



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

**Modelagem em Verilog-AMS de uma Tag
Passiva de RFID e Projeto Elétrico do
Demodulador ASK**

Autor: Marlon Carvalho Portugal Filho
Orientador: Dr. Wellington Avelino do Amaral

Brasília, DF
2014



Marlon Carvalho Portugal Filho

Modelagem em Verilog-AMS de uma Tag Passiva de RFID e Projeto Elétrico do Demodulador ASK

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: Dr. Wellington Avelino do Amaral

Brasília, DF

2014

Marlon Carvalho Portugal Filho

Modelagem em Verilog-AMS de uma Tag Passiva de RFID e Projeto Elétrico do Demodulador ASK/ Marlon Carvalho Portugal Filho. – Brasília, DF, 2014-107 p. : il. (algumas color.) ; 30 cm.

Orientador: Dr. Wellington Avelino do Amaral

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2014.

1. RFID. 2. Verilog-AMS. I. Dr. Wellington Avelino do Amaral. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Modelagem em Verilog-AMS de uma Tag Passiva de RFID e Projeto Elétrico do Demodulador ASK

CDU 02:141:005.6

Marlon Carvalho Portugal Filho

Modelagem em Verilog-AMS de uma Tag Passiva de RFID e Projeto Elétrico do Demodulador ASK

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, 18 de Novembro de 2014:

Dr. Wellington Avelino do Amaral
Orientador

Dr. Gilmar Silva Beserra
Convidado 1

Dr. Daniel Mauricio Muñoz Arboleda
Convidado 2

Brasília, DF
2014

*Este trabalho é dedicado à todos aqueles que, de alguma forma,
proporcionaram uma oportunidade de crescermos juntos.*

*“Eu poderia viver recluso numa casca de noz
e me considerar rei do espaço infinito...”
(Hamlet, Shakespeare, Ato 2, Cena 2)*

Resumo

Este trabalho visa desenvolver os blocos constituintes do *front-end* analógico de uma *tag* passiva de RFID para 13,56 MHz utilizando a linguagem Verilog-AMS. A metodologia adotada para o desenvolvimento do projeto foi a *Top-Down* que, tem como objetivo, verificar o sistema em um nível abstrato antes de iniciar seu projeto mais detalhado. Esse método é indicado para projetos extensos e mais complexos, onde, tem-se no planejamento da modelagem do sistema um ponto chave para seu desenvolvimento. É modelado em Verilog-AMS os blocos individuais que compõem o *front-end* analógico da *tag*, que são: o retificador, modulador, demodulador e oscilador. Os modelos criados foram validados utilizando o simulador CADENCE Virtuoso e obtido os resultados esperados e planejados. Com isso, foi feito o projeto elétrico do demodulador ASK com a tecnologia TSMC 180nm, enviando o *layout* do circuito para fabricação do chip. Ao mesmo tempo, simulações mistas dos blocos em Verilog-AMS com o circuito elétrico do demodulador ASK foram realizadas a fim de validar os modelos e a metodologia. Com o Verilog-AMS foi possível melhorar o desempenho de um projeto complexo a partir de uma modelagem em alto nível, seguindo essa metodologia aplicada para projetos de circuitos integrados. Portanto, por ter sido implementado blocos individuais constituintes do sistema, as mesmas, posteriormente, poderão ser utilizadas em outros projetos que não necessariamente são uma *tag* de RFID.

Palavras-chaves: Verilog-AMS. RFID. Simulações Mistas. Metodologia Top-Down.

Abstract

This work aims the development of the analog front-end blocks of a passive RFID tag to 13.56MHz using Verilog-AMS language. Top-Down was the adopted methodology to develop this project, that has, as goal, verify the system as an abstract level before beginning it's detailed design. This method is indicated to extensive and complexes projects, where, has the modeling plan of the system as a key to it's development. It is designed, in Verilog-AMS, the analog front-end individual blocks, they are: rectifier, modulator, demodulator and oscillator. The created models were validated using CADENCE Virtuoso as simulator and it's expected and planned results were obtained. With this, the ASK demodulator electric circuit was designed using TSMC 180 nm, sending it to manufacture. At the same time, mixed simulations with the blocks in Verilog-AMS and the electric circuit of the ASK demodulator are run in order to validate the models and methodology. With the Verilog-AMS was possible to enhance the performance of a complex project from a high level modeling, guided by this methodology applied to integrated circuits. Thus, as the individual blocks implemented, they, afterwards, can be used in another projects which, not necessarily, are a RFID tag.

Key-words: Verilog-AMS. RFID. Top-Down Design.

Lista de ilustrações

Figura 1 – Relação entre Verilog-AMS, Verilog-HDL e Verilog-A (KUNDERTH, 2004)	35
Figura 2 – (a) Denominação dos Blocos do <i>Tag</i> de RFID (b) Denominação dos Blocos Específicos do <i>Tag</i> de RFID (LI, 2009)	39
Figura 3 – Onda ASK (RAZAVI, 1998)	40
Figura 4 – Diagrama de Blocos do Demodulador ASK	41
Figura 5 – Esquemático de Funcionamento do Circuito de <i>Backscattering</i> (ASHRY; SHARAF; IBRAHIM, 2009)	42
Figura 6 – Esquemático padrão para LDO	45
Figura 7 – Esquemático para Bandgap em Nível de Circuito (PINTO et al., 2014)	48
Figura 8 – Esquemático de um Modulador BPSK	48
Figura 9 – Representação da impedância da chave quando fechada	49
Figura 10 – Esquemático de uma chave MOS	49
Figura 11 – Amplitude da Onda Refletida (ATMEL, 2005)	50
Figura 12 – Modulação BPSK da Subportadora de 847,5 kHz (ATMEL, 2005)	51
Figura 13 – Circuito externo da antena (MICROCHIP, 2004)	51
Figura 14 – Esquemático de um oscilador em anel padrão	52
Figura 15 – Tangente Hiperbólica Deslocada em +1	58
Figura 16 – Tangente Hiperbólica Multiplicada pela sua Componente Deslocada em +1	59
Figura 17 – Função Utilizada para Saturação do Amplificador com Exemplo	60
Figura 18 – Transiente do Modelo do Amplificador Operacional para uma Realimentação de Ganho -2 V/V	61
Figura 19 – Comparativo entre Simulações Transiente para Diferentes Tensões de Saturação	62
Figura 20 – Transiente do Charge Pump	63
Figura 21 – Gráfico da resposta matemática da função do bandgap	64
Figura 22 – Transiente do Circuito do Regulador de Tensão (LDO)	65
Figura 23 – Comparativo entre os Níveis de Tensão na Saída do Charge Pump e LDO	66
Figura 24 – Resposta de um comparador a uma entrada ruidosa (a) Comparador sem histerese (b) Comparador com histerese (ALLEN; HOLBERG, 2002)	67
Figura 25 – Transiente do Comparador com Histerese	69
Figura 26 – Transiente dos Modelos de Detector de Envoltória e Filtro de Média	71
Figura 27 – Transiente do Sinal Modulado	72
Figura 28 – Transiente do Oscilador	74
Figura 29 – Tempos de Subida e Descida do Oscilador	74

Figura 30 – Transiente do Sistema completo do Demodulador ASK em Verilog-AMS	75
Figura 31 – Esquemático do Detector de Envoltória do Sinal RF de Entrada	76
Figura 32 – Esquemático do Filtro de Média do Sinal Retificado	77
Figura 33 – Resultados Obtidos pelo Circuito (PINTO et al., 2014)	78
Figura 34 – Simulação Mista do Demodulador ASK Utilizando um Comparador com Histerese e o Filtro de Média em Verilog-AMS	79
Figura 35 – Simulação Mista do Demodulador ASK Utilizando um Comparador com Histerese em Verilog-AMS	79
Figura 36 – Esquemático do comparador com histerese.	105
Figura 37 – Layout do demodulador ASK.	107

Lista de tabelas

Tabela 1 –	Descrições dos Padrões de ISOs para RFID (RFIDCANADA, 2012) . . .	28
Tabela 2 –	Comparativo entre as principais características das faixas de frequência mais comuns em RFID	29
Tabela 3 –	Valores Lógicos em Verilog (KUNDERTH, 2004).	37
Tabela 4 –	Relação entre o índice de modulação com a profundidade (ATMEL, 2005)	50
Tabela 5 –	Descrição dos Pinos do Amplificador Operacional	54
Tabela 6 –	Descrição dos Pinos do Bandgap	54
Tabela 7 –	Descrição dos Pinos do Charge Pump	54
Tabela 8 –	Descrição dos Pinos da Chave	55
Tabela 9 –	Descrição dos Pinos do Comparador	55
Tabela 10 –	Descrição dos Pinos do Detector de Envoltória	55
Tabela 11 –	Descrição dos Pinos do Filtro de Média	55
Tabela 12 –	Descrição dos Pinos do Oscilador	55

Lista de abreviaturas e siglas

AC	<i>Alternate Current</i>
AM	<i>Amplitude Modulation</i>
ASK	<i>Amplitude Shifting Keying</i>
BPSK	<i>Binary Phase Shifting Keying</i>
CMOS	<i>Complementary Metal-Oxide Semiconductor</i>
DC	<i>Direct Current</i>
HDL	<i>Hardware Description Language</i>
HF	<i>High-frequency</i>
ID	<i>Identification</i>
IEC	<i>International Electrotechnical Commission</i>
ISO	<i>International Organization for Standardization</i>
LDO	<i>Low-dropout Regulator</i>
LF	<i>Low-frequency</i>
MOSFET	<i>Metal-Oxide Semiconductor Field Effect Transistor</i>
PSK	<i>Phase Shifting Keying</i>
RC	Resistor Capacitor
RFID	<i>Radio Frequency Identification</i>
TSMC	<i>Taiwan Semiconductors</i>
UHF	<i>Ultra High-frequency</i>
VCO	<i>Voltage-Controlled Oscillator</i>

Lista de símbolos

μ	Letra grega Mu representa um fator de 10^{-6} em uma unidade de medida
ω_c	Frequência angular de um sinal
ϕ	Fase de um sinal
$<+$	Atribuição em Verilog-AMS
\sim	Inversor lógico em Verilog-AMS

Sumário

1	INTRODUÇÃO	25
1.1	Contextualização	25
1.2	Objetivos	25
1.3	Motivação	26
1.4	Organização	26
2	FUNDAMENTAÇÃO TEÓRICA	27
2.1	Tipos e Padrões de RFID	27
2.2	Aplicação dos Padrões de RFID	28
3	METODOLOGIA <i>TOP-DOWN</i>	31
3.1	Aproximações Tradicionais Para Projetos de Sinais Mistos	31
3.1.1	Projeto <i>Bottom-Up</i>	31
3.1.2	Projeto <i>Top-Down</i>	31
3.2	Princípios da Metodologia <i>Top-Down</i>	31
3.3	Processo de Desenvolvimento <i>Top-Down</i>	32
3.3.1	Planejamento de Simulação e Modelagem	32
3.3.2	Verificação em Nível de Sistema	33
3.3.3	Simulação Mista	33
3.3.4	Verificação <i>Bottom-Up</i>	34
3.3.5	Verificação Final	34
3.3.6	Teste	34
3.3.7	Benefícios da Metodologia	34
3.4	Linguagens de Descrição de <i>Hardware</i>	34
3.5	Família de Linguagens Verilog	35
3.6	Linguagem Verilog-AMS	36
4	CONCEPÇÃO DO SISTEMA DA TAG DE RFID	39
4.1	Retificador	40
4.2	Demodulador	40
4.3	Modulador	41
4.3.1	PSK <i>Backscattering</i>	42
4.4	Oscilador	42
5	PLANEJAMENTO DA MODELAGEM	45
5.1	Bloco Retificador	45

5.2	Bloco Modulador	48
5.3	Bloco Demodulador	51
5.4	Bloco Oscilador	52
5.5	Planejamento das Simulações Mistas	53
5.6	Descrição dos Pinos	54
6	PROJETO E MODELAGEM DO SISTEMA	57
6.1	Amplificador Operacional	57
6.2	Charge Pump	62
6.3	Bandgap	63
6.4	Regulador de Tensão (LDO)	65
6.5	Comparador	66
6.6	Detector de Envoltória e Filtro de Média	69
6.7	Modulador	71
6.8	Oscilador	72
6.9	Demodulador ASK em Verilog-AMS	74
6.10	Demodulador ASK em Nível de Circuito	75
6.11	Simulações Mistas	78
7	CONCLUSÃO	81
	 Referências	 83
	 APÊNDICES	 85
	 APÊNDICE A – CÓDIGO DO AMPLIFICADOR OPERACIONAL	 87
	 APÊNDICE B – CÓDIGO DO BANDGAP	 89
	 APÊNDICE C – CÓDIGO DO CHARGE PUMP	 91
	 APÊNDICE D – CÓDIGO DA CHAVE	 93
	 APÊNDICE E – CÓDIGO DO COMPARADOR	 95
	 APÊNDICE F – CÓDIGO DO DETECTOR DE ENVOLTÓRIA	 97
	 APÊNDICE G – CÓDIGO DO FILTRO DE MÉDIA	 99
	 APÊNDICE H – CÓDIGO DO OSCILADOR EM ANEL	 101

APÊNDICE I – ESQUEMÁTICO DO COMPARADOR COM HISTERESE	105
APÊNDICE J – LAYOUT DO DEMODULADOR ASK	107

1 Introdução

1.1 Contextualização

A utilização das *tags* de RFID é crescente na medicina. Estudos sobre implantes médicos vem sendo desenvolvidos como, por exemplo, uma *tag* aplicada à uma prótese ortopédica onde o médico, após o implante, tem a possibilidade de monitorar a prótese, auxiliando no tratamento do paciente (LIU et al., 2013). Além disso, pacientes que precisam de um acompanhamento rigoroso na ingestão de medicamentos podem ser monitorados com pílulas que contém uma *tag* de RFID. Com isto, é possível saber se foi tomado o medicamento correto e se ele foi ingerido na quantidade certa (RAJAGOPALAN; RAHMAT-SAMII, 2010). Outro exemplo de aplicação é o monitoramento de pessoas que sofreram traumas leves em tecidos, causados por tratamento de câncer, feridas por estilhaços causados em combate, ataques terroristas ou acidentes de carro (RIGELSFORD; DAVENPORT, 2013).

No geral o RFID possui diversas aplicações em inúmeras áreas de interesses, que vão desde agronomia até logística de produção, passando pela área de saúde e controle de acesso. É bastante utilizado para monitoramento animal, onde o gado, por exemplo, é supervisionado e identificado via *tags* de radiofrequência, que apresentam melhor solução que o sistema de tecnologia baseado em códigos de barras, também utilizados em fazendas menores. Ainda nesse segmento, com o RFID pode-se automatizar alguns processos, como o de monitoramento do ciclo de leite produzido por vacas (STEVAN, 2012). Adicionalmente, o identificador pode ser aplicado em hospitais para identificação automática de materiais, medicações ou pessoas, visando otimizar a comunicação (JOSE; MONICA, 2011). Tanto em hospitais, como citado anteriormente, quanto em empresas e estabelecimentos em geral que requerem acesso controlado, o RFID é comumente utilizado para o controle do acesso de pessoas e identificação das mesmas.

1.2 Objetivos

Este trabalho visa o desenvolvimendo da modelagem do *front end* analógico de uma *tag* HF (*High-frequency*) passiva de RFID em linguagem de alto nível (Verilog-AMS) pela metodologia *Top-Down*. Essa modelagem será realizada tanto em nível de blocos, quanto em nível de sistema, possibilitando a realização de simulações mistas da *tag*.

Além do projeto da *tag* de RFID o trabalho visa:

- Validar os modelos projetados em linguagem de hardware (Verilog-AMS) utilizando

o software CADENCE Virtuoso.

- Validar o bloco demodulador em nível de sistema (Verilog-AMS) comparando os resultados obtidos com o projeto em nível de circuito.
- Disponibilizar os modelos desenvolvidos aos alunos de graduação da Universidade de Brasília para utilização em projetos de pesquisa e em disciplinas de microeletrônica analógica e RF.
- Aplicar as ferramentas de ponta utilizadas em *Design Houses* e sua metodologia de projeto.

1.3 Motivação

O desenvolvimento de um trabalho na área de microeletrônica, que aborde metodologias de mercado, desperta nos alunos maior interesse na área, fazendo com que mais projetos sejam iniciados, destacando tanto os alunos e professores, como a universidade, no cenário acadêmico e industrial ao expor que projetos na área estão em constante desenvolvimento.

Complementando, a criação de modelos em alto nível para uso de alunos da universidade irá auxiliar o desenvolvimento de novos projetos, ampliando a área de estudo em RF. O desenvolvimento de novas pesquisas irá gerar oportunidades de projetos junto à indústria, motivando o desenvolvimento de uma visão comercial pelos novos estudantes.

Adquirir conhecimentos em microeletrônica de radiofrequência proporcionando uma formação acadêmica mais completa.

1.4 Organização

Este trabalho se divide em seis capítulos. Inicialmente é apresentada uma introdução geral sobre os tipos de RFID e suas aplicações. Após essa apresentação, no capítulo 2 é abordada a metodologia *Top-Down*, onde o Verilog-AMS é utilizado. Em seguida, no capítulo 3, é apresentada uma visão geral do sistema a ser desenvolvido. No capítulo 4 é realizado o planejamento da modelagem do sistema proposto, apresentando características importantes e quais blocos serão desenvolvidos. No capítulo 5 é apresentado como foi conduzida a modelagem, estratégias tomadas para representar os blocos pela linguagem Verilog-AMS bem como os resultados obtidos individualmente, em conjunto e simulações mistas. E, por fim, no capítulo 6, as conclusões chegadas pelo desenvolvimento do projeto.

2 Fundamentação Teórica

RFID (*Radio-frequency Identification*) é uma forma de identificação automática que vem substituindo as tradicionais tecnologias, como código de barras, devido a sua superior capacidade de exercer essa função, possibilitando aplicações de rastreamento e identificação que as demais técnicas não eram capazes de realizar.

2.1 Tipos e Padrões de RFID

Dentre os tipos de RFID há a *tag* ativa, semi-passiva e a passiva. A diferença entre eles se deve a forma de como são alimentados. A *tag* ativa utiliza uma fonte de energia interna, utilizada para energizar todo seu circuito, nesse caso, uma bateria. A *tag* semi-passiva, apesar de usar uma bateria para alimentar seu circuito, depende do leitor para fornecer sua energia para comunicação. Já a *tag* passiva depende da energia do sinal de RF transferida do leitor para ser alimentado, ou seja, o próprio sinal que é recebido o energiza. A *tag* ativa e semi-passiva de RFID têm vantagem sobre as passivas quando se trata de distância de comunicação, velocidade de leitura e escrita e custo. Elas possuem aplicações em transporte, hospitais inteligentes, dentre outras (SHUNYU-SHI et al., 2013).

Os padrões RFID passivos são basicamente divididos em grupos baseados em sua frequência de operação, sendo elas: Baixa frequência (LF - 125 kHz a 135 kHz), Alta frequência (HF - 13,56 MHz), Ultra alta frequência (UHF - 433 MHz e 868 MHz a 956 MHz) e Microonda (2,4 GHz). Para cada grupo citado há uma aplicação mais adequada e padrões determinados por normas, sendo elas determinadas pela ISO (*International Organization for Standardization*) em conjunto com a IEC (*International Electrotechnical Commission*) que, no caso da última, participa do desenvolvimento de padrões internacionais através de comitês técnicos (ISO/IEC, 2004).

A norma ISO/IEC 18000, *Information technology – Radio Frequency identification for item management*, apresenta os parâmetros gerais para padrões de RFID. Cada padrão possui um conjunto de parâmetros de projeto para interface de comunicação. Os padrões são para comunicação em 135 kHz, 13,56 MHz, 2,45 GHz, 860 MHz a 960 MHz e 433 MHz. Além dessa norma há as normas ISO 11784 e 11785 para aplicações animais e a ISO/IEC 14443 que trata de padrões para cartões de identificação sem contato para operações de baixa proximidade com a antena (ATMEL, 2005). Similarmente, há a ISO/IEC 15693 e por fim as ISO/IEC 15961 e ISO/IEC 15962, onde ambas são necessárias para o completo entendimento do protocolo de dados, mas cada uma foca-se em um interface particular (ISO/IEC, 2004).

Sendo assim, como há um padrão para cada faixa de frequência e aplicações, cada função e etapa de desenvolvimento segue uma norma. Na Tabela (1) são apresentados os padrões de RFID determinados pelas ISOs e a descrição de cada um. Partindo desse ponto, o capítulo 2.2 mostrará onde cada faixa de frequência das *tags* passivas é utilizada e suas principais características.

Padrão ISO	Descrição
ISO 11784	Identificação por Rádio Frequência para animais – Estrutura de código
ISO 11785	Identificação por Rádio Frequência para animais – conceito técnico
ISO/IEC 14443A,B	Cartões de Identificação – Cartões de circuito(s) integrado(s) sem contato – Cartões de proximidade
ISO/IEC 15693	Cartões de Identificação – Cartões de circuito(s) integrado(s) sem contato – Cartões de vizinhança
ISO/IEC 18001	Tecnologia da Informação – Técnicas AIDC – RFID para gestão de item – Perfis de requisitos de aplicação
ISO/IEC 18000-1	Parâmetros genéricos para interface de comunicação aérea para frequências globalmente aceitas
ISO/IEC 18000-2	Parâmetros para comunicação de interface aérea até 135 kHz
ISO/IEC 18000-3	Parâmetros para comunicação de interface aérea até 13,56 MHz
ISO/IEC 18000-4	Parâmetros para comunicação de interface aérea a 2,45 GHz
ISO/IEC 18000-6	Parâmetros para comunicação de interface aérea a 860-960 MHz
ISO/IEC 18000-7	Parâmetros para comunicação de interface aérea a 433 MHz
ISO/IEC 15961	RFID para gestão de itens – Protocolo de dados: Interface de aplicação
ISO/IEC 15962	RFID para gestão de itens – Protocolo: Regras de codificação de dados e funções da memória lógica
ISO/IEC 15963	RFID para gestão de itens – Identificação única de etiqueta de RF.

Tabela 1 – Descrições dos Padrões de ISOs para RFID ([RFIDCANADA, 2012](#))

2.2 Aplicação dos Padrões de RFID

As frequências mais comuns utilizadas em RFID passivo de baixa frequência são 125 e 134,2 kHz. Uma característica interessante desse tipo de *tag* é que ela não é afetada por metais, sendo interessante para aplicações de identificações automotivas, identificação de ferramentas, contêineres, acesso à estacionamentos, dentre outros. Além de metais, outros materiais que podem ser penetrados incluem água e tecido biológico o que permite

aplicação em identificação animal. Porém sua utilização é limitada em ambientes industriais, onde pode ocorrer interferências por máquinas elétricas. A distância pode variar de poucos centímetros a alguns metros, dependendo do tamanho da antena e do receptor (RFIDCANADA, 2012).

O RFID passivo de alta frequência opera em 13,56 MHz. Sua utilização, apesar de ser mundialmente aceita e utilizada, varia um pouco em regiões, como por exemplo, na América do Norte é regulamentado que a potência da antena do receptor seja limitado a 3W enquanto que na Europa é permitido 4W. Comparando com a *tag* de baixa frequência, possui boa penetração em materiais, porém a interferência causada por metais é maior, seu custo é mais baixo, e possui uma melhor velocidade de comunicação. Esse tipo de *tag* é utilizada para aplicações que necessitem distâncias de 1m ou menor, tal como controle de acesso e sistemas de segurança. Se adicionada uma memória, permite que a segurança seja aumentada e haja integração com biometria como parte do sistema. Documentos e arquivos podem ser facilmente identificados também. Segundo a empresa RFID Canadá, cartões sem contato ou cartões RFID serão a nova geração de cartões de crédito. É possível ver que cartões RFID estão sendo utilizados para controle de acesso em transportes públicos no mundo. O RFID de alta frequência é ideal para aplicações que requerem identificações de baixo custo (RFIDCANADA, 2012).

A *tag* de ultra alta frequência trabalha no espectro de 300 MHz a 3 GHz. A faixa que possui maior mercado é a de 860-956 MHz, portanto será a tecnologia abordada. UHF é mais indicada quando a aplicação requer uma distância maior do leitor. Comparando com o RFID HF, pode ler até 4 vezes mais *tags* em um receptor, porém não trabalha bem com líquidos. Ela pode substituir as *tags* de HF, por exemplo em cadeias de suprimentos em mercados, onde é necessário maior distância do leitor (RFIDCANADA, 2012).

Na Tabela (2), é apresentado, de forma geral, um comparativo entre as faixas de frequência utilizadas para RFID.

Característica	LF	HF	UHF
Distância Etiqueta-Leitor	Baixa	Baixa	Alta
Taxa de Transmissão	Baixa	Alta	Alta
Custo do Transceptor	Alto	Médio	Baixo
Leitura Simultânea por Vários Tags no Receptor	Não	Sim	Sim
Penetração em Materiais	Muito Alta	Alta	Média

Tabela 2 – Comparativo entre as principais características das faixas de frequência mais comuns em RFID

3 Metodologia *Top-Down*

A metodologia *Top-Down* é comumente utilizada em projetos que possuem uma alta complexidade do sistema. Basicamente, o sistema é projetado e verificado em um nível mais abstrato, ou em um nível de diagrama de blocos, antes de ser iniciado o projeto mais detalhado dos blocos individuais que compõem o sistema (KUNDERTH, 2004).

3.1 Aproximações Tradicionais Para Projetos de Sinais Mistos

Há um sensível aumento da produtividade quando projetistas de sinais mistos migram para uma efetiva produção em metodologia *Top-Down*. Em projetos desenvolvidos com a metodologia *Bottom-Up*, com o aumento da complexidade do sistema, a verificação do mesmo pode ocorrer tardiamente na linha de produção, proporcionando re-ciclos de projeto e aumentando o tempo de desenvolvimento (KUNDERTH, 2004).

3.1.1 Projeto *Bottom-Up*

Nesta metodologia, o projeto é iniciado com o desenvolvimento dos blocos individuais, que são combinados para formar o sistema. O projeto dos blocos inicia com suas especificações e termina com o projeto dos circuitos em nível de transistor. Cada bloco é verificado isoladamente do contexto geral do sistema. Uma vez verificados, eles são combinados a fim de se fazer uma verificação geral, onde todo o sistema está em nível de transistor.

3.1.2 Projeto *Top-Down*

Nesse caso a arquitetura do chip é definida, simulada e otimizada exaustivamente. São retiradas dessa simulação em alto nível as especificações de cada bloco. A partir disso, os blocos são individualmente projetados para atingir essas especificações. Por fim, todo o chip é verificado com base nos requisitos originais.

3.2 Princípios da Metodologia *Top-Down*

Seguindo a metodologia *Top-Down*, evita-se o impacto proveniente de alterações que venham a ocorrer no final do ciclo de projeto. Dessa forma, o circuito teria que ser parcialmente remodelado. Essa metodologia permite que essas mudanças e suas verificações sejam feitas rapidamente (KUNDERTH, 2004).

Um efetivo projeto em metodologia *Top-Down* segue alguns princípios, são eles:

- Representação de projeto compartilhada;
- Cada mudança é verificada;
- Planejamento de verificação;
- Múltiplos passos;
- Planos e especificações executáveis.

Os engenheiros do sistema podem criar uma arquitetura que pode ser validada individualmente por cada projetista, permitindo que o projeto possa ser simulado por todos os membros da equipe.

Considerando que em nível de transistor as simulações são mais lentas e a verificação mais complicada, quando desenvolvido seus blocos individuais em linguagem alto nível, obtém-se uma drástica redução do tempo de simulação, permitindo que o sistema seja verificado exaustivamente.

O planejamento de verificação, foca em antecipar e prevenir problemas que possam surgir ao integrar os blocos ao sistema. Deve ser feita o mais cedo possível e com o máximo possível do sistema desenvolvido em alto nível.

O objetivo com o planejamento da modelagem é identificar o conjunto de modelos que devem ser desenvolvidos. É recomendado o desenvolvimento de modelos simples e otimizados.

São realizados múltiplos passos para reduzir os riscos do projeto enquanto toma-se passos para expor potenciais problemas antecipadamente, trabalhando completamente em um nível abstrato de representação. Enquanto o projeto evolui com seus blocos sendo implementados e mais informação sendo disponibilizada, os modelos são atualizados, podendo ter seus resultados refinados.

As especificações do sistema devem ser preferencialmente manifestados como modelos e *scripts* executáveis, ao invés de documentos escritos. Estes modelos executáveis são, então, utilizados e validados no dia-a-dia. Assim, eliminando ambiguidades que podem ser geradas em documentos escritos, que poderiam resultar em desentendimentos, acarretando no reprojetado de partes do sistema.

3.3 Processo de Desenvolvimento *Top-Down*

3.3.1 Planejamento de Simulação e Modelagem

Inicialmente, identificam-se as áreas de interesse do projeto para então desenvolver planos para como elas serão verificadas. Os planos devem especificar como esses testes se-

rão realizados, identificando quais blocos devem ser representados em nível de transistor. Kunderth menciona que deve-se resistir a tentações de especificar modelos mais complicados que o necessário, começando por modelos simples e modelar adicionais efeitos apenas se preciso. A ênfase ao escrever modelos deve-se dar ao comportamento do bloco e não em sua estrutura.

Outra importante consideração ao modelar os blocos de sistema é pela sua especificação, sendo assim, modelar um bloco quanto sua faixa de operação, evitando especificar uma região que o circuito nunca irá operar.

Seguindo essas regras, economiza-se tempo que seria utilizado escrevendo modelos. Assim, ao obter a descrição em alto nível do sistema, uma vez validado, pode ser aplicado às simulações em nível de transistor.

3.3.2 Verificação em Nível de Sistema

O objetivo é encontrar um algoritmo e arquitetura que implementa uma funcionalidade requisitada, providenciando adequado desempenho à mínimo custo. Esse custo está relacionado com a capacidade de se reduzir qualquer tipo de gasto desnecessário para o projeto. Como o desenvolvimento será feito especificamente para um sistema, pode-se reduzir a área do chip, tempo de desenvolvimento e otimizar o circuito para a aplicação desejada.

Essa fase de projeto é essencial para um entendimento geral do sistema logo no início de seu desenvolvimento. O principal objetivo dessa fase é o projeto preciso de modelos dos blocos e suas interfaces. O Verilog-AMS é utilizado nessa fase porque permite uma modelagem precisa das interfaces e suporta simulações mistas.

3.3.3 Simulação Mista

A simulação mista é necessária para verificar se determinado bloco, validado individualmente, continuará respeitando suas especificações após sua introdução no sistema.

Um bloco representado em alto nível é substituído pelo seu esquemático a nível de transistor antes de se rodar a simulação. Ele deve respeitar as especificações do sistema quando projetado em alto nível. Com relação à quantidade de pinos, impedância, polaridades e etc. Basicamente, deve-se projetar o circuito equivalente em baixo nível (transistores) do bloco descrito em alto nível.

De forma geral, o bloco em alto nível funciona como um *test bench* do circuito descrito em nível de transistor.

3.3.4 Verificação *Bottom-Up*

Após a simulação mista, comparando os resultados alcançados por simulações que envolvem o circuito, a funcionalidade e precisão do modelo podem ser verificadas. Em suma, essa verificação se dá em comparar os resultados obtidos em nível de transistor com o sistema desenvolvido em alto nível e validar suas especificações.

3.3.5 Verificação Final

Nessa etapa, todos os blocos são simulados a nível de transistor pelo contexto do sistema, verificando sua funcionalidade e interface. É possível identificar caminhos críticos de desenvolvimento, áreas de especial interesse, e a performance do sistema. Tudo que se é desejado verificar é feito a nível de transistor. A ideia é reduzir o tempo gasto com simulações SPICE, tornando a verificação do sistema mais efetiva.

3.3.6 Teste

Exclusivamente após o término do projeto são realizados todos os testes do sistema em *softwares*. Com esses testes, durante a fase de projeto, é possível reduzir o tempo de *debug* pós sílicio em 50%, melhorando o rendimento (KUNDERTH, 2004).

3.3.7 Benefícios da Metodologia

- Melhora a comunicação entre engenheiros
- Melhora a produtividade
- Melhora a habilidade de lidar com projetos complexos
- Permite execução paralela de tarefas de projeto
- Portabilidade dos modelos dos blocos em outros projetos

3.4 Linguagens de Descrição de *Hardware*

HDLs (*Hardware Description Language*) como propriamente diz, são linguagens para descrever *hardware*, diferentemente de linguagens tradicionais como C que descrevem algoritmos implementados em software.

As linguagens de descrição de *hardware* possuem duas aplicações principais: simulação e síntese. Em simulação é aplicado estímulos que serão aplicados a um modelo descrito pela linguagem que expressará uma saída conforme o sistema responderá.

Basicamente, a simulação permitirá que seja compreendido como um modelo complexo se comporta antes de ser propriamente implementado.

A síntese cria uma solução refinada com comportamento equivalente às entradas e saídas do sistema, usando componentes que possuem implementação física (KUNDERTH, 2004).

Atualmente, há dois tipos de linguagem de descrição disponíveis para sinais mistos: Verilog-AMS e VHDL-AMS, que são extensões das tradicionais linguagens para circuitos digitais com suporte à modelagem de sistemas analógicos e sinais mistos.

3.5 Família de Linguagens Verilog

Verilog-AMS é uma linguagem de modelagem para sistemas de sinais mistos (KUNDERTH, 2004).

Kunderth define o Verilog-AMS como sendo essencialmente desenvolvido para assistir simulações de sistemas de sinais mistos a ser descrito para o simulador.

O termo sinais mistos sugere um sistema feito por partes de sinais digitais e outra analógica. O Verilog-AMS suporta a descrição dos dois sinais, sendo basicamente a união das linguagens de descrição; Verilog-HDL e Verilog-A. Essas três linguagens citadas dão origem à família de linguagens de descrição Verilog®. Dessa forma, como o Verilog-AMS faz parte da descrição para sinais tanto analógicos quanto digitais, o Verilog-HDL descreve sistemas digitais. Por outro lado, o Verilog-A descreve sistemas analógicos. Mas, além disso, o Verilog-AMS permite que sejam descritos componentes de sinais mistos.

Adicionalmente, para exemplificar a família de linguagens Verilog tem-se a Fig. (1).



Figura 1 – Relação entre Verilog-AMS, Verilog-HDL e Verilog-A (KUNDERTH, 2004)

Por fim, o Verilog-AMS é uma linguagem utilizada para simulação e verificação de sistemas, ou seja, a partir dela não será feita síntese.

3.6 Linguagem Verilog-AMS

Para o estudo da estrutura da linguagem é analisado o exemplo de um resistor linear proposto por Kunderth:

1. `'include "disciplines.vams"`
- 2.
3. `module resistor(p,n);`
4. `parameter real r=0; //resistance (Ohms)`
5. `inout p,n;`
6. `electrical p,n;`
- 7.
8. `analog`
9. `V(p,n) <+ r*I(p,n);`
10. `endmodule`

Tendo como base o exemplo de um resistor linear proposto por (KUNDERTH, 2004), o código em Verilog é estruturado da seguinte maneira:

Deve-se incluir *disciplines*, que são basicamente bibliotecas, contendo coleções de tipos de sinais físicos, como, por exemplo, tensão e corrente. Para que a linguagem seja capaz de descrever modelos que operem em sinais físicos, as *disciplines* associadas devem ser declaradas. Uma comum *discipline* utilizada é a *disciplines.vams*, que foi incluída, conforme linha 1, utilizando:

```
'include "disciplines.vams"
```

Nela é fornecido os nomes *electrical*, *V* e *I*, que são utilizados no código (linha 6).

O bloco básico de construção é o *module*, que é a descrição de componentes individuais, seguido pelo nome e sua lista de portas. como na linha 3. Ou seja, criando-se o módulo resistor com entradas/saídas *p* e *n*.

Na linha 4, `// resistance (Ohms)`, as duas barras indicam comentário. Tudo após `//` não influenciará no projeto. Outra forma que pode ser feito comentários é utilizando `/*` e `*/`, onde o conteúdo dentre eles será comentado.

Um parâmetro é criado utilizando o mnemônico *parameter*, assim como na linha 4. Quando dado o valor 0, significa que o parâmetro será inicializado com o valor "0". O tipo, nesse caso *real*, é opcional. Esse mnemônico *parameter* é importante para a realização de simulação devido à sua capacidade de ser alterado sem a necessidade de editar o código fonte. Isso permite que, quando trabalhado no simulador, poderá ser modificado a qualquer

momento que necessário, essencial para valores que necessitam constantes alterações, como características de dispositivos ligadas à resistência, capacitância, dentre outros.

Ports são os pontos que serão conectados o componente, nesse exemplo, os terminais do resistor, como observado nas linhas 5 e 6. A direção da porta é dada por **inout**, são três possíveis direções, **input**(entrada), **output**(saída) e bidirecional(**inout**). O tipo descrito é o *electrical*, significa que as portas associadas esperam ser tensão ou corrente.

O comportamento do bloco é definido em *analog* conforme linhas 8 e 9. *Analog* introduz um processo analógico que é usado para descrever um comportamento em tempo contínuo. Nesse caso, o valor de V em suas portas (p e n), recebe o valor da expressão à direita. A expressão é basicamente a relação entre tensão e corrente de um resistor linear ideal.

Por fim, para finalizar este módulo, deve-se fechá-lo com o uso de **endmodule**. Qualquer expressão que segue não estará associado com esse módulo resistor.

Para quaisquer outros dispositivos, deve-se checar a lista de descrição da linguagem. Por exemplo, ao modelar um capacitor, que tem relação de derivada com a tensão e corrente, ao escrever sua função referente usa-se “*ddt*” para expressar a derivada.

Para o desenvolvimento de sinais mistos segue-se da mesma estrutura. Uma alternativa para esse desenvolvimento é utilizar recursos da linguagem HDL. Assim, pode-se declarar uma rede digital utilizando expressão **wire**.

Wire é um tipo de rede digital. Ela pode carregar um bit de informação, que pode receber um de 4 valores mostrados na Tab. (3).

Nome	Descrição	Constante
0	Zero, Baixo ou Falso	0 ou 1'b0
1	Um, Alto ou Verdadeiro	1 ou 1'b1
x ou X	Desconhecido ou Não inicializado	1'bx
z ou Z	Alta Impedância (Flutuante)	1'bz

Tabela 3 – Valores Lógicos em Verilog (KUNDERTH, 2004).

Pode-se também declarar um vetor digital como por exemplo:

wire [7:0] dados

Nesse caso, está sendo declarada uma conexão de 8 bits, que podem ser acessados por dados[x], onde x varia de 0 a 7, que é a posição do bit.

Para transmitir um valor continuamente, utiliza-se a função *assign*.

assign q = ~a;

No caso anterior, é indicado que “q” receberá o inverso de “a” (devido ao uso de ~ após a igualdade), que sempre será alterado com qualquer mudança de “a”.

Com isso, tem-se a base para o desenvolvimento de blocos em linguagem de descrição Verilog-AMS. Contudo, em função da grande quantidade de expressões que a linguagem fornece, será necessário um intenso treinamento, para que seja possível aproveitar toda a versatilidade que o Verilog-AMS fornece.

4 Concepção do Sistema da *Tag* de RFID

Este capítulo apresentará um resumo sobre o que é o *front-end* analógico e os blocos que os constituem, assim como uma breve explicação de cada um no sistema. Além disso, serão expostos conceitos importantes para o planejamento da modelagem dos blocos da *tag* de RFID.

Dessa forma, os blocos do *front-end* analógico da *tag* passiva especificada são: o retificador de tensão, demodulador, gerador de clock e um modulador, conforme mostrado na Fig. (2)

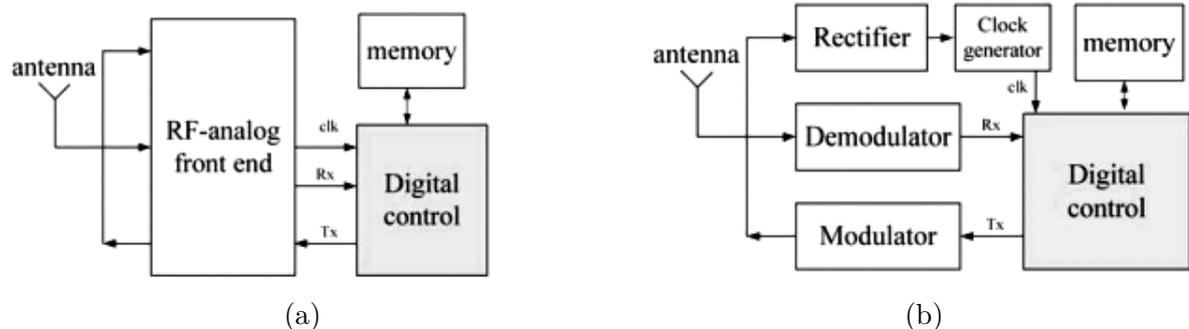


Figura 2 – (a) Denominação dos Blocos do *Tag* de RFID (b) Denominação dos Blocos Específicos do *Tag* de RFID (LI, 2009)

- Retificador: Esse bloco retifica a entrada do sinal de RF e gera o nível DC de tensão necessário para alimentar os demais blocos.
- Demodulador: O demodulador de onda ASK (*Amplitude Shift Keying*) utiliza, basicamente, um estágio de comparação que irá identificar a média do sinal retificado e comparar com o próprio sinal. Essa comparação fornecerá um sinal em nível digital.
- Controlador Lógico: É a parte digital do sistema que controla os demais blocos. Basicamente ele irá controlar quando a *tag* deverá receber e enviar o sinal. Armazena o ID do *tag* que será enviado ao leitor pelo modulador.
- Clock Interno: É um oscilador que gerará um *clock* para alimentar a parte digital do circuito.
- Modulador: O modulador do sistema é um Modulador BPSK (*Binary Phase Shift-keying*) que, a partir de um sinal digital de uma subportadora em BPSK, fará uma modulação ASK do sinal recebido, que o enviará para o *reader*.

4.1 Retificador

A maioria dos circuitos retificadores são baseados em diodos e capacitores (ASHRY; SHARAF; IBRAHIM, 2008).

A função desse bloco é transformar o sinal RF em um sinal DC para alimentar o restante do circuito. A onda AC do sinal de RF é amplificada e retificada para se atingir um nível de tensão que possa alimentar o circuito.

Para a alimentação da memória externa, a tensão deve ser elevada o suficiente, em torno de 5V , então utiliza-se um *charge pump* para atingir a tensão necessária.

Para alimentar os circuitos de RF é necessário uma tensão menor, abaixo de 2V ou 3,5V, dependendo da tecnologia, operando os circuitos em 1,8V ou 3,3V. Então é utilizado um regulador de tensão.

O regulador de tensão utilizado nesse caso é comumente um LDO (*Low-dropout Regulator*). Que para este projeto, receberá 3V que regulará para 1,8V.

De maneira geral, o sinal recebido pela antena é amplificado via *charge pump* para um nível mais alto de tensão, em seguida ele é retificado para se obter um nível DC que será regulado por um LDO.

4.2 Demodulador

O demodulador é o bloco responsável pela detecção dos dados enviados pelo *reader* para a *tag*. Devido a uma oportunidade de *tape-out*, juntamente com um projeto de iniciação científica (PINTO et al., 2014), esse bloco foi desenvolvido em nível de transistor e enviado para fabricação no dia 02 de Abril de 2014.

Esse demodulador projetado é um Demodulador ASK. O sinal ASK é uma contraparte digital do sinal AM (*Amplitude Modulation*). A característica do sinal ASK é mostrada na Fig. (3), onde a amplitude é chaveada no tempo, o que se tem é uma onda senoidal em meio período da subportadora e na outra metade tem-se o sinal nulo. Em sua estrutura ele possui três blocos gerais: Detector de envoltória, filtro de média e comparador. Na Fig. (4) é possível ver como esses blocos se compõem para formar a arquitetura completa do demodulador.



Figura 3 – Onda ASK (RAZAVI, 1998)

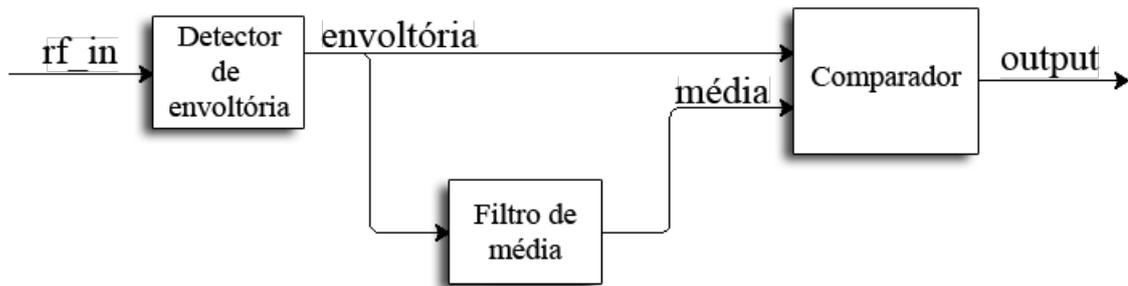


Figura 4 – Diagrama de Blocos do Demodulador ASK

O sinal recebido passa por um detector de envoltória projetado com capacitores e diodos. Na primeira parte do bloco têm-se uma multiplicação do sinal por um multiplicador de tensão. A sua saída recebe um filtro próximo à frequência do sinal (13,56 MHz), de forma que se obtenha o sinal de sua envoltória.

O sinal passa por um multiplicador de tensão para que seu nível DC seja elevado a um valor suficiente para sensibilizar as estruturas seguintes.

Foi desenvolvido um comparador com histerese para que se evite pequenos ruídos do sinal e tenha-se na saída a informação do sinal de entrada em níveis lógicos 1 e 0.

Os objetos de comparação são o sinal pós-envoltória e um sinal médio do sinal. Ou seja, a estrutura compara o sinal da envoltória com seu valor médio que é retirado por um filtro de média.

Dessa forma, como se compara um sinal com sua média, pode-se ver um sinal digital na saída do comparador, já que um comparador é basicamente um conversor analógico-digital de 1 bit. O que ocorre nessa etapa é transformar a faixa em que possui um sinal analógico senoidal em nível lógico alto e nas faixas em que o sinal tem amplitude nula, ou seja, que o valor do sinal seja um nível constante em zero, a saída terá nível lógico baixo.

4.3 Modulador

Dois tipos de moduladores podem ser utilizados em tags de RFID, o ASK e o PSK. A vantagem em se utilizar o PSK é a possibilidade de *full-duplex operation*, ou seja, pode operar para enviar o sinal para o leitor ao mesmo tempo que pode estar recebendo, o que não é necessário em aplicações de baixa taxa de transmissão, além de apresentar um menor consumo de energia.

4.3.1 PSK *Backscattering*

É desenvolvido para uma comunicação reversa, baseado na diferença de impedância entre a antena e o tag. Essa diferença resultará no retorno de uma fração do sinal que foi enviado. É transmitido um sinal do *tag* para o leitor na frequência da subportadora de 847,5kHz (ATMEL, 2005). *Backscattering* é um esquema de modulação de baixo consumo em que o tag age como um refletor, enviando parte da onda RF incidente de volta para o leitor (EPCGLOBAL, 2005). Na modulação PSK a impedância real do *tag* é casada com a impedância da antena, já a reatância varia entre valores capacitivos e indutivos, conforme a Fig. (5):

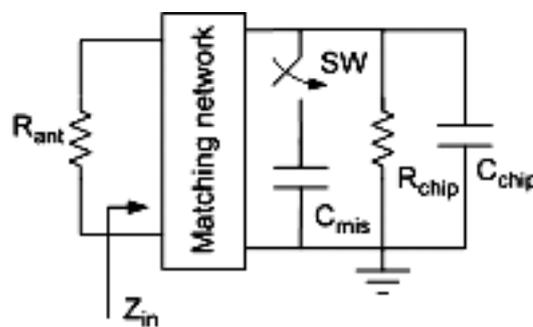


Figura 5 – Esquemático de Funcionamento do Circuito de *Backscattering* (ASHRY; SHARAF; IBRAHIM, 2009)

Com o chaveamento na saída a reatância altera-se, enviando um sinal refletido com o ID da *tag*.

O BPSK é a modulação digital com forma de onda da banda base binária. Nesse caso, PSK binário. No BPSK, a banda base binária seleciona uma das duas fases opostas da portadora (RAZAVI, 1998). Ou seja, o sinal PSK será chaveado entre dois valores de fase, $\phi = 0^\circ$ e 180° .

Esse sinal modulado em BPSK pode ser descrito como na Eq. (4.1), onde A_c é a amplitude do sinal, ω_c é a frequência angular do sinal e ϕ é a fase.

$$x_{BPSK} = A_c \cos(\omega_c t + \phi) \quad (4.1)$$

Como a subportadora, nesse caso, é um sinal digital, o que acontece é uma alteração na fase do sinal digital. Onde o sinal seria 1, torna-se 0, e vice-versa.

4.4 Oscilador

O oscilador fornece ao bloco digital o *clock* necessário para o funcionamento do mesmo. Existem diversos tipos de osciladores, dentre eles:

- *Voltage-Controlled Oscillator (VCO)*: Oscilador em que sua frequência de operação é controlada por um sinal de tensão.
- Oscilador em Anel: Oscilador baseado na instabilidade de portas lógicas inversoras conectadas em cascata, onde uma ao inverter o seu sinal tentará inverter a outra, como estão conectadas em anel (a saída é realimentada na entrada), isso se torna uma oscilação.
- Oscilador por Relaxamento (RC): Baseado em instabilidade da realimentação do sistema, onde há carga e descarga do capacitor, gerando uma oscilação.

O interessante ao se modelar em alto nível é que, independentemente da complexidade do circuito que será desenvolvido, o modelo deve ser capaz de fornecer o clock na frequência necessária em nível de sistema. O projeto de circuito é planejado e desenvolvido depois de validado.

5 Planejamento da Modelagem

Durante o processo de planejamento da modelagem, os blocos a serem desenvolvidos em nível de sistema são projetados de tal forma que os problemas que venham a ocorrer no projeto elétrico sejam previstos. Além disso, os blocos de interesse devem ser previamente definidos. Neste capítulo, será proposto o desenvolvimento de cada um. Em cada seção, serão destacados como será feita a modelagem do bloco e que características de cada um são necessárias para que a maioria dos problemas e situações sejam previstas durante as simulações. Sabendo disso, serão apresentadas as justificativas para as características extraídas para cada bloco a ser desenvolvido. Ao fim de cada planejamento, os pinos e suas descrições serão apresentadas.

5.1 Bloco Retificador

O bloco retificador é responsável por fornecer as tensões que alimentam os circuitos analógicos e digitais. Como o sistema se trata de uma *tag* passiva, a energia é produzida a partir do sinal de radiofrequência enviado do leitor para a *tag*. Sendo assim, ele possui um sistema composto de sub-blocos. Nesse contexto, o bloco principal é o LDO.

A Fig. (6) apresenta o esquemático de um LDO mais comum, utilizado para projetos de circuitos integrados.

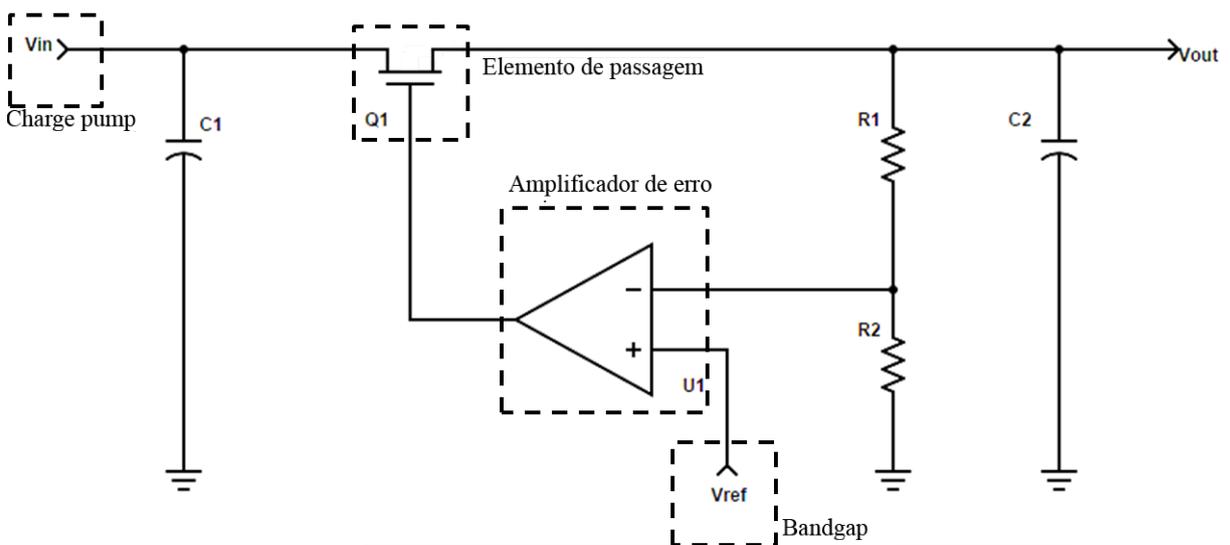


Figura 6 – Esquemático padrão para LDO

Então, para se evitar modelar uma fonte de tensão, o que não é possível prever os problemas que podem vir a ocorrer, é visto o sistema um nível abaixo, onde agora é

possível modelar os blocos fundamentais que constituem o sistema e, assim, prever uma simulação que se aproxime do bloco real.

Sendo assim, Os blocos destacados, com exceção do elemento de passagem Q1 que pode ser um transistor bipolar ou mosfet (tanto tipo n ou p, para os dois casos), serão os modelados em Verilog-AMS. O elemento Q1 opera na região linear para baixar a tensão de entrada para a tensão desejada. A saída é percebida pelo amplificador de erro que a compara com uma tensão de referência, o amplificador aciona o *gate* do transistor para o ponto de operação a fim de garantir que a saída esteja na tensão desejada. Ao passo que a tensão de entrada varia, o amplificador modula o elemento de forma que mantenha a saída constante (DAY, 2002).

V_{in} , a tensão de entrada, é uma tensão proveniente do *Charge Pump*. O amplificador U1 é chamado de *Error Amplifier* (Amplificador de erro), responsável por amplificar a diferença de tensão entre uma tensão V_{ref} de referência. Essa última, é modelada a partir de um *Bandgap Reference*, que é uma tensão invariante com a temperatura.

O *Charge Pump* providencia um nível DC maior ao circuito, isso é feito por dobradores de tensão. Uma topologia bastante utilizada é o dobrador de tensão Dickson. A vantagem desses tipos de circuitos está na capacidade de serem cascateados para elevar a tensão DC do circuito vários níveis.

Uma célula de dobrador de tensão é dado pela Eq. (5.1), onde V_p é a tensão de pico da onda e V_d a tensão de queda dos diodos que o compõe.

$$V_{DC} = 2(V_p - V_d) \quad (5.1)$$

O interessante dessa equação é que ela prevê a queda de tensão que os diodos virão a ter, e assim, o projetista poderá antecipar os problemas que isso poderá gerar no desenvolvimento do projeto elétrico desse bloco.

A Eq. (5.2), define o nível DC do circuito quando o mesmo é cascateado.

$$V_{n,DC} = 2n(V_p - V_d) \quad (5.2)$$

Onde n representa o número de estágios utilizados, podendo o projetista então, prever quantos estágios o elevador de tensão é composto. Desta forma, durante a etapa de projeto elétrico, tem-se a informação de quantos estágios o *Charge Pump* possui e o tipo de diodo a ser utilizado.

Essas equações, permitirão que um modelo em Verilog-AMS seja feito para um *Charge Pump*.

O retificador de onda, utilizado para se ter uma tensão de corrente contínua, é

comumente feito por um diodo e um filtro passa baixas, de forma que a onda seja o mais próximo de um valor constante possível. Ao modelar esse estágio, junto ao *Charge Pump*, obtêm-se a tensão V_{in} que irá para o LDO.

O amplificador de erro é um amplificador operacional convencional. Sendo assim, para prever a topologia a ser adotada e suas características para o projeto do circuito, alguns parâmetros devem ser modelados.

A Eq. (5.3) define o comportamento de um amplificador operacional ideal, onde a saída é uma entrada multiplicada por um ganho.

$$V_{out} = Av V_{in} \quad (5.3)$$

Porém, como dito anteriormente, isso não é suficiente para justificar uma modelagem em alto nível, visto que, isso não limitaria, por exemplo, a potência consumida por esse bloco. Sendo assim, os pontos a serem considerados para a modelagem do amplificador serão:

- Ganho;
- Potência;
- Saturação do sinal;

Com esses pontos, é possível que o projetista preveja a potência consumida, a saturação do sinal, caso ocorra, e o ganho que o amplificador deve possuir, em malha aberta, para que o circuito funcione de acordo com a modelagem em alto nível.

O bandgap é um circuito capaz de fornecer uma tensão como referência que não varie em condições usuais de temperatura. Ou seja, fornecerá um determinado valor de tensão independente das variações térmicas do sistema. Sendo assim, uma equação pode ser obtida para modelar o bloco em função da temperatura como variável para um determinado circuito. O circuito de base para o desenvolvimento deste bloco é apresentado na Fig. (7), e tem como característica a Eq. (5.4). Onde V_T é a tensão térmica e V_{BE} é a tensão entre base e emissor do transistor bipolar Q4. Essa tensão é variante na temperatura, na qual tem uma queda em função da temperatura.

$$V_{out1} = V_{BE4} + V_T \ln(n) \left(1 + \frac{R1}{R3} \right) \quad (5.4)$$

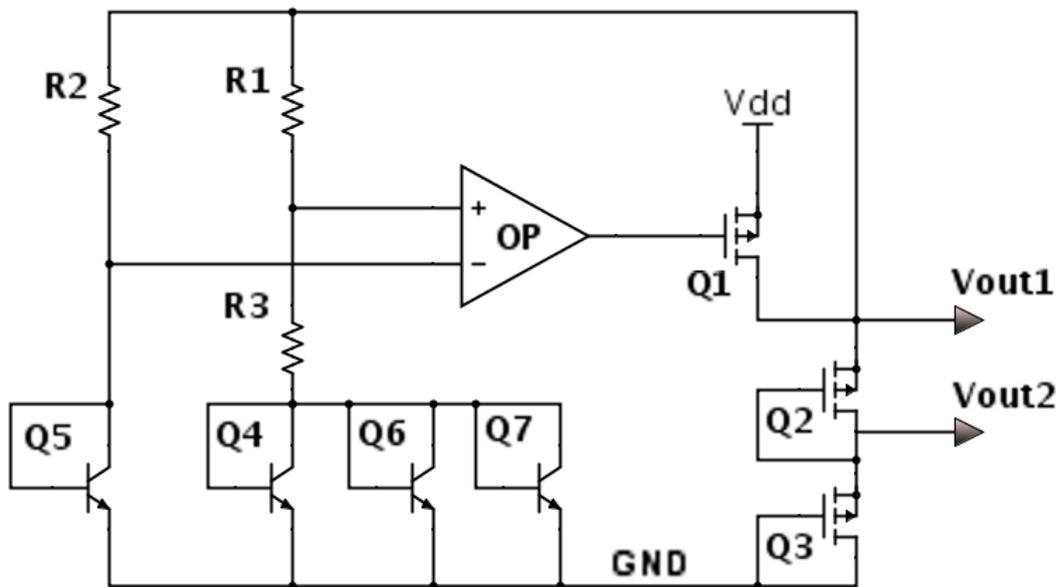


Figura 7 – Esquemático para Bandgap em Nível de Circuito (PINTO et al., 2014)

5.2 Bloco Modulador

O modulador BPSK, de maneira geral, é apresentado na Fig. (8), onde uma chave irá comutar de acordo com o ID da *tag*, variando a amplitude da onda refletida de acordo com um capacitor de descasamento.

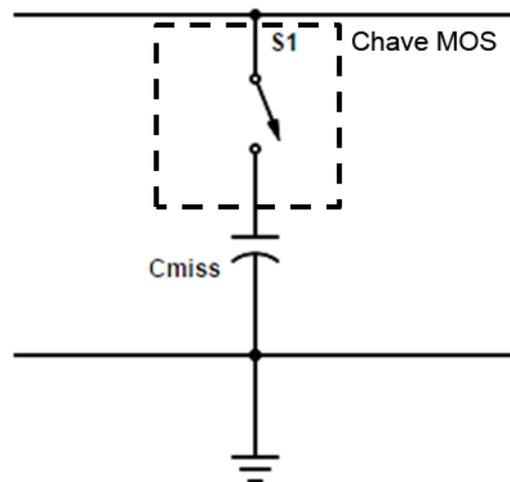


Figura 8 – Esquemático de um Modulador BPSK

Nesse caso, a chave utilizada é uma chave MOS, ou seja, uma chave projetada a partir de elementos MOSFET, então, transistores NMOS e PMOS são inseridos de forma que uma tensão no *gate* dos transistores, feche ou abra o circuito. Sabendo disso, modelar uma chave em Verilog-AMS deve levar em consideração as perdas e impedância da chave utilizada. Para isso, deve-se conhecer como é o funcionamento desse dispositivo.

Para ilustrar a impedância, é observado na Fig. (9) como o circuito se comporta quando a chave está ligada, ou seja, não há um circuito aberto nessa região. A resistência R_{on} representa essa impedância que a chave fornece ao trecho, observe que, quando R_{on} for nulo, trata-se de um curto-circuito, ou seja, a chave em questão seria ideal, sem perdas.

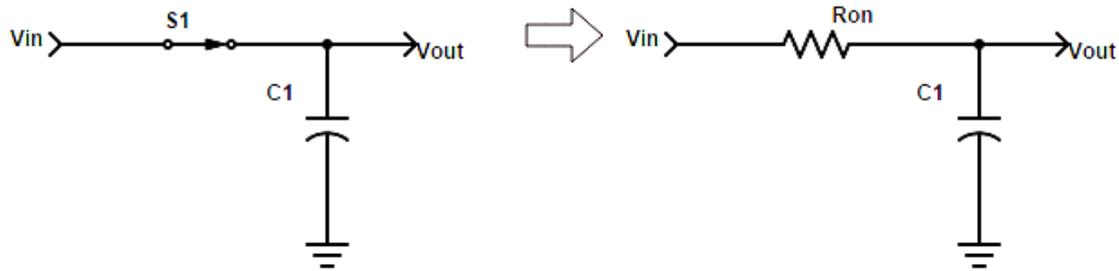


Figura 9 – Representação da impedância da chave quando fechada

A chave mais comum, em nível de transistor, é o próprio transistor CMOS, mas, devido às limitações de alcance dinâmico associadas às chaves de um único transistor (ALLEN; HOLBERG, 2002), pode ser feita a chave como mostrada na Fig. (10), que é chamada de chave complementar, que é construída conectando um transistor PMOS em paralelo com NMOS. Como ilustrado, CK e CK' irão abrir ou fechar a chave, quando CK em nível lógico 1, a chave está fechada e, em 0, aberta. CK' significa o inverso de CK, ou seja, ao implementar uma chave MOS em transistor, um inversor pode ser colocado, desde que essa chave não exerça a função de um dispositivo para *sample and hold* (amostragem) ou *switch capacitors*. Esses últimos necessitariam de *clocks* e ainda, para essas aplicações, o chamado *non overlapping clock*, que não será o caso para essa aplicação, podendo, na modelagem, prever a utilização de apenas um terminal de entrada.

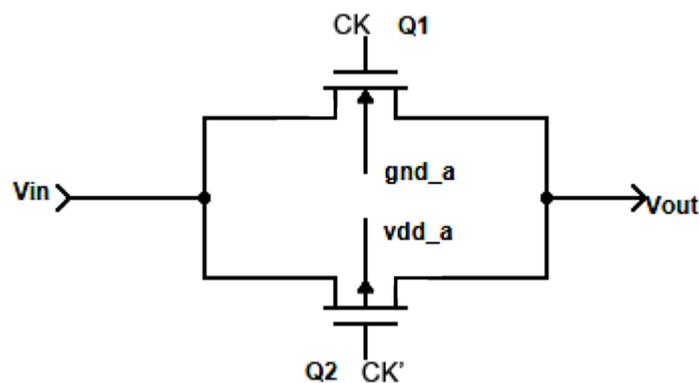


Figura 10 – Esquemático de uma chave MOS

Nesse caso, para o modulador, não é interessante modelar o descasamento da impedância, ou inserir um capacitor, isso é justificado pelo tipo de modulação feita. Por

ser um *backscattering*, o sinal recebido pela antena é refletido com uma subportadora de frequência diferente. Sendo assim, a simulação deve ser mista para que sejam alterados os parâmetros da rede de casamento, que fará essa função de *backscattering* da onda.

É esperado que, ao inserir um dispositivo na rede de casamento de impedância, essa impedância casada com a antena seja descasada, o que fará que o sinal refletido sofra alteração em sua amplitude. Deve ser refletida uma onda ASK com perda de amplitude de 10%. Esse cálculo é realizado a partir da Fig. (11) e a Tabela (4), onde se mostra a margem de quanto essa onda deve ser modulada (8% a 14%). Essa atenuação do sinal, que caracterizará numa onda ASK refletida, é feita a partir da subportadora de 847,5 kHz modulada em BPSK. Esse sinal recebido da subportadora digital chaveará o descasamento de impedância, isso justifica a preocupação apenas com a chave, já que a modulação BPSK vem do bloco digital.

Os percentuais de modulação são calculados a partir das Eq. (5.5) e (5.6), onde A é a amplitude do sinal não-modulado e B a amplitude do sinal modulado. (ATMEL, 2005).

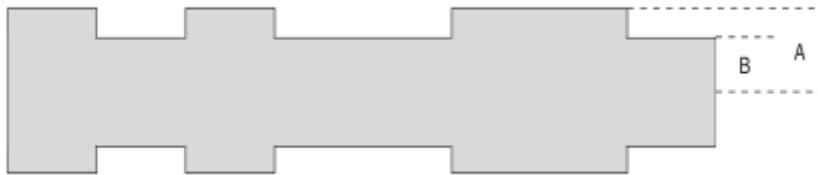


Figura 11 – Amplitude da Onda Refletida (ATMEL, 2005)

$$M_I = \frac{A - B}{A + B} \quad (5.5)$$

$$M_D = \frac{B}{A} \quad (5.6)$$

Índice de Modulação (M_I)	Profundidade da Modulação (M_D)
8%	85,2%
9%	83,5%
10%	81,8%
11%	80,2%
12%	78,6%
13%	77,0%
14%	75,4%

Tabela 4 – Relação entre o índice de modulação com a profundidade (ATMEL, 2005)

Para entender o motivo de ser 847,5 kHz a frequência da subportadora, observe-se o sinal recebido, a uma frequência de 13,56MHz, que ao se dividir por 16, tem-se a

subportadora, e isso é caracterizado por 8 bits (considerando cada bit um momento que a onda está em 1 ou 0). Ou seja, um período de bit possui oito ciclos da subportadora. Essa é modulada a partir de uma codificação, indicando o ID da *tag*, e que, cada nível lógico caracteriza uma fase dessa subportadora digital, como observado na Fig. (12).

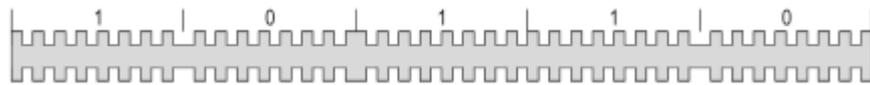


Figura 12 – Modulação BPSK da Subportadora de 847,5 kHz (ATMEL, 2005)

Isso é feito ao ajustar ou desajustar o circuito de ressonância da antena que, no caso, é um circuito externo. Uma das configurações desse circuito é apresentado na Fig. (13), onde a chave é inserida entre os pontos "Ant. Pad B" e Vss, de forma que sintonize a frequência em 13,56MHz quando fechada, ao abrir, essa frequência é desajustada em uma frequência próxima do sinal. Esse circuito será utilizado para os testes do modulador.

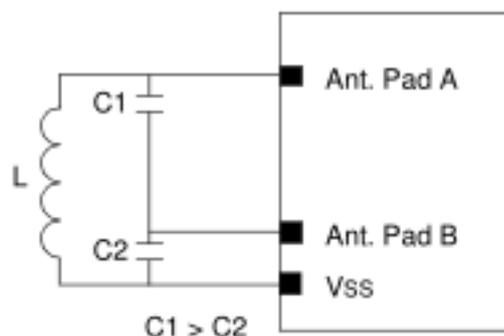


Figura 13 – Circuito externo da antena (MICROCHIP, 2004)

5.3 Bloco Demodulador

O bloco do demodulador ASK é composto de três elementos, que serão modelados em Verilog-AMS, são eles:

- Detector de Envoltória
- Filtro de Média
- Comparador com Histerese

O detector de envoltória é um bloco semelhante ao apresentado no bloco do refitificador, onde é feito um multiplicador de tensão Dickson e um retificador com diodo para realizar a detecção de envoltória. Sendo assim, o mesmo bloco pode ser utilizado nessa etapa, tendo apenas os parâmetros de simulação adaptados para essa função.

O filtro de média, nada mais é que detectar o ponto médio da envoltória. Esse sinal terá um valor máximo e mínimo em torno do nível DC, de característica simétrica, ou seja, desconsiderando o nível DC, o módulo do valor máximo da onda é muito próximo do módulo do valor mínimo da onda. Ou seja, esse bloco tende a identificar o valor do nível DC do sinal de envoltória (Equação 5.7). Contudo, esses dois blocos são filtros que poderão ser modelados em Verilog-AMS.

$$V_{m\acute{e}dia} = \frac{V_{max} + V_{min}}{2} = V_{DC} \quad (5.7)$$

O comparador com histerese terá os parâmetros de histerese como ponto chave da modelagem, o modelo deve ser capaz de manipular os valores de histerese e tempo de transição do sinal entre os valores digitais como pontos críticos do bloco. O sistema como um todo será apresentado em nível de transistor no Capítulo 5.

5.4 Bloco Oscilador

O oscilador é responsável por gerar o *clock* necessário para o módulo digital do sistema. O oscilador utilizado para a *tag* é um oscilador em anel, conforme a Fig. (14).

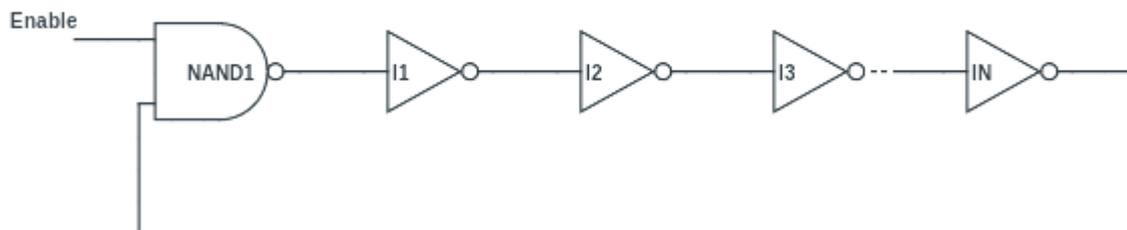


Figura 14 – Esquemático de um oscilador em anel padrão

Há duas maneiras interessantes de se modelar um oscilador em anel em Verilog-AMS. A primeira é modelar os blocos individuais do subsistema, ou seja, modelar um inversor e uma porta lógica NAND. A segunda, modelar o oscilador completo. Ao projetar os inversores, seu tempo de resposta deveria ser considerado e, com esse modelo, cascadeá-lo de forma que se crie um oscilador. Entretanto, há uma equação característica para osciladores em anel. A Eq. (5.8) representa o período de oscilação.

$$T_{clock} = 2 t_d N \quad (5.8)$$

Com isso é possível obter a frequência de oscilação necessária a partir do número de estágios N que o *clock* possuirá. Onde t_d representa o tempo de propagação de um

inversor simples. Sabendo que a frequência é o inverso do período da onda obtém-se a Eq. (5.9).

$$f_{clock} = \frac{1}{2 t_d N} \quad (5.9)$$

Com essa representação, como há o *delay* (t_d) em cada estágio unitário de inversor, é possível gerar um *clock* que dependa do número de estágios e da porta NOT. Ou seja, não é necessário modelar em Verilog-AMS a porta inversora, podendo projetar o bloco completo e, o projetista saberá, durante as simulações, o inversor a ser utilizado e sua quantidade.

A porta NAND, é utilizada como uma chave de *enable* (habilitar). Quando a porta é ativada em nível lógico 1, o oscilador inicia.

5.5 Planejamento das Simulações Mistas

O objetivo do trabalho é desenvolver os blocos do *front end* analógico do tag de RFID em Verilog-AMS, o que faz com que não seja abordado todas as etapas da metodologia *Top-Down* de forma rígida como: verificação *Bottom-Up*, verificação final e teste, focando-se no planejamento da modelagem e verificação a nível de sistema. Apesar de que os blocos serão modelados em Verilog-AMS, com o projeto em nível de transistor do demodulador ASK, este fará parte das simulações mistas, fazendo com que seja necessário também este planejamento.

Na primeira etapa da modelagem em alto nível se tem o planejamento da modelagem e das simulações, deve-se identificar quais blocos estão em nível de transistor e analisar os blocos a nível de sistema. Como foi projetado o bloco do demodulador ASK em nível de circuito e se tem validado seu comportamento, seus resultados entrarão como importante ponto para seu projeto.

O circuito do LDO será projetado com elementos já em nível de circuito utilizando a tecnologia TSMC 180nm, como citado anteriormente, e os seus módulos individuais serão em Verilog-AMS, caracterizando uma simulação com componentes em alto nível e em nível elétrico.

O circuito do demodulador ASK, inicialmente será completamente simulado em Verilog-AMS, alguns blocos serão substituídos para seus equivalentes em nível de transistor e, por fim, apresentado os resultados em nível de circuito por inteiro. Apresentando, portanto, algumas simulações mistas desse bloco.

Serão feitas 4 etapas de simulação para o demodulador, sendo a primeira os blocos completos em Verilog-AMS, posteriormente, será substituído os blocos de detector de envoltória para nível de transistor e simulado, em seguida, mantém-se apenas o comparador

com histerese em nível de sistema, e por fim, o circuito completo em nível de circuito e comparado os resultados obtidos. Tendo, apenas para esse bloco, a metodologia *Top-Down* por completo.

Ao fim dessa etapa, deve-se obter diversos blocos que poderão ser utilizados posteriormente em projetos em nível de transistor com qualquer tipo de tecnologia e ainda integrar o sistema como um todo, que não necessariamente, deve ser parte de uma *tag* de RFID.

5.6 Descrição dos Pinos

Durante o planejamento da modelagem é importante que os pinos dos blocos sejam previstos, isso se deve à necessidade dos projetistas de saberem os sinais de entrada e saída do bloco para que, quando o modelo em Verilog-AMS for substituído pelo modelo em nível de circuito, os mesmos pinos sejam utilizados, de forma que apenas sejam substituídos os modelos sem que sejam criadas portas não planejadas. Sendo assim, esse tópico irá apresentar os pinos dos blocos a serem modelados e suas respectivas descrições em forma de tabela. Esse formato simplifica a visualização das entradas e saídas do sistema, favorecendo o trabalho dos projetistas de sistema e de circuito. As Tabelas (5) a (12) descrevem os pinos utilizados em cada bloco.

Pino	Descrição
avdd	Alimentação positiva do circuito
agnd	Alimentação Negativa do circuito (<i>ground</i>)
in_p	Entrada Positiva do Ampop
in_n	Entrada Negativa do Ampop
vout	Pino de Saída do Ampop

Tabela 5 – Descrição dos Pinos do Amplificador Operacional

Pino	Descrição
avdd	Alimentação positiva do circuito
agnd	Alimentação negativa do circuito (<i>ground</i>)
vout	Tensão de referência (saída)

Tabela 6 – Descrição dos Pinos do Bandgap

Pino	Descrição
rf_in	Sinal de RF recebido pela antena
agnd	Tensão de alimentação negativa do circuito (<i>ground</i>)
out	Saída de sinal retificado do Charge Pump

Tabela 7 – Descrição dos Pinos do Charge Pump

Pinos	Descrição
avdd	Tensão de alimentação positiva do circuito
agnd	Tensão de alimentação negativa do circuito (<i>ground</i>)
key	Sinal que abre ou fecha a chave
v_in	Terminal de entrada da chave
v_out	Terminal de saída da chave

Tabela 8 – Descrição dos Pinos da Chave

Pinos	Descrição
avdd	Tensão de alimentação positiva do circuito
agnd	Tensão de alimentação negativa do circuito (<i>ground</i>)
ibias	Fonte de corrente externa para o circuito
vin_p	Entrada positiva do comparador
vin_n	Entrada negativa do comparador
vout	Saída do comparador

Tabela 9 – Descrição dos Pinos do Comparador

Pinos	Descrição
rf_in	Sinal de RF recebido pela antena
avdd	Tensão de alimentação positiva do circuito
agnd	Tensão de alimentação negativa do circuito (<i>ground</i>)
out	Saída do detector de envoltória

Tabela 10 – Descrição dos Pinos do Detector de Envoltória

Pinos	Descrição
rf_in	Sinal de RF recebido pelo detector de envoltória
avdd	Tensão de alimentação positiva do circuito
agnd	Tensão de alimentação negativa do circuito (<i>ground</i>)
out	Saída do filtro de média

Tabela 11 – Descrição dos Pinos do Filtro de Média

Pinos	Descrição
avdd	Tensão de alimentação positiva do circuito
agnd	Tensão de alimentação negativa do circuito (<i>ground</i>)
enable	Sinal que habilita o oscilador
saida	Saída do oscilador com o clock

Tabela 12 – Descrição dos Pinos do Oscilador

6 Projeto e Modelagem do Sistema

Neste capítulo será apresentado a modelagem em Verilog-AMS dos circuitos propostos e o projeto do demodulador ASK em nível de transistor, assim como seus resultados obtidos.

O sistema é um *front-end* analógico de uma *tag* passiva de RFID baseada na ISO/IEC 14443 (ATMEL, 2005). O sinal de entrada do sistema é uma onda ASK com frequência de 13,56 MHz e o sinal refletido pelo modulador é um ASK com subportadora de frequência 847,5 kHz modulada em BPSK. Devido à tecnologia utilizada (TSMC 180 nm), a tensão de alimentação dos blocos de RF é 1,8 V. São consumidos $5\mu\text{A}$ de corrente pelo bloco demodulador com uma potência total de $9\mu\text{W}$. O oscilador deve gerar um *clock* para a parte digital entre 1 MHz e 2,2 MHz.

6.1 Amplificador Operacional

Para modelar o amplificador operacional, é importante destacar sua saturação e ganho. Para o ganho em malha aberta do amplificador, o parâmetro foi definido em dB – caso mais usual para definir ganhos de amplificadores – esse ganho é transformado para escala absoluta como descrito na Eq. (6.1).

$$Ganho = 10^{\frac{Ganho[dB]}{20}} \quad (6.1)$$

Desta forma, a saída do amplificador operacional é o ganho absoluto multiplicado pela diferença entre suas entradas. E pode ser observado no código, logo na declaração das variáveis de projeto, conforme linha 13, onde é uma variável *parameter real*, que apesar de possuir um valor de projeto de 60dB como *default*, pode ser alterado no ambiente de simulação. Quando declarado a variável interna de ganho absoluto, é realizado a conversão de medidas, como mostra a linha 17. O trecho do código em que isso acontece é apresentado a seguir:

```

13. parameter real gain = 60;
14. parameter real pot = 5e-6;
15.
16. //variaveis locais
17. real abs_gain = pow(10, gain/20);

```

Dado sua saída, caso seja maior que as tensões de alimentação, o mesmo deve saturar e, para isso, foi utilizado como ferramenta, a função tangente hiperbólica. Essa função tem um comportamento em que $f(x)$ possui domínio entre -1 e $+1$, o que pode ser manipulado a fim de arbitrar esse domínio, ou seja, se multiplica-se a função $\tanh(x)$ por uma constante A , seu domínio estará entre $-A$ e A . Entretanto, isso garante que a função sature simetricamente, o que não é interessante para o amplificador já que deseja-se saturações diferentes para níveis de alimentação, que nem sempre serão simétricas. Para isso, é possível manipular a tangente hiperbólica de tal forma que haja uma região de 0 a 1 no domínio negativo e positivo. Ou seja, ao invés de uma única função em -1 e $+1$, o que se obtém é uma função em -1 e 0 e outra em 0 e $+1$ (Figura 15). Assim, ao somar as duas tangentes, satura-se em dois níveis distintos de alimentação.

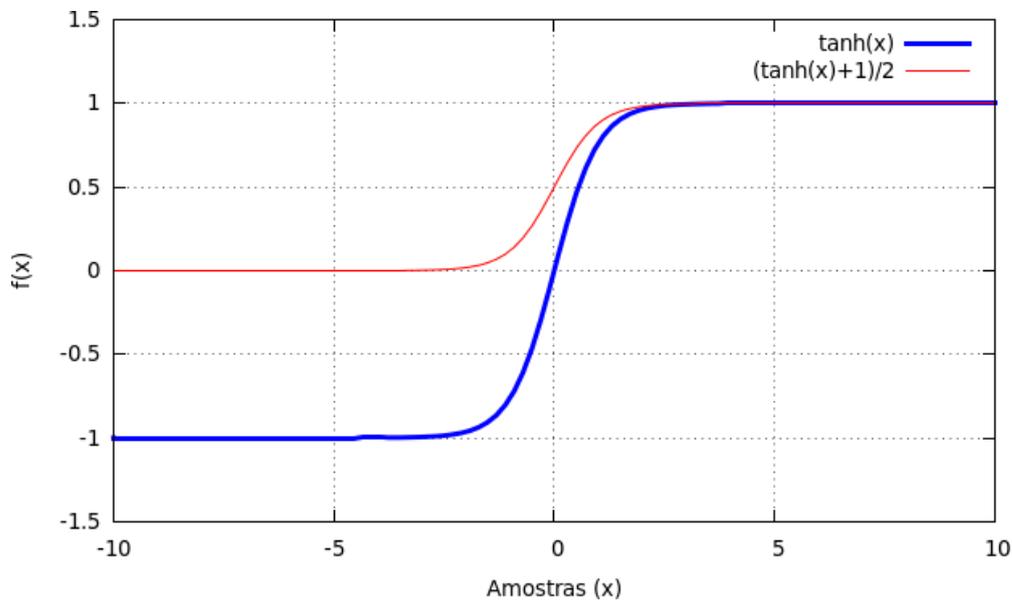


Figura 15 – Tangente Hiperbólica Deslocada em $+1$

A ideia dessa manipulação é simular uma função degrau em que pode-se eliminar a parte positiva ou a negativa, obtê-las separadamente e depois soma-las, de forma que se altere apenas a saturação, já que o Verilog-AMS não possui uma função degrau em sua linguagem matemática. Essa função simulando uma degrau pode ser observada conforme Fig. (16), multiplicando-a pela $\tanh(x)$. Porém, há uma pequena região em que a função ainda admite valores na região negativa, o que pode ser contornado multiplicando-a por um valor alto, tornando uma transição mais abrupta (Figura 16), o valor utilizado foi de 10 e pode-se observar uma maior estabilidade em 0.

Com essas manipulações de multiplicar uma tangente simulando uma degrau, pode-se somar as regiões de deslocamento da função positivas e negativas, conforme Eq. (6.2). O resultado aparenta ter apenas uma região positiva mas, note que, uma das alimentações será idealmente nula ou negativa, fazendo com que ela seja jogada para baixo do eixo $y = 0$. E ainda, se for inserida uma alimentação positiva, se manterá no eixo

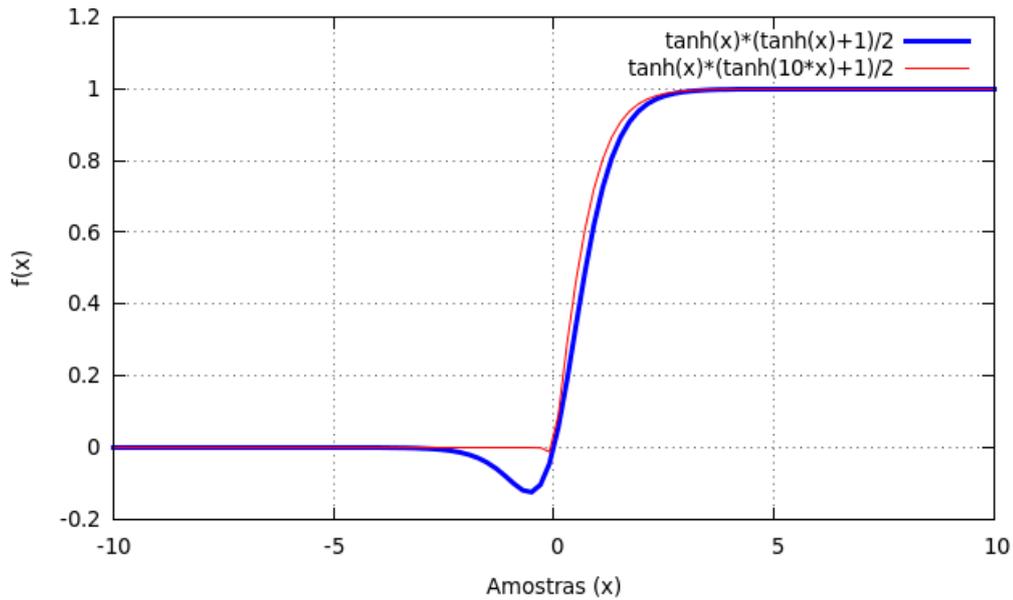


Figura 16 – Tangente Hiperbólica Multiplicada pela sua Componente Deslocada em +1 positivo, assim como pode ser visto na Fig. (17), com uma alimentação de +1,8V e -1,3V.

$$f(x) = V_+ \tanh(x) \frac{\tanh(10x + 1)}{2} + V_- \tanh(x) \frac{\tanh(10x - 1)}{2} \quad (6.2)$$

Como dito anteriormente, foi necessário multiplicar a função por uma constante para que fossem eliminados componentes na região próxima à $f(x) = 0$, Quando na alimentação negativa for colocado uma tensão maior que 0, essa componente volta a aparecer, conforme visto pelo gráfico da função na Fig. (17). Supõe-se que nessa alimentação seja inserido valores negativos ou o terra. Sendo assim, uma importante observação para esse modelo, já que antes de saturar na região inferior, caso haja uma tensão maior que zero, haverá um pequena região que não saturará quando devia. Então o modelo em Verilog-AMS possui a função descrita acima (Equação 6.2) para saturar seu sinal de saída.

O processo analógico do amplificador operacional é muito importante para o modelo, tendo em vista que é nesse processo em que essa equação de saturação é inserida, conforme o trecho do código em Verilog-AMS a seguir:

```

23. analog begin
24.
25. aux_out = abs_gain*(V(in_p)-V(in_n));
26.
27. //parte positiva
28.
29. aux_pos = V(avdd)*tanh(aux_out/V(avdd))*(tanh(10*aux_out/V(avdd))+1)/2;

```

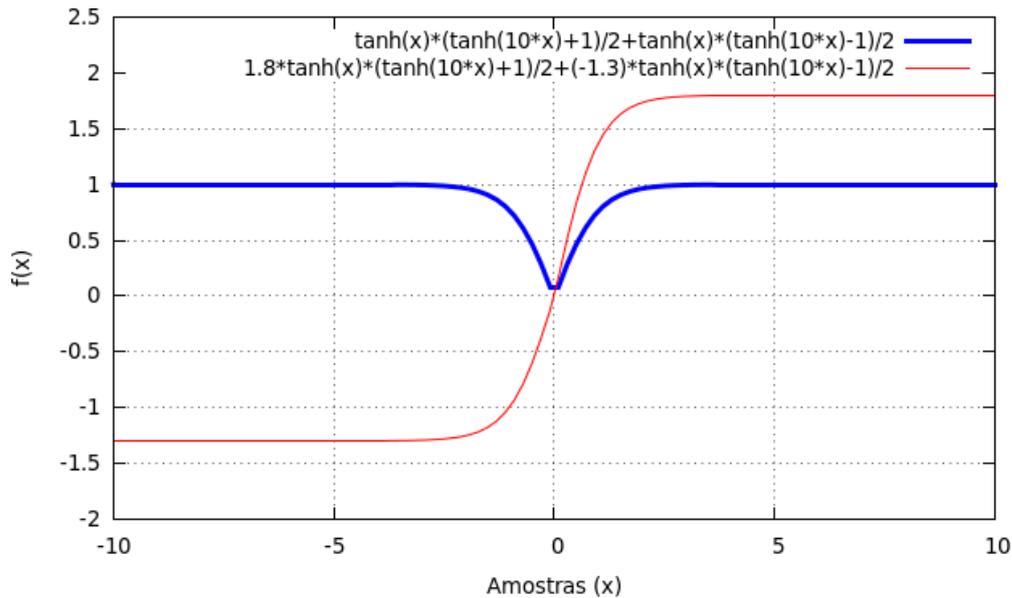


Figura 17 – Função Utilizada para Saturação do Amplificador com Exemplo

30.

31. //parte negativa

32.

33. aux_neg = V(agnd)*tanh(aux_out/V(avdd))*(tanh(10*aux_out/V(avdd))-1)/2;

34.

35. V(vout) <+ aux_pos+aux_neg;

36. I(avdd, agnd) <+ pot;

37.

38. end

A descrição foi dividida em duas partes, uma positiva e outra negativa, com uma variável auxiliar interna ao projeto, chamada `aux_pos` para a parte positiva, e `aux_neg` para a parte negativa, ambas de parâmetro *real*, fazendo com que receba um valor, então, com o cálculo realizado, ambas são somadas e convertidas em tensão para o pino de saída `vout`, conforme linha 35. A manipulação das tangentes hiperbólicas podem ser observadas nas linhas 29 e 33. E ainda, na linha 36 é calculada a potência consumida pelo sistema – corrente entre `avdd` e `agnd`.

Os resultados obtidos pelo modelo podem ser observados a seguir, onde, para avaliar o ganho, foi simulado um amplificador operacional com realimentação negativa com resistores ajustados de tal forma que, um ganho de -2 V/V fosse obtido, e um sinal de entrada aplicado suficiente para que não sature para uma alimentação de 0V a 1.6 . Conforme observado na Fig. (18). O sinal obtido foi a entrada, invertida, com as tensões

de vale e pico com o dobro do valor. Em -300mV a saída obtida foi 599mV e para -699mV , $1,39\text{V}$, caracterizando o ganho de -2 V/V .

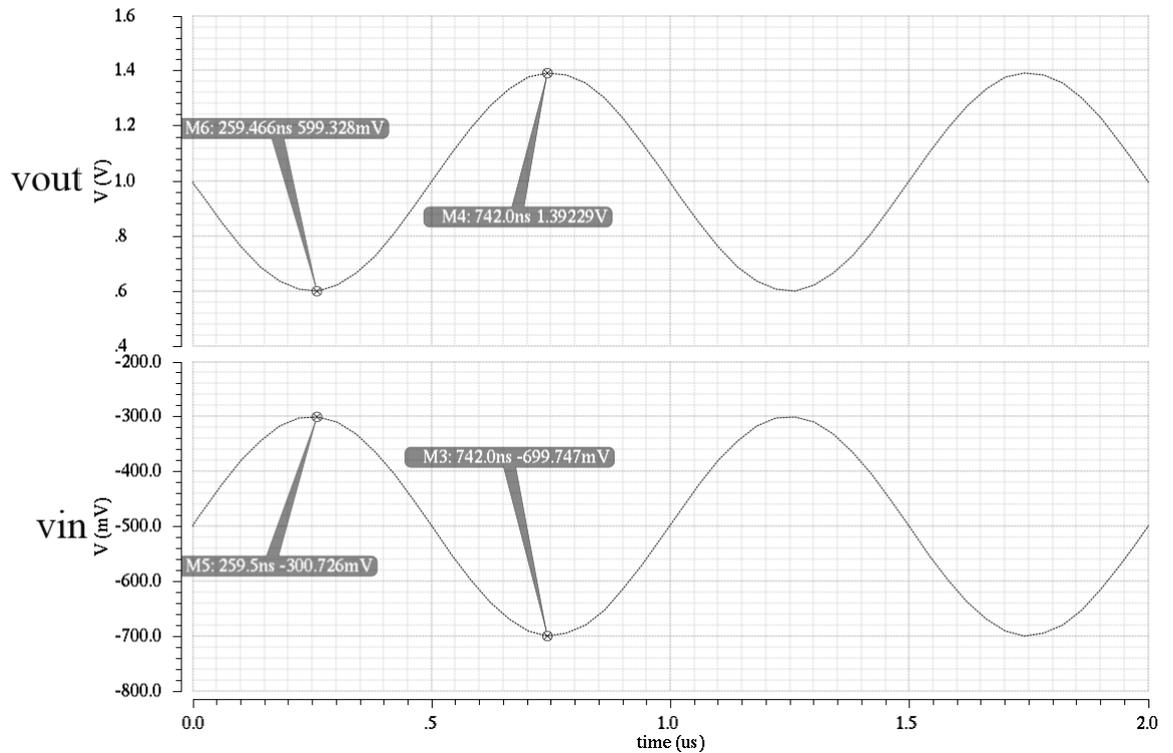


Figura 18 – Transiente do Modelo do Amplificador Operacional para uma Realimentação de Ganho -2 V/V

Para simular a saturação do sinal, foi inserido um sinal de entrada mais alto que, com o mesmo ganho, ultrapassasse os níveis de alimentação. Conforme a Fig. (19) o foi inserido uma alimentação positiva de $1,6\text{V}$ e 4 casos de alimentação negativa, $-0,2\text{V}$; 0V ; $0,1\text{V}$ e $0,2\text{V}$. Conforme previsto, o sinal satura para valores negativos e nulo, porém quando alimentado com uma tensão positiva, o sinal não satura imediatamente, que seria a influência da manipulação das tangentes hiperbólicas para o problema. Entretanto, o modelo foi projetado para alimentações usuais de valores negativos ou terra. Tornando um bom modelo de amplificador operacional para alimentações não simétricas.

O amplificador foi modelado de duas formas, um para excursão ideal, chamada *rail to rail* (que é possível se projetar a nível de circuito, onde não considera-se quedas dos transistores saturando o sinal antes da alimentação, e outro no qual há um parâmetro para prever esse caso (simulado acima, observado nas Fig. (18) e (19)). Para isso, altera-se as linhas 29 e 33 substituindo $V(\text{avdd})$ e $V(\text{agnd})$ para um $Vt+$ e $Vt-$ (tensão de limiar positivo e negativo, respectivamente), ou os subtrai para um valor de queda de tensão, realizando assim, o modelo que considera essa particularidade.

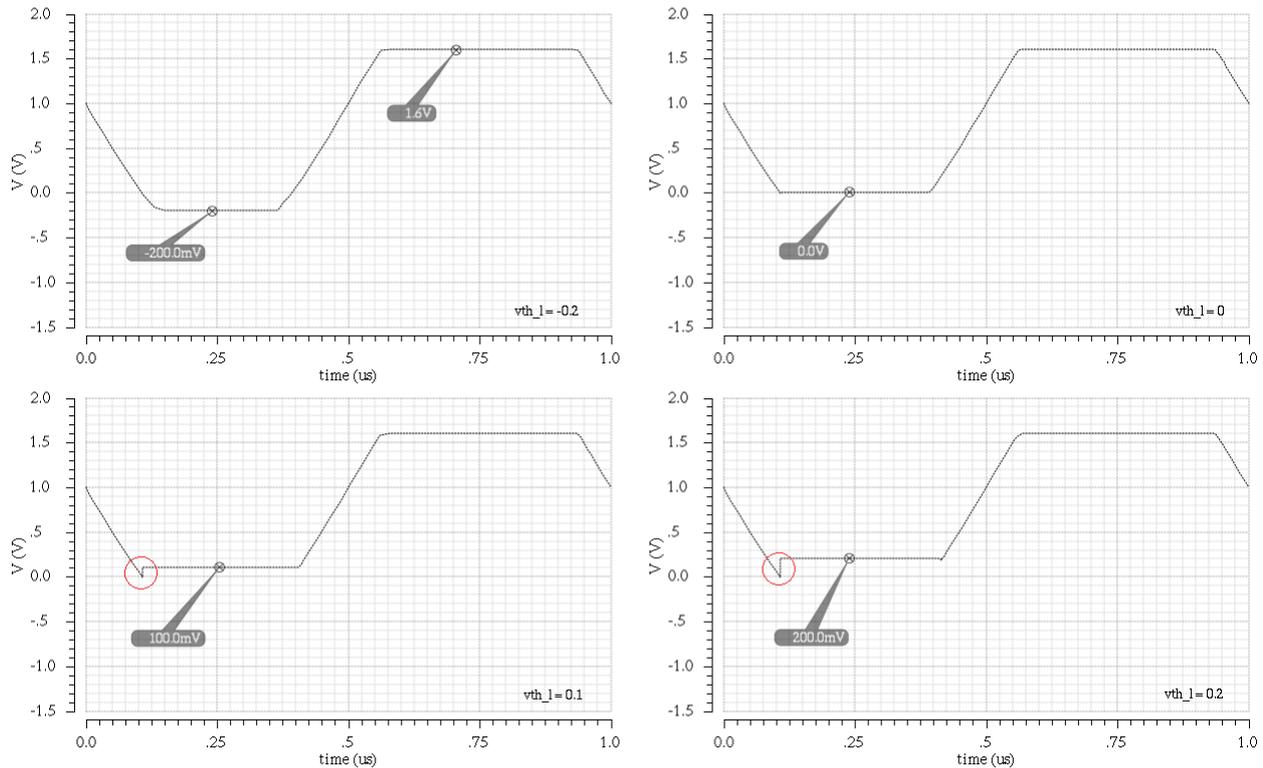


Figura 19 – Comparativo entre Simulações Transiente para Diferentes Tensões de Saturação

6.2 Charge Pump

O modelo do *Charge Pump* deve ser capaz de subir a tensão de RF que será recebido pela antena, e, para isso, é utilizada sua equação característica, e o importante nesse caso, é desenvolver um algoritmo que seja capaz de encontrar a tensão de pico do sinal recebido para que seja inserido na equação. Isso é realizado no processo analógico do modelo, conforme trecho do código a seguir:

```

27. analog begin
28.
29. @(initial_step) vpeak = 0;
30.
31. if (V(rf_in) > vpeak) vpeak = V(rf_in);
32. if (V(rf_in) < 0.05) vpeak = 0;
33.
34. V(out) <+ ((rect_mod)*V(rf_in))+((2*stages*(transition(vpeak, 0, tt, tf)-vd)));
35.
36. end

```

Nas linhas 31 e 32 é avaliado qual a tensão de pico do sinal recebido, como passo inicial, $vpeak$ (variável local para armazenar o valor da tensão de pico) recebe zero (linha 29), e assim, compara-se o sinal de entrada com $vpeak$, caso seja maior, $vpeak$ recebe a tensão RF de entrada. Em seguida, faz-se outra comparação, caso seja menor que um limiar próximo a zero, $vpeak$ é zerado, para que sempre avalie a tensão de pico. (valor arbitrário definido, já que se tem um prévio conhecimento do sinal que será recebido).

Com $vpeak$ avaliado, é inserido na equação característica do *Charge Pump*, como na linha 34, nesse caso, $vpeak$ é inserido na função *transition* para simular uma descarga do sistema proveniente da tensão de entrada indo para zero. Essa função *transition* faz com que, quando houver uma transição da variável para outro valor, ela não transitará abruptamente, haverá um tempo para a realização dessa transição, determinados por tt (variável utilizada para o tempo de subida), e tf (variável utilizada para o tempo de descida). De valores diferentes, é capaz de subir mais rápido que descarregar, porém, parâmetros que podem ser alterados no ambiente. Por fim, $rect_mod$ significa um termo para que haja atenuação do sinal arbitrária, decidida ao simular, como padrão é definida como zero, prevendo um retificador ideal.

Os resultados de simulação obtidos pelo *Charge Pump* são observados na Fig. (20). Onde é possível observar a tentativa do circuito de eventualmente descarregar, principalmente quando a tensão de entrada vai à zero.

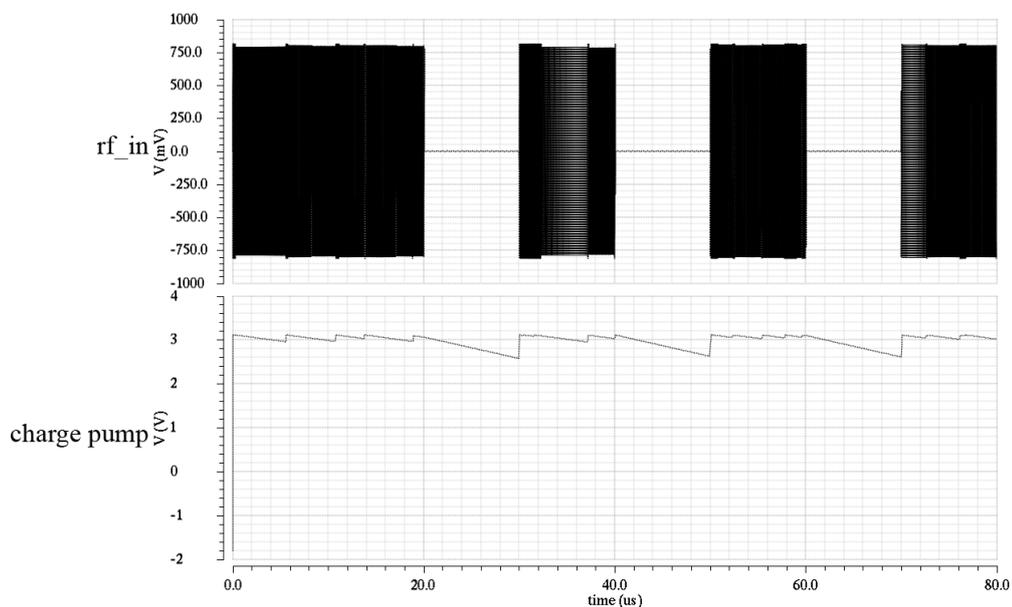


Figura 20 – Transiente do Charge Pump

6.3 Bandgap

O bandgap é um circuito que depende da temperatura, para isso, a linguagem Verilog-AMS possui funções na qual é possível extrair parâmetros de temperatura do

circuito que estão sendo utilizadas no simulador. O modelo se baseia, basicamente, na equação característica do circuito, e na potência que o mesmo consome da fonte. O trecho do processo analógico é apresentado a seguir:

```

20. analog begin
21. V(vout) <+ (Vbe-(2e-3*(temperature-273.15))) + $vt*ln(n)*(1+(R1/R2));
22.
23. I(avdd, agnd) <+ pot;
24.
25. end

```

A linha 21 do código apresenta a equação característica. Inicialmente, considerando que V_{be} tenha uma queda de tensão de 2mV por °C, então tem-se que $(V_{be}-(2e-3*(temperature-273.15)))$. V_{be} nesse caso está em função de $temperature$, este recebe a temperatura do ambiente de simulação em Kelvin, portanto, é transformada para escala Celsius. E, como há a tensão térmica na equação do bandgap, ela é inserida pelo termo $\$vt$, esse termo é uma característica da linguagem Verilog-AMS que calcula a tensão térmica a partir da temperatura do sistema simulado. Assim, foi possível gerar um modelo no qual dependa da temperatura e por fim, na linha 24 é inserido a corrente consumida pelo bloco.

Como o simulador não permite simulação DC para curvas de temperatura, a função foi plotada matematicamente e o resultado é observado na Fig. (21).

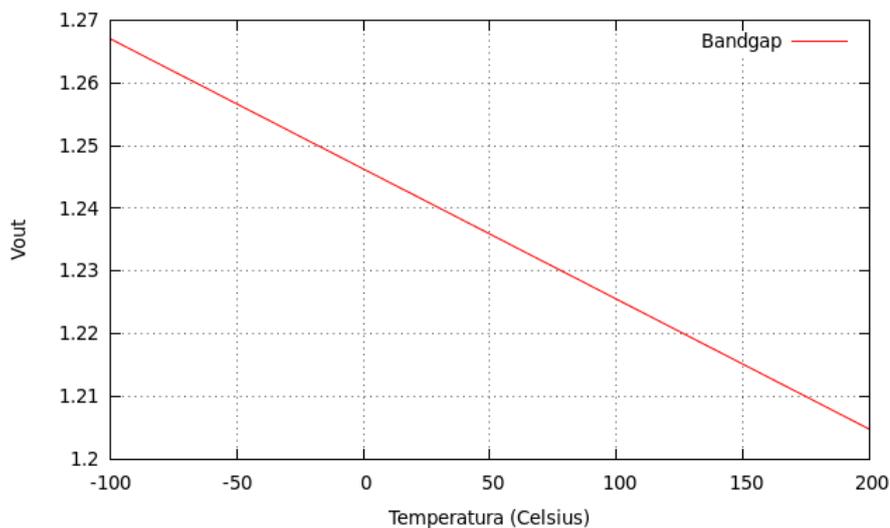


Figura 21 – Gráfico da resposta matemática da função do bandgap

Apesar de analiticamente o resultado obtido não se comportar como uma parábola característica de circuitos como esse, no simulador, obteve uma resposta bem regular,

constante e próximo à 1,22V numa longa faixa de temperaturas. (valor arbitrário, ajustado para se adequar a um projeto real desenvolvido por (PINTO et al., 2014)). Sendo assim, esse resultado pode ser melhorado ajustando melhor a variação da tensão V_{be} .

6.4 Regulador de Tensão (LDO)

O regulador de tensão é um bloco que utiliza os blocos citados anteriormente, o amplificador operacional, *charge pump* e *bandgap* modelados em Verilog-AMS são utilizados para simular o comportamento desse grande bloco. Como dito no planejamento, esse circuito não será completamente simulado em Verilog-AMS, isso se dá pois não é interessante modelar o elemento de passagem e os resistores. Sendo assim, pode-se dizer que, apesar de não ter substituído nenhum bloco em alto nível por circuitos, é feita uma simulação mista. O esquemático se dá com os blocos funcionando em Verilog-AMS juntamente com componentes da tecnologia TSMC 180nm.

Os blocos utilizados para o amplificador operacional, *charge pump* e *bandgap* estão em Verilog-AMS, representados por blocos apenas, ou seja, não circuitos elétricos representados por eles, apenas modelos. E, a regulação de tensão feita pelos resistores, elemento de passagem (implementado por um N-MOS) e os capacitores como dispositivos eletrônicos da tecnologia.

Sendo assim, com esse circuito é realizado a simulação transiente para a obtenção dos resultados apresentados na Fig. (22). Onde observa-se o resultado do *Charge Pump*, dado um sinal ASK de entrada, com o sinal de tensão do *bandgap* para temperatura ambiente de 27°C e a saída do LDO.

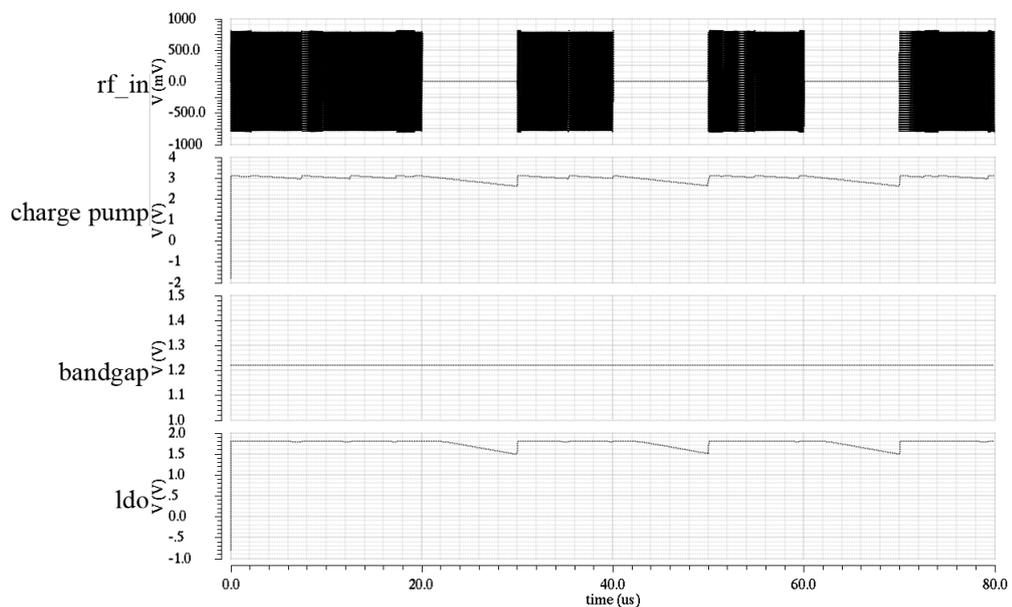


Figura 22 – Transiente do Circuito do Regulador de Tensão (LDO)

E como o objetivo do LDO é fornecer uma tensão estável na região desejada (1,8V), o sinal de saída esperado deve ser algo que retire as irregularidades que o *Charge Pump* vem a apresentar. Conforme Fig. (23), é possível ver que, ao comparar o sinal vindo do *charge pump* com a saída do LDO, há pouca variação do sinal, se mantendo bem regular aos 1,79V. Ainda, quando o sinal recebido é atenuado demais, o LDO não segura o bastante, já que ele tenta manter variações não muito grandes, mas mesmo assim, a mantém estável quando há uma grande variação, de quase 0,2V. É um resultado importante para o sistema, que precisa de uma alimentação regular.

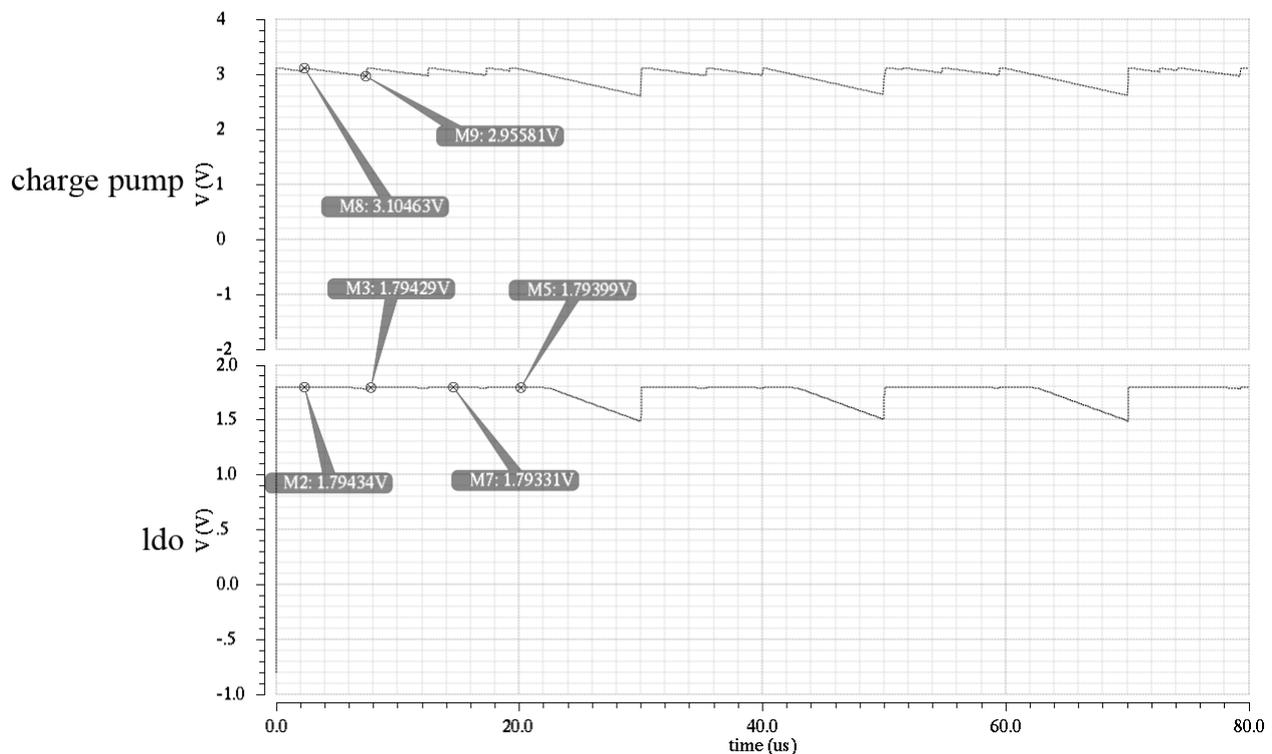


Figura 23 – Comparativo entre os Níveis de Tensão na Saída do Charge Pump e LDO

6.5 Comparador

O comparador projetado deve possuir histerese para evitar que pequenas variações (*ripple*) ou ruídos próximos ao outro sinal sejam desconsiderados, na Fig. (25) é possível ver o comportamento dos comparadores, o primeiro, sem histerese, possui uma saída bem ruidosa próximo à tensão de limiar (*Comparator threshold*) que prejudicaria o funcionamento de blocos dependentes à esse, e o segundo com a histerese, as tensões *trip* (VTRP) positivas e negativas garantem uma faixa para que haja a transição proporcionando uma saída estável ao sistema. Como os sinais de entrada ao comparador no módulo demodulador ASK são naturalmente ruidosas, o uso de histerese é essencial para se obter seu comportamento digital.

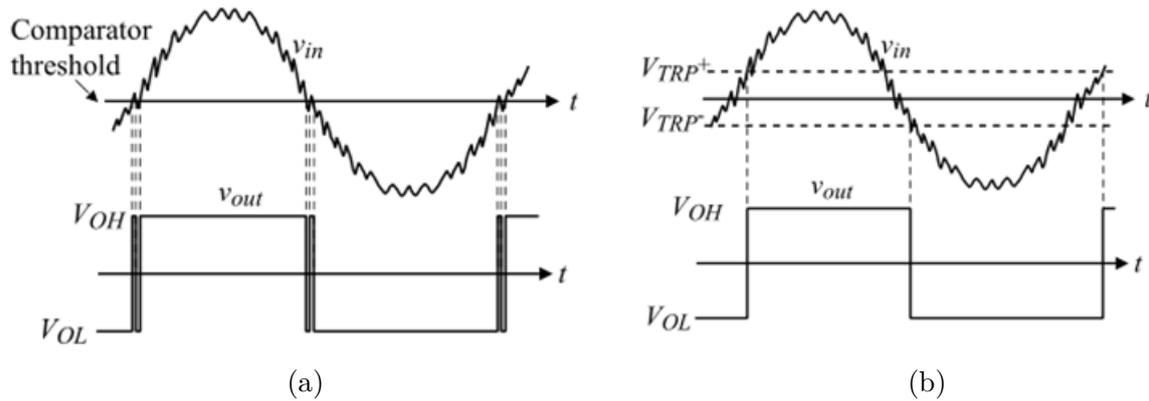


Figura 24 – Resposta de um comparador a uma entrada ruidosa (a) Comparador sem histerese (b) Comparador com histerese (ALLEN; HOLBERG, 2002)

Além de considerar a histerese, que é um ponto fundamental para o projeto do circuito e funcionamento de diversos sistemas, o comparador é capaz de realizar uma transição entre níveis lógicos mais suaves, ou seja, não instantaneamente, essas transições suaves são importante para o caso do sinal fazer parte de um outro sistema.

Para realizar a histerese, considere o seguinte trecho do código:

```

28. logic saida;
29. reg saida;
30.
31. //processo continuo
32. always @(above(V(vin_p, vin_n) - thrhi))
33. saida = 1;
34.
35. always @(above(thrlo - V(vin_p, vin_n)))
36. saida = 0;

```

A função **above** faz parte da chamada travessia de limiar (*Threshold Crossing*). Existem duas funções que monitoram esse cruzamento, o *cross* e o *above*. A função *above* é bem parecida com a função *cross*, exceto que enquanto *cross* apenas produz eventos em uma análise temporal, *above* irá disparar em análise DC (seja *DC operating point* ou análise DC). Sendo assim, a função *cross* é usada para gerar um evento analógico quando o resultado de uma expressão passe pelo zero em uma direção particular, que pode ser definida nos argumentos da função, já a função *above*, não especifica uma direção, o que acontece é que ela gera um evento quando o argumento passa a ser positivo.

De forma geral, ela gera um evento quando: O argumento transiciona do negativo para o positivo durante uma análise DC (DC *sweep*, caso utilizado comumente para avaliar histerese) ou em análise temporal (transiente, etc.). Ou durante uma análise de ponto de operação DC.

Sendo assim, partindo do código, quando a tensão transicionar o limiar, definido pelos **thrlo** e **thrhi**, será gerado um evento, esse evento gerará uma saída digital em nível lógico 1 ou zero (já que a variável que está sendo trabalhada nesse processo é de natureza *logic*, indicando um sinal digital), o primeiro caso quando a tensão ultrapassar **thrhi** ou no segundo quando ultrapassar **thrlo**. Como na função o valor de **thrlo** é subtraído de $V(p,n)$, ele só gera o evento quando a tensão diminuir ao ponto de passar, de forma decrescente, esse limiar, que será o caso de quando essa subtração possuir valor positivo.

Contudo, é necessário que, para que o circuito faça parte de um sistema, essa lógica digital possua um comportamento com níveis analógicos. Sendo assim, considere o trecho do código referente ao processo analógico a seguir:

```
39. analog begin
40. @(posedge saida) aux_out = 1;
41. @(negedge saida) aux_out = 0;
42. V(vout) <+ V(avdd)*transition(aux_out, td, tt, tf);
43. I(avdd, agnd) <+ pot;
44. end
```

Nas linhas 40 e 41 indicam que, quando houver uma transição, positiva e negativa, respectivamente, uma variável do tipo *real* receberá um valor para representar aquela transição. Com essa variável, é multiplicado ao valor de tensão de alimentação *avdd* e agregando ao valor de tensão de saída. Porém essa variável é transicionada a partir da função *transition* de forma que suavize o sinal de saída.

Com um sinal triangular de teste (*vin*), comparado a uma tensão de 1V, é feito a simulação desse modelo, e os resultados obtidos podem ser observados na Fig. (25). Mesmo comparado a um sinal de 1V, a transição não ocorre exatamente em 1V, o mesmo ocorre quando o sinal diminui. E, por fim, a transição não ocorre instantaneamente, levando $0,02\mu s$.

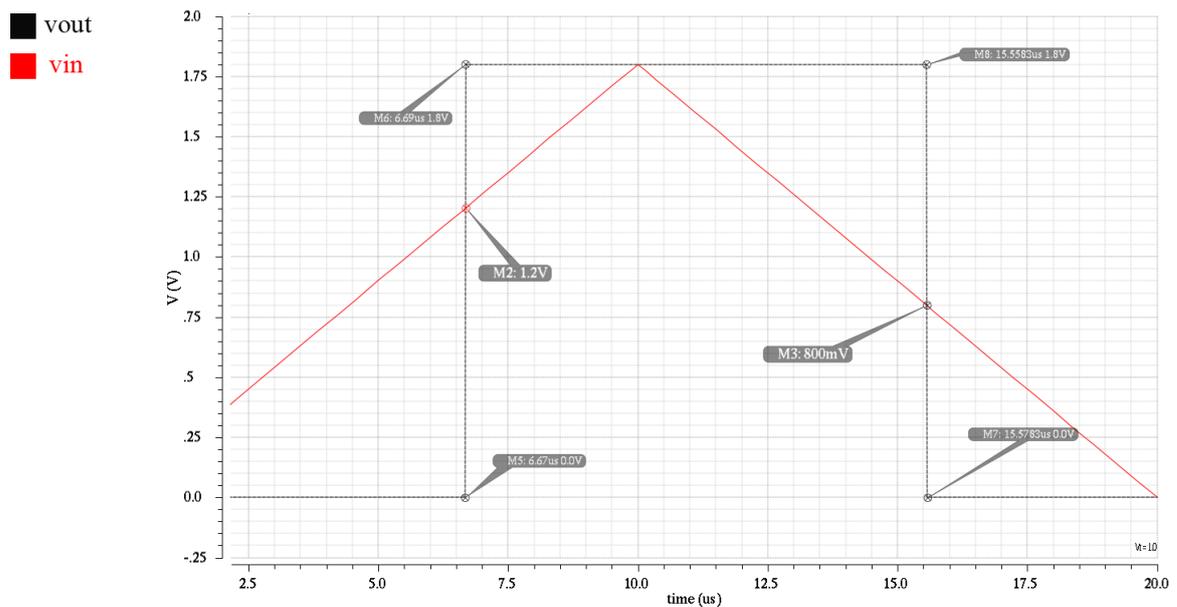


Figura 25 – Transiente do Comparador com Histerese

6.6 Detector de Envoltória e Filtro de Média

Os modelos do detector de envoltória e filtra de média são projetados baseado em suas características de filtro. Como os dois são um filtro passa baixa de um circuito RC, isso é modelado com a diferença de que, para o detector de envoltória, há um estágio de um retificador antes do filtro e valores de resistência e capacitância diferentes, que podem ser alterados dado os parâmetros de simulação.

Sabendo disso, considere o código referente ao processo analógico do circuito do detector de envoltória:

```

23. analog begin
24.
25. //retificador
26. if(V(rf_in) > vlim)
27. vout = multiplier*V(rf_in);
28. else
29. vout = vlim;
30.
31. //filtro
32. I(out) <+ -vout/res;
33. I(out) <+ cap*ddt(V(out));

```

```

34. I(out) <+ V(out)/res;
35.
36. I(avdd, agnd) <+ pot;
37.
38. end

```

O retificador de onda está descrito nas linhas 26 a 29, note que, ele compara o sinal de entrada com um limiar *vlim*, se for maior, a saída *vout* (variável temporária interna ao projeto do tipo *real*) recebe a entrada multiplicada por uma constante (caso esse circuito passe por um retificador com um multiplicador de tensão), caso contrário, recebe apenas o limiar. Sendo $vlim = 0$. Esse trecho irá fornecer apenas a parte positiva da onda.

Em seguida, nas linhas 32 a 34, é realizado o filtro. Esse filtro é feito simplesmente a partir do nó de saída do circuito no qual haverá o filtro RC, ou seja, inicialmente recebe a corrente de saída em função da entrada, e conseqüentemente, as correntes do capacitor e do resistor que estão filtrando o sinal. Simulando um filtro analógico, porém descrito em linguagem de hardware.

O filtro média não possui a necessidade de retificar o sinal, apenas realizar um filtro na frequência do sinal. Sendo assim, é retirado o trecho de retificador do código do Detector de Envoltória e deixado apenas o trecho do filtro, conforme o trecho a seguir:

```

22. analog begin
23.
24. //filtro
25.
26. I(out) <+ -V(rf_in)/res;
27. I(out) <+ cap*ddt(V(out));
28. I(out) <+ V(out)/res;
29.
30. I(avdd, agnd) <+ pot;
31.
32. end

```

Como a entrada do circuito do filtro de média é a saída do detector de envoltória, os dois são simulados e obtidos os resultados conforme Fig. (26), onde há o sinal de entrada, em seguida o sinal retificado e o resultado pelo filtro de média.

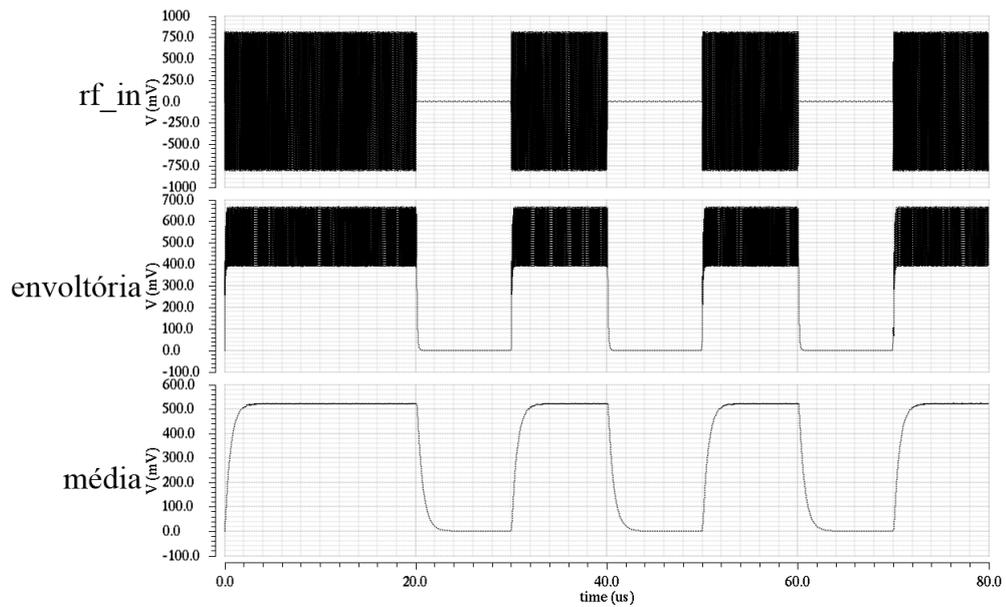


Figura 26 – Transiente dos Modelos de Detector de Envoltória e Filtro de Média

6.7 Modulador

Como dito no planejamento da modelagem, o bloco responsável por realizar a modulação do sinal é uma chave que capaz de abrir ou fechar o circuito em uma determinada região. Para isso, o modelo é capaz de se tornar um resistor para determinados sinais de controle. Sendo assim, considere o seguinte trecho do código:

```

18. analog begin
19. @(cross(V(key)-v_th, 0));
20. if (V(key) > v_th)
21. V(v_out,v_in) <+ Ron*I(v_out,v_in);
22. else
23. V(v_out,v_in) <+ Roff*I(v_out,v_in);
24. end

```

Como dito anteriormente a função *cross* quando cruzar a tensão *v_th* disparará um evento analógico, com isso, irá avaliar a direção na qual foi cruzado, caso, a tensão de controle da chave seja maior após o cruzamento, o circuito se tornará um resistor de resistência *Ron* (Resistência quando fechado – ligado – de valor bem baixo, que pode ser alterado nas simulações), caso contrário, *Roff*, com resistência bem alta, a fim de simular um circuito aberto.

Utilizando o esquemático utilizado para antenas, com a chave alterando a sintonia da frequência de ressonância do circuito externo, é obtido os resultados da Fig. (27).

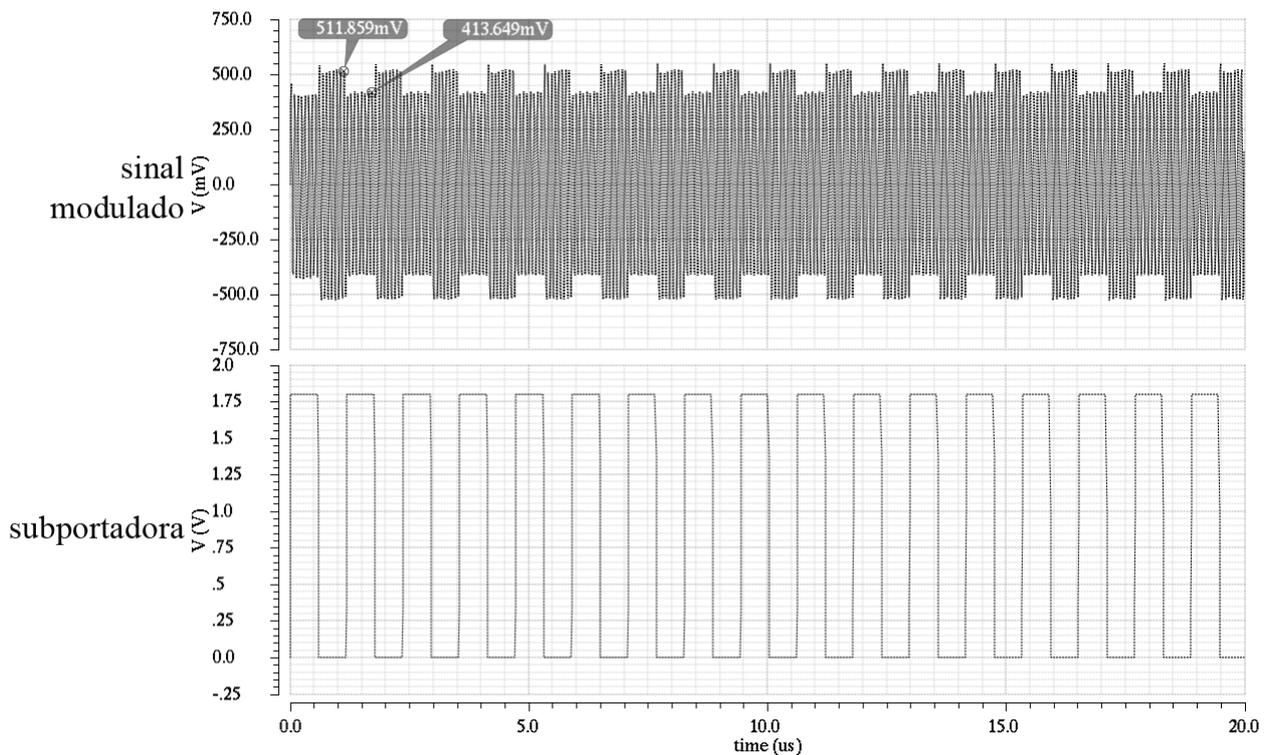


Figura 27 – Transiente do Sinal Modulado

Com um clock de 847,5 kHz para simular a subportadora digital controlando a chave, é possível observar a amplitude da onda resultando modulada em ASK pela subportadora digital em BPSK. Com a amplitude em sintonia com 13,56 MHz, foi obtido 511,85 mV de pico enquanto que, descasando o circuito ressonante, foi obtido 413,64 mV de pico. Com esses resultados o índice de modulação obtido foi de 10,61%, conforme ISO 14443.

6.8 Oscilador

O modelo do oscilador foi planejado para se comportar com características de um oscilador em anel, sendo assim, é possível utilizar sua equação característica para definir o período do clock que será gerado. Considere o trecho a seguir:

3. 'timescale 1ns/1ps
42. always begin
43. $\#((2*(1e9)*stages*inv_delay)/2)$
44. vout = ~vout;
45. end
55. @(posedge vout) aux_out = 1;

```
56. @(negedge vout) aux_out = 0;
```

Na linha 43 do código o operador `#` indica que o código deve esperar por aquele determinado tempo para prosseguir com a execução do código. Por estar inserido em um trecho de processo contínuo (indicado por *always*), esse trecho irá se repetir sempre que houver aquela espera, esse tempo é definido de acordo com a expressão do oscilador em anel em que as características do seu circuito determinarão esse valor. Ainda, por se tratar de processo digital, é declarado o *timescale* na linha 3, o que pede um ajuste do *delay* à escala, sendo a expressão multiplicada pelo valor de $1e9$ (10^9 , para compensar a escala em nanosegundos). Logo, dado o cada meio período, uma variável interna *vout* é invertida.

Analogamente ao processo do comparador, utilizando as funções de identificação de borda de subida ou descida como nas linhas 55 e 56. é feito o processo de *transition* para suavizar o tempo de transição enquanto o mesmo é enviado à tensão de saída como nível analógico.

Ainda, como foi planejado, o circuito há um enable junto à uma porta NAND para iniciar o oscilador ou desligá-lo. Similarmente ao comparador é utilizado um processo contínuo, conforme trecho do código a seguir:

```
47. always @(above(V(enable) - 1))
48. hab = 1;
49.
50. always @(above(1 - V(enable)))
51. hab = 0;
58. V(saida) <+ hab*atenu*V(avdd)*transition(aux_out, td, tt, tf);
```

Quando houver a transição entre um valor de limiar (linhas 47 a 51), arbitrário a 1 para ativá-lo, o sinal é multiplicado por 0 (desliga o oscilador) ou por 1 (mantém o sinal na porta de saída), conforme linha 58, que possui um parâmetro de atenuação, sendo possível decidir um percentual do valor máximo do sinal para a tensão de alimentação.

Para um oscilador arbitrário de 1kHz, manipulando os valores de número de estágio, atraso do inversor, o resultado obtido pelo modelo é observado na Fig. (28). E a transição na Fig. (29).

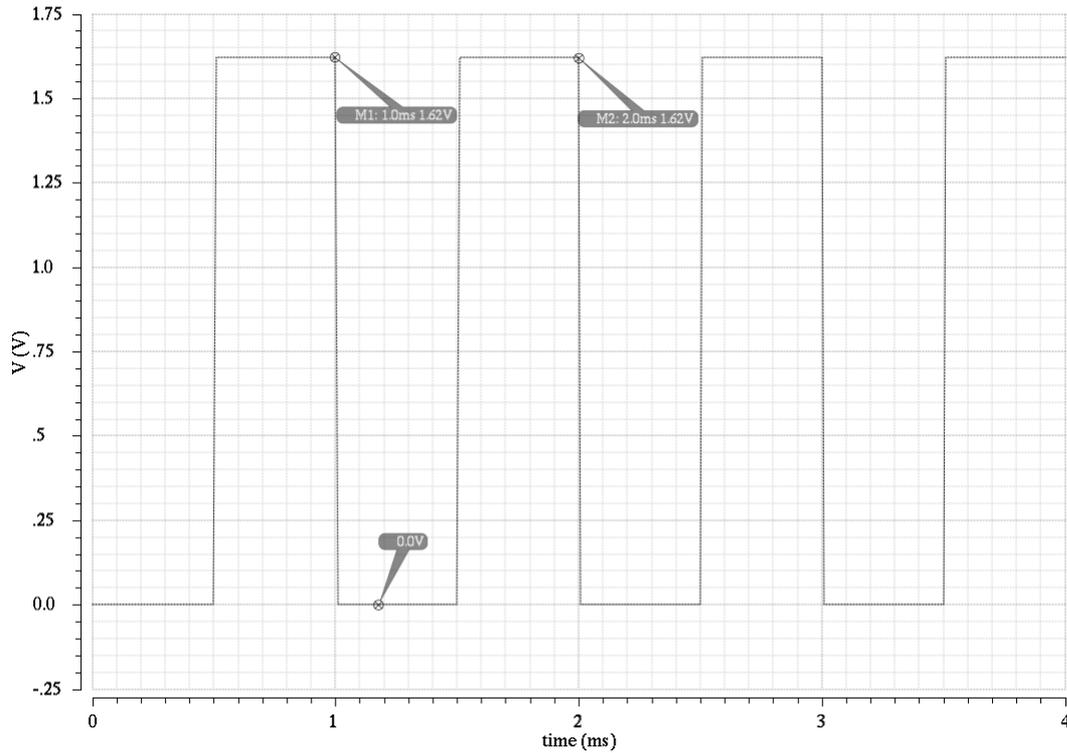


Figura 28 – Transiente do Oscilador

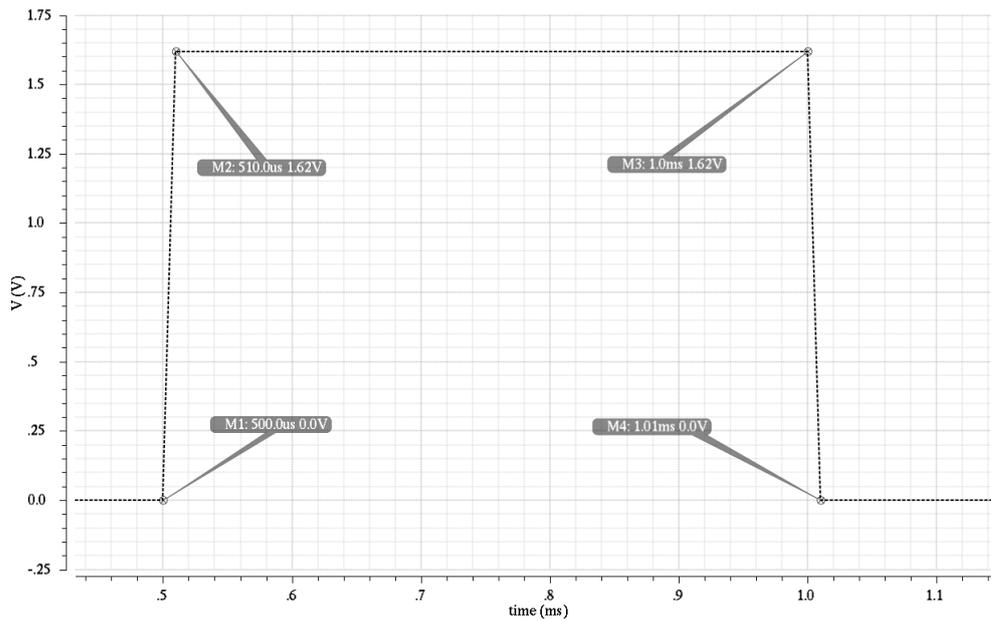


Figura 29 – Tempos de Subida e Descida do Oscilador

6.9 Demodulador ASK em Verilog-AMS

Nessa etapa de projeto, os blocos projetados são conectados de forma a formar o demodulador ASK. Os blocos do Regulador de Tensão, comparador, e filtros compoem o sistema do demodulador. Onde, o regulador de tensão é utilizado como a tensão de alimentação do circuito, e a envoltória do sinal é comparada com a média do mesmo pelo comparador com histerese. A Fig. (30) apresenta os resultado do demodulador ASK em

Verilog-AMS, juntamente com as respostas dos blocos individuais em conjunto.

Pode-se observar que, onde há o sinal senoidal do sinal de entrada modulada em ASK, o sinal de saída apresenta nível lógico 1 e quando em zero, nível lógico 0. Note que, o sinal de saída possui o nível lógico referente ao sinal do LDO, tendo em vista que o mesmo alimenta o circuito para seu funcionamento.

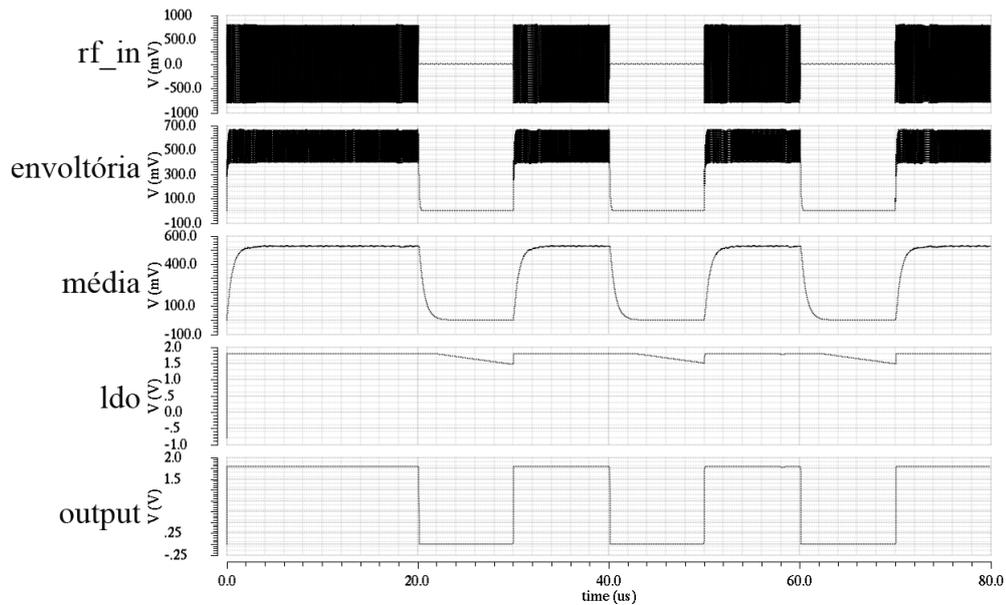


Figura 30 – Transiente do Sistema completo do Demodulador ASK em Verilog-AMS

6.10 Demodulador ASK em Nível de Circuito

Devido à oportunidade de um *tape-out*, como não houve tempo hábil para aplicar a metodologia *Top-Down*, foi definido que esse bloco em específico seria projetado em nível de transistor para fabricação. Esse capítulo apresentará o projeto do demodulador ASK e seus resultados que serão utilizados para a modelagem do bloco em Verilog-AMS.

Como dito anteriormente, a função do demodulador é transformar esse sinal ASK em um sinal digital. Então, onde há a função senoidal deve-se obter um sinal em nível alto e quando o sinal é nulo e constante, recebe-se um sinal digital em nível baixo. Para realizar esta função é feita a detecção da envoltória do sinal. Em seguida, utilizando um filtro de média, estes dois sinais (envoltória e média) são comparados, obtendo-se como resultado os níveis lógicos 1 ou 0.

O sinal RF passa por um detetor de envoltória, é elevado seu nível DC em um dobrador de tensão com capacitores e diodos como na Fig. (31). Nessa topologia são utilizados os diodos de barreira schottky, pois devido à sua baixa tensão de condução direta e sua capacidade de chaveamento em altas frequências, são ideais para a utilização em detectores de RF. Sua barreira de potencial está entre 0,2 V e 0,3 V, enquanto que

os diodos tradicionais de silício possuem algo em torno de 0,6 V e 0,7 V. Isso faz com que tenham praticamente a mesma tensão que um diodo de germânio. Além disso, eles possuem um tempo de recuperação rápido devido a seu baixo armazenamento de carga, possibilitando sua utilização em aplicações com alta frequência de chaveamento.

A tecnologia utilizada permite o uso de capacitores de até 1,79 pF. Isso justifica o fato de se utilizar capacitores em paralelo, necessários para obter uma capacitância suficiente para o armazenamento de energia para o multiplicador.

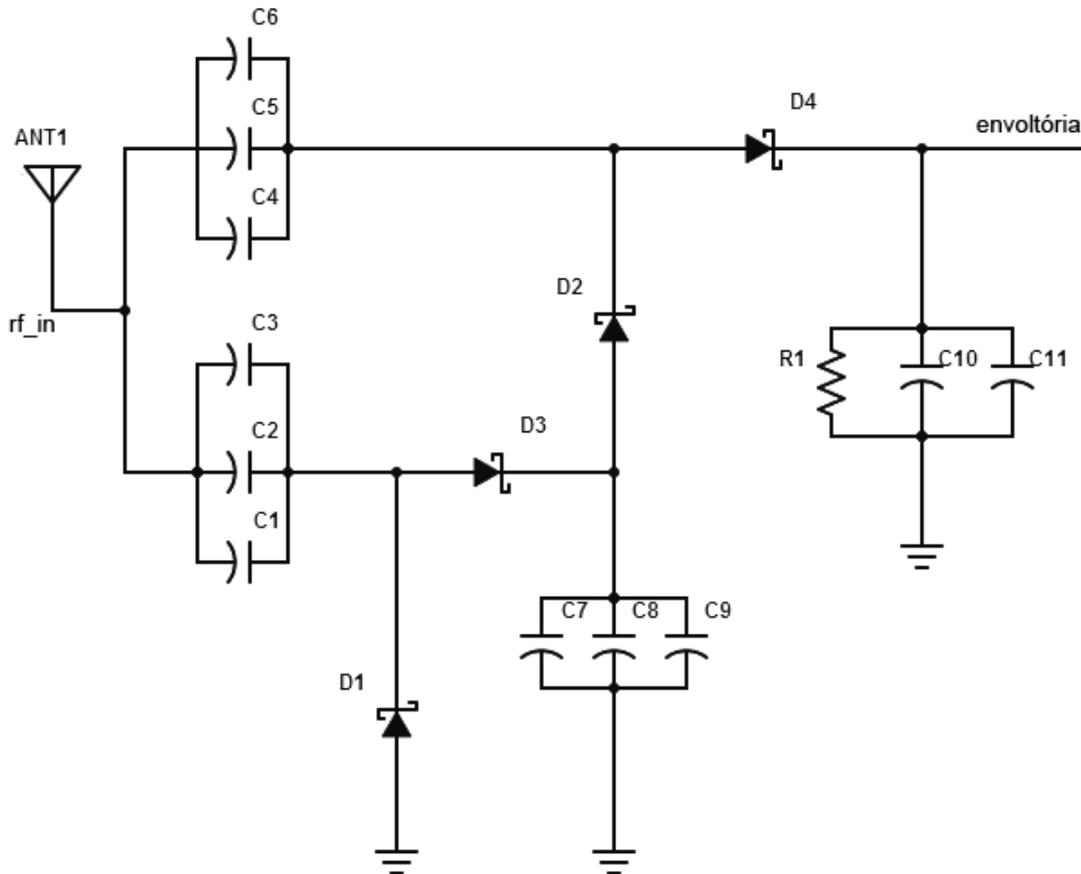


Figura 31 – Esquemático do Detector de Envoltória do Sinal RF de Entrada

Então, como observado no esquemático da Fig. (31), é feita uma elevação da tensão de entrada, realizado pela parte do circuito que antecede o diodo D4 por meio de um multiplicador de tensão Dickson, para se obter 600mV após a detecção de envoltória. Em seguida, com o nível de tensão elevado, é feita a detecção de envoltória com diodo e um filtro, pouco abaixo da frequência da onda (13,56 MHz), com um resistor e capacitor em paralelo. O valor do resistor é ajustado a fim de se reduzir o uso de capacitores.

Essa elevação de tensão se dá pelo motivo de que o sinal recebido pela antena é baixo e é importante que se consiga no comparador uma tensão suficiente para polarizar os transistores para seu funcionamento. Além disso, há uma queda de tensão proveniente do diodo utilizado para a detecção de envoltória. Ainda assim, esse nível não pode ultrapassar o valor de 2 V, porque, devido à tecnologia 180 nm, se a tensão é maior que esse limite

o circuito será danificado nos blocos que possuem transistores MOSFET (*Metal Oxide Semiconductor Field Effect Transistor*), comprometendo o chip.

Para a média, é utilizado o mesmo princípio, porém, ao invés de se utilizar um resistor de grandes dimensões, é utilizado um MOSFET com canal tipo p (PMOS) com o gate aterrado a fim de se ter a resistência entre fonte e dreno em função das dimensões de largura e comprimento do canal. Inicialmente, o resistor foi projetado com $318\text{k}\Omega$, que ocuparia uma grande área de circuito, por esse motivo, foi substituído pelo MOSFET e ajustado o capacitor para se obter a média. A configuração pode ser vista na Fig. (32).

Pela simulação da saída do demodulador em função da temperatura, o circuito pode operar na faixa de -15°C a 95°C sem que seu comportamento sofra alterações.

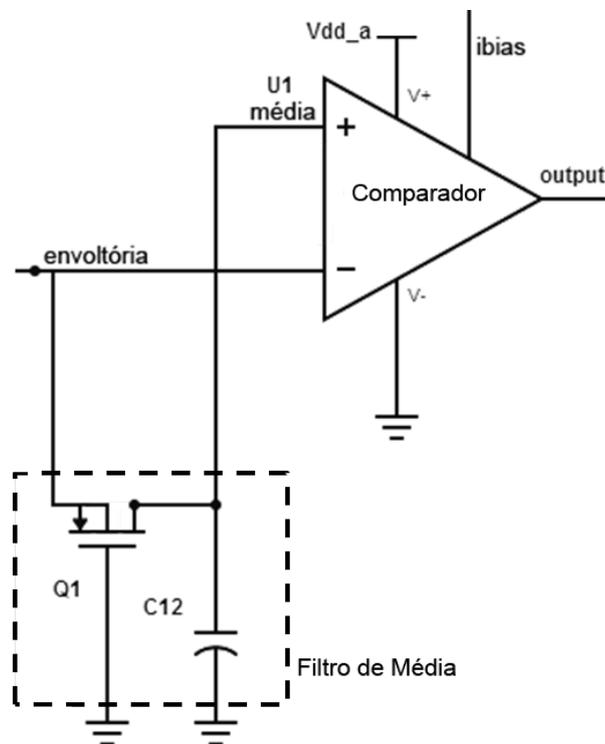


Figura 32 – Esquemático do Filtro de Média do Sinal Retificado

O sinal obtido pelo circuito simulado pelo Virtuoso CADENCE é observado na Fig. (33), onde, respectivamente, obtém-se o sinal de entrada, sinal retificado, média do sinal e saída do sinal pelo comparador.

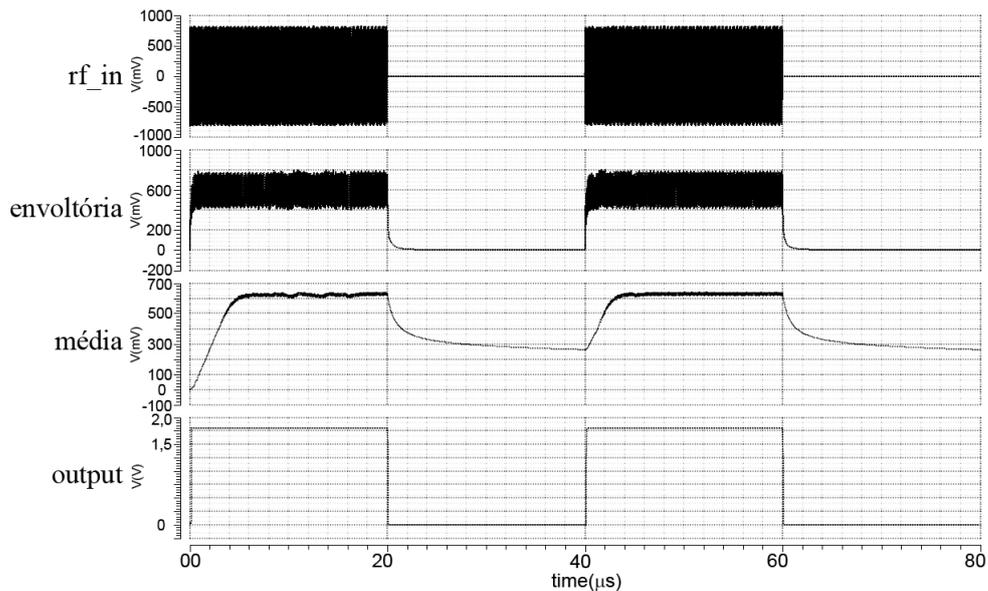


Figura 33 – Resultados Obtidos pelo Circuito (PINTO et al., 2014)

6.11 Simulações Mistas

As simulações mistas foram conduzidas utilizando o bloco Demodulador ASK, isso se deve ao fato de que o mesmo foi tanto projetado em Verilog-AMS e em nível de transistor. Contudo, foi definido duas simulações mistas para comparar os resultados com o projeto final. Além disso, como dito anteriormente, o bloco LDO possui em sua estrutura, elementos em Verilog-AMS e em dispositivos elétricos.

O foco neste capítulo é conduzir como as etapas das simulações mistas são desenvolvidas até que se obtenha o circuito final, tendo uma visão de como os resultados vão sendo obtidos com o decorrer do projeto.

Inicialmente, suponha o circuito completo em Verilog-AMS, projetado e observado seus resultados, conforme o Capítulo 6.9. Em seguida, é substituído o modelo em Verilog-AMS do detector de envoltória e simulado seu comportamento com apenas o filtro de média e o comparador em Verilog-AMS. Ao simular, sua resposta transiente é obtida conforme Fig. (34).

É observado que, sua resposta se mantém como ao sistema em Verilog-AMS, o sinal que houve alteração, apesar de pouca, foi o detector de envoltória, mas ainda, o circuito se comporta como planejado.

Subsequentemente, é projetado o filtro de média com um transistor e um capacitor para exercer a função de filtro passa baixas para frequência do sinal. Nessa etapa, apenas o comparador está modelado em Verilog-AMS. Ressaltando que, o comparador obteve os resultados esperados utilizando uma histerese de 0,3V, o que antecipa o projeto do circuito, sabendo quanto de histerese o comparador deve possuir, já que metade do circuito já está

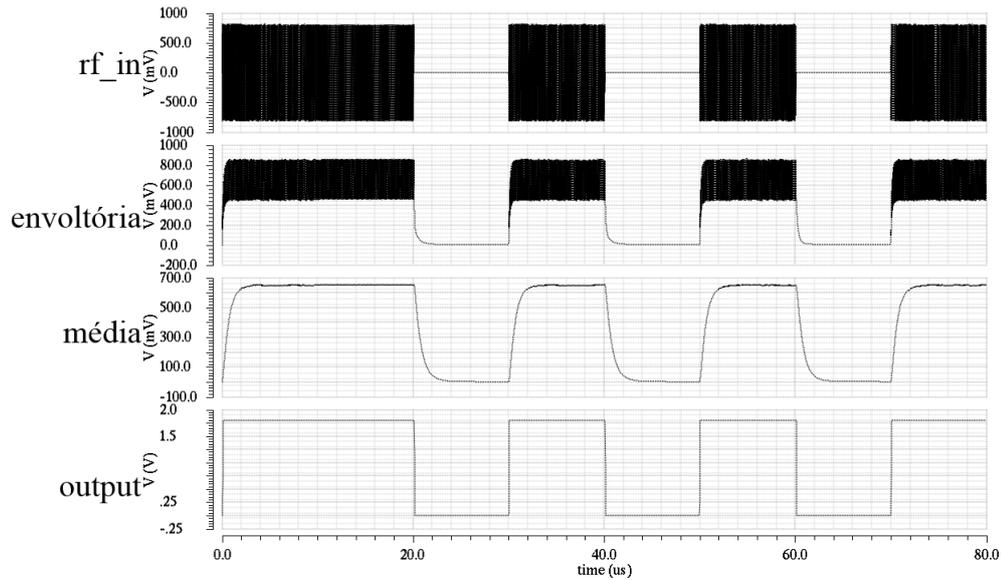


Figura 34 – Simulação Mista do Demodulador ASK Utilizando um Comparador com Histerese e o Filtro de Média em Verilog-AMS

projetado eletricamente. O resultado obtido é observado na Fig. (35).

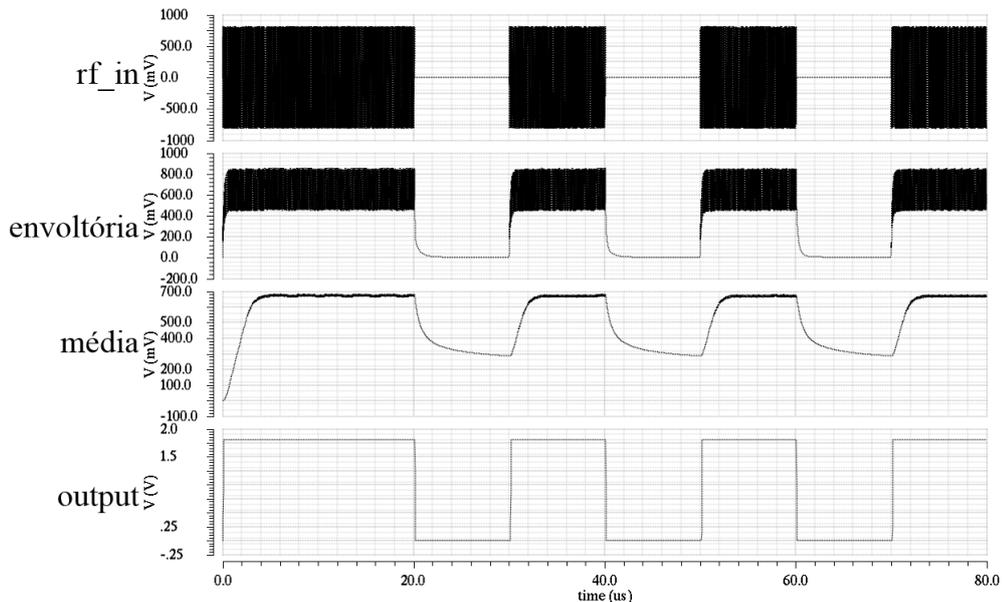


Figura 35 – Simulação Mista do Demodulador ASK Utilizando um Comparador com Histerese em Verilog-AMS

Nessa simulação, observa-se que o circuito tem todo o comportamento similar ao projeto em nível de circuito, inclusive ao comparador, mas sabe-se que o mesmo ainda não foi projetado. Por fim, projeta-se o circuito completo em nível de transistor. E o resultado obtido, como observado no capítulo anterior, possui a resposta transiente do sistema completo, muito similar ao projeto em Verilog-AMS e suas etapas de simulação mista, podendo então observar a importância dessa metodologia, em que, há o planejamento dos blocos e rapidamente sabe-se o que deve ser esperado do circuito elétrico. Aos poucos

pode ser desenvolvido blocos do sistema e mesmo assim continuar com suas simulações até a etapa de teste final.

7 Conclusão

A modelagem de um *front-end* analógico de uma *tag* de RFID passiva de 13,56MHz em linguagem Verilog-AMS foi realizada nesse trabalho. Foram apresentados os padrões e aplicações do RFID. Foi abordada a metodologia *Top-Down* onde a linguagem proposta é aplicada. Baseados na estrutura do *tag* de RFID, foi proposto a modelagem dos blocos Retificador, demodulador, modulador e oscilador. Foi projetado um demodulador ASK em nível de transistor utilizando a tecnologia TSMC 180 nm que foi enviada para fabricação. Essa oportunidade de *tape-out* proporcionou um projeto completo do sistema do demodulador ASK, que foi, desde as especificações do bloco, extração de características para o seu modelo, o projeto em Verilog-AMS, o projeto elétrico, a produção do *Layout* do seu circuito para fabricação e ainda, em paralelo, as tarefas foram executadas de tal forma que simulações mistas foram realizadas.

Para o bloco do Retificador, foi obtido um amplificador operacional com saturação não-simétrica, dependente da tensão de alimentação, o modelo do charge pump obteve resultados com descarga do sinal que podem ser arbitradas pelo projetista e o bandgap uma função dependente da temperatura, seus modelos são capazes de prever a corrente consumida entre fonte de terra. Para o oscilador, o modelo pode se adaptar à frequência desejada de acordo com as características de um oscilador em nível de transistor. Para o bloco do modulador, foi simulado utilizando um circuito externo conforme especificações da antena, obtendo os resultados previsto na ISO 14443 com índice de modulação de 10,61%. Para o bloco do demodulador, o modelo completo em Verilog-AMS foi simulado, conforme o projeto em nível de transistor, ambos tiveram as mesmas respostas, chegando ao resultado esperado, previsto pela metodologia. Ainda, tendo os dois projetos com resultados obtidos conforme o planejado, simulações mistas foram feitas em paralelo e para cada etapa de simulação mista feita, os resultados se mantiveram conforme previsto.

Sendo assim, a metodologia *Top-Down* favorece projetos com alta complexidade, esse projeto demonstrou que, conforme se tem o comportamento do sistema conhecido, os resultados que se esperam ao longo do projeto em nível de circuito já estão consolidados, e ainda, durante as simulações mistas, erros são evitados, sendo previstos antes de gerar os circuitos, com o auxílio da linguagem Verilog-AMS, os blocos são rapidamente desenvolvidos com seus comportamentos em função de entradas e saídas, que são de grande importância ainda durante o planejamento.

Portanto, o desenvolvimento de um projeto pela metodologia *Top-Down*, permite que o produto final tenha melhor qualidade possível dentro de suas especificações com um bom entendimento dos projetistas. Logo, a linguagem Verilog-AMS, proporcionou

para o projeto seu completo comportamento que condiz com projeto elétrico. Por fim, como no trabalho foi projetado os blocos individuais do sistema, os modelos poderão ser facilmente aplicados à outros projetos de circuitos integrados para a universidade que venham a utilizar tais sistemas, que, não necessariamente, sejam um projeto de uma *tag* de RFID.

Referências

- ALLEN, P.; HOLBERG, D. *CMOS Analog Circuit Design*. Oxford, New York: Oxford University Press, 2002. Citado 3 vezes nas páginas 13, 49 e 67.
- ASHRY, A.; SHARAF, K.; IBRAHIM, M. A simple and accurate model for rfid rectifier. Ain-Shams University, 2008. Citado na página 40.
- ASHRY, A.; SHARAF, K.; IBRAHIM, M. A compact low-power uhf rfid tag. University of Sheffield, 2009. Citado 2 vezes nas páginas 13 e 42.
- ATMEL. *Understanding the Requirements of ISO/IEC 14443 for Type B Proximity Contactless Identification Cards*. San Jose, Estados Unidos, 2005. 28 p. Citado 7 vezes nas páginas 13, 15, 27, 42, 50, 51 e 57.
- DAY, M. Understanding low drop out (ldo) regulators. Texas Instruments, Dallas, 2002. Citado na página 46.
- EPCGLOBAL. *EPCTM Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz – 960 MHz Version 1.0.9*. [S.l.], 2005. 94 p. Citado na página 42.
- ISO/IEC. *Information technology — Radio Frequency Identification for Item Management — Application Requirements Profiles*. Genebra, Suíça, 2004. 89 p. Citado na página 27.
- JOSE, P.; MONICA, H. K. Rfid en el sector salud: Aplicaciones, beneficios e incertidumbres. Puerto Ordaz, Venezuela, 2011. Citado na página 25.
- KUNDERTH, K. S. *The designer's guide to verilog-ams*. Boston, Estados Unidos, 2004. Citado 7 vezes nas páginas 13, 15, 31, 34, 35, 36 e 37.
- LI, H. Development and implementation of rfid technology. Chinese Academy of Sciences, 2009. Citado 2 vezes nas páginas 13 e 39.
- LIU, X. et al. A touch probe method of operating an implantable rfid tag for orthopedic implant identification. 2013. Citado na página 25.
- MICROCHIP. *microID[®] 13.56 MHz RFID System Design Guide*. [S.l.], 2004. 208 p. Citado 2 vezes nas páginas 13 e 51.
- PINTO, J. et al. Design of an ask demodulator and a bandgap reference for a passive rfid tag for 13,56mhz. Universidade de Brasília, Brasília, 2014. Citado 6 vezes nas páginas 13, 14, 40, 48, 65 e 78.
- RAJAGOPALAN, H.; RAHMAT-SAMII, Y. Ingestible rfid bio-capsule tag design for medical monitoring. University of California, 2010. Citado na página 25.
- RAZAVI, B. *RF Microelectronics*. University of California, Los Angeles: Prentice Hall, 1998. Citado 3 vezes nas páginas 13, 40 e 42.
- RFIDCANADA. *Understanding RFID (Radio Frequency Identification)*. Markham, Canadá, 2012. 19 p. Citado 3 vezes nas páginas 15, 28 e 29.

RIGELSFORD, J. M.; DAVENPORT, C. A passive rfid implant for soft tissue trauma monitoring. University of Sheffield, 2013. Citado na página [25](#).

SHUNYU-SHI et al. A design of active rfid tags based on nrf24l01. Ocean University of China, 2013. Citado na página [27](#).

STEVAN, S. Dairy cow monitoring by rfid. Piracicaba, Brasil, 2012. Citado na página [25](#).

Apêndices

APÊNDICE A – Código do Amplificador Operacional

```
1. 'include "constants.vams"
2. 'include "disciplines.vams"
3.
4. module amplificador_operacional(in_p, in_n, avdd, agnd, vout);
5.
6. //declaracao de pinos
7. input in_p, in_n, avdd, agnd;
8. output vout;
9.
10. electrical vout, in_p, in_n, avdd, agnd;
11.
12. //declaracao de parametros
13. parameter real gain = 60;
14. parameter real pot = 5e-6;
15.
16. //variaveis locais
17. real abs_gain = pow(10, gain/20);
18. real aux_out;
19. real aux_pos;
20. real aux_neg;
21.
22. //processo analog
23. analog begin
24.
25. aux_out = abs_gain*(V(in_p)-V(in_n));
26.
```

```
27. //parte positiva
28.
29. aux_pos = V(avdd)*tanh(aux_out/V(avdd))*(tanh(10*aux_out/V(avdd))+1)/2;
30.
31. //parte negativa
32.
33. aux_neg = V(agnd)*tanh(aux_out/V(avdd))*(tanh(10*aux_out/V(avdd))-1)/2;
34.
35. V(vout) <+ aux_pos+aux_neg;
36. I(avdd, agnd) <+ pot;
37.
38. end
39.
40. endmodule
```

APÊNDICE B – Código do Bandgap

```
1. 'include "constants.vams"
2. 'include "disciplines.vams"
3.
4. module bandgap(vout, avdd, agnd);
5.
6. //declaracao dos pinos
7. input avdd, agnd;
8. output vout;
9.
10. electrical vout, avdd, agnd;
11.
12. //declaracao de parametros
13. parameter real R1=15;
14. parameter real R2=1;
15. parameter real Vbe=0.7;
16. parameter real n=4;
17. parameter real pot=5e-6;
18.
19. //processo analog
20. analog begin
21. V(vout) <+ (Vbe-(2e-3*($temperature-273.15))) + $vt*ln(n)*(1+(R1/R2));
22.
23. I(avdd, agnd) <+ pot;
24.
25. end
26.
27. endmodule
```


APÊNDICE C – Código do Charge Pump

```
1. 'include "constants.vams"
2. 'include "disciplines.vams"
3. 'timescale 1ns/1ps
4.
5. module chargepump(rf_in, out, agnd);
6.
7. //declaracao de pinos
8. input rf_in, agnd;
9. output out;
10. electrical rf_in, out;
11.
12. //variaveis locais
13. real vpeak;
14.
15. //variaveis digitais
16. reg aux;
17. logic aux;
18.
19. //declaracao de parametros
20. parameter real vd = 0.3;
21. parameter real stages = 1;
22. parameter real rect_mod = 0;
23. parameter real tt = 1e-9;
24. parameter real tf = 5e-6;
25.
26. //processo analogico
27. analog begin
28.
```

```
29. @(initial_step) vpeak = 0;
30.
31. if (V(rf_in) > vpeak) vpeak = V(rf_in);
32. if (V(rf_in) < 0.05) vpeak = 0;
33.
34. V(out) <+ ((rect_mod)*V(rf_in))+((2*stages*(transition(vpeak, 0, tt, tf)-vd)));
35.
36. end
37.
38. endmodule
```

APÊNDICE D – Código da Chave

```

1. 'include "constants.vams"
2. 'include "disciplines.vams"
3.
4. module chave(key, v_in, v_out, avcc, agnd);
5.
6. //declaracao dos pinos
7. inout v_in, v_out;
8. input key, avcc, agnd;
9.
10. electrical key, v_in, v_out;
11.
12. //declaracao de parametros
13. parameter real v_th=1;
14. parameter real Ron=0;
15. parameter real Roff=1000000;
16.
17. //processo analog
18. analog begin
19. @(cross(V(key)-v_th, 0));
20. if (V(key) > v_th)
21. V(v_out,v_in) <+ Ron*I(v_out,v_in);
22. else
23. V(v_out,v_in) <+ Roff*I(v_out,v_in);
24. end
25.
26. endmodule

```


APÊNDICE E – Código do Comparador

```
1. 'include "constants.vams"
2. 'include "disciplines.vams"
3. 'timescale 1ns/1ps
4.
5. module comparador(vout, vin_p, vin_n, avdd, agnd,ibias);
6.
7. //declaracao de parametros
8. parameter real offset = 0;
9. parameter real hyst = 0.2 from [0:inf);
10. parameter real thrlo = offset - hyst;
11. parameter real thrhi = offset + hyst;
12.
13. parameter real toff=0 from [0:inf);
14. parameter real td=0 from [0:inf); //delay
15. parameter real tt= (1e-6)/100 from [0:inf); //timerise
16. parameter real tf= (1e-6)/100 from [0:inf); //timefall
17. parameter real pot= 5e-6;
18.
19. //declaracao de pinos
20. input vin_p, vin_n, avdd, agnd,ibias;
21. output vout;
22.
23. //variaveis de sistema
24. electrical vin_p, vin_n, avdd, agnd, vout,ibias;
25.
26. real aux_out;
27.
28. logic saida;
```

```
29. reg saida;
30.
31. //processo continuo
32. always @(above(V(vin_p, vin_n) - thrhi))
33. saida = 1;
34.
35. always @(above(thrlo - V(vin_p, vin_n)))
36. saida = 0;
37.
38. //processo analog
39. analog begin
40. @(posedge saida) aux_out = 1;
41. @(negedge saida) aux_out = 0;
42. V(vout) <+ V(avdd)*transition(aux_out, td, tt, tf);
43. I(avdd, agnd) <+ pot;
44. end
45.
46. endmodule
```

APÊNDICE F – Código do Detector de Envoltória

```
1. 'include "constants.vams"
2. 'include "disciplines.vams"
3. 'timescale 1ns/1ps
4.
5. module detector_envoltoria(rf_in, out, avdd, agnd);
6.
7. //declaracao dos pinos
8. input rf_in, avdd, agnd;
9. output out;
10. electrical rf_in,out, avdd, agnd;
11.
12. //parameters
13. parameter real vlim = 0;
14. parameter real cap = 7e-12;
15. parameter real res = 20e3;
16. parameter real multiplier = 2;
17. parameter real pot = 5e-6;
18.
19. //variavel interna
20. real vout;
21.
22. //processo analog
23. analog begin
24.
25. //retificador
26. if(V(rf_in) > vlim)
```

```
27. vout = multiplier*V(rf_in);
28. else
29. vout = vlim;
30.
31. //filtro
32. I(out) <+ -vout/res;
33. I(out) <+ cap*ddt(V(out));
34. I(out) <+ V(out)/res;
35.
36. I(avdd, agnd) <+ pot;
37.
38. end
39.
40. endmodule
```

APÊNDICE G – Código do Filtro de Média

```
1. 'include "constants.vams"
2. 'include "disciplines.vams"
3. 'timescale 1ns/1ps
4.
5. module filtro_media(rf_in, out, avdd, agnd);
6.
7. //declaracao dos pinos
8. input rf_in, avdd, agnd;
9. output out;
10. electrical rf_in,out,avdd,agnd;
11.
12. //declaracao dos parametros
13. parameter real vlim = 0;
14. parameter real cap = 1.79e-12;
15. parameter real res = 318e3;
16. parameter real multiplier = 1;
17. parameter real pot = 5e-6;
18.
19. //variavel interna
20. real vout;
21.
22. analog begin
23.
24. //filtro
25.
26. I(out) <+ -V(rf_in)/res;
27. I(out) <+ cap*ddt(V(out));
28. I(out) <+ V(out)/res;
```

29.

30. I(avdd, agnd) <+ pot;

31.

32. end

33.

34. endmodule

APÊNDICE H – Código do Oscilador em Anel

```
1. 'include "constants.vams"
2. 'include "disciplines.vams"
3. 'timescale 1ns/1ps
4.
5. module oscilador_anel(avdd, agnd, enable, saida);
6.
7. //declaracao dos pinos
8. input enable, avdd, agnd;
9. output saida;
10. electrical saida;
11.
12. electrical avdd, agnd, enable;
13.
14. //pinos digitais
15. reg vout;
16. logic vout;
17.
18. //declaracao de parametros
19. parameter real stages=1 from [0:inf);
20. parameter real inv_delay = 1e-4;
21. parameter real atenu = 0.9;
22. parameter real pot = 5e-6;
23.
24. //variaveis internas
25. real period, toff, td, tt, tf;
26. real aux_out;
```

```
27. real hab;
28.
29. //parametros iniciais das variaveis
30. initial begin
31. period=(2*stages*inv_delay);
32. toff=0;
33. td=0;
34. tt=period/100;
35. tf=period/100;
36. end
37.
38. initial vout = 0;
39. initial hab = 0;
40.
41. //processo continuo
42. always begin
43. #((2*(1e9)*stages*inv_delay)/2)
44. vout = ~vout;
45. end
46.
47. always @(above(V(enable) - 1))
48. hab = 1;
49.
50. always @(above(1 - V(enable)))
51. hab = 0;
52.
53. //processo analog
54. analog begin
55. @(posedge vout) aux_out = 1;
56. @(negedge vout) aux_out = 0;
57.
```

58. $V(\text{saida}) <+ \text{hab} * \text{atenu} * V(\text{avdd}) * \text{transition}(\text{aux_out}, \text{td}, \text{tt}, \text{tf});$

59. $I(\text{avdd}, \text{agnd}) <+ \text{pot};$

60.

61. end

62.

63. endmodule

APÊNDICE J – Layout do Demodulador ASK

A Figura (37) apresenta o *Layout* do demodulador ASK projetado para fabricação.

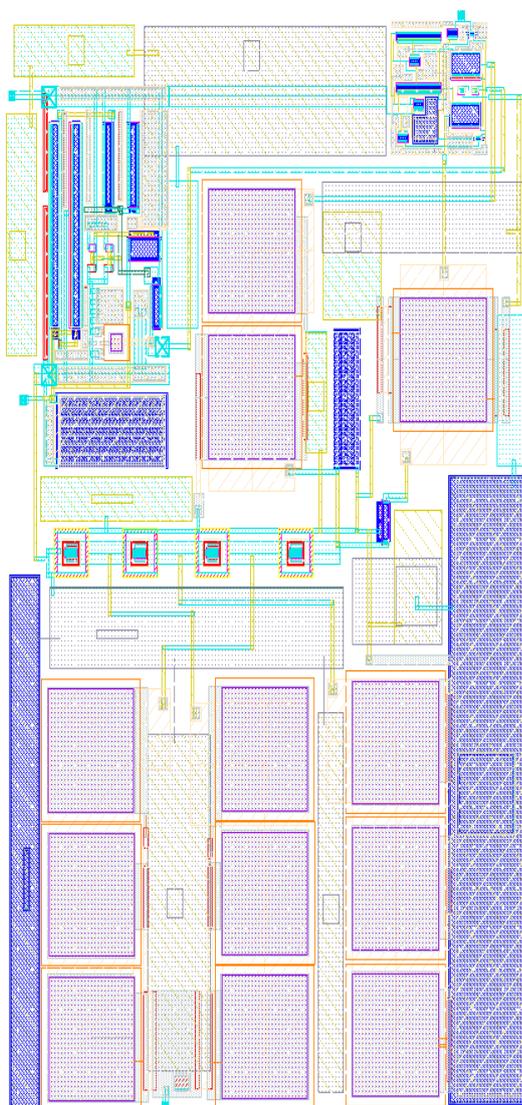


Figura 37 – Layout do demodulador ASK.