



Universidade de Brasília – UnB
Faculdade UnB Gama – FGA
Engenharia Eletrônica

Receptor de curtas distâncias para comunicação através da terra

Autor: Vitor Guedes da Silva
Orientador: Prof. Dr. Wellington Avelino

Brasília, DF
2020



Vitor Guedes da Silva

Receptor de curtas distâncias para comunicação através da terra

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.

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Prof. Dr. Wellington Avelino

Brasília, DF

2020

Vitor Guedes da Silva

Receptor de curtas distâncias para comunicação através da terra/ Vitor Guedes da Silva. – Brasília, DF, 2020-

122 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Wellington Avelino

Trabalho de Conclusão de Curso – Universidade de Brasília – UnB
Faculdade UnB Gama – FGA , 2020.

1. transceptor. 2. TTE. I. Prof. Dr. Wellington Avelino. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Receptor de curtas distâncias para comunicação através da terra

CDU 02:141:005.6

Vitor Guedes da Silva

Receptor de curtas distâncias para comunicação através da terra

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.

Trabalho aprovado. Brasília, DF, 16 de dezembro de 2020 – Data da aprovação do trabalho:

Prof. Dr. Wellington Avelino
Orientador

Prof. Dr. Leonardo Aguayo
Convidado 1

Prof. Dr. Adorian Judson
Convidado 2

Brasília, DF
2020

Resumo

Devido ao alto número de acidentes que vêm ocorrendo em ambientes de mineração, sistemas de comunicação Through the earth estão se tornando mais frequentes, possibilitando um melhor auxílio no planejamento preventivo contra acidentes. Esse tipo de sistema utiliza a indução eletromagnética em baixas frequências para se propagar por meio do solo, rochas, resíduos em barragens, entre outros meios. Neste âmbito, este trabalho apresenta inicialmente um embasamento teórico para o projeto de um receptor de curtas distâncias com aplicação em nós sensores, que podem ser utilizados no monitoramento de barragens ou resgate de vítimas. Em seguida é apresentado o projeto dos circuitos de amplificação de baixo ruído, do filtro passa faixa e do amplificador de ganho programável com aplicação em transceptores em sistemas TTE para curtas distâncias, além de uma explicação sobre o funcionamento dos mesmos. Por fim, são apresentadas simulações na plataforma Cadence e LTSpice do amplificador de baixo ruído, filtro passa faixa e PGA, além de suas respectivas caracterizações em bancada.

Palavras-chaves: Throgh the Earth. Amplificador de Baixo Ruído. Filtro Passa Faixa. Amplificador de Ganho Programável. Nó sensores. Transceptor.

Abstract

Due to the high number of accidents that have been occurring in mining environments, Through the earth communication systems are becoming more frequent, enabling better assistance in accident prevention planning. This type of system uses low frequency electromagnetic induction to propagate through soil, rocks, dam debris, and other channels. In this context, this work initially presents a theoretical basis for the design of a short distance receiver with sensor nodes application, which can be used to monitor dams or rescue victims. After that, the design of the low noise amplifier circuits, the bandpass filter and programmable gain amplifier applied to transceivers in short distance TTE systems, as well as a brief explanation of how they work. Finally, simulations are presented in the Cadence and LTSpice platforms of the low noise amplifier, bandpass filter and programmable gain amplifier, in addition their respective practical characterization.

Key-words: Through the Earth. Low Noise Amplifier. Programmable Gain Amplifier. Bandpass Filter. Sensor node.

Lista de ilustrações

Figura 1 – Função de transferência do canal quando $\sigma = 0.01\text{S/m}$ e $r = 200\text{ m}$. (1)	29
Figura 2 – Distribuição de energia de um ruído Gaussiano. (2)	31
Figura 3 – Ruído térmico. (2)	32
Figura 4 – Características dos ruídos do amp op TLV2772. (2)	34
Figura 5 – Modelo equivalente do amp op com ruído. (2)	34
Figura 6 – Modelo equivalente do amp op inversor com ruído. (2)	35
Figura 7 – Modelo equivalente do amp op não inversor com ruído. (2)	35
Figura 8 – Representação figurativa da ocorrência de aliasing em um sinal. (3)	37
Figura 9 – Resposta em frequência real de um filtro. (4)	38
Figura 10 – Classificação dos filtros.	38
Figura 11 – Localização dos polos em relação a variação do fator de qualidade Q.	39
Figura 12 – Resposta em frequência do filtro passa faixa de segunda ordem.	40
Figura 13 – PGA utilizando controle de ganho na malha externa (a) e seu detalhamento (b)(13).	41
Figura 14 – Antena tag 2.0.	44
Figura 15 – Antena tag escudo.	44
Figura 16 – Esquemático do LNA não inversor.	46
Figura 17 – Declaração dos pinos, parâmetros e variáveis	47
Figura 18 – Módulo do amplificador utilizado no LNA.	48
Figura 19 – Símbolo do amplificador modelado em Verilog-A.	48
Figura 20 – Esquemático do LNA inversor.	48
Figura 21 – Esquemático do LNA não inversor.	49
Figura 22 – Símbolo do LNA inversor.	49
Figura 23 – Símbolo do LNA não inversor.	50
Figura 24 – Layout do LNA.	50
Figura 25 – Protótipo do LNA.	51
Figura 26 – Testes em bancada do LNA.	51
Figura 27 – Topologia para o projeto de um FPF de quarta ordem. (5)	53
Figura 28 – Esquemático do FPF de quarta ordem.	54
Figura 29 – Declaração dos pinos, parâmetros e variáveis	55
Figura 30 – Módulo do amplificador utilizado no FPF.	55
Figura 31 – Esquemático do filtro passa alta.	56
Figura 32 – Símbolo do filtro passa alta.	56
Figura 33 – Pinos, parâmetros e variáveis do filtro passa alta.	57
Figura 34 – Módulo do filtro passa alta.	57
Figura 35 – Layout do FPF de quarta ordem.	58

Figura 36 – PCI do FPF de quarta ordem.	58
Figura 37 – Segundo layout do FPF de quarta ordem.	59
Figura 38 – Típica aplicação do CI LTC6911-1.	60
Figura 39 – Esquemático do PGA.	61
Figura 40 – Diagrama de funcionalidade do PGA para uma alimentação single supply de 3.3V.	62
Figura 41 – Função detetor de pico do PGA.	63
Figura 42 – Função detetor de nível do PGA.	64
Figura 43 – Declaração dos pinos, parâmetros e variáveis do PGA.	65
Figura 44 – Modelo do PGA.	66
Figura 45 – Símbolo do PGA.	67
Figura 46 – Layout da placa de circuito impresso do PGA.	67
Figura 47 – Protótipo do PGA.	68
Figura 48 – Testes em bancada do PGA.	68
Figura 49 – Montagem do primeiro procedimento do PGA.	69
Figura 50 – Montagem do segundo procedimento do PGA.	70
Figura 51 – Esquemático do receptor com LNA inversor e filtro passa baixa de 2º ordem.	71
Figura 52 – Esquemático do receptor com LNA não inversor e filtro passa baixa de 2º ordem.	72
Figura 53 – Esquemático do receptor com LNA não inversor e filtro passa baixa de 4º ordem.	72
Figura 54 – Esquemático do receptor com LNA não inversor, filtro passa baixa de 4º ordem e PGA.	73
Figura 55 – Testbench do LNA inversor.	76
Figura 56 – Testbench do LNA não inversor.	77
Figura 57 – Resposta em frequência da saída em dB20 do LNA inversor.	77
Figura 58 – Resposta em frequência da saída em dB20 do LNA não inversor.	78
Figura 59 – Testbench do filtro passa alta de 2º ordem.	79
Figura 60 – Resposta em frequência da saída em dB20 do filtro utilizando o esquemático.	79
Figura 61 – Resposta em frequência da saída em dB20 do filtro utilizando o modelo Verilog-AMS.	80
Figura 62 – Testbench do PGA.	81
Figura 63 – Resposta transiente da saída do PGA para uma entrada digital G=010 e amplitude de entrada de 1V pico a pico.	81
Figura 64 – Testbench dos circuitos de recepção utilizado LNA inversor e LPF de 2º ordem, LNA não inversor e LPF de 2º ordem, LNA não inversor e LPF de 4º ordem	82

Figura 65 – Testbench dos circuitos de recepção utilizado LNA não inversor, LPF de 4º ordem e PGA	83
Figura 66 – Resposta em frequência do ganho em dB20 dos circuitos de recepção utilizado LNA inversor e LPF de 2º ordem, LNA não inversor e LPF de 2º ordem, LNA não inversor e LPF de 4º ordem.	83
Figura 67 – Resposta em frequência da magnitude do ganho dos circuitos de recepção utilizado LNA inversor e LPF de 2º ordem, LNA não inversor e LPF de 2º ordem, LNA não inversor e LPF de 4º ordem.	84
Figura 68 – Resposta em frequência do ganho em dB20 do circuito de recepção utilizando LNA não inversor, LPF de 4º ordem e PGA.	84
Figura 69 – Modelo spice do do OP213.	85
Figura 70 – Testbench do LNA utilizando o modelo spice do OP213.	86
Figura 71 – Ganho em dB20 da resposta em frequência do LNA utilizando modelo spice do OP213.	86
Figura 72 – Densidade de ruído da saída do LNA com topologia não inversora.	87
Figura 73 – Testbench do filtro utilizando o modelo spice do LTC1568.	87
Figura 74 – Resposta em frequência da saída do filtro utilizando o modelo spice do LTC1568.	88
Figura 75 – Resposta transiente da saída do filtro utilizando o modelo spice do LTC1568.	88
Figura 76 – Testbench do PGA utilizando o modelo spice do LTC6911-1.	89
Figura 77 – Resposta transiente da saída do PGA utilizando o modelo spice do LTC6911-1 com alimentação single supply de 3.3V.	89
Figura 78 – Resposta transiente da saída do PGA utilizando o modelo spice do LTC6911-1 com alimentação simétrica de 3.3V.	90
Figura 79 – Sinal de entrada do LNA.	91
Figura 80 – Sinal de saída do LNA com ganho de 95.2V/V.	92
Figura 81 – Sinal de saída do PGA para G=111 e alimentação single supply.	94
Figura 82 – Sinal de saída do PGA para G=010.	95

Lista de tabelas

Tabela 1 – Cor dos ruídos(2).	33
Tabela 2 – Tabela de resultados da caracterização da antena tag 2.0 e da antena tag escudo.	43
Tabela 3 – Pinos amplificador LNA.	47
Tabela 4 – Pinos do amplificador utilizado no FPF.	54
Tabela 5 – Pinos do filtro passa alta de segunda ordem.	56
Tabela 6 – Ganhos e propriedades do CI LTC6911-1.	60
Tabela 7 – Pinos de entrada e saída do PGA.	65
Tabela 8 – Tabela de resultados do LNA.	92
Tabela 9 – Tabela de resultados do PGA.	94

Lista de abreviaturas e siglas

Amp op	Amplificador operacional
ENB	Banda equivalente
FPF	Filtro passa faixas
GPIO	Entradas e saídas para propósito geral
LNA	Amplificador de baixo ruído
LPF	Filtro passa baixa
PCI	Placa de circuito impresso
PDF	Função densidade de probabilidade
PGA	Amplificador de ganho programável
TTE	Através da terra
ULF	Frequências ultra baixas
Verilog-HDL	Verilog hardware description Language
VGA	Amplificador de ganho variável
VLF	Frequências muito baixas

Lista de símbolos

∇	Operador Gradiente
μ	Permeabilidade do meio
ε	Permissividade do meio
σ	Condutividade do meio
δ	Profundidade peculiar
ω	Frequência angular
Ω	Resistência elétrica

Sumário

1	INTRODUÇÃO	23
1.1	Contextualização	23
1.2	Objetivos	24
1.3	Metodologia	24
2	CONTEXTUALIZAÇÃO TEÓRICA	27
2.1	TTE	27
2.1.1	Ambiente de propagação TTE	27
2.1.2	Modelo de propagação do campo magnético	28
2.1.3	Ruído em sistemas de comunicação TTE	29
2.2	Ruído em amplificadores operacionais(Amp op)	30
2.2.1	<i>Shot noise</i>	31
2.2.2	Ruído Térmico	32
2.2.3	<i>Flicker noise</i>	32
2.2.4	Cor do ruído	33
2.3	Ruído da frequência de <i>Corner</i>	33
2.4	Modelo do ruído em Amp ops	34
2.4.1	Ruído no Amp op inversor	34
2.4.2	Ruído no Amp op não inversor	35
2.5	Fórmula de Friis	36
2.6	LNA	36
2.6.1	Figura de ruído	36
2.6.2	Ganho	36
2.6.3	Estabilidade	36
2.6.4	Largura de banda	36
2.6.5	Potência de dissipação	37
2.7	Aliasing	37
2.8	Filtros	37
2.8.1	filtros de segunda ordem	39
2.8.2	Filtro passa faixa	39
2.9	PGA	40
2.10	Verilog A e Verilog AMS	41
3	PROJETO	43
3.1	Antena	43
3.1.1	Caracterização da antena <i>loop</i>	43

3.2	LNA	45
3.2.1	Projeto	45
3.2.2	Modelagem	46
3.2.3	Protótipo	50
3.3	Filtro passa faixa	52
3.3.1	Projeto	52
3.3.2	Modelagem	54
3.3.3	Protótipo	57
3.4	PGA	59
3.4.1	Projeto	60
3.4.2	Lógica de programação	61
3.4.3	Modelagem	65
3.4.4	Protótipo	67
3.5	Receptor TTE	70
3.5.1	Modelagem	70
4	SIMULAÇÕES E ANÁLISE DOS RESULTADOS	75
4.1	Simulações da modelagem utilizando Verilog-A	75
4.1.1	LNA	75
4.1.2	Filtro passa baixa	78
4.1.3	PGA	80
4.1.4	Receptor TTE	82
4.2	Simulações dos blocos utilizando modelos <i>Spice</i>	84
4.2.1	LNA	85
4.2.2	Filtro passa faixa	87
4.2.3	PGA	88
4.3	Caracterização em bancada dos blocos	90
4.3.1	LNA	90
4.3.2	Filtro passa faixa	93
4.3.3	PGA	94
5	DISCUSSÕES	97
5.1	Discussões	97
	REFERÊNCIAS	99

APÊNDICES	101
APÊNDICE A – MODELAGEM EM VERILOG-AMS DO AMP OP UTILIZADO NO CIRCUITO LNA	103
B – MODELAGEM EM VERILOG-AMS DO AMP OP UTILIZADO NO FILTRO PASSA ALTA	105
C – MODELAGEM EM VERILOG-AMS DO FILTRO PASSA ALTA . .	107
D – CÓDIGO DO PGA	109
E – MODELAGEM EM VERILOG-AMS DO PGA	121

1 Introdução

1.1 Contextualização

A indústria mineradora segue em constante desenvolvimento, como consequência disto se tem um aumento expressivo dos rejeitos gerados a serem armazenados em barragens, de maneira que ficam evidentes novos desafios, como a necessidade de uma estrutura maior para as barragens e a implementação de um sistema de manutenção para barragens em processo de envelhecimento. (6)

Mediante a estes desafios, é de suma importância estabelecer um bom planejamento, como de sistemas de monitoramento planejados, nos quais são compostos por circuitos de comunicação e instrumentação eficazes. Além disso, barragens de contenção de rejeitos são estruturas de grande porte e necessitam de um monitoramento incessante de seu funcionamento durante sua vida útil. (6)

Não obstante, a necessidade de comunicação em monitoramento de barragens e em atividades mineradoras, sistemas automatizados estão se tornando mais recorrentes, com isso é necessário fornecer meios que possibilitem a comunicação entre máquinas e equipamentos em diferentes níveis das minas ou barragens (1).

Uma maneira eficaz de fazer o monitoramento de barragens é através da utilização de nós sensores, nos quais são pequenos equipamentos compostos por um instrumento de medição e um transceptor que seja capaz de realizar comunicações através dos rejeitos presentes na barragem. Contudo, para se realizar a comunicação entre dois transceptores através dos rejeitos é necessário a implementação de um mecanismo de comunicação TTE(through-the-earth).

Estes nós sensores são compostos por uma sequência importante de circuitos: antena, filtro passa faixa(FPF), LNA e PGA.

A antena tem como princípio receber ou transmitir uma mensagem por meio da indução magnética. O FPF irá filtrar interferências de linha AC, filtrar o *aliasing* e delimitar a banda de frequência desejada. O LNA tem como objetivo aumentar o ganho sinal-ruído, além de melhorar a sintonização da antena. Por fim, o PGA irá fazer o controle de ganho do circuito, permitindo que o ganho seja alterado de acordo com a proximidade entre os circuitos de transmissão e recepção.

1.2 Objetivos

Objetivo geral: Desenvolver um transceptor para curtas distâncias, com o objetivo de utilizar em nós sensores para monitoramento de barragens, de forma a evitar possíveis rompimentos de barragens. Além disso, há a possibilidade desses transceptores serem utilizados em situações de desastres, auxiliando na busca de vítimas.

Objetivos específicos:

- Estudar sobre a comunicação TTE com foco na sua influência sobre o canal de comunicação e suas interferências nos dispositivos de comunicação;
- Caracterizar antenas tags e escudo a serem utilizadas em transceptores em curtas distâncias;
- Estudos relacionados ao filtro passa faixas de quarta ordem juntamente com o projeto e posteriormente sua prototipação;
- Estudos relacionados ao LNA, projeto do LNA e sua prototipação;
- Estudos relacionados ao PGA, projeto do bloco individual, desenvolvimento da lógica de comando do PGA utilizando a Raspberry pi 3B+ e por fim sua prototipação;
- Caracterização dos circuitos projetados;
- Modelagem em verilog-A dos seguintes blocos: LNA, FPF e PGA.

1.3 Metodologia

- Estudos teóricos e pesquisas sobre assuntos relacionados a comunicação TTE;
- Estudos relacionados aos circuitos que implementam a parte de recepção do transceptor;
- Caracterização das antenas loop e escudo;
- Utilização da ferramenta Cadence para simulação do circuito LNA e FPF a ser implementado;
- Utilização da ferramenta Altium para a confecção do layout das placas de circuito impresso(PCI) dos circuitos LNA e FPF

Este documento será organizado da seguinte forma:

- Introdução composta por aspectos gerais, objetivos e metodologias utilizadas.

-
- Contextualização teórica sobre comunicação TTE, LNA, FPF e PGA.
 - Projeto, composto por materiais e métodos utilizados para a caracterização da circuitaria que compõe o receptor.
 - Resultados obtidos em simulações mistas desenvolvidas no software Cadence.
 - Resultados obtidos pela caracterização dos blocos prototipados do LNA, FPF e PGA.

2 Contextualização Teórica

2.1 TTE

O sistema TTE se opõe a sistemas tradicionais que operam com antenas de RF em longas distâncias e com meios que possuem características semelhantes às do vácuo. Todavia, em ambientes TTE, existem adversidades como meio composto por pedras, água, areia e entre outros materiais com condutividade elétrica não desprezível. Com base nessas condições, o sistema TTE tem uma grande perda na propagação proporcionalmente ao aumento da frequência e da condutividade elétrica do meio (1).

Partindo deste contexto, surgiu-se o conceito de comunicações sem fio através da terra, no qual utiliza-se ondas eletromagnéticas, por meio da indução magnética através do solo, em frequências ultra baixas (ULF), que estão em torno de 300 Hz e 3 kHz, ou frequências muito baixas (VLF), que estão em torno de 3 kHz e 30 kHz. Neste trabalho utilizaremos frequências VLF (1).

A resposta em frequência do canal é diretamente dependente das individualidades do solo juntamente com sua profundidade, isso se deve a forma em que o sinal se propaga e dos circuitos magnéticos presentes (1). Além disso, existem outras dificuldades intrínsecas a este tipo de comunicação como: limitação da banda causada pelo canal, proporcionando uma diminuição da taxa de transmissão de dados, ruídos gaussianos térmicos e alguns componentes de ruídos não gaussianos, podendo ser provocados por fenômenos atmosféricos, como de relâmpagos ou pela interferência de linhas de transmissão e dispositivos elétricos(1).

2.1.1 Ambiente de propagação TTE

De maneira geral, a penetração de uma onda eletromagnética em um meio condutor é descrita pela equação $\nabla^2 H = \mu\varepsilon(\partial^2 H)/(\partial t^2)$ porém, devido as características intrínsecas do solo, este meio obedece a seguinte equação de difusão $\nabla^2 H = \mu\sigma(\partial H)/(\partial t)$, em que H é o campo magnético dentro do meio condutor, σ condutividade, ε a permissividade e μ a permeabilidade (7).

A justificativa para se utilizar frequências ULF e VLF em comunicações TTE é dado pelo grau de decaimento no meio condutor, no qual é dado pela profundidade particular $\delta = \sqrt{2/\omega\mu\sigma}$, que é inversamente proporcional a raiz quadrada da frequência, de forma que $\sigma/(\varepsilon\omega) \gg 1$. Além disso, a onda eletromagnética que se propaga em um meio condutor ou dielétrico irá sofrer alterações em sua velocidade de propagação e seu comprimento de onda (7).

Para este projeto, a antena loop é utilizada como um transdutor magnético por conta da utilização da comunicação de campo reativo. A capacidade de transmissão indutiva da antena loop pode ser dado pelo momento magnético $md = N_{tx}I_{tx}S_{tx}$, de maneira que N_{tx} é o número de espirais, I_{tx} é o valor RMS da corrente e S_{tx} é a área de transmissão da antena (7).

Para campos magnéticos próximos as antenas são consideradas eletricamente curtas, e com uma corrente uniformemente distribuída por todos os espirais da antena. Considerando para este trabalho um campo quase estático, no qual se pode aproximar a transmissão no vácuo que é em sua média homogênea e uniforme. Portanto o cálculo para um campo quase estático e para curta distâncias pode ser expresso por (7):

$$H_{qs} = \frac{m_d}{4\pi r^3} 2 \cos(\theta) \hat{r} + \sin(\theta) \hat{\theta}, \quad (2.1)$$

em que \hat{r} e $\hat{\theta}$ são vetores na coordenadas esféricas nas direções radial e de elevação. m_d é o momento magnético, r a distância entre o local de transmissão e o local de medição. Para comunicação TTE a antena fica na vertical, de modo que o θ é igual a zero, resultando no seguinte valor para o campo magnético(7):

$$H_{qs} = \frac{m_d}{2\pi r^3}, \quad (2.2)$$

2.1.2 Modelo de propagação do campo magnético

Supondo que o solo seja um meio condutor de extensões infinitas, o campo magnético produzido por uma antena loop pode ser aproximado em coordenadas esféricas por(1):

$$H = \frac{m_d}{4\pi r^3} e^{-jT} e^{-T} \{2 \cos \theta (1 + (1 + j)T) \hat{r} + \sin \theta (1 + (1 + j)T + 2jT^2) \hat{\theta}\}, \quad (2.3)$$

em que $T = r\sqrt{\mu\sigma\omega/2}$, μ é a permeabilidade magnética do solo, σ é a condutividade elétrica do solo, ω é a frequência angular, r é a distância entre o transmissor e o ponto de indução, m_d é o momento magnético da antena loop e θ é o ângulo entre o ponto que produz a indução e o ponto que sofre a indução(1).

O campo magnético gerado pela antena transmissora induz uma tensão na antena receptora, de modo que ambas antenas estejam na vertical e alinhadas, gerando um ângulo θ igual a 0. Dessa forma, a tensão induzida na antena receptora pode ser expressa por(1):

$$V_{rx}(\omega) = -j\omega\mu N_{rx}S_{rx}H_r \quad (2.4)$$

em que H_r é a componente do campo magnético na direção \hat{r} , N_{rx} é número de espirais e S_{rx} área da antena loop receptora(1).

A impedância de transferência é definida pela razão entre a tensão induzida na antena receptora, V_{rx} , e a corrente elétrica na antena transmissora, I_{tx} (1):

$$Z(T) = V_{rx}(T)/I_{tx}(T) = N_{tx}N_{rx}S_{tx}S_{rx}F_r, \quad (2.5)$$

em que F_r é função de transferência expressa por:

$$F_r = \frac{T^2}{\pi\sigma r^5} e^{-T} \sqrt{1 + 2T + 2T^2} \exp \left\{ j \left[\arctan \frac{T}{T+1} - T - \frac{\pi}{2} \right] \right\} \quad (2.6)$$

Portanto, a impedância de transferência é a combinação entre os ganhos intrínsecos das antenas de recepção e transmissão e a função de transferência. Rejeitando os ganhos e analisando as distorções causadas pelo canal, restando apenas F_r . Exemplificando, defini-se $r = 200m$ e $\sigma = 0.01S/m$, obtém-se a função de transferência do canal mostrada na figura 20(1).

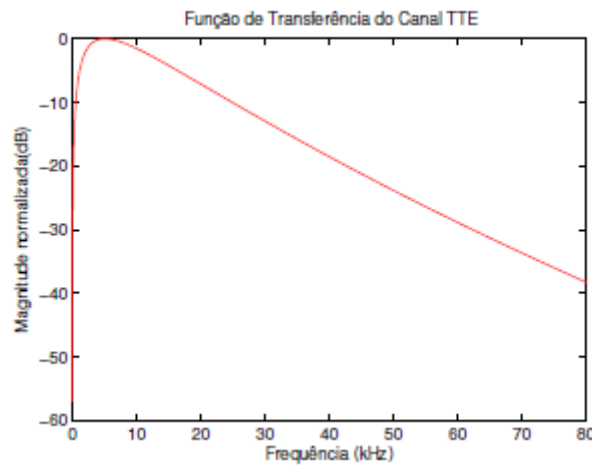


Figura 1 – Função de transferência do canal quando $\sigma = 0.01S/m$ e $r = 200 m$. (1)

2.1.3 Ruído em sistemas de comunicação TTE

Como já apontado, umas das dificuldades da comunicação TTE é a presença de ruídos atmosféricos, que de acordo com sua impulsividade pode deteriorar ainda mais a comunicação, e ruídos antrópicos, que são causados por meio da presença de harmônicos criados principalmente pela presença de equipamentos existentes em mineradoras e por conta das linhas de transmissão de potência (8).

Os ruídos atmosféricos estão ligados com forças da natureza, que comumente se propagam através da superfície terrestre, sendo consequência de descargas elétricas. A comunicação TTE é altamente afetada por este tipo de ruído, pois possui como meio de comunicação a terra, de forma que para se obter um menor ruído é interessante a utilização do enlace de subida no momento em que for feito a comunicação, pois há uma grande atenuação de ruídos através da terra (8).

Em teoria, o ruído atmosférico afeta diretamente no receptor, sendo necessário avaliar em função de temperatura de ruído.

Os Ruídos antrópicos são ruídos produzidos pelo homem, sendo em sua maior parte componentes harmônicas produzidas por energia elétrica. Em contraponto ao ruído atmosférico que tem como predominância frequências entre $10kHz$ e $30kHz$, os ruídos antrópicos estão em frequências próximas a $1kHz$, podendo ser observado através de quilômetros de distância da linha de transmissão, de acordo com sua tensão, podendo estar entre dezenas a centenas de kV (8).

O ruído externo pode ser considerado como o somatório entre o ruído atmosférico e o harmônico, da mesma maneira que o ruído interno é a junção entre o ruído do amplificador e ADC na placa de aquisição de dados. Levando em consideração estes ruídos, é possível medir e avaliar a sua contribuição à temperatura do ruído atmosférico (9).

Gibson em (10) apresenta um modelo entre a razão do ruído atmosférico e térmico, $F_a = 10\log(T_a/T_0)$, dado por:

$$F_a = PSD_a - 20\log(N_R S_R) - 40\log(f) + 492, \quad (2.7)$$

em que a PSD_a é densidade de potência do ruído atmosférico em dBV/\sqrt{Hz} , estimado pelas medidas de tensão.

2.2 Ruído em amplificadores operacionais(Amp op)

De maneira simplificada, ruído é simplesmente um sinal aleatório, no qual sua amplitude e fase instantânea não podem ser pré determinadas em nenhum momento. O ruído em um amp op pode ser gerado internamente, estando associado diretamente com componentes passivos ou alimentações externas(2).

A amplitude instantânea de um ruído pode ser negativa ou positiva, de forma que ao serem plotadas formam um padrão aleatório centrado em zero, de forma que estas amplitudes podem ser estimadas através de um função densidade de probabilidade(PDF). A PDF mais comum é a Gaussiana, no qual é caracterizada por seus momentos, tendo como um dos seus principais momento sua média, de forma que a FDP é simétrica em torno do seu centro, como mostrado na figura 2(2).

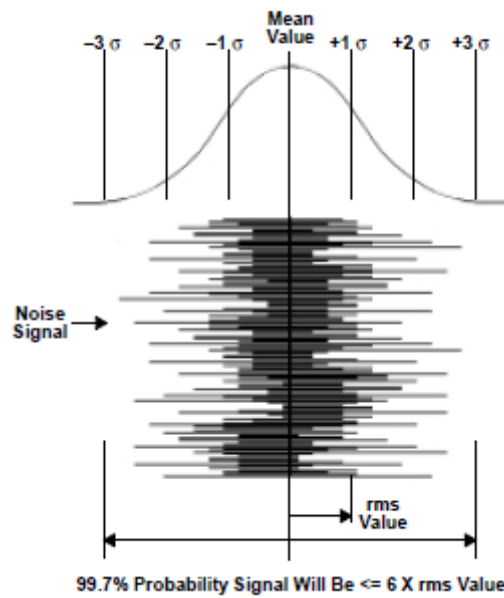


Figura 2 – Distribuição de energia de um ruído Gaussiano. (2)

σ é o desvio padrão da PDF Gaussiana e o valor rms da tensão e corrente do ruído, de forma que a tensão instantânea está entre $\pm\sigma$ 68% do tempo e está 97.7% do tempo entre $\pm 3\sigma$. A variância é dada por σ^2 , ou seja, é a variação da raiz quadrada da média em relação ao valor da média(2).

Quando se tem a presença de múltiplos ruídos, o valor total do sinal de ruído é dado pela seguinte equação (2):

$$E_{Totalrms} = \sqrt{e_{1rms}^2 + e_{2rms}^2 + \dots + e_{nrms}^2} \quad (2.8)$$

Para esta relação o valor resultante do ruído é dominado pelo ruído com maior amplitude, por exemplo, considerando um ruído com $10V_{rms}$ e um com $1V_{rms}$, o valor total do ruído fica (2):

$$E_{Totalrms} = \sqrt{10^2 + 1^2} = 10.05V_{rms} \quad (2.9)$$

Em circuitos existem cinco tipos de ruídos mais comuns: *Shot noise*, Ruído térmico, *Flicker noise*, *Burst noise* e Ruído de avalanche.

2.2.1 Shot noise

Este ruído é causado por flutuações randômicas no movimento das cargas na portadora do condutor. Os elétrons presentes em correntes se chocam e geram um potencial de energia criado pela quebra de barreira entre eles, de forma que cada elétron tem uma pequena contribuição neste potencial(2).

Este ruído tem algumas características importantes:

- Ele está sempre relacionado com a corrente de fluxo;
- Ele não depende da temperatura;
- Sua densidade espectral de potência é uniforme;
- *Shot noise* está sempre presente em materiais condutores.

2.2.2 Ruído Térmico

O ruído térmico é causado pela agitação dos elétrons no condutor. As perturbações de calor causadas em elétrons são respostas da aplicação de algum potencial, em consequência é adicionado uma componente aleatória ao seu movimento(figura 3)(2).

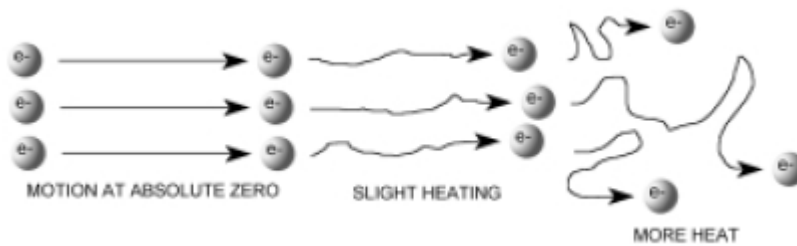


Figura 3 – Ruído térmico. (2)

É importante ressaltar que ao se utilizar resistores com valores grandes na entrada do amp op, o seu ruído térmico será amplificado pelo ganho do circuito.

2.2.3 *Flicker noise*

Este tipo de ruído é presente em diversos componentes passivos, estando relacionado pela imperfeição de estruturas cristalinas em semicondutores(2).

Algumas características do *flicker noise*:

- É inversamente proporcional a frequência;
- Está associado a correntes DC em equipamentos elétricos;
- Tem a mesma potência em cada década.

Ao se aumentar o consumo de potência em amp ops é possível reduzir este tipo de ruído, em contraponto há o aumento do ruído térmico.

2.2.4 Cor do ruído

Para distinguir melhor os ruídos eles são divididos em cores, de maneira que estas cores estão relacionadas diretamente com sua frequência, como mostrado na tabela 2.2.4(2).

Cor	Frequência
Roxo	f^2
Azul	f
Branco	1
Rosa	$\frac{1}{f}$
Vermelho/Marrom	$\frac{1}{f^2}$

Tabela 1 – Cor dos ruídos(2).

Neste documento serão abordados rapidamente sobre ruídos brancos e rosas.

O ruído branco possui espectro de frequência e potência constante independentemente da frequência e teoricamente possui energia infinita para frequências infinitas (2).

Já o ruído rosa possui frequência $1/f$ para o espectro de frequência e potência com exceção do espectro DC, ou seja, a amplitude desse ruído decai exponencialmente de acordo com a frequência (2).

2.3 Ruído da frequência de Corner

Um gráfico de corrente normalmente possui duas regiões distintas: para baixas frequências, nos quais tem-se o ruído rosa como predominante, e para altas frequências, em que o ruído branco tem um efeito dominante (2).

Ao se separar o ruído branco do rosa é possível determinar a frequência de Corner(f_c) visualmente através do gráfico, sendo a intersecção entre o ruído branco e o ruído rosa($1/f$), como apresentado na figura 4(2).

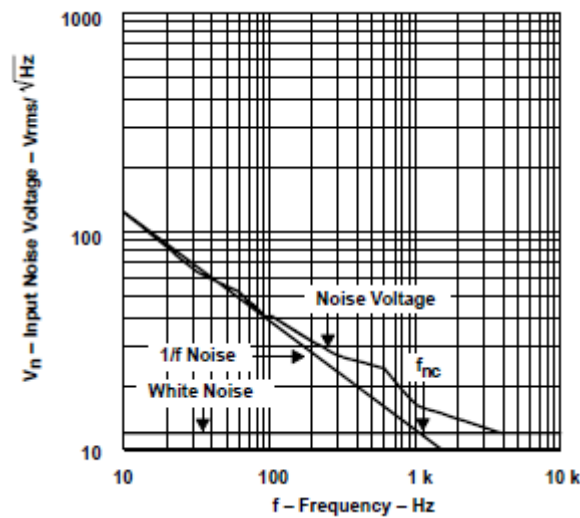


Figura 4 – Características dos ruídos do amp op TLV2772. (2)

Sabendo o valor da frequência de Corner é possível calcular o componente de ruído presente no amplificador.

2.4 Modelo do ruído em Amp ops

Na prática, circuito amplificadores são modelados para possuírem uma baixa impedância na fonte de entrada, de forma que para essas baixas impedâncias apenas a tensão do ruído é importante (2).

O modelo do circuito amplificador pode ser representado pela figura 5.

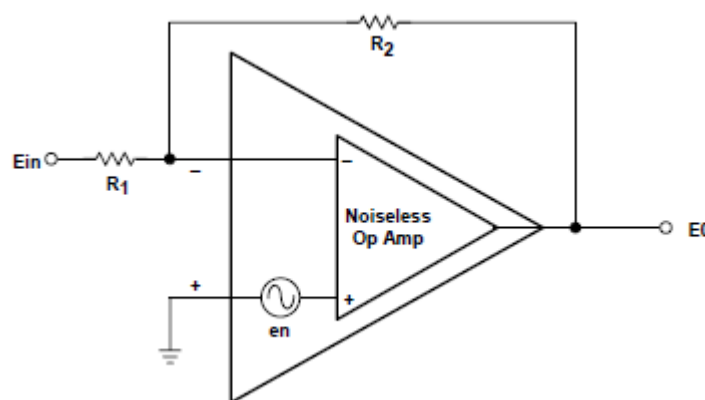


Figura 5 – Modelo equivalente do amp op com ruído. (2)

2.4.1 Ruído no Amp op inversor

Tomando como base o circuito amplificador apresentado na figura 5, é possível modelar o circuito do amplificador inversor (figura 6) (2).

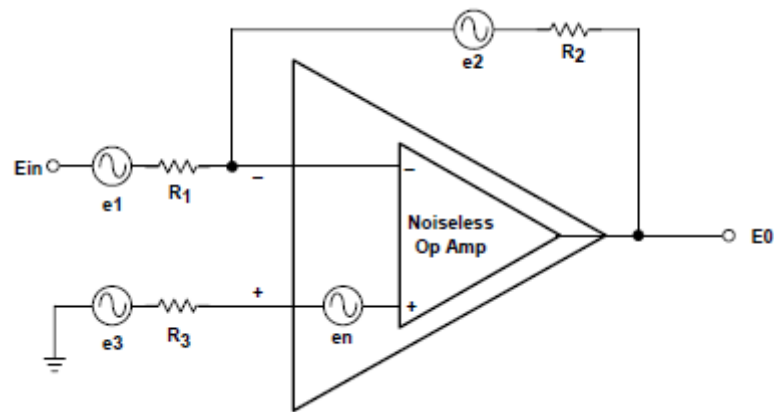


Figura 6 – Modelo equivalente do amp op inversor com ruído. (2)

Seu ganho pode ser calculado pela seguinte equação:

$$E_0 = \sqrt{\left(E_{in} \frac{R_2}{R_1}\right)^2 + \left(e_n \left(1 + \frac{R_2}{R_1}\right)\right)^2}, \quad (2.10)$$

em que e_n é o ruído total presente na largura de banda de interesse.

2.4.2 Ruído no Amp op não inversor

De forma semelhante ao amp op inversor, o amp op não inversor pode ser modelado tomando como base o amp op apresentado na figura 7 (2).

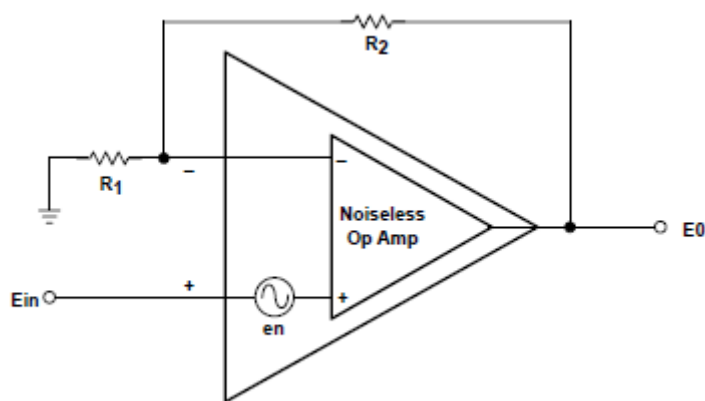


Figura 7 – Modelo equivalente do amp op não inversor com ruído. (2)

Seu ganho pode ser calculado pela seguinte equação:

$$E_0 = \sqrt{\left(E_{in} \left(1 + \frac{R_2}{R_1}\right)\right)^2 + \left(e_n \left(1 + \frac{R_2}{R_1}\right)\right)^2} \quad (2.11)$$

2.5 Fórmula de Friis

Pra um circuito em que há o cascadeamento de estágios de amplificação, os estágios anteriores contribuem para redução do ruído dos estágios subsequentes (11), sendo possível observar através da fórmula de Friis, que é expressa por:

$$NF_{tot} = 1 + (NF_1 - 1) + \frac{NF_2 - 1}{A_{P1}} + \dots + \frac{NF_m - 1}{A_{P1} \dots A_{P(m-1)}}, \quad (2.12)$$

no qual NF é a figura de ruído presente em cada estágio do cascadeamento de estágios e A_{Pn} representa o ganho de potência do estágio (11).

2.6 LNA

Em circuitos receptores o LNA tem um papel crucial em seu desempenho, de forma que o LNA possui alguns parâmetros de desenvolvimento importantes e que da formam que são estabelecidos geram um melhor desempenho (11).

2.6.1 Figura de ruído

A figura de ruído do LNA é adicionada diretamente no circuito receptor, de forma que essa figura de ruído é minimizada com o aumento dos estágios de amplificação, sendo possível observar através da equação 2.12 (11).

2.6.2 Ganho

O ganho do LNA deve ser grande o suficiente a fim de minimizar o máximo possível a contribuição do ruído nos estágios subsequentes (11).

2.6.3 Estabilidade

Da mesma maneira que os outros circuito presentes em um receptor, o LNA também pode sofrer interferências externas, por conseguinte, é interessante que o LNA seja estável na banda de frequência que está sendo utilizada. Porém, é importante tomar cuidado com a destabilização do LNA até mesmo em frequências em que não estão se trabalhando, pois ele pode gerar um alta não linearidade e por consequência o ganho irá sofrer uma alta compressão (11).

2.6.4 Largura de banda

O LNA deve fornecer uma resposta em frequência relativamente plana para a banda de interesse, de preferência com menos de 1 dB de variação de ganho (11).

2.6.5 Potência de dissipação

O LNA tradicionalmente possui uma troca direta entre ruído, linearidade e potência de dissipação. Não obstante, na maior parte dos circuitos receptores o LNA consome apenas uma pequena parte da potência, tendo-se como principal fator crítico o ruído (11).

2.7 Aliasing

Ao se processar sinais discretos no tempo, é necessário amostrar o sinal de interesse, de forma que existem componentes que podem interferir no processo de amostragem, sendo necessário fazer o devida filtragem para se remover frequências indesejadas dentro da banda de interesse. O processo pelo qual aparecem componentes indesejados dentro da banda de amostragem é conhecido como *aliasing*(figura 8), sendo necessário uma filtragem destas frequências, limitando a banda indesejada antes de se realizar o processo de amostragem (3).

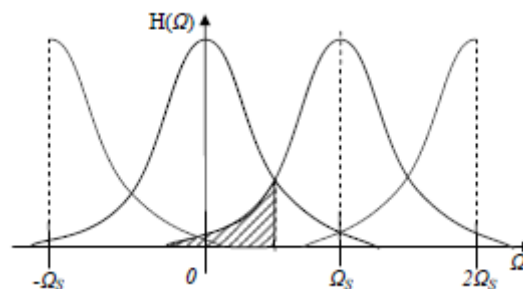


Figura 8 – Representação figurativa da ocorrência de aliasing em um sinal. (3)

2.8 Filtros

Para que um filtro seja eficaz é necessário que ele possua algumas características importantes (4):

- Resposta em frequência plana na largura de banda desejada;
- Atenuação do sinal interferente , ou seja, a medida em que ele for saindo da região em que se encontra a banda passante ocorre um decaimento até que se torne muito baixo(figura 9);

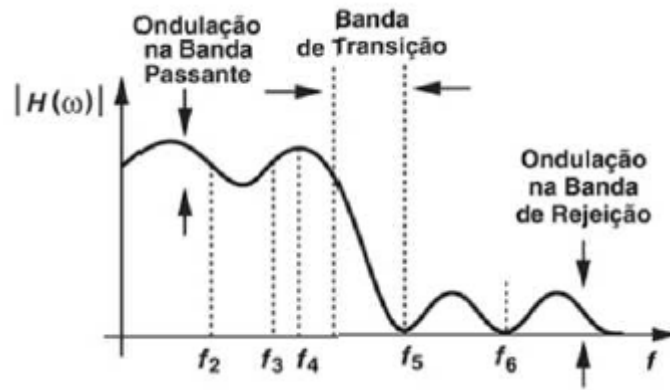


Figura 9 – Resposta em frequência real de um filtro. (4)

- Ondulações(ripple) das amplitudes na resposta em frequências(figura 9), sendo interessante um menor ripple possível, não atenuando amplitudes em faixas de frequências desejadas;
- Banda de transição estreita, determinando a parcela que a interferência continua afetando o sinal;
- Atenuação suficiente da banda de rejeição, de forma que ele fique com amplitudes muito baixas.
- Linearidade, ruído, dissipação de potência e complexidade(de acordo com a ordem o filtro).

Os filtro podem ser classificados em: passa baixas, passa altas, passa faixa e rejeita faixa(figura 10), de forma que daremos ênfase ao filtro passa faixa (4).

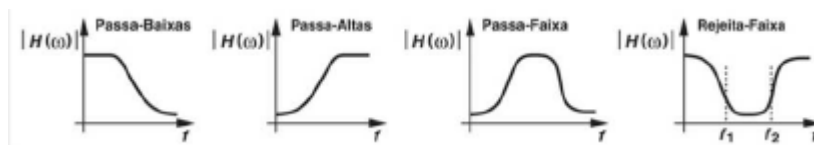


Figura 10 – Classificação dos filtros.

Algumas características dos filtro são influenciadas diretamente pela função de transferência, na qual determina como a resposta em frequência irá se comportar. De maneira genérica, a resposta em frequência é (4):

$$H(s) = \alpha \frac{(s - z_1)(s - z_2)\dots(s - z_m)}{(s - p_1)(s - p_2)\dots(s - p_n)}, \quad (2.13)$$

no qual z_k e p_k (reais ou complexos) são os zeros e polos do filtro, respectivamente, sendo interessante sempre manter o fatores m com uma quantidade inferior aos n, caso contrário $H(s) \rightarrow \infty$, fazendo com que se torne algo irreal (4).

2.8.1 filtros de segunda ordem

Um filtro de segunda ordem possui a seguinte função de transferência (4):

$$H(s) = \frac{\alpha s^2 + \beta s + \gamma}{s^2 + \frac{\omega_n}{Q}s + \omega_n^2}, \quad (2.14)$$

em que diferentemente do denominador o numerador é expresso pela frequência natural do circuito ω e pelo fator de qualidade Q . Ao se analisar os polos tem-se (4):

$$p_{1,2} = -\frac{\omega_n}{2Q} \pm j\omega_n \sqrt{1 - \frac{1}{4Q^2}}, \quad (2.15)$$

ao passo que o fator de qualidade aumenta (figura 11), a parte real tende a zero e a parte imaginária tende a ω_n tornando os polos "mais imaginários", resultando em um circuito mais instável (4).

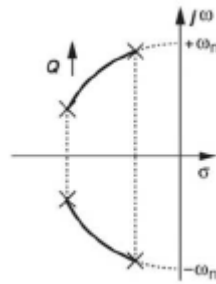


Figura 11 – Localização dos polos em relação a variação do fator de qualidade Q .

Vale ressaltar, que para se obter um filtro de quarta ordem basta fazer o cascadeamento de dois filtros de segunda ordem e de forma similar, para obter um filtro de oitava ordem basta cascadear dois de quarta ordem.

2.8.2 Filtro passa faixa

Para filtros passa faixa a função os valores de α e γ são iguais a zero, de forma que sua função de transferência fica (4):

$$H(s) = \frac{\beta s}{s^2 + \frac{\omega_n}{Q}s + \omega_n^2} \quad (2.16)$$

Quando $s \rightarrow 0$ e $s \rightarrow \infty$ a magnitude da função transferência tende a ser zero ou próximo de zero. De forma que entre os extremos o valor da magnitude atinge um pico na frequência natural (figura 12) (4).

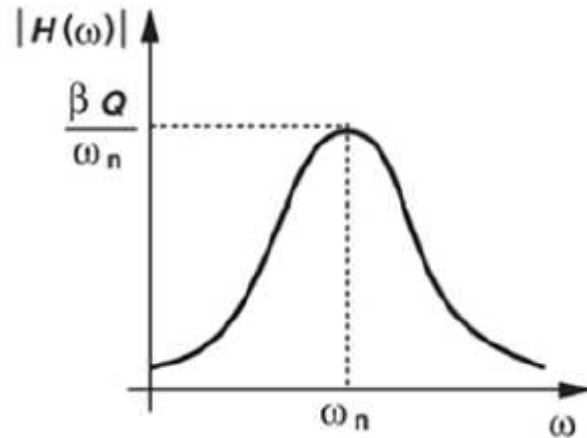


Figura 12 – Resposta em frequência do filtro passa faixa de segunda ordem.

2.9 PGA

Em amplificadores utilizados em circuitos é possível alterar seu ganho através da mudança discreta de seus valores, podendo assim, caracterizar a variação senoidal do ganho (12).

Quando o ganho é controlado por meio analógico, seja por meio de uma tensão ou corrente de entrada, é possível caracterizar um amplificador de ganho variável(VGA), que pode atingir infinitos valores por possuir um sinal de controle analógico, estando limitado dentro da faixa que foi projetado (12).

Similarmente é possível determinar um amplificador de ganho programável(PGA), em que o ganho agora possui valores discretos dentro de limites estabelecidos no projeto do circuito. Em muitas aplicações com VGA é necessário utilizar um conversor analógico-digital(A/D) para promover uma variação discreta do ganho. Tendo isto em mente, é possível definir o PGA como um VGA que possui níveis discretos de ganho (12).

Para o controle de ganho digitalmente existem algumas técnicas, de modo que neste trabalho será utilizado a realimentação do amplificador, no qual o controle será realizado pela mudança do valor da resistência da malha externa, como mostrado na figura 13. A vantagem do uso desta topologia é uma melhor linearidade na tensão de saída (13).

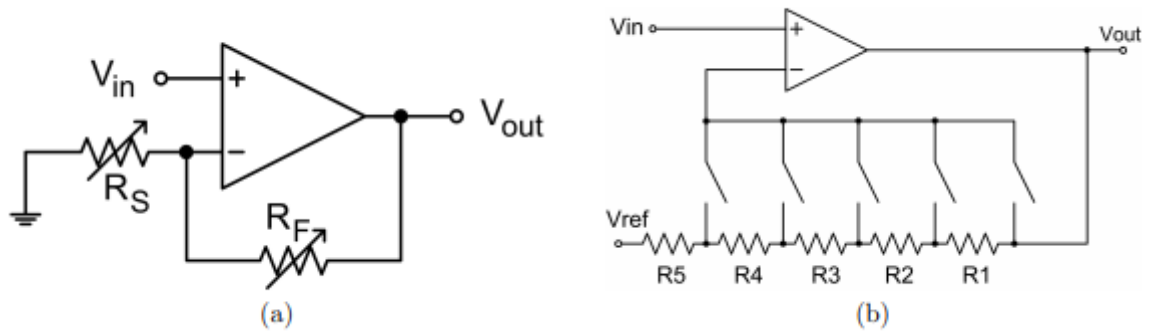


Figura 13 – PGA utilizando controle de ganho na malha externa (a) e seu detalhamento (b)(13).

2.10 Verilog A e Verilog AMS

O *Verilog Hardware Description Language* (Verilog-HDL) tem sido a linguagem mais utilizada para descrever complexos hardwares digitais, tendo como propósito suportar simulações de circuitos digitais (14).

Com o intuito de se desenvolver uma linguagem capaz de suportar simulações de sistemas analógicos e simulações mistas, um grupo chamado *Analog Verilog International* desenvolveu o Verilog-A, que é uma linguagem com uma sintaxe e semântica semelhante a do Verilog-HDL para modelagem de sistemas analógicos que sejam compatíveis com circuitos de simulação *SPICE-class* (14).

Contudo, com o objetivo de unir sistemas digitais e analógicos foi criado o Verilog-AMS que combina o Verilog-HDL juntamente com o Verilog A, além de se poder desenvolver sistemas com sinais mistos, possibilitando assim a integração entre sistemas analógicos, digitais e sinais mistos (14).

Dessa maneira, o Verilog-A é comumente utilizado na modelagem de circuitos para simulação e construção de sistemas analógicos e com o desenvolvimento do Verilog-AMS está crescendo cada vez mais rápido (14).

3 Projeto do Receptor

Este capítulo tem como objetivo apresentar o projeto dos blocos que compõe um receptor TTE utilizando circuitos integrados presentes no mercado, além de suas respectivas prototipações realizadas através do software *Altium Designer* com o objetivo de serem realizados suas caracterizações em bancada. Não obstante, será apresentado a modelagem em Verilog-AMS do LNA, filtro passa alta de 2º ordem, PGA e do receptor TTE utilizando a ferramenta *Virtuoso Analog Design Environment*, desenvolvida pela empresa *Cadence Design Systems*, no qual é a ferramenta ideal para produção de circuitos integrados.

3.1 Antena

3.1.1 Caracterização da antena *loop*

Para a caracterização das antenas *loop*, foram feitos testes de transmissão em recepção utilizando somente o conversor AD com capacitores, no quais tinham papel de ajustar a sintonização do sinal para 5040Hz. Os testes foram realizados em conjunto com os estudantes Breno Dantas, Gabriel Teixeira e o mestre Josua Peña.

Nesta caracterização foi utilizado a antena tag 2.0(figura 14) como antena do transmissor e a antena tag escudo(figura 15) como antena do receptor, de forma que foram feitos testes com distâncias de 7.5 e 10 metros ao passo que se variava a potência de transmissão(tabela 2).

	Pot. de TX (dBm)	Irms (mA)	Cap. RX (nF)	SNR (dB)	BER	Freq. Teste (kHz)	Distância
Tag 1.0 para Tag Escudo	26,43	111,9000	58,7000	17,9300	0,0000	5,0400	7,5m
Tag 2.0 para Tag Escudo	28,7976	192,9916	58,7000	32,4300	0,0000	5,0400	
Tag 2.0 para Tag Escudo	34,7700	385,0192	58,7000	39,6483	0,0000	5,0400	
Tag 2.0 para Tag Escudo	42,0707	891,7189	58,7000	46,3800	0,0000	5,0400	
Tag 2.0 para Tag Escudo	36,4500	481,1900	58,7000	30,3455	0,0000	5,0400	10m
Tag 2.0 para Tag Escudo	21,6626	85,0917	1(uF)	8,1386	0,0100	5,0400	
Tag 2.0 para Tag Escudo	24,6730	120,3148	1(uF)	9,2770	0,0000	5,0400	
Tag 2.0 para Tag escudo	27,6535	169,5748	1(uF)	13,6610	0,0000	5,0400	
Tag 2.0 para Tag Escudo	30,6712	239,9976	1(uF)	18,2623	0,0000	5,0400	
Tag 2.0 para Tag Escudo	33,1480	319,3489	1(uF)	20,2221	0,0000	5,0400	
Tag 2.0 para Tag Escudo	36,6659	479,1601	1(uF)	22,9300	0,0000	5,0400	

Tabela 2 – Tabela de resultados da caracterização da antena tag 2.0 e da antena tag escudo.



Figura 14 – Antena tag 2.0.



Figura 15 – Antena tag escudo.

3.2 LNA

O LNA é o primeiros bloco da cadeia de recepção, tendo assim um papel importante de amplificar o sinal recepcionado para que possa ser feito a leitura através de um conversor A/D que não necessite de uma alta sensibilidade. Não obstante, ao se inserir um amplificador todos os sinais recepcionados são amplificados, incluindo ruídos, dessa maneira é importante que o primeiro estágio de amplificação necessite de uma baixa figura de ruído e conseqüentemente uma figura de ruído total baixa, minimizando o impacto do ruído no momento em que o sinal for processado.

3.2.1 Projeto

Na equação 3.1 (15) é expresso o cálculo da tensão de ruído em amplificadores, em que ENB é a banda equivalente do ruído (aproximadamente 1.57 a frequência de corte do amplificador), $A=(R1+R2)/R1$, R1 é a resistência série, R2 é a resistência de realimentação, f_H e f_L são as frequências limite de operação, f_{enc} é a frequência de *corner* e e_W é a tensão de ruído do amp op.

$$E_{T_{rms}} = \sqrt{ENB4kTR_{2A} + e_W^2 A^2 \left(f_{enc} \ln \frac{f_H}{f_L} + ENB \right)} \quad (3.1)$$

Para se minimizar o ruído no amplificador deve-se escolher um amplificador com baixo ruído (e_W), além de resistências pequenas. Buscando um amplificador que possuísse uma boa relação custo versus e_W , optou-se pelo OP213. Em consequência, o CI OP213 possui uma densidade de ruído constante de $4.7nV/\sqrt{Hz}$ e produto ganho banda de $3.4MHz$ para frequências acima de 1kHz.

Para o projeto do LNA utilizou-se a topologia não inversora com alimentação single supply, como apresentado na figura 16, devido a sintonização do sinal de transmissão, pois em circuitos não inversores a impedância de entrada é alta, fazendo com que a banda de sintonização fique mais estreita, ou seja, interfere diretamente na largura de banda da resposta em frequência do circuito.

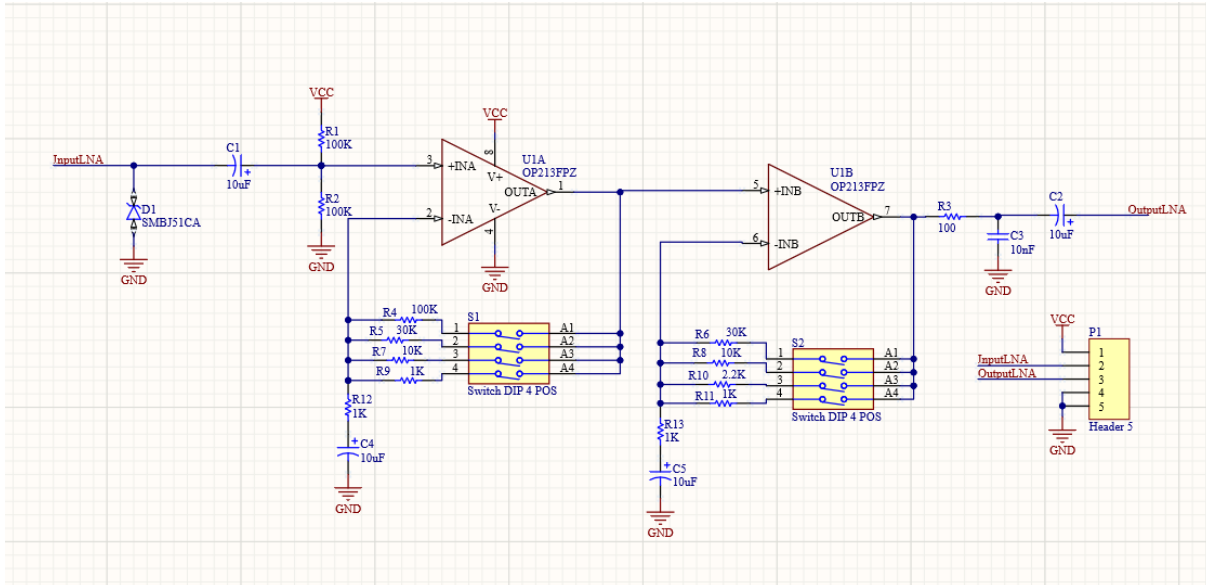


Figura 16 – Esquemático do LNA não inversor.

O resistor de entrada juntamente com o capacitor de acoplamento formam um filtro passa alta, possibilitando a fixação de uma frequência de -3dB e a rejeição de frequências próximas da rede, assim buscou-se escolher os resistores levando a relação da filtragem dos espúrios versus valores das resistências.

O resistor de 100Ω juntamente com o capacitor de 6.2nF resultam em um filtro RC passa baixa, auxiliando na rejeição de altas frequências. O diodo TVSD1 (*Transient Voltage Suppression*) protege o circuito de sobretensões eminentes da antena. O capacitor de $10\mu\text{F}$ na saída atua como um capacitor de desacoplamento DC. Como não se tinha um ganho pré estabelecido, foi necessário acrescentar duas chaves de 4 terminais de modo que pudessem ser testados diferentes ganhos. Os demais componentes (resistores e capacitores) atuam na polarização do circuito.

3.2.2 Modelagem

A modelagem do amplificador utilizado foi baseado nas características do amplificador OP213. O código do modelo completo do amplificador se encontra no apêndice A. A relação dos pinos de entrada e saída que compõe o amplificador apresentados na tabela 3.

Nome	Descrição
out	Saída
inn	Entrada inversora
inp	Entrada não inversora
agnd	Ground
avdd	Alimentação

Tabela 3 – Pinos amplificador LNA.

Inicialmente no código foi feito a declaração das bibliotecas padrões fornecidas automaticamente pelo Verilog-AMS. Em seguida são declarados os pinos e suas disciplinas correspondentes, além de declarar um nó interno e parâmetros e variáveis locais.

```

1 'include "constants.vams"
2 'include "disciplines.vams"
3
4 module rx_lna_opamp_v0(out, agnd, avdd, inp, inn);
5     //pins declaration
6     output out;
7     electrical out;
8     inout agnd;
9     electrical agnd;
10    inout avdd;
11    electrical avdd;
12    input inp, inn;
13    electrical inp, inn;
14    electrical out1; //internal node
15    //parameters declaration
16    localparam real gain_db=126;
17    parameter real zin=2k;
18    parameter real zout=10;
19    //local variables declaration
20    parameter real realgain = pow(10, gain_db/20.0);
21    parameter real zbias = 5K;

```

Figura 17 – Declaração dos pinos, parâmetros e variáveis

A parte do código que modela o comportamento do amplificador é determinado pelo processo "always", no qual descreve o comportamento discreto no tempo. São expressas as tensões entre as portas e o nó interno, além da potência consumida pelo circuito. A tensão de saída é determinada pelo parâmetro "realgain" e pela diferença de tensão entre a entrada não inversora e inversora.

```

1 analog begin
2   V(inp , inn) <+ zin*I(inp , inn); //input impedance
3   V(out1 , agnd) <+ realgain*V(inp , inn);
4   V(out , out1) <+ zout*I(out , out1);
5   //power consumption
6   V(avdd , agnd) <+ zbias*I(avdd , agnd);
7 end

```

Figura 18 – Módulo do amplificador utilizado no LNA.

A partir da modelagem do amplificador em Verilog-AMS foi possível criar o símbolo, 19 e em sequência as topologias do LNA inversor e não inversor, apresentados pelas figuras 20 e 21 respectivamente.

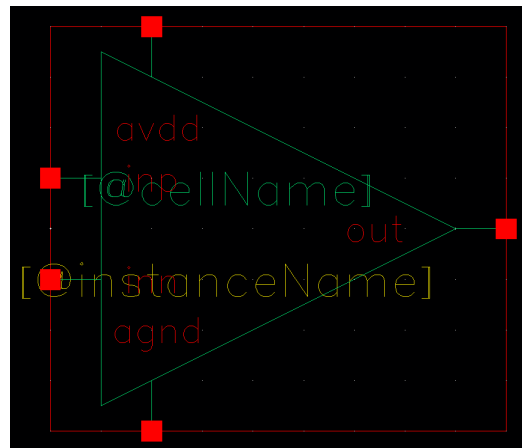


Figura 19 – Símbolo do amplificador modelado em Verilog-A.

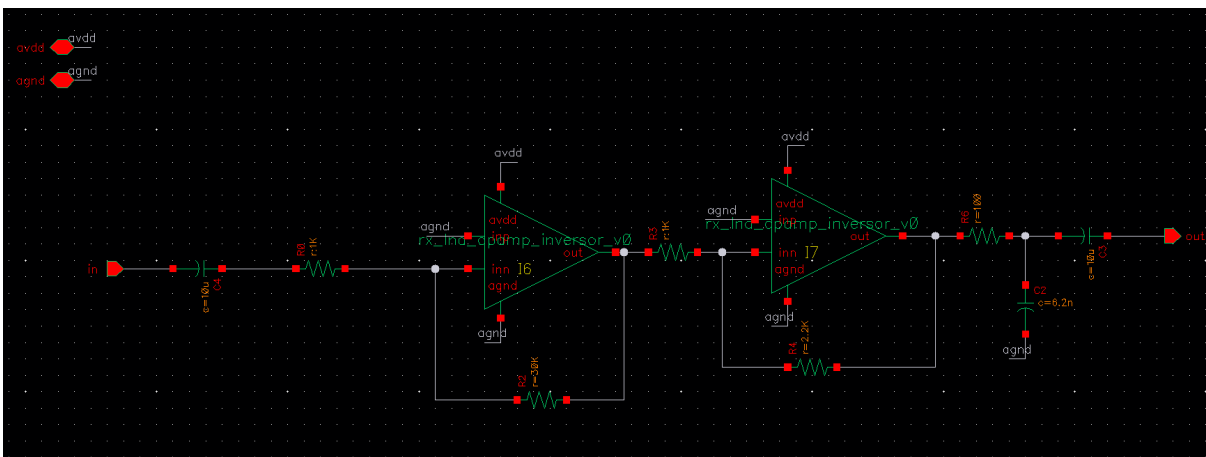


Figura 20 – Esquemático do LNA inversor.

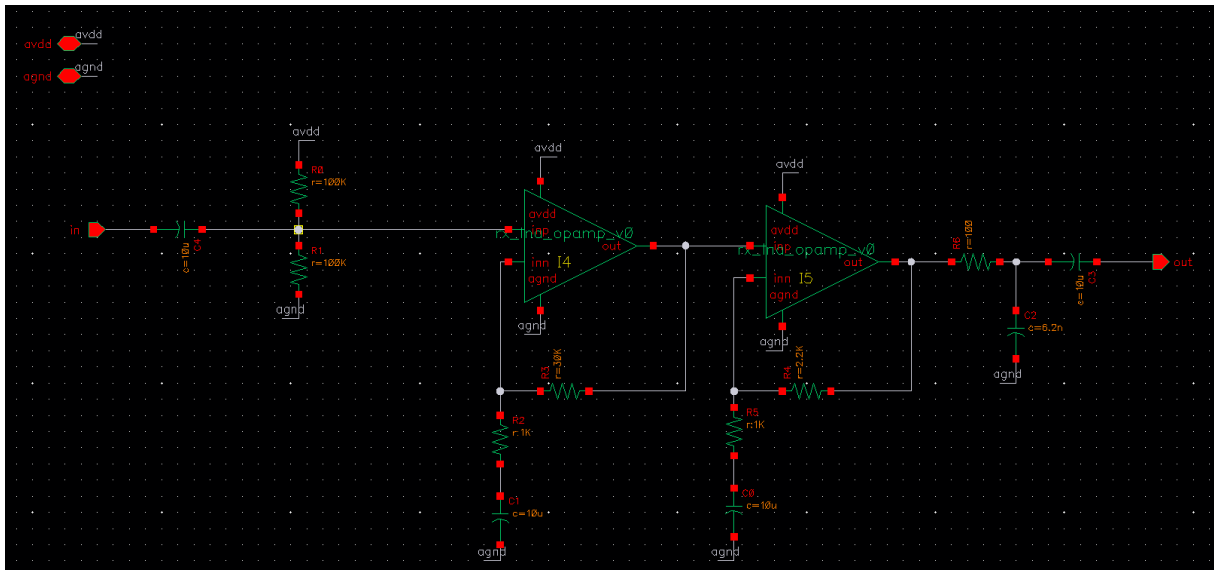


Figura 21 – Esquemático do LNA não inversor.

Por fim, foram criados símbolos a partir das topologias inversoras e não inversoras, como mostrado nas figuras 22 e 23 respectivamente.

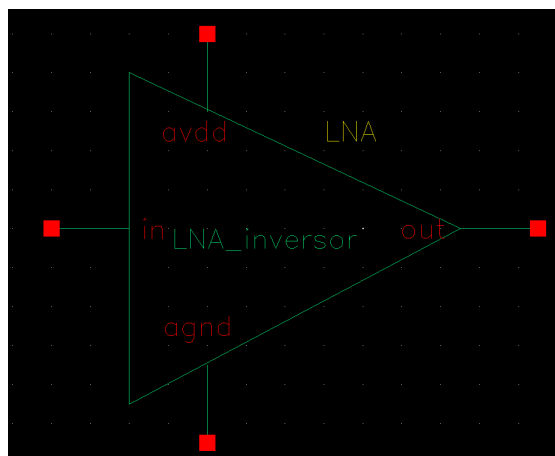


Figura 22 – Símbolo do LNA inversor.

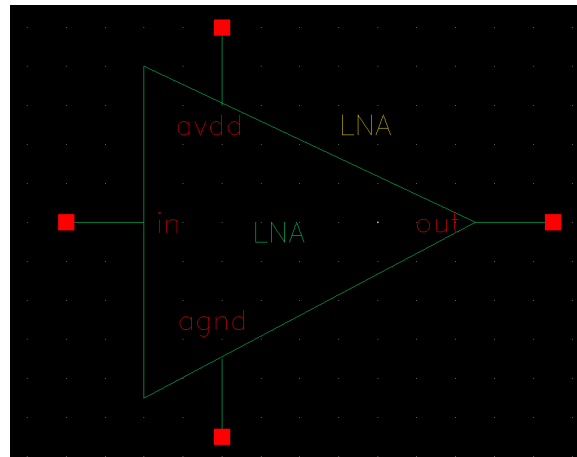


Figura 23 – Símbolo do LNA não inversor.

3.2.3 Protótipo

Com o objetivo de se realizar a caracterização em bancada foi feito o layout do LNA apresentado na figura 24 e sua respectiva prototipação, como mostrado na figura 25.

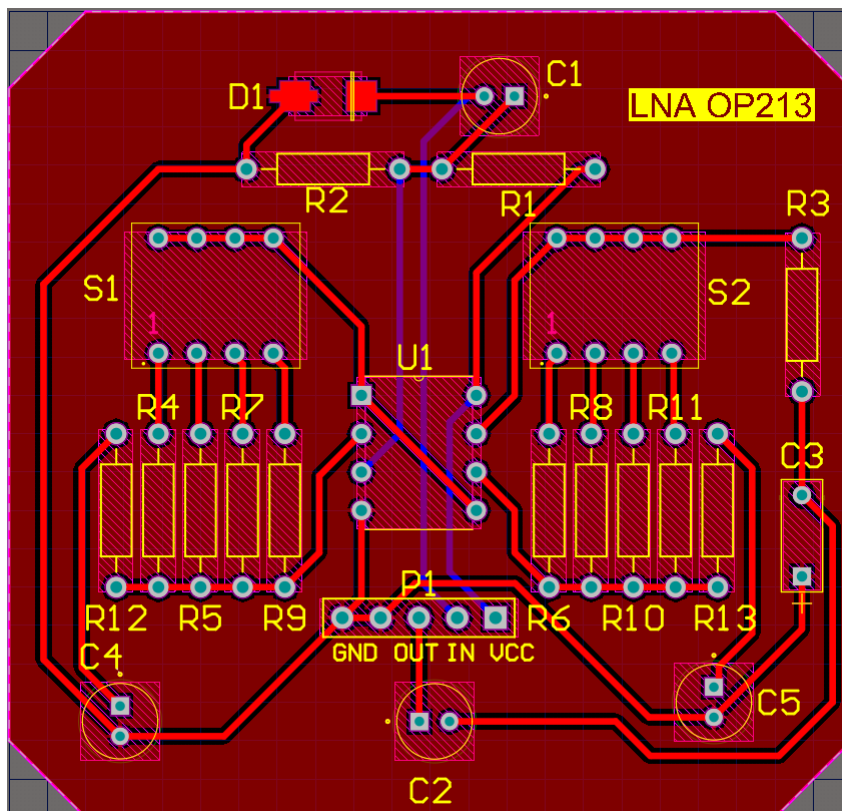


Figura 24 – Layout do LNA.

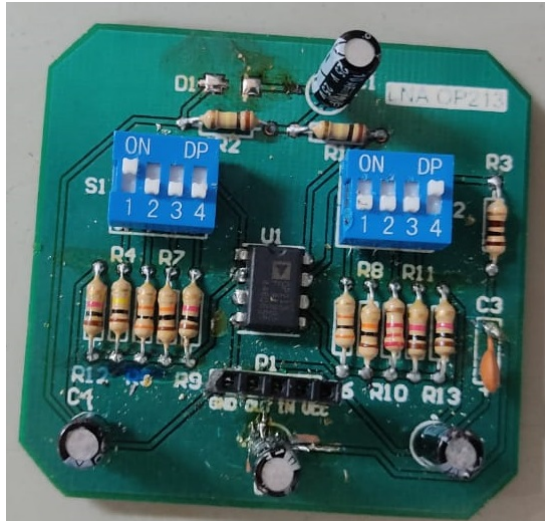


Figura 25 – Protótipo do LNA.

Para caracterização do LNA foram feitos testes em bancada, de maneira que utilizou-se os seguintes instrumentos e materiais: Fonte de tensão, osciloscópio, gerador de funções, cabos com conector BNC, jumpers, protótipo do LNA, multímetro e uma protoboard.

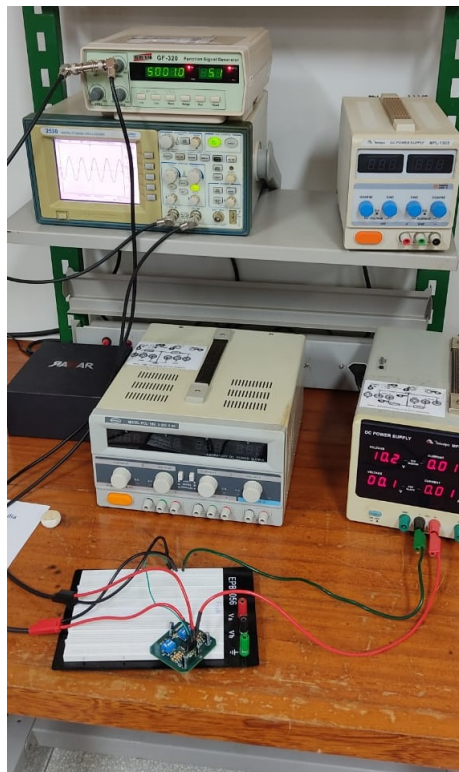


Figura 26 – Testes em bancada do LNA.

Para realização dos testes foram efetuados os seguintes procedimentos:

1. Ajustou-se a fonte de tensão para 10V single supply, dessa maneira a tensão de 10V irá no “VCC” e o ground irá no “GND” do protótipo;
2. Definiu-se uma senoide de 10mVpp com frequência de 5040Hz no gerador de funções, de modo que essa senoide é a entrada do circuito, representado por “IN” na figura [24](#);
3. No canal 01 do osciloscópio analisou-se o sinal gerado pelo gerador de funções e no canal 02 a saída do circuito, representado por “OUT” na figura [24](#);
4. Por fim, observou-se diferentes ganhos do circuito através da mudança dos valores dos resistores por meio das chaves S1 e S2.

3.3 Filtro passa faixa

Filtros são muito utilizados em circuitos de recepção devido a presença de interferências e ruídos dentro da faixa de frequência operante do circuito. Como para este projeto estamos trabalhando em baixa frequência é necessário utilizar um filtro passa faixa, devido a presença de ruídos e interferências já citados nesse texto.

3.3.1 Projeto

O FPF foi projetado utilizando o circuito integrado LT1568 da Linear Technology, pois é um CI que possui ótimas características para circuitos que necessitem ter um baixo ruído devido sua relação sinal ruído acima de 90dB.

Para a confecção do filtro foi utilizado um topologia de quarta ordem(figura [27](#)), permitindo uma rejeição de ruídos e interferências próximos a frequência atuante, além de ser um filtro anti-aliasing.

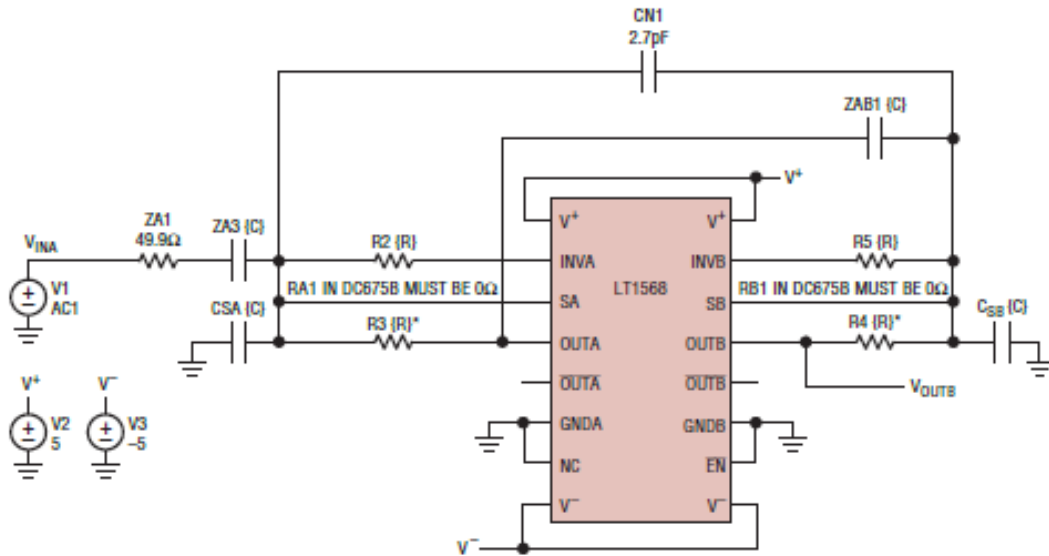


Figura 27 – Topologia para o projeto de um FPF de quarta ordem. (5)

Para o cálculo dos valores dos componentes, para um frequência de corte (f_C) de 5040Hz com ganho 1, utilizou-se as seguinte equações:

$$C = 70\sqrt{Gain} \quad (3.2)$$

$$R_2 = R_3 = R_4 = R_5 = R \quad (3.3)$$

$$R = \frac{34 \cdot 10^9}{\sqrt{10C + 700f_C}} \quad (3.4)$$

Por não possuir componentes comerciais dos valores encontrados a partir dos cálculos feitos, utilizou-se uma frequência de corte de 5084Hz ao invés de 5040Hz e substituiu-se o capacitor CN1 por um capacitor variável, permitindo com que seja possível ajustar a frequência de corte.

Portanto, foi projetado o circuito mostrado na figura 28, possuindo resistores comerciais de $180k\Omega$ e capacitores de 68pF.

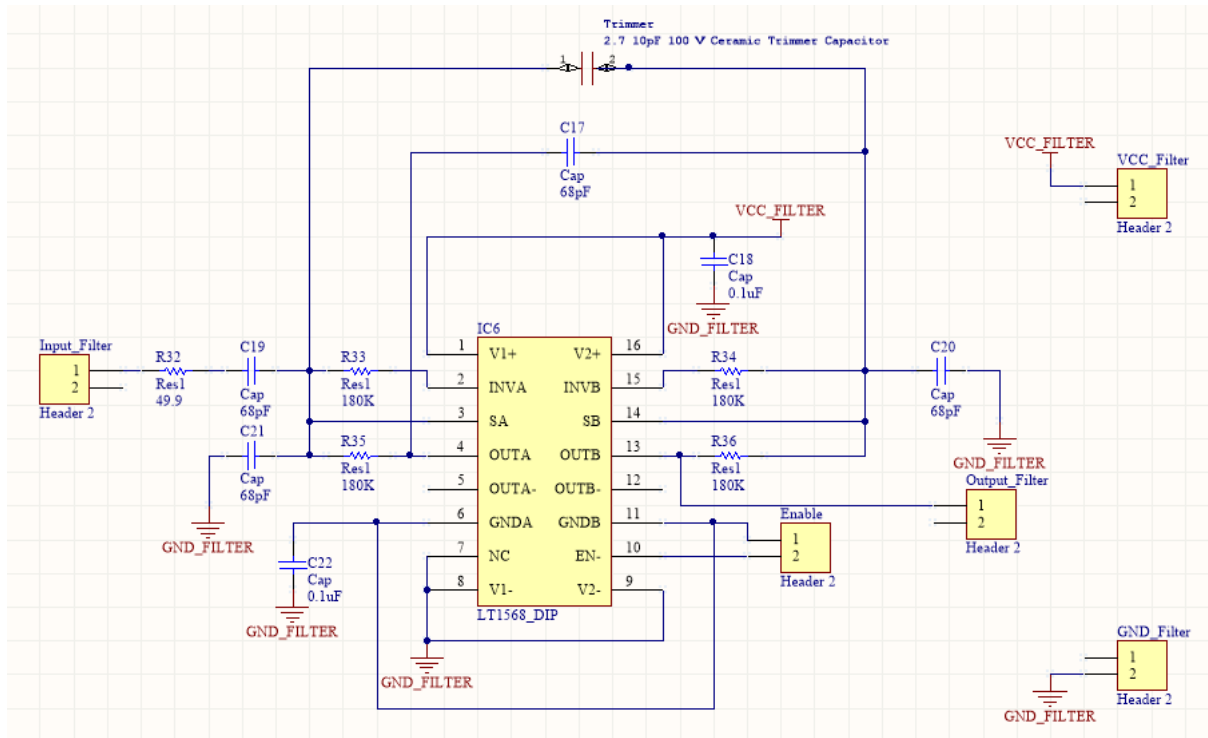


Figura 28 – Esquemático do FPF de quarta ordem.

3.3.2 Modelagem

A modelagem em Verilog-AMS do amplificador utilizado no esquemático do filtro utilizado foi baseado nas características do CI LTC1568. O código do modelo completo do amplificador se encontra no apêndice B. A relação dos pinos de entrada e saída que compõe o amplificador apresentados na tabela 4.

Nome	Descrição
out	Saída
inn	Entrada inversora
inp	Entrada não inversora
agnd	Ground
avdd	Alimentação

Tabela 4 – Pinos do amplificador utilizado no FPF.

Inicialmente no código foi feito a declaração das bibliotecas padrões fornecidas automaticamente pelo Verilog-AMS. Em seguida são declarados os pinos e suas disciplinas correspondentes, além de declarar um nó interno e parâmetros e variáveis locais.


```

1 'include "constants.vams"
2 'include "disciplines.vams"
3
4 module rx_lpf_opamp(out, agnd, avdd, inp, inn);
5 //pins declaration
6 output out;
7 electrical out;
8 inout agnd;
9 electrical agnd;
10 inout avdd;
11 electrical avdd;
12 input inp, inn;
13 electrical inp, inn;
14 electrical out1; //internal node
15 //parameters declaration
16 localparam real gain_db=130;
17 parameter real zin=40K;
18 parameter real zout=10;
19 //local variables declaration
20 parameter real realgain = pow(10, gain_db/20.0);
21 parameter real zbias = 5K;

```

Figura 29 – Declaração dos pinos, parâmetros e variáveis

No modelo são expressos as tensões entre as portas e o nó interno, a tensão de saída que é expressa pelo ganho e a diferença de tensão entre a entrada não inversora e inversora e a potência consumida pelo circuito.

```

1 analog begin
2 V(inp,inn) <+ zin*I(inp,inn); //input impedance
3 V(out1,agnd) <+ realgain*V(inp,inn);
4 V(out,out1) <+ zout*I(out,out1);
5 //power consumption
6 V(avdd,agnd) <+ zbias*I(avdd,agnd);
7 end

```

Figura 30 – Módulo do amplificador utilizado no FPF.

A fim de validar o circuito de filtragem, foi desenvolvido o esquemático de um filtro passa alta de segunda ordem apresentado pela figura 31 e respectivamente seu símbolo apresentado pela figura 32.

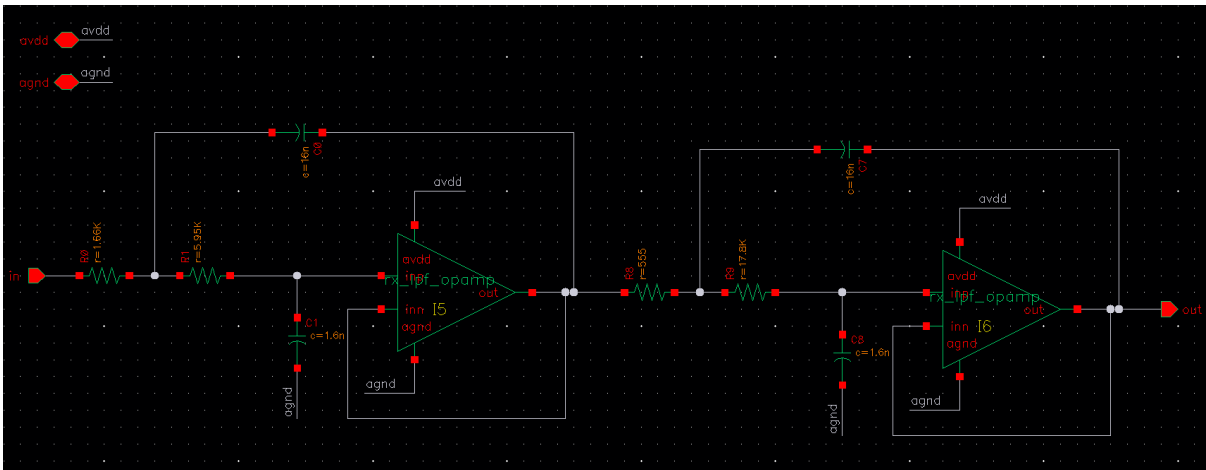


Figura 31 – Esquemático do filtro passa alta.

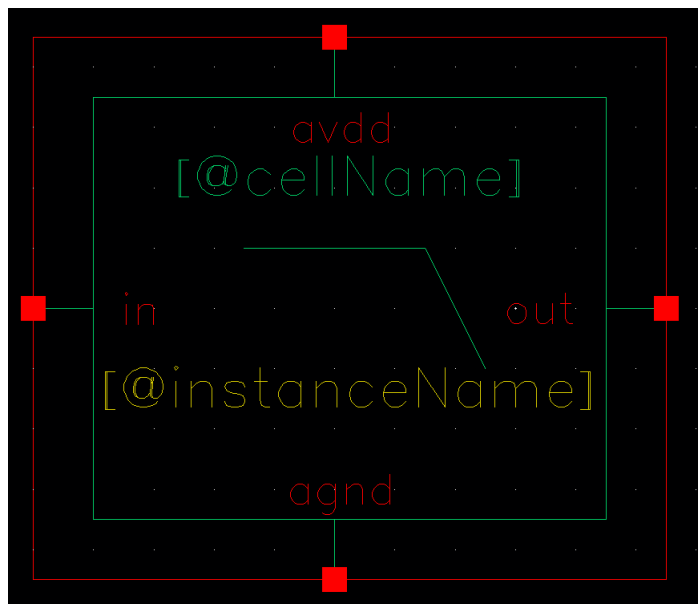


Figura 32 – Símbolo do filtro passa alta.

Além disso, com a intenção de comparar o comportamento do circuito da figura 31 e do comportamento ideal de um filtro passa alta, foi desenvolvido o modelo do filtro passa alta em Verilog-AMS, de modo que seus pinos estão apresentados na tabela 5.

Nome	Descrição
out	Saída
in	Entrada
agnd	Ground
avdd	Alimentação

Tabela 5 – Pinos do filtro passa alta de segunda ordem.

De início é feito a declaração padrão das bibliotecas em conjunto com os parâmetros e variáveis pertencentes ao modelo do filtro passa alta.

```

1 'include "constants.vams"
2 'include "disciplines.vams"
3
4 module rx_lpf(out, agnd, avdd, in);
5     //pins declaration
6     output out;
7     electrical out;
8     inout agnd;
9     electrical agnd;
10    inout avdd;
11    electrical avdd;
12    input in;
13    electrical in;
14    electrical out1; //internal node
15    //parameters declaration
16    localparam real gain_db=10;
17    parameter real zin=1K;
18    parameter real zout=10;
19    parameter real f3db=5.04k;
20    //local variables declaration
21    parameter real realgain = pow(10, gain_db/20.0);
22    parameter real zbias = 5K;

```

Figura 33 – Pinos, parâmetros e variáveis do filtro passa alta.

No modelo são expressas as tensão entre os pinos o nó interno, de modo que a tensão entre o nó interno "out1" e o ground é expresso por uma fórmula disponibilizada pelo Verilog-AMS, a fórmula é a "laplace_nd", no qual necessita do operador, dos polos e zeros da função de transferência que o representa, como possível observar na figura 34.

```

1 analog begin
2     V(in, agnd) <+ zin*I(in, agnd); //input impedance
3     V(out1, agnd) <+ realgain*laplace_nd(V(in, agnd), {1}, {1, 1/'M_TWO_PI/f3db
4         });
5     V(out, out1) <+ zout*I(out, out1);
6     //power consumption
7     V(avdd, agnd) <+ zbias*I(avdd, agnd);
8 end

```

Figura 34 – Módulo do filtro passa alta.

3.3.3 Protótipo

Para realização da caracterização em bancada foi feito o primeiro layout da PCI (figura 35) e sua placa de circuito impresso (figura 36).

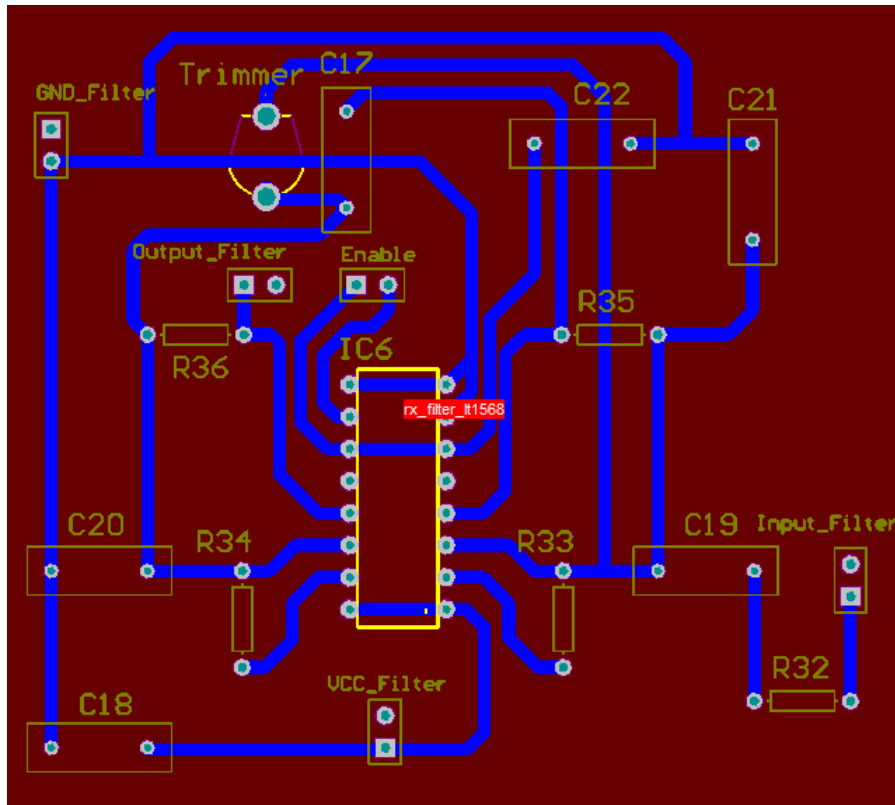


Figura 35 – Layout do FPF de quarta ordem.

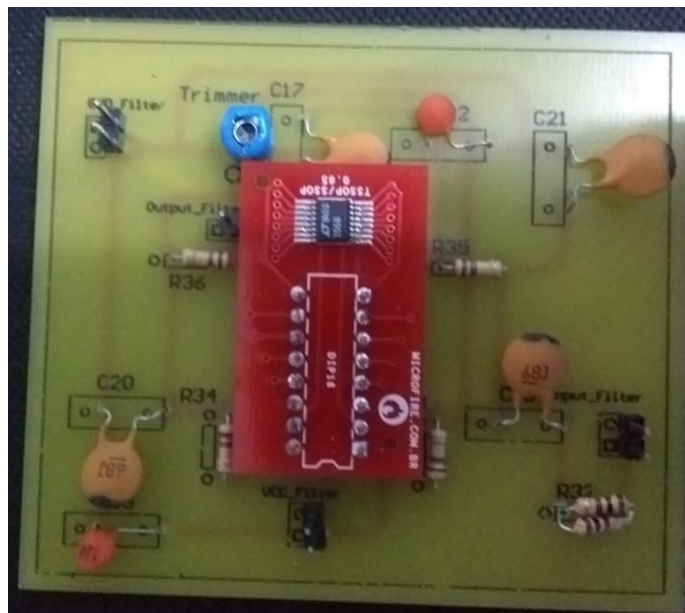


Figura 36 – PCI do FPF de quarta ordem.

Todavia, por conta dos problemas encontrados na primeira caracterização, que serão apresentados na seção de resultados, necessitou-se o re-projeto da placa de circuito impresso do FPF, como apresentado na na figura 37.

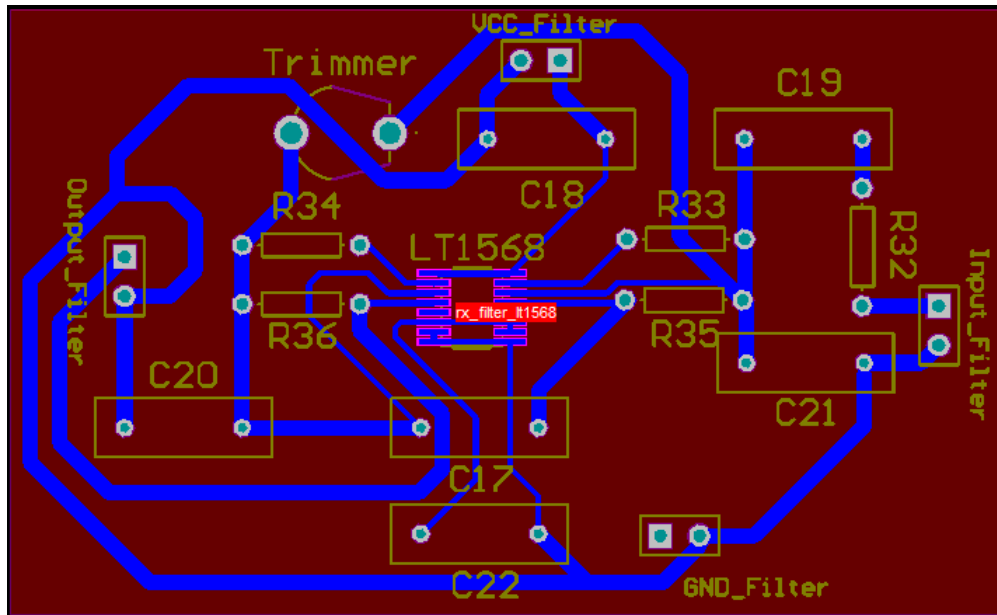


Figura 37 – Segundo layout do FPF de quarta ordem.

De maneira similar ao LNA, para a caracterização do circuito necessitou-se de alguns instrumentos e materiais: Fonte de tensão, osciloscópio, gerador de funções, cabos com conector BNC, jumpers, protótipo do LNA, multímetro e uma protoboard.

Para realização dos testes foram efetuados os seguintes procedimentos:

1. Ajustou-se a fonte de tensão para 5V single supply, dessa maneira a tensão de 5V irá no “VCC_Filter” e o ground irá no “GND_Filter” do protótipo;
2. Definiu-se uma senoide de 1Vpp com frequência de 5040Hz no gerador de funções, de modo que essa senoide é a entrada do circuito, representado por “Input_Filter” na figura 37;
3. No canal 01 do osciloscópio analisou-se o sinal gerado pelo gerador de funções e no canal 02 a saída do circuito, representado por “Output_Filter” na figura 37;
4. Observou-se se o sinal foi mantido para a frequência desejada e apresentou pouca atenuação ou não apresentou;
5. Por fim, ajustou-se a frequência do sinal gerado para cerca de 10kHz e posteriormente 100kHz, podendo assim analisar a atenuação presente no sinal.

3.4 PGA

O PGA é empregado em um sistema quando o sinal de entrada pode variar dentro de uma grande faixa dinâmica de tensões. Portanto, um sinal de entrada com baixa

amplitude sofrerá um alto ganho, já um sinal com alta amplitude irá ter um baixo ou nenhum ganho (13).

3.4.1 Projeto

Para o projeto do PGA foi utilizado o CI LTC6911-1 da *Linear Technology* devido seu baixo ruído de entrada, cerca de $10nV/\sqrt{Hz}$, 3 entradas digitais para controle do ganho e configuração single supply. A topologia adotada está representada na figura 38. Para este trabalho projetou-se um PGA com alimentação single supply de 3.3V, deste modo, a relação entre o ganho, entradas digitais e máxima tensão de entrada estão apresentados na tabela 6.

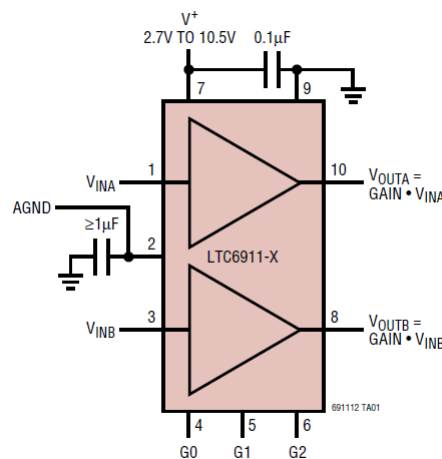


Figura 38 – Típica aplicação do CI LTC6911-1.

Entradas Digitais			Ganho Nominal)		Máxima tensão de entrada Linear (Vp-p)
G2	G1	G0	Volts/Volt	(dB)	Single 3V Supply
0	0	0	0	-120	3
0	0	1	-1	0	3
0	1	0	-2	6	1.5
0	1	1	-5	14	0.6
1	0	0	-10	20	0.3
1	0	1	-20	26	0.15
1	1	0	-50	34	0.06
1	1	1	-100	40	0.03

Tabela 6 – Ganhos e propriedades do CI LTC6911-1.

O controle das portas digitais do PGA foi feito pelo microprocessador *Raspberry pi 3 Model B* da empresa *Raspberrypi*, já que este microprocessador será utilizado na demodulação do sinal. Para se ter um controle de ganho automático, o sinal de entrada do PGA é lido pela *Raspberry* e posteriormente é feita a tomada de decisão dos valores

das entradas digitais do PGA. Contudo, a *Raspberry* não possui nenhum conversor A/D, sendo necessário utilizar o conversor externo ADS1115 da empresa *Texas Instruments*, pois ele possui 16 bits de resolução. Por ser possível realizar apenas a leitura de sinais com amplitudes positivas no conversor A/D, utilizou-se o amplificador LM358 da empresa *Texas Instruments*, no qual se projetou um amplificador somador antes do sinal ser lido pela *Raspberry*, desta maneira é possível adicionar um tensão DC de 1.65V ao sinal, pois a máxima amplitude de entrada no PGA é de 3.3V pico a pico. Em adição, inseriu-se um *buffer* entre a entrada do PGA e a entrada analógica do ADS1115, prevenindo que o sinal sofra alguma alteração ao entrar no PGA. O esquemático final pode ser observado na figura 39.

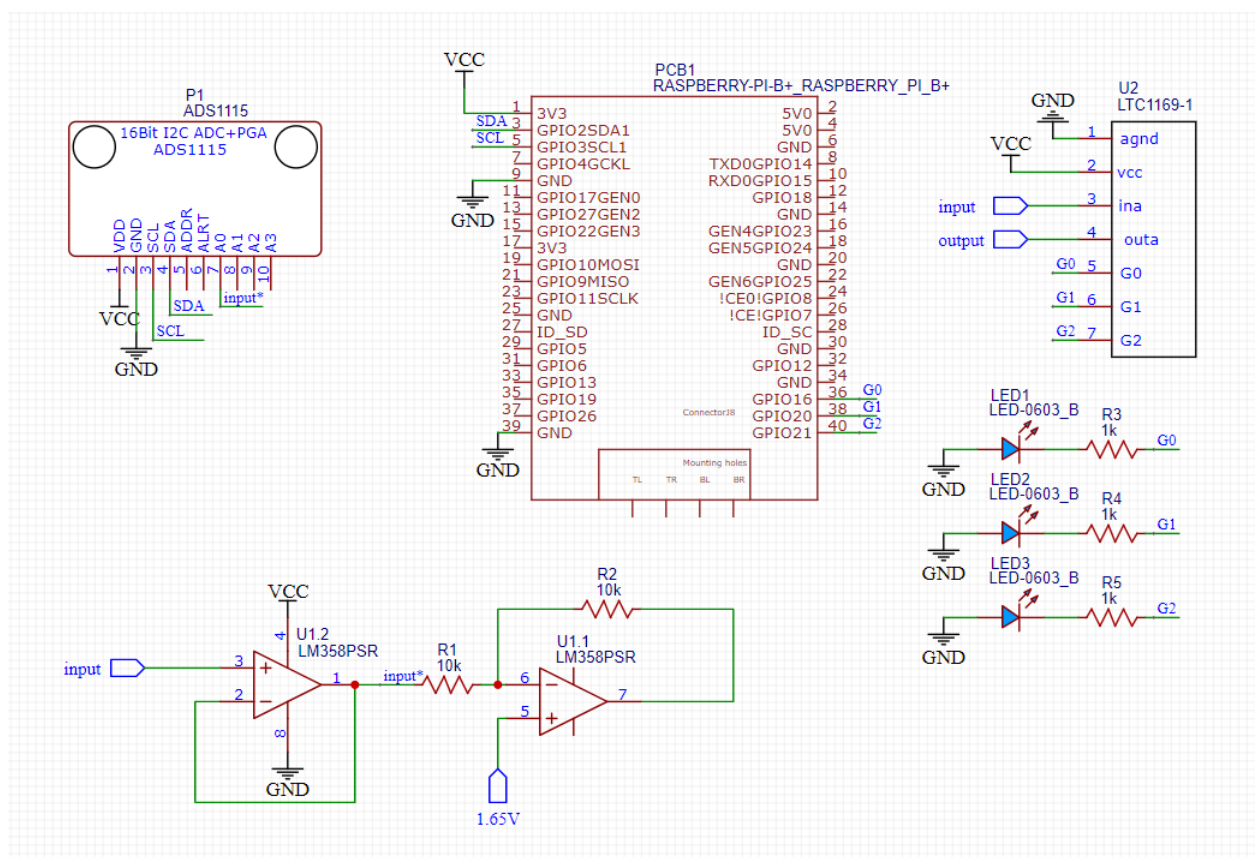


Figura 39 – Esquemático do PGA.

3.4.2 Lógica de programação

Com o intuito de se realizar um controle automático do ganho, desenvolveu-se um código em C, no qual foi baseado no fluxograma de funcionamento (figura 40).

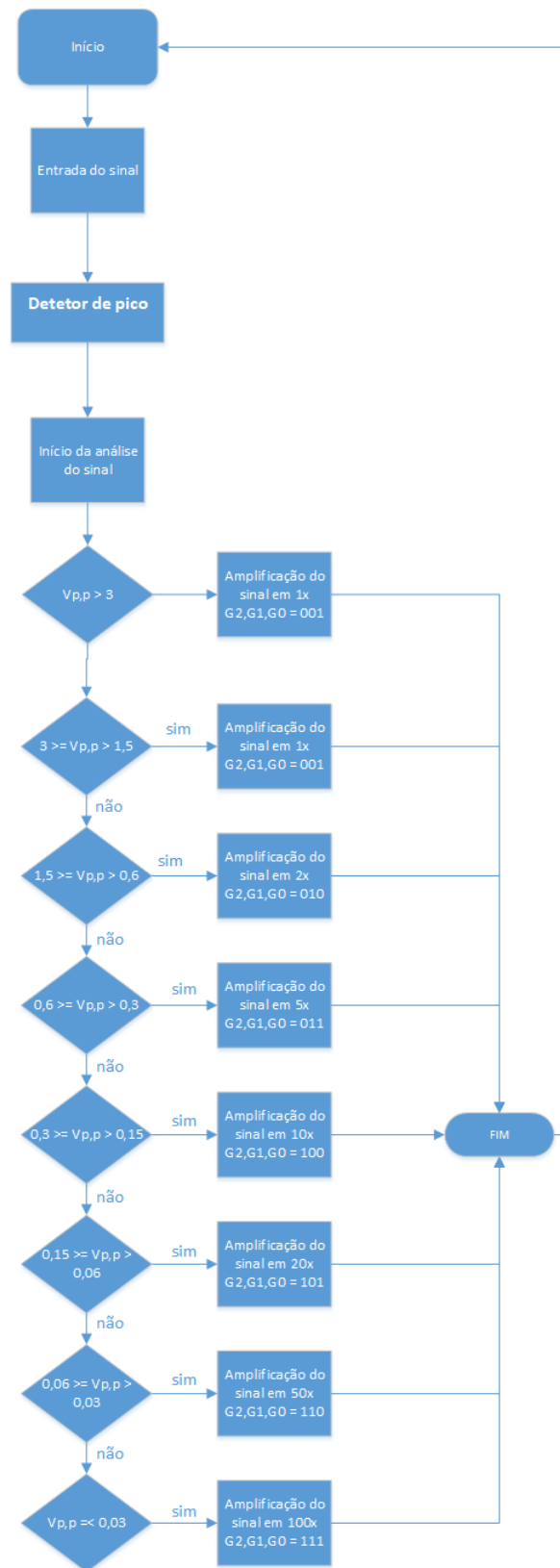


Figura 40 – Diagrama de funcionalidade do PGA para uma alimentação single supply de 3.3V.

No bloco detetor de pico é feito a leitura do sinal analógico, no qual é armazenado numa variável de transição. Em sequência esta variável de transição é comparada a uma variável que irá armazenar a maior amplitude e a outra variável que será armazenada a menor amplitude do sinal. Por fim, é feito o cálculo da tensão pico a pico do sinal. Esta lógica pode ser analisada através da figura 41.

```

1 void detetorDePico() {
2
3     if(openI2CBus("/dev/i2c-1") == -1)
4     {
5         printf("EXIT_FAILURE");
6     }
7     setI2CSlave(0x48);
8
9     while(1)
10    {
11        valorDeTransicao = readVoltage(0);
12        if (valorDePicoPos <= valorDeTransicao && valorDeTransicao <= 3.3) {
13            if (gpio(26,0,0)){
14                valorDePicoNeg = 3.3;
15                valorDePicoPos = 0;
16            }
17            valorDePicoPos = valorDeTransicao;
18        } else if (valorDePicoNeg >= valorDeTransicao && valorDeTransicao >=
19            0) {
20            if (gpio(26,0,0)){
21                valorDePicoNeg = 3.3;
22                valorDePicoPos = 0;
23            }
24            valorDePicoNeg = valorDeTransicao;
25        } else {
26            if (gpio(26,0,0)){
27                valorDePicoNeg = 3.3;
28                valorDePicoPos = 0;
29            }
30            valorDeTransicao = valorDeTransicao;
31        }
32        Vpp = valorDePicoPos - valorDePicoNeg;
33
34        printf("Valtage pico a pico = %.3f V \n", Vpp);
35        detetorDeNivel();
36        delay(0.3);
37    }

```

Figura 41 – Função detetor de pico do PGA.

Após analisar qual a tensão pico a pico do sinal, é feito a análise em qual intervalo essa tensão se encontra, para assim poder determinar qual são os valores das entradas digitais do PGA e por consequência realizar a amplificação ou não do sinal. É possível observar a função desenvolvida na figura 42. O código completo pode ser visto no apêndice D.

```

1 void detetorDeNivel() {
2   if (Vpp > 3.3){
3     Vout = Vpp*1;
4     G2 = 0; G1 = 0; G0 = 1;
5     gpio(21, 1, 0);
6     gpio(20, 1, 0);
7     gpio(16, 1, 1);
8   } else if (3.3 >= Vpp && Vpp > 1.5){
9     Vout = Vpp*1;
10    G2 = 0; G1 = 0; G0 = 1;
11    gpio(21, 1, 0); //G2
12    gpio(20, 1, 0); //G1
13    gpio(16, 1, 1); //G0
14  } else if (1.5 >= Vpp && Vpp > 0.6){
15    Vout = Vpp*2;
16    G2 = 0; G1 = 1; G0 = 0;
17    gpio(21, 1, 0); //G2
18    gpio(20, 1, 1); //G1
19    gpio(16, 1, 0); //G0
20  } else if (0.6 >= Vpp && Vpp > 0.3){
21    Vout = Vpp*5;
22    G2 = 0; G1 = 1; G0 = 1;
23    gpio(21, 1, 0); //G2
24    gpio(20, 1, 1); //G1
25    gpio(16, 1, 1); //G0
26  } else if (0.3 >= Vpp && Vpp > 0.15){
27    Vout = Vpp*10;
28    G2 = 1; G1 = 0; G0 = 0;
29    gpio(21, 1, 1); //G2
30    gpio(20, 1, 0); //G1
31    gpio(16, 1, 0); //G0
32  } else if (0.15 >= Vpp && Vpp > 0.06){
33    Vout = Vpp*20;
34    G2 = 1; G1 = 0; G0 = 1;
35    gpio(21, 1, 1); //G2
36    gpio(20, 1, 0); //G1
37    gpio(16, 1, 1); //G0
38  } else if (0.06 >= Vpp && Vpp > 0.03){
39    Vout = Vpp*50;
40    G2 = 1; G1 = 1; G0 = 0;
41    gpio(21, 1, 1); //G2
42    gpio(20, 1, 1); //G1
43    gpio(16, 1, 0); //G0
44  } else if (Vpp <= 0.03){
45    Vout = Vpp*100;
46    G2 = 1; G1 = 1; G0 = 1;
47    gpio(21, 1, 1); //G2
48    gpio(20, 1, 1); //G1
49    gpio(16, 1, 1); //G0
50  } else {
51    Vout = 0;
52    G2 = 0; G1 = 0; G0 = 0;
53    gpio(21, 1, 0); //G2
54    gpio(20, 1, 0); //G1
55    gpio(16, 1, 0); //G0
56  }
57  printf("Tens o de sa da = %.3f V \n", Vout);
58  printf("Sa da G = %d %d %d\n", G2, G1, G0);
59 }

```

Figura 42 – Função detetor de nível do PGA.

3.4.3 Modelagem

O PGA foi modelado de acordo com as características do CI LTC6911-1, de modo que o modelo completo do amplificador se encontra no apêndice B. A relação dos pinos de entrada e saída que compõe o PGA estão apresentados na tabela 7.

Nome	Descrição
out	Saída
in	Entrada inversora
agnd	Ground
avdd	Alimentação
G2	Bit 3 da entrada digital
G1	Bit 2 da entrada digital
G0	Bit 1 da entrada digital

Tabela 7 – Pinos de entrada e saída do PGA.

Por padrão, de início foi feito a a declaração das bibliotecas padrões fornecidas automaticamente pelo Verilog-AMS. Em seguida são declarados os pinos e suas disciplinas correspondentes, além de declarar um nó interno e parâmetros e variáveis locais.

```

1
2 'include "constants.vams"
3 'include "disciplines.vams"
4
5 module rx_pga1(out , agnd , avdd , in , G0 , G1 , G2);
6
7 //pins declaration
8 output out;
9 electrical out;
10 inout agnd;
11 electrical agnd;
12 inout avdd;
13 electrical avdd;
14 input in;
15 electrical in;
16 input G0;
17 electrical G0;
18 input G1;
19 electrical G1;
20 input G2;
21 electrical G2;
22 electrical out1; //internal node
23
24 //parameters declaration
25 real gain;
26 real zin;
27 real realgain;
28 parameter real zout=10;
29 parameter real zbias = 5K;

```

Figura 43 – Declaração dos pinos, parâmetros e variáveis do PGA.

No modelo do PGA levou-se em consideração a logica apresentada na seção de lógica de programação, de modo que as entradas digitais podem ser 5V representado pelo valor binário 1 ou 0V representado pela valor binário 0. Ao ser realizado a análise das entradas digitais é definido o ganho e a impedância de entrada do PGA. Por fim, é feito são definidos as equações que definem a tensão entre entrada e terra, nó interno e terra, saída e nó interno, além da potência consumida pelo circuito. O símbolo gerado pelo modelo Verilog-AMS do PGA é apresentado na figura 45.

```

1 analog begin
2     if(V(G2, agnd) == 0 && V(G1, agnd) == 0 && V(G0, agnd) == 0)
3         gain=0;zin=10k;
4     if(V(G2, agnd) == 0 && V(G1, agnd) == 0 && V(G0, agnd) == 5)
5         gain=0;zin=5k;
6     if(V(G2, agnd) == 0 && V(G1, agnd) == 5 && V(G0, agnd) == 0)
7         gain=6;zin=2k;
8     if(V(G2, agnd) == 0 && V(G1, agnd) == 5 && V(G0, agnd) == 5)
9         gain=14;zin=1k;
10    if(V(G2, agnd) == 5 && V(G1, agnd) == 0 && V(G0, agnd) == 0)
11        gain=20;zin=1k;
12    if(V(G2, agnd) == 5 && V(G1, agnd) == 0 && V(G0, agnd) == 5)
13        gain=26;zin=1k;
14    if(V(G2, agnd) == 5 && V(G1, agnd) == 5 && V(G0, agnd) == 0)
15        gain=34;zin=1k;
16    if(V(G2, agnd) == 5 && V(G1, agnd) == 5 && V(G0, agnd) == 5)
17        gain=40;zin=1k;
18
19    realgain = pow(10, gain/20.0);
20
21    V(in, agnd) <+ zin*I(in, agnd);
22    V(out1, agnd) <+ realgain*V(in, agnd);
23    V(out, out1) <+ zout*I(out, out1);
24
25    V(avdd, agnd) <+ zbias*I(avdd, agnd);
26 end

```

Figura 44 – Modelo do PGA.

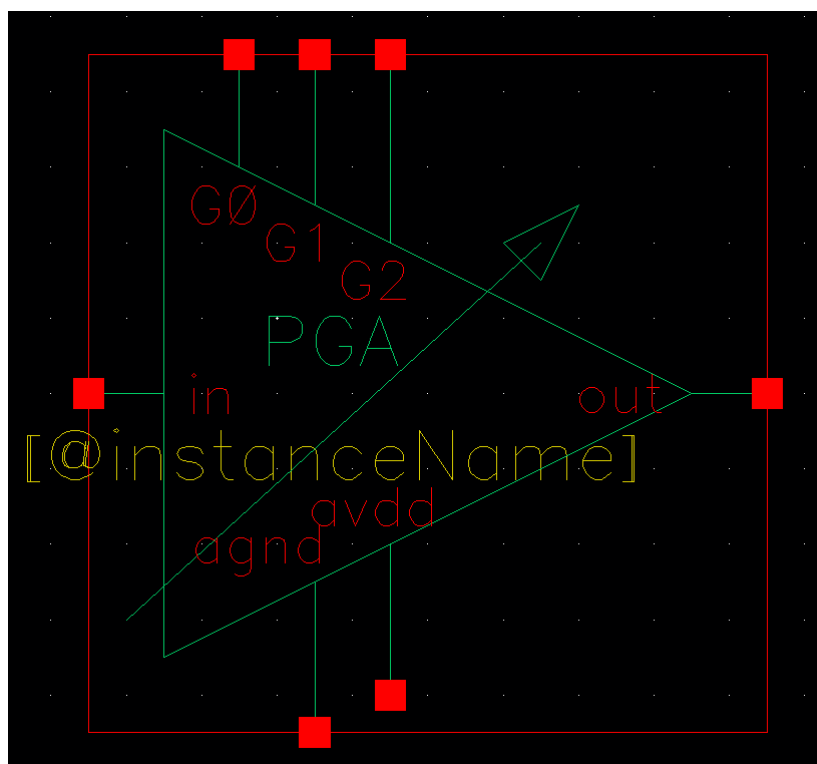


Figura 45 – Símbolo do PGA.

3.4.4 Protótipo

A fim de serem feitos testes em bancada do PGA foi desenvolvido o layout apenas do CI LTC6911-1 junto com capacitores de polarização (figura 46) e seu respectivo protótipo (figura 47).

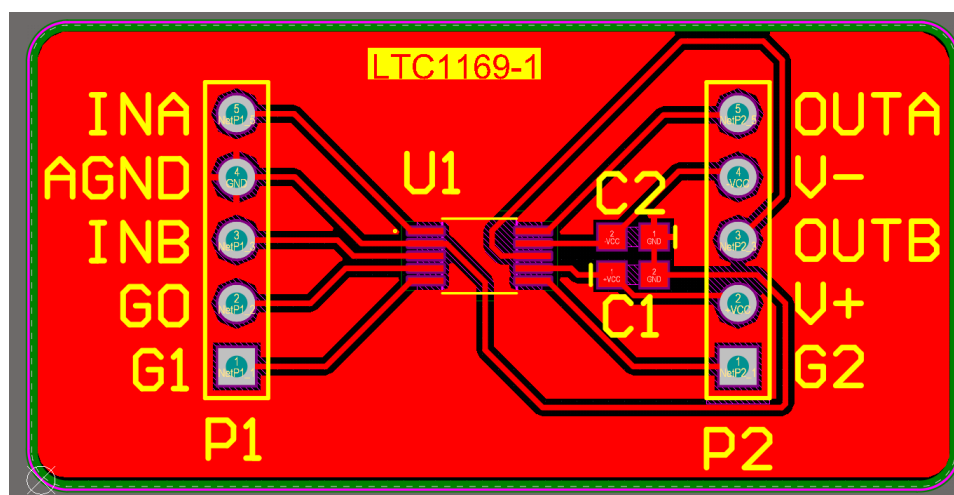


Figura 46 – Layout da placa de circuito impresso do PGA.

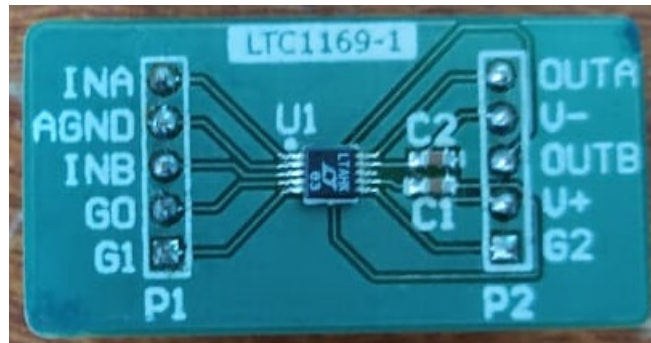


Figura 47 – Protótipo do PGA.

Para caracterização do protótipo do PGA foi necessário utilizar um microprocessador *Raspberry pi 3B+*, um conversor analógico digital ADS1115, um amplificador LM358, leds, resistores, potenciômetro, botão, um notebook, fonte de tensão, osciloscópio, gerador de funções, cabos com conector BNC, jumpers, protótipo do LNA, multímetro e uma protoboard.

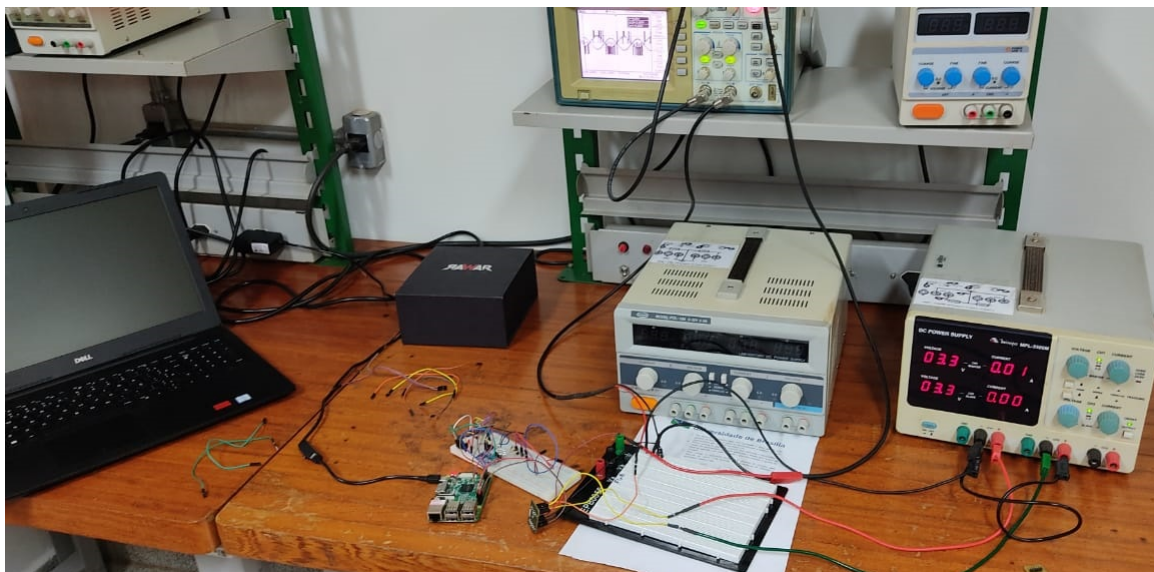


Figura 48 – Testes em bancada do PGA.

Para os testes foram realizados dois procedimentos distintos, no primeiro procedimento foram realizados os seguintes passos:

1. Montagem do circuito em protoboard(figura 49):

- Alimentar uma trilha com 3.3V da *Raspberry* e outra com o *ground*;
- Conectar o SDA e SCL do ADS1115 com os GPIOs 2 e 3 da *Raspberry* respectivamente;

- Conectar o sinal de entrada no amplificador somador feito a partir do LM358 e a entrada "INA" do PGA;
 - Conectar a saída do amplificador somador ao A0 do ADS1115;
 - Conectar os GPIOs 21, 20 e 16 da *Raspberry* ao G2, G1 e G0 do PGA respectivamente;
 - A fim de restabelecer os valores iniciais das variáveis lidas pelo sistema, foi inserido um botão no GPIO26, de modo que ao apertar será acionado o led vermelho ligado ao GPIO 19.
2. Ajustar a fonte de tensão para 3.3V single supply, dessa maneira o 3.3V irá no "VCC" e o ground irá no "GND" do protótipo;
 3. No canal 01 do osciloscópio será analisado o sinal gerado pelo gerador de funções e no canal 02 será analisado a saída do circuito, representado por "OUTA" na figura 01;
 4. Para a análise de ganho do circuito será utilizada a ferramenta de cursor do osciloscópio, podendo analisado a tensão pico a pico do sinal de entrada e do sinal de saída, e por fim calcular o ganho do circuito;
 5. Para validar os diferentes estágios de ganhos do circuito será inserido no "INA" uma senoide com frequência de 5040Hz e diferentes valores de amplitude.

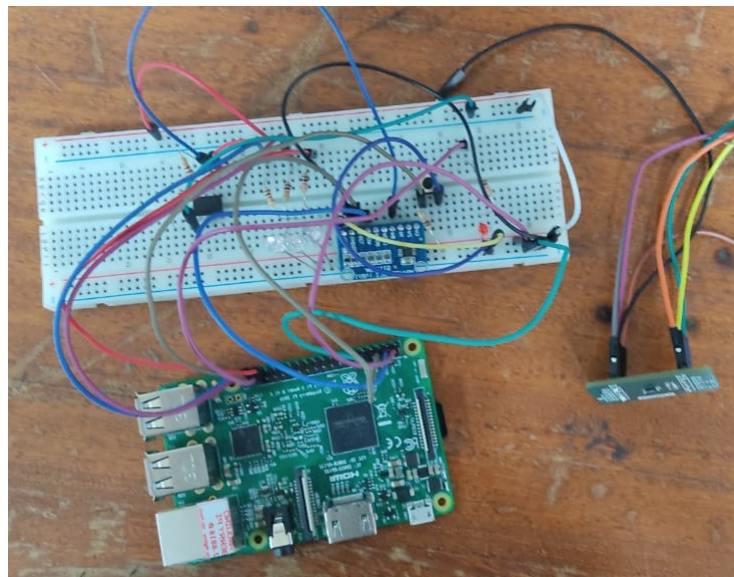


Figura 49 – Montagem do primeiro procedimento do PGA.

Para o segundo procedimento se difere apenas o processo de montagem do circuito em protoboard, no qual é:

1. Montagem do circuito em protoboard(figura 50):

- Alimentação simétrica do PGA com 3.3V;
- Conectar o SDA e SCL do ADS1115 com os GPIOs 2 e 3 da *Raspberry* respectivamente;
- Aplicar um tensão de 3.3V em um potenciômetro de $10k\Omega$, de modo que a tensão será lida diretamente pela porta analógica A0 do ADS1115, e por consequência gerar as entradas digitais do PGA;
- Conectar os GPIOs 21, 20 e 16 da *Raspberry* ao G2, G1 e G0 do PGA respectivamente;

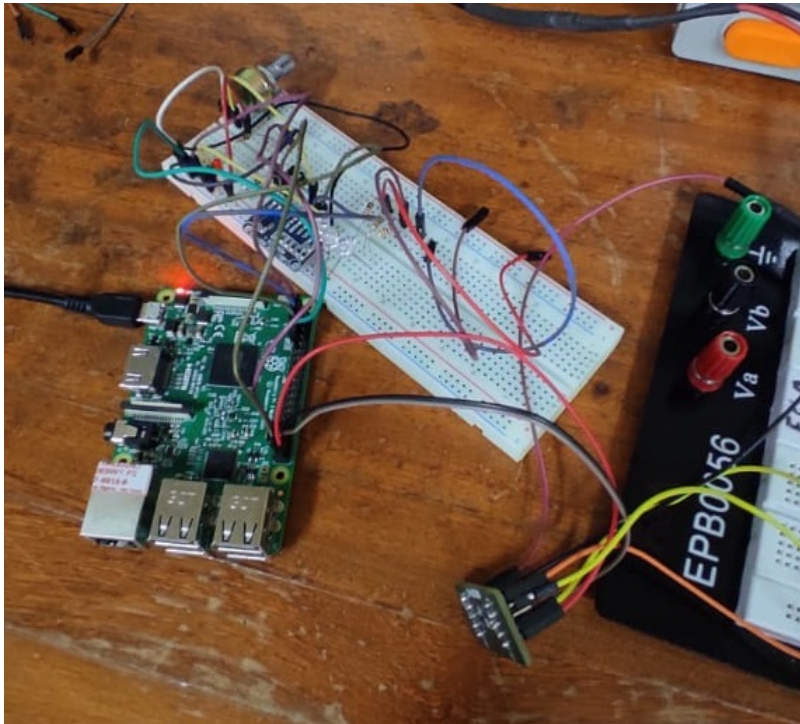


Figura 50 – Montagem do segundo procedimento do PGA.

3.5 Receptor TTE

3.5.1 Modelagem

Para validação de todo circuito de recepção foi realizado um simulação mistas com os blocos criados e apresentados anteriormente, além da criação de diferentes topologias de circuitos de recepção, podendo ser realizado a comparação entre os blocos. Em todos os circuitos se inseriu um oscilador LDO modelado em Verilog-AMS e um circuito RLC para simular a antena de recepção.

Inicialmente foram criados circuitos de recepção sem a presença do PGA, avaliando apenas o comportamento do filtro em conjunto com o LNA. Na intenção de diferir as topologias do LNA inversor e não inversor e do filtro de 2º ordem e 4º ordem, foram feitos 3 esquemáticos distintos: LNA inversor com filtro de segunda ordem (figura 51), LNA não inversor com filtro de segunda ordem (figura 52) e LNA não inversor com filtro de quarta ordem (figura 53). E por fim, foi projetado um filtro com um LNA inversor, filtro passa baixa de 4º ordem e o PGA (figura 54).

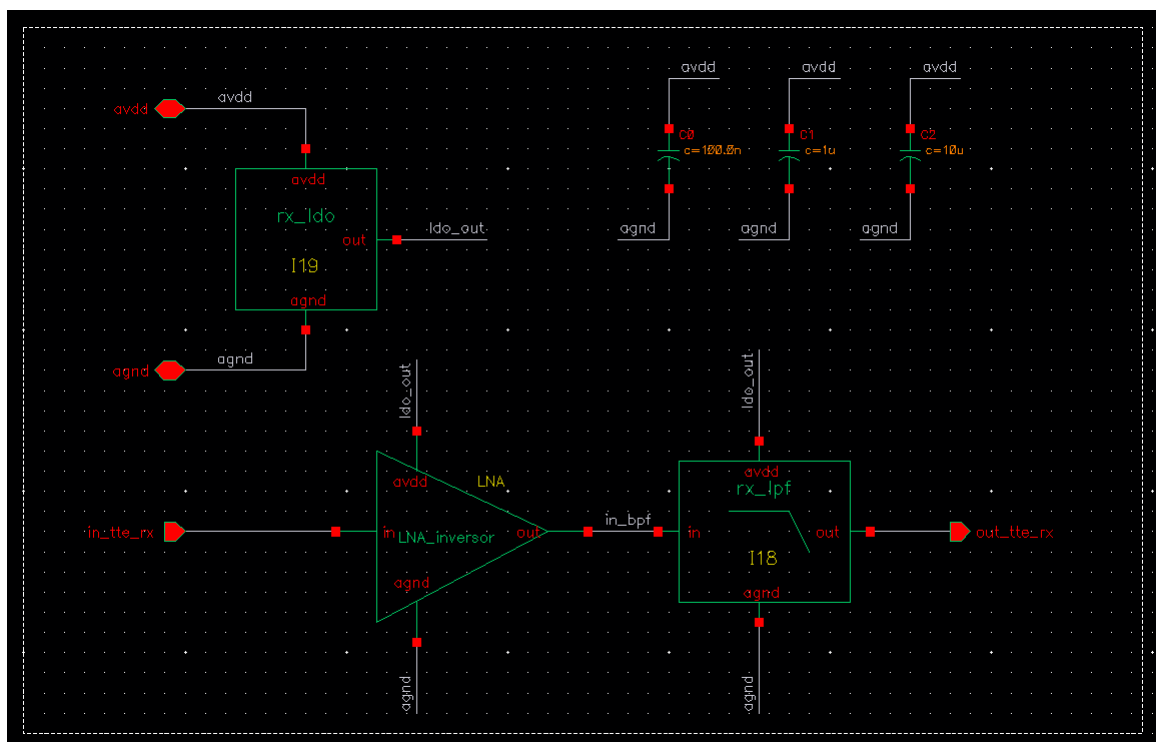


Figura 51 – Esquemático do receptor com LNA inversor e filtro passa baixa de 2º ordem.

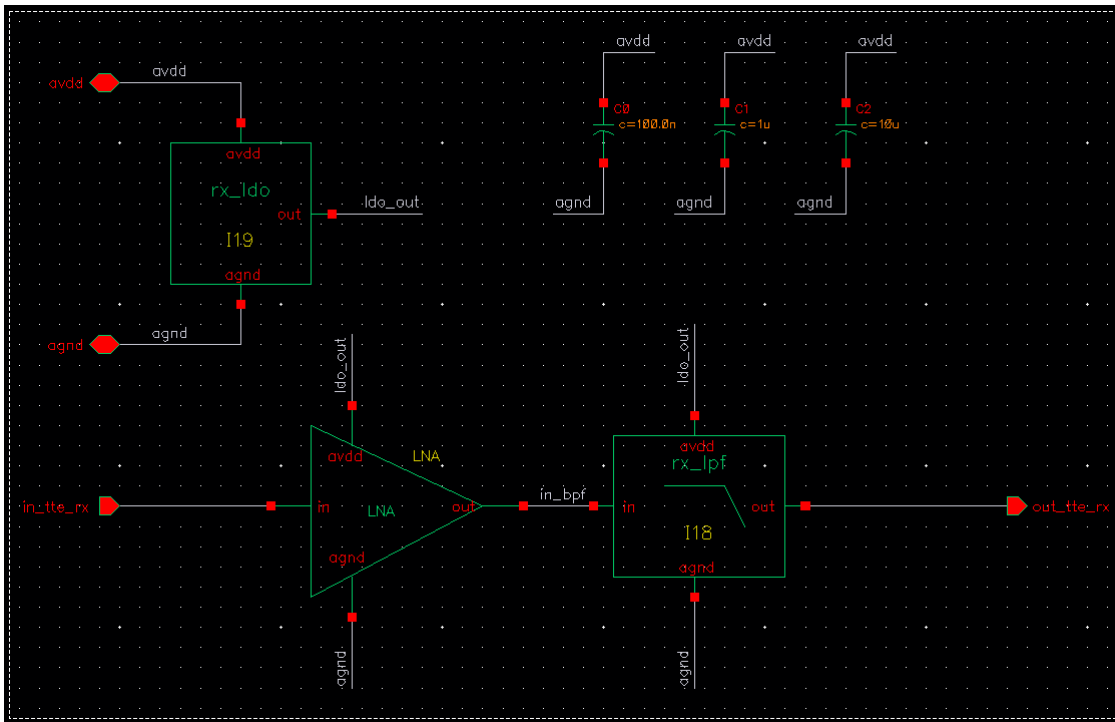


Figura 52 – Esquemático do receptor com LNA não inversor e filtro passa baixa de 2^o ordem.

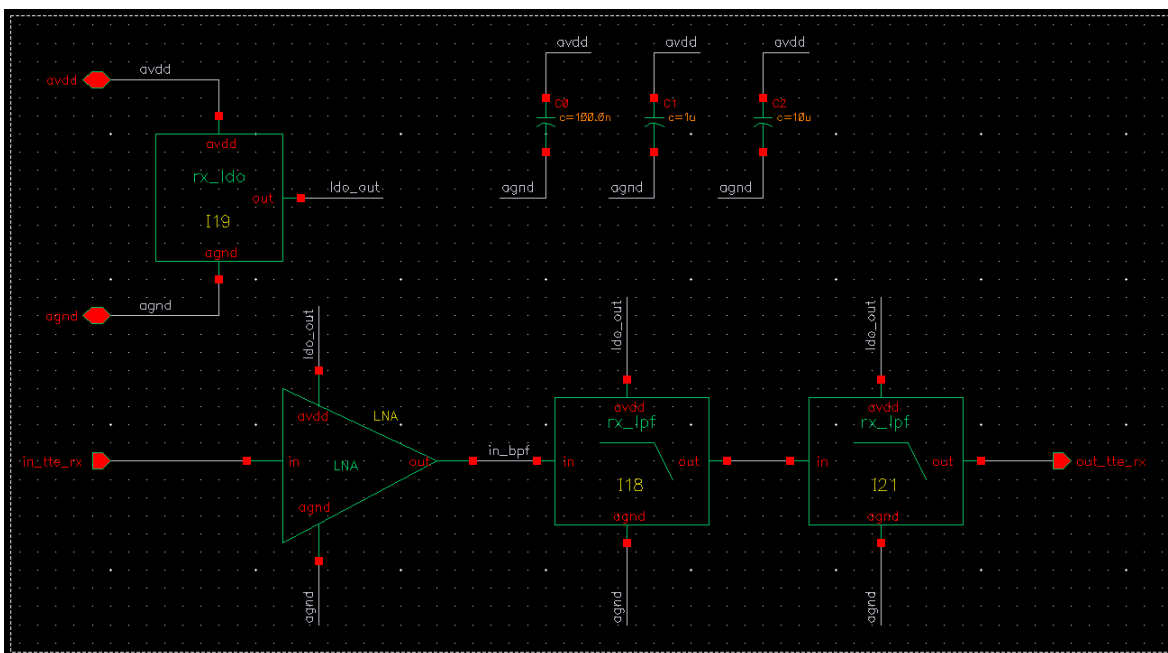


Figura 53 – Esquemático do receptor com LNA não inversor e filtro passa baixa de 4^o ordem.

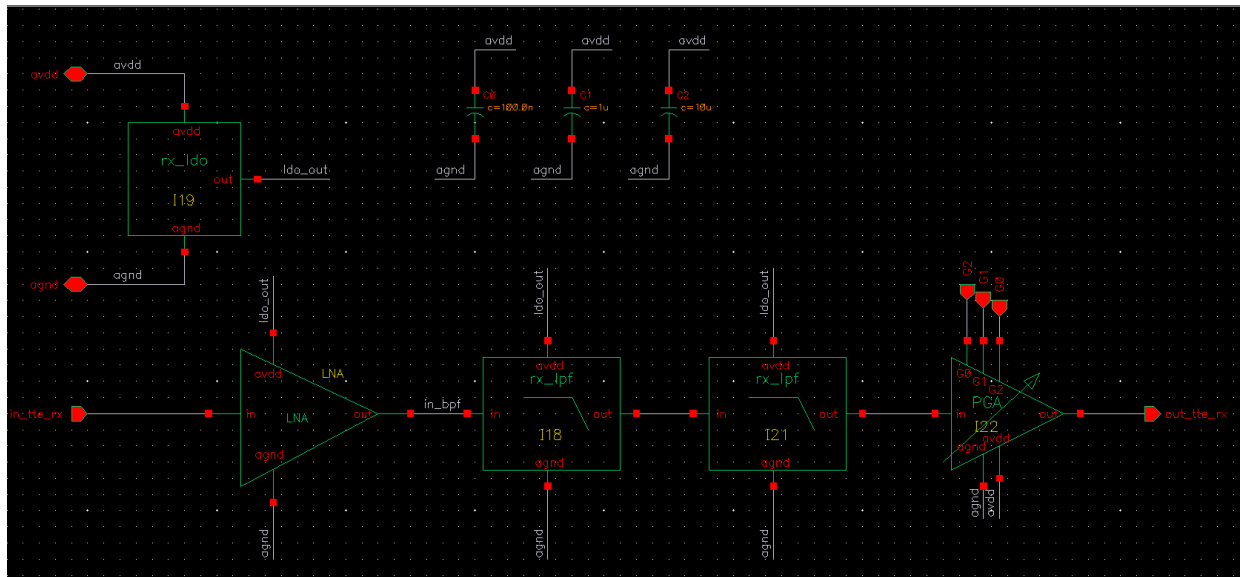


Figura 54 – Esquemático do receptor com LNA não inversor, filtro passa baixa de 4^o ordem e PGA.

4 Resultados

Neste capítulo será feito a análise individual dos resultados obtidos em simulações através do software *Cadence* e *LTSPice*, além da análise da caracterização prática em bancada.

É um importante destacar que o sinal de entrada do receptor pode ter uma amplitude de entrada dentro de uma faixa muito dinâmica, de forma que com a finalidade de caracterizar o circuito levou-se em consideração que o sinal terá um amplitude de entrada entre 1mV pico a pico até 10mV pico a pico.

4.1 Simulações da modelagem utilizando Verilog-A

Os resultados obtidos nas simulações feitas através do software *Cadence* foram feitas através de simulações mistas, nas quais parte é composta por modelos em Verilog-AMS e outra parte por modelos reais de componentes.

Em todos os blocos foi desenvolvido um testbench com os modelos criados, de modo que em sua maioria é feito uma simulação da resposta em frequência de 1Hz até 100kHz com 20 pontos por década e simulações transientes de aproximadamente 5ms.

4.1.1 LNA

No projeto deste LNA utilizou-se uma topologia não inversora, isso se deve por conta da alta impedância de entrada presente na entrada não inversora, desta forma a banda de frequência em que o circuito opera é muito mais estreita, auxiliando na rejeição de interferências e ruídos indesejados.

Desta forma, com o intuito de se comparar as topologias inversora e não inversora e o motivo do escolha não inversora foram realizados simulações da resposta em frequências de ambos os circuitos, de modo que os gráficos estão representados nas figuras 57 e 58 respectivamente. Mais a frente iremos mostra a resposta em frequência de todo o circuito de recepção ficando mais evidente a escolha do LNA não inversor.

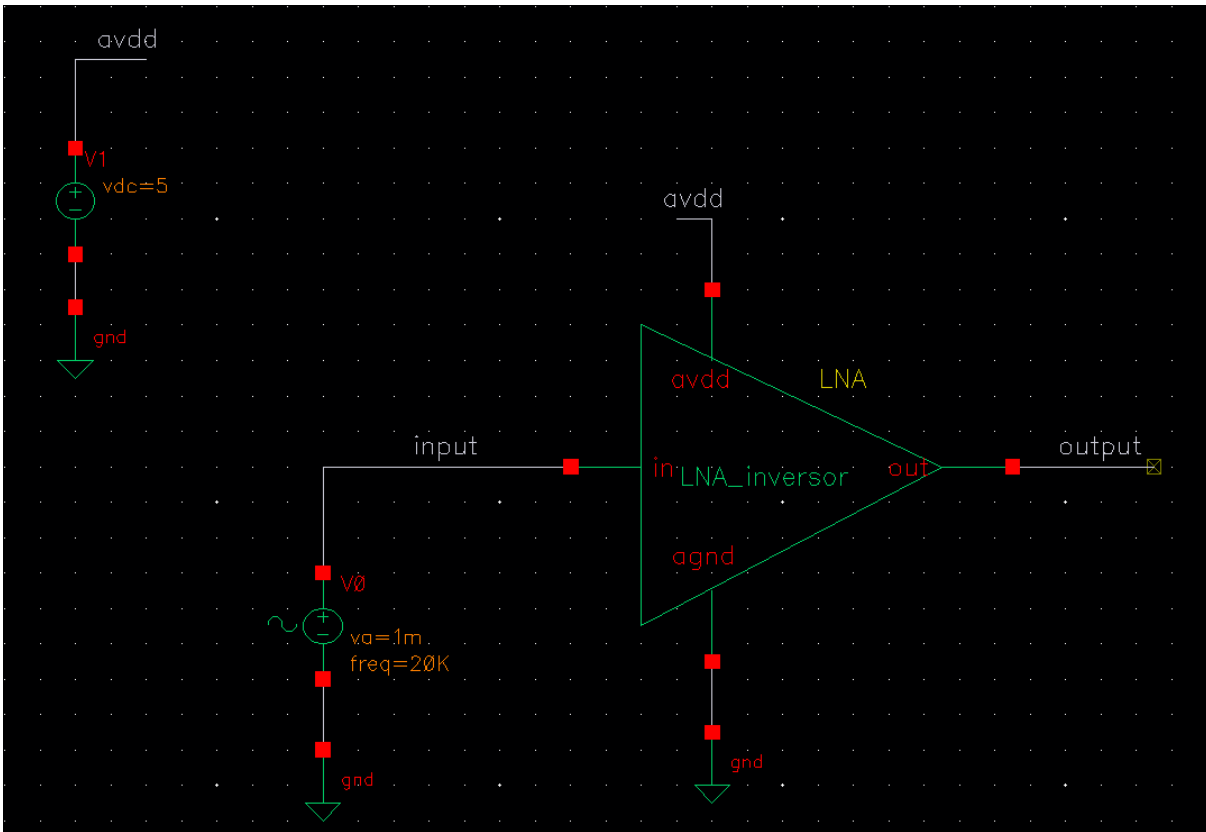


Figura 55 – Testbench do LNA inversor.

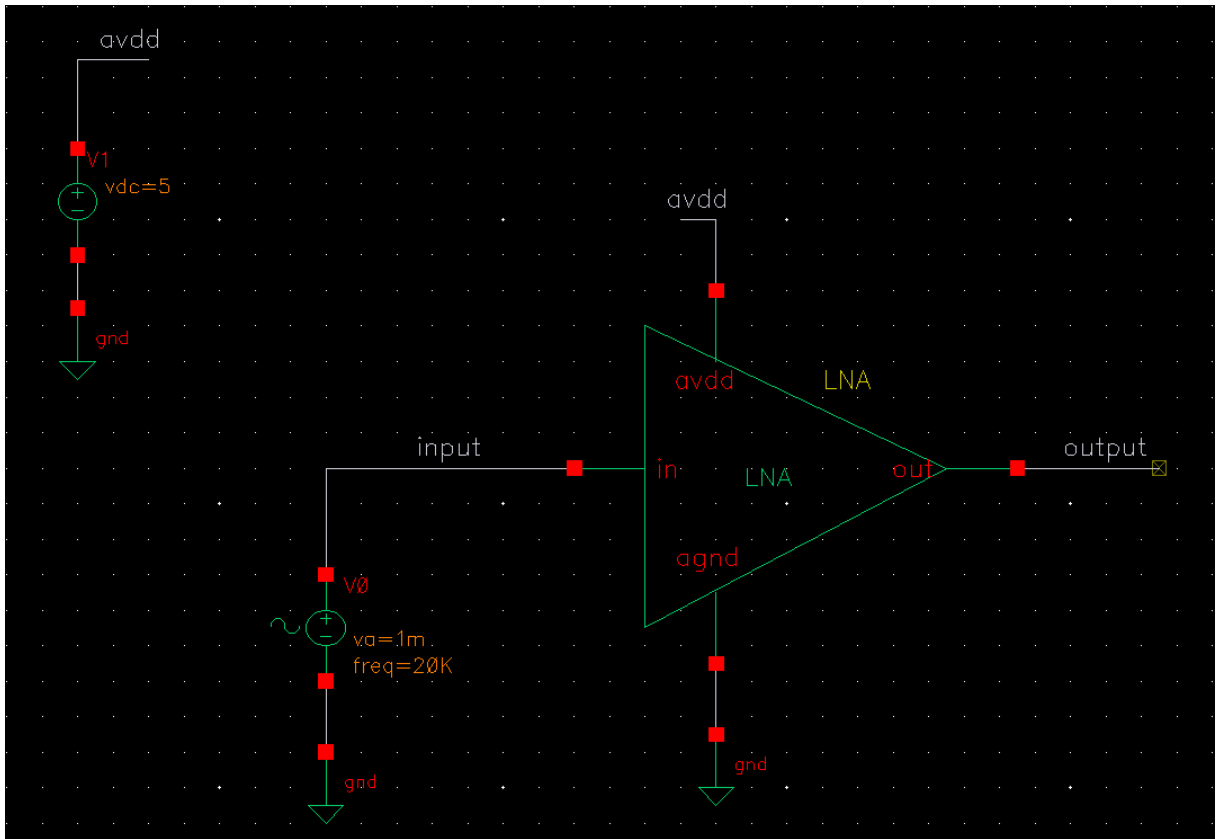


Figura 56 – Testbench do LNA não inversor.

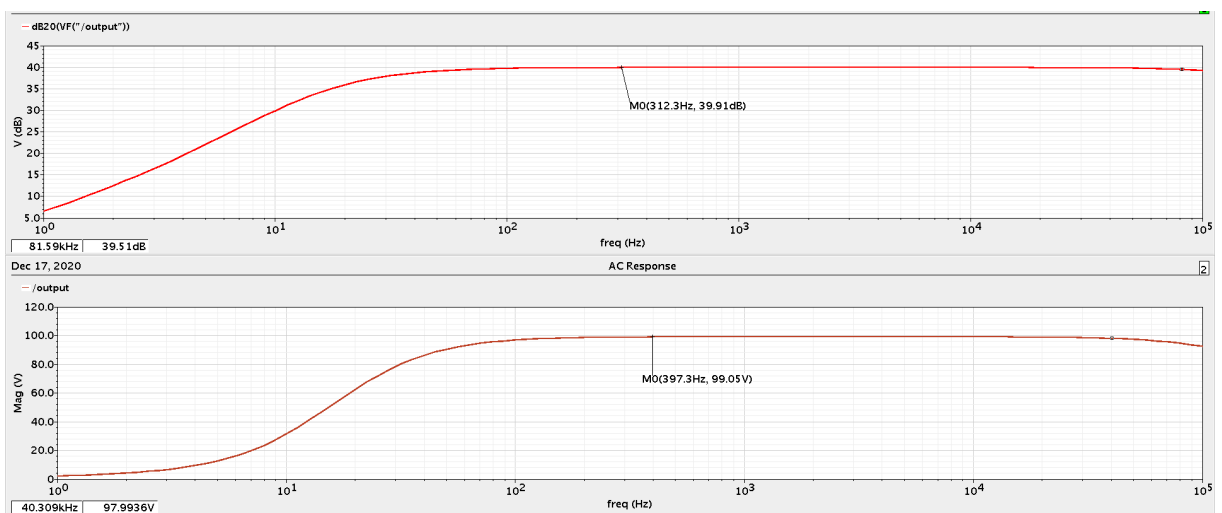


Figura 57 – Resposta em frequência da saída em dB20 do LNA inversor.

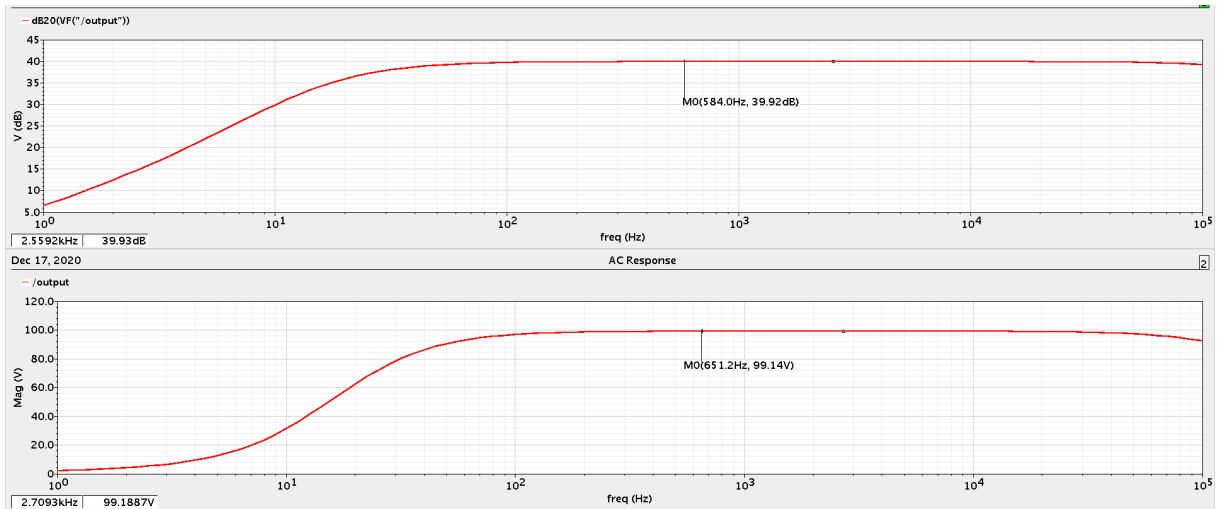


Figura 58 – Resposta em frequência da saída em dB20 do LNA não inversor.

4.1.2 Filtro passa baixa

O filtro foi modelado em Verilog-AMS e além disso foi desenvolvido seu esquemático apresentado na seção anterior. Utilizou-se um filtro passa baixa apenas com o intuito de validar o circuito de recepção. Para a análise do filtro foi realizado uma simulação em frequência, de modo que através da ferramenta "config" do *Cadence* foi possível realizar esta simulação utilizando a estrutura do esquemático, como visto na figura 60, e o modelo em Verilog-AMS, como visto na figura 61. Com isso, é possível observar que ambos apresentaram um comportamento similar, e para a realização dos testes com o todo o receptor utilizou-se a estrutura do esquemático.

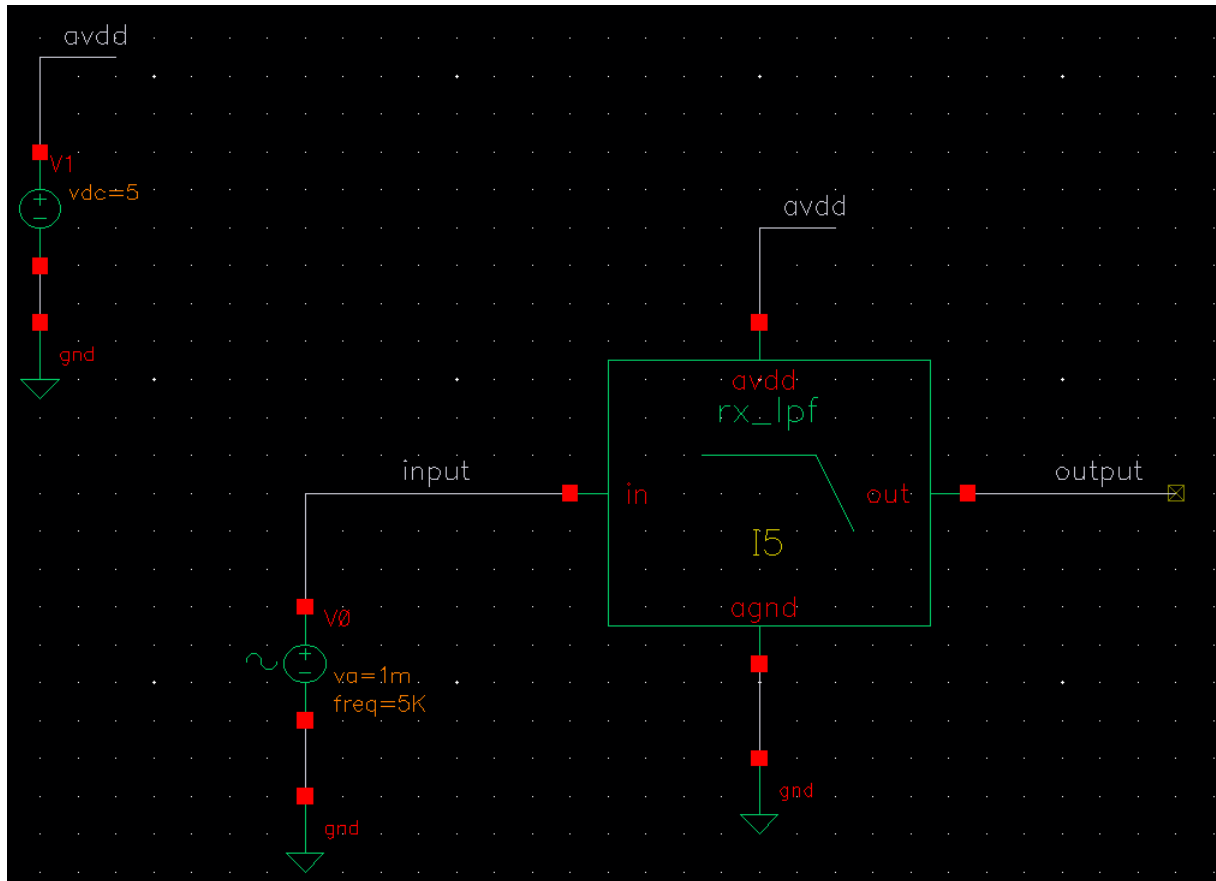


Figura 59 – Testbench do filtro passa alta de 2º ordem.

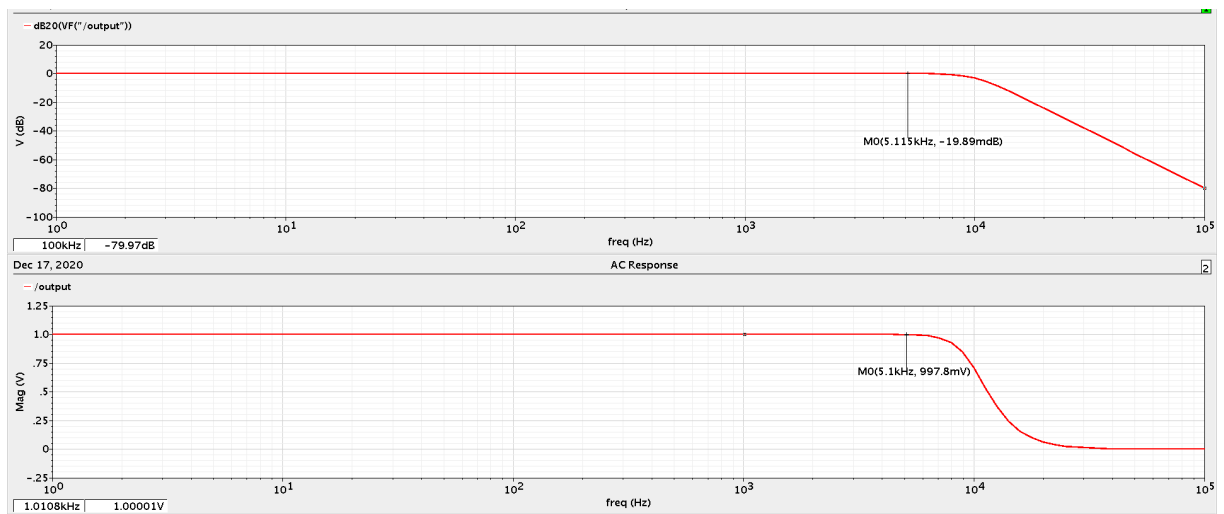


Figura 60 – Resposta em frequência da saída em dB20 do filtro utilizando o esquemático.

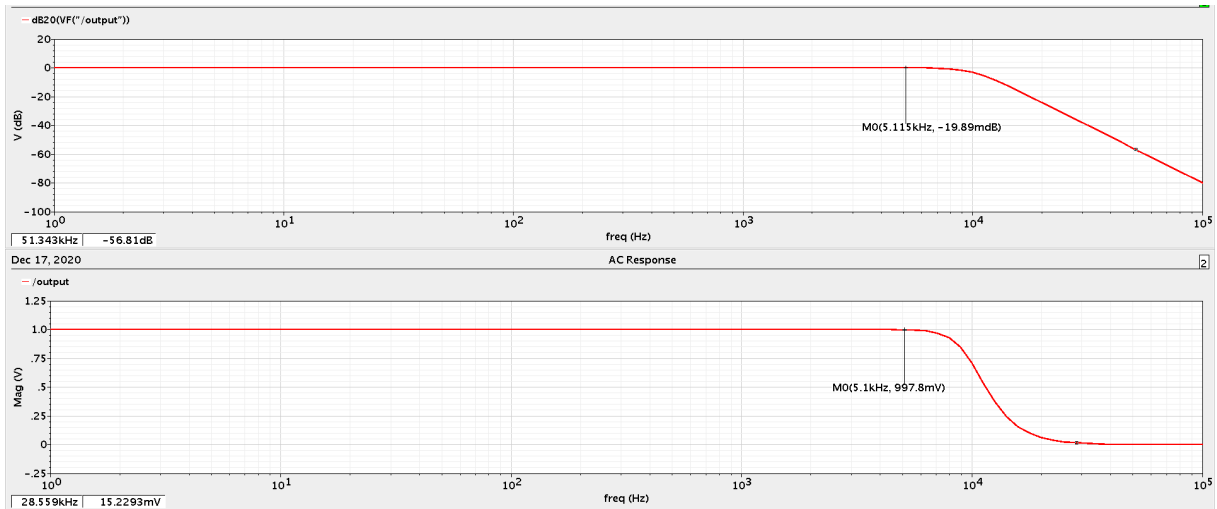


Figura 61 – Resposta em frequência da saída em dB20 do filtro utilizando o modelo Verilog-AMS.

4.1.3 PGA

No testbench do PGA utilizou-se uma entrada digital igual a 010, pois assim é possível realizar a comparação com os testes realizados em bancada, e uma tensão de entrada de 1V pico a pico com uma frequência de 5040Hz. Realizou-se uma simulação transiente, como apresentado na figura 63, e obteve-se o resultado esperado que seria de um sinal de saída com 2V pico a pico, ou seja, uma amplificação de 2V/V.

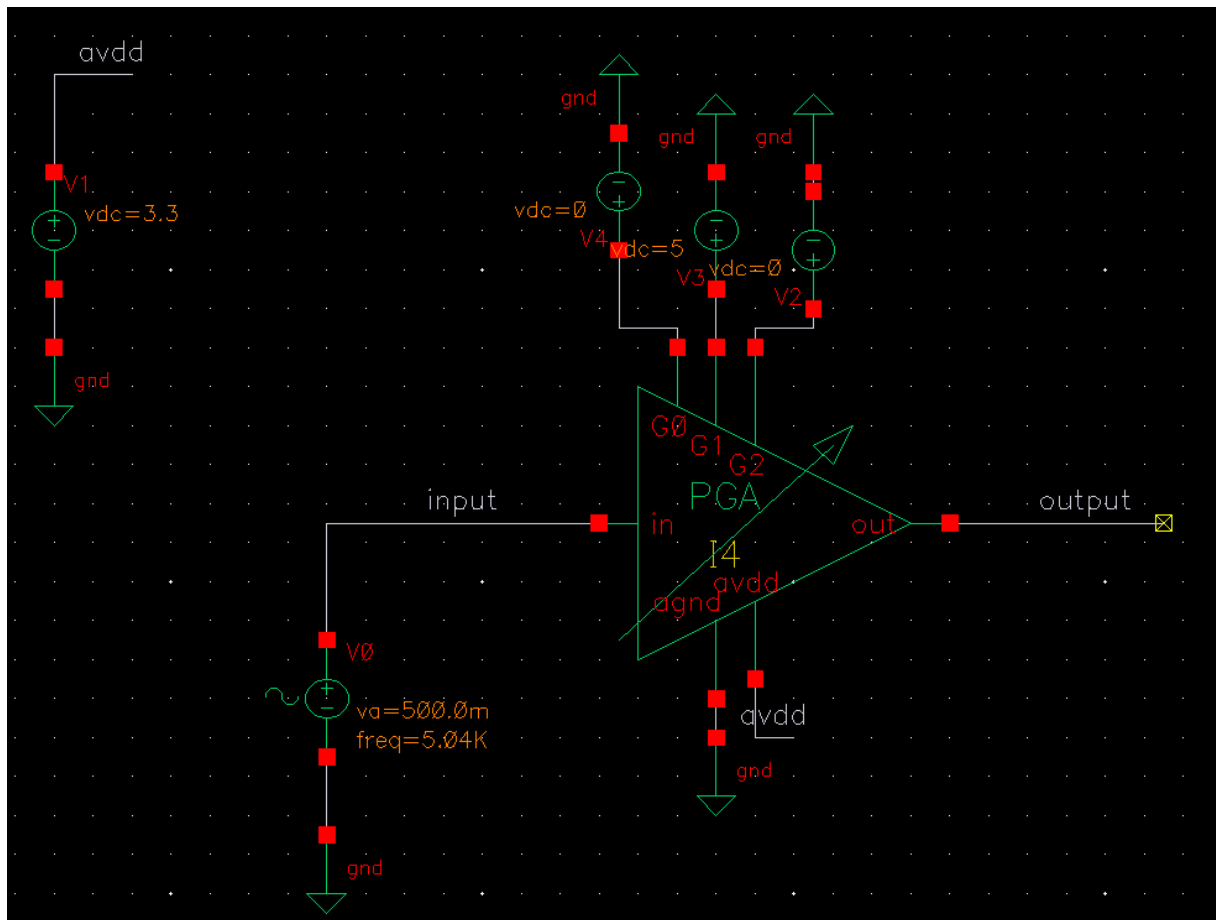
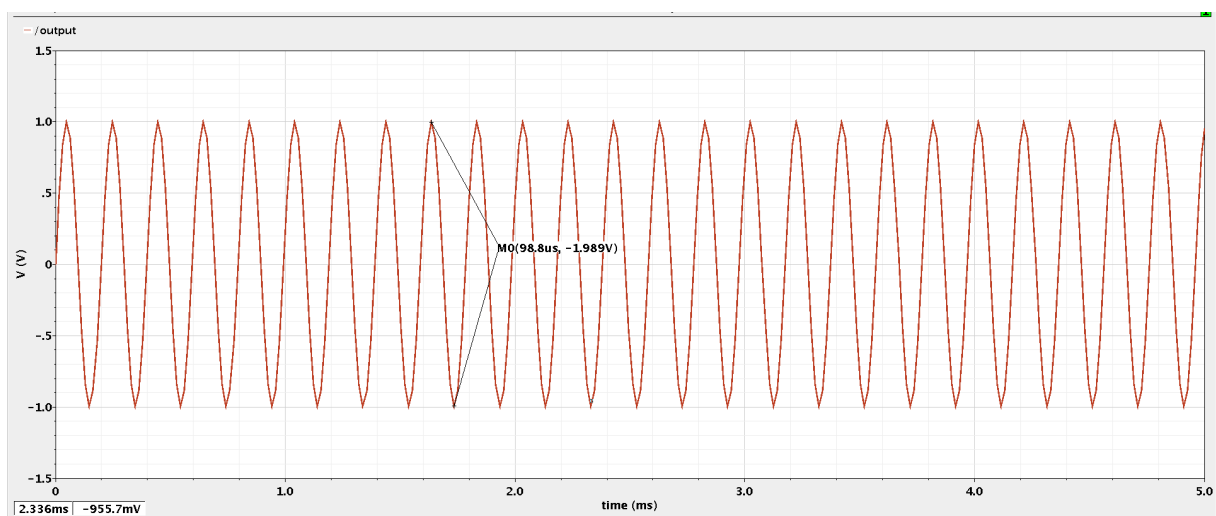


Figura 62 – Testbench do PGA.

Figura 63 – Resposta transiente da saída do PGA para uma entrada digital $G=010$ e amplitude de entrada de 1V pico a pico.

4.1.4 Receptor TTE

Na validação do receptor foram criados 4 circuitos diferentes com o objetivo de ser analisar os circuitos com topologia do LNA inversor e não inversor, filtro passa alta de segunda ordem com o de quarta ordem, além da simulação do circuito de recepção final projetado, composto por um LNA não inversor, um filtro passa alta de quarta ordem e um PGA para ajustar o ganho.

Comparando a resposta em frequência dos três esquemáticos citados inicialmente, é possível observar que o circuito com filtro de quarta ordem possui uma atenuação maior que os outros dois circuitos nas frequências indesejadas, os circuito com filtro de segunda ordem apresentam um decaimento de 10dB/dec e o circuito de quarta ordem apresenta um decaimento de 20dB/dec, como apresentado na figura 66.

Já em relação a análise da magnitude destes circuitos em frequência é possível observar que os circuitos que possuem o LNA não inversor possui uma resposta muito mais estreita, auxiliando na atenuação de frequências abaixo de acima de 5040Hz, como é possível observar na figura 67.

Por fim, é feito uma simulação AC do circuito final de recepção, figura 68, resultando em um ganho de 46dB composto por um ganho de 40dB do LNA e 6dB do PGA. Além disso, o filtro de quarta ordem em conjunto com a topologia não inversora resultam em uma atenuação próxima a frequência de 5040Hz.

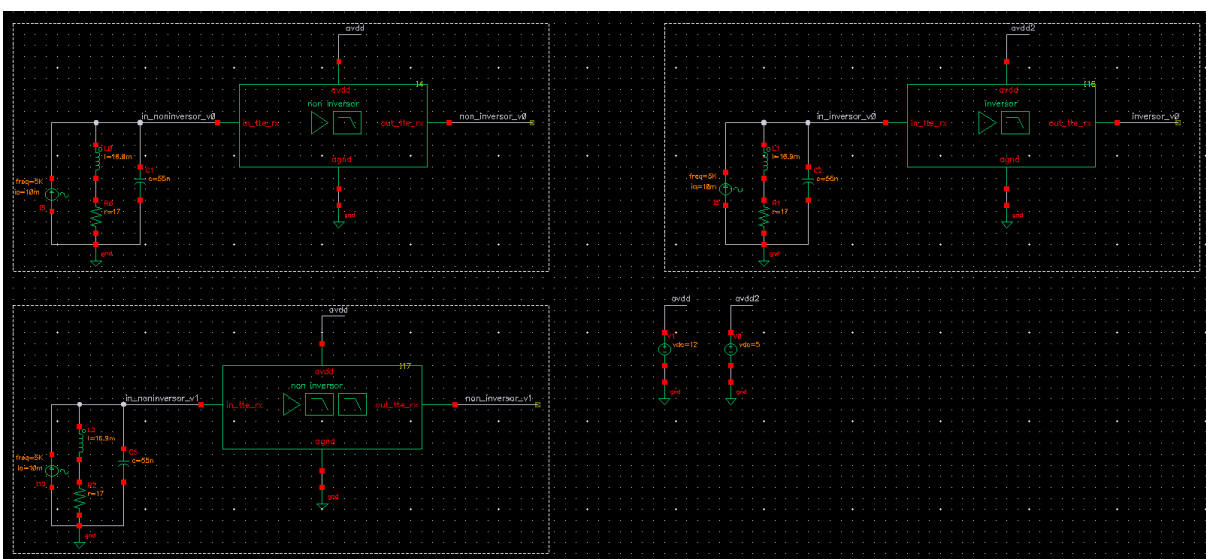


Figura 64 – Testbench dos circuitos de recepção utilizado LNA inversor e LPF de 2º ordem, LNA não inversor e LPF de 2º ordem, LNA não inversor e LPF de 4º ordem .

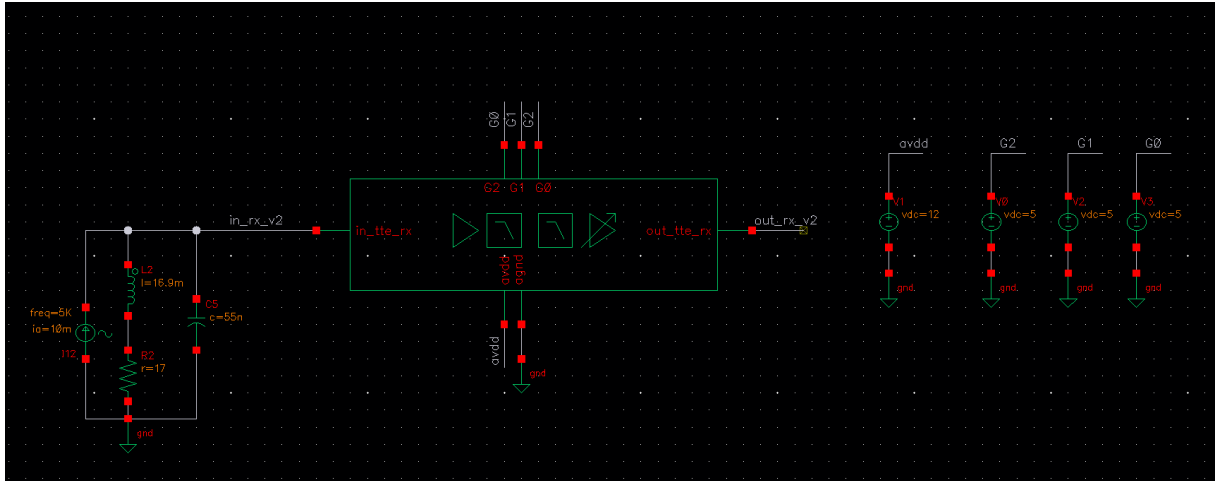


Figura 65 – Testbench dos circuitos de recepção utilizado LNA não inversor, LPF de 4º ordem e PGA .

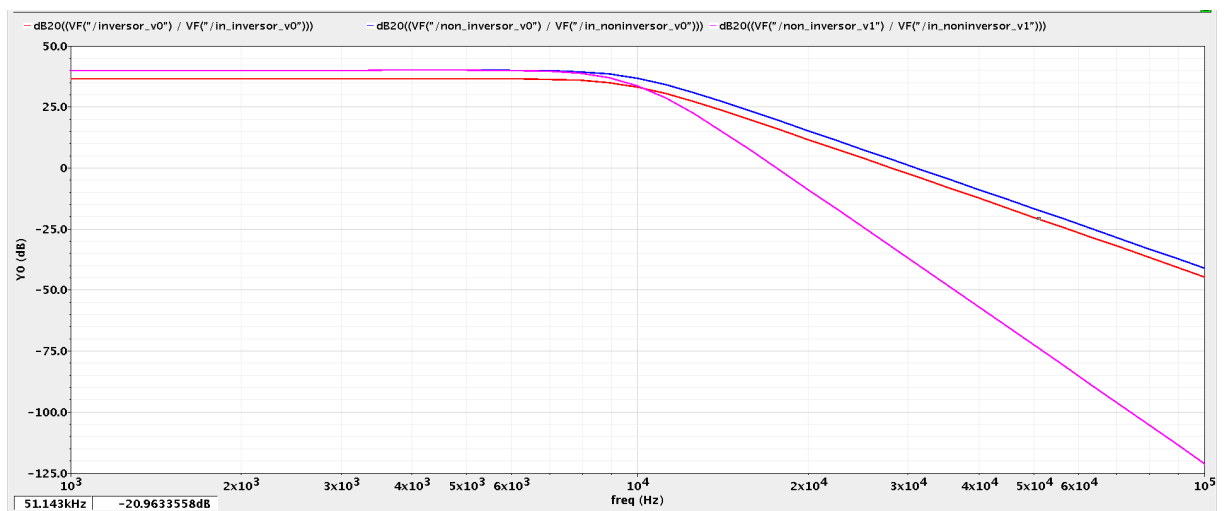


Figura 66 – Resposta em frequência do ganho em dB20 dos circuitos de recepção utilizado LNA inversor e LPF de 2º ordem, LNA não inversor e LPF de 2º ordem, LNA não inversor e LPF de 4º ordem.

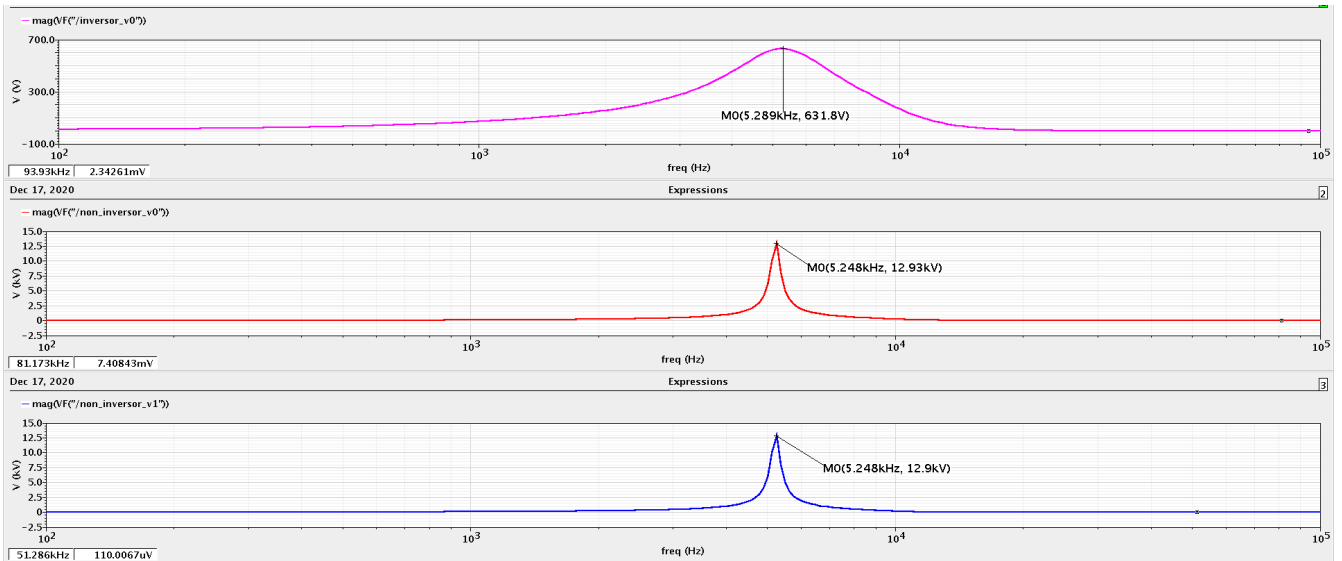


Figura 67 – Resposta em frequência da magnitude do ganho dos circuitos de recepção utilizado LNA inversor e LPF de 2º ordem, LNA não inversor e LPF de 2º ordem, LNA não inversor e LPF de 4º ordem.

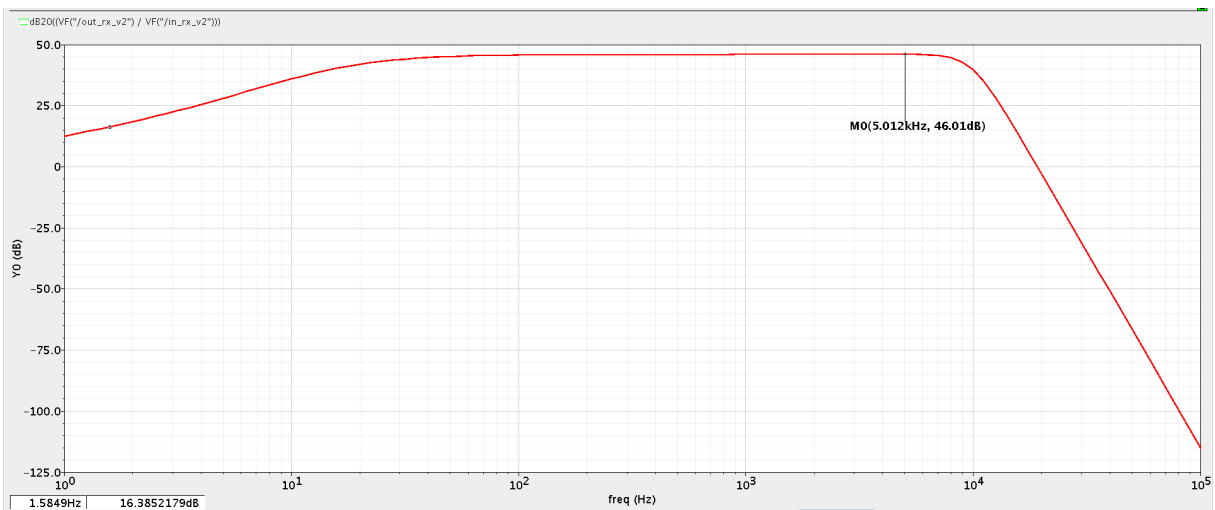


Figura 68 – Resposta em frequência do ganho em dB20 do circuito de recepção utilizando LNA não inversor, LPF de 4º ordem e PGA.

4.2 Simulações dos blocos utilizando modelos *Spice*

Com o intuito de se validar os CIs e suas respectivas configurações de circuitos, realizou-se simulações utilizando seus modelos *LTSpice*, dessa forma, gerou-se uma maior confiabilidade ao ser realizado a prototipação dos blocos individuais.

4.2.1 LNA

Para simular o LNA utilizou-se o software *Cadence*, de modo que foi necessário importar o modelo spice do OP213. O Ci apresentou um funcionamento correto e obteve o ganho de acordo com o calculado que seria de 10.5V/V(figura 71). Além disso, foi realizado uma simulação da densidade de ruído da saída do LNA com topologia não inversora(figura 72), obtendo uma baixa densidade para frequência desejada e próxima ao esperado que seria de $4.7n/\sqrt{Hz}$.

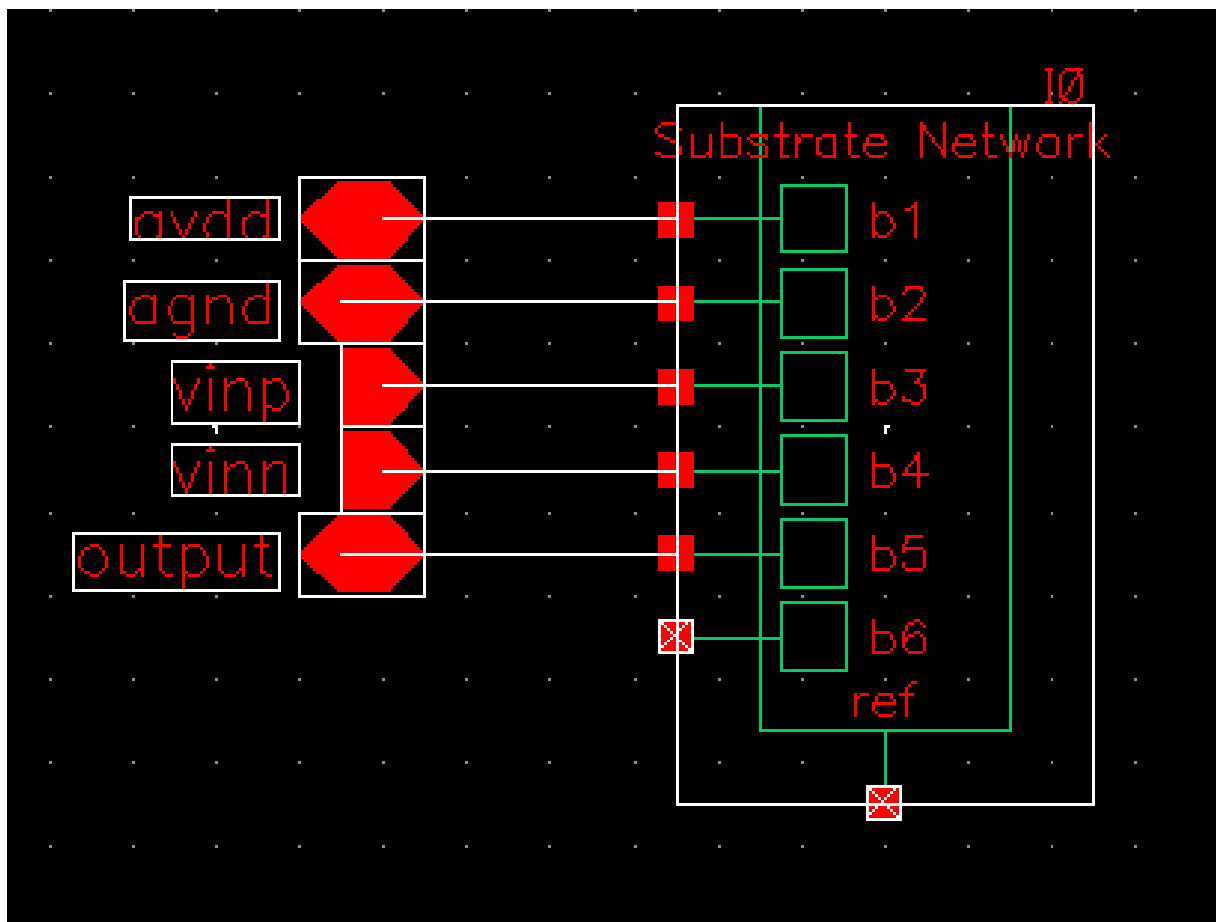


Figura 69 – Modelo spice do do OP213.

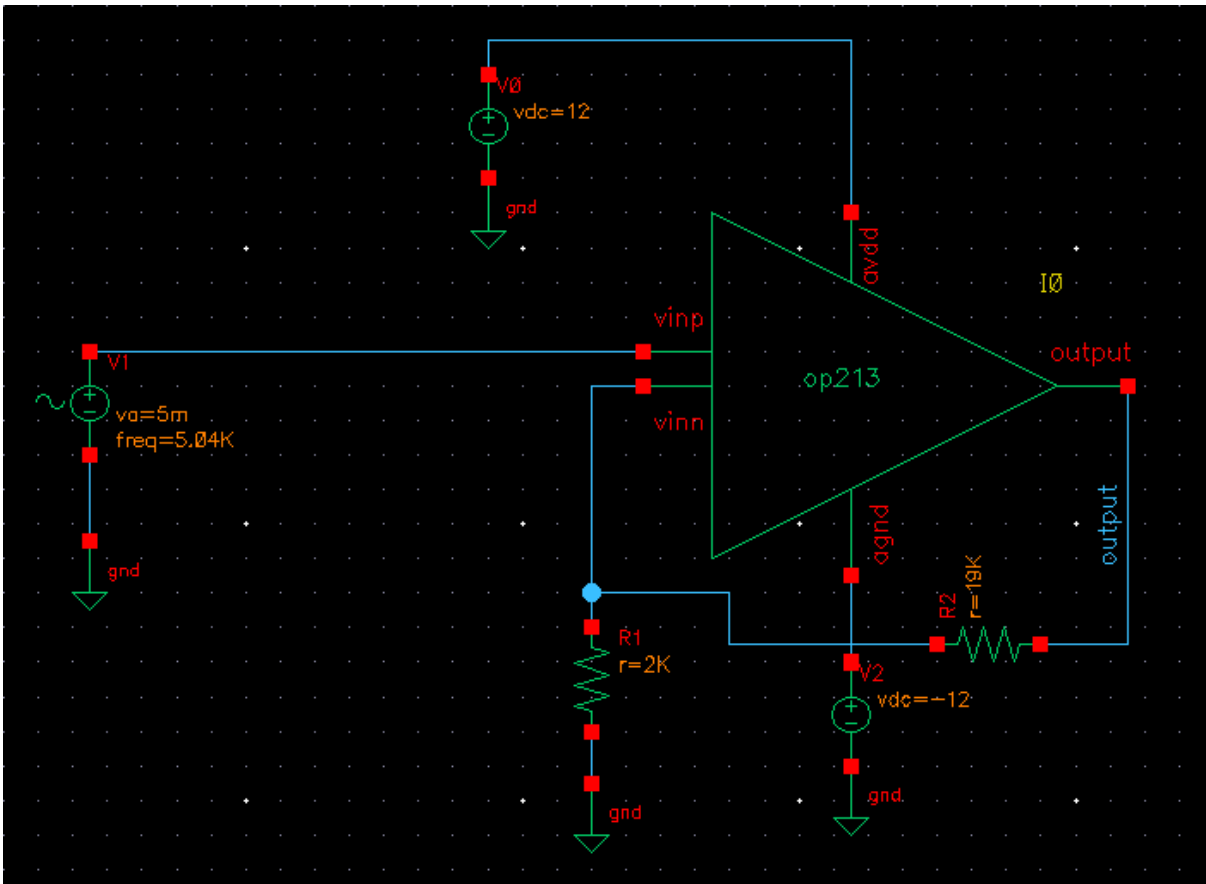


Figura 70 – Testbench do LNA utilizando o modelo spice do OP213.

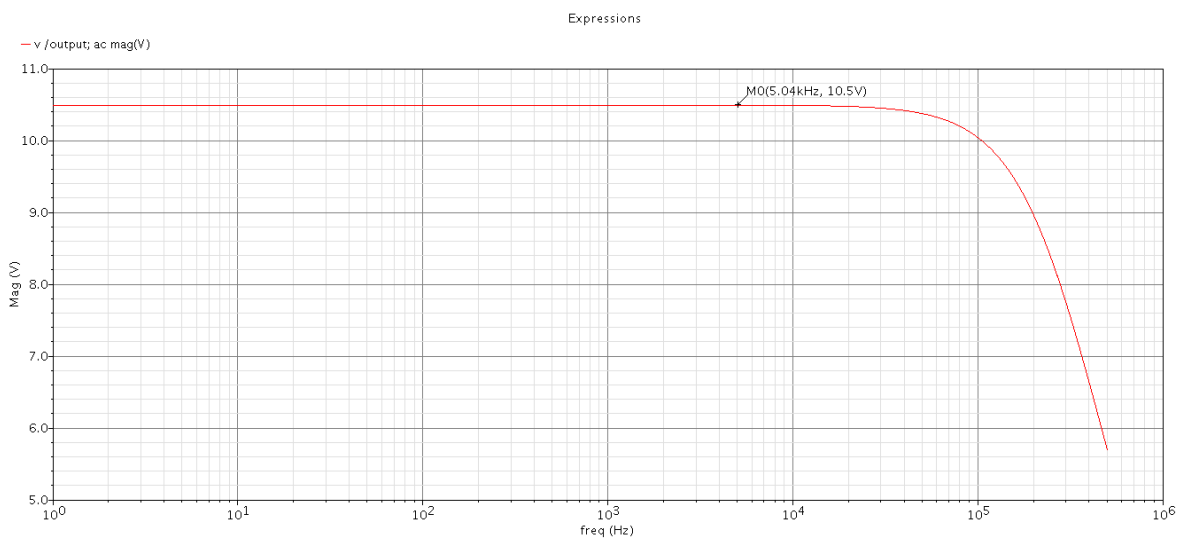


Figura 71 – Ganho em dB20 da resposta em frequência do LNA utilizando modelo spice do OP213.

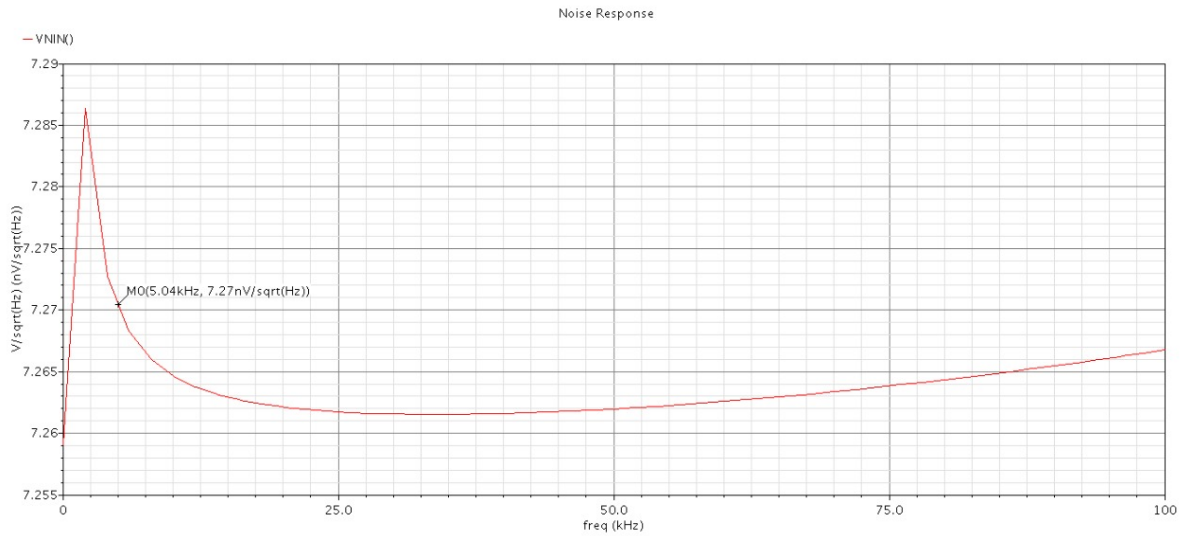


Figura 72 – Densidade de ruído da saída do LNA com topologia não inversora.

4.2.2 Filtro passa faixa

A simulação do modelo spice do CI LTC1568 foi feito através do *LTSpice*, de modo que obteve-se um resultado esperado, sendo possível analisar através da resposta em frequência (figura 74) e uma frequência centrada em 5040 Hz, assim como projetado.

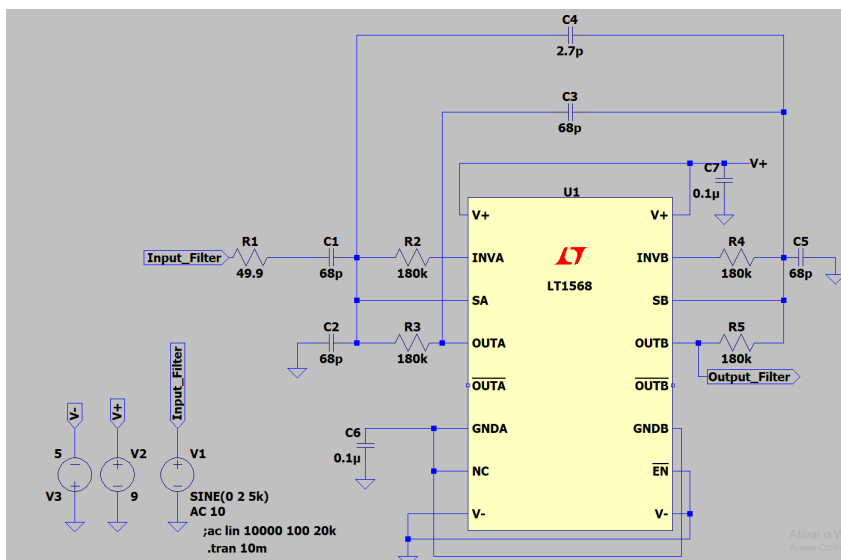


Figura 73 – Testbench do filtro utilizando o modelo spice do LTC1568.



Figura 74 – Resposta em frequência da saída do filtro utilizando o modelo spice do LTC1568.

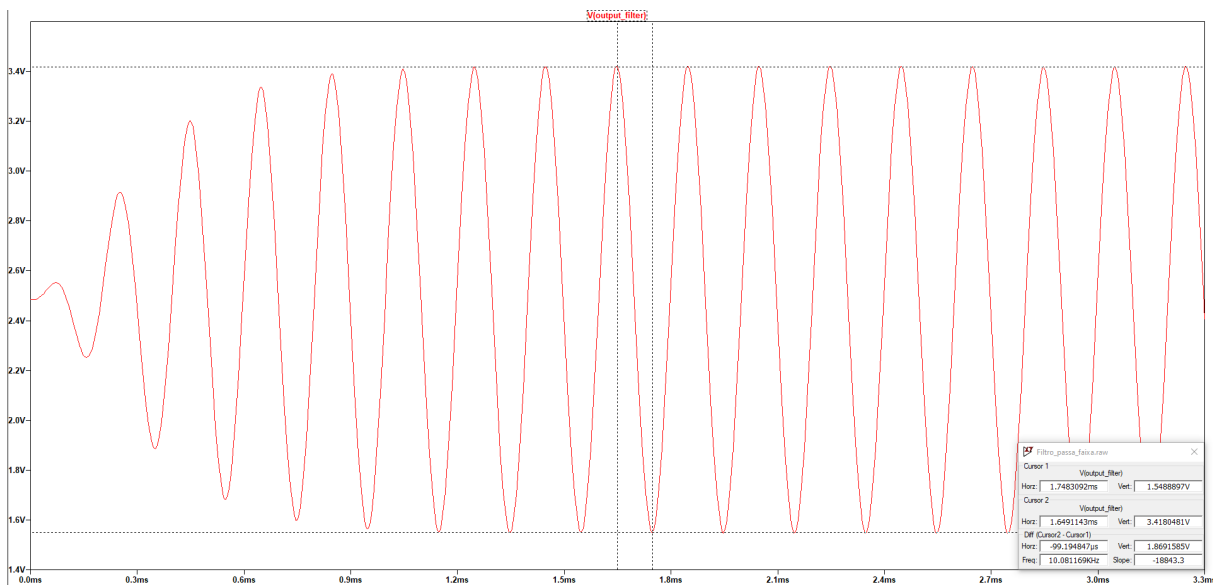


Figura 75 – Resposta transiente da saída do filtro utilizando o modelo spice do LTC1568.

4.2.3 PGA

Inicialmente foi realizado a simulação com alimentação single supply de 3.3V do circuito, de maneira que o sinal de saída sofreu uma distorção (figura 77). Por consequência projetou-se o circuito com alimentação simétrica de 3.3V e obteve-se o resultado esperado (figura 78) e sem nenhuma distorção.

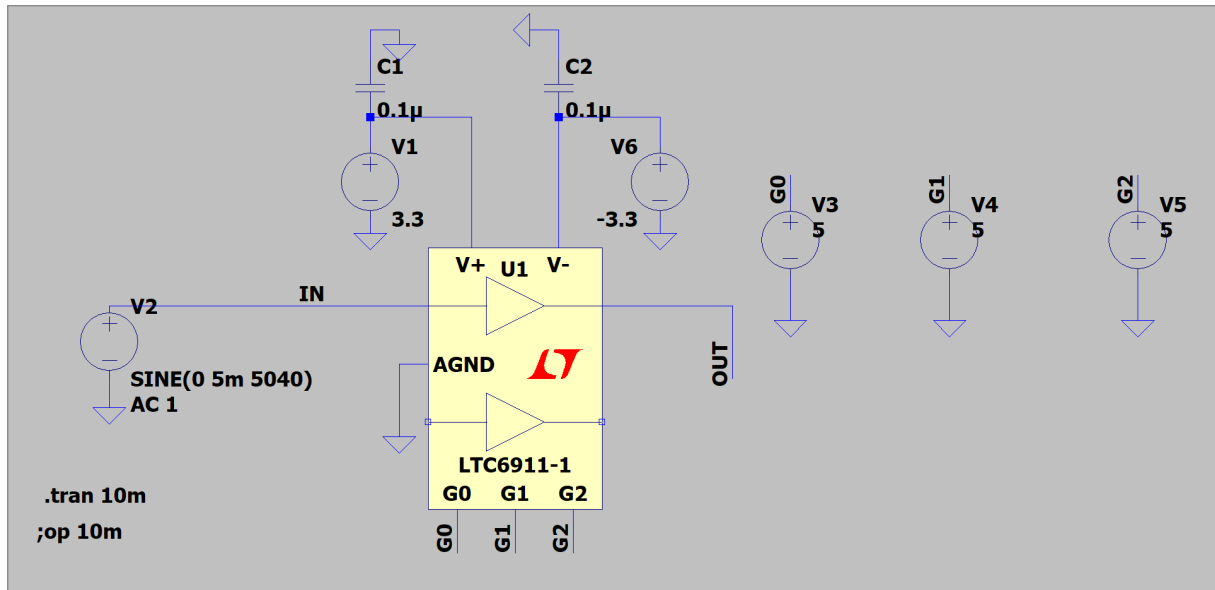


Figura 76 – Testbench do PGA utilizando o modelo spice do LTC6911-1.

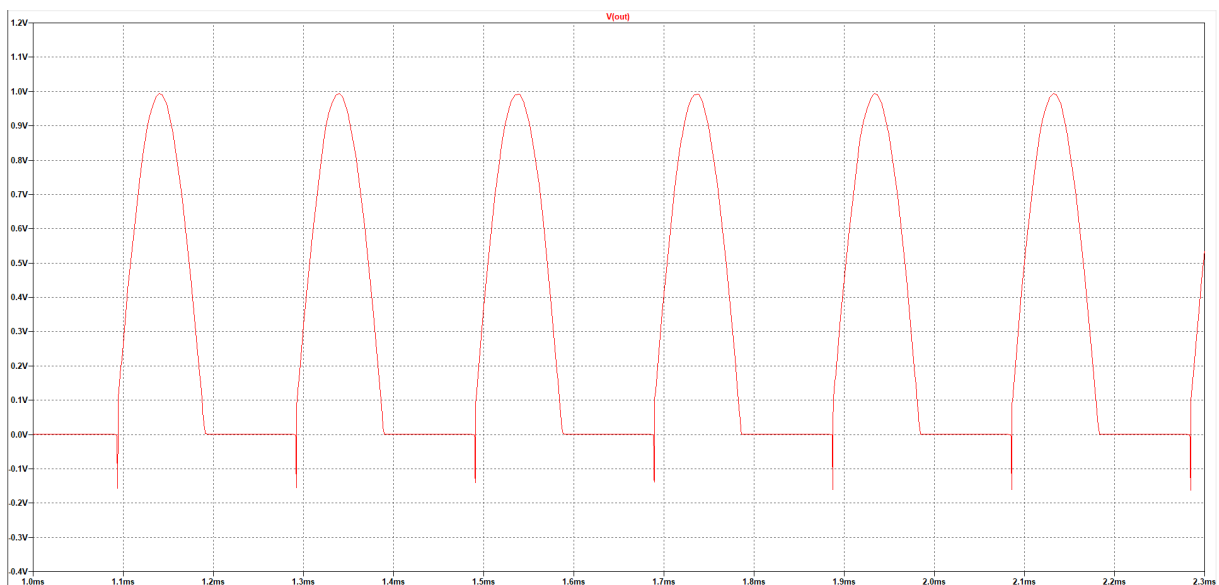


Figura 77 – Resposta transiente da saída do PGA utilizando o modelo spice do LTC6911-1 com alimentação single supply de 3.3V.

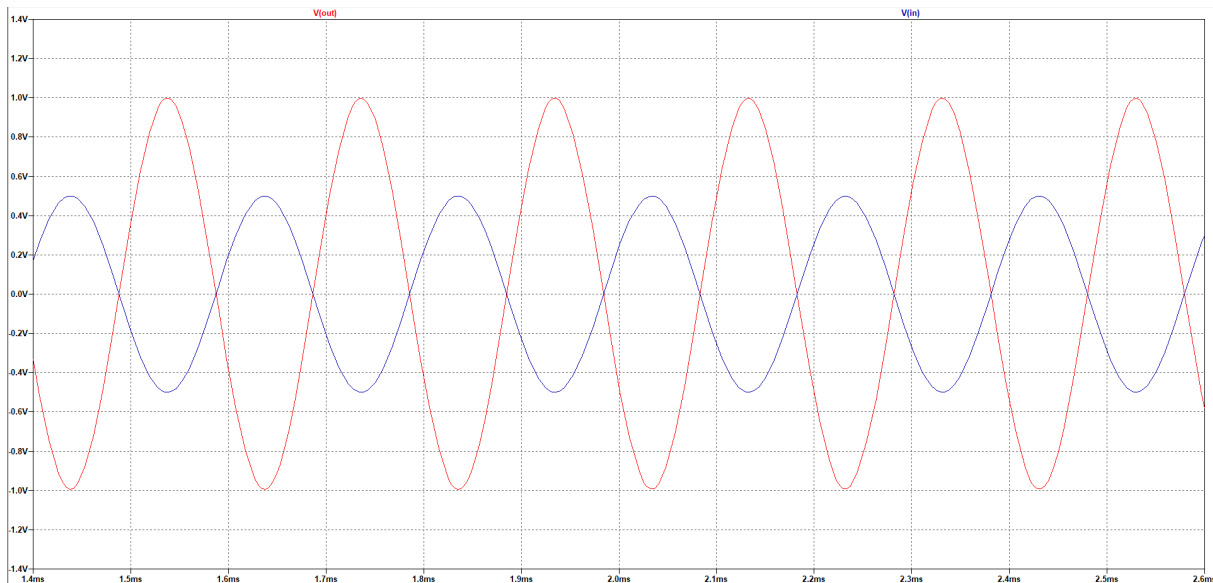


Figura 78 – Resposta transitória da saída do PGA utilizando o modelo spice do LTC6911-1 com alimentação simétrica de 3.3V.

4.3 Caracterização em bancada dos blocos

Para realizar a caracterização dos blocos individuais foi necessário realizar testes no laboratório NEI da Faculdade do Gama.

4.3.1 LNA

A caracterização do LNA foi feita através da submissão de diferentes ganhos, para isso foi necessário alterar os ganhos através das chaves S1 e S2 que realizavam a mudança da resistência de realimentação do circuito.

O sinal de entrada utilizado em todos os testes foi de 10mV pico a pico com uma frequência de 5.04kHz, figura 79.

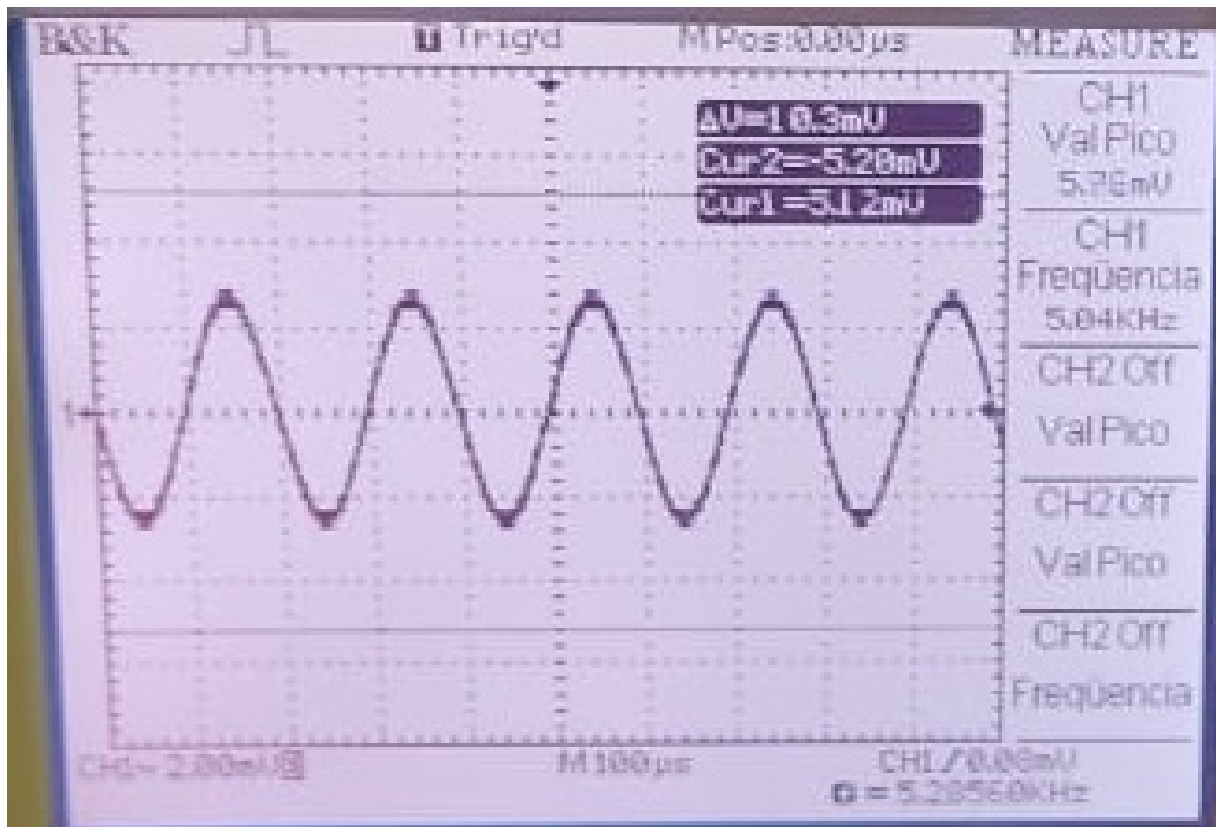


Figura 79 – Sinal de entrada do LNA.

Foram realizadas 16 combinações diferentes de ganho através dos valores de resistência disponibilizados, obtendo assim uma variada faixa de ganhos para se utilizar no circuito de recepção, os valores de resistência utilizados e o seus respectivos ganhos estão apresentados na tabela 4.3.1.

Chaves		Saída (Vp-p)	Ganho (V/V)	Ganho (dB)
S1	S2			
R4 = 100k Ω	R11 = 1k Ω	1.88	188	45.48
R4 = 100k Ω	R10 = 2.2k Ω	2.98	298	49.48
R4 = 100k Ω	R8 = 10k Ω	10	1000	60
R4 = 100k Ω	R6 = 30k Ω	Saturou	Saturou	Saturou
R5 = 30k Ω	R11 = 1k Ω	640m	64	36.12
R5 = 30k Ω	R10 = 2.2k Ω	952m	95.2	39.57
R5 = 30k Ω	R8 = 10k Ω	3.22	322	50.15
R5 = 30k Ω	R6 = 30k Ω	8.80	880	58.88
R7 = 10k Ω	R11 = 1k Ω	248m	24.8	27.88
R7 = 10k Ω	R10 = 2.2k Ω	344m	34.4	30.73
R7 = 10k Ω	R8 = 10k Ω	1.2	120	41.58
R7 = 10k Ω	R6 = 30k Ω	3.2	320	50.1
R9 = 1k Ω	R11 = 1k Ω	40m	4	12.04
R9 = 1k Ω	R10 = 2.2k Ω	60m	6	15.56
R9 = 1k Ω	R8 = 10k Ω	200m	20	26.02
R9 = 1k Ω	R6 = 30k Ω	600m	60	35.56

Tabela 8 – Tabela de resultados do LNA.

Levando em consideração que a amplitude de entrada do sinal de recepção estará entre 1mV e 10mV pico a pico, um ganho adequado para não saturar o sinal de saída do circuito de recepção seria de 95.2V/V, dado pelos resistores com de 30k Ω e 2.2k Ω . De modo que o primeiro estágio gera um ganho de 31V/V e o segundo estágio um ganho de 3.2V/V. O sinal com ganho de 39.57dB pode está representado na figura 80.

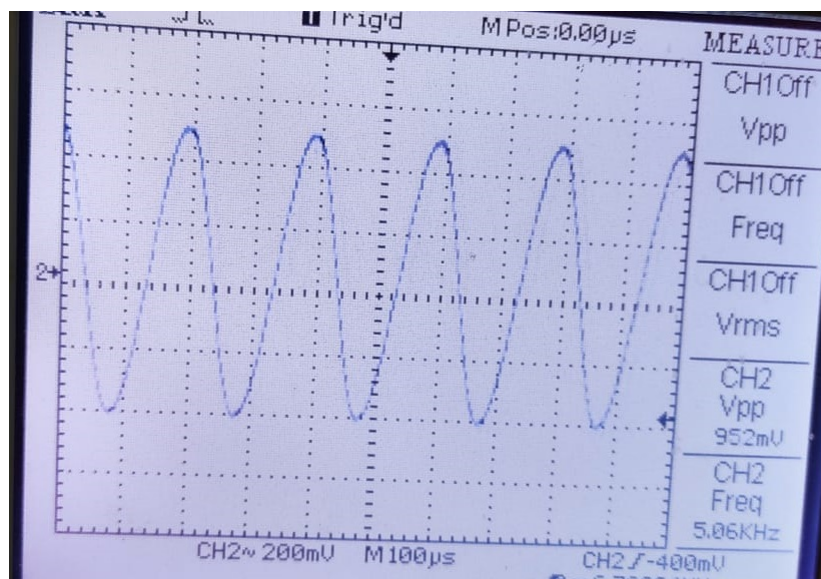


Figura 80 – Sinal de saída do LNA com ganho de 95.2V/V.

4.3.2 Filtro passa faixa

O protótipo desenvolvido para o filtro passa faixa não apresentou resultados esperados, de modo que ao se inserir uma amplitude de 1V de entrada com frequência de 5.04kHz o sinal de saída atenuava para cerca de 100mV, além de apresentar muito ruído.

Isso pode ter ocorrido por conta da má prototipação da placa de circuito impresso ou por conta dos valores dos componentes que necessitavam ser muito próximos dos valores ideais para se obter a frequência de corte desejada. Além disso, foram feitos testes do circuito em protoboard a fim de se encontrar o problema, porém o circuito se comportou da mesma forma que o protótipo.

Por fim, foi elaborado um novo layout, porém por não ter obtido nenhum resultado promissor em protoboard e nem no circuito prototipado optou-se por não realizar a segunda prototipação, já que por conta da situação atual em que vivemos não seria possível ter o auxílio do orientador para encontrar o problema real.

4.3.3 PGA

Em relação ao primeiro procedimento realizado no PGA não foi possível obter o resultado esperado, isso se deve principalmente por conta do condicionamento do sinal para ser realizado a leitura através do amplificador somador. Como não foi possível realizar muitos testes no laboratório optou-se por uma caracterização mais eficiente, gerando as entradas digitais para o PGA de maneira isolada.

Inicialmente foi realizado um teste do PGA com alimentação single supply, porém o sinal de saída (figura 81) apresentou a mesma distorção apresentada através da simulação realizada no *LTSpice*.



Figura 81 – Sinal de saída do PGA para $G=111$ e alimentação single supply.

Em seguida o PGA foi alimentado simetricamente, obtendo uma tensão de saída de acordo com o esperado. Para os testes no PGA utilizou-se diferentes tensões de entrada, de maneira que para cada tensão de entrada são gerados diferentes entradas digitais G . Nota-se na 4.3.3 que foram feitas medidas para diferentes entradas digitais, podendo concluir que o ganho obtido estava próximo do ideal.

Entrada digital (G)	Entrada (Vp-p)	Saída (Vp-p)	Ganho (V/V)	Ganho (dB)
001	3.36	3.36	1	0
001	2	2	1	0
010	1.48	2.88	1.94	5.66
010	960m	1.6	1.67	4.45
011	500m	2.26	4.52	13.1
100	200m	2.04	10.2	20.17
101	50m	2.34	46.8	33.4
101	20m	840m	42	32.46

Tabela 9 – Tabela de resultados do PGA.

É possível observar o sinal de saída de 1.6 V pico a pico para uma entrada de 960mV pico a pico, figura 82, de maneira que este ganho seria suficiente levando em consideração um sinal de entrada de 10mV, tendo um ganho de aproximadamente 100V/V no estágio do LNA, chegando ao valor próximo de 960mV. A presença da grande quantidade de ruído pode ser por conta de diversos fatores, como utilização de jumpers, protoboard, má conexão, ruído gerado pela fonte de alimentação, entre outros fatores.

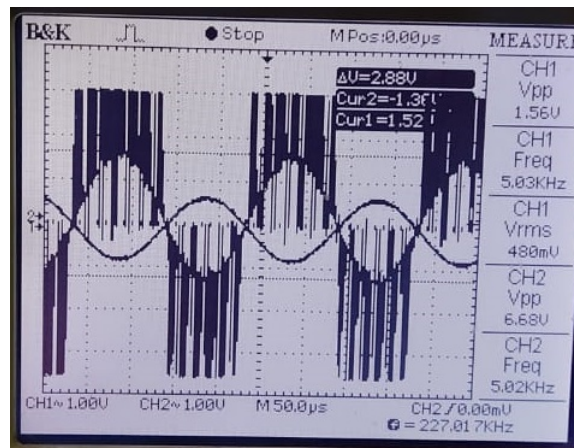


Figura 82 – Sinal de saída do PGA para G=010.

5 Discussões

5.1 Discussões

O problema deste trabalho está correlacionado com comunicações realizadas através da terra, tendo como foco realizar monitoramento de barragens e resgate de vítimas em acidentes. Por conseguinte, com base em toda a análise feita foi possível identificar mais profundamente os principais problemas nos sistemas de comunicações TTE, possibilitando desenvolver uma solução mais eficaz.

Devido as diversas restrições que o solo impõe, este tipo de comunicação acaba se tornando um grande desafio. A utilização de frequências ULF e VLF são imprescindíveis para este projeto, pois através delas é possível minimizar as atenuações causadas pelo solo. Em contraponto, ao se utilizar frequências muito baixas se obtém um comprimento de onda com um valor alto, resultando em uma largura de banda pequena e conseqüentemente limitando as alternativas de modulação digital.

A partir disso, esse trabalho tem como intenção o desenvolvimento do protótipo de um receptor para curtas distâncias, ao passo que para chegar neste resultado foram desenvolvidos estudos teóricos relacionados, simulações através de modelos, simulações dos modelos spice dos componentes e caracterizações práticas dos blocos individuais.

Os objetivos propostos para este trabalho foram parcialmente cumpridos, faltando a caracterização prática do filtro passa faixa, uma caracterização mais eficiente do PGA e testes com todo o circuito TTE. Porém, por conta da pandemia em que estamos imersos houve uma dificuldade em realizar os testes práticos, necessitando de uma burocracia muito maior para utilizar o laboratório.

Contudo, para minimizar a falta de testes práticos foram desenvolvidos simulações extras, como simulações dos modelos spice dos componentes, com o intuito de se ter uma validação extra de que o circuito é eficiente. Além disso, os resultados práticos do LNA e do PGA se mostraram bastante promissores.

Referências

- 1 SOUSA, L. et al. Desafios para a transmissão de dados em um sistema de comunicação through-the-earth (tte). Simpósio Brasileiro de Telecomunicações (SBrT). Citado 5 vezes nas páginas 9, 23, 27, 28 e 29.
- 2 CARTER, B. *Op Amp Noise Theory and Applications*. [S.l.], 2008. Citado 8 vezes nas páginas 9, 13, 30, 31, 32, 33, 34 e 35.
- 3 MORENO, R. F. L. Filtro anti-aliasing gm-c totalmente diferencial e sintonizável. Universidade Federal do Rio de Janeiro, Junho 2009. Citado 2 vezes nas páginas 9 e 37.
- 4 RAZAVI, B. *Fundamentos da Microeletrônica*. 1st edition. ed. Rio de Janeiro, Brasil: [s.n.], 2010. Citado 4 vezes nas páginas 9, 37, 38 e 39.
- 5 LINEAR TECHNOLOGY CORPORATION. *DEMO MANUAL DC675B*. [S.l.], 2004. Citado 2 vezes nas páginas 9 e 53.
- 6 MACHADO, W. G. de F. *Monitoramento de barragens de contenção de rejeitos da mineiração*. Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo, 2007. Citado na página 23.
- 7 MENDES, H. B. S. et al. Experimental platform for through-the-earth communication. Dept. of Electrical Engineering of University of Brasilia and Vale Institute of Technology, Brazil. Citado 2 vezes nas páginas 27 e 28.
- 8 MENDES, H. B. S. et al. Modelo empírico-estatístico de canal para comunicações em minas subterrâneas. XXXIV simpósio brasileiro de telecomunicações, Agosto 2016. Citado 2 vezes nas páginas 29 e 30.
- 9 MENDES, H. B. S. *Plataforma experimental para Comunicação Through-The-Earth (TTE)*. Dissertação (Mestrado) — Universidade de Brasília (UnB), 2007. Citado na página 30.
- 10 GIBSON, D. *Channel Characterisation and System Design for Sub-Surface Communications*. Dissertação (Mestrado) — School of Electronic and Electrical Engineering, February 2003. Citado na página 30.
- 11 RAZAVI, B. *RF Microelectronics*. 2nd edition. ed. United States at Hamilton Printing Company in Castleton, New York: [s.n.], 2011. Citado 2 vezes nas páginas 36 e 37.
- 12 MATIAS, M. L. *Amplificador de ganho programável aplicado em um receptor de frequência com subamostragem em dupla quadratura*. Dissertação (Mestrado) — Universidade Federal do Paraná, 2017. Citado na página 40.
- 13 MORAES, G. dos Santos de. *AMPLIFICADOR DE GANHO PROGRAMAVEL INTEGRADO PARA APLICACÃO EM MEDIDORES DE ENERGIA ELETRICA INTELIGENTES*. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro, 2012. Citado 4 vezes nas páginas 9, 40, 41 e 60.

- 14 KUNDERT, K. *The designer's guide book series*. 1st edition. ed. New York: [s.n.], 2004. Citado na página 41.
- 15 TEXAS INSTRUMENTS. *Noise Analysis in Operational Amplifier Circuits*. [S.l.], 2007. Citado na página 45.

Apêndices

APÊNDICE A – Modelagem em Verilog-AMS do Amp op utilizado no circuito LNA

```

1 // VerilogA for tte_unb_v2, rx_lna, veriloga
2 'include "constants.vams"
3 'include "disciplines.vams"
4
5 module rx_lna_v1(out, agnd, avdd, in);
6
7     //pins declaration
8     output out;
9     electrical out;
10    inout agnd;
11    electrical agnd;
12    inout avdd;
13    electrical avdd;
14    input in;
15    electrical in;
16    electrical out1; //internal node
17
18    //parameters declaration
19    parameter real gain_db=10;
20    parameter real zin=1K;
21    parameter real zout=10;
22    //local variables declaration
23    parameter real realgain = pow(10, gain_db/20.0);
24    parameter real zbias = 5K;
25
26    analog begin
27        V(in,agnd) <+ zin*I(in,agnd); //input impedance
28        V(out1,agnd) <+ realgain*V(in,agnd);
29        V(out,out1) <+ zout*I(out,out1);
30        //power consumption
31        V(avdd,agnd) <+ zbias*I(avdd,agnd);
32    end
33 endmodule

```


B Modelagem em Verilog-AMS do Amp op utilizado no filtro passa alta

```

1 // VerilogA for tte_unb_v2, rx_lna, veriloga
2
3 'include "constants.vams"
4 'include "disciplines.vams"
5
6 module rx_lpf_opamp(out, agnd, avdd, inp, inn);
7     //pins declaration
8     output out;
9     electrical out;
10    inout agnd;
11    electrical agnd;
12    inout avdd;
13    electrical avdd;
14    input inp, inn;
15    electrical inp, inn;
16    electrical out1; //internal node
17    //parameters declaration
18    localparam real gain_db=130;
19    parameter real zin=40K;
20    parameter real zout=10;
21    //local variables declaration
22    parameter real realgain = pow(10, gain_db/20.0);
23    parameter real zbias = 5K;
24
25 analog begin
26     V(inp,inn) <+ zin*I(inp,inn); //input impedance
27     V(out1,agnd) <+ realgain*V(inp,inn);
28     V(out,out1) <+ zout*I(out,out1);
29     //power consumption
30     V(avdd,agnd) <+ zbias*I(avdd,agnd);
31 end
32
33 endmodule

```


C Modelagem em Verilog-AMS do filtro passa alta

```

1 // VerilogA for tte_unb_v2, rx_demod_lpf, veriloga
2
3 'include "constants.vams"
4 'include "disciplines.vams"
5
6 module rx_lpf(out, agnd, avdd, in);
7     //pins declaration
8     output out;
9     electrical out;
10    inout agnd;
11    electrical agnd;
12    inout avdd;
13    electrical avdd;
14    input in;
15    electrical in;
16    electrical out1; //internal node
17    //parameters declaration
18    localparam real gain_db=10;
19    parameter real zin=1K;
20    parameter real zout=10;
21    parameter real f3db=5.04k;
22    //local variables declaration
23    parameter real realgain = pow(10, gain_db/20.0);
24    parameter real zbias = 5K;
25
26 analog begin
27     V(in,agnd) <+ zin*I(in,agnd); //input impedance
28     V(out1,agnd) <+ realgain*laplace_nd(V(in,agnd), {1}, {1,
29         1/'M_TWO_PI/f3db});
30     V(out,out1) <+ zout*I(out,out1);
31     //power consumption
32     V(avdd,agnd) <+ zbias*I(avdd,agnd);
33
34 end
endmodule

```


D Código do PGA

```
1  /*
2  =====
3  Name      : ads1115.c
4  Author    : Vitor Guedes da Silva
5  Description : ADS1115
6  =====
7  */
8
9  #include <stdio.h>
10 #include <stdlib.h>
11 #include <unistd.h>
12 #include <linux/i2c-dev.h>
13 #include <sys/ioctl.h>
14 #include <fcntl.h>
15
16 #include "ads1115_rpi.h"
17
18 //char *bus = "/dev/i2c-1";
19 int i2cFile;
20 //int deviceAddr = 0x48;
21 unsigned char writeBuf[3] = {0};
22
23 int openI2CBus(char *bus)
24 {
25     if ((i2cFile = open(bus, O_RDWR)) < 0)
26     {
27         printf("Failed to open the bus. \n");
28         return -1;
29     } else {
30         printf("Bus open \n");
31         return 1;
32     }
33 }
34
35 int setI2CSlave(unsigned char deviceAddr)
36 {
```

```

37     if(ioctl(i2cFile, I2C_SLAVE, deviceAddr) < 0)
38     {
39         printf("Failed to set I2C_SLAVE at address: 0x%x.
40             \n", deviceAddr);
41         return -1;
42     } else {
43         printf("I2C_SLAVE set at address: 0x%x \n",
44             deviceAddr);
45         return 1;
46     }
47 }
48 float readVoltage(int channel)
49 {
50     unsigned char readBuf[2] = {0};
51     unsigned int analogVal;
52     float voltage;
53     unsigned int config = 0;
54
55     config =          CONFIG_REG_OS_SINGLE          |
56                  CONFIG_REG_PGA_4_096V          |
57                  CONFIG_REG_MODE_CONTIN          |
58                  CONFIG_REG_DR_860SPS
59                  |
60                  CONFIG_REG_CMODE_TRAD          |
61                  CONFIG_REG_CPOL_ACTIV_LOW       |
62                  CONFIG_REG_CLATCH_NONLATCH      |
63                  CONFIG_REG_CQUE_NONE;
64
65     void configDevice(unsigned int config)
66     {
67         writeBuf[0] = 0x01;
68         writeBuf[1] = config >> 8;
69         writeBuf[2] = config && 0xFF;
70         write(i2cFile, writeBuf, 3);
71         usleep(25);
72     }
73
74     switch (channel) {
75         case 0:
76             config |= CONFIG_REG_MUX_CHAN_0;

```



```
76         break;
77     case 1:
78         config |= CONFIG_REG_MUX_CHAN_1;
79         break;
80     case 2:
81         config |= CONFIG_REG_MUX_CHAN_2;
82         break;
83     case 3:
84         config |= CONFIG_REG_MUX_CHAN_3;
85         break;
86     default:
87         printf("Give a channel between 0-3\n");
88     }
89     configDevice(config);
90     usleep(135000);
91
92     writeBuf[0] = 0x00;
93     write(i2cFile, writeBuf, 1);
94
95     if(read(i2cFile, readBuf, 2) != 2) // read data and check
96         error
97     {
98         printf("Error : Input/Output Error \n");
99     }
100    else
101    {
102        analogVal = readBuf[0] << 8 | readBuf[1];
103        voltage = (float)analogVal*4.096/32767.0;
104    }
105
106    return voltage;
107 }
108
109 /*
110 =====
111 Name      : gpio.c
112 Author    : Vitor Guedes da Silva
113 Description : INOUT GPIO
114 =====
```

```
115  */
116
117
118  #include <sys/stat.h>
119  #include <sys/types.h>
120  #include <fcntl.h>
121  #include <stdio.h>
122  #include <stdlib.h>
123  #include <unistd.h>
124  #include <stdbool.h>
125  #include <time.h>
126  #include <signal.h>
127
128  //---Macros---//
129  #define HIGH    1
130  #define LOW     0
131  #define INPUT   0
132  #define OUTPUT  1
133
134  //---Prototipos das funcoes---//
135
136  bool access_gpio(int pin);
137  bool export_gpio(int pin);
138  bool direction_gpio(int pin, int direction);
139  bool value_out_gpio(int pin, int value);
140  bool value_in_gpio(int pin, int value);
141  bool gpio(int pin, int direction, int power);
142
143  void delay(float time);
144  //-----//
145
146  //---Variaveis Globais-----//
147  int button;
148  float timeSleep = 0.5;
149  int arquite;
150  char buffer[3];
151  char path[35];
152  //-----//
153
154  bool gpio(int pin, int direction, int power)
155  {
156      // === Verifica se o pino ja foi exportado === //
```

```
157     access_gpio(pin);
158     // === Export do Pino === //
159     export_gpio(pin);
160     // === Configurando Direcao do Pino === //
161     if(direction == 1){
162         direction_gpio(pin, OUTPUT);
163         //=== Escrevendo no pino ===//
164         if(power == 1 || power == 0){
165             value_out_gpio(pin, power);
166
167         } else{
168             printf("Erro: valor do power incomum\n");
169         }
170     }
171     else{
172         direction_gpio(pin, INPUT);
173         if((value_in_gpio(pin, 0) == 0)) //Button press
174         {
175             gpio(19,1,1);
176             delay(timeSleep);
177             printf("\n\nButton press \n\n\n");
178             return true;
179         }
180         else
181         {
182             gpio(19,1,0);
183             delay(timeSleep);
184             //printf("Button not press\n");
185             return false;
186         }
187     }
188 }
189
190 bool access_gpio(int pin)
191 {
192     snprintf(path, 35, "/sys/class/gpio/gpio%d/direction",
193             pin);
194     if (access(path, 0) == -1) // Arquivo nao existe
195     {
196         return true;
197     } else // Arquivo existe
198     {
```

```
198         return false;
199     }
200 }
201
202 bool export_gpio(int pin)
203 {
204     arquivo = open ("/sys/class/gpio/export", O_WRONLY);
205     if (arquivo==-1)
206     {
207         printf("Arquivo abriu incorretamente\n");
208         return false;
209     }
210     snprintf(buffer, 3, "%d", pin);
211     if(write(arquivo, buffer, 3) == -1)
212     {
213         close(arquivo);
214         return false;
215     }
216
217     close(arquivo);
218
219     return true;
220 }
221
222 bool direction_gpio(int pin, int direction)
223 {
224     arquivo=0;
225     snprintf(path, 35, "/sys/class/gpio/gpio%d/direction",
226             pin);
227     arquivo = open (path, O_WRONLY);
228     if (arquivo==-1)
229         return false;
230
231     snprintf(buffer, 3, "%d", pin);
232     if (write( arquivo, ((direction == INPUT)?"in":"out"), 3
233             )== -1)
234     {
235         close(arquivo);
236         return false;
237     }
238
239     close(arquivo);
240
241     return true;
242 }
```

```
238 }
239
240 bool value_in_gpio(int pin, int value)
241 {
242     arquivo=0;
243     char retorno[3];
244     snprintf(path, 35, "/sys/class/gpio/gpio%d/value", pin);
245     arquivo = open(path, O_RDONLY);
246     //printf("Descritor do arquivo: %d \n", arquivo);
247     if (arquivo == -1)
248     {
249         return false;
250     }
251     if (read(arquivo, retorno, 3) == -1)
252     {
253         close(arquivo);
254         return false;
255     }
256     close(arquivo);
257     return atoi(retorno);
258 }
259
260 bool value_out_gpio(int pin, int value)
261 {
262     arquivo=0;
263     snprintf(path, 35, "/sys/class/gpio/gpio%d/value", pin);
264     arquivo = open(path, O_WRONLY);
265     if (arquivo == -1)
266     {
267         return false;
268     }
269     if (write (arquivo, ((value == HIGH)?"1":"0"), 1) == -1)
270     {
271         close(arquivo);
272         return false;
273     }
274     close(arquivo);
275     return true;
276 }
277
278 void delay(float time)
279 {
```

```
280     struct timespec t;
281     int seg;
282     seg = time;
283     t.tv_sec = seg;
284     t.tv_nsec = (time-seg)*1e9;
285     nanosleep(&t, NULL);
286 }
287
288
289 /*
290 =====
291 Name      : detetorDePicoENivel.c
292 Author    : Vitor Guedes da Silva
293 Description : Detetor of pico for analog signal
294 =====
295 */
296
297 #include <stdio.h>
298 #include <stdlib.h>
299 #include <unistd.h>
300 #include <linux/i2c-dev.h>
301 #include <sys/ioctl.h>
302 // #include <conio.h>
303 #include <fcntl.h>
304
305 #include "ads1115_rpi.h"
306
307 #define LOW 0
308 #define HIGH 1
309
310 _Bool gpio(int pin, int direction, int power);
311
312 void detetorDePico();
313 void detetorDeNivel();
314 void delay(float time);
315
316 float valorDePicoPos = 0; float valorDePicoNeg = 3.3; float
    valorDeTransicao = 0;
317 float Vpp, Vout;
318 int i = 0;
```

```
319 int G0 = 0, G1 = 0, G2 = 0;
320
321 void detetorDePico() {
322
323     if(openI2CBus("/dev/i2c-1") == -1)
324     {
325         printf("EXIT_FAILURE");
326     }
327     setI2CSlave(0x48);
328
329     while(1)
330     {
331         valorDeTransicao = readVoltage(0);
332         if (valorDePicoPos <= valorDeTransicao &&
333             valorDeTransicao <= 3.3) {
334             if (gpio(26,0,0)){
335                 valorDePicoNeg = 3.3;
336                 valorDePicoPos = 0;
337             }
338             valorDePicoPos = valorDeTransicao
339                 ;
340         } else if (valorDePicoNeg >=
341             valorDeTransicao && valorDeTransicao >=
342             0) {
343             if (gpio(26,0,0)){
344                 valorDePicoNeg = 3.3;
345                 valorDePicoPos = 0;
346             }
347             valorDePicoNeg = valorDeTransicao
348                 ;
349         } else {
350             if (gpio(26,0,0)){
351                 valorDePicoNeg = 3.3;
352                 valorDePicoPos = 0;
353             }
354
355             valorDeTransicao =
356                 valorDeTransicao;
357
358         }
359
360         Vpp = valorDePicoPos - valorDePicoNeg;
361     }
362 }
```

```
354         printf("Valtage pico a pico = %.3f V \n", Vpp);
355         detetorDeNivel();
356         delay(0.3);
357     }
358 }
359
360 void detetorDeNivel() {
361     if (Vpp > 3.3){
362         Vout = Vpp*1;
363         G2 = 0; G1 = 0; G0 = 1;
364         gpio(21, 1, 0);
365         gpio(20, 1, 0);
366         gpio(16, 1, 1);
367     } else if (3.3 >= Vpp && Vpp > 1.5){
368         Vout = Vpp*1;
369         G2 = 0; G1 = 0; G0 = 1;
370         gpio(21, 1, 0); //G2
371         gpio(20, 1, 0); //G1
372         gpio(16, 1, 1); //G0
373     } else if (1.5 >= Vpp && Vpp > 0.6){
374         Vout = Vpp*2;
375         G2 = 0; G1 = 1; G0 = 0;
376         gpio(21, 1, 0); //G2
377         gpio(20, 1, 1); //G1
378         gpio(16, 1, 0); //G0
379     } else if (0.6 >= Vpp && Vpp > 0.3){
380         Vout = Vpp*5;
381         G2 = 0; G1 = 1; G0 = 1;
382         gpio(21, 1, 0); //G2
383         gpio(20, 1, 1); //G1
384         gpio(16, 1, 1); //G0
385     } else if (0.3 >= Vpp && Vpp > 0.15){
386         Vout = Vpp*10;
387         G2 = 1; G1 = 0; G0 = 0;
388         gpio(21, 1, 1); //G2
389         gpio(20, 1, 0); //G1
390         gpio(16, 1, 0); //G0
391     } else if (0.15 >= Vpp && Vpp > 0.06){
392         Vout = Vpp*20;
393         G2 = 1; G1 = 0; G0 = 1;
394         gpio(21, 1, 1); //G2
395         gpio(20, 1, 0); //G1
```



```
396         gpio(16, 1, 1); //G0
397     } else if (0.06 >= Vpp && Vpp > 0.03){
398         Vout = Vpp*50;
399         G2 = 1; G1 = 1; G0 = 0;
400         gpio(21, 1, 1); //G2
401         gpio(20, 1, 1); //G1
402         gpio(16, 1, 0); //G0
403     } else if (Vpp <= 0.03){
404         Vout = Vpp*100;
405         G2 = 1; G1 = 1; G0 = 1;
406         gpio(21, 1, 1); //G2
407         gpio(20, 1, 1); //G1
408         gpio(16, 1, 1); //G0
409     } else {
410         Vout = 0;
411         G2 = 0; G1 = 0; G0 = 0;
412         gpio(21, 1, 0); //G2
413         gpio(20, 1, 0); //G1
414         gpio(16, 1, 0); //G0
415     }
416     printf("Tens o de sa da = %.3f V \n", Vout);
417     printf("Sa da G = %d %d %d\n", G2, G1, G0);
418 }
419
420
421 /*
422 =====
423 Name      : main_PGA.c
424 Author    : Vitor Guedes da Silva
425 Description : PGA
426 =====
427 */
428
429 #include <stdio.h>
430 #include <stdlib.h>
431 #include <unistd.h>
432 #include <linux/i2c-dev.h>
433 #include <sys/ioctl.h>
434 #include <fcntl.h>
435
```

```
436 #include "ads1115_rpi.h"
437
438 float detetorDePico();
439 _Bool gpio(int pin, int direction, int power);
440
441 int main(void) {
442     detetorDePico();
443 }
```

E Modelagem em Verilog-AMS do PGA

```

1 // VerilogA for tte_unb_v2, rx_vga, veriloga
2
3 'include "constants.vams"
4 'include "disciplines.vams"
5
6 module rx_pga1(out, agnd, avdd, in, G0, G1, G2);
7
8     //pins declaration
9     output out;
10    electrical out;
11    inout agnd;
12    electrical agnd;
13    inout avdd;
14    electrical avdd;
15    input in;
16    electrical in;
17    input G0;
18    electrical G0;
19    input G1;
20    electrical G1;
21    input G2;
22    electrical G2;
23    electrical out1; //internal node
24
25    //parameters declaration
26    real gain;
27    real zin;
28    real realgain;
29    parameter real zout=10;
30    parameter real zbias = 5K;
31
32    analog begin
33        if(V(G2, agnd) == 0 && V(G1, agnd) == 0 && V(G0,
34            agnd) == 0)
35            gain=0;zin=10k;
36        if(V(G2, agnd) == 0 && V(G1, agnd) == 0 && V(G0,
37            agnd) == 5)
38            gain=0;zin=5k;

```

```
37     if(V(G2, agnd) == 0 && V(G1, agnd) == 5 && V(G0,
38         agnd) == 0)
39         gain=6;zin=2k;
40     if(V(G2, agnd) == 0 && V(G1, agnd) == 5 && V(G0,
41         agnd) == 5)
42         gain=14;zin=1k;
43     if(V(G2, agnd) == 5 && V(G1, agnd) == 0 && V(G0,
44         agnd) == 0)
45         gain=20;zin=1k;
46     if(V(G2, agnd) == 5 && V(G1, agnd) == 0 && V(G0,
47         agnd) == 5)
48         gain=26;zin=1k;
49     if(V(G2, agnd) == 5 && V(G1, agnd) == 5 && V(G0,
50         agnd) == 0)
51         gain=34;zin=1k;
52     if(V(G2, agnd) == 5 && V(G1, agnd) == 5 && V(G0,
53         agnd) == 5)
54         gain=40;zin=1k;
55
56     realgain = pow(10, gain/20.0);
57
58     V(in,agnd) <+ zin*I(in,agnd);
59     V(out1,agnd) <+ realgain*V(in,agnd);
60     V(out,out1) <+ zout*I(out,out1);
61
62     V(avdd,agnd) <+ zbias*I(avdd,agnd);
63
64     end
65
66 endmodule
```