múltiplas disciplinas. A disciplina é uma coleção de tipos de sinais físicos relacionados, que em Verilog-AMS são referidos como naturezas. Por exemplo, a natureza *electrical* consiste em tensões e correntes, sendo que tensões e correntes são naturezas. Já a natureza *logic* consiste em sinais digitais.
- *nature*: é usada para descrever grandezas físicas básicas. As naturezas contêm três partes de informação: as unidades da grandeza, um nome para quando ela for acessando em num nó, e um número que dá uma indicação do tamanho esperado para a

mesma. Essencialmente, a natureza combina estas três informações em uma única unidade e dá-lhe um nome, tornando-a fácil de incorporá-las em tanto em uma disciplina quanto em um grupo. Por exemplo:

```
nature Current
      units = "A";
      access = I;
      abstol = 1e-12;
endnature
```

- *include*: é uma diretiva de compilação em que arquivos externos, como as bibliotecas padrão, são acessadas como arquivos durante todo o processamento.
- *module*: os módulos são descritores de componentes individuais. Trata-se de um bloco de instruções que começa com o nome *module*, seguido pelo nome dado a componente que se deseja descrever seguida da lista de pinos ou portas finalizados com um ponto e vírgula . Exemplo: *module switch (p,n, s)*. A definição do módulo é terminada como a palavra *endmodule*. Qualquer definição após essa palavra não estará mais associada com esse módulo.
- *parameter*: é atribuído ao identificador um valor constante padrão, que não muda no decorrer da simulação, porém pode ser alterado posteriormente no *testbench*. Ele pode ser real ou inteiro. A sintaxe permite definir um intervalo no qual o parâmetro pode estar contido.
- *ports*: São pontos onde as conexões podem ser feitas. Deve ser definido o tipo e as direções dos terminais.
- A palavra *analog* introduz um processo analógico. Ele é utilizado para descrever um comportamento contínuo no tempo. Já um procedimento digital é sempre introduzido por um comando *always* ou *initial*, uma tarefa (*task*) ou função (*function*). Os comandos dentro de um bloco sequencial (comandos que aparecem entre um *begin* e um *end*) que é parte de um procedimento são executados sequencialmente na ordem em que aparecem. Um bloco sequencial

pode aparecer em um comando *always*, no caso em que o bloco deve ser executado repetidamente. Por outro lado, um comando *initial* especifica um bloco sequencial que é executado apenas uma vez, no início de uma simulação.

- Há um tipo de controle de temporização que atrasa uma atribuição até que um evento específico ocorra. Um controle de evento é especificado por um @. No exemplo da figura 9, @(posedge s) indica um evento referente a uma borda de subida do sinal s.
- transition: esta função é interessante ser usada sempre que possível, uma vez que ela suaviza uma mudança abrupta já que o simulador não trabalha bem com sinais quadrados ideais já que busca trabalhar o mais próximo possível do real.
- Wreal: este é um tipo de sinal que permite transformar um sinal elétrico em uma variável com qualquer valor real. Ela também reduz o tempo de simulação permitindo que blocos analógicos sejam simulados com softwares digitais. Ou seja, ela é do tipo real discreta no tempo. Isso significa que uma variável do tipo wreal apenas assume valores diferentes “gatilhada” por determinados eventos no tempo especificados no modelo.

2.3. TRANSCEPTOR RF

Dentre os componentes que fazem parte de uma rede de sensoriamento ou monitoramento, mostrados na primeira seção, Figura1, o módulo de comunicação da rede, propriamente dito, é a etapa de interesse do presente trabalho. Trata-se do sistema de transmissão/recepção que opera na faixa de radiofrequência (RF) de 2.4GHz. Neste sentido, é feita uma melhor caracterização da arquitetura típica deste dispositivo e de seus componentes básicos, evitando-se o aprofundamento já que não é o foco do trabalho.

2.3.1. Características

Na comunicação sem fio são utilizados sistemas de transmissão e recepção que servem de interface entre o usuário e o meio de comunicação.

Um transceptor é um dispositivo formado pela união destes dois sistemas, os quais utilizam componentes de circuitos tipicamente comuns para que se tenha ambas as funções num só aparelho. Quando contrário, ou seja, os componentes são diferentes, esse aparelho é chamado de transmissor-receptor. A maioria dos blocos presentes em um transceptor, tais como amplificadores, filtros, osciladores e misturadores, fazem parte tanto da etapa transmissora quanto da etapa receptora. Há duas maneiras de se classificar o transceptor de acordo com a capacidade de promover suas funcionalidades. Quando este apresenta a função de recepção e transmissão, simultaneamente, é denominado full-duplex e, em contrapartida se as funcionalidades forem tidas em momentos distintos, é chamada half-duplex.

Crítérios como complexidade, custo, dissipação de potência, taxas de transmissão, frequências de operações, bem como tamanho e número de componentes externos, são de fundamental importância na seleção de arquiteturas de transceptores. No que diz respeito ao bloco receptor, é ele que determina o desempenho geral do sistema de rádio. A escolha da topologia deve ser feita sob a ótica de algumas características básicas como simplicidade, custo, tamanho, alcance e desempenho. As topologias mais utilizadas são homódina, heteródina, low -IF e a super-heteródina, sendo esta última a mais utilizada, entretanto, existem outras que permitem uma escolha mais adequada para cada aplicação. Com relação à transmissão, um transmissor RF executa a modulação, a conversão ascendente, e uma amplificação de potência muitas vezes com a primeira combinação dos blocos. Isto porque, em contraste com as várias abordagens criadas para a recepção de RF, são encontradas apenas algumas poucas arquiteturas de transmissão. O motivo é que questões como ruído, rejeição de interferência e banda seletiva são mais abordadas no transmissor do que no receptor.

2.3.2. Componentes básicos

A figura 9 ilustra um transceptor completo, utilizando circuitos CMOS, para sinais RF. Os blocos básicos do circuito são: amplificador de baixo ruído (LNA – *Low Noise Amplifier*), amplificador de potência (PA – *Power Amplifier*),

oscilador local (LO – *Local Oscillator*), misturador (*Mixer*), filtros, conversor digital/analógico, buffers e antena. É possível observar que o bloco referente à recepção está projetado com a topologia super-heteródino, uma vez que apresenta a região de frequência intermediária (IF – *Intermediate Frequency*) entre a banda base e a região de RF.

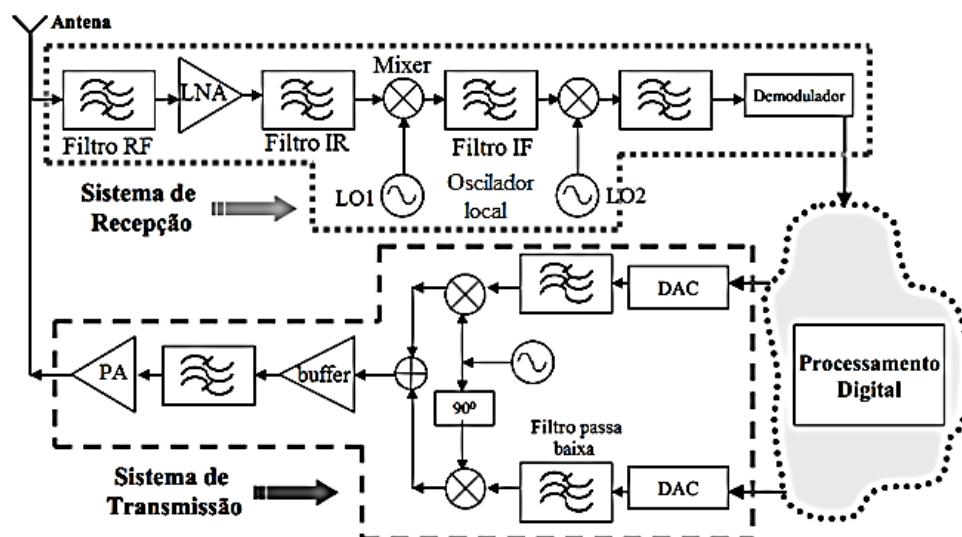


Figura 9: Arquitetura típica de uma etapa transceptora de um sistema Zigbee (Ferreira, 2006).

Basicamente, o sistema de transmissão tem como função modular a informação de banda base numa portadora de frequência elevada, que será posteriormente radiada por uma antena emissora. Assim, de acordo com a figura acima, o sinal de banda base é utilizado para modular um sinal senoidal de frequência intermédia, sinal IF (*Intermediate Frequency*). A modulação é a variação da amplitude, fase ou frequência do sinal senoidal, a qual pode ser analógica ou digital, recebendo assim o nome de portadora. Posteriormente o sinal IF é deslocado para uma frequência superior, frequência de RF, através do *Mixer*. O *Mixer* promove a soma ou diferença das frequências do sinal IF e do sinal do oscilador local (LO). Em seguida o sinal RF é passado por um amplificador de potência (PA), com a intenção de que sua potência seja aumentada para finalmente ser transmitido. Os filtros apresentam a função de selecionar as bandas do sinal desejado, para eliminar sinais que não pertencem à banda de frequência do sistema, ou seja, são utilizados para

eliminar ruídos. São filtros, normalmente, passa-baixa com alto fator de qualidade (Q).

Já o sistema de recepção RF tem como função, basicamente, recuperar os dados transmitidos e isto é feito através do processo inverso da transmissão. O sinal é recebido pela antena de recepção e pode passar ou não por um filtro para a escolha da banda de interesse. Ao chegar ao LNA ele é amplificado, por normalmente ser de baixa amplitude, com a mínima inserção de ruído. Em seguida, há o *Mixer*, que translada o sinal de RF, através de um sinal do oscilador local, para um sinal de frequência mais baixa (*downconversion*), chamado mais uma vez de sinal de IF. Um amplificador de frequência intermédia de alto ganho promove o aumento do nível de potência do sinal de IF, de forma que a informação de banda base possa ser recuperada mais facilmente. O processo de recuperação da informação em banda base é chamado de demodulação.

Como cada tipo de demodulação possui um demodulador específico, nas arquiteturas mais recentes esta etapa, e muitas vezes também a etapa de modulação, tendem a ser feitas no domínio digital, através de circuitos dedicados ao processamento digital de sinal. Dessa forma, por não fazer parte do foco do trabalho e por serem feitas de forma muito mais simples digitalmente, as especificações referentes à demodulação serão omitidas.

2.3.3. LNA

O LNA é normalmente o primeiro bloco do receptor, ele pode ou não ser precedido por um filtro. Pode-se defini-lo como um amplificador eletrônico utilizado para amplificar os sinais possivelmente fracos, provenientes da antena. Ele é o primeiro dispositivo a adicionar um ganho ao sinal, sendo responsável pela primeira inserção de figura de ruído que irá se propagar no restante do sistema, sendo então suas especificações consideradas críticas.

Os parâmetros importantes no projeto de um LNA são: ganho, figura de ruído, não-linearidade e casamento de impedância. Assim, para que ele não interfira de forma tão expressiva no sistema como um todo, busca-se um alto ganho com uma baixa figura de ruído, o que implicará na transmissão de um

ruído baixo ao resto do sistema, não comprometendo a recepção do sinal. A figura 10 apresenta uma arquitetura geral de um LNA, a qual é composta por três estágios, sendo o inicial e o final referentes a estágios de casamento de impedância e o do meio ao LNA propriamente dito (Nascimento, 2010).

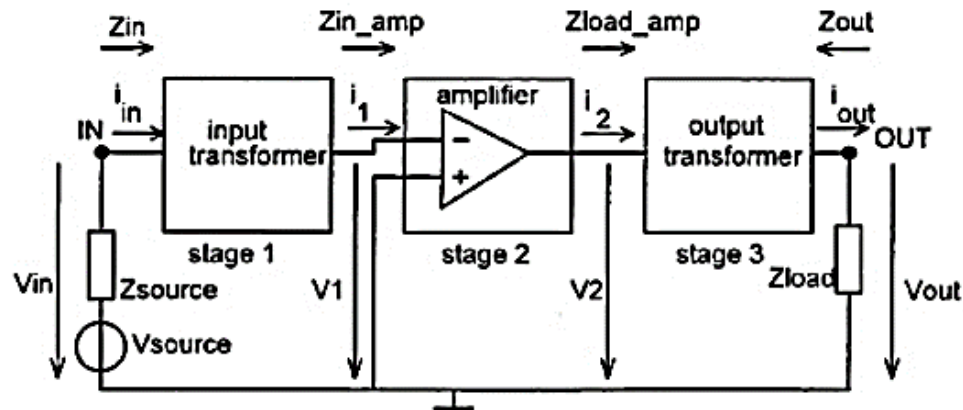


Figura 10: Arquitetura geral de um LNA com estágios de casamento de impedância (Nascimento,2010).

2.3.4. Mixer

O misturador, ou *Mixer*, é também um dispositivo elétrico não-linear que possui a função de criar uma nova frequência a partir de duas frequências de sinais aplicados a suas entradas. Ou seja, o bloco realiza a translação de um sinal de entrada através da operação de multiplicação deste com outro sinal. Isto é feito no sentido de levar o sinal de entrada para uma frequência específica intermédia (IF), em geral uma frequência mais baixa na recepção (*downconversion*) e mais alta na transmissão (*upconversion*).

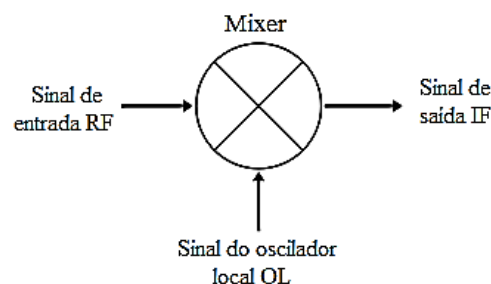


Figura 11: Diagrama *Mixer*.

Este dispositivo apresenta três portas, como pode ser observado na figura 11, sendo duas entradas distintas denominadas: entrada de RF e entrada de oscilador local (LO); e uma saída denominada IF. Na entrada RF é aplicado o sinal a ser transladado e na entrada LO é inserida a forma de onda periódica, pura e limpa, gerada pelo oscilador local. Como resultado, Fig. 12, na saída se pode obter a diferença ($W_{RF} - W_{LO}$) entre as frequências RF e LO, para o caso da recepção, ou a soma delas ($W_{RF} + W_{LO}$), para o caso da transmissão.

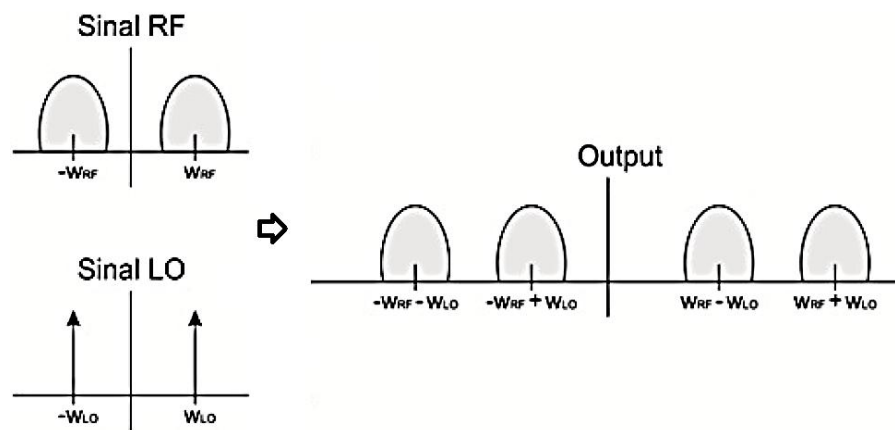


Figura 12: Operação na frequência efetuada pelo *Mixer*.

Logo, no caso da recepção, este procedimento tem por objetivo levar a informação original contida em um sinal de alta frequência para um sinal de frequência inferior. As vantagens oferecidas por sinais de baixa frequência são: maior imunidade a capacitâncias e indutâncias parasitas do circuito, facilidade para serem amplificados (como por AMPOPs), e maior facilidade de demodulação. Neste sentido, um sinal quando chega à antena poderia certamente ser convertido diretamente para baixas frequências e não amplificado primeiramente no LNA. Entretanto isto não ocorre pelo fato destes blocos conversores de frequência serem bastante ruidosos, o que implicaria severamente na sensibilidade do receptor.

É interessante que a faixa de frequência intermédia para qual o sinal RF será convertido seja de baixa frequência, como dito anteriormente, de maneira a satisfazer de forma mais adequada cada aplicação. Porém, a escolha de uma faixa muito baixa, como dezenas de KHz, acarreta num ruído no sinal, injetado pelo *Mixer*, significativamente alto devido ao ruído $1/f$ presente nos

componentes ativos de circuito e também presente de forma mais intensa em transistores integrados MOS (Soares, 2008).

Há duas maneiras de se classificar o *Mixer*: ativo ou passivo. O primeiro caso ocorre quando o sinal na saída do dispositivo apresenta um ganho com relação ao sinal de entrada. Já o segundo caso ocorre quando há apenas a conversão de frequência, sem que nenhum ganho seja dado ao sinal transladado.

2.3.5. Oscilador Local

O oscilador local é um circuito usado para gerar sinal periódico de referência, tanto no transmissor como no receptor, do qual obtém-se a frequência portadora para sintonizar um canal com a finalidade de converter um sinal de interesse para uma frequência diferente, utilizando um *Mixer*. O desempenho de um sistema de processamento de sinal depende das características do oscilador local. Ele deve permanecer numa frequência estável e produzir potência suficiente para conduzir eficazmente as fases subsequentes, como mais uma vez o *Mixer*. Deveria também gerar, idealmente, sinais com baixo ruído de fase e sem variações devido à temperatura ou a tensão de alimentação, já que o sincronismo do sinal é crítico, entretanto, os osciladores nem sempre conseguem alcançar estes objetivos.

Um oscilador local bastante comum é o oscilador LC, também conhecido como tanque, composto por um indutor e um capacitor. A característica deste circuito é que a frequência do sinal produzido é determinada por uma bobina e um capacitor. A escolha da frequência de oscilação é feita através da utilização desses circuitos ressonantes LC como estruturas de realimentação dos osciladores. É interessante fazer a comunicação em altas frequências uma vez que:

$$F = \frac{1}{2\pi\sqrt{LC}} \quad (1)$$

Sendo L a indutância e C a capacitância, quanto maior a frequência de oscilação (F) menor o tamanho do indutor, o que é visto com bons olhos já que o tamanho deste componente é crítico devido a grande área ocupada no chip.

Nos atuais sistemas de comunicação os osciladores locais são controlados por tensão, VCO, isto é, a frequência de oscilação depende do valor de uma tensão de controle. O VCO apresenta também uma sensibilidade de frequência para que o oscilador consiga mudar a frequência de trabalho de acordo com o sinal de entrada. Com relação ao modelo comportamental, o LO é constituído basicamente de um gerador de sinais que pode apresentar uma defasagem de 90° para o componente em quadratura (Zapata, 2007).

2.3.6. Conversor A/D e D/A

O conversor analógico digital é um estágio pertencente ao sistema de recepção de sinais, sua função é quantizar o sinal analógico transformando-o em digital sem alterar seu conteúdo. A importância dele se dá no fato de que em determinadas etapas o processamento de sinais só é feito de forma digital. Agindo de forma totalmente oposta ao citado anterior, está o conversor digital analógico que pertence ao sistema de transmissão de sinais, este gera sinais para excitar e validar o sistema. A etapa de conversão de sinais necessita de uma referência devido à discretização, sendo esta dada pelo *clock*.

2.3.7. Amplificador de potência (PA)

O amplificador de potência é o último estágio no sistema de transmissão de um sinal RF. Seu objetivo é proporcionar potência ao sinal de saída em níveis suficientes para que a transmissão ocorra com bom rendimento energético. É comum haver um estágio de amplificador de tensão precedendo o de potência, no sentido de promover o isolamento (*buffer*) entre a fonte do sinal RF e o PA prevenindo sobrecarga.

Devido aos níveis elevados de tensão e corrente que este estágio apresenta, é possível verificar uma expressiva distorção nesta etapa devido a não-linearidade dos dispositivos e a forma com que os mesmos são dispostos na intenção de maximizar o rendimento. Assim, essa amplificação deve ser

feita levando em consideração critérios que resultarão em figuras de mérito tais como rendimento, capacidade de potência e ganho. Há diferentes técnicas de projeto que podem definir diferentes classes de amplificação ficando a critério a escolha da que melhor se adapta ao projeto.

2.3.8. Filtros

Os filtros são circuitos especialmente projetados para fornecer sinais de saída com amplitude dependente da frequência do sinal aplicado na entrada. Esta definição, apesar de poder ser aplicada a outros circuitos como o amplificador, salienta a principal característica de um filtro: sua sensibilidade à frequência do sinal. Sendo esta característica chamada de seletividade. Assim, os filtros são projetados para permitir a passagem de um sinal com uma faixa específica de frequência e, simultaneamente, rejeitar ou atenuar sinais cujo espectro de frequência esteja fora da faixa de interesse. São quatro os filtros mais comuns, sendo eles: os passa-baixa, que permitem a passagem das baixas frequências e rejeitam as altas frequências; os filtros passa-alta, que permitem a passagem das altas frequências e bloqueiam as baixas; os filtros passa-banda, que permite a passagem de uma banda específica e rejeita todas as frequências fora desta faixa; e os filtros rejeita-banda, que são projetados para rejeita uma banda de frequência particular e permitir a passagem das demais frequências.

A quantidade e os tipos de filtros presentes tanto no sistema de transmissão quanto no de recepção podem variar de acordo com a arquitetura utilizada. São diversas as arquiteturas existentes para implementar estes sistemas, o que representa um comprometimento com o custo do sistema, a dissipação de potência, o tamanho, as taxas de transmissão e recepção, as frequências de operação, etc, por esses motivos é importante ter claro o que se deseja obter do sistema de comunicação.

Os filtros que podem estar presentes no sistema de transmissão são: o filtro passa-baixa, localizado antes do mixer, com a finalidade de eliminar sinais espúrios e harmônicos que surgem no processo de translado da frequência; e o passa-banda para o caso de vários estágios. Com relação à etapa de

recepção, os filtros possíveis são: o filtro RF, que é um passa banda responsável por filtrar o sinal em alta frequência captado pela antena; o filtro IR que também pode ser um passa-banda para rejeição do canal imagem, quando houver; e o filtro IF passa-banda para seleção do canal com frequência intermediária, podendo ser vários se houver vários estágios de frequência intermediária. Caso a arquitetura permita que o sinal de RF seja convertido direto para banda-básica, o problema de frequência imagem é eliminado não necessitando mais do filtro, e os filtros de IF e os estágios subsequentes de conversão com misturadores são substituídos por filtros passa-baixas e possíveis amplificadores de banda básica.

3. METODOLOGIAS DE PROJETO

Este capítulo aborda o estudo das metodologias de fluxo de projeto com o objetivo de se definir a melhor metodologia a ser aplicada no projeto do sistema em questão.

3.1. ABORDAGEM DAS METODOLOGIAS EXISTENTES

A metodologia de projeto, em essência, é um conjunto de abordagens, técnicas e processos utilizados para formular e implementar uma ou mais etapas de um projeto, de maneira sistemática, fazendo uso de conhecimentos objetivos. Alguns exemplos de metodologia são: metodologia de modelagem de um somador utilizando portas lógicas elementares, metodologia de modelagem de uma porta lógica utilizando transistores pMos e nMos, metodologia de verificação de erros em um circuito digital e muitos outros.

A adoção de uma metodologia para confecção de um projeto é extremamente importante para uma perfeita conclusão do mesmo, satisfazendo todos os requisitos especificados, uma vez que, a sua utilização tanto assegura o cumprimento de tais requisitos de maneira formal à medida que o projeto avança como potencializa o uso de ferramentas de automação das tarefas (tal como CAD), e também facilita a comunicação entre as equipes de desenvolvimento. No decorrer de todo fluxo de projeto há três ações ditas fundamentais: especificação e modelagem, validação e síntese.

A especificação e modelagem consistem no processo que inicia com a descrição de uma especificação e termina com a descrição de um modelo. O modelo é o cerne de qualquer processo de design, pois se refere ao desenvolvimento de um sumário de descrições que servem para validar as características de uma parte ou totalidade do sistema concebido, como funcionalidades ou performances. Estes modelos são basicamente modelos de simulação ou modelos executáveis que produzem determinadas respostas quando aplicados alguns estímulos. Os modelos podem descrever o comportamento e/ou a estrutura do sistema concebido em vários níveis de abstração. Assim, ele também demonstra um conjunto de índices de

desempenho como custo, confiabilidade, velocidade, etc, (em nível de equações) e um conjunto de restrições sobre estes índices de desempenho.

A etapa de validação determina se os requisitos estão corretos, completos, precisos e consistentes, assegurando a qualidade do produto. Aplica-se a softwares e hardwares utilizando como ferramentas simulações e verificação formal. Quando ocorre a detecção de um erro ou não conformidade o projeto deve retornar ao processo de modelagem.

Já a síntese corresponde ao processo de se transformar uma especificação mais abstrata em uma menos abstrata. Ou seja, é uma transformação que ocorre na descrição de um sistema de maneira que detalhes são adicionados à implementação, partindo sempre de uma descrição mais abstrata para uma que se aproxima da implementação física.

Os níveis de abstração estabelecem o grau de detalhamento com que um dado sistema é descrito. Quanto maior o nível de abstração menor o nível de detalhamento, gerando uma descrição mais simples o que resulta em uma compreensão também simples. Em contrapartida, quanto menos abstratas mais complexas serão as descrições, resultando em uma compreensão mais difícil com mais detalhamento de informações. De maneira geral há cinco níveis de abstração, adotando uma ordem decrescente de abstração, tem-se: o nível funcional (equações), nível de blocos, nível de portas lógicas, nível de transistores e nível *layout*, conforme ilustrado na figura 13.

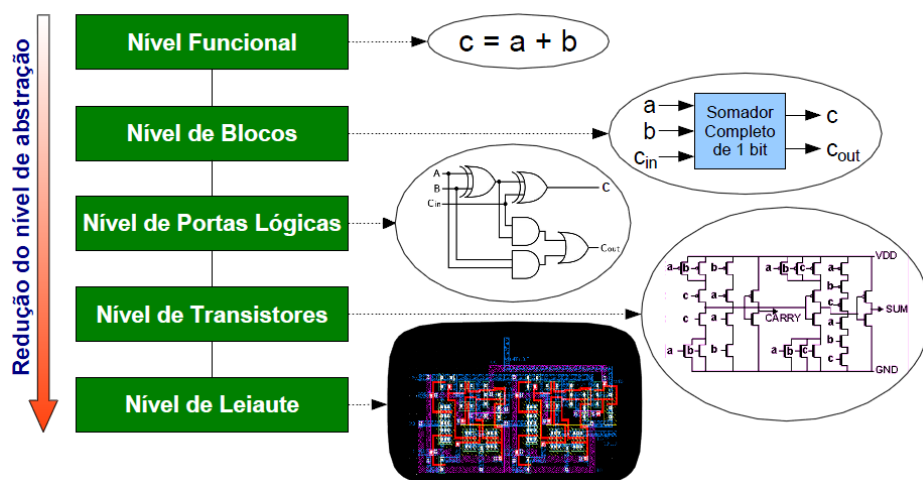


Figura 13: Níveis de abstração. (Zurita, 2013)

O número de níveis de abstração pode variar com o tipo de projeto e com a metodologia adotada, sendo também definido de acordo com as ferramentas de projeto a serem utilizadas. O fluxo de projeto corresponde a uma sequência de passos que vai da especificação do sistema até sua implementação final em silício (confeção do chip propriamente dita). Este fluxo quando completo envolve diferentes metodologias a cada passo, como metodologia de modelagem, metodologia de verificação e outras. Porém, pode-se classificar as metodologias de fluxos existentes em três categorias, conforme a sequência em que se dá a evolução do projeto, sendo elas: *Bottom-up*, *Top-down* e *Middle-out*.

A metodologia *Bottom-up*, também conhecida como ascendente, é a metodologia mais clássica e se inicia com o projeto dos componentes elementares através de um conjunto de especificações, para apenas posteriormente implementar as partes mais complexas até a descrição total do sistema. Trata-se do projeto individual dos blocos antes de se concentrar no projeto do diagrama de bloco do sistema como um todo. Depois do projeto das partes individuais, estes são ligados entre si para formar o sistema, sendo todos em nível de transistores.

Esta metodologia é eficiente apenas para projetos pequenos e mais simples, uma vez que para projetos grandes e complexos a junção dos blocos pode gerar alguns problemas inesperados. Um dos problemas é a verificação completa que se torna inviável já que leva muito tempo, pois todos os blocos estão projetados a nível de transistores. Assim, é realizada uma quantidade muito reduzida de verificações no sistema resultando na não detecção de alguns problemas o que leva consequentemente a uma nova rodada de fabricação. Outro problema está relacionado à arquitetura do sistema que está diretamente ligada ao desempenho do mesmo. Como a eficiência da arquitetura é pouco explorada, os erros arquiteturais quando não detectados são caros e demorados de resolver uma vez que é preciso fazer redesenho dos blocos. Há também o problema da comunicação entre os projetistas. Nesta metodologia a comunicação entre os profissionais é feita de maneira informal e sujeita a erros, já que eles possuem uma visão limitada do sistema. Quando

ocorre um erro de comunicação, geralmente, a consequência é o reprojetado do sistema. Outra desvantagem é a necessidade dos projetistas estarem no mesmo local de trabalho devido ao fato de terem que assegurar a concepção adequada da obra quando os blocos forem unidos.

A metodologia *top-down*, descendente, trabalha do nível mais abstrato para o mais detalhado. Ela se inicia com a formulação geral das características finais do sistema desejado de maneira abstrata, sem detalhamento. Com o avanço do projeto vai ocorrendo o refinamento do sistema, com a decomposição do problema e solução do mesmo em cada nível, gerando assim subsistemas. Ao final, o sistema é descrito em termos de componentes elementares. O uso desta metodologia está relacionada à automação das etapas do projeto. Com ela, os blocos individuais, ou subsistemas, que são necessários ao desempenho geral, são minuciosamente estudados e compreendidos antes de serem desenvolvidos. Esta estratégia minimiza a necessidade de redesenho dos blocos individuais, pois, uma vez constatado que a performance de algum bloco é irrealizável, torna-se necessária uma revisão da topologia adotada para o sistema.

Já a topologia *Middle-out* nada mais é que a mistura da topologia *Top-down* com a *Bottom-up*. O projeto inicia em um nível intermediário de abstração e a partir dele se avança nos dois sentidos: refinando a descrição (*top-down*) e compondo subsistemas maiores a partir dos subsistemas descritos no nível intermediário (*bottom-up*).

Uma tendência mundial observada nas últimas décadas é o avanço na tecnologia de semicondutores correlacionando partes analógicas, digitais, de radiofrequência, ópticas e microssistemas integrados (MEMS). Houve nos últimos anos um aumento exponencial da complexidade dos projetos como: menor área, menor consumo, menor custo, maior velocidade e etc. Segundo Ribas (2000), o interesse de pesquisadores e indústrias com relação aos microssistemas vem aumentando cada vez mais nos últimos tempos, devido à grande quantidade de dispositivos e aplicações potenciais nas mais diversas áreas como: telecomunicações, automobilística, médica, biomédica,

aeronáutica, automação industrial e muitas outras. Circuitos microeletrônicos no início com capacidade de apenas um simples transistor, atualmente possui a capacidade de centenas de milhares de transistor em um único chip. Assim, em decorrência destas complexidades é fundamental a utilização de uma metodologia mais eficiente de projeto de CI's analógicos e mistos. Com isto, a metodologia *Top-down* é recomendada por suportar melhor as diversidades apresentadas.

3.2. METODOLOGIA TOP-DOWN

Nas ultimas décadas foi observado uma evolução muito mais expressiva nos projetos de sinais digitais quando comparados aos de sinais analógicos. Como se tratam de tipos de projetos bastante distintos, as formas com que são abordados também diferem, o que faz com que projetos analógicos sejam mais lentos, trabalhosos e propensos a erros quando comparados ao digital que melhora cada vez mais sua metodologia de design e automação de projeto.

Estes projetos evoluíram desta forma por terem sido expostos à necessidade de implementar sistemas grandes e complexos antes que os analógicos, e por serem mais propensos a automação. Hoje, enquanto os chips complexos digitais possuem dezenas de milhões de transistores e podem ser projetados corretamente na primeira tentativa, os complexos analógicos possuem dezenas de milhares de dispositivos podendo atingir 3 ou 4 redesenhos e levar de um ano a um ano e meio para ficar pronto.

Segundo Kundert (2004), na maioria dos casos os sistemas digitais são implementados com máquinas de estado finito (FSM) e construídos utilizando bibliotecas de células padrão. As FSM possuem a função de unificar, homogeneizar o design e conceber uma fundamentação matemática. O resultado desta forma de implementação são ferramentas de síntese lógica que fazem uma descrição em nível de transferência de registros (RTL), que é uma descrição de alto nível que apresenta uma descrição otimizada do sistema em nível de portas lógicas. Assim, essas ferramentas criam uma descrição matemática nas quais executam transformações no intuito de criar uma concepção otimizada em termos de potência, velocidade e área.

Ainda com base no mesmo autor, a construção de sistemas analógicos é basicamente o oposto da anterior, uma vez que não tem equivalência à RTL e nem às FMS, não tendo, então, formulação unificada e nem fundamentação matemática comum. Também não existem processos de transformação diretos, por equivalência, que permitem que seja facilmente realizada uma modificação na topologia de um circuito sem que exista o risco de que ele venha a não funcionar. De uma maneira geral, seus projetos de design são muito mais lentos e sujeitos a erros, pois são feitos praticamente à mão desde a concepção até a implementação.

Com relação a projetos complexos de sinais mistos, existe uma grande diferença no cronograma entre as partes analógica e digital, e riscos, tornando difícil a combinação dos dois no mesmo chip. Isto gera uma incerteza quanto ao tempo de disponibilidade do produto, devido à demora e aos erros, e também à disponibilidade de projetistas analógicos. Segundo a pesquisa de Roll Collet, em 1998, no *Design Automation Conference*, em que 21 projetos de chips de 14 empresas líderes de semicondutores, o desenvolvimento de circuitos analógicos e de sinal misto exige três a sete vezes mais esforço do que o digital.

Numa analogia a quando os designers digitais estavam tentando projetar sistemas um pouco mais complexos, pode-se comparar a tentativa de projetar sistemas mistos também mais complexos. Os projetos digitais, na época, eram parecidos com os projetos mistos atuais, e precisaram de uma mudança de metodologia devido à incapacidade de avanço dos projetos. Da mesma forma, para que projetos de sinais mistos e sinais analógicos atendam as necessidades e expectativas atuais, é necessário uma metodologia de projeto mais estruturada, que seja mais eficiente e que permita aos designers lidar com o crescimento do tamanho dos circuitos e com a complexidade dos mesmos.

Uma solução para os problemas e desafios, muito procurada pelas equipes de projeto, é a adoção da metodologia *top-down*, por possuir uma abordagem diferente da tradicional *bottom-up*. Ela trabalha de cima para baixo, ou seja, do mais abstrato para o menos. A arquitetura é definida, inicialmente, como um

diagrama de blocos, que é simulado e testado com ferramentas de simulação como Matlab ou Simulink. A partir destas simulações, são retiradas as especificações para os blocos inferiores e individuais, sendo então projetados os circuitos destes. Ao final, o chip como um todo é verificado a respeito dos requisitos e especificações originais. O problema desta metodologia é que existe uma descontinuidade bastante significativa no fluxo do projeto devido à incompatibilidade da representação utilizada na fase inicial, referente à arquitetura, com representação da fase final, referente à implementação. A consequência desta incompatibilidade é a dificuldade de garantia que os blocos trabalharam juntos conforme o esperado, e dificuldade de comunicação, levando muitas vezes ao erro e ao redesenho do projeto.

Um processo de projeto *top-down* bem elaborado metodicamente, procede da arquitetura para design em nível de transistor. A redução do tempo gasto para completar o design se dá através da estratégia de conceber cada nível totalmente antes de passar para o próximo, sendo este reaproveitado totalmente na concepção do próximo. Assim ocorre a divisão do projeto em blocos menores e bem definidos que podem ser distribuídos para mais designers trabalharem em conjunto. Há também a redução de falhas com a melhora da comunicação entre os designers, que passa a ser feita de maneira mais formal. Os profissionais podem estar em diferentes locais que a eficácia desta comunicação não é comprometida.

A Figura 14 apresenta um modelo sequencial de projeto top-down, em que apenas quando cada etapa é completamente realizada é que se passa a diante.

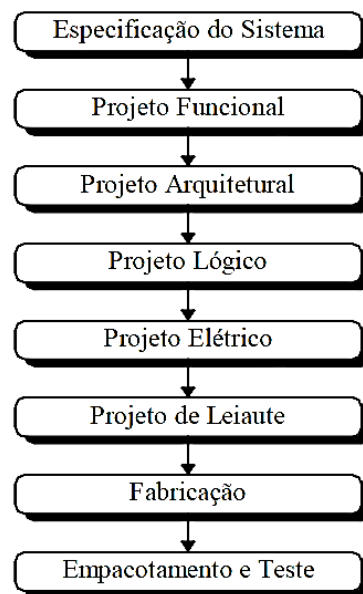


Figura 14: Ciclo de um projeto de CI com metodologia *top-down*.

A representação de um projeto comum compartilhado pode ser usado como auxílio para toda a equipe em todos os tipos de descrições (comportamental, circuito, *layout*). Isto é possível com a utilização de uma representação comum de design simulável, como Verilog-AMS. Com ela, os engenheiros de sistema podem construir uma descrição em nível arquitetônico do projeto, a partir de modelos comportamentais de cada bloco avaliados por cada design do circuito. Assim, exemplos executáveis do que é esperado são passados para os projetistas de circuitos. Como tanto os projetistas de circuito quanto os projetistas de sistema estarão trabalhando com mesmo ambiente, a comunicação é melhorada e potenciais problemas e soluções podem ser tratados em conjunto. Com estes modelos executáveis, a necessidade de especificações que descrevem o comportamento desejado de cada bloco são dispensáveis.

A metodologia *top-down* também permite mudanças parciais de circuitos com o mínimo impacto no final do ciclo de projeto. As infraestruturas colocadas em prática permitem que os modelos sejam atualizados e o impacto sobre o resto do sistema pode ser rapidamente avaliado, uma vez que com Verilog-AMS é possível criar vários co-simuladores tanto para as descrições a nível de sistemas quanto para a nível de circuitos. Com eles, tanto os blocos individuais

quanto conjuntos de blocos podem ser representados em nível de transistor ou em nível de *layout* e ser co-simulados com o restante do sistema, que é descrito com modelos de comportamento de alto nível. Ou seja, blocos modificados podem ser verificados com relação à representação do resto do sistema. Assim, um banco de ensaios para verificação dos blocos pode ser criado com o aproveitamento das simulações do sistema.

Para que o processo de projeto com esta metodologia seja bem concebido, há a etapa de planejamento de verificação. Nela, ocorre a antecipação e prevenção dos problemas durante a montagem dos blocos do sistema. O processo de design é movido para esta etapa o mais cedo possível quando o sistema ainda é descrito em alto nível, para que a verificação seja eficaz. Quando os erros são detectados mais cedo, eles são mais fáceis e menos caros de consertar. Benefícios importantes desta etapa é o fato da verificação permitir que as equipes de design reajam a mudanças tardias nos requisitos de design, e que possíveis áreas problemáticas possam receber mais atenção da equipe.

Como problemas imprevistos geram risco de iteração de projetos, estes devem ser reduzidos com a utilização de medidas para expor possíveis problemas no início. Essas medidas são formuladas através de estimativas da representação de design geradas por informações confiáveis. Essas informações são obtidas começando com abstrações de alto nível e depois com o refino destas conforme o progresso do projeto. Assim, através dos refinamentos é possível chegar a estimativas exatas da arquitetura desejada, levando a compreensão de como os blocos deverão ser implementados. Com a simulação dos mesmos, potenciais problemas podem ser detectados antes dos blocos estarem disponíveis, o que pode resultar em alterações precoces na arquitetura, nas especificações, ou na melhoria do plano de verificação. Como essas alterações ocorrem cedo, a quantidade de redesenhos no processo de design é bastante reduzida.

Com a intenção de evitar erros no projeto devido a falhas de comunicação, as especificações e os planos devem se manifestar como modelos executáveis

e scripts, na medida do possível, ao invés de documentos escritos, uma vez que são usados no processo de projeto numa frequência diária. Como os modelos executáveis e scripts são muito específicos, eles eliminam a ambiguidade dos documentos escritos que podem gerar equívocos que levam à reformulação do projeto. A utilização deles melhora expressivamente tanto o processo de design para a versão inicial do chip como permite a reutilização não só dos blocos como do projeto como um todo, utilizados na construção do chip. Assim numa visão geral, os benefícios do projeto *top-down* são mostrados a seguir.

- Ele proporciona a melhora da comunicação entre os engenheiros, com modelos de alto nível do sistema compartilhado para verificação de todos os processos, e a utilização de especificações e planos executáveis, assim como faz com que a produtividade seja melhorada, devido à eliminação de erros e redesenhos de projeto e devido à maior quantidade de designers trabalhando juntos, uma vez que aos sistemas são bem definidos.
- Essa metodologia proporciona a capacidade de lidar com projetos complexos, devido à melhor exploração do sistema e maior compreensão do projeto, além do uso de simulação de níveis mistos. Ele também possibilita a execução paralela de tarefas de design. Com a existência de um modelo de alto nível compartilhado executável, o desenvolvimento do programa de teste permite que possam ser feito em paralelo o desenho e montagem de blocos.

Dadas as altas pressões que os designers atuais são submetidos e a pouca formação disponível nos centros de educação continuada, é difícil a transição para esta metodologia. Uma série de princípios devem ser seguidos para que todos os benefícios desta rigorosa metodologia de projeto, *top-down*, sejam atingidos, tendo ela o Verilog-AMS como seu alicerce de construção.

3.3. METODOLOGIA DA MODELAGEM DO TRANSCEPTOR

Conforme já foi visto, a linguagem HDL é utilizada para descrição de *hardware*, podendo ser aplicada tanto para simulação quanto para síntese. Como a síntese de circuitos analógicos ainda não chegou a um estágio satisfatório, então os projetos analógicos e de RF ainda são muito dependentes do projeto manual. Porém, à medida que o circuito analógico fica mais complexo, aumenta-se a dependência pelo HDL para estar modelando determinadas situações mesmo em circuitos de RF e analógicos.

Neste sentido, a linguagem Verilog-AMS se torna interessante por modelar circuitos mistos, podendo no mesmo bloco ser utilizada a linguagem HDL puramente digital e a linguagem Verilog-A que é analógica, sendo então a mistura destas duas últimas linguagens. Trata-se no geral, de uma linguagem utilizada apenas para verificação, sendo, portanto uma ferramenta que auxilia a verificar a consistência do sistema diante das especificações de aplicação, buscando a otimização do mesmo através de simulações mistas.

Esta linguagem é compatível com a metodologia *top-down* e se encaixa perfeitamente com o fluxo da CADENCE. O objetivo do projeto é modelar o sistema transceptor Zigbee com a linguagem Verilog-AMS, seguindo do nível mais abstrato para o menos. Assim, muitos dos *test-benches*, que serão utilizados para simular o circuito em alto nível, serão posteriormente reutilizados perto do *tape-out* para testar o circuito completo em nível de transistores, otimizando, assim, o tempo com o reaproveitamento de trabalhos feitos no início.

Para aplicar tudo que foi exposto, foi feita a implementação de um dos blocos do sistema (em nível de transistores) utilizando o software *Advanced Desing System* (ADS), com o intuito de validar a tecnologia utilizada. O bloco escolhido para ser projetado foi o Amplificador de Baixo Ruído (LNA). Embora sua implementação em nível de transistores na fase inicial do projeto pareça vir de encontro com os princípios da metodologia *top-down*, ela é justificada por permitir a possibilidade da utilização do circuito em simulações mistas, demonstrando assim a aplicabilidade e importância de tal metodologia.

4. PROJETO

Este capítulo tem por objetivo apresentar a modelagem funcional em alto nível dos principais blocos da seção RF de um transceptor, e a modelagem comportamental em baixo nível do LNA. A modelagem comportamental do LNA será feita para a simulação mista do sistema completo. Para o projeto do mesmo, utilizou-se especificações de LNA Zigbee encontradas nas tecnologias atuais. O capítulo inicia-se com uma descrição mais profunda do LNA, seguido da apresentação das topologias existentes, mostrando as vantagens e desvantagens de cada uma no intuito de chegar à topologia utilizada, o mesmo ocorre com os tipos de casamento de impedância. Posteriormente são feitos os cálculos para o modelo elétrico. Por último é feita a modelagem funcional dos demais blocos, utilizando o Verilog-AMS, bem como o levantamento das especificações dos mesmos para Zigbee. Tanto a cadeia completa de recepção quanto a cadeia de transmissão serão do tipo diferencial. As principais vantagens do par diferencial é que o circuito se torna mais robusto, mais imune ao ruído externo, com tensão de modo comum e maior excursão de saída.

4.1. ARQUITETURA DO AMPLIFICADOR DE BAIXO RUÍDO (LNA)

O amplificador de baixo ruído é, normalmente, nas arquiteturas de transceptor, o primeiro bloco do sistema de recepção de radio frequência. A função do LNA é receber o sinal de radio proveniente da antena e amplificá-lo para que o sinal seja fornecido ao misturador (*Mixer*). Este bloco deve fornecer um ganho suficiente para superar o ruído dos estágios subsequentes, além de introduzir a menor potência de ruído possível para que assim a figura de ruído do sistema como um todo seja minimizada, o que é o mesmo que maximiza a sensibilidade do receptor. A formula de Friis para o ruído é dada da seguinte forma, para o bloco de recepção representado na Fig. 15:

$$F = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 G_2} + \frac{F_4 - 1}{G_1 G_2 G_3} + \dots \quad (2)$$

$$NF = 10 * \text{Log}(F) \quad (3)$$

Onde F_i representa o ruído de cada estágio, G_i é o ganho de cada estágio e F é o ruído total do sistema. Já NF representa a figura de ruído do sistema total.

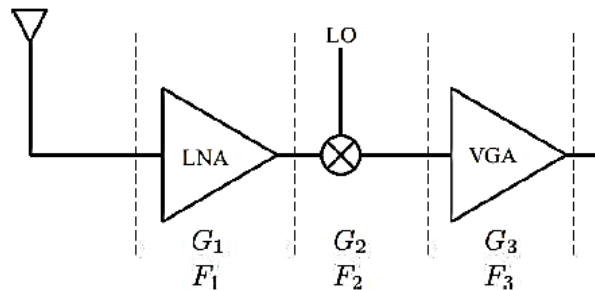


Figura 15: Bloco receptor com vários estágios(Razavi, 1998).

O LNA deve garantir, ao ser projetado, um ganho suficientemente alto na faixa de interesse, sem degradar a relação sinal-ruído (SNR) e deve também ser capaz de acomodar grandes sinais com baixa distorção e baixo consumo de potência na saída do LNA já que normalmente ele vem após a antena ou filtro e antes do *Mixer*. Para que um bom isolamento do sinal reverso seja feito e a máxima transferência de potencia do sinal seja alcançada, o casamento deve ser feito, tipicamente, com um valor de 50Ω .

Logo, de um modo geral, o LNA requer um compromisso entre: bom ganho, baixa figura de ruído, alta linearidade, casamento de impedância na entrada e saída, isolamento do sinal reverso e baixo consumo de potência. O fato do LNA geralmente possuir poucos componentes trás a falsa aparência de simplicidade no projeto, o que não é verdade tendo em vista a necessidade de conciliação dos requisitos citados e o compromisso com os parâmetros de uma especificação.

A indústria de circuitos integrados para radiofrequência está em contínua evolução na busca de processos de fabricação que possam conciliar todos os requisitos e ainda obter um baixo custo, já que o projeto do LNA em CMOS necessita de uma boa caracterização dos dispositivos para circuitos analógicos, que por sua vez é ineficiente numa tecnologia CMOS padrão (Fuentes, 2003). Ainda segundo o mesmo autor, esta caracterização esta

relacionada com a precisão, que por sua vez está diretamente relacionada a uma boa caracterização física dos efeitos que governam os dispositivos RF. A Figura 16 mostra um diagrama com os principais parâmetros para especificação de projeto e a relação entre eles na busca das características mais adequadas.

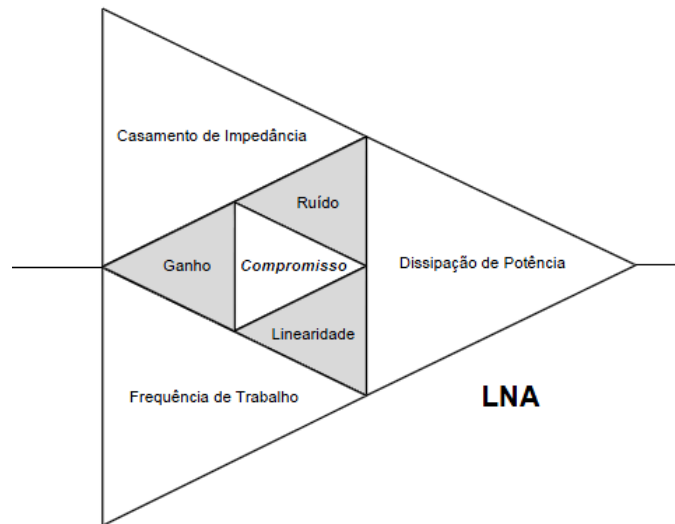


Figura 16: Compromisso dos parâmetros no projeto de um LNA.

4.1.1. Topologia LNA

Conforme visto, um dos pontos fundamentais em sistemas de comunicação é o casamento de impedância envolvendo o LNA, objetivando sempre a máxima transferência de potência entre os dispositivos. Ainda com relação à potência, deseja-se também o máximo ganho de potência e a minimização da potência de ruído introduzida no circuito. Assim, com a chegada do sinal ao circuito receptor, torna-se desnecessário manter o sinal em altos níveis de potencia devido ao grande consumo de cada bloco que compõem o receptor, e em contra partida, é interessante mantê-lo em elevados níveis de tensão para que seja garantido um pequeno ruído introduzido pelo sistema, com benefício de redução substancial de consumo com menores dissipações de potência.

O projeto de LNA está intimamente ligado à figura de ruído global do sistema, apesar das técnicas existentes para a redução do ruído somado ao sinal, o ruído intrínseco presente nos componentes eletrônicos não podem ser eliminados devido a sua natureza física. Neste sentido, a solução encontrada foi à utilização do menor numero possível de elementos ativos e passivos ruidosos no caminho do sinal, tais como resistores e transistores. Sendo assim, é possível projetá-lo baseando-se nas topologias comumente usadas em projetos de amplificadores MOS de estágio simples, já que eles apresentam o mínimo de elementos ativos (transistores), conforme mostrado na figura 17, as topologias são: fonte comum, porta comum e dreno comum.

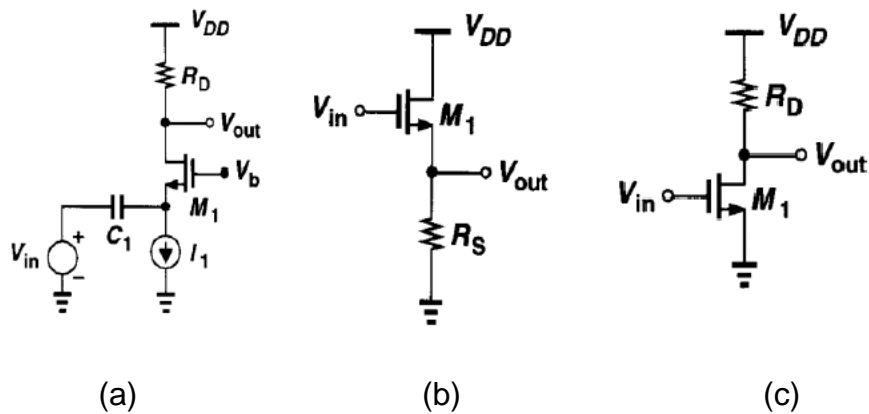


Figura 17: Amplificadores MOS de estágio simples (a) porta comum (b) dreno comum (c) fonte comum. Fonte: Soares, 2008.

O amplificador porta comum, figura 17 (a), é utilizado para aplicações que necessitem do ajuste da resistência de entrada para baixos valores, sendo ideal para promover casamento de impedâncias. Porém, para o caso do casamento de impedância do LNA com a antena ele não é adequado uma vez que aumenta a figura de ruído do sistema, já que a impedância disponibilizada é puramente resistiva.

O amplificador dreno comum, figura 17 (b), é utilizado como estágio de saída em amplificadores formados por vários estágios, para que estes apresentem uma baixa resistência de saída gerando assim uma alta corrente sem perda de ganho. Porém essa topologia é tida como inadequada já que o ganho de tensão na sua saída chega no máximo à unidade.

Já o amplificador fonte comum, figura 17 (c), ao contrário das topologias apresentadas anteriormente, apresenta altos ganhos de tensão com uma baixa impedância de entrada puramente capacitiva, o que o torna a topologia mais apropriada e utilizada. Entretanto, ela ainda não está na sua forma ótima, sendo preciso tratar: a natureza capacitiva da impedância de entrada do amplificador e a capacitância parasita entre a porta e o dreno.

Iniciaremos a solução do problema da impedância de entrada através da inclusão de um circuito que promova o casamento de impedância entre a antena e o LNA. Segundo Razavi (1998), considerando-se o LNA como um amplificador de tensão, poderia se esperar que o valor ideal para a impedância de entrada dele fosse infinito. Do ponto de vista do ruído, pode se requerer uma rede de transformação para preceder o LNA de modo a se obter a menor figura de ruído (NF). Já do ponto de vista da potência do sinal, sendo esta a abordagem mais utilizada, pode-se utilizar o casamento conjugado entre a antena e o LNA já que este é projetado para ter uma impedância de entrada de 50Ω . Isto porque o filtro passa-banda seguido da antena é geralmente projetado para ser usado em vários sistemas transceptores e, portanto, deve operar com uma impedância de terminação padrão. Dessa forma, serão apresentadas diversas maneiras de se criar este casamento, Figura 18, enfatizando sempre suas vantagens e desvantagens na busca da melhor arquitetura.

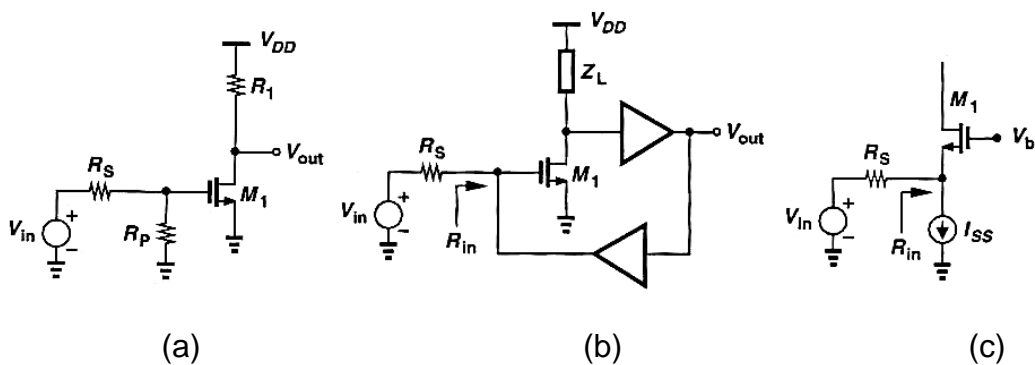


Figura 18: Casamento impedância de entrada (a) casamento resistivo (b) casamento por realimentação (c) estágio porta comum. Fonte: Razavi, 1998.

Na arquitetura da figura 18 (a) um resistor de 50Ω é colocado em paralelo com a entrada, e com o auxílio de um indutor externo, a parte capacitiva da impedância de entrada é cancelada. Porém, essa terminação resistiva gera um ruído levando a uma figura de ruído, neste estágio, que excede 3dB. A solução desejada seria que o circuito tivesse 50Ω de impedância de entrada sem ter o ruído térmico do resistor incorporado (Razavi, 1998).

A figura 18 (b) representa uma arquitetura de realimentação negativa paralela que fornece uma baixa impedância de entrada, podendo gerar uma parte real de 50Ω com design apropriado. Porém, são duas as dificuldades deste caso. Primeiro, o sinal realimentado pode conter ruído substancial, levando a figura de ruído a níveis inaceitáveis. E segundo, a realimentação pode gerar um deslocamento de fase em torno do laço criando instabilidade para certas impedâncias de fonte e carga (Razavi, 1998).

Na arquitetura referente a figura 18 (c) tem-se um estágio porta comum projetado para apresentar impedância de entrada 50Ω , isto é, $1/(g_m + g_{mb}) = 50\Omega$. A capacitância de entrada pode ser anulada por meio de um indutor externo. A principal inconveniência deste método é que a transcondutância do transistor de entrada não pode ser elevada de qualquer maneira, o que impõem um limite ao parâmetro de interesse do projeto (Razavi, 1998).

O fato de se inserir elementos resistivos na entrada do transistor acaba por aumentar a figura de ruído o que degrada a desempenho do bloco. Neste sentido, outro método para criar uma resistência de entrada de 50Ω é adicionar uma degeneração indutiva ao transistor e negligenciar as capacitâncias de porta-dreno e fonte-*bulk*, com ilustrado na figura 19 (a). Sendo assim, a impedância pode ser calculada da seguinte forma (Razavi, 1998):

$$Z_{in} \approx \frac{g_m L_1}{C_{GS}} + L_1 s + \frac{1}{C_{GS} s} \quad (4)$$

Onde,

C_{GS} é a capacitância porta-fonte;

g_m é a transcondutância de pequenos sinais do transistor.

Então, com a escolha adequada dos parâmetros g_m , L_1 e C_{GS} , é possível controlar o valor da parte real de impedância, podendo ser tipicamente 50Ω. Os dois últimos termos, na prática, podem não ecoar na frequência de interesse, necessitando do uso de componentes off-chip na entrada. Isto significa que como a impedância de entrada ainda terá uma componente reativa, predominantemente capacitiva, é necessário adicionar um indutor em serie com a entrada para neutralizar tal componente, de acordo com a figura 19 (b). com isto, uma outra maneira de analisar a impedância de entrada é:

$$I_i = V_{gs} \cdot sC_{gs} \quad \text{para} \quad s = j\omega \quad (5)$$

$$V_{ant} = I_i \cdot sL_g + \frac{I_i}{sC_{GS}} + sL_g I_i + \frac{g_m \cdot I_i}{sC_{GS}} \quad (6)$$

$$V_{ant} = \frac{I_i(g_m L_S)}{C_{GS}} + s(L_g + L_S) + \frac{1}{C_{GS}S} \quad (7)$$

Onde:

V_{ant} é a tensão de saída da antena e vista na entrada do LNA;

C_{GS} é a capacitância porta-fonte;

L_S e L_g são calculados, juntamente com C_{GS} , para estar em ressonância na frequência $\omega = \omega_0$;

g_m é a transcondutância de pequenos sinais do transistor.

Logo,

$$s(L_g + L_S) + \frac{1}{C_{GS}S} = 0 \quad (8)$$

E a impedância de entrada pode ser escrita como:

$$Z_{in} = \frac{g_m L_S}{C_{GS}} \quad Z_{in} \approx L_S \cdot \omega_T \quad (9)$$

Onde ω_T é a frequência de ganho unitário de corrente.

Para o modelo da figura 19 (c), o fator de ruído é dado, de acordo com Lee (2004) é:

$$F \geq 1 + \frac{W_0^2}{W_T^2} \gamma g_{d0} R_S \quad (10)$$

Contudo, conforme esperado, a desempenho do bloco não sofre degradação, o que torna esta arquitetura a mais adequada pra o projeto.

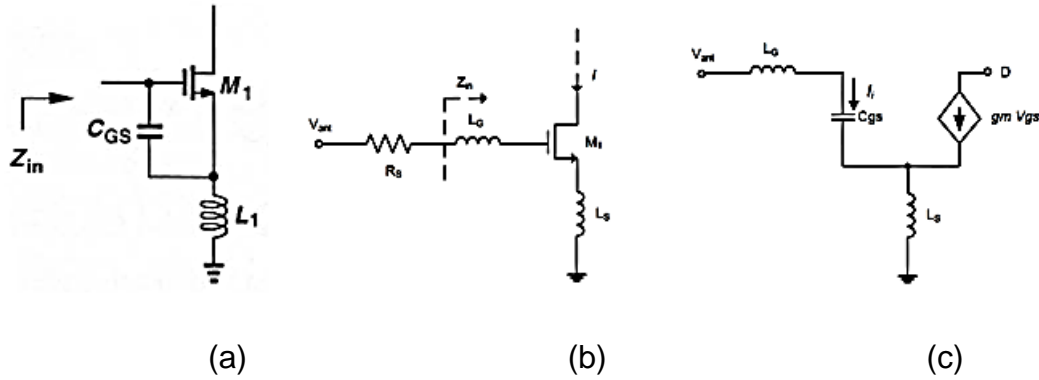


Figura 19: Arquitetura degenerativa indutiva por fonte: (a) circuito (b) circuito com indutor na entrada (c) modelo simplificado para cálculo de Z_{in} .

O segundo problema a ser considerado é com relação à capacitância parasita entre a porta e o dreno do transistor, que até então estava sendo desconsiderada. Neste caso, devido ao efeito Miller, a capacitância parasita, efetivamente vista, está totalmente relacionada ao ganho do amplificador, sendo dada pela capacitância porta-dreno multiplicada por um fator equivalente ao ganho do amplificador. A consequência disto é a deterioração da resposta em frequência do dispositivo. Há também o fato de se colocar a carga conectada diretamente no dreno, que faz com que o nó se torne um terminal de saída tornando o circuito realimentado, o resultado disto é a possibilidade de levar o amplificador à instabilidade.

Há disponíveis na literatura duas soluções para estes efeitos. A primeira consiste no processo de neutralização da capacitância parasita, figura 20 (a), através de um indutor em paralelo. A desvantagem é que neste método L_1 e C_μ ressoam em torno da frequência de interesse apenas por uma banda muito próxima a frequência de ressonância da rede. Além do que, inserir mais um indutor ao projeto compromete ainda mais a área do chip (Soares, 2008). A

segunda solução, mais adequada ao projeto, consiste em adicionar mais um estágio ao amplificador, seria ligar um amplificador base-comum à saída do transistor fonte-comum, conforme figura 20 (b), resultando em um amplificador cascode. A utilização deste estágio é justificada por cominar na eliminação do efeito Miller, no aumento do ganho e na melhora da estabilidade.

No sistema de recepção analisado até então, o bloco do LNA é seguido pelo *Mixer*, o que torna necessário a inserção de um indutor na saída do LNA. Ao escolher um valor adequado para este elemento, é possível garantir que haja ressonância entre a capacitância de entrada do *Mixer* e a indutância de carga do LNA. Podendo assim, ser alcançado altos ganhos de tensão e promover uma filtragem passa-faixas ao sinal de RF recebido.

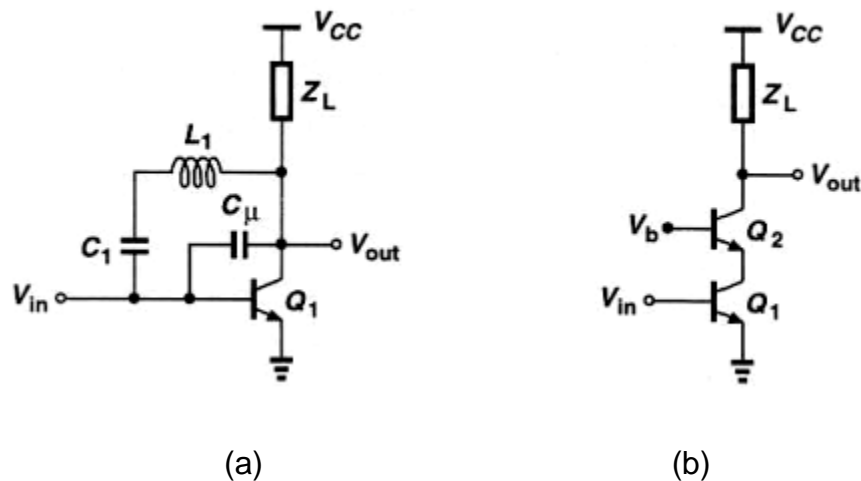


Figura 20: Estabilização por (a) neutralização e (b) cascode

A topologia final considerada ideal para o projeto do LNA é conforme a representada na figura 21. Nela, os indutores LS e L1 fornece o casamento conjugado com a entrada, LD proporciona o ganho de tensão significativo e M2 aumenta o isolamento reverso do LNA desempenhando dois papeis importantes: primeiro, reduz o vazamento LO produzido pelo *Mixer* que vem em seguida, e segundo, melhora a instabilidade por minimizar a realimentação conforme já foi dito.

Como na indústria há um predomínio de circuitos com entrada diferencial, neste projeto acompanharemos esta tendência. Assim, esta mesma topologia *single-ended* da figura 21, pode ser feita de forma diferencial bastando duplicar a estrutura e ligar os dois circuitos pelo terminal inferior do indutor de degeneração L_S . Os circuitos serão ligados a uma fonte de corrente, através do nó comum, e em seguida ao gnd. Essa topologia apresenta pontos fortes com relação à capacidade de rejeição de interferências de modo comum e à imunidades aos parasitas introduzidos pelo encapsulamento. Como ponto fraco, está a necessidade do dobro da área no chip e o consumo do dobro da potencia para um desempenho similar ao *single-ended* (Lee, 2004).

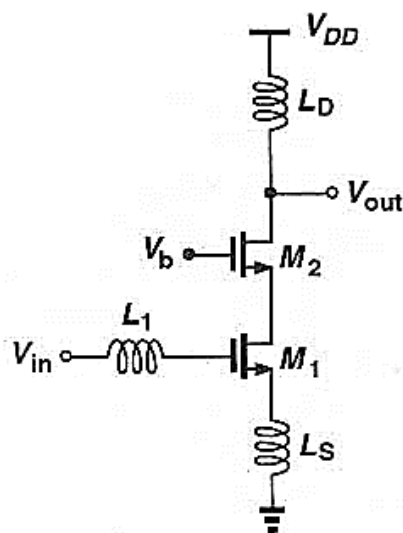


Figura 21: Topologia adotada para o LNA.

4.1.2. Projeto Elétrico LNA

O projeto elétrico do LNA irá utilizar a metodologia *bottom-up* por motivos já explicados no decorrer do texto. A topologia adotada está apresentada na figura 22. Trata-se de um amplificador diferencial com dois estágios, baseado nas discussões anteriores. O primeiro estágio, formado pelos transistores M1, M2, M3 e M4, é um amplificador Cascode diferencial com degeneração indutiva. Esta configuração oferece aumento no ganho do sinal, melhora na estabilidade, além de fornecer um bom isolamento entre a entrada e saída deste estágio, uma vez que o sinal de entrada é aplicado no *Gate* de M1 e M3, e o sinal de saída é apresentado no dreno de M2 e M4. O segundo estágio é

[illegible]

Tabela 2: Especificações do projeto

A implementação do projeto foi fundamentada em algumas especificações básicas utilizadas atualmente, que são apresentadas na tabela 2. Sendo que,

elas estão de acordo com as especificações encontradas na literatura atual, como nos *papers* fornecidos pela IEEE (*Institute of Electrical and Electronics Engineers*). Alguns exemplos são: “A Switchless Zigbee Frontend Transceiver with Matching Component Sharing of LNA and PA”, “CMOS Low-Noise Amplifier Design Optimization Techniques”.

O projeto do LNA foi iniciado com o cálculo dos parâmetros para posteriormente ser simulado no ADS adotando a tecnologia XFAB 0.35 μ m. Para tanto, foi utilizado como referência, Lee (2004), no qual é possível observar a dedução de todas as equações utilizadas, uma vez que isto não será feito do decorrer deste texto.

Os cálculos serão efetuados para apenas um lado do circuito (*single-ended*), onde os resultados encontrados se equivalerão ao outro lado uma vez que o amplificador é diferencial, logo ambos os lados devem ser iguais para que possam ressoar na mesma frequência.

Inicia-se o projeto com o cálculo da largura ótima do *gate* (W_{opt}). Esta largura corresponde ao tamanho do transistor que produz o fator de ruído mínimo para o LNA, portanto este parâmetro será utilizado por todos os transistores do projeto, seu valor é dado pela equação 11.

$$W_{opt} = \frac{1}{3\omega LC_{ox}R_s} \quad (11)$$

Onde,

$$\omega = 2\pi f \quad (12)$$

Os valores de C_{ox} e L são determinados pela tecnologia sendo então de $4,46 \times 10^{-3} \text{ PF}/\mu\text{m}^2$ e $0,35\mu\text{m}$. O parâmetro R_s corresponde a resistência de entrada que foi definida como 50Ω , metade da resistência de entrada total. Além disso, como o sistema deverá operar em uma faixa de frequência de 2,4 – 2,5 GHz, foi utilizada no projeto a frequência central desta faixa, ou seja, 2.45 GHz. Logo, o valor aproximado de ω é $1,54 \times 10^{10} \text{ rad/sec}$. Assim, com base nestes valores o parâmetro W_{opt} encontrado foi de aproximadamente $277,3\mu\text{m}$.

A próxima etapa corresponde ao cálculo da corrente desconsiderando *buffer* de saída, pois ele será utilizado apenas para efeito de teste. Este cálculo pode ser feito utilizando a alimentação e a potência consumida e através da divisão da corrente entre os lados do circuito, de acordo com a equação 13.

$$P_t = \frac{I_D V}{2} \quad (13)$$

$$I_D = \frac{2P_t}{V} \quad (14)$$

Uma vez que tanto a potência consumida quanto a tensão de alimentação são fornecidas nas especificações, o cálculo torna-se simples. O valor da corrente encontrada foi de aproximadamente 1,39 mA para cada lado do circuito. A corrente total do circuito foi utilizada para a polarização do amplificador, fixada por uma fonte de corrente.

Para o cálculo dos indutores é preciso uma estimativa da capacitância parasita C_{gs} do transistor M1, sendo que o seu cálculo é fornecido pela equação 15.

$$C_{gs} = \frac{2}{3} W_{opt} L C_{ox} \quad (15)$$

Como os termos que compõem a equação já são conhecidos, o valor encontrado para C_{gs} foi de 0,29 pF. Foi feito também o cálculo do V_{gs} , equação 16, para que o transistor de entrada permaneça sempre em saturação não comprometendo o desempenho do circuito.

$$V_{gs} = \sqrt{\frac{I_D}{\frac{1}{2} K_N \frac{W_{opt}}{L}}} + V_t \quad (16)$$

Para V_t igual a 0,55 V e K_N igual a $165 \times 10^{-6} A/V^2$, ambos fornecidos pela tecnologia, o valor encontrado para V_{gs} foi de 0,69 V. A transcondutância é uma grandeza que funciona como uma medida da "força" do dispositivo, em que um valor mais alto corresponde a uma maior alteração na corrente de dreno, para uma dada alteração em V_{gs} . Seu calculo é dado pela equação 17.

$$gm = \sqrt{2I_D K_N \frac{W_{opt}}{L}} \quad (17)$$

Fazendo as substituições dos parâmetros o gm encontrado foi de $0.02\Omega^{-1}$. Com isso, a frequência de ganho unitário pode ser facilmente calcula (eq. 18) com a utilização dos parâmetros já encontrados gm e C_{gs} .

$$\omega_t = \frac{gm}{C_{gs}} \quad (18)$$

Portanto, o valor de ω_t é de $69,3 \text{ } G^{rad}/sec$.

De posse destes parâmetros foi possível prosseguir com o design do circuito para a determinação dos indutores. Para o cálculo destes foi preciso analisar a equação 19, correspondente a impedância de entrada do estágio Cascode.

$$Z_{in} = SL + \frac{1}{SC_{gs}} + \omega_t \cdot L_S \quad (19)$$

Analisando seus termos nota-se que a parte real é dada por $\omega_t \cdot L_S$. Logo, este deve ser igual a 50, para que o LNA apresente uma impedância de entrada de 50Ω . Com isto, o valor do indutor presente no source é dado pela equação 20.

$$L_S = \frac{R_S}{\omega_t} \quad (20)$$

Como a frequência de ganho unitário (ω_t) resultante foi de $69,3 \text{ } G^{rad}/sec$ e a resistência foi fixada em 50Ω , L_S é aproximadamente $0,72 \text{ nH}$.

Para que o circuito seja sintonizado na faixa de operação estipulada, a malha de entrada deve ressoar na frequência de interesse, obedecendo a equação 21.

$$\omega = \frac{1}{\sqrt{L_T C_{gs}}} \quad (21)$$

Onde,

$$L_T = L_S + L_g \quad (22)$$

Logo,

$$L_g = \frac{1}{\omega^2 C_{gs}} - L_S \quad (23)$$

Para ω igual a $15,4 \text{ G rad/sec}$, sendo esta a frequência de operação central da faixa de operação do circuito, L_g é igual a 13,8 nH.

O calculo do indutor de saída (L_d) é feito de acordo com a equação 24, a qual faz com que o circuito tanque (LC) na saída ressoe na frequência desejada. Sendo que neste caso, C_d é apenas estimado.

$$\omega = \frac{1}{\sqrt{L_d C_d}} \quad (24)$$

Logo,

$$L_d = \frac{\left(\frac{1}{\omega}\right)^2}{C_{out}} \quad (25)$$

Adotando um pequeno C_d na saída igual a 1pF, o L_d encontrado foi de 4,22 nH. O valor de L_d normalmente é um pouco empírico através de C_d na busca do máximo ganho e menor figura de ruído.

Os indutores utilizados no projeto são considerados ideais porque a modelagem dos indutores integrados seria realizada por um software específico da Cadence e isso não foi possível devido aos problemas apresentados. O fator de qualidade do indutor de saída (Q) está diretamente relacionado ao ganho, neste sentido foi adotado um Q aproximadamente igual a 10, sendo este um valor comumente encontrado em circuitos integrados. Este fator é utilizado para determinar a resistência série ou paralela associada ao indutor de saída. O cálculo é apresentado na equação 26.

$$Q = \frac{\omega L_d}{r_s} \quad (26)$$

Onde,

$$r_s = \frac{\omega L_d}{Q} \quad (27)$$

Assim, fazendo as substituições necessárias, o valor de r_s encontrado foi de $6,5 \Omega$. Para transformar essa resistência série em paralela, basta seguir a equação 28.

$$r_p = r_s(Q^2 + 1) \quad (28)$$

Logo, r_p é dado por 655Ω e será acoplada ao circuito tanque.

De posse destes valores, já seria possível simular o primeiro estágio do amplificador. Porém, para finalizar o circuito por completo foi preciso definir a corrente do estágio dreno comum.

$$r_0 = \frac{1}{g_m} = 50\Omega \quad g_m = \sqrt{2I_{D2}K_N \frac{W_{opt}}{L}} \quad (29)$$

Em que a corrente pode ser definida pela equação 30.

$$I_{D2} = \frac{1}{2 \times 50^2 K_N \frac{W_{opt}}{L}} \quad (30)$$

A corrente de dreno encontrada foi de $1,53 \text{ mA}$, sendo que esta influência diretamente no casamento da impedância de saída. Os valores dos parâmetros obtidos estão dispostos na tabela 3 de forma aproximada, uma vez que em projetos reais são utilizados apenas valores exatos para não gerar problemas no encapsulamento.

Tabela 3: Parâmetros obtidos nos cálculos

Largura ótima dos transistores (W_{opt})	$288 \mu\text{m}$
Corrente do estágio <i>Single-ended</i> (I_D)	$1,5 \text{ mA}$
Transcondutância (g_m)	$0,02 \Omega^{-1}$
Capacitância parasita (C_{gs})	$0,3 \text{ pF}$
Frequência de ganho unitário	70 nrad/sec
Indutor de fonte (L_s)	$0,7 \text{ nH}$

Indutor de gate (Lg)	14 nH
Indutor de dreno (Ld)	4 nH
Resistência série (rs)	6,5Ω
Resistência paralela (rp)	655Ω
Tensão (Vgs)	0,7 V
Corrente do segundo estágio (I _{D2})	1,5 mA

Estes parâmetros serão utilizados posteriormente na confecção do projeto elétrico do bloco.

4.2. PROJETO FUNCIONAL DO LNA

O LNA é um dos primeiros estágios da cadeia de recepção. De uma maneira geral, sabendo que todo circuito gera alguma forma de ruído, a principal função do LNA é amplificar o sinal recebido adicionando o mínimo de ruído possível para que o mesmo possa ser processado adequadamente nos estágios seguintes, conforme já foi visto. Porém, a análise do ruído inserido pelo mesmo deverá ser feita em outro nível de sistema, menos abstrato, o que não corresponde à modelagem funcional aqui utilizada.

O LNA apresenta dois estágios de ganho, sendo estes comandados pelo ganho (G). Este ganho deve ter apenas dois *steps* de valor: para o nível mais baixo (0V) o ganho deverá ser mínimo e para o nível alto (1.8V) o mesmo deverá ser máximo. Ele também contará com um pino *power-down*, assim como os demais blocos, para a comunicação *half-duplex*. Isto significa que quando o transmissor estiver em funcionamento o receptor deverá permanecer desligado e vice-versa. A equação 31 descreve o comportamento funcional do bloco em análise.

$$V_{out} = G * V_{in} \quad (31)$$

Sendo V_{out} a tensão de saída, G o ganho final do bloco e V_{in} a tensão de entrada. Assim, a funcionalidade do LNA é tida como um ganho aplicado ao sinal de entrada.

4.2.1. Especificação

As especificações elétricas para este bloco foram feitas com base em estudos de trabalhos sobre LNA operando em sistemas Zigbee. Sendo assim, para um projeto completo são estabelecidas as especificações contidas na tabela 4, porém, para este projeto foram utilizadas apenas algumas delas como tensão de alimentação, ganho e frequência.

Tabela 4: Especificações para o LNA

Parâmetros	Min	Tipo	Max	Unidade
Tensão de alimentação	1.6	1.8	2.0	V
Temperatura	-40	27	125	$^{\circ}\text{C}$
Alto ganho	13	20	30	dB
Baixo ganho	-	0	-	dB
Figura de ruído	2.3	2	3	dB
IP1dB (100 Ω)	-9	-10	-15	dBm
Frequência	2.4	2,44	2.5	GHz
Zin (diferencial)	-	100	-	Ω
Zout (diferencial)	-	100	-	Ω
S11 (coef. Reflexão na entrada)	-	-	-12	dB
S22 (isolação reversa)	-	-	-12	dB
IIP3	-1	-	-	dBm
Potencia dissipada	-	3.6	5	mW
Corrente	1	3	10	mA

4.2.2. Modelagem

O LNA foi modelado inicialmente de forma ideal para que sua funcionalidade pudesse ser validada. Os parâmetros reais do bloco, referentes ao Zigbee, serão utilizados apenas nas simulações das cadeias de recepção e transmissão, da mesma maneira será com os demais blocos. Os códigos com os modelos completos, tanto do LNA quanto dos demais, se encontram no anexo B.

A relação dos pinos que irão compor o esquemático é apresentada pela tabela 5. Eles representam as entradas e saídas do bloco.

Tabela 5: Pinos do LNA

Nome	Descrição
VDD	Fonte de alimentação
GND	Terra
In_p	Entrada positiva
In_n	Entrada negativa
Out_p	Saída positiva
Out_n	Saída negativa
G	Controle de ganho
Pd	Pino de desligamento

O código com a descrição comportamental do LNA se inicia com as declarações das bibliotecas padrões já fornecidas automaticamente pelo Verilog-AMS. Em seguida são declarados os pinos que estarão presentes no testbench, sendo estes classificados por: entradas ou saídas, e disciplinas correspondentes. Declaram-se também os parâmetros com valores pré-estabelecidos, bem como as variáveis locais que serão utilizadas no decorrer do código. Como pode ser observado na figura 23.

```

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

module lna ( in_p, in_n, out_p, out_n, pd, gnd, vdd);

    //declaracao dos pinos
    input in_n, in_p, pd, gnd, vdd;
    output out_n, out_p;
    wreal gnd, vdd, in_n, in_p, out_n, out_p;
    logic pd;

    //declaracao dos parametros
    parameter real gain = 30;
    parameter real vinsat = 0.1;

    //declaracao das variaveis locais
    real vrfn, vrfout;
    real realgain;

```

Figura 23: Declaração das variáveis LNA

A parte do código que modela a funcionalidade do bloco é implementada como um processo discreto, “always”, ou seja, ele descreve um comportamento discreto no tempo. A primeira operação a ser feita é a conversão do ganho de

dB para um numero real. Em seguida é feito o tratamento da entrada diferencial RF, uma vez que o simulador não trabalha bem com sinais quadrados ideais, já que variações abruptas não existem no mundo real. Posteriormente foi atribuído a uma variável local o sinal de entrada acrescido de um ganho, sendo esta a linha principal do código, pois descreve, de fato, a funcionalidade do LNA. Finalizando atribui-se às variáveis de saída o sinal de saída de forma diferencial juntamente com o pino de desligamento. Como pode ser observado na figura 24.

```

always @(in_p, in_n) Begin
    realgain = pow(10,gain/20);
    vrfin = vinsat* tanh(in_p - in_n)/vinsat;
    vrfout = vrfin* realgain;

    end

    assign out_p = (vrfout/2)* (~pd);
    assign out_n = -(vrfout/2)* (~pd);

```

Figura 24: módulo LNA

4.3. PROJETO FUNCIONAL DO MIXER

O mixer é o bloco do sistema de recepção ou transmissão, que apresenta duas entradas com frequências distintas podendo realizar operações com essas frequências. Logo, conforme já foi explicado anteriormente, ele é responsável pela translação da frequência do sinal de entrada para outra frequência. A translação de frequência ocorre através da multiplicação de dois sinais, sendo o primeiro o sinal proveniente do LNA e o segundo proveniente do oscilador local (LO). A utilidade fundamental desta multiplicação pode ser entendida a partir da seguinte identidade trigonométrica.

$$(A \cos w_1 t)(A \cos w_2 t) = \frac{AB}{2} [\cos(w_1 - w_2)t + \cos(w_1 + w_2)t] \quad (32)$$

Sendo que os sinais de entrada são:

$$v_{RF} = A_{RF} \cos w_{RF} t \quad (33)$$

$$v_{LO} = A_{LO} \cos w_{LO} t \quad (34)$$

Na saída do mixer tem-se:

$$v_{out} = v_{RF} \times v_{LO} \quad (35)$$

Substituindo na identidade:

$$v_{out} = \frac{A_{RF}A_{LO}}{2} [\cos(w_{RF} - w_{LO})t + \cos(w_{RF} + w_{LO})t] \quad (36)$$

Assim, o resultado na saída do mixer é a soma e a diferença dos sinais de entrada, cujas amplitudes são proporcionais ao produto das amplitudes de RF e LO.

4.3.1. Especificação

As especificações elétricas para o Mixer também foram obtidas de projetos operando em sistemas Zigbee. Elas podem ser observadas na tabela 6, logo abaixo.

Tabela 6: Especificações para o Mixer

Parâmetros	Tipo	Unidade
Tensão de alimentação	1.8	V
Temperatura	27	⁰ C
Ganho	12	dB
Figura de ruído	15	dB
IP1dB (100Ω)	-8	dBm
Frequência	2.45	Ghz
Zin (diferencial)	100	Ω
Zout (diferencial)	100	Ω
S11	-15.2	dB
S12	40	dB
Dissipação de corrente	10	mA
IIP3	-10	dBm

4.3.2. Modelagem

O primeiro passo na modelagem do bloco foi definir os pinos que iriam compor o esquemático do Mixer, sendo estes mostrados na tabela 7.

Tabela 7: Pinos do Mixer

Nome	Descrição
VDD	Fonte de alimentação
GND	Terra
In_RF_P	Entrada RF positiva

In_RF_n	Entrada RF negativa
In_LO_P	Entrada LO positiva
In_LO_n	Entrada LO negativa
Out_p	Saída positiva
Out_n	Saída negativa
Pd	Pino de desligamento

Posteriormente iniciando de fato a construção do código foi feita a declaração das bibliotecas e da diretiva de tempo. Esta última define a unidade de tempo e a precisão do tempo para os módulos que o seguem. Conforme a figura 25, abaixo.

```
`timescale 1ns/10fs
`include "constants.vams"
`include "disciplines.vams"
```

Figura 25: Definição de bibliotecas e diretiva de tempo

Iniciando o módulo de construção do modelo, foi feita a declaração: dos pinos de entrada e saída, das disciplinas para cada pino, dos parâmetros que poderão ser alterados no *testbench* e das variáveis locais que auxiliarão no desenvolvimento da lógica de funcionamento do Mixer. A figura 26 corresponde a esta parte do módulo.

```
module mixer ( vlo_n, vlo_p, vrf_n, vrf_p, vout_p, vout_n, gnd, vdd, pd);

    //declaração dos pinos
    input vlo_n, vlo_p, vrf_n, vrf_p, gnd, vdd, pd;
    output vout_n, vout_p;

    //declaração das disciplinas
    wreal vlo_n, vlo_p;
    wreal vrf_n, vrf_p;
    wreal vout_n, vout_p, gnd, vdd;
    logic pd;

    //declaração dos parametros
    parameter real gain = 1;
    parameter real vcm_out = 1;
    parameter real vsatrf = 10;

    //declaração das variaveis locais
    real vinlo, vinrf, outmix;
    real realgain;
```

Figura 26: Declaração das variáveis do Mixer

A etapa principal do modulo é executada dentro de um processo discreto e infinito (“*always*”), ou seja, suas declarações são repetidas inúmeras vezes podendo parar em pontos específicos de tempo ou por alterações de algum processo externo. Como pode ser visto na figura 27. A implementação se inicia com a conversão do ganho de dB para um numero real, para que os cálculos possam ser simplificados. Em uma variável, atribui-se o valor das entradas diferenciais advindas do VCO e, em outra, os das entradas RF. Entretanto o sinal da entrada RF deve ser tratado com a função tangente hiperbólica no intuito de amenizar uma mudança abrupta do sinal, colocando limites na variação da entrada. Em seguida, realiza-se o produto do sinal de RF por LO e também pelo ganho que o sistema deve atribuir ao sinal. Finalizando utiliza-se a função “*assign*” para transmitir continuamente o valor para a variável de saída, adiciona-se a ela uma tensão de modo comum e o controle *Power-Down*.

```

always @(vrf_n, vrf_p, vlo_n, vlo_p, pd) begin
    realgain = pow(10,gain/20);
    vinlo = vlo_p - vlo_n;
    vinrf = vsatrf*tanh((vrf_p - vrf_n)/vsatrf);
    outmix = realgain*vinlo*vinrf;

    end
    assign vout_p = ((vcm_out + outmix)/2)*(~pd);
    assign vout_n = ((vcm_out - outmix)/2)*(~pd);
endmodule

```

Figura 27: Módulo do Mixer

4.4. PROJETO FUNCIONAL DO PGA

O PGA (amplificador de ganho programável) é muito utilizado em pré-amplificação em sistemas com conversão analógico-digitais ou em sinais cujas faixas dinâmicas sejam muito elevadas. Este sistema é de fundamental importância em receptores WLAN e WPAN, além de outros tipos de receptores de comunicação com e sem fio. O controle automático de ganho é um sistema adaptativo encontrado em diversos dispositivos eletrônicos. Este sistema é comumente utilizado quando a amplitude do sinal de entrada varia sobre uma

faixa dinâmica, que é a relação dos valores máximo e mínimo de uma grandeza, em que neste caso corresponde a tensão de entrada [Moraes, 2012].

Os PGA's são amplificadores com ganhos pré-estabelecidos e que podem ser selecionados digitalmente. O seu funcionamento, de maneira geral, se dá através da amplitude do sinal de saída, quando este valor torna-se conhecido, o mesmo é realimentado para ajustar apropriadamente o ganho do amplificador e, assim, adequar a amplitude de saída. Portanto, um sinal de entrada de baixa amplitude implicará numa configuração de alto ganho do amplificador, enquanto um sinal de alta amplitude implicará em pouca ou nenhuma amplificação [Moraes, 2012]. O tipo mais simples, apresentado na Figura 28, possui entradas digitais cuja lógica combinacional é capaz de selecionar um entre diversos ganhos possíveis (2^4 no caso do exemplo). Entretanto, o PGA específico para Zigbee possui quatro bits de controle, sendo possível selecionar 16 possibilidades de ganhos.

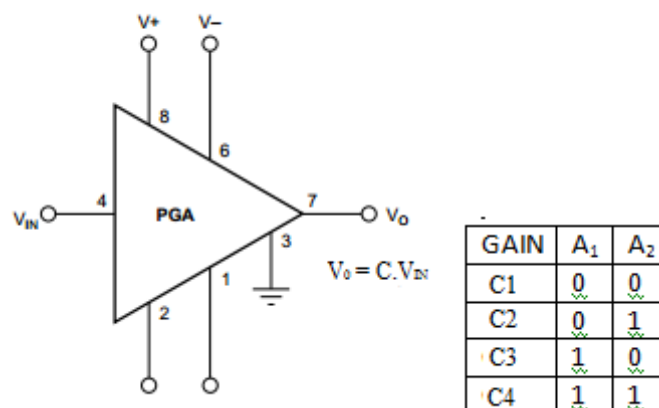


Figura 28: PGA e dois bits

Como o PGA, analisado em alto nível, se comporta como um simples amplificador, a equação que caracteriza seu comportamento funcional é dada por:

$$V_{out} = C \times V_{in} \quad (37)$$

Sendo V_{out} a tensão de saída, C o ganho final do bloco e V_{in} a tensão de entrada.

4.4.1. Especificação

As especificações elétricas para este bloco operando em sistemas Zigbee são fornecidas pela tabela 8.

Tabela 8: Especificações para o PGA

Parâmetros	Tipo	Unidade
Tensão de alimentação	1.8	V
Temperatura	27	⁰ C
Alto ganho	-22	dB
Baixo ganho	43	dB
Zin (diferencial)	100	Ω
Zin (diferencial)	100	Ω

4.4.2. Modelagem

O projeto do PGA se diferencia dos amplificadores comuns por necessitar de um controle de ganho, sendo este feito por um barramento com quatro fontes distintas para os quatro bits que o compõe. Os pinos que irão compor o esquemático do PGA são apresentados pela tabela 9.

Tabela 9: Pinos do PGA

Nome	Descrição
VDD	Fonte de alimentação
GND	Terra
In_p	Entrada positiva
In_n	Entrada negativa
Out_p	Saída positiva
Out_n	Saída negativa
C_a	Controle de ganho (bit 1)
C_b	Controle de ganho (bit 2)
C_c	Controle de ganho (bit 3)
C_d	Controle de ganho (bit 4)
pd	Pino de desligamento

No código do PGA existe a necessidade de se declarar uma variável de entrada, no caso “c”, na forma de vetor. Ela é do tipo lógica e representa o barramento de controle, como pode ser visto na figura 29.

```
//declaracao dos pinos  
input in_n, in_p, pd, gnd, vdd;
```

```

input [0:3] c;
output out_n, out_p;
wreal gnd, vdd, in_n, in_p, out_n, out_p;
logic pd;
logic [0:3] c;

//declaracao das variaveis locais
real vrfn, vrfout;
real realgain, gain;
real G = maxgain/16;

```

Figura 29: Declaração dos pinos e variáveis locais PGA

Como são quatro os bits de controle, haverá 2^4 possibilidades de ganho. Sendo assim, o ganho máximo, que poderá ser definido no *testbench*, foi dividido por 2^4 e atribuído a uma variável local “G”. Conforme mostrado na figura 30, este ganho é repassado ao ganho total do amplificador de acordo com os valores dos bits de controle.

```

always @(in_p, in_n) Begin
    case(c)
        4'b0000: begin gain = G*1; end
        4'b0001: begin gain = G*2; end
        4'b0010: begin gain = G*3; end
        4'b0011: begin gain = G*4; end
        4'b0100: begin gain = G*5; end
        4'b0101: begin gain = G*6; end
        4'b0110: begin gain = G*7; end
        4'b0111: begin gain = G*8; end
        4'b1000: begin gain = G*9; end
        4'b1001: begin gain = G*10; end
        4'b1010: begin gain = G*11; end
        4'b1011: begin gain = G*12; end
        4'b1100: begin gain = G*13; end
        4'b1101: begin gain = G*14; end
        4'b1110: begin gain = G*15; end
        4'b1111: begin gain = G*16; end
    endcase
end

```

Figura 30: Controle de ganho do PGA

Posteriormente, já com o ganho definido, segue-se a mesma lógica utilizada no LNA, figura 31. O sinal de entrada é tratado com a tangente hiperbólica e em seguida é incorporado o ganho no sinal. Por fim, o sinal é

transmitido para as variáveis de saída de forma diferencial, acrescido do *power-down*.

```

    realgain = pow(10,gain/20);
    vrfin = vinsat* tanh(in_p - in_n)/vinsat;
    vrfout = vrfin* realgain;
    end

    assign out_p = (vrfout/2)* (~pd);
    assign out_n = -(vrfout/2)* (~pd);

endmodule

```

Figura 31: Modulo principal do PGA

4.5. DESCRIÇÃO FUNCIONAL DO VCO

O oscilador local nada mais é do que uma fonte que gera uma senóide de frequência variável, de acordo com a frequência de operação. Um oscilador local é um oscilador eletrônico utilizado para gerar um sinal, geralmente com o objetivo de converter um sinal de interesse, com uma frequência diferente, utilizando um misturador. Contudo, a frequência da maioria dos osciladores de RF deve ser regulável, devido à necessidade de selecionar um dos muitos canais, já que a um determinado receptor é atribuído diferentes frequências de portadoras em momentos também diferentes. Assim, a frequência de LO em cada caso, deve variar em passos bem definidos. Se a frequência de um oscilador de saída pode ser variada por uma tensão, o circuito é chamado um oscilador controlado por tensão (VCO).

Em implementações de LC, a capacitância do tanque pode ser fornecido por um diodo polarizado inversamente de modo que a tensão DC através da junção controla a frequência de ressonância, variando assim a frequência de um oscilador. Um VCO ideal é um circuito que gera uma saída periódica, cuja frequência é uma função linear da tensão de controle, V_{count} , dada pela equação XX.

$$W_{out} = W_{RF} + (K_{VCO} * V_{count}) \quad (38)$$

Onde W_{RF} é a frequência de oscilação e K_{vco} é o ganho do VCO. A existência de W_{RF} na equação acima indica simplesmente que, para o range

prático de V_{count} , W_{out} não pode aproximar-se de zero. Em outras palavras, V_{count} cria uma mudança em torno W_{FR} . Uma vez que a fase é a integral da frequência com relação ao tempo, a saída de um VCO senoidal pode ser expressa como:

$$y(t) = A \cos \left(W_{RF} t + K_{vco} \int_{-\infty}^t V_{count} dt \right) \quad (39)$$

Onde A é a amplitude do sinal. Em particular, para uma modulação senoidal, $V_{count} = V_m \cdot \cos(W_m t)$. Em que se tem a característica funcional representada por:

$$y(t) = A \cos \left(W_{RF} t + \frac{K_{vco}}{W_m} V_m \sin W_m t \right) \quad (40)$$

Indicando que o VCO tem uma tendência para rejeitar componentes de alta frequência que aparecem na sua entrada de controle.

4.5.1. Especificação

Logo abaixo, tabela 10, encontram-se as especificações elétricas para o VCO que opera em módulos Zigbee.

Tabela 10: Especificações para o VCO

Parâmetros	Tipo	Unidade
Tensão de alimentação	1.8	V
Temperatura	27	$^{\circ}\text{C}$
Ganho	20	dB
Frequência	2.405	Mhz

4.5.2. Modelagem

A representação das entradas e saídas do bloco encontra-se, a seguir, na tabela 11.

Tabela 11: Pinos do VCO

Nome	Descrição
VDD	Fonte de alimentação
GND	Terra
Out_p	Saída positiva
Out_n	Saída negativa
In	Entrada de controle

pd	Pino de desligamento
----	----------------------

O VCO apresenta certa quantidade de parâmetros a mais que os demais blocos que podem ser alterados, como podem ser observados na figura 32. Nele é necessário definir a tensão máxima e mínima do sinal de entrada, bem como as frequências máxima e mínima, faixa de operação, que o mesmo pode oferecer ao circuito, além da amplitude do sinal de saída. As declarações dos parâmetros são semelhantes aos apresentados nos blocos anteriores, exceto que os parâmetros e limites para alguns parâmetros são definidos em termos de parâmetros previamente definidos.

```
//declaração dos parâmetros
parameter real vmin = 0;
parameter real vmax = vmin +1.8 from (vmin:inf);
parameter real fmin = 2.4G from (0:inf);
parameter real fmax = 2.5G from (fmin:inf);
parameter real ampl = 1;
```

Figura 32: Parâmetros do VCO

A etapa principal do modulo, figura 33, é executada dentro de um processo analógico (“analog”), ou seja, processo regido por comportamento contínuo no tempo. O processo se inicia com o cálculo da frequência que o VCO irá fornecer ao sistema. Este cálculo é feito vinculando a tensão de entrada do bloco, uma vez que é esta que controla o VCO, com as faixas de tensão e frequência fixadas para o mesmo. Em seguida, é feita a certificação de que a frequência que o módulo irá fornecer ao sistema está dentro da faixa pré-estabelecida.

```
analog Begin

    freq = (V(in)-vmin)*(fmax - fmin)/(vmax - vmin)+fmin;

    if (freq > fmax) freq = fmax;
    if (freq < fmin) freq = fmin;

    phase = 2*M_PI*idtmmod(freq,0.0,1.0,-0.5);
    V(out_p) <+ (sin(phase)/2)*(~pd);
    V(out_n) <+ (-sin(phase)/2)*(~pd);
    $bound_step (0.1/freq);

end
```

Figura 33: Módulo principal do VCO

A fase do sinal de saída é calculada através do produto de 2π por uma integral circular, ou integral de caminho fechado. Esta integral é preferida sobre o operador convencional por manter a potência de saída e as variáveis de estado internas limitadas, o que evita os problemas de tolerância e de arredondamentos. Este operador apresenta cinco argumentos: operando (*operand*), uma condição inicial (*ic*), um módulo (*modulus*), um offset e uma tolerância absoluta (*abstrof*). Por fim o sinal de saída é atribuído de forma diferencial e com pino para desligamento, para uma variável de saída. A função “*bound_step*” é utilizada para informar ao simulador que mudanças suaves estão ocorrendo. Ela limita o espaço de tempo a não ser maior do que um décimo do tamanho de um período de saída do VCO.

4.6. DESCRIÇÃO FUNCIONAL DO PA

Amplificador de potência é o estágio de um transmissor de RF (radiofrequência) que eleva o sinal de RF fornecido pelo mixer à níveis de tensão e impedância adequados, suficientemente altos para que ocorra a transmissão. Pelo fato de estar conectado a um módulo que possui uma carga de baixa impedância (antena) e pelo fato de fornecer uma potência muito alta para transmitir o sinal, ele é um dos blocos que mais consomem potência na cadeia de transmissão.

Como o PA, analisado em alto nível, se comporta como um simples amplificador, a equação que caracteriza seu comportamento funcional é dada por XX:

$$V_{out} = A \times V_{in} \quad (41)$$

Sendo V_{out} a tensão de saída, A o ganho final do bloco e V_{in} a tensão de entrada.

4.6.1. Especificação

As especificações elétricas do PA em módulos Zigbee são fornecidas pela tabela 12.

Tabela 12: Especificações para o PA

Parâmetros	Valor (tipo)	Unidade
Tensão de alimentação	1.8	V
Temperatura	27	⁰ C
Ganho de Potência	10	dB
Frequência	2.45	Ghz
Zin (diferencial)	100	Ω
Zout (diferencial)	100	Ω
Dissipação de Potência DC	4.5	mW
Dissipação de corrente	2.5	mW
Tecnologia	0.18	um
Potencia de maior eficiência	22%	

4.6.2. Modelagem

Os pinos de entrada e saída que compõem o PA são mostrados na tabela 13.

Tabela 13: Pinos do PA

Nome	Descrição
VDD	Fonte de alimentação
GND	Terra
In_p	Entrada positiva
In_n	Entrada negativa
Out_p	Saída positiva
Out_n	Saída negativa
A	Controle de ganho de potência
pd	Pino de desligamento

O PA se comporta de maneira idêntica ao LNA. A figura 34, apresenta os parâmetros reais pré-estabelecidos que fazem parte do modelo funcional, sendo apenas os valores destes a diferença entre ambos.

```
//declaracao dos parametros
parameter real gain = 30;
parameter real vinsat = 10;
```

Figura 34: Parâmetros reais do PA

A etapa principal do PA, figura 35, que é onde de fato é descrito o comportamento funcional do mesmo, também é idêntica ao LNA. A diferença entre os módulos está no local que cada qual se encontra no sistema

transceptor, sendo o LNA na cadeia de recepção e o PA na cadeia de transmissão conforme já foi visto.

```
always @(in_p, in_n) begin
    realgain = pow(10,gain/20);
    vrfin = vinsat* tanh(in_p - in_n)/vinsat;
    vrfout = vrfin* realgain;
end

assign out_p = (vrfout/2)* (~pd);
assign out_n = -(vrfout/2)* (~pd);
```

Figura 35: Modelo funcional PA

4.7. DESCRIÇÃO FUNCIONAL DO FILTRO PASSA-BAIXA

O filtro passa-baixa é utilizado tanto na recepção quanto na transmissão do sinal. Sua utilização na recepção é justificada pelo fato do mixer gerar um sinal em baixa frequência resultando em alguns espúrios de alta frequência, assim o filtro será responsável por eliminar estes espúrios. Já na transmissão, se deve a necessidade de estar adequando o sinal para a banda que o transmissor estará ocupando, sendo esta regulada pela Anatel.

Filtro passa-baixa é o nome comum dado a um circuito Eletrônico que permite a passagem de baixas frequências sem dificuldades e atenua (ou reduz) a amplitude das frequências maiores que a frequência de corte. Sendo assim, sua característica funcional pode ser representada pela convolução do sinal de entrada com a função transferência do filtro. Logo:

$$V_{out} = H(w) * V_{in} \quad (42)$$

Sendo V_{out} a tensão de saída, $H(W)$ a função de transferência do filtro e V_{in} a tensão de entrada.

4.7.1. Especificação

As especificações elétricas para este bloco encontram-se na tabela 14, logo abaixo, tais informações foram obtidas do estudo de projetos feitos na área de transceptores Zigbee.

Tabela 14: Especificações para o Filtro

Parâmetros	Tipo	Unidade
Tensão de alimentação	18	V
Temperatura	27	⁰ C
Frequência TX	2.5	MHz
Frequência RX	2.5	MHz
Ordem TX	2	
Ordem RX	3	
modelo	Butterworth	

4.7.2. Modelagem

A tabela 15 apresenta a relação dos pinos que compõem o esquemático do filtro.

Tabela 15: Pinos do filtro

Nome	Descrição
VDD	Fonte de alimentação
GND	Terra
In_p	Entrada positiva
In_n	Entrada negativa
Out_p	Saída positiva
Out_n	Saída negativa
pd	Pino de desligamento

O filtro não apresenta parâmetros que possam ser modificados no *testbench* assim, sua atuação é fixa para determinada ordem e frequência de corte. A modelagem desse bloco é feita de forma bastante simples, uma vez que há no Verilog-AMS uma função que o implementa de maneira direta. A função é a “laplace_nd(operand, n,d)” que necessita do operador e dos polos e zeros da função transferência que o represente, como pode ser visto na figura 36.

```

analog begin

    V(out_n) <+ (laplace_nd(V(in_n),{8e018},{8e018,8e012,4e006,1.0}))*(~pd);
    V(out_p) <+ (laplace_nd(V(in_p),{8e018},{8e018,8e012,4e006,1.0}))*(~pd);

End

```

Figura 36: Modelo funcional do Filtro

Para encontrar a função transferência com suas respectivas especificações utilizou-se o software Matlab, figura 37.

```

[A,B] = butter (3,2000000,'s')
%H=freqs(B,A,w);
bode(A,B)
%t 0:0.0001:1;
%x=sin(2*3.14*1000*t)
%y=filter(A,B,'s',x)
%plot(y)
TF (A,B)

```

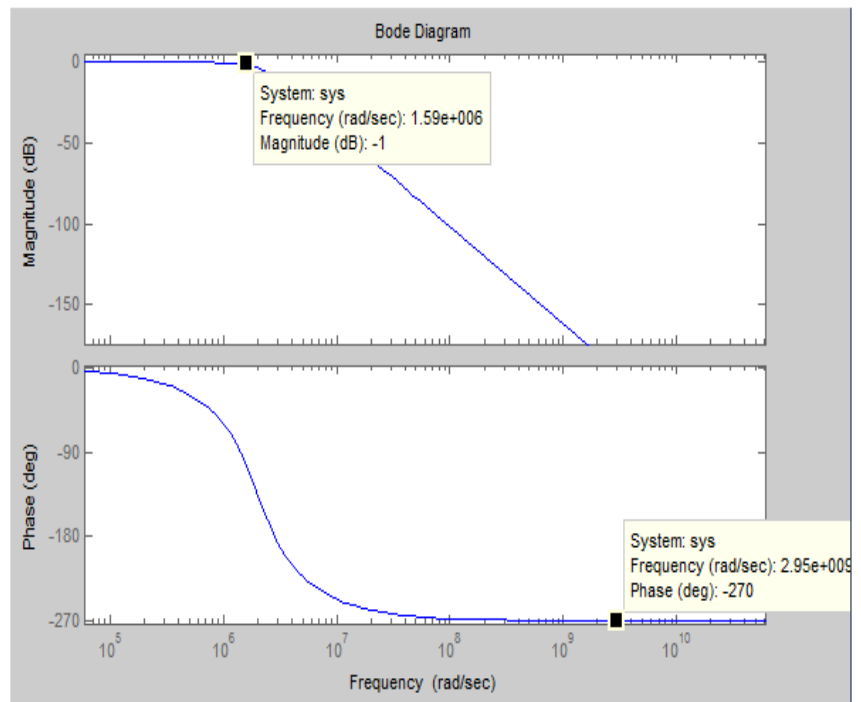


Figura 37: Função no matlab para filtro

Assim a função de transferência do Butterworth de terceira ordem e frequência de corte 2MHz obtida foi:

$$H_s = \frac{8.10^{18}}{S^3 + 4.10^6 S^2 + 8.10^{12} S + 8.10^{18}} \quad (43)$$

5. SIMULAÇÃO E ANÁLISE DOS RESULTADOS

Este capítulo tem por objetivo apresentar e discutir resultados obtidos a partir de simulações realizadas da modelagem dos blocos de acordo com o exposto nas seções anteriores. Primeiramente será feita as simulações dos blocos individuais, no intuito de validar o modelo. Em seguida será feita a simulação das cadeias de recepção e transmissão individualmente e por fim será apresentada a simulação do sistema transceptor completo com as especificações Zigbee.

5.1. VALIDAÇÃO BLOCOS INDIVIDUAIS

5.1.1. Simulação LNA

O modelo funcional deste bloco foi implementado em alto nível de abstração, ou seja, sem uma análise de circuitos e nem as equações de elementos ativos ou passivos foram consideradas neste primeiro momento. A figura 38 mostra o *testbench* do LNA, ele foi criado com o símbolo gerado pelo simulador assim que o código em verilog-AMS foi compilado.

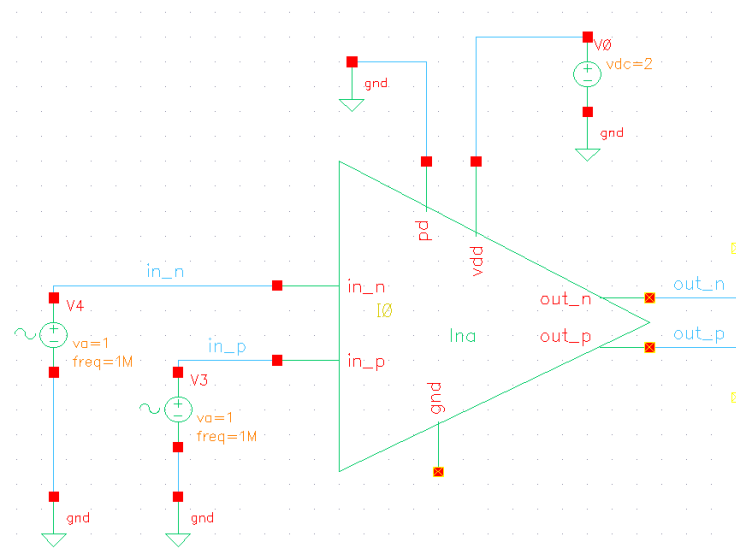


Figura 38: *Testbench* LNA

Como a principal função deste bloco é amplificar, ele está composto por um ganho na entrada que será computado de acordo com o valor especificado pelo usuário. Os parâmetros do modelo que poderão ser modificados no *testbench* são apenas dois: o ganho (dB) do bloco e a tensão de saturação do sinal de entrada.

Para o teste de validação do LNA projetado foram especificados um ganho de 30dB e uma tensão de saturação de 20V. O sinal de entrada corresponde a uma senóide com amplitude de 1V e frequência de 1MHz, sendo que a entrada *in_n* está defasada 180° da entrada *in_p*. O resultado pode ser observado na figura 39.

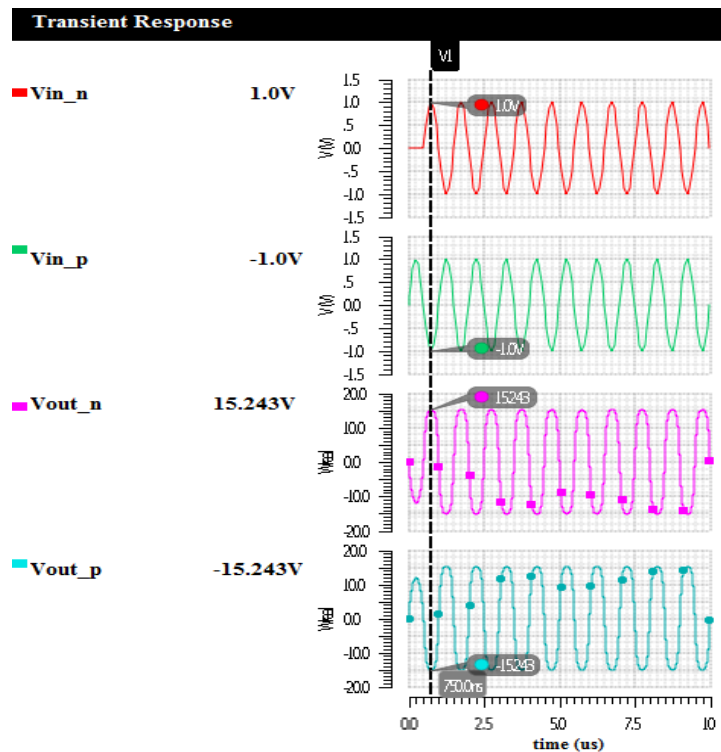


Figura 39: Simulação LNA

De acordo com as curvas, o ganho aplicado ao sinal de entrada pode ser calculado da seguinte maneira:

$$ganho = 20 * \log\left(\frac{15.243 * 2}{1.0}\right) \quad (44)$$

$$ganho = 29.68 \quad (45)$$

$$frequência = \frac{1}{250\mu s * 4} = 1MHz \quad (46)$$

Logo, é possível constatar que o valor obtido é bastante coerente com o desejado e que a frequência do sinal continua a mesma, o que valida o funcionalmente do bloco.

5.1.2. Simulação Mixer

A simulação do mixer foi feita para uma conversão *downconversion*, ou seja, o sinal na saída vai diretamente para uma frequência menor, que no caso é a frequência de banda base do sinal definido. A figura 40 representa o *testbench* do bloco. Como pode ser observado, os sinais tanto na entrada RF quanto na entrada LO são fornecidos por fontes de correntes senoidal a título de simulação.

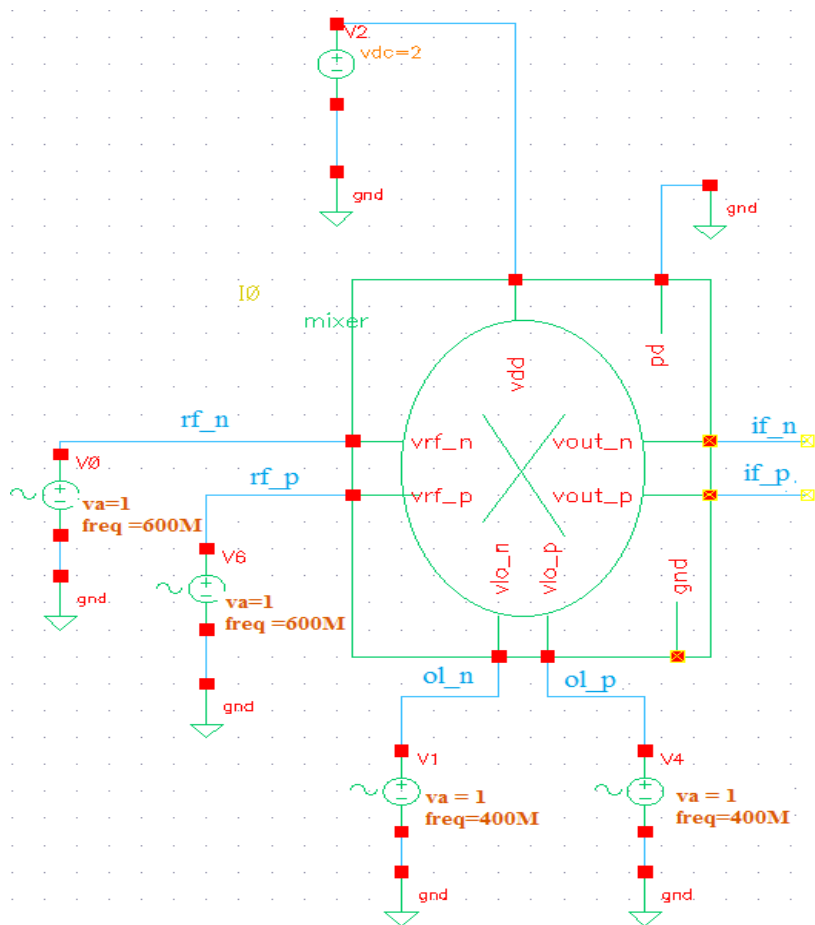


Figura 40: *Testbench* do mixer

Na saída do bloco encontra-se um conjunto de frequências que são as somas e diferenças das frequências múltiplas dos dois sinais de entrada, RF e LO, ou seja:

$$f_{saída} = mf_{RF} \pm nf_{LO} \quad (47)$$

Os sinais na saída do mixer, como pode ser visto na figura 41.

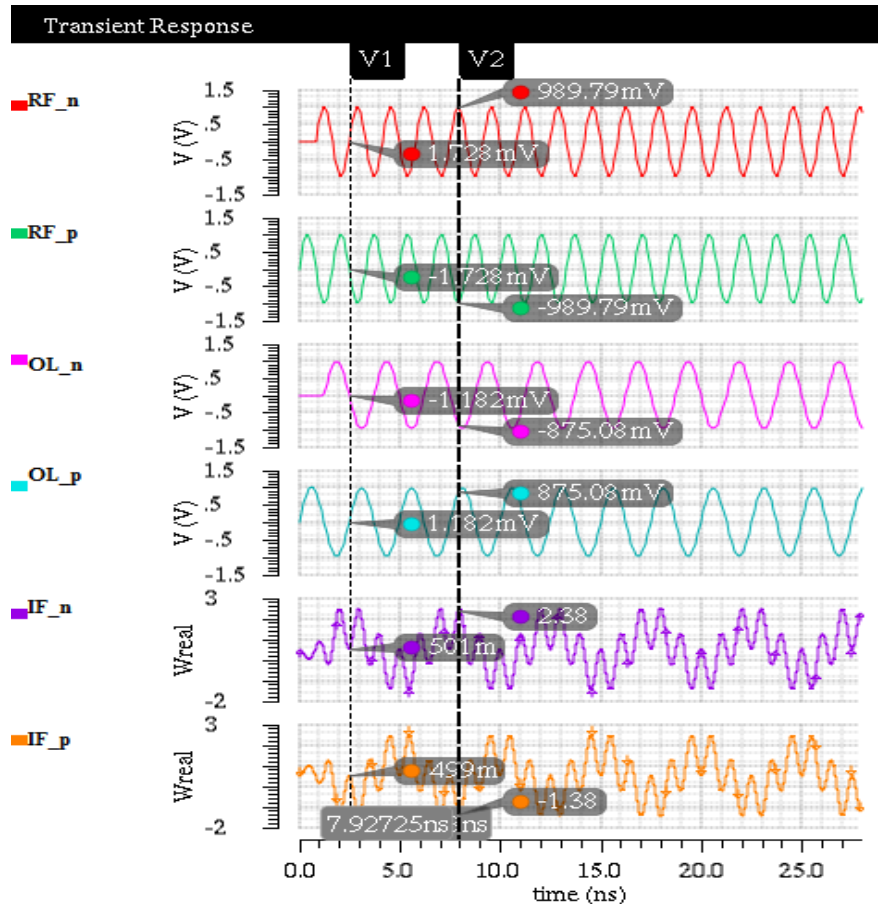


Figura 41: Simulação Mixer

Observa-se uma oscilação nos sinais uma vez que a frequência central fica variando, entretanto a amplitude do sinal continua o mesmo.

5.1.3. Simulação PGA

O PGA possui um comportamento funcional similar ao LNA. A diferença entre eles é que o ganho definido pelo usuário pode variar de acordo com uma tensão de controle. No *testbench* da figura 42 é possível observar as quatro fontes de sinal que alimentam os quatro bits de controle do mesmo. Estas

fontes se comportarão de forma lógica, sendo 1.8V representando o nível lógico alto e 0V o nível baixo.

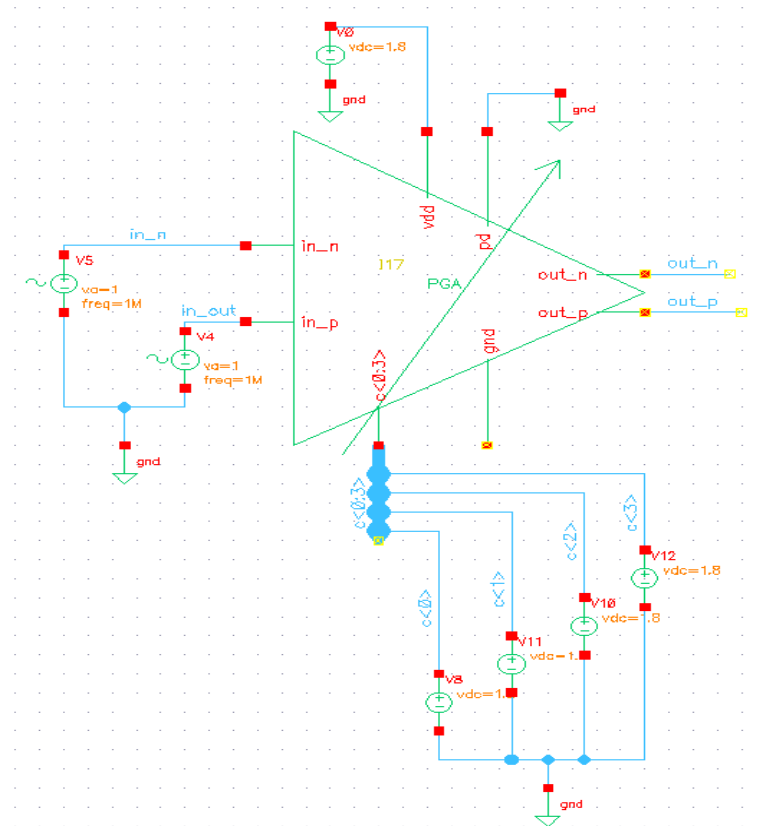


Figura 42: *Testbench* PGA

Da mesma maneira que ocorre com LNA, os únicos parâmetros que poderão ser modificados no testbench são: o ganho e a tensão de saturação do sinal de entrada. Essa liberdade permite que o modelo seja utilizado para diferentes especificações e aplicações. O controle do ganho foi feito para o máximo (1111) e para o mínimo (0000). O resultado pode ser observado na figura 43.

Para o teste de validação foram especificados um ganho de 20dB e uma tensão de saturação de 10V. O sinal de entrada corresponde a uma senóide com amplitude de 1V e frequência de 1MHz.

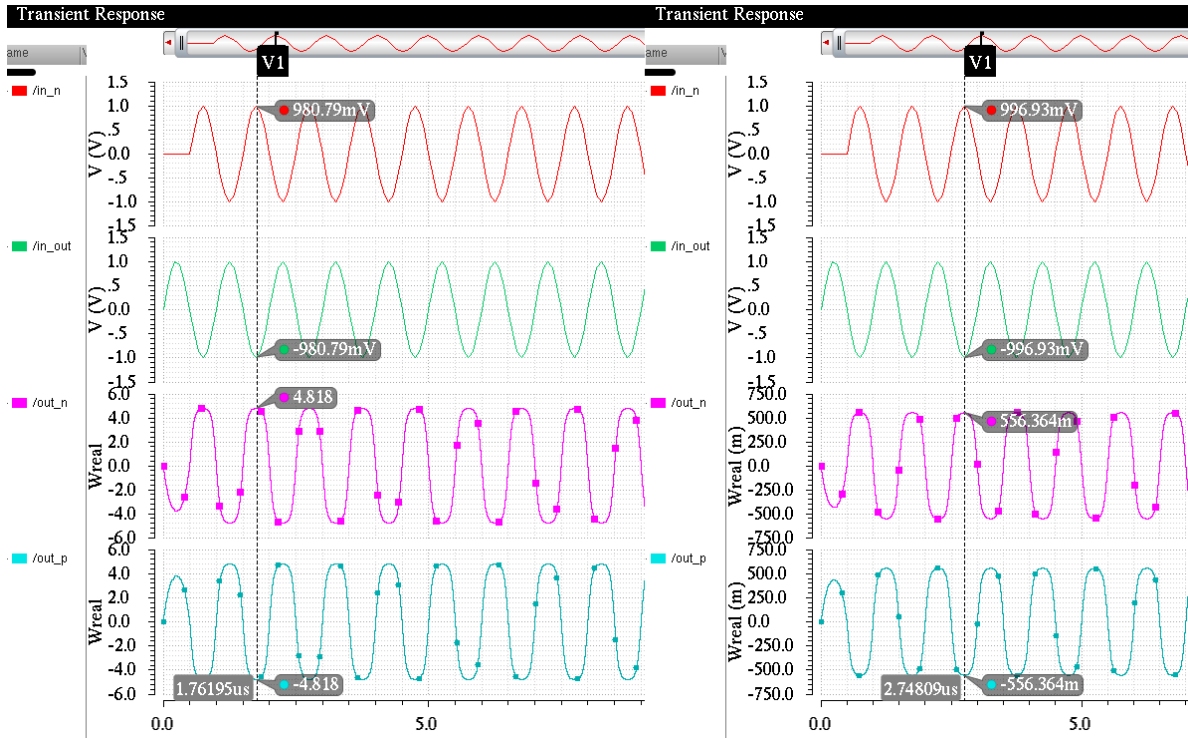


Figura 43: Simulação PGA

De acordo com as curvas obtidas, pode-se calcular o ganho aplicado ao sinal de entrada da seguinte maneira:

$$ganho_{max} = 20 * \log\left(\frac{4.818 * 2}{1.0}\right) \quad (48)$$

$$ganho_{max} = 19.677 \quad (49)$$

$$freq_{max} = \frac{1}{2.4985\mu s * 4} = 1MHz \quad (50)$$

$$ganho_{min} = 20 * \log\left(\frac{556.364 * 10^{-3} * 2}{998.85 * 10^{-3}}\right) \quad (51)$$

$$ganho_{min} = 0.9377 \quad (52)$$

$$freq_{min} = \frac{1}{250.01\mu s * 4} = 1MHz \quad (53)$$

Sendo, $freq_{max}$ a frequência máxima e $freq_{min}$ a frequência mínima. Logo, os valores dos ganhos retirados da simulação são muito coerentes com os desejados, considerando se mais uma vez que a frequência do sinal permanece a mesma, o que valida funcionalmente o PGA.

Na figura 44 foi feita uma simulação em que as fontes de controle dos bits foram programadas de maneira se pudesse verificar todos os estágios de ganho possíveis do PGA.

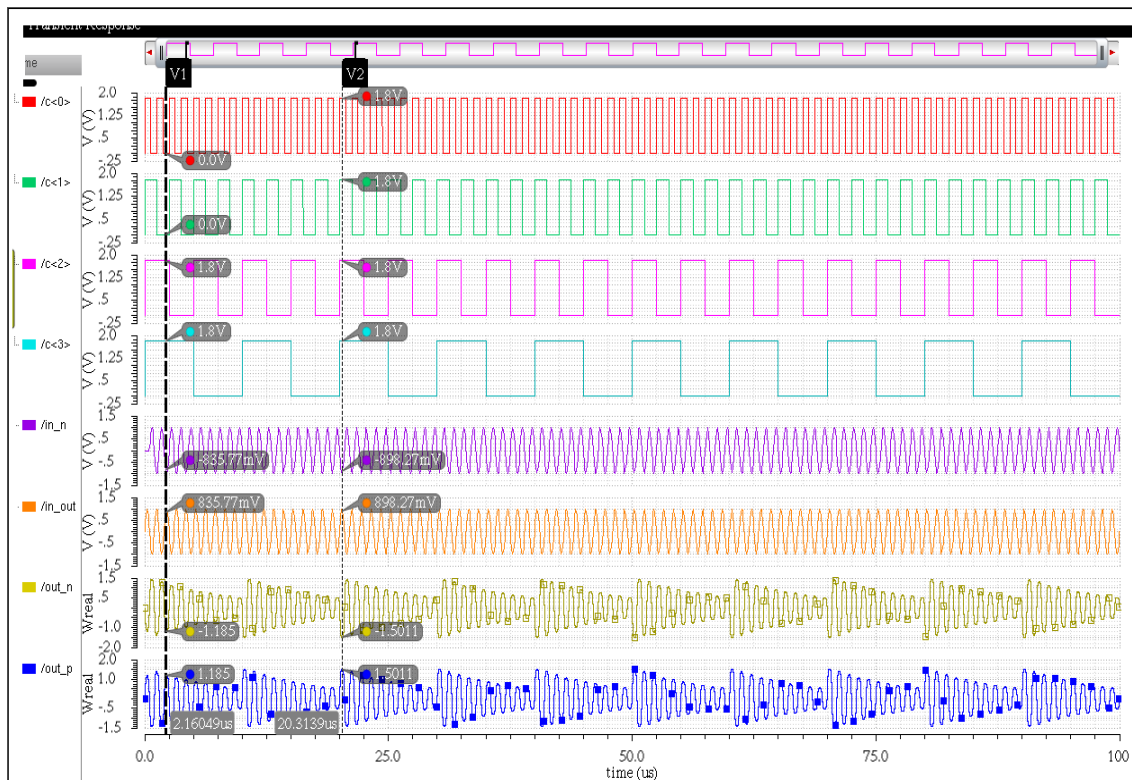


Figura 44: Simulação PGA com todos os estágios de ganho

A tabela 16 contém os dezesseis estágios da relação entre os bits de controle e o ganho do amplificador.

Tabela 16: Bits de controle x ganho

Estágio de ganho	Bits de controle	Ganho (dB)
1	0000	0.625
2	0001	1.25
3	0010	1.875
4	0011	2.5
5	0100	3.125
6	0101	3.75
7	0110	4.375
8	0111	5
9	1000	5.625

10	1001	6.25
11	1010	6.875
12	1011	7.5
13	1100	8.125
14	1101	8.75
15	1110	9.375
16	1111	10

5.1.4. Simulação VCO

A funcionalidade principal do VCO é gerar a portadora necessária para o canal. Ele constitui-se basicamente de um gerador de sinais, que opcionalmente pode ter uma defasagem de 90^0 para a componente em quadratura. A figura 45 mostra o testbench de simulação do bloco.

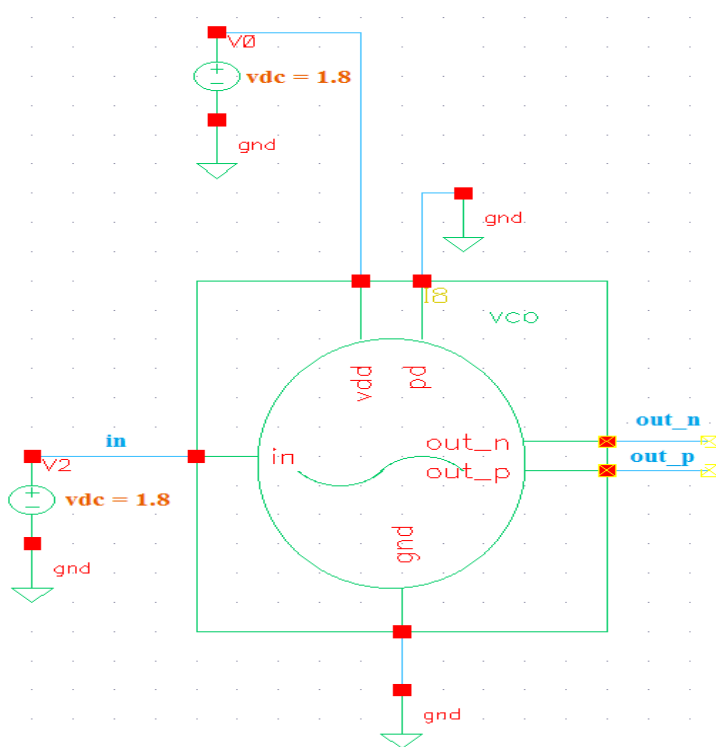


Figura 45: Testbench VCO

Uma das figuras de mérito importantes para um VCO é a faixa de frequência que este é capaz de sintonizar. Assim, para o teste do modelo foram definidos os seguintes parâmetros: a mínima tensão de entrada em 0V, a máxima tensão de entrada em 1.8V, a mínima frequência de saída em 2.4GHz,

a máxima frequência de saída em 2.5GHz e a amplitude do sinal de saída em 1V. O resultado pode ser observado na figura 46.

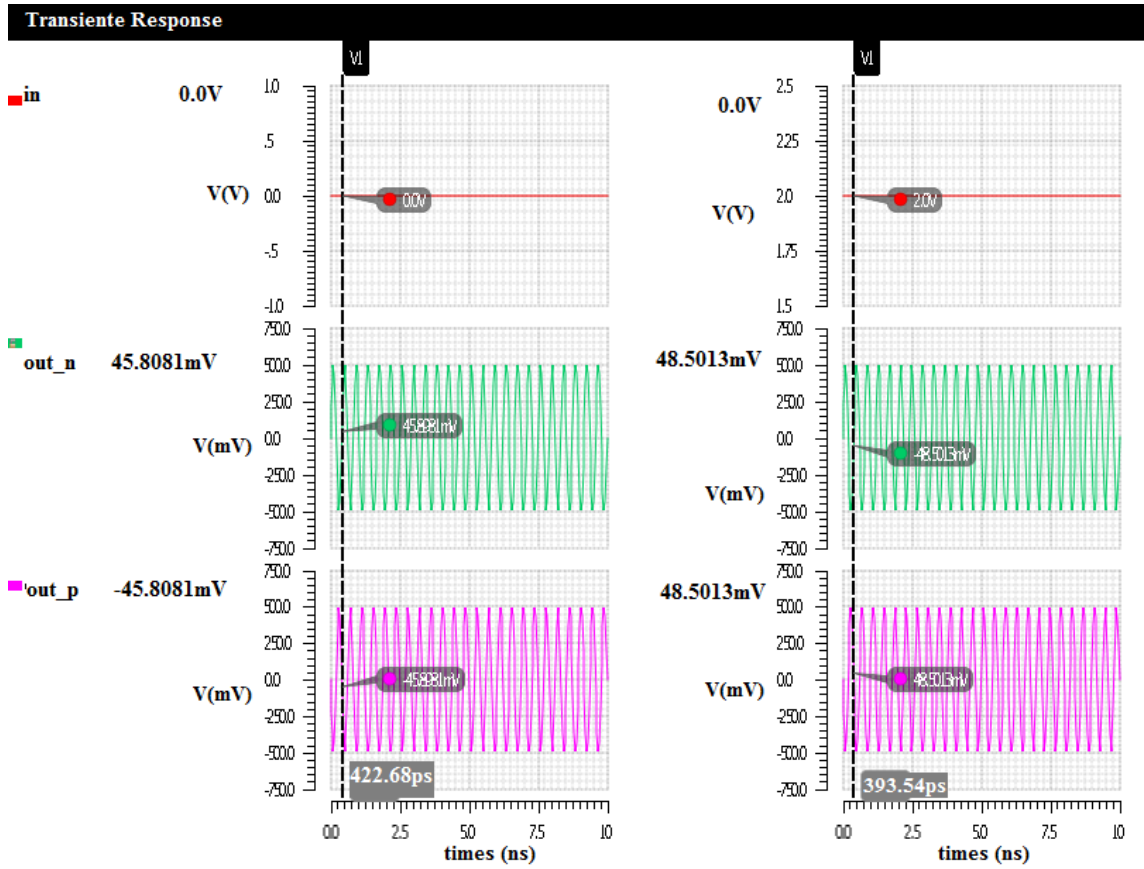


Figura 46: Simulação VCO para mínima e máxima tensão de controle

A figura 46 possui duas simulações, sendo a primeira com a mínima tensão de controle e a segunda com a máxima. É possível analisar se o comportamento do mesmo está adequado com o seguinte cálculo:

$$freq_{min} = \frac{1}{T} = \frac{1}{422.68 * 10^{-12}} \quad (54)$$

$$freq_{min} = 2.365GHz \quad (55)$$

$$freq_{max} = \frac{1}{T} = \frac{1}{393.54 * 10^{-12}} \quad (56)$$

$$freq_{max} = 2.54GHz \quad (57)$$

Sendo $freq_{max}$ a frequência máxima, $freq_{min}$ a frequência mínima e T o período da onda. Assim observa-se que os resultados obtidos, 2.36GHz e 2.54GHz, ficaram próximos do esperado. Observa-se também nas ondas do

sinal, figura 46, que a amplitude de saída se encontra exatamente com 1V pico a pico. Já a figura 47, logo abaixo, apresenta a variação da frequência de oscilação com relação à variação do sinal de controle em passos de 0.09V.

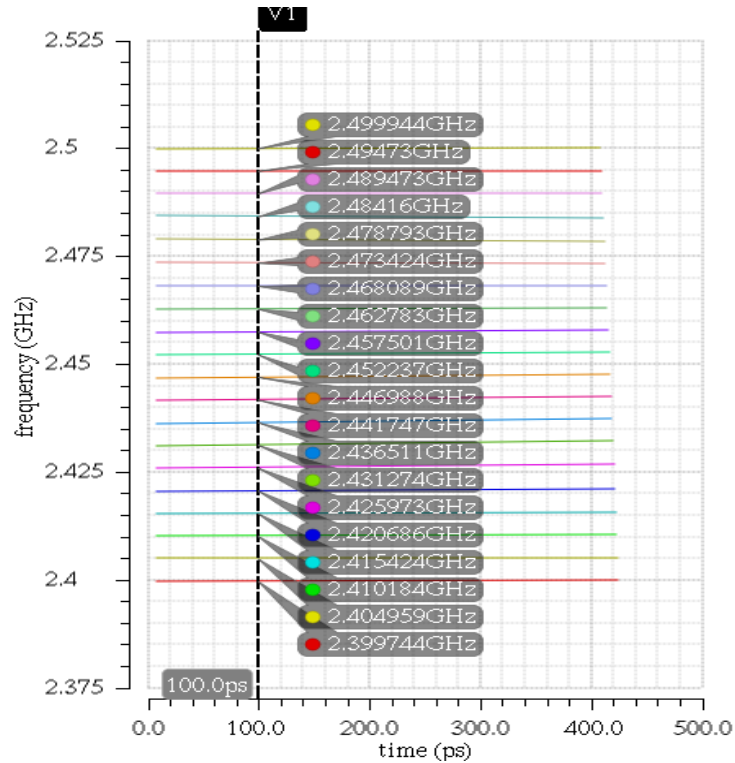


Figura 47: Variação da frequência devido à variação do sinal de controle

5.1.5. Simulação PA

O Amplificador de potência possui comportamento funcional idêntico ao do LNA, Apesar de apenas fornecer potência, ele também adiciona um ganho ao sinal. Entretanto, o PA é utilizado apenas na cadeia de transmissão enquanto o LNA pertence a cadeia de recepção. Os parâmetros desse bloco também são diferentes, uma vez que sua localização exige uma aplicabilidade diferente. A figura 48 mostra o *testbench* do PA.

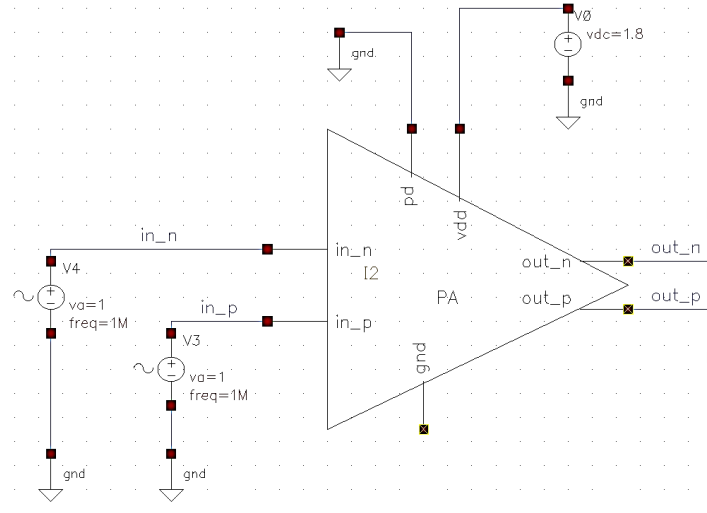


Figura 48: Testbench do PA

Os parâmetros utilizados no modelo foram: um ganho de 10dB e uma tensão de saturação do sinal de entrada de 0.1V. O sinal de entrada permanece o mesmo, uma senóide com amplitude de 1V e frequência de 1MHz. O resultado está explícito na figura 49.

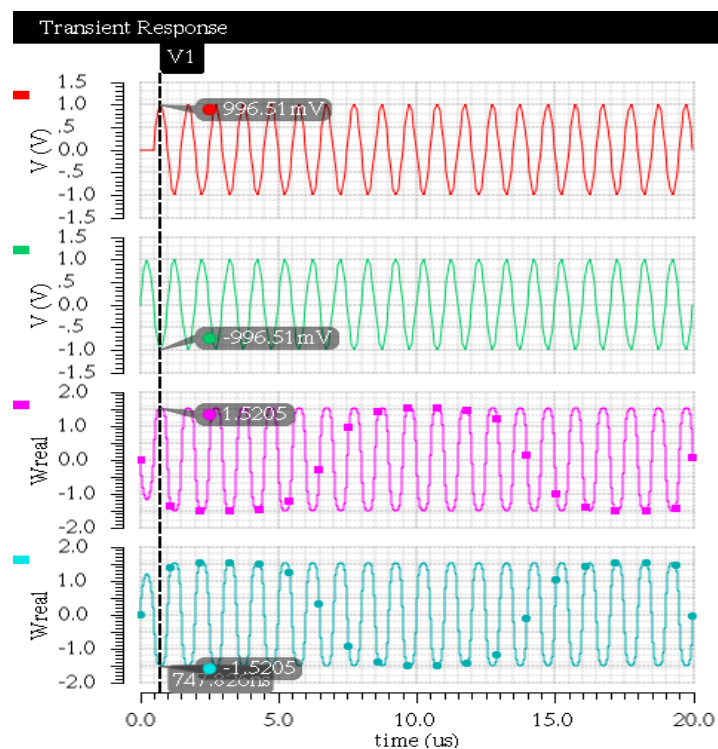


Figura 49: Simulação PA

O Ganho do circuito simulado é dado por:

$$ganho = 20 * \log\left(\frac{1.5205 * 2}{996.51 * 10^{-3}}\right) \quad (58)$$

$$ganho = 9.69 \quad (59)$$

$$frequência = \frac{1}{0.2493\mu s * 4} = 1MHz \quad (60)$$

Com os cálculos é possível concluir que os valores obtidos são próximos do esperado, o que também valida funcionalmente do PA.

5.1.6. Simulação Filtro

O modelo funcional de um filtro passa baixas foi implementado, no intuito de se permitir a passagem até uma determinada frequência específica. O *testbench* utilizado para o teste do mesmo pode ser visto na figura 50.

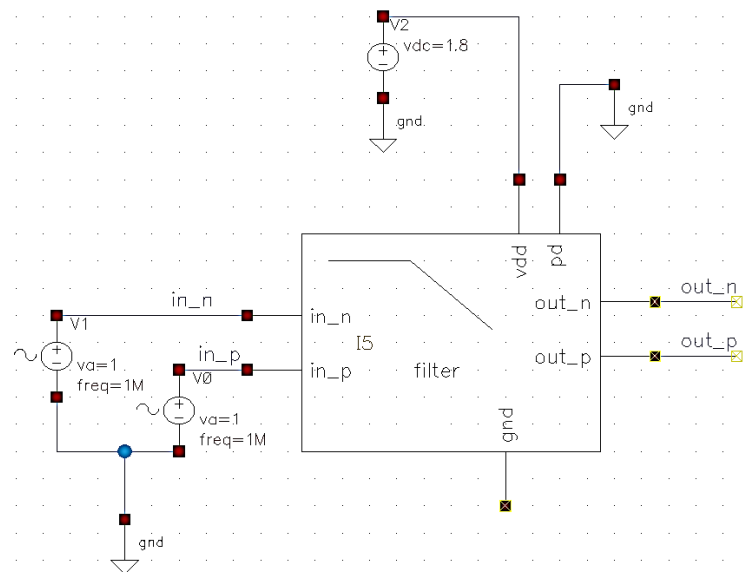


Figura 50: Testbench filtro passa-baixas

Foi projetado um filtro do tipo butterworth de terceira ordem para uma frequência de corte de 2MHz. O projeto de um butterworth permite ter uma função de transferência com o mínimo de oscilações tanto na banda passante como na banda de corte. Diferentemente dos demais blocos, o filtro não apresenta parâmetros que poderão ser modificados no *testbench* uma vez que ele é responsável apenas por eliminar as frequências indesejadas na banda de operação já específica.

A escolha da frequência de corte em 2MHz se deve ao fato do zigbee possuir a largura do canal de exatamente 2MHz restando ainda 1.5MHz de distância do próximo canal. Quando projetado no matlab, no intuito de se obter a função transferência, o filtro funcionava em perfeitas condições conforme esperado. Entretanto, ao inserir as informações obtidas na função laplace do verilog-A, observou-se que a mesma não funciona de forma eficiente já que não respondia de forma satisfatória. Observou-se que a frequência de corte mudou para 0.1MHz e que a partir 2MHz já ocorria a rejeição total, o que poderia resultar em perda de informação. A solução encontrada foi aumentar a frequência de corte de maneira que o erro pudesse ser compensado. Assim, a frequência de corte foi alterada para 20MHz o que permitiu que toda a faixa de 1MHz pudesse fazer parte da banda passante, e que mais de 70% do sinal fosse atenuado em 5MHz (limite para o próximo canal). O resultado pode ser observado na figura 51.

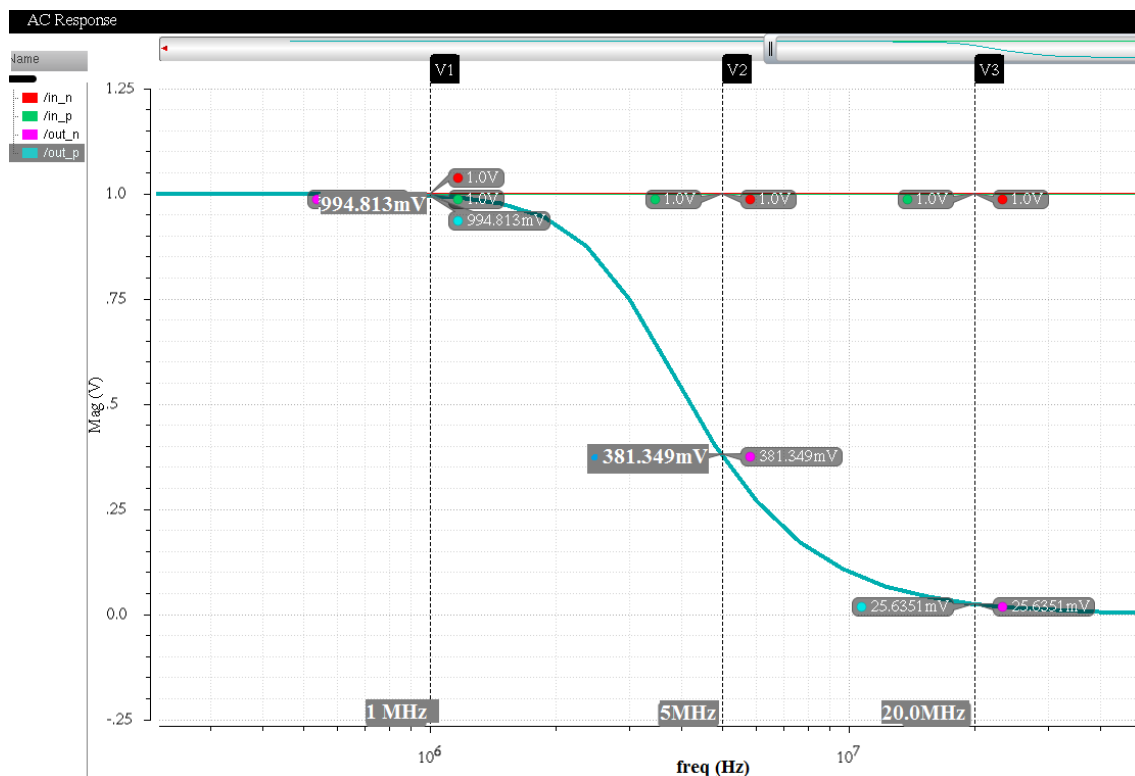


Figura 51: Simulação Filtro passa baixa

5.2. VALIDAÇÃO PROJETO ELÉTRICO LNA

Após os cálculos dos dimensionamentos feitos no capítulo 4.1.2 foi feita a simulação do circuito através dos Parâmetros S, onde é possível obter uma análise mais apurada do circuito e, portanto, realizar um refinamento nos valores dos componentes a serem utilizados tendo como base os dispositivos calculados durante o projeto. Em sistemas de RF, parâmetros de espalhamento (ou parâmetros-S) desempenham um importante papel uma vez que, para altas frequências as medidas em curto-circuito e em circuito aberto não são aplicáveis devido a indutância e capacitância acoplada ao circuito. Os parâmetros-S são utilizados para definir a relação de entrada e saída de uma rede em forma de coeficientes de reflexão e transmissão, utilizando-se o fato de que uma linha de transmissão terminada em sua impedância característica (Z_0) não tem reflexões e define as variáveis de entrada e saída em termos de ondas de tensão incidentes e refletidas / espalhadas (Lee, 2004). Neste caso, a impedância característica é de 50Ω . A figura 52 mostra o projeto sendo simulado. Circuito com os parâmetros ajustados pode ser encontrado em anexo.

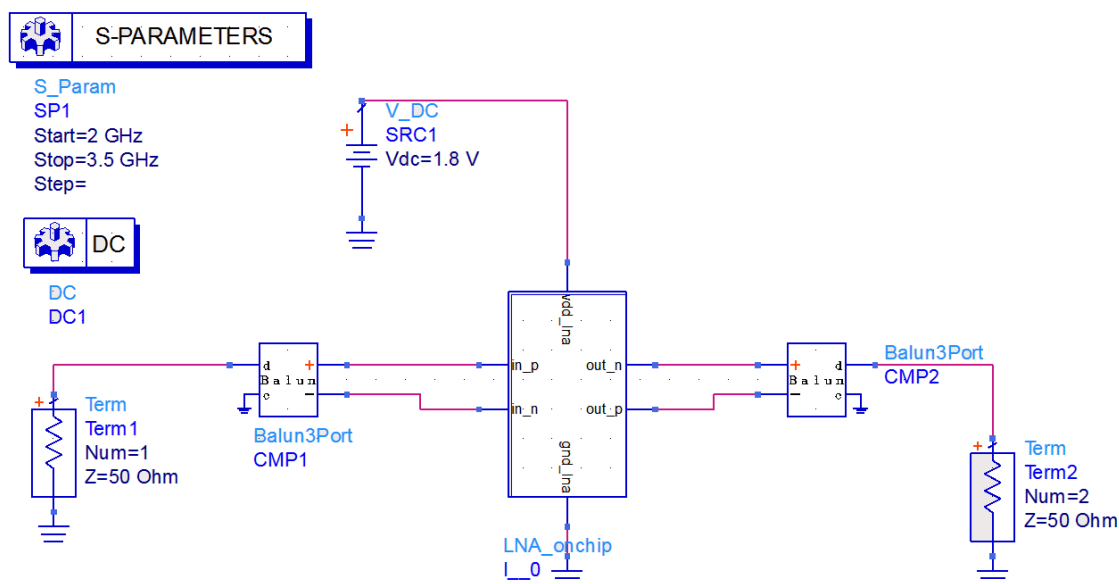


Figura 52: Simulação do *test-bench* off-chip.

A primeira análise a ser feita foi perante o parâmetro S_{11} , que corresponde ao coeficiente de reflexão do sinal entrada. Ele permite a verificação da sintonia

do circuito de casamento feito na entrada do amplificador na faixa da frequência de interesse (2,4 – 2,5GHz). Observou-se que, a medida que era feito o decremento do valor de L_g , o vale da curva de S11 se aproximava da frequência de interesse. Deste modo, o valor de L_g encontrado foi de 11nH. O resultado obtido na simulação está apresentado na figura 53.

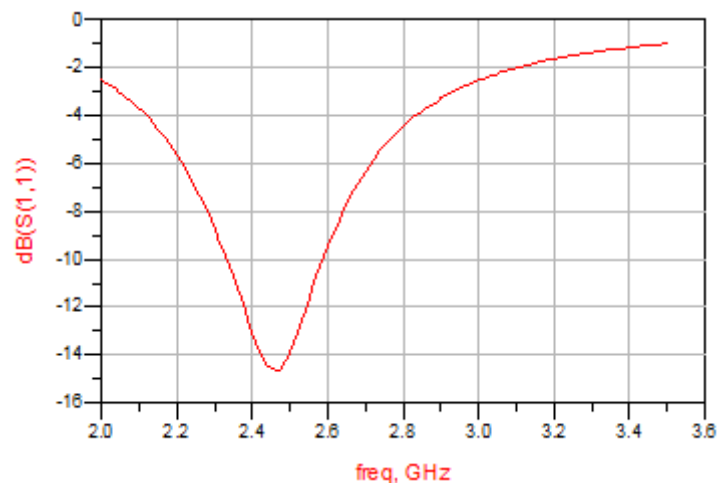


Figura 53: Casamento feito na entrada

A segunda análise, figura 54, corresponde ao parâmetro S22, que é o coeficiente de saída. Este, assim como o anterior, verifica a sintonia do casamento feito na saída do amplificador também na faixa de interesse. Observou-se que, a medida que a corrente de polarização ou o tamanho da transistor era incrementado, o vale se aproximava da frequência desejada. Entretanto, o casamento estava muito acima do esperado. A solução foi encontrada ao fazer análise da carta de Smith.

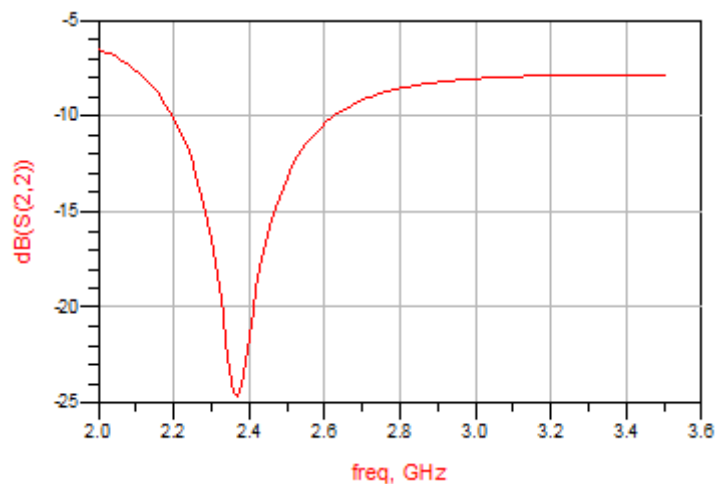


Figura 54: Casamento feito na saída

A carta de Smith é uma forma alternativa de analisar a impedância de saída, figura 55. Nela é possível analisar a impedância direto na frequência de interesse, sendo que a parte real multiplicada a Z_0 corresponde a resistência de entrada e a parte imaginária também multiplicada por Z_0 corresponde a reatância. Para diminuir a parte real da impedância há duas opções: aumenta o tamanho do transistor ou aumenta a corrente de polarização do segundo estágio. Assim, optou-se por modificar a corrente de polarização, para I_{D2} igual a 10 mA obteve-se uma resistência de 54Ω , o que representa uma boa resistência. Entretanto observou-se que a frequência estava no lado superior da carta, o que indicava a presença de uma componente indutiva na impedância de saída. Ao diminuir o valor do capacitor de saída de 10pF (que se comportava como capacitor ideal) para 2pF, o resultado foi como se adicionasse uma impedância capacitiva em serie com a saída ao circuito o que eliminou parte da indutância da saída. Como resultado o circuito ressoou na frequência correta e com ganho baixo na saída, conforme o esperado. Isto fez o circuito ressoar próximo da 2,4GHz com um ganho de -24 dB, o que era desejado.

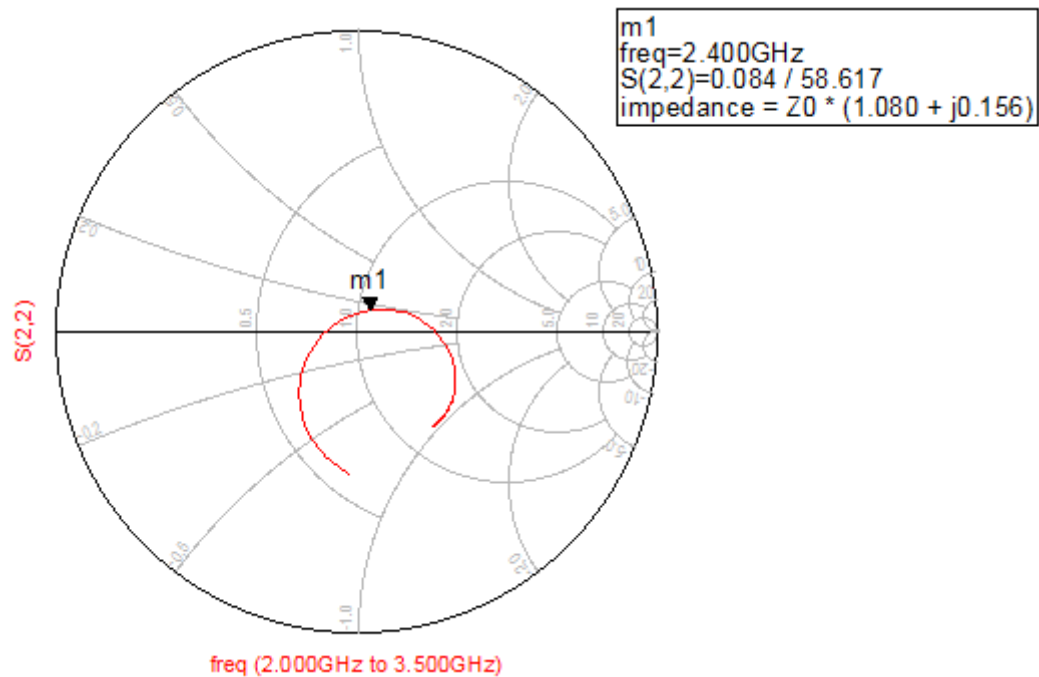


Figura 55: Carta de Smith

A análise do parâmetro S21 corresponde ao ganho direto do amplificador, figura 56, sendo possível observá-lo direto na frequência desejada. Inicialmente observou-se que o ganho estava baixo do desejado e que o mesmo não ressoava na frequência correta. Com a redução do capacitor ou do indutor, observou-se que o circuito ressoava mais próximo da frequência. Assim, adotou-se um capacitor de 0,5 pF que fez o circuito ressoar próximo de 2.4GHz com ganho de 18.5 dB, valor muito próximo do esperado que era de 20 dB.

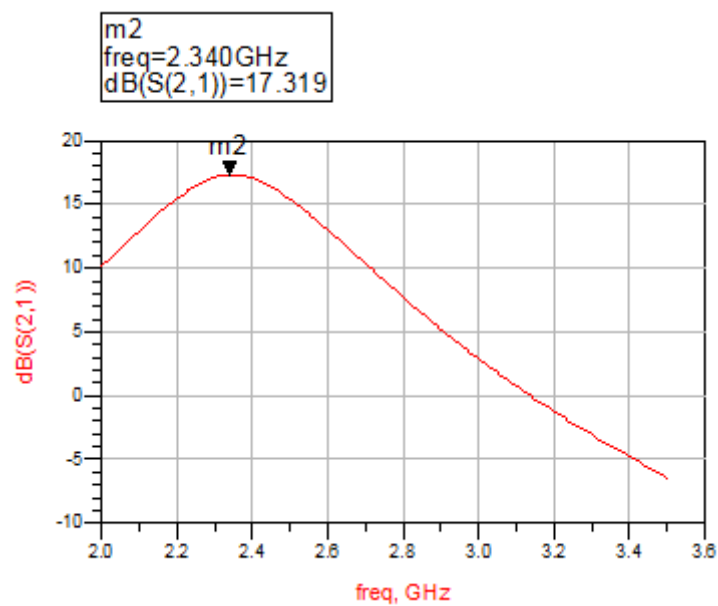


Figura 56: Ganho do amplificador

O último gráfico, figura 57, corresponde à figura de ruído do circuito. Que é considerada satisfatória, 2.2, já que se aproxima muito da esperada que era 2.

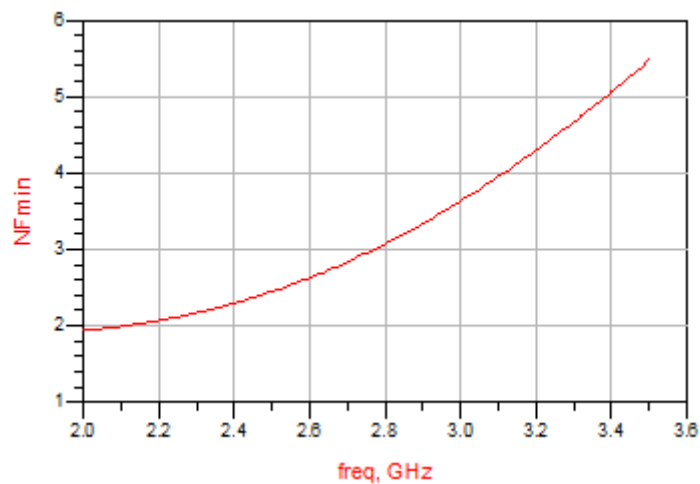


Figura 57: Representação da figura de ruído

5.3. VALIDAÇÃO CADEIA DE RECEPÇÃO

A cadeia de recepção de um transceptor é constituída basicamente por um amplificador de baixo ruído, um mixer que neste caso faz o translado do sinal

de RF para uma frequência menor (downconversion), um demodulador encarregado de recuperar a informação do sinal que chega, além dos filtros e do oscilador local.

Após a simulação e verificação do correto funcionamento dos principais blocos pertencente à cadeia, foi feita a modelagem da recepção completa no intuito de validar sistema como um todo, bem como a comunicação entre os módulos. A recepção implementada é composta por um LNA, seguido de um mixer que é acompanhado por um VCO, em seguida vem o filtro passa baixas e por fim o PGA. A figura 58 apresenta os componentes do símbolo gerado e a figura 59 o *testbench* construído para simular a recepção.

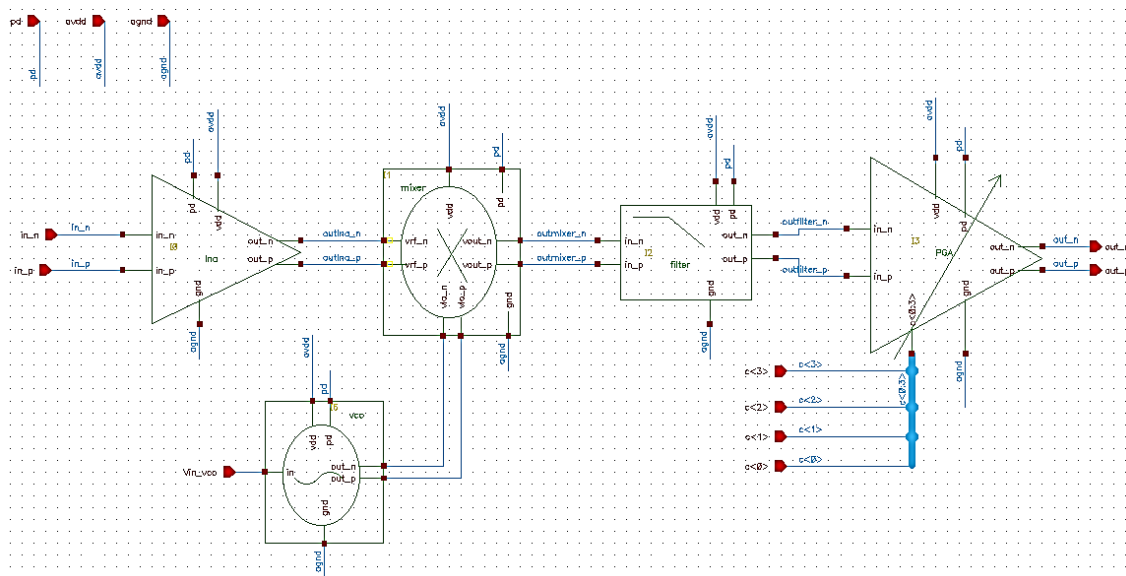


Figura 58: Componentes do símbolo de um sistema de recepção

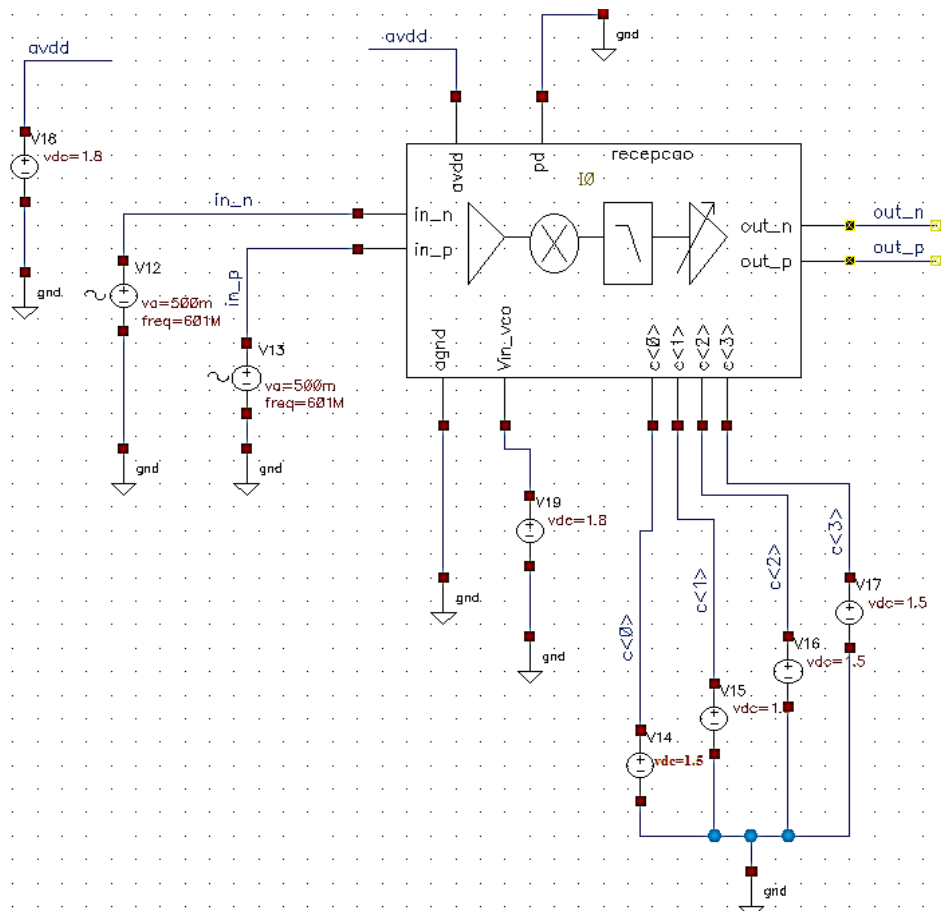


Figura 59: Testbench de um sistema de recepção

Optou-se por fazer o teste da recepção utilizando as especificações Zigbee, porém, primeiramente, com frequências da ordem da banda base e posteriormente com as frequências na ordem de operação do mesmo. Ressalta-se também, que a análise da resposta da recepção será feita acompanhando o comportamento bloco a bloco.

Sendo assim, na figura 58 é possível observar que o primeiro bloco da seção é o LNA. Ele foi configurado para o ganho de 4dB e tensão de saturação de 20V. A entrada do circuito é alimentada por uma onda senoidal de 500mV e frequência de 601MHz. Tanto o sinal de entrada quanto o sinal na saída do LNA podem ser observados na figura 60, logo abaixo.

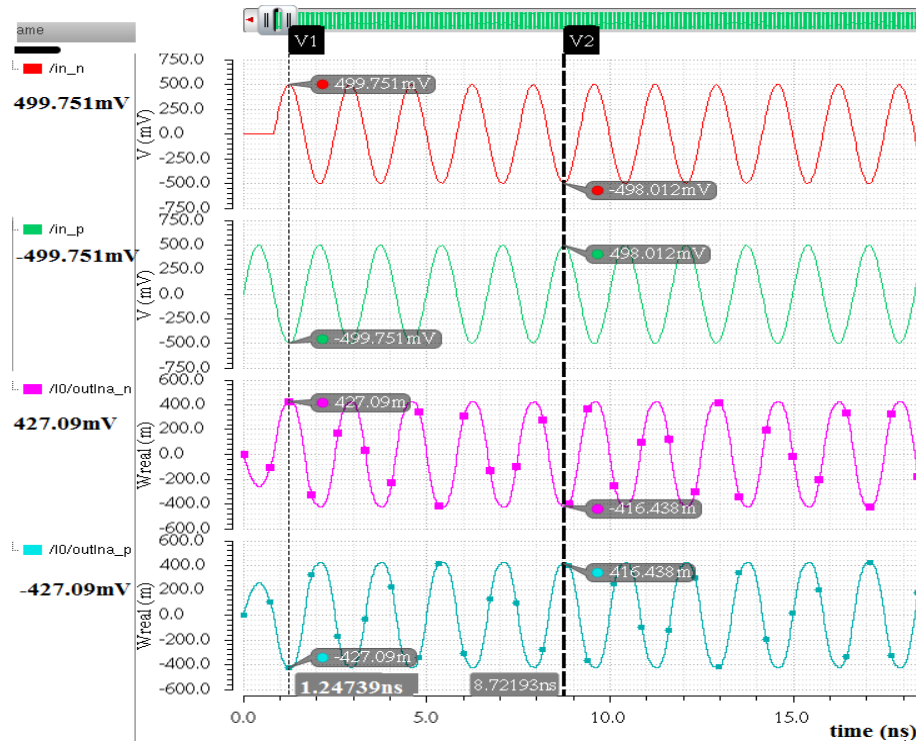


Figura 60: Resposta entrada da seção e saída do LNA

Assim,

$$ganho = 20 * \log \left(\frac{427.09 * 10^{-3} * 2}{499.751 * 10^{-3}} \right) = 4.6 \quad (61)$$

$$frequência = \frac{1}{T} \quad (62)$$

$$T = \frac{1.24739 * 10^{-9} * 4}{3} = 1.6631 * 10^{-9} \quad (63)$$

$$frequência = \frac{1}{1.6631 * 10^{-9}} = 601.25 MHz \quad (64)$$

Os valores calculados mostram que o ganho obtido se aproxima do desejado e que a frequência encontrada nada mais é que a frequência inicial inserida no circuito.

Em seguida é feita a análise da resposta do segundo bloco, o mixer. Ele recebe na entrada RF o sinal proveniente do LNA e na entrada OL o sinal proveniente do VCO. O VCO está configurado com uma tensão mínima de 0V, tensão máxima de 1.8V, amplitude do sinal de saída de 1V, frequência de

600MHz. Enquanto o mixer possui os parâmetros definidos por: ganho de 1dB, tensão de saturação de 10V e nível dc de 0V. O resultado pode ser visto na figura 61.

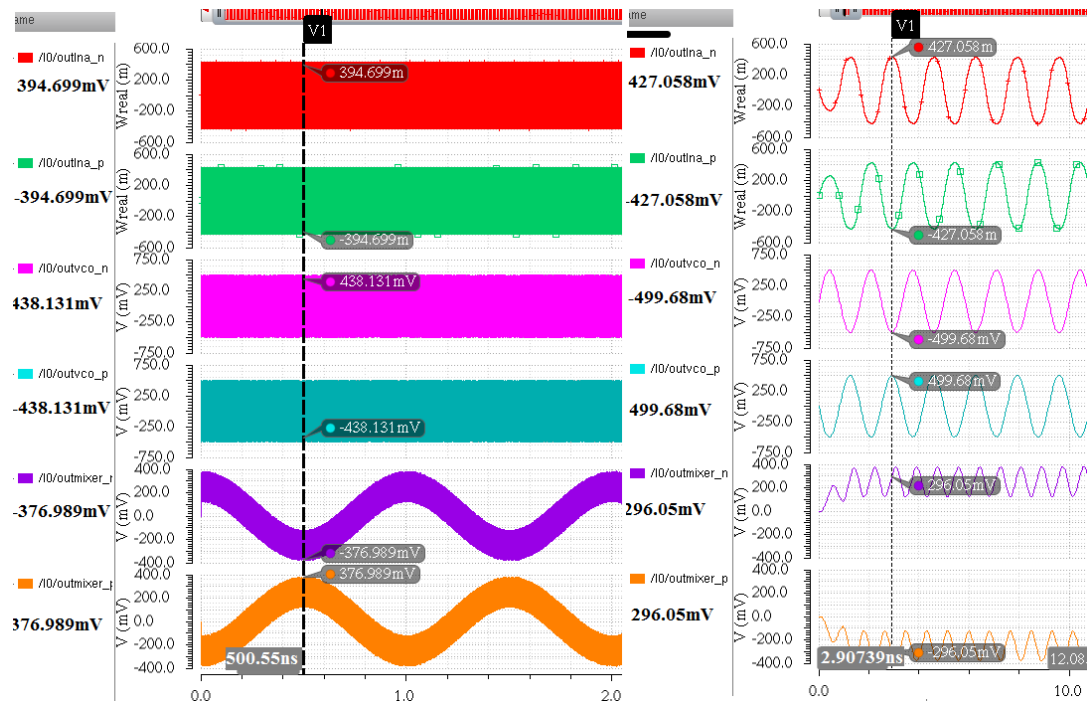


Figura 61: Resposta das saídas do LNA, VCO e Mixer

É possível observar que o mixer promove um *downconversion* com os sinais de entrada provenientes do LNA e VCO. A variação observada na primeira figura configura a formação da portadora, ocorre a constante mudança da frequência central do mixer devido aos cálculos de soma e subtração das frequências de entrada.

O seguindo o fluxo do sinal, o próximo bloco a ser analisado é o filtro. Ele possui como entrada o sinal resultante do mixer. O filtro projetado continua sendo um butterworth de segunda ordem com frequência de 20MHz. Observando a figura 62, é possível ver nitidamente a atuação do filtro. Uma vez que ele promove toda a filtragem do sinal rejeitando as frequências indesejadas.

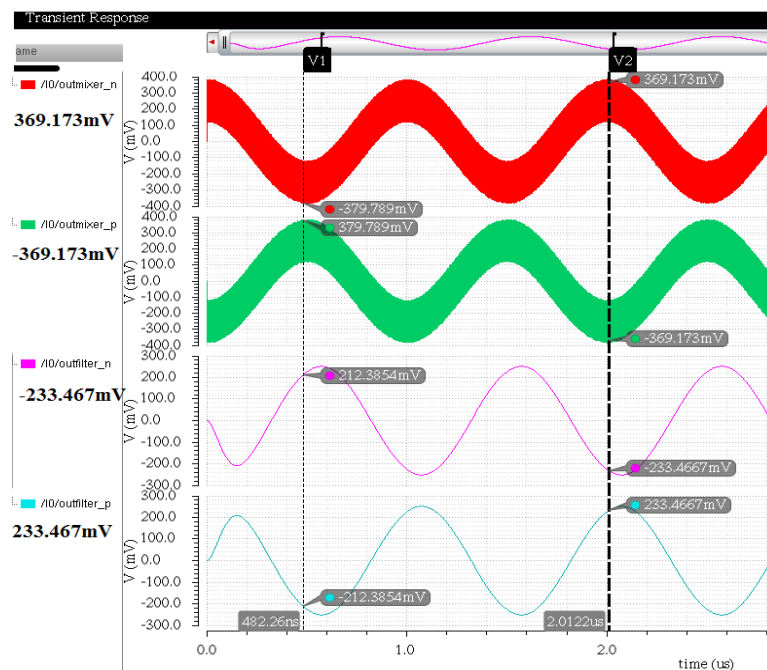


Figura 62: Resposta da saída mixer e da saída filtro

Por fim, tem-se a amplificação do sinal promovida pelo PGA. A figura 63, contém as curvas de simulação do sinal na saída do filtro, que alimentará o PGA, e na saída do PGA, que corresponde a saída da cadeia.

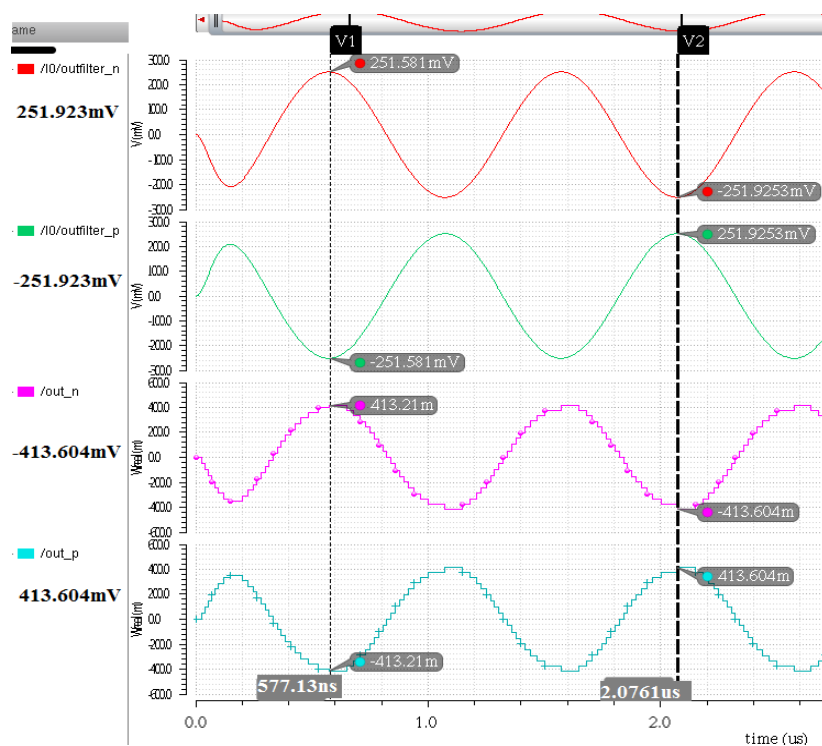


Figura 63: Resposta da saída filtro e da saída PGA

O PGA está configurado para inserir o ganho máximo de 10dB no sinal. A verificação do ganho aplicado ao sinal pode ser verificado da seguinte forma:

$$ganho = 20 * \log\left(\frac{413.21 * 10^{-3} * 2}{251.581 * 10^{-3}}\right) \approx 10.3 \quad (65)$$

$$T = \frac{[(2.07619 * 10^{-6}) - (577.13 * 10^{-9})] * 4}{6} = 9.9931 * 10^{-7} \quad (66)$$

$$frequência = \frac{1}{9.9931 * 10^{-7}} \approx 1MHz \quad (67)$$

Observa-se que o ganho inserido está próximo do estipulado e que o sinal se encontra na frequência de banda base.

O resultado final mostra o perfeito funcionamento da cadeia de recepção. Uma vez que um sinal de baixa amplitude é inserido no LNA, os ganhos são devidamente inseridos no decorrer da cadeia e o *downconversion* é aplicado de forma precisa ao sinal. Ressalta-se que o sinal de entrada de 500mV é considerado alto para o LNA, uma vez que ele é crítico podendo influenciar de forma expressiva os demais blocos ou mesmo saturar, entretanto, para a tecnologia AMS promover as conexões e sínteses é necessário que o sinal esteja acima de 100mV, sendo este valor já o limite para LNA.

5.4. VALIDAÇÃO CADEIA DE TRANSMISSÃO

A cadeia de transmissão foi construída com os blocos testados individualmente. Ela tem por objetivo tratar um sinal pequeno, lhe inserindo ganho, potência e transladando sua frequência, de maneira que o mesmo possa ser transmitido com força suficiente para que a recepção seja capaz de identifica-lo sem perdas, trata-lo e extrair a informação ali contida. A simulação será feita no sentido validar a metodologia de modelagem a desenvolvida, mostrando a perfeita comunicação entre os blocos projetados.

A seção de transmissão de um transceptor é constituída basicamente por um filtro passa baixa, por um mixer que neste caso irá transladar um sinal em banda base para uma frequência maior RF (*upconversion*), um VCO e um

amplificador de potencia PA. A figura 64 apresenta os blocos constituintes do símbolo gerado, e a figura 65 apresenta o *testbench* para a simulação do sistema de transmissão.

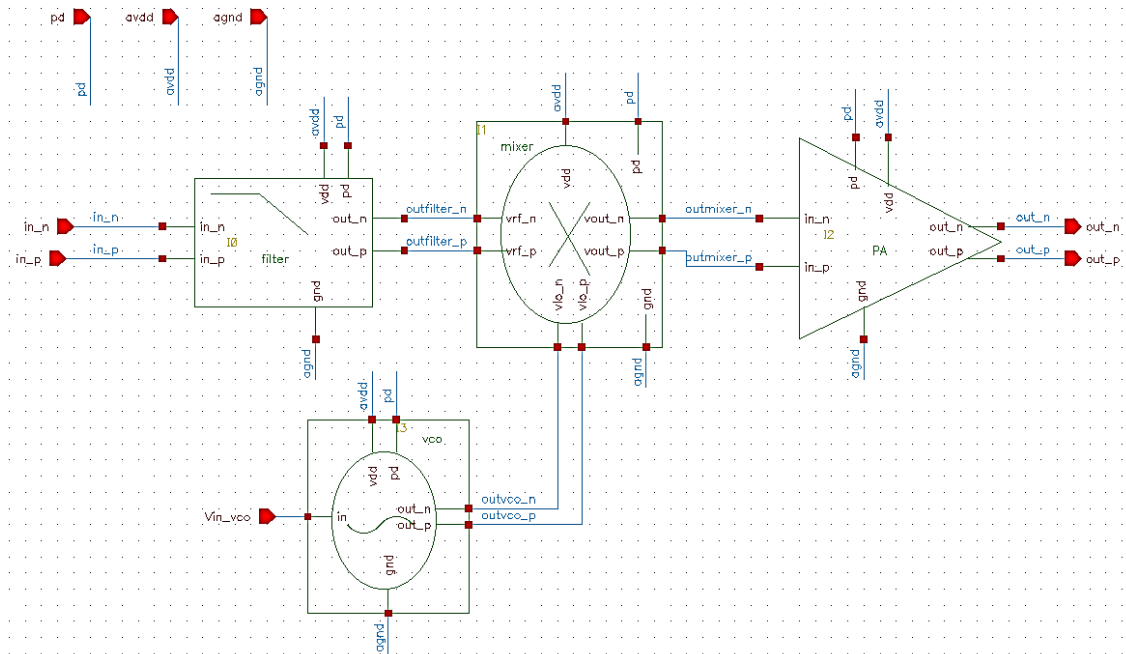


Figura 64: Componentes do símbolo de um sistema transceptor

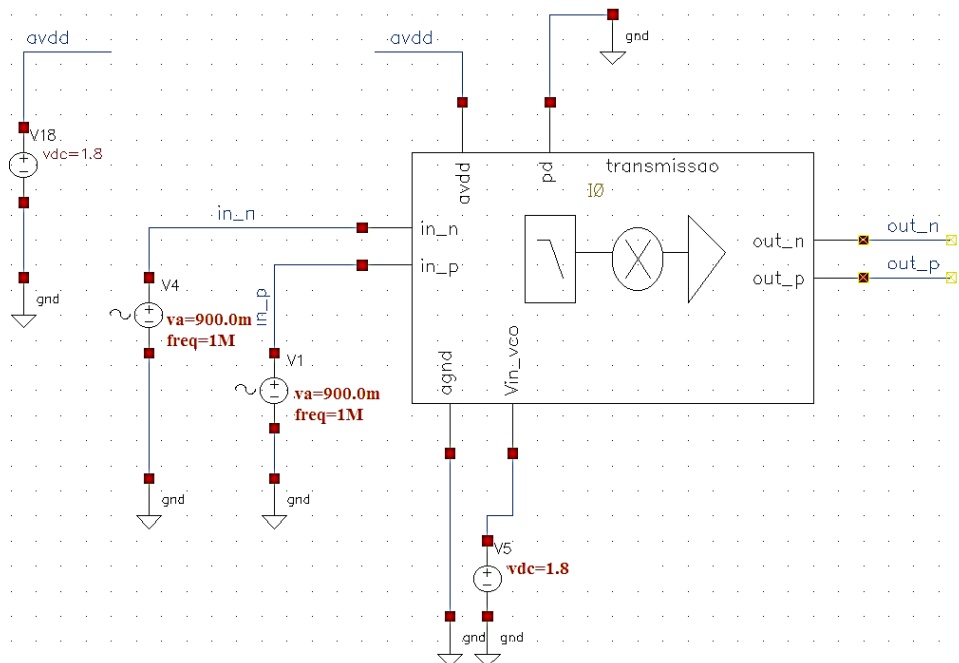


Figura 65: *Testbench* de um sistema transceptor

Nos testes da cadeia de transmissão optou-se por utilizar as especificações zigbee, porém, num primeiro momento as simulações irão ocorrer em uma frequência abaixo da faixa de operação Zigbee sendo definida, então, por 600MHz.

O primeiro bloco da cadeia de transmissão é o filtro, figura 66. Ele foi projetado utilizando o polinômio de butterworth de segunda ordem e frequência de 20MHz. O circuito é alimentado utilizando duas fontes senoidais de 900mV e frequência de 1MHz, sendo uma defasada da outra em 180°.

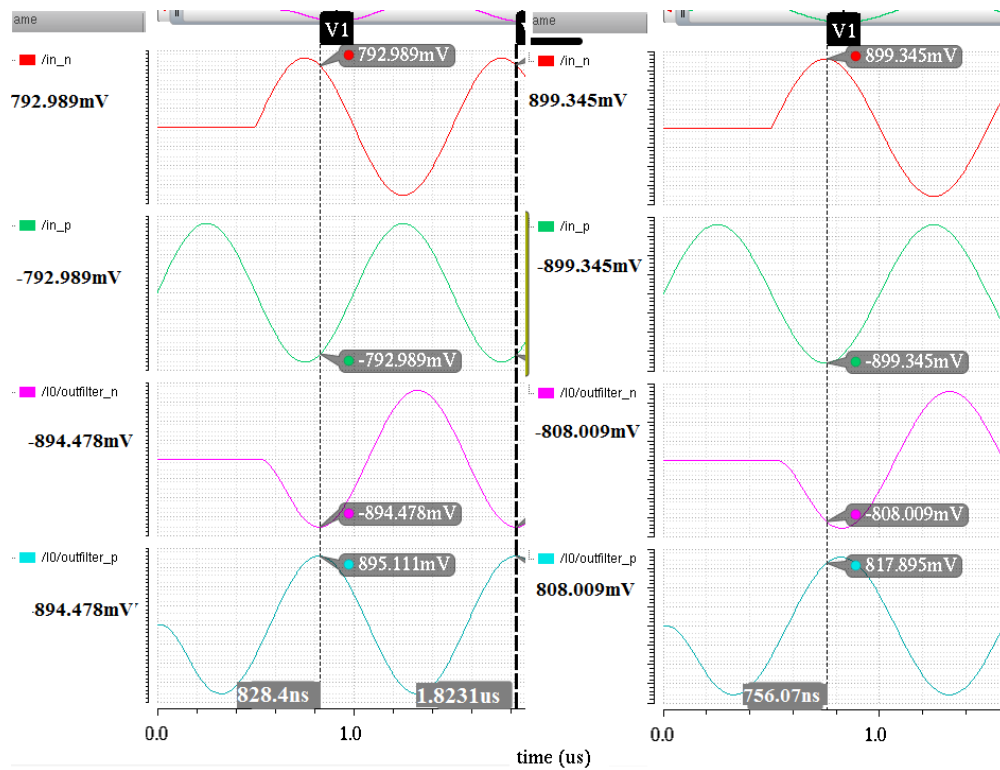


Figura 66: Resposta da entrada da seção e saída do filtro

Como já foi explicado, a frequência utilizada no projetado do filtro permite a passagem integral de sinais a 1MHz. Como pode ser observado na imagem, foi mantida a amplitude do sinal em aproximadamente 895, 11mV e também a frequência.

$$T = (1.8231 * 10^{-6}) - (828.4 * 10^{-9}) = 9.947 * 10^{-7} \quad (68)$$

$$frequência = \frac{1}{T} = \frac{1}{9.947 * 10^{-7}} \approx 1MHz \quad (69)$$

O sinal flui, então, para o segundo bloco, o mixer. Este receberá em sua entrada OL o sinal proveniente do VCO. A tensão de controle do VCO foi estabelecida na máxima amplitude o que resulta em uma frequência de 599MHz. A figura 67 evidencia o sinal gerado pelo VCO.

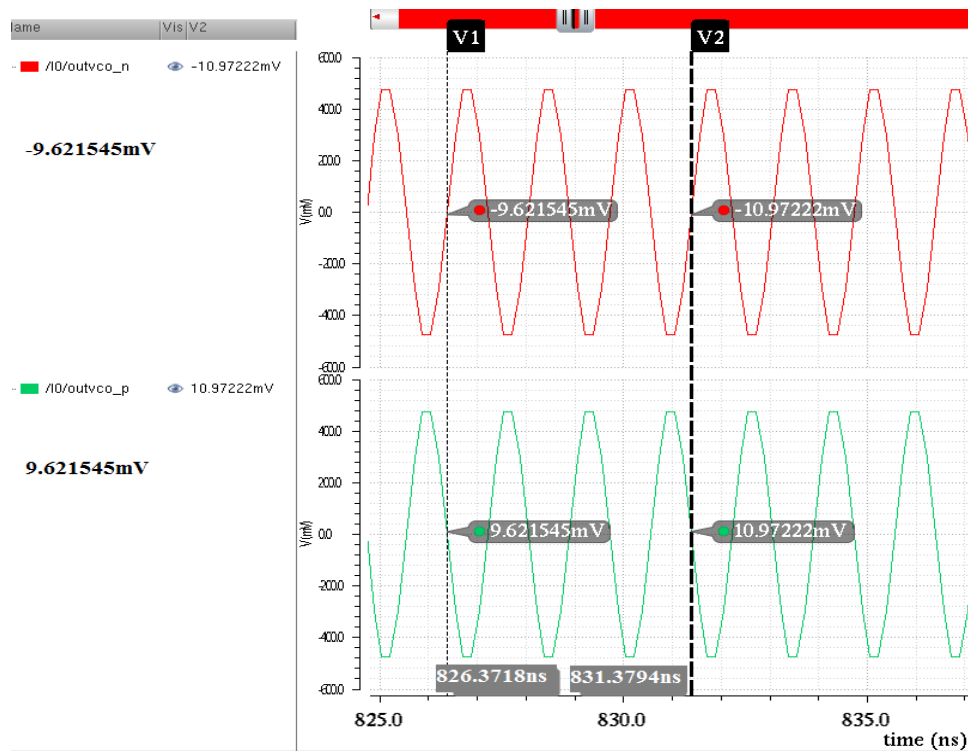


Figura 67: Resposta do VCO

A frequência fornecida é dada por:

$$T = \frac{(831.3794 * 10^{-9}) - (826.3719 * 10^{-9}) * 4}{11} = 1.6832 * 10^{-9} \quad (70)$$

$$frequência = \frac{1}{T} = \frac{1}{1.6832 * 10^{-9}} \approx 594.106MHz \quad (71)$$

O que está de acordo com o esperado.

O mixer, por sua vez, foi configurado para obter um ganho de 1dB, nível dc de 0V e tensão de saturação de 10V. A figura 68 mostra o resultado da simulação deste bloco, que receberá em uma entrada o sinal fornecido pelo filtro e na outra o sinal fornecido pelo VCO.

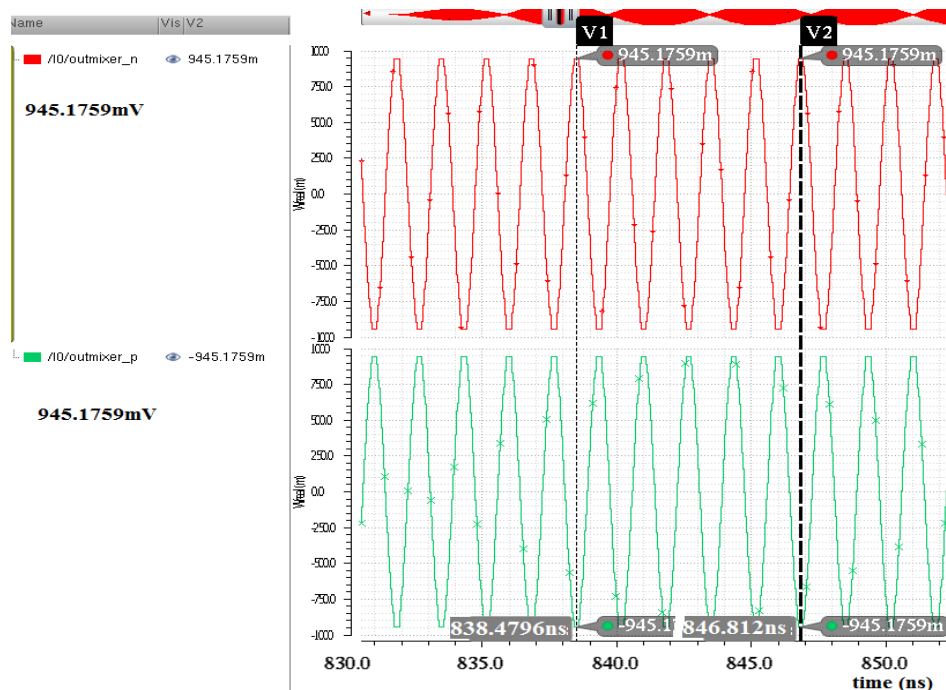


Figura 68: Resposta do Mixer

O mixer tem a obrigação de transladar a frequência do sinal para uma frequência maior, que no caso deve ser de 600MHz. O calculo para análise dos resultados é apresentado a seguir:

$$T = \frac{(846.812 * 10^{-9}) - (838.4796 * 10^{-9}) * 4}{20} = 1.66648 * 10^{-9} \quad (72)$$

$$frequência = \frac{1}{T} = \frac{1}{1.66648 * 10^{-9}} \approx 600.067MHz \quad (73)$$

O resultado acima comprova a funcionalidade do bloco, uma vez que os valores encontrados ficaram bem próximos dos especificados.

A última posição da cadeia de transmissão pertence ao PA. Ele deverá inserir uma potência suficientemente alta ao sinal para que o mesmo seja transmitido. Ele foi configurado para uma tensão de saturação de 0.9V e um ganho máximo de 20dB. O resultado obtido pode ser observado na figura 69.

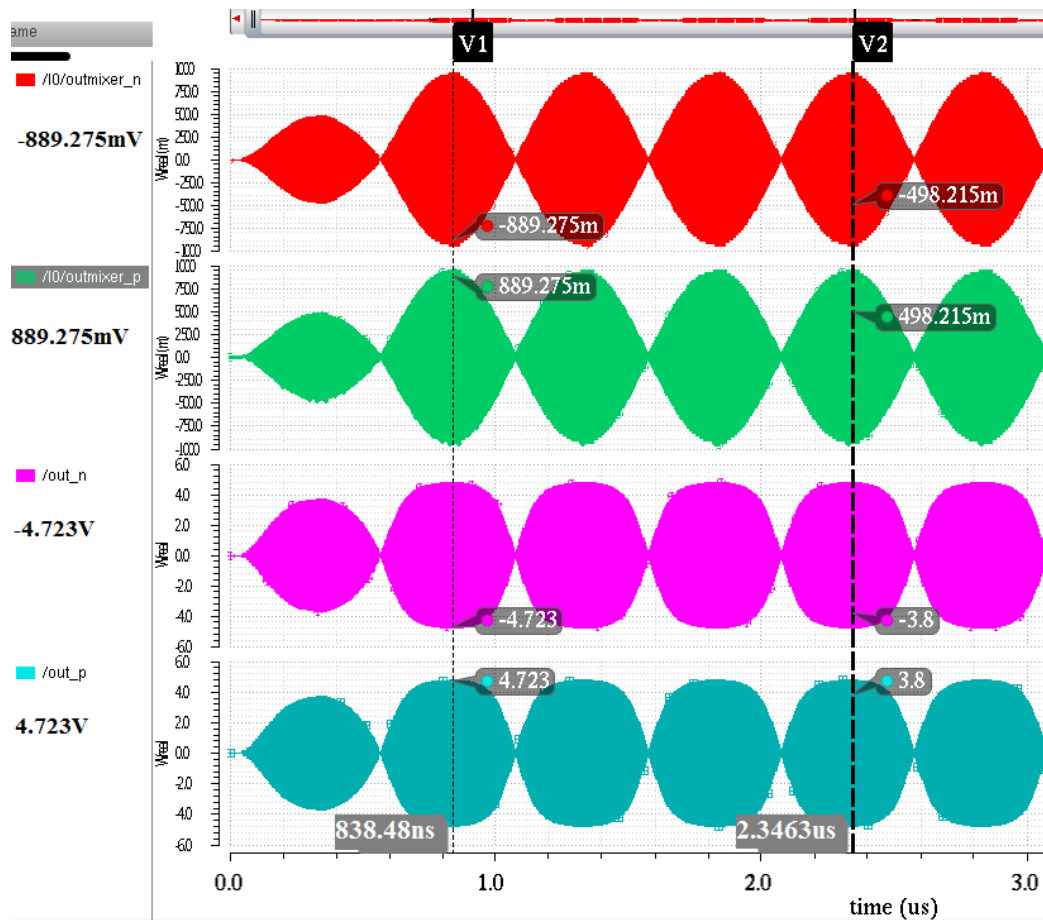


Figura 69: Resposta saída Mixer e saída PGA

A validade do ganho é verificada, com o seguinte cálculo:

$$ganho = 20 * \log\left(\frac{4.723 * 2}{889.275 * 10^{-3}}\right) \approx 20.52 \quad (74)$$

A figura 70 mostra nitidamente o sinal de saída. Com o mesmo é possível verificar a frequência de saída da seção transmissora.

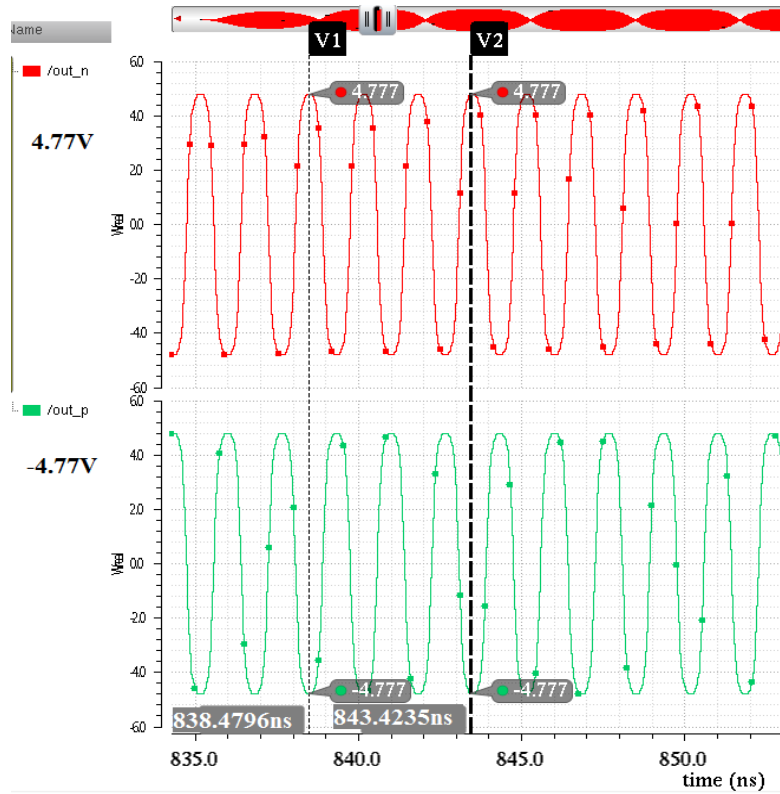


Figura 70: Resposta saída da cadeia de transmissão

$$T = \frac{[(843.4235 * 10^{-9}) - (838.4796 * 10^{-9})] * 4}{12} = 1.65966 * 10^{-9} \quad (75)$$

$$frequência = \frac{1}{9.9931 * 10^{-7}} \approx 602MHz \quad (76)$$

Observa-se que os valores encontrados ficaram bem próximos do esperado, o que valida a funcionalidade da cadeia de transmissão como um todo.

5.5. SIMULAÇÃO MISTA

Para obter a simulação mista optou-se então por substituir o filtro implementado em verilog-AMS por um novo filtro projetado de maneira simples em nível de transistor. Para alcançar as especificações propôs-se a função de transferência do filtro por uma aproximação do polinômio de *Butterworth* de segunda ordem. A topologia escolhida para a implementação foi a de Sallen-Key ou filtros Vcvs de acordo com o esquema elétrico da Figura 71.

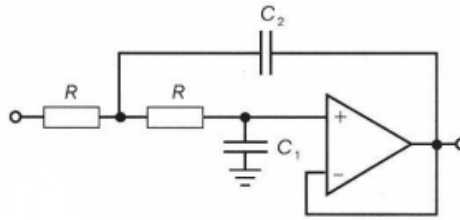


Figura 71: Esquemático elétrico topologia Sallen-Key

A função transferência referente ao modelo foi obtida da seguinte maneira:

$$\frac{v_{in} - v_1}{r} = \frac{v_1 - v_{out}}{1/sc} + \frac{v_1 - v_{out}}{r} \quad (77)$$

$$\frac{v_{in} - srcv_{out}}{r} = \frac{srcv_{out}}{1/sc} + \frac{srcv_{out}}{r} \quad (78)$$

$$\frac{v_{in}}{r} = \frac{v_{out}(src + 1 + s^2r^2cc + scr)}{r} \quad (79)$$

$$\frac{v_{out}}{v_{in}} = \frac{1}{(s^2r^2c^2 + sc(r + r) + 1)} \quad (80)$$

Para encontrar a função transferência com suas respectivas especificações utilizou-se o software Matlab, de acordo com a figura 72.

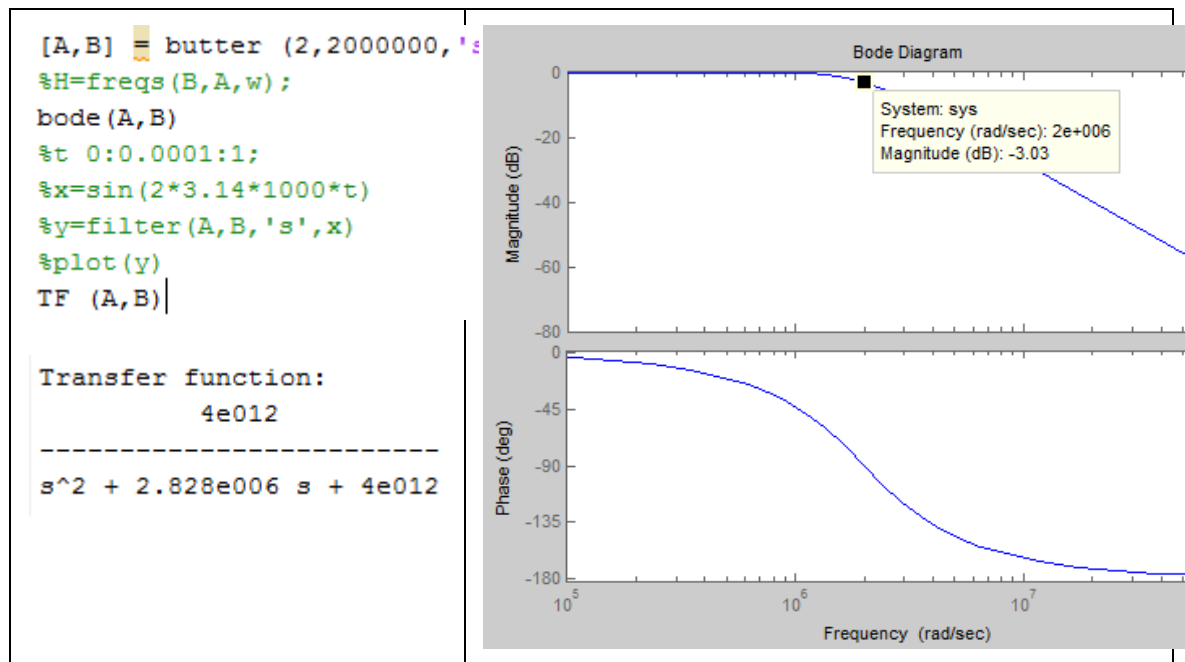


Figura 72: Função no Matlab para filtro de ordem 2

Sendo assim a função transferência do filtro obtida foi:

$$H_s = \frac{4.10^6}{s^2 + s.2,824.10^3 + 4.10^6} \quad (81)$$

Com base nesses resultados foi possível projetar o bloco do filtro no *software* Cadence. O projeto abrange os valores dos resistores e capacitores. Para a implementação, para teste de eficiência da função de transferência obtida utilizou-se um amplificador ideal, como mostrado na figura 73.

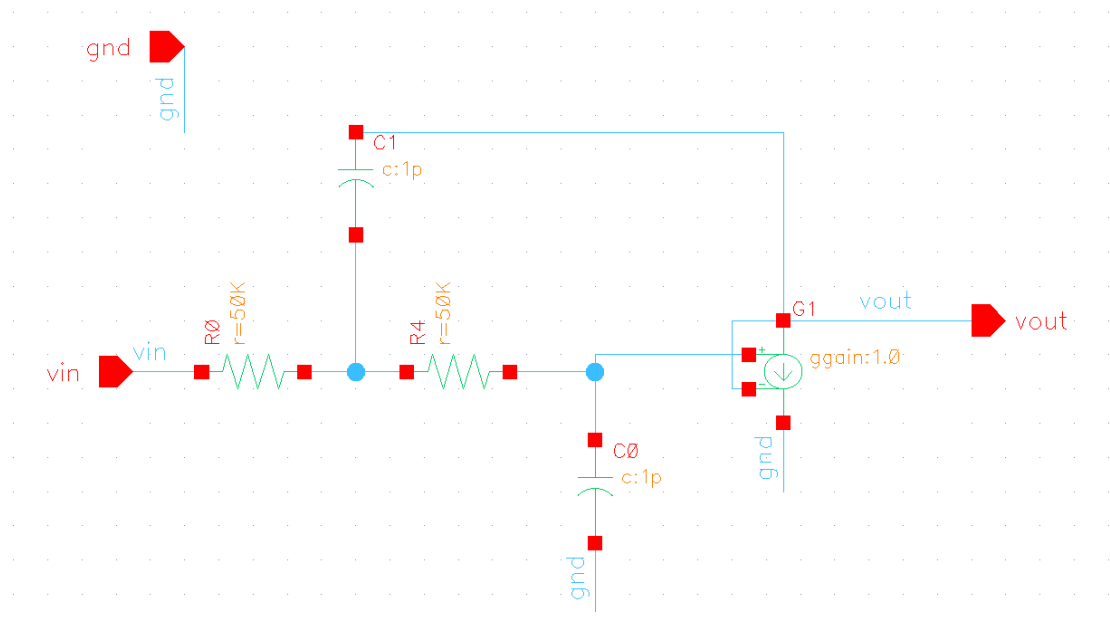


Figura 73: Esquemático elétrico do filtro no *software* Cadence.

Assim, com a implementação do filtro foi possível testar a simulação mista na frequência 2.4GHz, ou seja, agora utilizando todas as especificações Zigbee.

Desta maneira foi feita a união da cadeia de transmissão com a cadeia de recepção sendo estas intermediadas por um atenuador para conformar a entrada do sinal na recepção. Nesta simulação todos os blocos foram projetados em verilog-AMS exceto o filtro, que está em nível de transistor, uma vez que o objetivo é validar a capacidade do projeto ser utilizado para simulações mistas. A figura 74 apresenta os resultados.

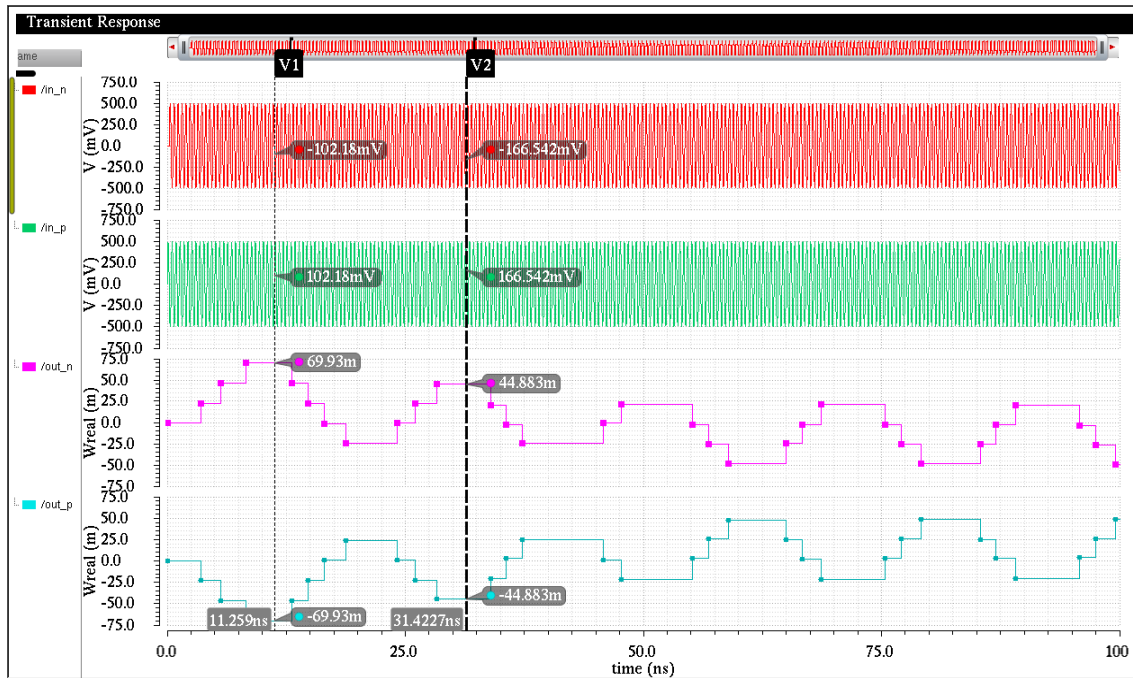


Figura 74: Resultado da simulação mista do sistema transceptor completo

O gráfico da figura 74 demonstra a perfeita comunicação e interrelação entre os blocos projetados tanto em nível mais abstrato, linguagem de descrição de hardware, quanto em nível menos abstrato, nível de transistor. O resultado foi o perfeito funcionamento da cadeia completa do transceptor Zigbee, validando assim a metodologia utilizada.

6. CONCLUSÕES

Este capítulo promove o fechamento do projeto expondo assim as conclusões obtidas e as propostas para a segunda parte.

6.1. CONCLUSÃO

A confecção de um transceptor Zigbee envolve o domínio de uma gama de conceitos na área de radiofrequência. Por se tratar de uma área totalmente nova, foi necessário promover um estudo aprofundado em Eletrônica de Rádio Frequência e em Modelagem de Circuitos RF utilizando Verilog-AMS oferecido na Escola de Microeletrônica do Distrito Federal. Diante disto, a maior parte deste trabalho se concentrou em apresentar um amplo estudo de temas referentes aos elementos básicos necessários ao entendimento do tema proposto na busca de uma ferramenta de auxílio ao desenvolvimento do projeto.

Na revisão bibliográfica foram abordados conceitos a respeito de sistemas de comunicação sem fio baseados no padrão Zigbee / IEEE 802.15.4, na qual foram expostos os fundamentos e as vantagens deste com relação aos demais padrões para a aplicação proposta, em que se evidenciou atender muito bem os requisitos para as redes de monitoramento e controle sem fio. Posteriormente foram feitos estudos a respeito de linguagens de descrição de hardware para a modelagem do projeto destacando-se a linguagem Verilog-MAS. Esta apresenta como principal vantagem, perante as demais, ser capaz de trabalhar com sistemas mistos possibilitando maior facilidade perante circuitos complexos, descrevendo os mais variados comportamentos, possibilitando a decomposição hierárquica do projeto, melhorando a verificação de todo sistema, disponibilizando simuladores e menor tempo entre modelagem e confecção dos chips. Por fim, foram estudados conceitos e características de transceptor RF, considerando a funcionalidade de seus principais blocos internos.

Com a relação a melhor metodologia a ser utilizada para o desenvolvimento do transceptor RF, o estudo realizado demonstrou que a *Top-down* proporciona resultados bem imediatos dando uma base melhor para o projeto do sistema completo, uma vez que com modelos de alto nível executáveis em mãos é possível executar a etapa de testes juntamente com a modelagem do circuito e montagem dos blocos.

Na primeira parte da confecção do projeto, foi feita a implementação em baixo nível de um importante bloco que compõem o módulo de recepção, o Amplificador de Baixo Ruído (LNA), na ferramenta *Advanced Design System* (ADS). O LNA foi realizado em uma topologia diferencial fonte-comum *cascode* com degeneração indutiva e carga LC ressonante. A avaliação do modelo foi feita configurando o bloco com parâmetros reais utilizados na confecção comercial do mesmo atualmente

Os resultados obtidos podem ser considerados satisfatórios atendendo as especificações básicas de um LNA Zigbee, embora outras contribuições no modelo possam ser acrescentadas num futuro para uma melhora ainda maior do seu desempenho, em que análises de outras especificações não verificadas poderão ser realizadas como: ponto de interseção de terceira ordem (IIP_3), ponto de compressão de 1dB (P_{1dB}), índice de rejeição da fonte de alimentação (PSRR).

A segunda etapa do projeto corresponde à modelagem dos principais blocos da seção RF de um transceptor numa arquitetura mais abstrata utilizando o software *Cadence*, sendo eles: LNA, Mixer, VCO, PGA, Filtro e PA. Foi feita a modelagem funcional de cada bloco utilizando a linguagem Verilog-AMS. As simulações feitas geraram resultados satisfatórios a todos os blocos, em que se evidenciou a funcionalidade individual dos mesmos. Posteriormente foi feita a confecção tanto de uma seção de recepção quanto de uma transmissão. A simulação e posteriormente análise dos resultados evidenciaram a capacidade de interação dos blocos validando assim a metodologia utilizada. Os modelos foram testados utilizando parâmetros Zigbee, porém a uma frequência menor que a especificada pela IEEE 802.15.4.

Isto se deve a problemas gerados pela linguagem Verilog-A, já que esta é normalmente utilizada para modelagens de filtros. Observou-se durante as tentativas de simulação em altas frequências que a função específica para este fim, não funciona de forma eficiente com frequências na faixa de giga hertz, o que inviabilizou a modelagem do transceptor Zigbee. Como consequência, a simulação mista não pode ser feita uma vez que os cálculos de todos os parâmetros utilizados na implementação em baixo nível do LNA foram realizados sob a frequência de 2.4GHz.

6.2. PROPOSTAS PARA TRABALHOS FUTUROS

Como proposta para trabalhos futuros, tem-se a continuação do projeto dos blocos da seção RF na busca das especificações mais profundas não realizadas no projeto. Assim como, um estudo detalhado das vantagens e desvantagens das topologias do sistema Zigbee completo que são utilizadas atualmente, visando a escolha da que melhor se adéqua a proposta do projeto.

A modelagem sistêmica poderá ser feita na busca de características comportamentais e funcionais que darão suporte a implementação dos subsistemas de forma mais detalhada. Assim, os módulos de RF poderão ser projetados em nível de transistor atendendo as especificações determinadas pelo protocolo Zigbee. Para tanto, para que simulações mistas possam ser feitas utilizando o LNA para promover a verificação dos blocos diante da aplicação das especificações gerais e na busca da otimização do sistema, é preciso que o problema do filtro seja solucionado. Por fim, propõem-se testes nos modelos e circuitos, e validação dos mesmos a fim de que possíveis projetos de layout sejam feitos para fabricação.

4. REFERÊNCIAS BIBLIOGRÁFICAS

- Accellera – “Verilog-AMS Language Reference Manual”, 2003. Analog and mixed signal extensions to Verilog HDL, version 2.1.
- Aisola R. “ Verilog – AMS Language Reference Manual” v. 2.1, Accellera, 2003.
- Braga, T. C. “Monitorização Ambiental em Espaço Florestal com Rede de Sensores Sem fios.” Dissertação de mestrado. Universidade da Madeira, 2010.
- Dr. Ahamed, S. S. “The Role of Technology in Future Data Communication System” Publicado em: Jornal of Theoretical and Applied Information Technology, 2005.
- Evangelista, D. S. “Integração de Redes de Sensores Zigbee para Automação predial Utilizando Módulos Mehbean”. Monografia de graduação. Universidade de Brasília, Faculdade de Tecnologia, 2010.
- Farahani, Shahin. “Zigbee Wireless Network and Transceivers”. Burlington: Newnes, 2008.
- Ferreira, F. J. F. – “Projeto de um Misturador em tecnologia CMOS de $0,25\mu\text{m}$ ”, 2006. Dissertação de mestrado. Universidade Federal de Itajubá.
- Fuentes, E. F. R. “Metodologia de Projeto para Amplificador de Baixo Ruído em CMOS”. Dissertação de mestrado. Universidade de São Paulo, 2003. Disponível em: <http://www.lsi.usp.br/~dmprsv/download/Disserta%E7%E3o-LNA.pdf>. Acessado em: outubro 2013.
- Gislason, D. – “Zigbee Wireless \networking”, Newnes, 2008.
- IEEE, Standard (2006) – “Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low – Rate Wireless Personal Area Networks (WPANs)”, IEEE Computer Society, Revision of IEEE Std 802.15.4-2003.

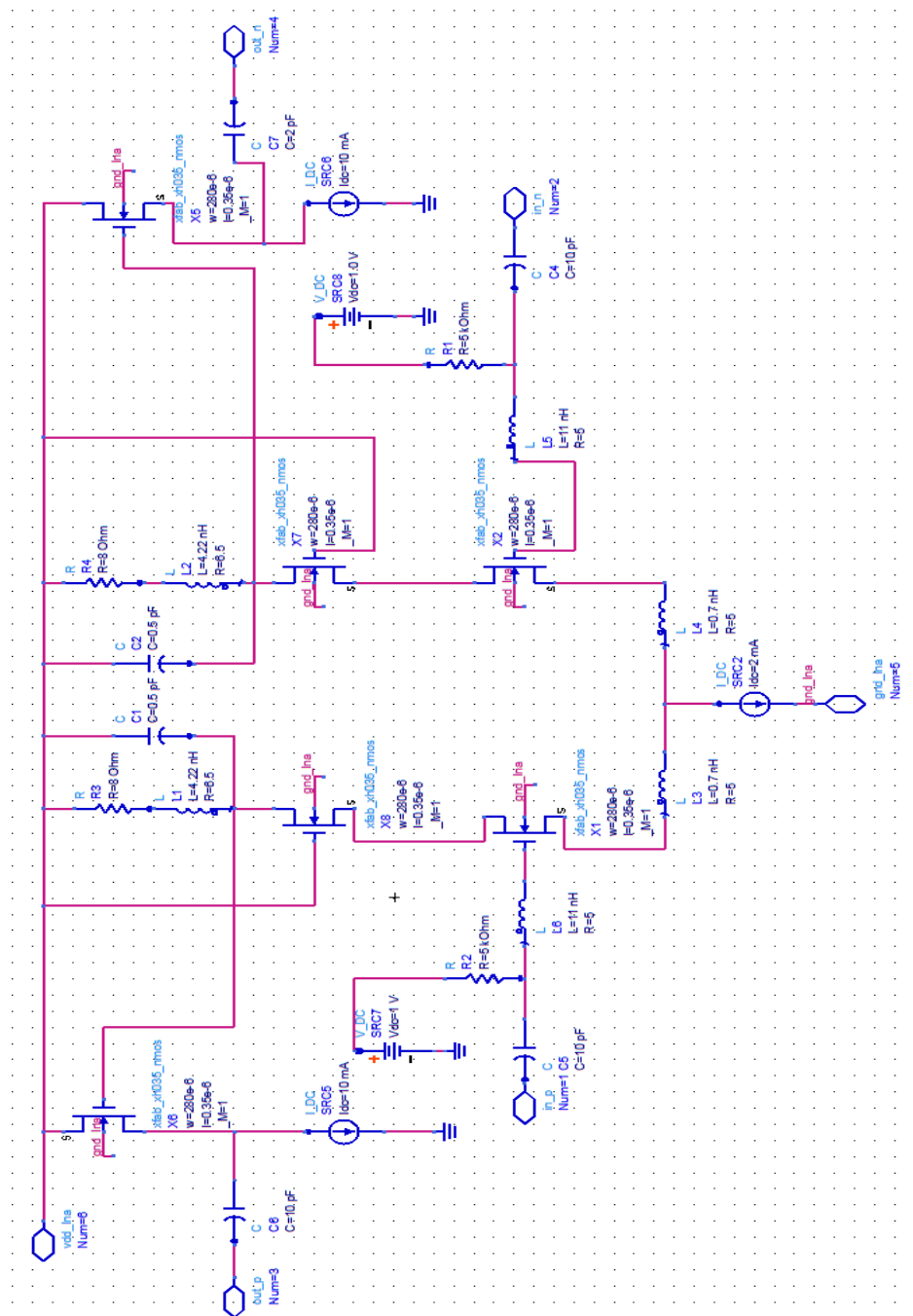
- Kanna, R. "Design of Zigbee Transceiver for IEEE 802.15.4 Using Matlab/Simulink", 2011. Thesis submitted in partial fulfilment of the requirements for the degree of master of technology in telematics and signal processing.
- Kundert, k. S. – "The designer's Guide To Verilog-AMS", 2004. First edition, Consulting editor.
- Kundert, K. S; Zinke, O. "The Designer's Guide to Verilog-AMS", 1. Ed. [S.1]: kluwer Academic Publishers, 2004.
- Lee, T. "The design of CMOS radio-frequency integrated circuits." [S.1]: Cambrige Univ Pr, 2004.
- Malafaya, H., Tomás, L., Sousa, J. P. – "Sensorização sem fios sobre Zigbee e IEEE 802.15.4". Departamento de Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto.
- Midorikama, E. T. "Uma Introdução às Linguagens de Descrição de Hardware", Escola Politécnica da Universidade de São Paulo, 2001.
- Moraes, Elisabete Nakoneczny "Implementação de um modelo do transistor MOS em VERILOG-AMS" (Dissertação de mestrado) – Setor de Tecnologia -Universidade Federal doPará - Curitiba, 2008.
- Moraes, G. D. S D. – "Amplificador de Ganho Programável Integrado para Aplicação em Medidores de Energia Elétrica Inteligentes.", 2012. Dissertação de mestrado. Universidade Federal do Rio de Janeiro.
- Nascimento, B. "Modelagem em Alto Nível da Seção de Recepção de um Transceptor RF", 2010. Relatório de graduação. Universidade de Brasília.
- Pêcheux F. et all. "VHDL-AMS and Verilog-AMS as Alternative Hardware Description Languages for Efficient Modeling of Multi-Discipline Systems". Publicado em: IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, v. 24, num. 2, p 204-225, 2005.
- Razavi, B. "RF Microeletronics." 1 ed. [S.1.]: Prentice Hall PTR Upper Saddle River, NJ, 1998.

- Ribas, R. P. “Microssistemas Integrados (MEMS)”. Universidade Estadual de Campinas, 2000. Disponível em: <http://www.ccs.unicamp.br/crusos/fee107-/download/cap14.pdf>. Acesso em: outubro 2013.
- Silva, A. T. D – “Módulos de Comunicação Wireless para Sensores”, 2007. Faculdade de Engenharia da Universidade do Porto.
- Silva, I. M. D. – “Análise de desempenho de sistema de comunicação sem-fio para monitoramento de unidade de produção de poços petrolíferos terrestres”, 2008. Dissertação de mestrado. Universidade Federal do Rio Grande do Norte.
- Soares, V. F. “Projeto de Módulos de RF para Sistema em Chip CMOS.” Universidade de Brasília, 2008. Disponível em: [http://bdm.bce.unb.br/bitstream/10483/866/1/2008_V%-C3%ADtorFonseca Soares.pdf](http://bdm.bce.unb.br/bitstream/10483/866/1/2008_V%-C3%ADtorFonseca%20Soares.pdf). Acessado em: outubro 2013.
- Sohraby, Kazem, Daniel Minoli e Taieb Znati. “Wireless Sensor Networks: Technology, Protocols, and Applications, Wiley-intrescience”, 2007.
- Tavares, P. L. – “Redes de Sensores Sem-Fio”. Disponível em: http://www.gta.ufrj.br/grad/02_2/Redes%20de%20sensores/Redes%20de%20Sensores%20Sem-fio.htm. Acessado em : outubro 2013.
- Trung K. N., Chung H. K, et all. – “CMOS Low-Noise Amplifier Design Optimization Techniques. IEEE Transactions on Microwave theory and Techniques”, Vol.52, NO. 5, May 2004.
- Young-Jin Kim, In-chul, et all. – “A Switchless Zigbee Frontend Transceiver With Matching Component Sharing of LNA and PA – IEEE Microwave and Wireless Components Letters”, Vol. 20, No. 9, september 2010.
- Zapata, C. “Ferramentas para modelagem de sistemas de comunicação.” Dissertação de mestrado. Escola Politécnica da Universidade de São Paulo.
- Zigbee – Alliance. (2013). Disponível em: <<http://zigbee.org/>>. Acessado em: 20/9/2013.

Zurita, M. - “Metodologia e Fluxo de Projeto de Sistemas VLSI Digitais”, 2013.
Material didático da Universidade Federal do Piauí, curso de Engenharia Elétrica.

ANEXO A

Circuito elétrico do LNA com ajuste fino nos cálculos realizados.



ANEXO B

- Verilog AMS Amplificador de Baixo Ruído (LNA)

//Verilog-AMS HDL for "Regina", "Ina" "verilogams"

`include "constants.vams"

`include "disciplines.vams"

module Ina (in_p, in_n, out_p, out_n, pd, gnd, vdd);

 //declaracao dos pinos

 input in_n, in_p, pd, gnd, vdd;

 output out_n, out_p;

 wreal gnd, vdd, in_n, in_p, out_n, out_p;

 logic pd;

 //declaracao dos parametros

 parameter real gain = 30;

 parameter real vinsat = 20; //tensao de saturação de entrada

 //declaracao das variaveis locais

 real vrfin, vrfout;

 real realgain; //conversao dB para real

 always @(in_p, in_n) begin

 realgain = pow(10,gain/20); //converte ganho para ganho real

 vrfin = vinsat* tanh(in_p - in_n)/vinsat; //coloca limites na

variação de entrada

 vrfout = vrfin* realgain;

 end

 assign out_p = (vrfout/2)* (~pd);

 assign out_n = -(vrfout/2)* (~pd);

endmodule

- Verilog AMS Mixer

```
//Verilog-AMS HDL for "Regina", "mixer" "verilogams"

`timescale 1ns/10fs
`include "constants.vams"
`include "disciplines.vams"

module mixer ( vlo_n, vlo_p, vrf_n, vrf_p, vout_p, vout_n, gnd, vdd, pd);
    //declaração dos pinos
    input vlo_n, vlo_p, vrf_n, vrf_p, gnd, vdd, pd;
    output vout_n, vout_p;

    //declaração das disciplinas
    wreal vlo_n, vlo_p;
    wreal vrf_n, vrf_p;
    wreal vout_n, vout_p, gnd, vdd;
    logic pd;

    //declaração dos parametros
    parameter real gain = 1;
    parameter real vcm_out = 1;      //tensão de modo comum na saída
    parameter real vsatrf = 10; //tensão saturação na entrada

    //declaração das variáveis locais
    real vinlo, vinrf, outmix;
    real realgain;

    always @(vrf_n, vrf_p, vlo_n, vlo_p, pd) begin
        realgain = pow(10, gain/20);
        vinlo = vlo_p - vlo_n;
        vinrf = vsatrf*tanh((vrf_p - vrf_n)/vsatrf);
        outmix = realgain*vinlo*vinrf;
    end
    assign vout_p = ((vcm_out + outmix)/2)*(~pd);
    assign vout_n = ((vcm_out - outmix)/2)*(~pd);

endmodule
```

- Verilog AMS amplificador programável (PGA)

```
//Verilog-AMS HDL for "Regina", "PGA" "verilogams"

`include "constants.vams"
`include "disciplines.vams"
`timescale 1ns/10ps

module PGA (in_p, in_n, c, out_p, out_n, pd, gnd, vdd );
```

```

//declaracao dos parametros
parameter real maxgain = 10;
parameter real vinsat = 0.1;      //tensao de saturação de entrada

//declaracao dos pinos
input in_n, in_p, pd, gnd, vdd;
input [0:3] c;
output out_n, out_p;
wreal gnd, vdd, in_n, in_p, out_n, out_p;
logic pd;
logic [0:3] c;

//declaracao das variaveis locais
real vrfin, vrfout;
real realgain, gain;              //conversao dB para real
real G = maxgain/16;

always @(in_p, in_n) begin

    case(c)
        4'b0000: begin gain = G*1; end
        4'b0001: begin gain = G*2; end
        4'b0010: begin gain = G*3; end
        4'b0011: begin gain = G*4; end
        4'b0100: begin gain = G*5; end
        4'b0101: begin gain = G*6; end
        4'b0110: begin gain = G*7; end
        4'b0111: begin gain = G*8; end
        4'b1000: begin gain = G*9; end
        4'b1001: begin gain = G*10; end
        4'b1010: begin gain = G*11; end
        4'b1011: begin gain = G*12; end
        4'b1100: begin gain = G*13; end
        4'b1101: begin gain = G*14; end
        4'b1110: begin gain = G*15; end
        4'b1111: begin gain = G*16; end
    endcase

    realgain = pow(10,gain/20);      //converte ganho para ganho real
    vrfin = vinsat* tanh(in_p - in_n)/vinsat; //coloca limites na variação
de entrada
    vrfout = vrfin* realgain;

    end

    assign out_p = (vrfout/2)* (~pd);
    assign out_n = -(vrfout/2)* (~pd);

endmodule

```


- Verilog AMS oscilador controlado por tensai (VCO)

```
//Verilog-AMS HDL for "Regina", "vco" "verilogams"

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

module vco ( in, out_n, out_p, pd, vdd, gnd);
    //declaração dos pinos
    input in, vdd, gnd;
    output out_n, out_p;
    wreal vdd, gnd;
    electrical out_n, out_p, in;
    logic pd;

    //declaração dos parametros
    parameter real vmin = 0;
    parameter real vmax = vmin +1.8 from (vmin:inf);
    parameter real fmin = 2.4G from (0:inf);
    parameter real fmax = 2.5G from (fmin:inf);
    parameter real ampl = 1;

    //declaração das variaveis locais
    real freq, phase;

    analog begin

        freq = (V(in)-vmin)*(fmax - fmin)/(vmax - vmin)+fmin; //calculo da
freq da tensao de entrada

        if (freq > fmax) freq = fmax;
        if (freq < fmin) freq = fmin;

        phase = 2*M_PI*idtmod(freq,0.0,1.0,-0.5);
        V(out_p) <+ (sin(phase)/2)*(~pd); //gerando saida p
        V(out_n) <+ (-sin(phase)/2)*(~pd); //gerando saida n
        $bound_step (0.1/freq); //ligado o passo de tempo

    end

endmodule
```

- Verilog AMS amplificador de potencia (PA)

```
//Verilog-AMS HDL for "Regina", "PA" "verilogams"

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

```

module PA (in_p, in_n, out_p, out_n, pd, gnd, vdd);

    //declaracao dos pinos
    input in_n, in_p, pd, gnd, vdd;
    output out_n, out_p;
    wreal gnd, vdd, in_n, in_p, out_n, out_p;
    logic pd;

    //declaracao dos parametros
    parameter real gain = 10;
    parameter real vinsat = 0.1;      //tensao de saturação de entrada

    //declaracao das variaveis locais
    real vrfin, vrfout;
    real realgain;                    //conversao dB para real

    always @(in_p, in_n) begin
        realgain = pow(10,gain/20);    //converte ganho para ganho real
        vrfin = vinsat* tanh(in_p - in_n)/vinsat; //coloca limites na
        //variação de entrada
        vrfout = vrfin* realgain;
    end

    assign out_p = (vrfout/2)* (~pd);
    assign out_n = -(vrfout/2)* (~pd);

endmodule

```

- Verilog AMS Filtro Passa baixa

```

//Verilog-AMS HDL for "Regina", "filter" "verilogams"

```

```

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

```

```

module filter(in_p, in_n, out_p, out_n, pd, gnd, vdd);

```

```

    //declaracao dos pinos
    input in_n, in_p, pd, gnd, vdd;
    output out_n, out_p;

    //declaracao das disciplinas
    electrical in_n, in_p, out_n, out_p;
    wreal gnd, vdd;
    logic pd;

    analog begin

```

```
        V(out_n) <+  
(laplace_nd(V(in_n),{8e018},{8e018,8e012,4e006,1.0}))*(~pd);  
        V(out_p) <+  
(laplace_nd(V(in_p),{8e018},{8e018,8e012,4e006,1.0}))*(~pd);  
    end  
endmodule
```