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

PROJETOS DOS BLOCOS CHARGE PUMP, LOOP FILTER E PFD DO PHASE LOCKED LOOP DE UM TRANSCEPTOR ZIGBEE

Autor: Wesley de Jesus Gomes
Orientador: Wellington Avelino do Amaral

Brasília, DF

2015

WESLEY DE JESUS GOMES

TÍTULO: PROJETO DOS BLOCOS CHARGE PUMP, LOOP FILTER E PFD DE UM TRANSCEPTOR ZIGBEE

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

Orientador: Prof. Dr. Wellington A. do Amaral

Brasília, DF
2015

CIP – Catalogação Internacional da Publicação

Gomes, Wesley de Jesus.

Título da Monografia: Projeto dos blocos Charge Pump, Loop Filter e PFD de um tansceptor ZigBee/Wesley de Jesus Gomes. Brasília: UnB, 2015. 76p. : il. ; 29,5 cm.

Monografia (Graduação) – Universidade de Brasília

Faculdade do Gama, Brasília, 2015. Orientação: Wesley de Jesus Gomes.

1. PLL. 2. Charge Pump. 3. Loop Filter I. Avelino do Amaral, Wellington. II. Dr.

CDU Classificação

PROJETO DOS BLOCOS CHARGE PUMP, LOOP FILTER E PFD DE UM TRANSCEPTOR ZIGBEE

Wesley de Jesus Gomes

Monografia submetida como requisito parcial para a obtenção de Título de Bacharel em Engenharia Eletrônica da Faculdade UnB Gama – FGA, da Universidade de Brasília, em data/data/data apresentada e aprovada pela banca examinadora abaixo assinada:

_____

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

Orientador

_____

Prof.Dr.: Gilmar Silva Bezerra, UnB/FGA

Membro Convidado

_____

Prof. Dr.: Diogo Caetano Garcia, UnB/FGA

Membro Convidado

Brasília, DF

2015

A todos aqueles que de alguma forma estiveram e estão próximos de mim, fazendo esta vida valer cada vez mais a pena.

AGRADECIMENTOS

Agradeço ao meu professor orientador Wellington Avelino de Amaral por permitir desenvolver esse tema tão interessante e importante em microeletrônica.

Também agradeço a minha família por estar presente em momentos de alegrias e tristeza, além de me apoiar em todas as decisões que tomei durante essa jornada.

Agradeço também aos meus amigos e colegas pela convivência nesses anos, pelo apoio e palavras de incentivo.

Um agradecimento especial para minha Avó Maria Pontes Gomes por sempre ter me apoiado e incentivado nessa caminhada, sempre com palavras de incentivo e motivação. Além de ensinar a nunca desistir diante de uma adversidade, sempre lutando por uma vida melhor e mais feliz. A senhora foi um exemplo de perseverança e amor à vida. Obrigado por todos os ensinamentos que foram muito importantes para formar o meu caráter.

"O que as vitórias têm de mau é que não são definitivas. O que as derrotas têm de bom é que também não são definitivas."

Saramago

RESUMO

A proposta deste trabalho consiste no desenvolvimento de três blocos essenciais de um *Phase Locked Loop* (PLL), que será utilizado em um transceptor *ZigBee* com frequência de operação de 2,4GHZ, e fará parte um uma banco de modelos de protocolos de rádio frequência RF geridos pela FGA. Os blocos projetados são o *Phase Frequency Detector* (PFD), a Charge Pump e o Loop Filter. O projeto segue uma metodologia tradicional Top-Down, usando uma linguagem Verilog-AMS para os modelos de alto nível. Portanto, no início são desenvolvidos os modelos de alto nível, com o objetivo de extrair a especificação blocos. Em seguida, os blocos serão concebidos no nível do transistor para, no final, os esquemas serem gerado. O projeto foi desenvolvido com a tecnologia de TSMC 0.18µm e as ferramentas CADENCE para a simulação e validação dos blocos projetados.

Palavras Chaves: phase locked loop, *Charge Pump*, *Loop filter*, phase frequency detector, Protocolo *Zigbee*, *Top-Down*, Radiofrequência.

ABSTRACT

The purpose of this work is to design and model three essential building blocks of a phase locked loop (PLL); the Phase Frequency Detector (PFD), the Charge Pump and Loop Filter. This PLL will be used in a ZigBee transceiver with operating frequency of 2.4GHz. This project is part of an effort of the FGA researchers, aiming the development of a library with high level models of RF building blocks and transceivers used in traditional RF protocols. The design will follow a traditional top-down methodology, using Verilog-AMS language for the high level models. Therefore at the beginning the high level models will be developed, aiming to extract the blocks specification. Then, the blocks will be designed in the transistor level for, at the end, the layouts can be generated. The project will be developed using the 0.18µm CMOS TSMC technology installed in the CADENCE tools.

Keywords: PLL, Charge Pump, Loop filter, PFD, Zigbee Protocol, Top-Down, Radio Frequency.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

ADS	*Advanced Design System*
AMS	*Analog Mixed-Signal*
BIAS	Polarização do circuito
BW	*BandWidth*
CI	Circuito Integrado
CMOS	*Complementary Metal-Oxide-Semiconductor*
CP	*Charge Pump*
DC	*Direct Current*
GND	*Graund*
HDL	*Harware Description Language*
IEEE	*Institute of Electrical and Electronics Enginners*
LF	*Loop Filter*
NMOS	Transistor MOS de canal N
PD	*Phase De*tector
PFD	*Phase Frequency Detector*
PLL	*Phase Locked Loop*
PMOS	Transistor MOS de canal P
RF	*Radio Frequency*
SNR	*Signal Side Band*
TCC	Trabalho de Conclusão de Curso
TSMC	*Taiwan Semiconductor Manufacturing Company*
VCO	*Voltage Controlled Oscillator*
Verilog	VERIfing LOGic
VHDL	*VHSIC Hardware Description Language*
VLSI	*Very-Large-Scale Integration*

WPAN *Wiriless Personal Area Network*

Sumário

1. INTRODUÇÃO

Atualmente a utilização de redes sem fios (*wireless*) em diversos tipos de aplicações se tornou comum visto sua praticidade e facilidade de uso em ambientes adversos e que necessitam de uma rede de comunicação e monitoramento. Desta forma, a rede *wireless* vem substituindo gradativamente as redes a cabo devido às suas vantagens como a flexibilidade (visto que, pode aumentar a área de cobertura e alcançar de lugares aonde as redes a cabo não chegam), a facilidade de instalação, a redução de custos e a possibilidade de utilizar diversas topologias, objetivando maior flexibilidade para a rede.

1.1 ZIGBEE

O desenvolvimento da tecnologia *wireless* possibilitou uma infinidade de protocolos de redes sem fios, cada um com características e públicos alvos distintos. Dentre eles podemos citar os padrões *Wi-Fi*, *Bluetooth* e o *ZigBee*. O protocolo 802.11 (*WiFi*) possuí alta taxa de transferência se comparado ao *Bluetooth* e *ZigBee*, entretanto possui um alto consumo de energia. Já a tecnologia *Bluetooth* possui média transferência de dados, possibilitando a troca de imagens e vídeos, entretanto possui limitações na quantidade de dispositivos conectados, visto que, só podem ser conectados sete dispositivos simultâneos, além de possuir alto consumo de energia. Já o protocolo 802.15.4 (*ZigBee*) possui uma baixa transferência de dados se comparado às duas tecnologias supracitadas, contudo possui um baixo consumo de energia, sendo até 100 vezes menor que no Bluetooth, podendo assim funcionar ininterruptamente por longos períodos. A tecnologia também possibilita a conexão de mais de 65mil dispositivos simultaneamente. A tabela 1 mostra o comparativo entre essas três tecnologias.

Tabela 1 Comparação Técnica entre Dispositivos de Transmissão sem fio (Campos, 2014).

Características Técnicas	**ZigBee**	**WiFi**	**Bluetooth**
Frequência de Trabalho (MHz)	2400/868/915	2400	2400
Alcance de transmissão (m)	10~100	100	10
Velocidade de transmissão (kbps)	20/250	11000	1000
Numero máximo de nós	65000	32	8
Autonomia	Muito Alta	Média	Baixa
Consumo	Baixo	Alto	Muito Alto

Diferente das tecnologias *WiFi* e *Bluetooth*, que possuem uma frequência padrão em todo o mundo, o padrão *ZigBee* possui três frequências de operação dependendo de onde o dispositivo será utilizado: 915MHz nos EUA, 868MHz na Europa e 2,4GHZ no restante do mundo. Possui uma transferência de dados de até 250 Kbps e alta densidade de nós da ordem de 65 mil, o que o torna um protocolo ideal para redes de sensores sem fio.

O protocolo *ZigBee* foi desenvolvido pela *ZigBee Aliance*, um grupo de empresas formado para desenvolver um protocolo para o estabelecimento de redes que pudessem ser utilizadas em diversos ambientes, como indústria, comércio e meio ambiente. O *Zigbee* foi projetado para implementar uma rede de dados sem fio e suas funcionalidades. Onde os dados são transmitidos para nós intermediários que fazem o redirecionamento das informações até chegar ao seu destino. Esta funcionalidade é utilizada em aplicações onde se pretende efetuar a transmissão de dados entre nós que estão fora do alcance um do outro (Saleiro et AL. 2010).

1.2 TOPOLOGIAS

Antes de definir as topologias da rede, vale ressaltar os tipos de dispositivos que podem coexistir no protocolo *ZigBee*. Existem dispositivos *Full Function Device* (FFD), caracterizados por funcionar como coordenadores de rede, e *Reduced Function Device* (RFD), caracterizado por se comunicarem

com o coordenador da rede. O protocolo faz distinção de três dispositivos lógicos: coordenador*, router* e *endpoit.* Cada um deles possui funções específicas, conforme é mostrado a seguir (Silva, 2007).

- *Coordenador*: É o dispositivo responsável por formar a rede, além de fazer a manutenção da mesma. Só existe um dispositivo desse por rede. Contudo, ele pode fazer a conexão entre outras redes.
- *Router*: Permite que mais nós se agrupem à rede aumentando, desta forma, o seu alcance físico. Pode fazer funções de controle e monitoramento.
- *Endpoint*: Permite o controle e monitoramento por meio de dispositivos associados a eles com sensores. É o que consome menos energia.

O *ZigBee* pode utilizar três tipos de topologias: estrela, arvore e malha, como se pode ver na Figura 1.

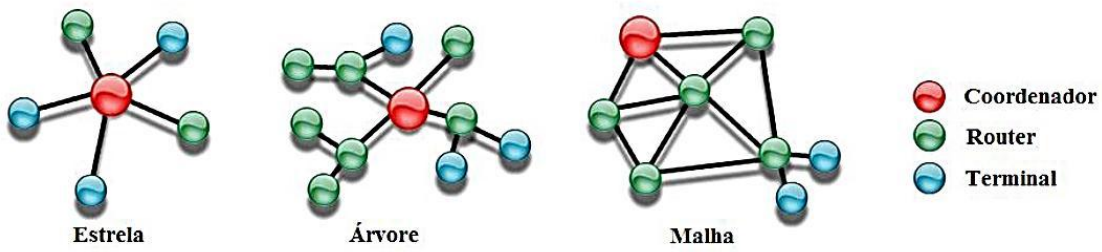

Figura 1 Topologias de rede (Campos, 2014)

Na topologia estrela, o coordenador controla toda a rede, fazendo uma comunicação direta com todos os outros dispositivos. Na topologia malha, os dispositivos coordenadores e *routers* são livres para enviarem informações para qualquer dispositivo da rede. Deste modo, a centralização da rede é menor, o que possibilita sua expansão sem grandes dificuldades. A topologia árvore e muito semelhante à malha. Entretanto, ela estabelece uma estrutura hierárquica de um nó principal chamado raiz, que possui elementos denominados ramos ou filhos. Estes ramos levam a outros elementos que não possui ramos e são conhecidos como folhas.

Desta forma, nota-se que o protocolo *ZigBee* possibilita a implementação de várias topologias, dotando os dispositivos de grande versatilidade, com a possibilidade de expandir o tamanho da rede facilmente, se comparado a outras tecnologias.

1.3 APLICAÇÕES

O protocolo *ZigBee* possui inúmeras aplicações na indústria devido a sua versatilidade e reduzido consumo de potência, agregado à capacidade de expandir indefinidamente o tamanho da rede. Por ter uma baixa velocidade de transmissão, sua principal função é conectar sensores e dispositivos com baixa transmissão de dados. Uma possível aplicação é o monitoramento de estação de tratamento de esgotos(Tose, 2010).

Um dos métodos mais utilizados para o tratamento do esgoto é a utilização de lagoas de estabilização com ou sem aeração. Ambas são construídas com diques de terra, onde o esgoto entra em uma das extremidades e sai do outro lado. A única diferença entre as lagoas é que as areadas possuem um dispositivo que oxigenam a lagoa, desta forma, auxiliando no tratamento. Contudo no Brasil as estações de tratamento de esgoto (ETE) não possuem um sistema eficiente de monitoramento das suas unidades de tratamento, desta forma, dificultando o controle operacional das estações.

Um exemplo que consta na literatura é a da automação da ETE Camburi, que pertence a Companhia Espírito Santense de Saneamento (CESAN). A estação é do tipo lagoa estabilizadora, e possui três lagoas, sendo que a primeira possui um sistema de aeração. Essa lagoa possui 18 aeradores superficiais, sendo treze de 12KW e cinco de 15KW. A estação de tratamento possui uma vazão de 300 L/s (Tose, 2010).

Mesmo com todos esses equipamentos, a estação não possuía um sistema de monitoramento de suas atividades. Desta forma, foi utilizado o protocolo *ZigBee* para monitorar seu funcionamento. Na unidade foram instalados:

- Coletores de amostras automatizados;
- Dispositivos temporizadores;
- Medidores, transmissores e registradores, para confiabilidade e segurança operacional;
- Monitoramento remoto da planta;
- Sensores de temperatura e umidade do ar;

Na ETE Camburi, foi utilizada a arquitetura de rede tipo malha com oito nós no sistema inteiro. O protocolo *ZigBee* possibilita essa flexibilidade de tamanho da rede e da quantidades de nós para um único coordenador. Na unidade de tratamento, foram feitos testes de campo para medir o alcance do sinal, levando em conta uma área de atuação dos sensores ser de aproximadamente 1,2Km².

A rede de nós sensores foi montada na estação, onde foram escolhidos três aeradores superficiais para monitoramento (nós 2, 3 e 4), a sala elétrica para acionamento desses conjuntos (nó 5), a elevatória de circulação interna (nó 7) e um kit móvel para ser utilizado em qualquer local da estação. Além do nó coordenador (1) e do nó roteador (8), responsável por concentrar e disponibilizar os dados via protocolo TCP/IP. A Figura 2 mostra as dimensões da lagoa e as disposições e configurações escolhidas para a instalação dos kits (Tose, 2010).

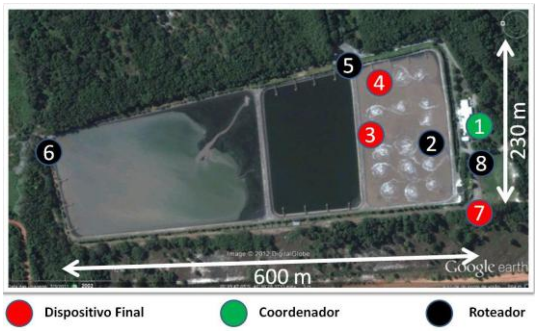

Figura 2 Disposição dos nós sensores na ETE Camburi (Tose, 2010)

O desenvolvimento de todo o projeto foi feito com um kit de desenvolvimento Arduino e placas com o padrão *ZigBee* para a transmissão dos sinais. Deste modo, foi desenvolvido um projeto para automação da estação utilizando a tecnologia *ZigBee*, tendo em visto o baixo consumo de

energia e quantidade elevada de nós do sistema. Logo, é possível ver uma aplicação efetiva do sistema *ZigBee* e comprovar a qualidade do sistema para sensoriamento remoto.

2. REVISÃO BIBLIOGRÁFICA

Este capítulo trata do funcionamento básico do PLL, e sua utilidade prática em projetos eletrônicos. Também aborda de maneira introdutória todos os blocos que o compõem, explicando o funcionamento e a importância de cada um dentro de um PLL. O capítulo também define os dois tipos de PLL, Inteiro e Fracionário, e a partir disso analisa a melhor topologia para um projeto de PLL para a tecnologia ZigBee. Por fim são descritos detalhadamente os três blocos que serão o foco principal deste trabalho: o PFD (*Phase Frequency Detector*), o LP (*Loop Filter*) e o *Charge Pump*, e são propostas topologias para cada um deles.

2.1. PLL *(Phase Locked Loop)*

O PLL (*Phase Locked Loop*) consiste em um circuito de realimentação negativa, cujo objetivo é sincronizar um sinal de saída, gerado por um oscilador interno, com um sinal de referência, tanto em frequência como em fase. Quando o PLL está sincronizado *(locked),* o erro de fase entre os sinais do oscilador e o sinal de referência é nulo. (Gomes, 2007)

O PLL básico utilizado neste trabalho é apresentado na figura 3 e consiste em um PFD (*Phase Frequency Detector*), um *Loop Filter*; *um Charge Pump*, um VCO *(Voltage control oscilator*) e um divisor (Prescaler).

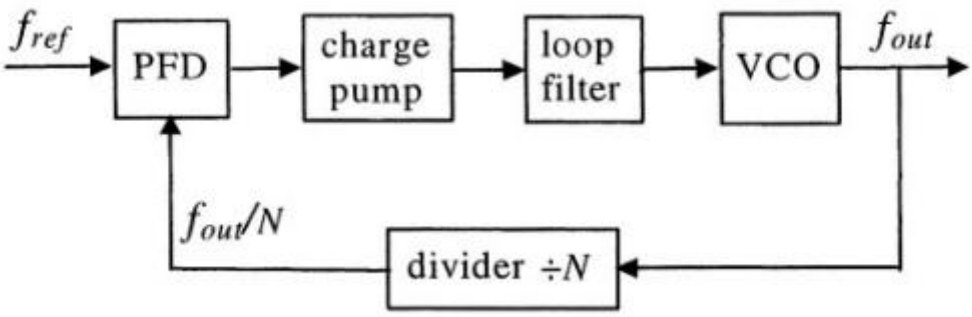

Figura 3 Blocos PLL (Cardoso, 2009)

De modo geral, o funcionamento de um circuito PLL começa com comparação entre dois sinais, o de referência e o de saída do bloco *divider* (*Prescaler*). Essa comparação é aplicada na entrada do PFD que mede a

diferença de fase e frequência entre os sinais de referência e o VCO. O *Charge Pump* converte a saída diferencial do PFD em um único sinal de saída, que passa pelo *Loop Filter* para ajustar os parâmetros de acordo com a frequência, e por fim o VCO gera uma frequência proporcional ao sinal da saída do *Loop Filter. O Prescalar* possibilita a seleção de frequências, onde o sinal pode ser um múltiplo da frequência do VCO. A seguir será descrito o funcionamento de cada um dos blocos de um PLL.

2.1.1 *Phase Frequency Detector* (PFD)

O PFD tem com principal função medir a diferença de fase e frequência do sinal de entrada*,* produzindo assim um sinal proporcional. O PFD pode ser considerado um bloco de controle visto que pode variar o valor da frequência que sai do VCO. Quando a diferença de fase entre os sinais é reduzida o erro é reduzido proporcionalmente, de forma que o erro se reduz linearmente.

2.1.2 CHARGE PUMP (CP)

O *Charge Pump* tem como função converter a saída do PFD em corrente, para desta forma fornecer ou retirar corrente do capacitor do *Loop FIlter*. O mesmo possui dois transistores que funcionam com chaves e controlam duas fontes de corrente.

2.1.3. LOOP FILTER

O *Loop Filter* é um filtro passa baixa responsável por duas funções básicas dentro de um PLL: filtrar os ruídos e frequências altas gerados pelo *Charge Pump* e converter a corrente da saída do CP para uma tensão proporcional ao erro de fase na entrada do VCO. O filtro pode ser passivo ou ativo. Comumente em projetos de PLL se utiliza filtros passivos, visto que, filtros ativos geram mais ruído (Neves, 2010).

2.1.4. *Voltage controler oscilator(VCO)*

O VCO é um circuito que responde de acordo com a tensão de entrada, produzindo uma variação da frequência na saída. Assim, o VCO é a talvez a

parte mais importante de um PLL, pois é ele que vai gerar a onda na frequência adequada. Os VCO mais utilizados na eletrônica são os pares cruzados CMOS e os osciladores em anel. Para aplicações em altas frequências o ideal é utilizar um VCO do tipo par cruzado, visto seu melhor desempenho com relação a ruídos.

2.1.5. PRESCALER

É um tipo de circuito que tem por finalidade dividir uma frequência de entrada. Np PLL essa frequência é obtida da saída do VCO. O circuito divide a frequência por N ou (N+1), dependendo de uma variável de controle. Tipicamente circuitos *prescaler* são constituídos de duas partes; são elas: um contador síncrono e um contador assíncrono. Desta forma é possível diminuir o numero de *Flip-Flops,* isso permitindo que o circuito opere em frequências maiores e com o consumo de potência menor. (Giusti, 2007)

2.1.6. *PLL INTEGER-N*

O PLL *Integer-N* possui divisor (*Pulse Swollow*), que utiliza um *Prescaler* e dois contadores: um principal, que é carregado com um valor fixo M e um programável com valor S. O valor do contador programável sempre será menor que o valor fixo, logo, S < M. No instante inicial os valores do *Modulus Control* estão em nível lógico alto, e o *prescaler* dividirá a frequência do VCO por N+1, até que o valor do contador chegue a zero, logo após o sinal *Modulus Control* é alterado e o *prescaler* passa a ser divido por N até o contador chegar a zero. Assim, os contadores são reiniciados e o ciclo estará completo. O resultado final será que o sinal de saída será composto da divisão por um valor P.

$$P = (N + 1)S + N(M - S) = NM + S \qquad (2.1)$$

Desta forma quando o PLL estiver no estado travado (*locked*), a frequência de saída será:

$$F = M * Fref = (NP + S) * Fref \qquad (2.2)$$

A figura 4 mostra um PLL integer – N .

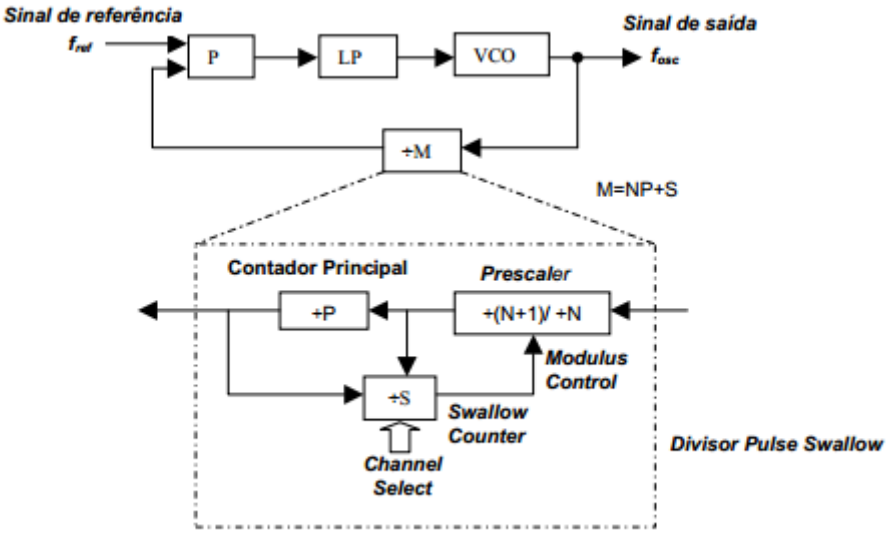

Figura 4 Arquitetura Integer-N (Arguello,2004)

2.1.7. *PLL FRACTIONAL-N*

Na arquitetura *Fractional-N,* um pulso do sinal do VCO é removido, gerando uma frequência programada *1/Tp*. A saída do removedor é ligada ao *precaler, e* quando o PLL estiver em estado *locked com* uma frequência *Nref,* assim a frequência de saída é dada por: *Fo = NFref + 1/Tp. L*ogo a frequência de saída é ajustada por 1*/Tp*.

Assim, a diferença entre as arquiteturas é que o passo de variação de frequência não depende de *Fref,* e sim de *1/Tp*. Isso possibilita maior liberdade ao projeto, além de reduzir o ruído de fase do VCO.

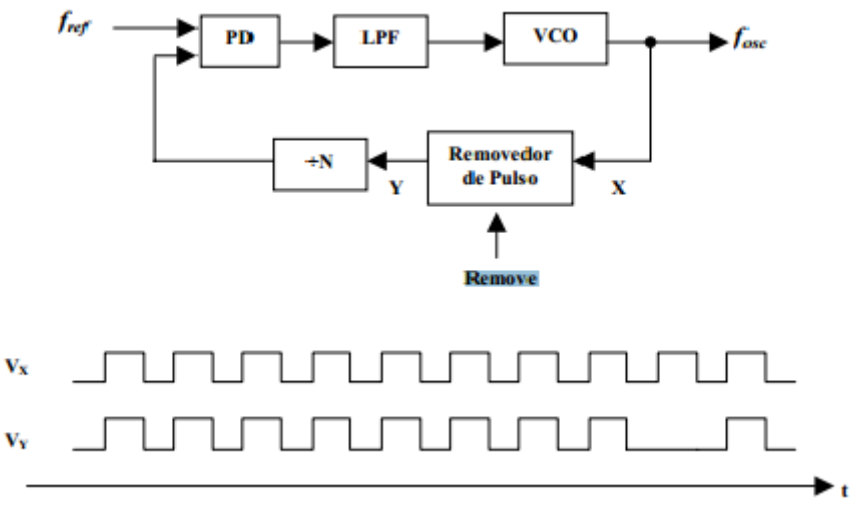

Figura 5 Arquitetura Fractional-N(Arguello, 2004)

2.2. ESTUDO PFD

O PFD é responsável por produzir um sinal proporcional à diferença de fase entre os sinais aplicados a entrada. Logo, em um sistema PLL ele compara o sinal de referência com o sinal da saída do VCO, gerando em sua saída uma tensão diferencial. A função da saída do PFD pode ser descrita pela equação 2.3.

$$Ve(s) = Kfase[\theta r - \theta o]; \qquad (2.3)$$

onde θr é a fase do sinal de referência, θo é a fase do sinal do VCO, $Kfase$ é uma constante de proporcionalidade e Ve é a tensão diferencial resultante. Logo, a equação mostra que o PFD efetua a subtração das fases dos sinais e multiplica por uma constante. Caso a diferença seja nula, o valor do sinal de saída Ve também é zero, o que caracteriza um erro zero entre as fases.

Em projetos mais simples o Detector de Fase pode ser uma simples porta lógica XOR ou XNOR. Mesmo assim, vale ressaltar o funcionamento deste tipo detector de fase. Um PD (*Phase detector*) possui duas regiões de operação, a ativa e a de saturação. Levando em conta a utilização destas portas lógicas em detectores de fase, o funcionamento do sistema se da no limite a da região ativa. A Figura 6 mostra o gráfico da função.

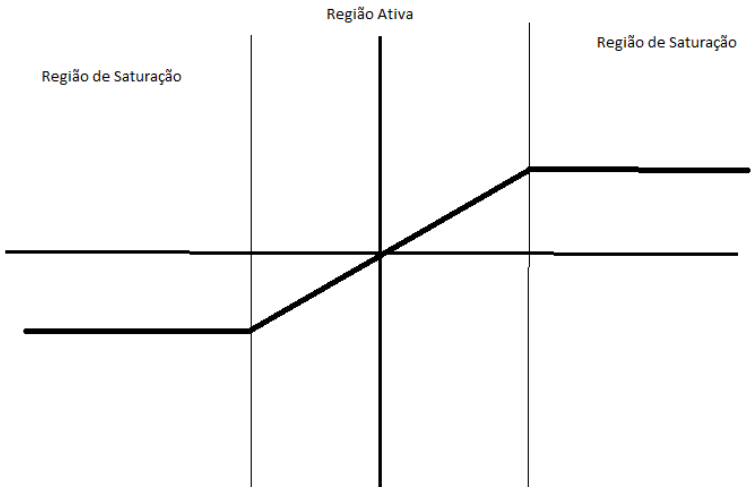

Figura 6 Gráfico porta XOR

Outra forma de fazer um detector de fase é utilizando um *mixer*. Contudo, não entraremos em detalhes a respeito desta técnica por se encontrar fora do escopo do trabalho.

Em PLL's, um simples detector de fase não é utilizado, visto que eles não conseguem detectar a frequência do sinal de entrada. Assim, o circuito indicado é o chamado *tristate phase detector,* mais conhecido com PFD. A Figura 7 mostra a topologia de um PFD.

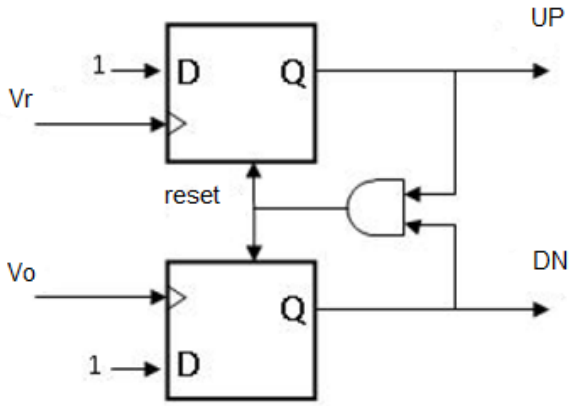

Figura 7 topologia PFD (Dabhi, 2014)

Essa topologia é composta de dois *Flip-Flops* do tipo "D", onde em um deles entra o sinal de referência e no outro entra o sinal do VCO. O circuito

também conta com uma porta AND, onde a saída dessa porta está conectada ao *reset* dos dois *Flip-Flops.* Já as entradas da porta AND são conectadas a saída do PFD.

O funcionamento do bloco é explicado a seguir. Caso a fase do sinal de referência esteja adiantada em relação ao VCO o circuito produz uma saída UP, que faz com que o VCO se adiante, avançando a fase para que ela possa acompanhar o Vr. Caso Vr esteja atrasado em relação ao VCO o circuito produz um sinal DN, que faz com que o VCO atrase sua fase. Se o sinal estiver em fase o circuito não gera nenhum sinal na entrada do *Charge Pump*.

O sinal gerado por um PFD é digital. Entretanto, o VCO não reconhece este tipo de sinal, sendo necessário utilizar um circuito que faça a conversão digital para analógico novamente. Para isso, é utilizado o *charge pump* que será detalhado mais a frente. A Figura 8 mostra o diagrama de estados de um PFD.

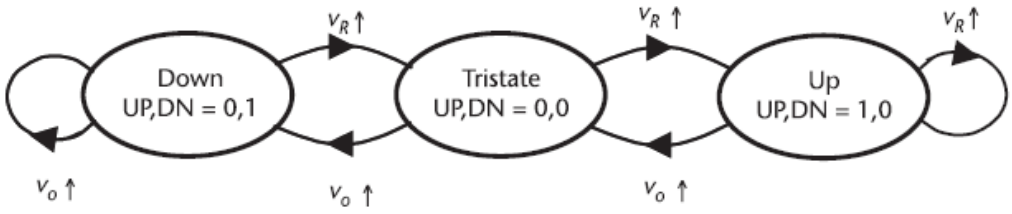

Figura 8 Diagrama de estados PFD (Rogers, 2006)

As transições para outro estado acontecem nas bordas de subida de $Vo\ e\ Vr$. O estado *tristate* é acionado quando os sinais então em fase. Outro ponto importante é analisar o modo de operação do PFD, como pode ser visto da figura 9.

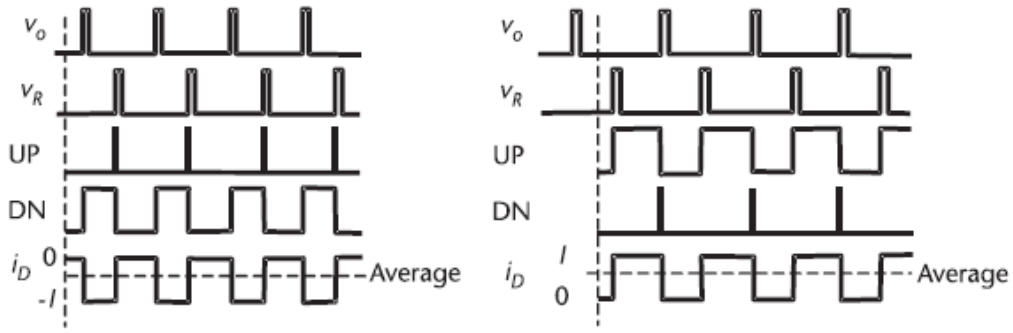

Figura 9 Operação de um PDF (a) antes de Vo (b) antes de Vr (Rogers,2006)

Existem dois modos de analisar os gráficos gerados por um PFD. Quando o sinal Vr está adiantado o valor de $Up = 0$ e o valor de $Dn = 1$, de modo que esse sinal alerta o VCO para atrasar a fase do seu sinal. Já quando Vr está adiantado em relação à Vo, o valor de $Up = 1$ e de $Dn = 0$, e a corrente do circuito é proporcional à saída Up.

2.3. *CHARGE PUMP*

A principal função desse circuito é converter o sinal digital recebido da saída do PFD para um sinal analógico. Além disso, o *Charge Pump* é responsável por inserir ou retirar carga do *Loop Filter.* Ele deve ter um projeto que minimize o ruído de fase, que prejudica o funcionamento do VCO.

Diante das pesquisas, e objetivando aplicações em alta frequência a topologia escolhida foi a de *Charge Pump Single-Ended,* composto por dois espelhos de corrente e um amplificador de um estágio *cascode* como visto na Figura 10.

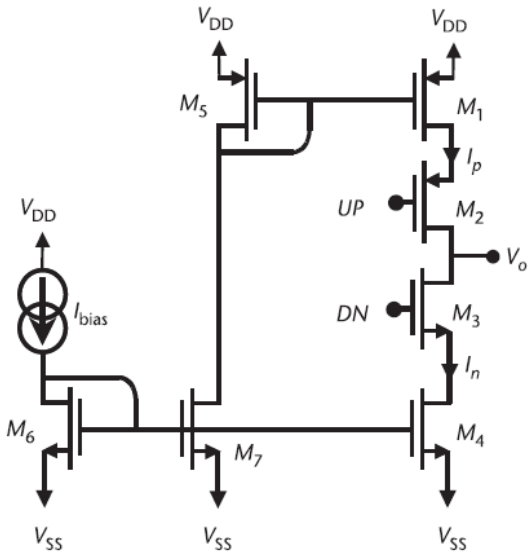

Figura 10 Figura 10: Topologia Charge Pump (Rogers, 2006)

Analisando a Figura 10 observa-se que os transistores M1 e M4 são utilizados como fonte de corrente, enquanto os transistores M2 e M3 funcionam como chaves, cuja entrada está conectada ao PFD, e a saída, ao *Loop Filter.*

Para o funcionamento adequado é importante garantir que os transistores M1 e M4 operem na região de saturação. Assim, temos que garantir que:

$$\mathrm{vdssat} > vgs - vth \tag{2.4}$$

Para os transistores PMOS a equação pode ser descrita como:

$$|\mathrm{vdssat}| > |vgs| - |vth| \tag{2.5}$$

Assim, garantimos que os transistores então em saturação. Considerando que o transistor está em saturação, a equação da corrente é:

$$\mathrm{id} = \frac{1}{2}\mu\mathrm{cox}\frac{\mathrm{W}}{\mathrm{L}}(\mathrm{vgs} - \mathrm{vth})^2 \tag{2.6}$$

Onde W é a largura, L o comprimento, μ é a mobilidade dos elétrons e Cox e a capacitância por unidade de área. Assim, podemos substituir o valor de $(vgs - vth)$ e encontrar outra equação utilizada para calcular $vdssat$, na equação 2.6.

$$vdssat = (vgs - vth) = \sqrt{\frac{2idL}{\mu coxW}} \tag{2.7}$$

De forma geral o *Charge Pump* funciona bem com uma grande razão $\frac{W}{L}$ e uma baixa corrente de dreno (Rogers, 2006). Isso possibilita uma grande faixa de sintonia para ser usada pelo PLL. Além de manter o circuito em saturação é importante aumentar a impedância de saída do mesmo. Em circuitos ideais o valor da impedância deve ser infinito. Entretanto, isso não é possível. Assim, temos que garantir valores elevados da impedância. A impedância de saída é dada por pela equação 2.7.

$$Rds = \frac{1}{\lambda ids} = \frac{L}{ids} \tag{2.8}$$

A solução encontrada na topologia foi utilizar um amplificador cascode de um estágio, aumentando assim consideravelmente o valor da resistência de saída.

2.4 ESTUDO LOOP FILTER

O *Loop Filter* é um filtro passa baixas de transimpedância convertendo a corrente em Tensão enquanto filtra. Possui duas funções básicas em um PLL. A primeira é filtrar as frequências e definir a largura de banda do mesmo. A outra função é transformar a corrente gerada na saída do *Charge Pump* em uma tensão, visto que os VCO são acionados por uma tensão de entrada. O circuito também é responsável pela estabilidade de todo o PLL.

A resposta em frequência conjunta do PFD, Charge Pump e Loop filter é determinada praticamente pelo LP (Rogers, 2006). Desta forma, podemos analisar a resposta em frequência considerando apenas a função de transferência do mesmo. No presente trabalho foi utilizada uma topologia de filtro passas baixas passivo de segunda ordem construído com capacitores. A figura 11 mostra a topologia escolhida.

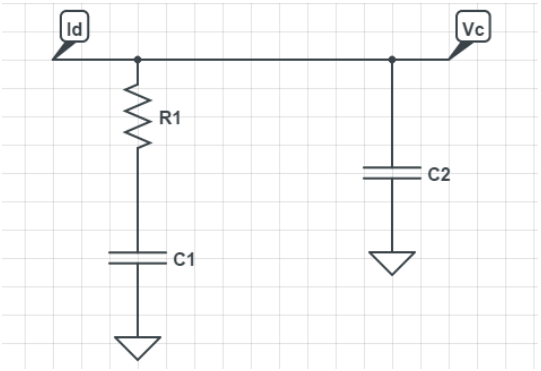

Figura 11 Topologia Loop Filter

Podemos deduzir a função de transferência do circuito, encontrando a admitância Y do circuito:

Onde temos:

$$Cs = \frac{C1C2}{C1 + C2} \tag{2.10}$$

Assim.

$$Y = \frac{s(C1 + C2)(1 + sCsR1)}{sC1R1 + 1} \tag{2.9}$$

Utilizando Y, pode-se determinar a tensão Vc na entrada do VCO é dada por:

$$Vc = \frac{id}{Y} = \frac{1(\theta r - \theta o)(1 + sC1R)}{2\pi s(C1 + C2)(1 + sCsR)} \quad (2.9)$$

Onde temos:

$$Cs = \frac{C1C2}{C1 + C2} \quad (2.10)$$

Outra equação importante para a analise do LP é o ganho de malha aberta e fechada. Suas deduções podem ser encontradas em Razavi e Behzad (1998) e Srinivasan (2006). O ganho de malha aberta é dado pela equação (2.11). Onde Kvco é o ganho do VCO e N é o fator de multiplicação da frequência.

$$G(s) = \frac{Kvco.Id}{2\pi NC1}.\frac{(1 + \frac{s}{Wz1})}{s^2.(1 + \frac{s}{Wp1})} \quad (2.11)$$

Onde $Wz1 = \frac{1}{R1C1}$ e $Wp1 = \frac{1}{R1C2}$.

É possível notar pela função que aparecem dois polos no eixo imaginário, o que torna a função potencialmente instável. Assim é necessário acrescentar um zero na função de transferência. O zero é obtido com o resistor em série com o capacitor C1.

Outra equação importante é a ganho em malha aberta, que se caracteriza por ser uma função de transferência de segunda ordem. Que é obtida a partir da equação da variação da tensão de entrada do VCO (2.8). A equação é dada por.

$$H(s) = \frac{\frac{1}{2\pi C1}(RC1 + 1)Kvco}{S^2 + \frac{Icp}{2\pi C1N}Kvco} \quad (2.12)$$

Da equação (2.12) é possível achar a frequência natural do circuito dado por (2.13).

$$\omega n = \sqrt{\frac{IcpKvco}{2\pi C1N}} \quad (2.13)$$

O fator de amortecimento é dado pela equação (2.14).

$$\zeta = \frac{R}{2}\sqrt{\frac{IcpC1Kvco}{2\pi N}} = \frac{\omega nRC1}{2} \tag{2.14}$$

Por questões de estabilidade do sistema, o fator de amortecimento escolhido é igual a 0.7.

A largura de banda pode ser encontrada pela equação (2.15).

$$\omega 3dB = \omega n\left[2\zeta^2 + 1 + \sqrt{(2\zeta^2+1)^2+1}\right]^{0.5} \tag{2.15}$$

É desejável maximizar a largura de banda, para dimunuir o tempo de acomodação do sistema ou melhorar a resposta do ruído se fase. O tempo de acomodação de um sistema criticamente amortecido é dado por:

$$Tlock = \frac{1}{\zeta\omega n}\ln\left(\frac{\Delta f}{af_0}\right) \tag{2.16}$$

Como esses parâmetros pode-se fazer um esboço da resposta em frequência do *Loop Filter,* tanto em magnitude como em fase como se pode ver na Figura 12.

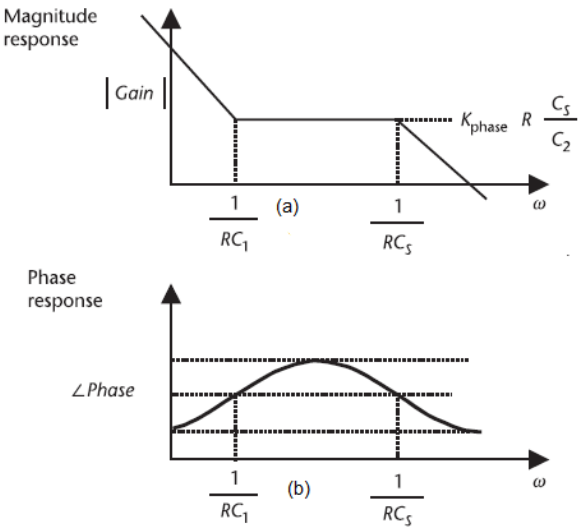

Figura 12 Resposta em frequência PFD, Charge Pump e Loop Filter (a) Magnitude (b) Fase (Rogers, 2006)

Pode-se notar pelo gráfico e pela função de transferência que o circuito possui dois polos e um zero e se comporta com um filtro passa baixas. O capacitor C2 foi adicionado ao LF para agregar outro polo ao sistema para reduzindo o *glitch.* Vale notar que se C2 <0.1C1 o filtro ainda pode ser tratado como de segunda ordem.

2.5. RUÍDO

O PLL é caracterizado por requerer uma alta precisão no sincronismo de fase. Logo, em sua modelagem devem ser levados em consideração os ruídos presentes no sistema, além de métodos eficientes para quantificar e reduzir os níveis de ruído do PLL.

2.5.1. RUÍDO DE FASE

O ruído de fase é o mais prejudicial para o funcionamento do sistema PLL, visto que interfere diretamente no desempenho em frequência e no espalhamento da frequência do sinal. Logo, é necessário reduzir a intensidade do ruído de fase a níveis que sejam considerados aceitáveis para o funcionamento adequado do PLL, produzindo um sinal mais puro e próximo do ideal (Cardoso,2009). A Figura 13 mostra diferença entre um sinal ideal e o sinal real gerado pelo PLL.

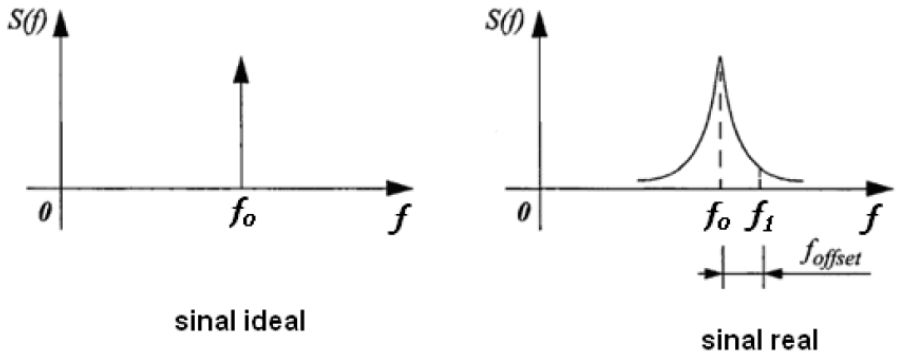

Figura 13 Ruído de fase aplicado a um sinal Ideal (Cardoso, 2009)

Nota-se que quanto menor o ruído de fase do sinal real, menor será o *offset* e o comportamento do circuito será mais próximo do ideal.

O ruído de fase é caracterizado por variações aleatórias de fase, frequência e deformação da forma de onda na saída do PLL, podendo ser quantificado pela magnitude do ruído de fase ou *jitter.* Este pode ser interpretado como uma variação de fase de um sinal periódico, onde o período da n-ésima oscilação é dado por T_n e o intervalo de tempo é dado por Δ_n (Razavi, 2000).

Na literatura existem diversas definições de *jitter,* porém neste trabalho serão citadas as mais relevantes para a análise de ruídos do PLL. A primeira definição é do *jitter* absoluto que é caracterizado pela equação 2.11.

$$\Delta T_{abs}(N) = \sum_{n=1}^{N} \Delta Tn \tag{2.16}$$

A equação 2.16 representa a acumulação do *jitter* no período de N ciclos. O cálculo do *jitter* absoluto não é muito utilizado para osciladores livres. Entretanto é muito utilizado em um PLL, ele é muito útil visto que o mesmo reseta o *jitter* a cada interação. Desta forma, é possível calcular o ruído gerado no VCO.

A segunda definição é dada pelo ruído de fase cíclico, que é calculado em apenas um ciclo onde é aferido a variação entre cada período e o período médio, e é dado pela seguinte equação:

$$\Delta T_c = \lim_{n\to\infty} \sqrt{\frac{1}{N}\sum_{n=1}^{n} \Delta Tn^2} \tag{2.17}$$

A fórmula representa a média de flutuações da frequência de referência. Esse *jitter* é fundamental para o cálculo do ruído de fase gerado em blocos como o PFD e o *Charge Pump*. O cálculo do ruído é fundamental para medir os parâmetros de desempenho do PLL. A Figura 14 mostra o sinal com o *jitter*.

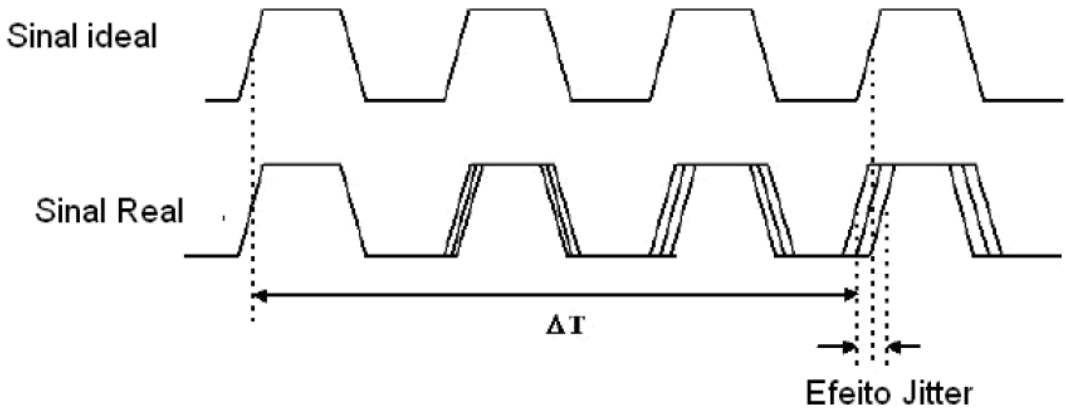

Figura 14 Efeito jitter na geração do sinal (Cardoso, 2009)

Para uma análise completa sobre a influência do ruído de fase é necessária a utilização de software de modelagem, como o CADENCE, e a utilização da linguagem de hardware Verilog-MAS, que serão vista mais adiante.

3. METODOLOGIA DE PROJETO

Em projetos eletrônicos, é importante que se utilizem técnicas e metodologias precisas. Pode-se citar três razões que justificam a importância de se seguir metodologias claras: permitir que o cumprimento dos requisitos de projeto possam ser assegurados de maneira formal à medida que o projeto avança; potencializar o uso de ferramentas de automação das tarefas (como as ferramentas CAD), visto que cada etapa é conhecida e pode ser dividida em pequenas tarefas automatizáveis, e facilitar a comunicação entre as equipes de desenvolvimento, visto que por meio dessas ferramentas é possível dividir as responsabilidades, de forma, a permitir que cada integrante da equipe tenha conhecimento do que está acontecendo nas demais partes do projeto, o que possibilita um conhecimento geral do que esta sendo desenvolvido (Wolf, 2008).

A metodologia escolhida deve ser capaz de três ações básicas: especificação/modelagem, validação e síntese.

A modelagem ou especificação do sistema começa no processo de descrição dos componentes e suas funções e termina na descrição de um modelo formal Segundo Zurita, um modelo formal deve conter: especificações funcionais, de forma a demonstrar as relações entre entradas e saídas do sistema; o que o projeto deve satisfazer, levando em conta as entradas e saídas do sistema; índices de desempenho, para avaliar a qualidade do projeto e onde deve ser melhorado; e restrições para os índices de desempenho, ou seja, parâmetros mínimos que devem ser alcançados. Com isso, o processo de modelagem pode seguir esses passos para a implementação de um modelo formal.

O próximo passo é a validação dos resultados obtidos na modelagem de componentes ou sistemas completos. Isso se dá pela análise de requisitos do sistema, verificando se estão corretos, precisos, completos e consistentes ou se precisam de ajustes na modelagem. Nessa etapa são utilizadas simulações com ferramentas de CAD e linguagem de descrição de hardware como o

Verilog-AMS. Desta forma, os projetistas podem verificar possíveis erros no modelo e corrigir-los.

Por fim, depois da validação do modelo, pode-se sintetizar o mesmo. A síntese consiste no processo de aumentar o nível de detalhamento das especificações, diminuindo o nível de abstração do projeto. Nessa etapa, é importante compreender o conceito de níveis de abstração, que é a divisão do projeto em vários níveis onde o nível inferior possui maior detalhamento do sistema que o nível superior. Em projetos de sistemas eletrônicos são utilizados cinco níveis de abstração: funcional; blocos; portas lógicas; transistores e *layout*. Deste modo, é possível criar um fluxo de projeto ordenado.

Para obter mais eficiência e organização é importante ter uma metodologia adequada que possibilite um desenvolvimento rápido e eficiente. A literatura fornece as seguintes alternativas: *Bottom-Up e Top-Down*

3.1 BOTTOM-UP

Na metodologia *Bottom-Up,* o projeto se inicia com a descrição detalhada dos componentes. Esses são conectados para formar subsistemas, e esses, por sua vez formam o sistema completo. Nessa metodologia, os componentes são armazenados em bibliotecas, com o objetivo de serem utilizados em outros níveis de abstração e esse passo é repetido em outras fases do projeto. Uma das vantagens da metodologia é a separação dos níveis de abstração em suas determinadas bibliotecas, o que pode facilitar a organização do projeto. No fim todos os componentes são unidos para formar o sistema em nível de transistores.

Contudo, a metodologia só é considerada vantajosa em projetos mais simples e com um número pequeno de blocos, visto que para projetos com uma grande quantidade de blocos, a verificação do sistema torna-se demorada e complexa, o que pode torna-la inviável. Desta forma, o número de simulações e verificações do sistema completo são reduzidas e alguns problemas de projeto podem passar despercebidos. Outro problema sistemático desta metodologia é a dificuldade de comunicação entres as equipes de trabalho, o

que pode acarretar em erros de projeto, visto que o grupo não possui uma visão completa do sistema. Além desses erros, também é comum ocorrerem erros de arquitetura, causando a necessidade de reprojetar partes do sistema, o que acarreta em atrasos e custos adicionais. Diante de todas as desvantagens citadas, nota-se que a metodologia Buttom-Up não é a mais eficiente em desenvolvimento de circuitos integrados, sendo necessário utilizar outra metodologia mais eficiente.

3.2 TOP-DOWN

Nas ultimas décadas houve um nítido aumento de projetos de sistemas integrados mistos, ou seja, que possuem tanto parte analógica quanto a digital. Entretanto, os projetos digitais e analógicos de sistemas possuem peculiaridades, sendo desenvolvidos de formas distintas. Nos grandes projetos existem equipes de projetistas para as partes digitais e analógicas. Entretanto em diversos projetos nota-se que o tempo de desenvolvimento da parte digital segundo Roll Collet, é de três a sete vezes inferior ao desenvolvimento da parte analógica do sistema, desta forma, o subsistema analógico se tornava responsável pelo aumento do tempo do projeto. Essa demora pode acarretar até mesmo que o sistema se torna obsoleto quando produzido em série. Assim, é necessário desenvolver metodologias de trabalho que permitam que projetos analógicos sejam desenvolvidos de maneira rápida e eficiente.

Para solucionar esse tipo de problema crônico foram desenvolvidos métodos de projetos semelhantes ao usados na área digital, onde se utiliza até hoje linguagem de descrição de Hardwares (HDL) e programas de simulação mista. Além de metodologias de projeto que possibilitem a organização adequada do fluxo de trabalho. Por cumprir todos estes requisitos, a metodologia *Top-Down* é a mais utilizada em fluxos de projetos de circuitos integrados analógicos.

Diferente das práticas vigentes antes da utilização sistemática da metodologia *Top-Down,* o projeto se inicia com a formulação geral das características finais do projeto, sem detalhes de como será implementado. À medida que o projeto avança, o sistema vai sendo definido até ser descrito em

nível de transistor. Vale ressaltar que um nível de abstração só é iniciado com o término do nível superior como mostra a Figura 15.

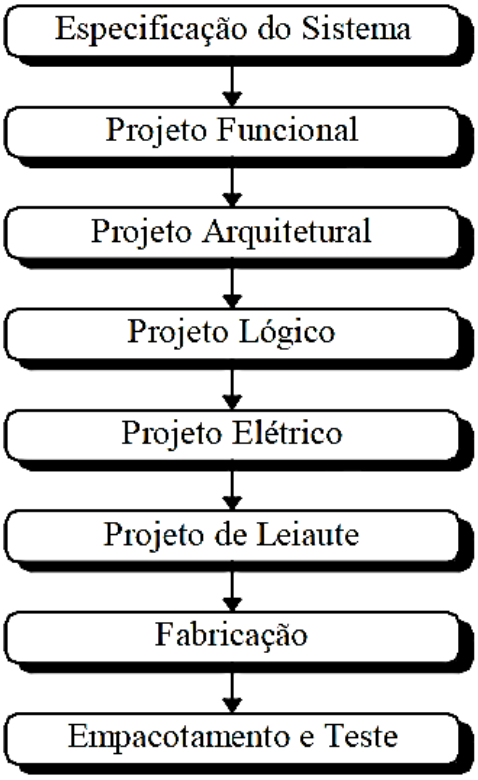

Figura 15 Ciclo de projeto de uma CI com topologia Top-Down

A metodologia é considerada intuitiva e possui uma rápida assimilação pelos projetistas, além de possuir a vantagem de permitir um bom grau de automação, com a utilização de softwares de simulação como o CADENCE e de linguagens de descrição de *hardware* como o Verilog-AMS. Desta forma, é possível reduzir o tempo de projeto, além de minimizar erros cometidos pela equipe. Nesta metodologia são empregados os chamados estimadores, ferramenta que auxiliam na obtenção das características do sistema como consumo, desempenho e área ocupada dentre outros parâmetros relevantes para o sistema. Esses métodos e ferramentas tem efeito direto no número de interações necessárias, diminuindo consideravelmente o tempo se comparado à topologia *Bottom-Up.* Assim, o projeto é realizado em um prazo e custo de desenvolvimento menores.

Para a otimização dos resultados a metodologia deve seguir modelos rigorosos, tendo em vista o aumento da efetividade. Devem existir planos de modelagem para identificar e especificar testes de desempenho do sistema e para criar modelos comportamentais dos blocos. No início do plano é característico que sejam criados diferentes modelos para cada bloco, escolhendo o mais fiel para representar determinado bloco. Os modelos desenvolvidos não precisam ter muitos detalhes e nem prever todas as variáveis possíveis, visto que o objetivo é fazer modelos comportamentais e não estruturais. Assim, um detalhamento maior pode não trazer vantagens relevantes, além de atrasar o desenvolvimento do projeto.

Seguindo essas regras, pode-se reduzir o tempo gasto descrevendo modelos, o que possibilita uma rápida passagem para a etapa de simulação em nível de transistores. Entretanto, nem sempre é possível fazer modelos comportamentais precisos, de forma que em alguns casos é necessário o desenvolvimento de um modelo estrutural que descreve fielmente o bloco, visto que, é necessário atestar o seu comportamento. Feito isto, é possível fazer o planejamento e a validação dos modelos para serem usados em nível de transistor.

Outra vantagem da metodologia é a utilização de simulações de sinais mistos, visto que mesmo que um bloco funcione adequadamente quando projetado, isso pode não acontecer quando no sistema completo. Desta forma, a simulação de sinais mistos verifica se os blocos estão se comportando como o esperado no sistema completo.

Para fazer a simulação é necessário que o modelo de blocos seja substituído por um modelo em nível de transistor, o que possibilita ver os efeitos e imperfeições do bloco sobre o sistema. Este tipo de simulação é a única forma viável para a verificação de grandes sistemas de sinais complexos, como o PLL, tornando sua utilização fundamental.

4. LINGUAGENS DE DESCRIÇÃO DE HARDWARE

Este capítulo apresenta o conceito da linguagem de descrição de hardware, com especial atenção ao Verilog-AMS, visto que esta será utilizada na modelagem dos blocos desenvolvidos neste documento. Serão abordados os benefícios de sua utilização quando utilizada no desenvolvimento de projetos de circuitos integrados, permitindo a verificação e descrição dos diversos comportamentos dos circuitos.

4.1. VISÃO GERAL

As linguagens de descrição de hardware foram desenvolvidas no início dos anos 80, com vista a auxiliar o desenvolvimento de hardware, bem como descrever funcionalidades e aplicações. Ou seja, tratavam-se de linguagens com suporte semântico e sintático para modelagem do comportamento temporal e espacial do hardware. Esses dispositivos possuem diversos blocos, onde a partir desses, é possível construir o sistema completo. Contudo, testar o sistema completo muitas vezes é inviável, dada a infinidade de componentes, tornando necessário desenvolver linguagens que possibilitem a simulação do sistema, antes mesmo de sua produção.

As primeiras linguagens de hardwares (HDL) desenvolvidas e amplamente difundidas foram o VHDL e o Verilog, sendo este desenvolvido entre 1984 e 1985 pela *Gateway Design Automation*. Em 1989, a *Gateway* foi comprada pela *CADENCE Design System,* que tornou o Verilog uma linguagem de domínio publico regulado pelo padrão IEEE 1364 – 1995. O VHDL foi desenvolvido por empresas contratadas pelo exército americano para projetos desenvolvidos pelo mesmo e foi padronizada pela IEEE em 1987 (Midorikawa, 2001).

Uma HDL pode ser usada na descrição do sistema para os diversos níveis de desenvolvimento do circuito. Partindo de uma descrição em alto nível, ela pode ser usada para particionar o sistema em descrições de níveis mais baixos durante o processo de desenvolvimento. A descrição final deve contar com componentes primitivos e blocos funcionais.

A grande razão para o uso de HDL's é a síntese lógica. A descrição de componentes é usada como uma ferramenta de síntese para a geração automática de um circuito digital. Além disso, essas ferramentas incluem uma etapa de armazenamento, da lógica combinatória e da estrutura de conexão dos componentes (*netlist*). A figura 16 mostra o diagrama das principais etapas da síntese lógica.

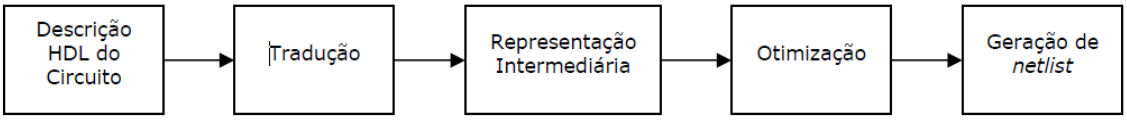

Figura 16 Fluxo das etapas de alto nível da síntese lógica (Midorikawa, 2001).

Mesmo com todas as vantagens das HDL's citadas, elas só possuem capacidade de simulação e compilação de circuitos digitais. Logo é necessário estender os conceitos de HDL no intuito de ampliar as possibilidades de modelagem para sistemas analógicos e sinais mistos. Para isso foram desenvolvidas as linguagens VHDL-AMS e Verilog-AMS. Na próxima seção o Verilog-AMS será abordado mais detalhadamente.

4.2. VERILOG-AMS

O Verilog-AMS é uma linguagem de descrição de hardware derivada do Verilog. Nela é possível simular sistemas digitais e analógicos, além de ser possível fazer simulações de sinais mistos com o próprio nome deixa claro (AMS – *Analog and Mixed Signal*). Desta forma, é possível realizar a descrição completa de sistemas.

O padrão foi finalizado em 2005 a partir da fusão entre o Verilog-HDL e o Verilog-A (Figura 17), responsável por descrever sistemas digitais e analógicos. Além disso, a linguagem também permite a simulação de sinais mistos, o que a deixa ainda mais completa (Kundert, 2004).

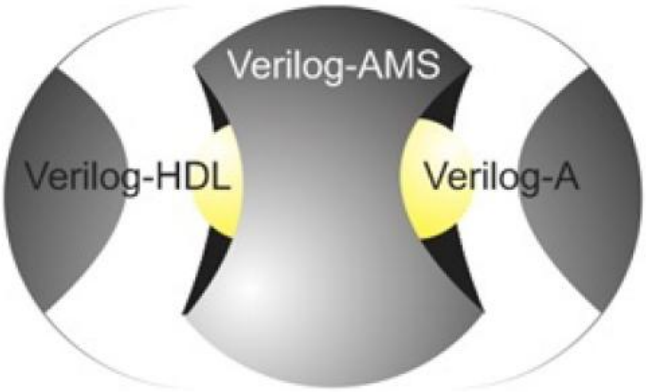

Figura 17 Universo Verilog-AMS (Nascimento, 2010)

A linguagem possibilita especificar sistemas desde as fases iniciais até as fases mais avançadas. Também possibilita a simplificação e padronização da simulação, visto que podem ser feita simulações digitais, analógicas e de sinais mistos. O objetivo da padronização é testar todas as funcionalidades do sistema em uma única plataforma, possibilitando que projetos possuam apenas um fluxo de desenvolvimento. Sendo assim, a linguagem é útil tanto para o design digital quanto para o analógico.

4.2.1 CARACTERISTICAS

O Verilog-AMS possibilita que projetos sejam decompostos hierarquicamente e, além disso, podem-se adicionar modelos às simulações como resistores, capacitores, indutores, transistores, dentre diversos outros modelos. Os códigos escritos em Verilog-AMS seguem um padrão, onde no corpo principal encontra-se a definição dos parâmetros na escala do tempo, instanciação de bibliotecas e descrição de sistemas.

Outra característica relevante é a capacidade de descrever diferentes tipos de comportamento, tais como, componentes lineares, não lineares, integro diferencial e simulações de eventos analógicos. A estrutura da linguagem permite que o projetista defina o fluxo dos sinais, além de avaliar aspectos elétricos e mecânicos.

Outra característica importante da linguagem Verilog-AMS é que ela facilita o uso da metodologia *Top-Down*. Também é possível executar automaticamente a interface entre modelos digitais e analógicos.

O Verilog-AMS possui um conjunto de estruturas que tipificam as características da linguagem, e possibilitam a descrição de componentes digitais e analógicos. Na Figura 18 é mostrado um exemplo de como utilizar o Verilog-AMS, principais estruturas da linguagem são comentadas.

```
// Junction diode

`include "disciplines.vams"

module diode (a, c);
    parameter real is=10f from (0:inf);      // saturation current (A)
    parameter real tf=0 from [0:inf);        // forward transit time (s)
    parameter real cjo=0 from [0:inf);       // zero-bias junction capacitance (F)
    parameter real phi=0.7 exclude 0;        // built-in junction potential (V)
    inout a, c;
    electrical a, c;
    branch (a, c) res, cap;
    real qd;

    analog begin
        I(res) <+ is*(limexp(V(res)/$vt) – 1);
        qd = tf*I(res) – 2*cjo*phi*sqrt(1 – V(cap)/phi);
        I(cap) <+ ddt(qd);
    end
endmodule
```

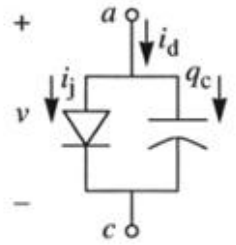

Figura 18 Módulo que descreve a junção de um diodo (Kundert, 2004).

A seguir, são apresentadas suas principais estruturas:

- Disciplinas: são compostas por sinais físicos relacionados, ou seja, sinais de mesma natureza física como: Elétricos, mecânicos, lógicos, dentre outros.
- *Nature:* é a definição quantitativa de grandezas do domínio analógico. Possui três informações básicas, divididas em unidades básica de grandeza, nome para quando ela for utilizada e um número que indica o tamanho da mesma;
- *Module:* é a unidade básica de projeto;

- *Parameter: é o* identificador de um valor constante, não devendo mudar no decorrer da simulação e podendo ser real ou inteiro;
- *Port:* são pontos onde as conexões podem ser feitas. Existem três tipos, *input, output* e *inout.;*
- *Include: é uma* derivativa de compilação de arquivos externos;
- *System Task:* são funções que retornam valores reais e tem como entrada os valores atribuídos aos parâmetros;
- *Wreal:* Sinal que permite a transformação de um sinal elétrico em uma variável real. Possui a vantagem de reduzir o tempo de simulação dos blocos em simuladores digitais.

5. PROJETO, SIMULAÇÕES E RESULTADOS

O trabalho consistiu no projeto de três blocos de um *Phase Looked Loop* (PLL) desenvolvido na tecnologia TSMC 0.18µ e frequência de referência de 5Mhz. A saída do PLL fornece um sinal com frequência igual a 2,4GHz. Utilizando divisores este sinal em 2.4 GHz é dividido para que possa ser comparado com a frequência de referência (5MHz). Logo os blocos projetados neste trabalho realizam esta comparação e operam na frequência de referência. Optou-se por utilizar topologias já testadas e modificá-las para a realidade do projeto, visando suprir as exigências do sistema *ZigBee*.

Partindo das topologias escolhidas e modificando-as para obter um melhor resultado, foram testadas duas topologias de *Flip-Flop D* e sendo escolhida a de melhor desempenho para compor o PFD. Já a topologia inicialmente proposta para o *Charge Pump* foi mantida sofrendo pequenas alterações para se enquadrar nos parâmetros do projeto. O *Loop Filter* não sofreu alterações.

A seguir serão descritos os blocos projetados e demonstrados os resultados obtidos.

5.1 RESULTADOS PFD

O PFD consiste em um bloco que detecta a diferença de fase e frequência entre um sinal de referência e outro vindo da malha de realimentação. A diferença de fase ativa os sinais UP ou DN do PFD. A saída UP é ativada quando o sinal da malha de realimentação está adiantado. Já a saída DN é ativada quando o sinal de referência está adiantado. As saídas do PFD são as entradas *do Charge Pump,* controlando a corrente da saída do mesmo.

Para o desenvolvimento do PFD são utilizados dois *Flip-Flop D* (ffD) e uma porta lógica AND. O circuito é construído de forma que as saídas do ffD são conectadas as entradas da porta AND e a sua saída é conectada no *reset* dos ffD. Já os sinais $Vref$ e $Vvco$ são conectados as entradas CK do ffD, e por fim é necessário conectar a entrada D em VDD. A Figura 19 mostra os blocos do

PFD. .

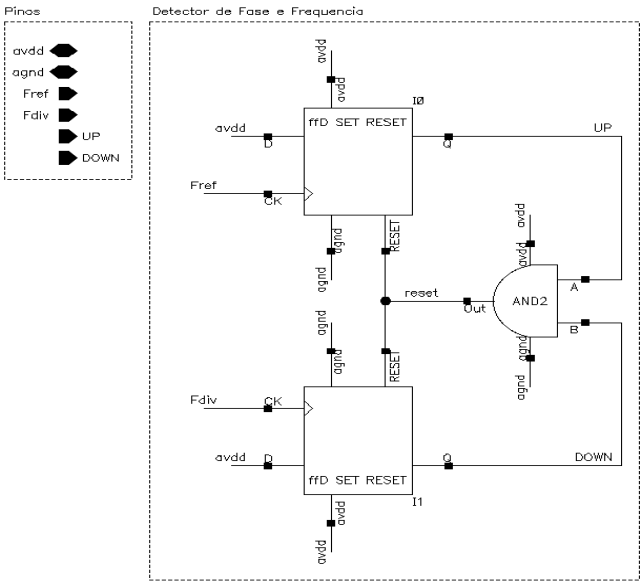

Figura 19 PFD.

A borda de subida de $Vref$ ativa a saída UP, assim aumentando a carga na saída do CP. Analogamente uma borda de subida de $Vvco$ ativa o saída DN, reduzindo desta forma a carga na saída do CP. Quando a porta AND é ativada o *reset* do ffD é ativado, forçando UP e DN a irem para nível baixo.

Foram testadas duas topologias de ffD, e escolhida a que teve o melhor desempenho. Foi utilizada uma variante modificada da topologia TSPC, apresentada na Figura 20. O circuito possui sinal de *reset* essencial para as saída UP e DN irem para nível baixo. A entrada “D” está conectada em VDD. Quando o *reset* e o CK estão em nível baixo o nó A conecta-se a VDD por meio dos transistores m7 e m8. Já quando CK é invertido o nó B é conectado ao terra pelos transistores m3 e m4 e a saída do ffD vai para nível alto permanecendo assim até o *reset* ser acionado novamente. Quando acionado o nó A é conectado ao terra e o nó B muda para nível lógico alto, e a saída vai para nível baixo..

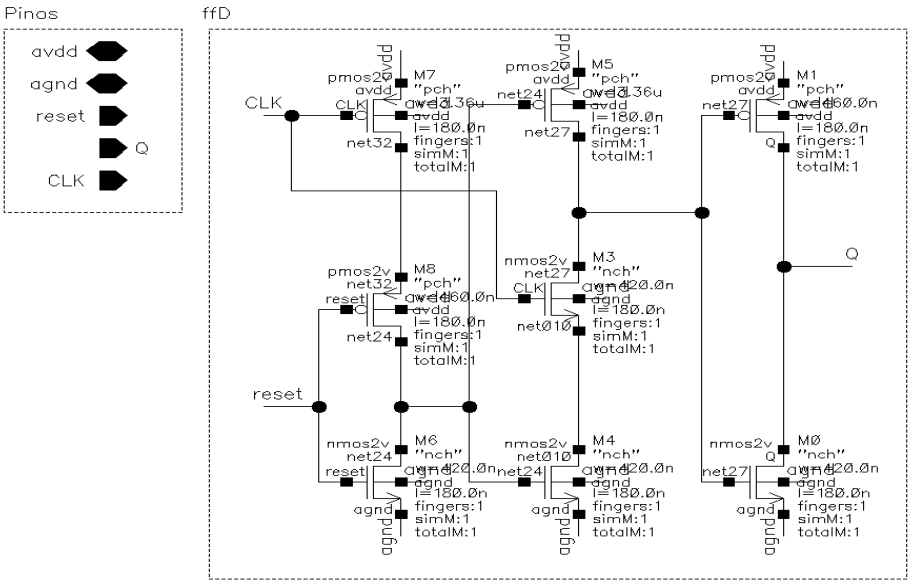

Figura 20 Topologia TSPC modificada.

Já a outra topologia testada foi a de um ffD biestável. O projeto do mesmo consiste na utilização de portas lógicas inversoras, NAND e portas de transmissão ou bilaterais. As chaves bilaterais consistem na união de transistores pmos e nmos (Figura 21).

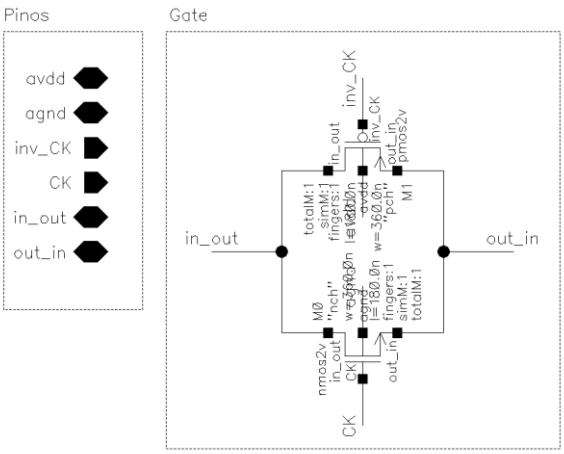

Figura 21 Topologia porta bilateral.

O ffD biestável, apresentada na Figura 22 se comporta da seguinte forma o inversor N1 e o buffer B1 são responsáveis pelos sinais CK e inv_CK (o buffer e o inversor têm atrasos idênticos). Inicialmente deve-se assumir que CK e a

entrada D então em níveis baixos e consequentemente inv_CK está em nível alto. Deste modo, a porta de transmissão TG1 estará em alta impedância, impedindo assim que a entrada D exerça qualquer efeito sobre o ffD.

Quando CK está em nível alto e inv_CK está em nível baixo, TG1 irá para nível baixo e D será invertida por N2 de modo que a saída Q irá para nível lógico alto. Já se a entrada D for alterada para nível alto, a borda de subida de CK ativa TG1, que por sua vez transmitirá o sinal para a entrada de N2, fazendo que Q vá para nível baixo.

Enquanto CK estiver acionado qualquer alteração em D será transmitida por TG1. Contudo, no momento da borda de descida TG1 é desligado e TG2 é ativado, isolando as portas N1 e N2. Assim pode-se afirmar que o ffD funciona com borda de subida.

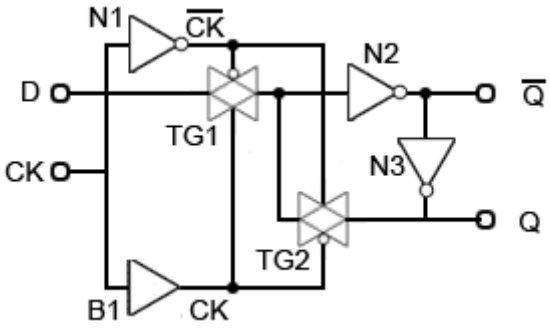

Figura 22 topologia base ffD biestável (2015, learnabout-electronics).

Contudo, essa topologia não possui estradas de SET e nem de RESET. Desta forma, foi necessário modificá-la para que essas entradas fossem inseridas no ffD. Para que fosse possível a mudança foi necessário substituir as portas inversoras por portas NAND, exceto a porta inversora N1. A Figura 23 mostra as modificações necessárias para a adição das entradas SET e RESET.

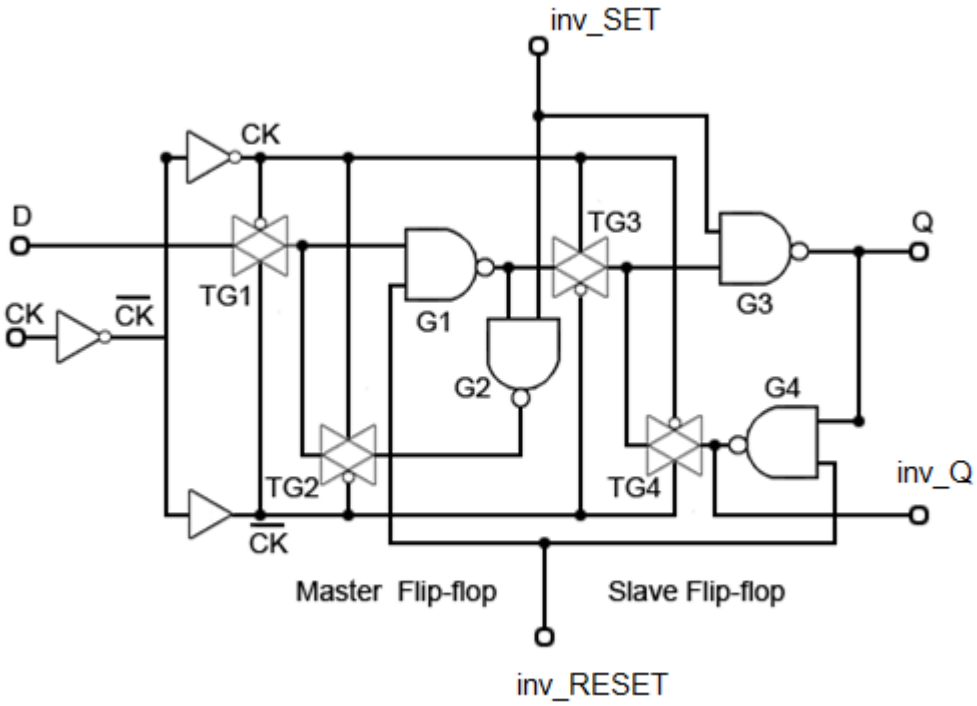

Figura 23 Topologia biestável com SET e RESET.

O circuito funciona da seguinte forma: quando o nível lógico baixo é aplicada inv_SET, a saída G3 (Q) vai para nível lógico alto. Já quando inv_SET = 0 o ffD será desabilitado, forçando que ambas as saídas G3 e G2 a nível alto. Desta forma, não será possível alterar o valor da saída. Com inv_RESET o comportamento é semelhante, apenas mudando as portas de ação que no caso são G1 e G4. A topologia apresentada na figura 23 sofreu algumas modificações visando reduzir o consumo de potência do projeto, além de adaptar a saída inv_RESET para nível alto.

No projeto do PFD a entrada SET não é utilizada, desta forma ela foi suprimida. Outra alteração foi a retirado do buffer. Essa mudança não trouxe problemas para o funcionamento do circuito. Além disso, a saída inv_Q não foi utilizada pois não tem utilidade na construção do PFD . Essas mudanças visaram reduzir o consumo de potência do bloco, além de reduzir o espaço ocupado no *layout.* A Figura 24 mostra o circuito do ffD.

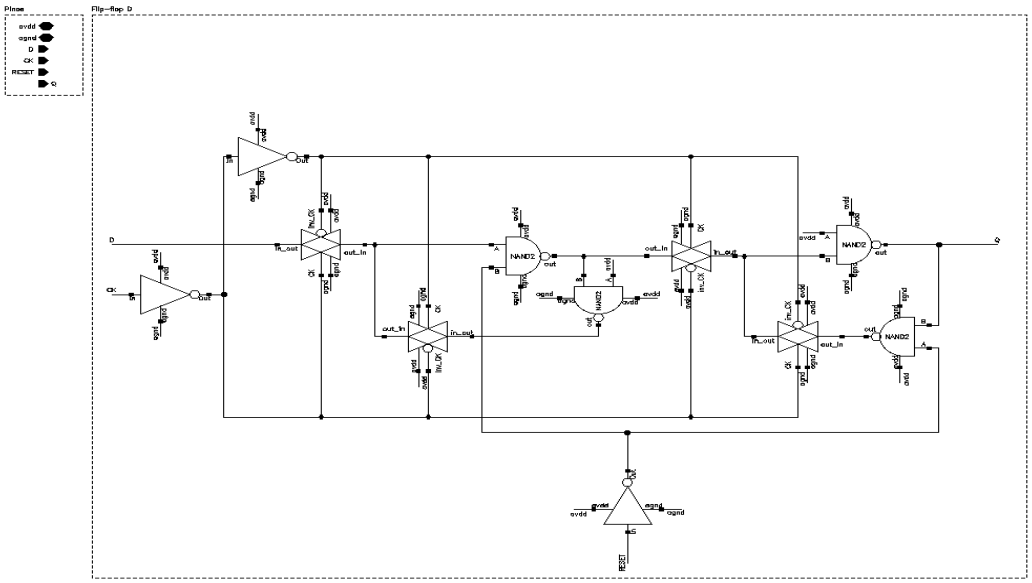

Figura 24 ffD biestável projetado.

Uma das preocupações do projeto do PFD foi de reduzir o consumo de potência de forma a ficar dentro das especificações do sistema. Outra preocupação foi a dimensão dos transistores, que foram construídos mantendo uma relação de $\frac{W}{L} = 2$ possibilitando um *layout* mais homogêneo e a redução da área ocupada pelo PFD.

Comparando as duas topologias, foi escolhida o ffD biestável, porque seu desempenho se mostrou mais adequado. Além disto, foi observada uma falha no projeto com a tecnologia TSPC, visto que o pulso da saída DN não estava chegando ao valor de VDD. Já na topologia biestável não ocorreu esse problema. Uma das desvantagens da mesma é a maior quantidade de transistores, o que fatalmente aumentará o consumo de potência do ffD.

Na Figura 25 é mostrada a simulação do PFD com as duas topologias de ffD. Em (a) o PFD funcionado com TSPC e o (b) a topologia biestável.

(a)

(b)

Figura 25 (a) Simulação PFD com ffD TSPC (b) Simulação PFD com biestável.

.

Para terminar a construção do PFD foi utilizada uma topologia padrão de porta AND construída com uma porta NAND e um inversor. Na construção optou-se deixar o comprimento do canal $L = 180\mu m$ e a largura em $W =$

$360\mu m$, mantendo a proporção citada anteriormente. A simulação e a topologia da porta AND então disponíveis no anexo.

Depois de validadas as portas lógicas e o ffD, foi montado o bloco do PFD, de forma que a saída D do ffD fosse conectado ao VDD.

Como pode ser visto, a saída DN no PFD construído com o ffD TSPC não atinge VDD, podendo desta forma causa o mal funcionamento do PLL. Desta forma, optou-se pela tecnologia biestável, mesmo considerando o fato de este circuito apresentar um maior consumo de potência. Contudo, a potência consumida a mais não ultrapassa os limites especificados para o projeto.

Uma última consideração a respeito o PFD construído com o ffD biestável é que para sua correta simulação as saídas UP, DN e o RESET devem ser configuradas em nível lógico baixo. O Cadence possui opções de simulação onde é possível ativar as condições iniciais através do simulador ADE L em *options* e *initial condition set*.

Depois de ativada as condições iniciais é possível simular o funcionamento do bloco, sendo extraídos do mesmo a corrente de operação e o consumo de potência, que não podendo ultrapassar os 20µW de potência. Tendo em vista o consumo, o PFD é um bloco critico dada a quantidade de transistores. O PFD escolhido tem um consumo de potência de 7.59µW, respeitando as especificações de projeto. A figura 26 mostra o consumo de potência e a corrente drenada polo circuito.

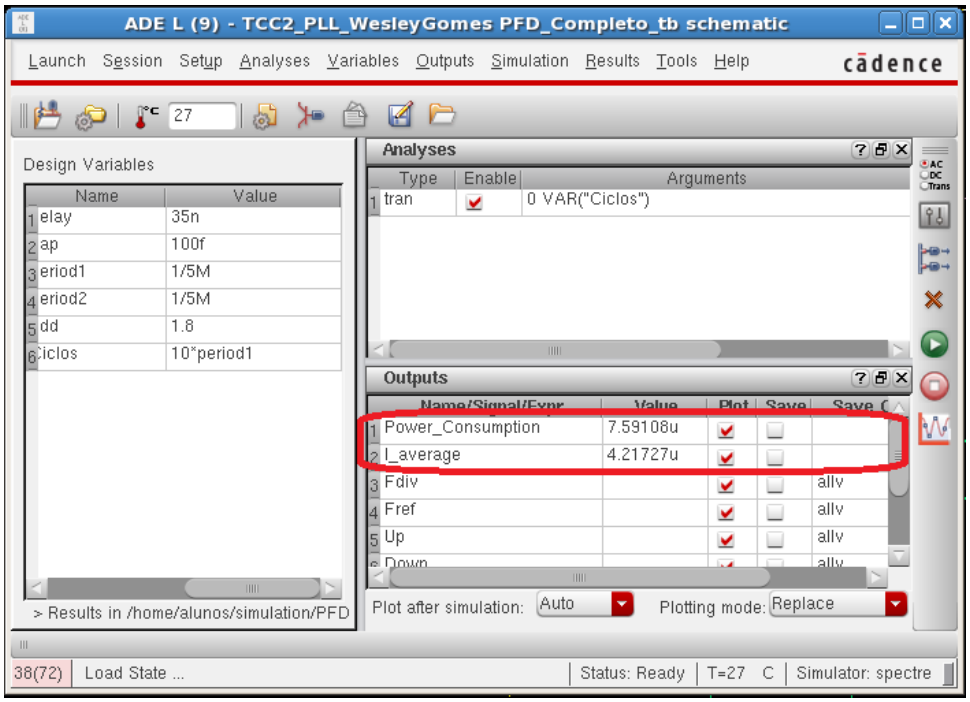

Figura 26 Consumo de potência e corrente.

Para validar o projeto do PFD é necessária ainda a realização de simulações mistas, que consiste na simulação do sistema completo do PLL, onde uma parte dos blocos é modelada linguagem de hardware verilog_AMS e o bloco a ser validado é representado em nível de componentes elétricos. O projeto dos blocos em verilog-AMS não foi realizado neste trabalho. Desta forma, esta simulação foi realizada utilizando o PFD projetado neste trabalho. Foi observado que o comportamento do PLL não foi alterado. Um dado importante é o tempo de acomodação do PLL, visto que o sistema é criticamente amortecido. De acordo com os cálculos feitos no projeto do *Loop Filter*, obteve-se um tempo de acomodação no pior caso de 50µs. Já como pode ser visto na figura 27 o sistema se acomodou com 13.2µs. Com base nesses dados nota-se que o bloco PFD se comportou adequadamente e cumpriu de forma satisfatória sua função no PLL. A Figura 27 mostra o resultado da simulação mista com o PFD.

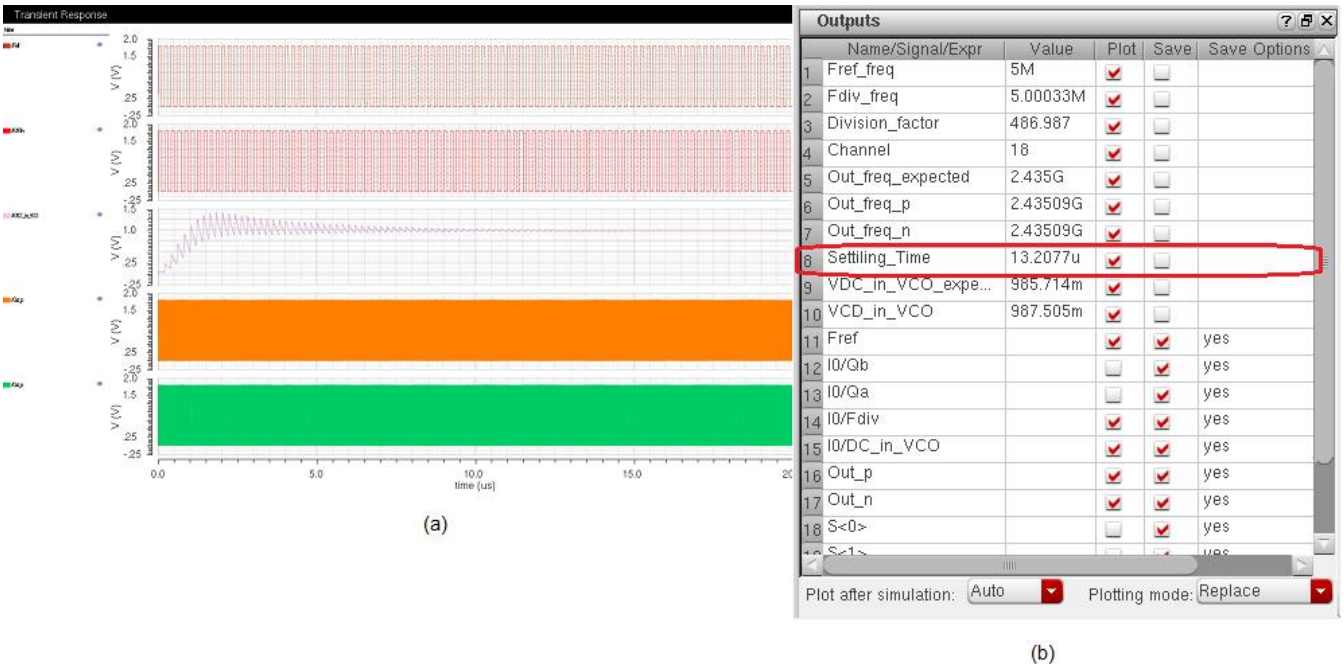

Figura 27 (a)Gráfico da simulação Mista (b) Saídas do PLL com ênfase no tempo de acomodação.

Como pode ser visto no gráfico da Figura 27 (a), o circuito se comportou de forma esperada com a introdução do bloco real do PFD. Já na figura 27 (b) nota-se os dados da saída do PLL, com ênfase especial no tempo de acomodação, com resultados dentro do esperado.

5.2 RESULTADOS DO CHARGE PUMP E LOOP FILTER

O *Charge Pump* é um bloco que normalmente acompanha o PFD e tem a finalidade de converter o sinal digital e diferencial da saída do PFD em um sinal analógico utilizado para controlar o VCO. Contudo a saída do CP são pulsos de corrente, tornando necessário converter esta corrente de saída em uma tensão de controle que será utilizada pelo VCO, o responsável por esta conversão. A Figura 29 mostra a topologia inicialmente proposta para o CP.

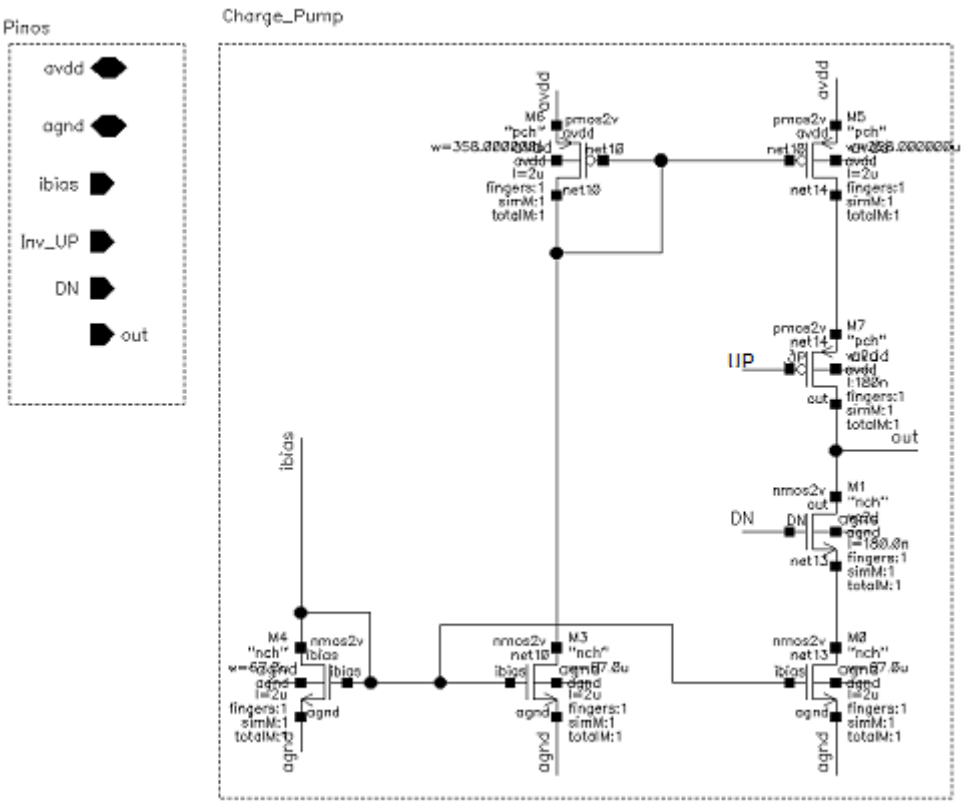

Figura 28 Topologia inicial CP.

Pode-se projetar o CP utilizando a topologia mostrada na Figura 28 onde os transistores m0, m3, m4, m5 e m6 atuam como espelhos de corrente e os transistores m1 e m7 funcionam com chaves, conectando as saídas UP e DN do PFD. Contudo o CP não funcionou como previsto com a utilização dos transistores. A saída do CP deve ser um degrau amplitude de 20 µA(corrente de alimentação do CP). Entretanto a corrente na saída não ultrapassava poucos na de amplitude, e a forma de onda não era a forma característica na saída de um CP. Foram feitas inúmeras alterações tanto no espelho de corrente como na relação $^{W}/_{L}$ dos transistores m1 e m7, de forma a corrigir os problemas de funcionamento do CP. Contudo não foram obtidas melhoras significativas na saída. Sabendo que os transistores m1 e m7 funcionam como chaves, optou-se por substituí-los por chaves complementares já utilizadas no PFD. A Figura 30 mostra a modificação feita na topologia.

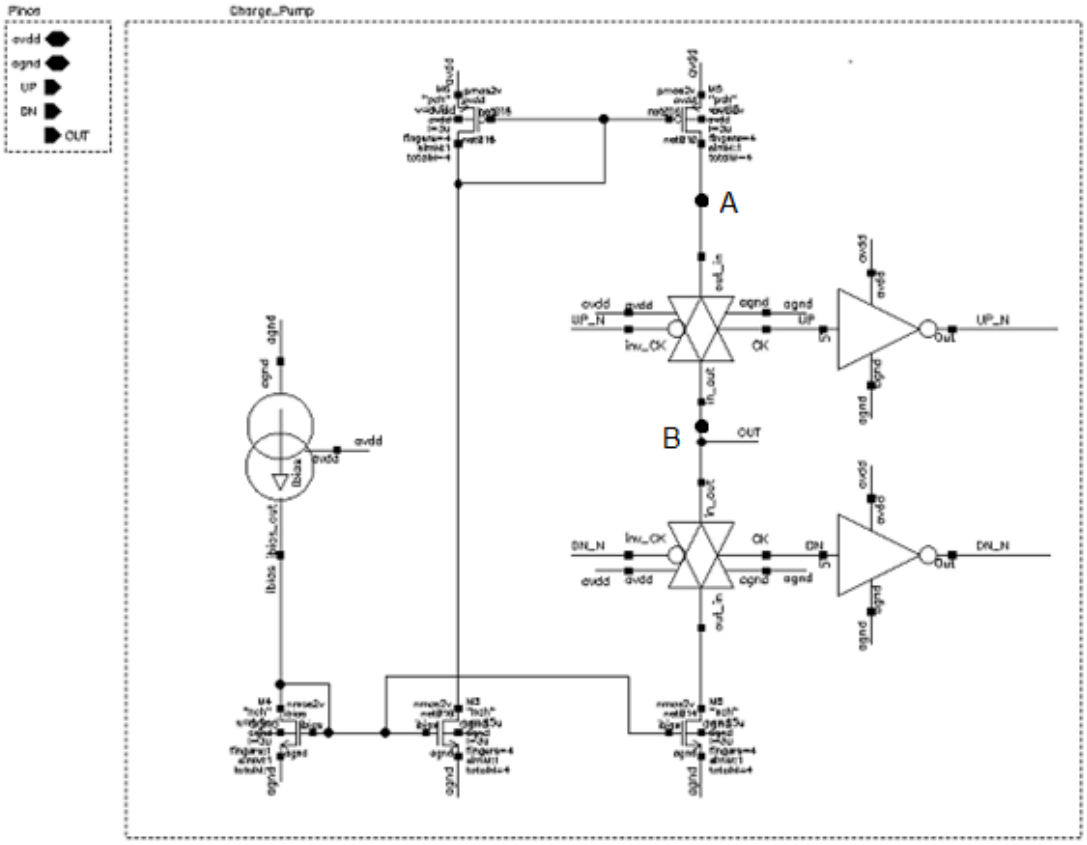

Figura 29 Topologia modificada do CP.

Com a modificação proposta o circuito funcionou como esperado. No projeto do CP é importante que os espelhos de corrente estejam bem projetados. No projeto dos espelhos de corrente foi definido o valor do comprimento do canal de 2µm. Logo, a única variável a ser alterada no espelho é a largura do canal do transistor. Outro aspecto é que o bloco deve operar com uma corrente de 20µA. A fonte de corrente utilizada possui uma corrente ibias de aproximadamente de 5µA. Desta forma, o espelho deve fornecer um ganho de corrente de 4 vezes Ibias:

$$Icópia = \left(\frac{\frac{W2}{L2}}{\frac{W1}{L1}} \right) Ibias \qquad (5.1)$$

Considerando também que o comprimento do canal é constante temos:

$$Icópia = \frac{W2}{W1} Ibias \quad (5.2)$$

Considerando que $W1 = 4\mu m$, logo o valor de $W2$ deve ser igual a $16\mu m$, possibilitando dessa forma a corrente de 20µA no CP.

A saída do CP gera um degrau de corrente, com o valor da corrente de alimentação. Deve-se então convertê-la em uma tensão. Isso é feito com o *Loop Filter.* A corrente na saída do CP é controlada pelas entradas UP, UP_N, DN e DN_N que saem do PFD. Quando as duas chaves complementares estão desativadas, ou seja, os sinais UP e DN estão em nível lógico baixo, o nó de controle fica em alta impedância, e a tensão de controle deve se manter constante. Contudo, existe uma pequena queda devido à corrente de fuga. Já quando as entradas UP e DN estiverem em nível lógico alto haverá o descasamento dos transistores, que funcionam como espelho de corrente, assim também havendo uma queda na tensão de controle. Quando houver uma transição de estados as cargas das chaves são redistribuídas, causando a variação da tensão de controle. Quando as duas chaves estão desligadas, a tensão do nó A vai para VDD e a tensão do nó B vai para zero. Quando a entrada UP está em nível alto, existe uma redistribuição de carga do nó A para o nó de controle, aumentando assim a tensão de controle na saída do *Loop Filter.* No entanto, é observada a existência de *glitches* na corrente que sai do CP.

Outro ponto que deve ser destacado é a forma de onda da corrente do CP, que deve ser um degrau de corrente com um valor aproximado de 20µA, visto este ser o valor da corrente de alimentação do CP. A corrente da saída estará próxima de zero quando a entrada UP estiver em nível lógico baixo e será máxima quando UP estiver em nível alto. Logo é possível notar uma correlação entre o acionamento de UP e o aumento da carga na saída do CP. A figura 30 mostra as formas de onda gerada na saída do *Loop Filter.*

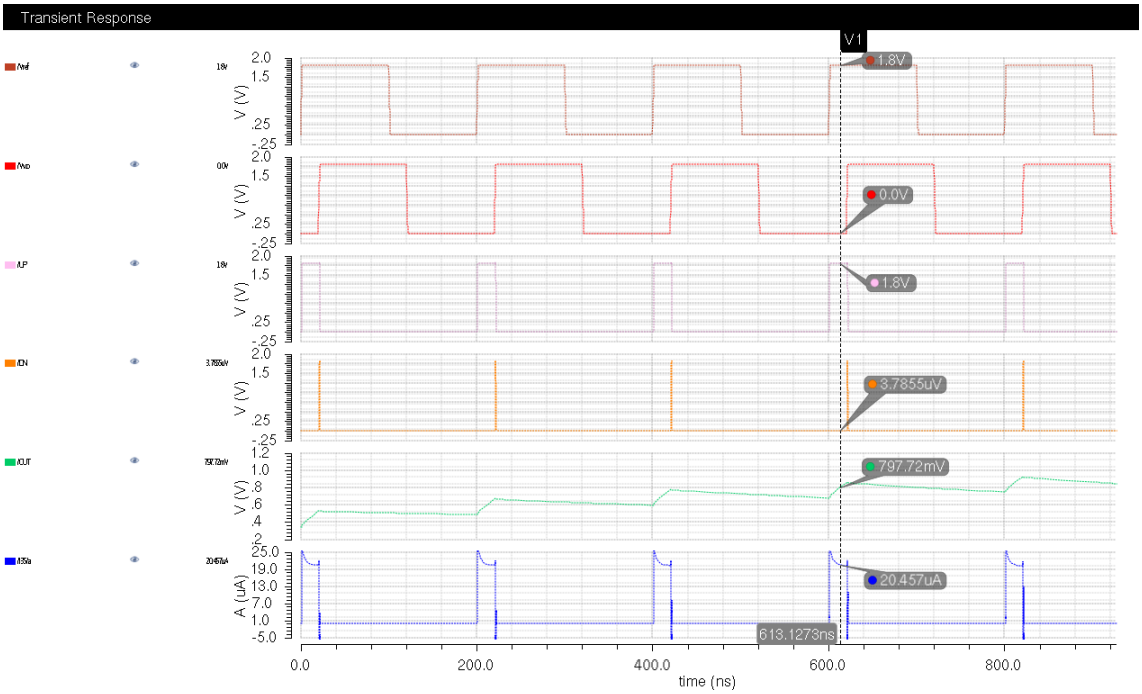

Figura 30 Simulação CP e LF.

Pode-se notar um *glitch* tanto na borda de subida da corrente, quanto na borda de descida. Eles podem ser minimizados reduzindo dimensões dos transistores, desde que mantida a proporção de $^{W}/_{L}$. A saída out do *Loop Filter* tem a forma de uma função dente de serra crescente. Isso acontece porque a tensão OUT é a intergral da corrente da saída do CP. A tensão se estabiliza com o tempo e o valor final depende da diferença de fase entre Vref e Vvco. Quanto maior for a diferença de fase, maior será a tensão de saída do LF, sendo possível controlar o VCO. A Figura 31 mostra a saída do CP e do LF até a estabilização da tensão.

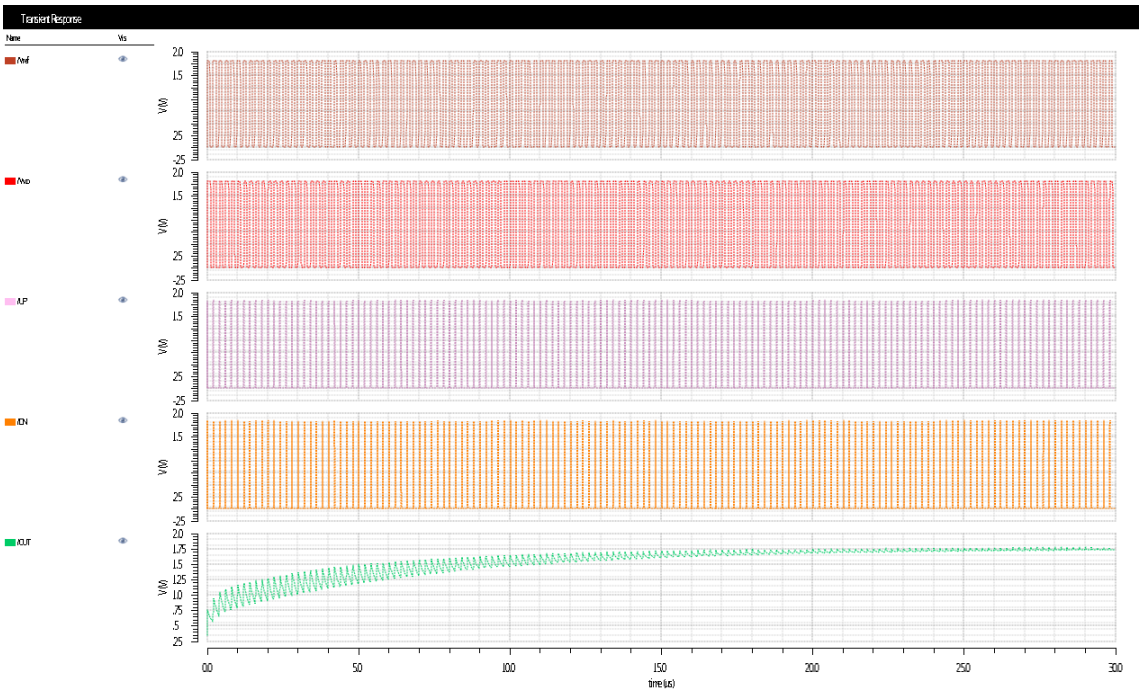

Figura 31 Estabilização da tensão do LF.

Como já mencionado anteriormente na seção 2.4, o LF converte uma corrente de entrada em uma tensão de saída, além de possuir a função de dar estabilidade ao PLL. Os parâmetros foram calculados utilizando as equações da seção 2.4. A tabela 2 mostra os valores obtidos para os componentes do LF.

Tabela 2 Componentes do Loop Filter

Componentes	Valores Obtidos
Resistor – R	265 KΩ
Capacitor – C1	16pF
Capacitor – C2	1pF

O filtro foi simulado juntamente com o CP, já que os dois devem ser analisados conjuntamente e as figura 30 e 31 mostradas anteriormente mostram o funcionamento do sistema. Os capacitores usados foram os “cfmon” da tecnologia TSMC 0.18µm e o resistor usado foi o “mhpolly” da mesma tecnologia. Tanto os capacitores quanto o resistor são compatíveis para o desenvolvimento do bloco. A utilização destes componentes só é possível

devido à frequência de operação ser baixa (5MHz). Em blocos com o VCO e o divisor devem ser utilizados componentes específicos para operar em altas frequências. A Figura 32 mostra o topologia do *Loop FIlter.*

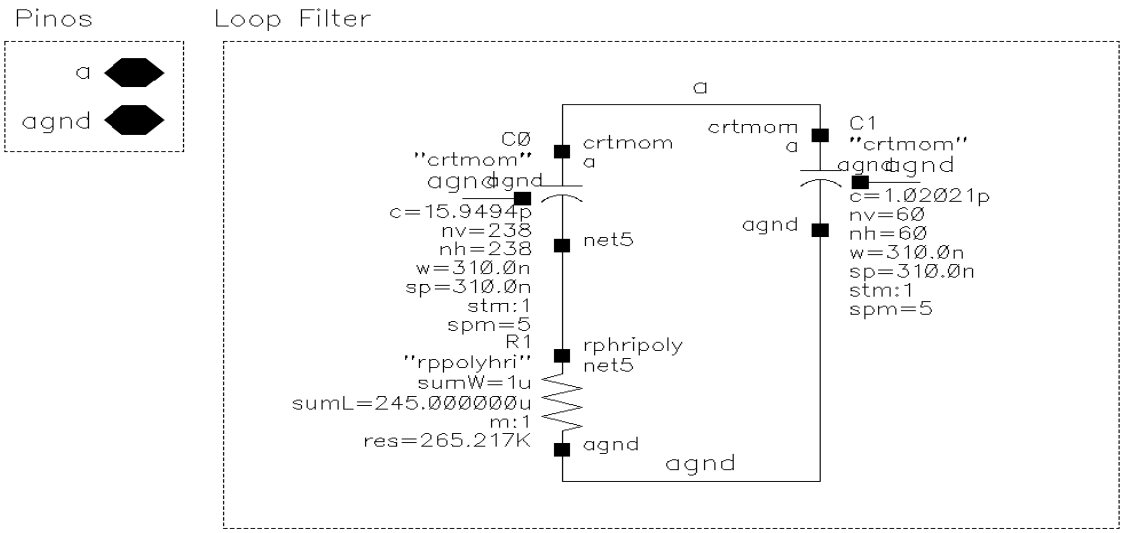

Figura 262 Topologia LF..

Para verificar o correto funcionamento do CP e do LF, deve-se fazer a simulação mista dos blocos, onde apenas o bloco a ser validado está em nível de componentes elétricos, e os demais estão representados em Verilog-AMS. A Figura 33 mostra a simulação dos dois blocos.

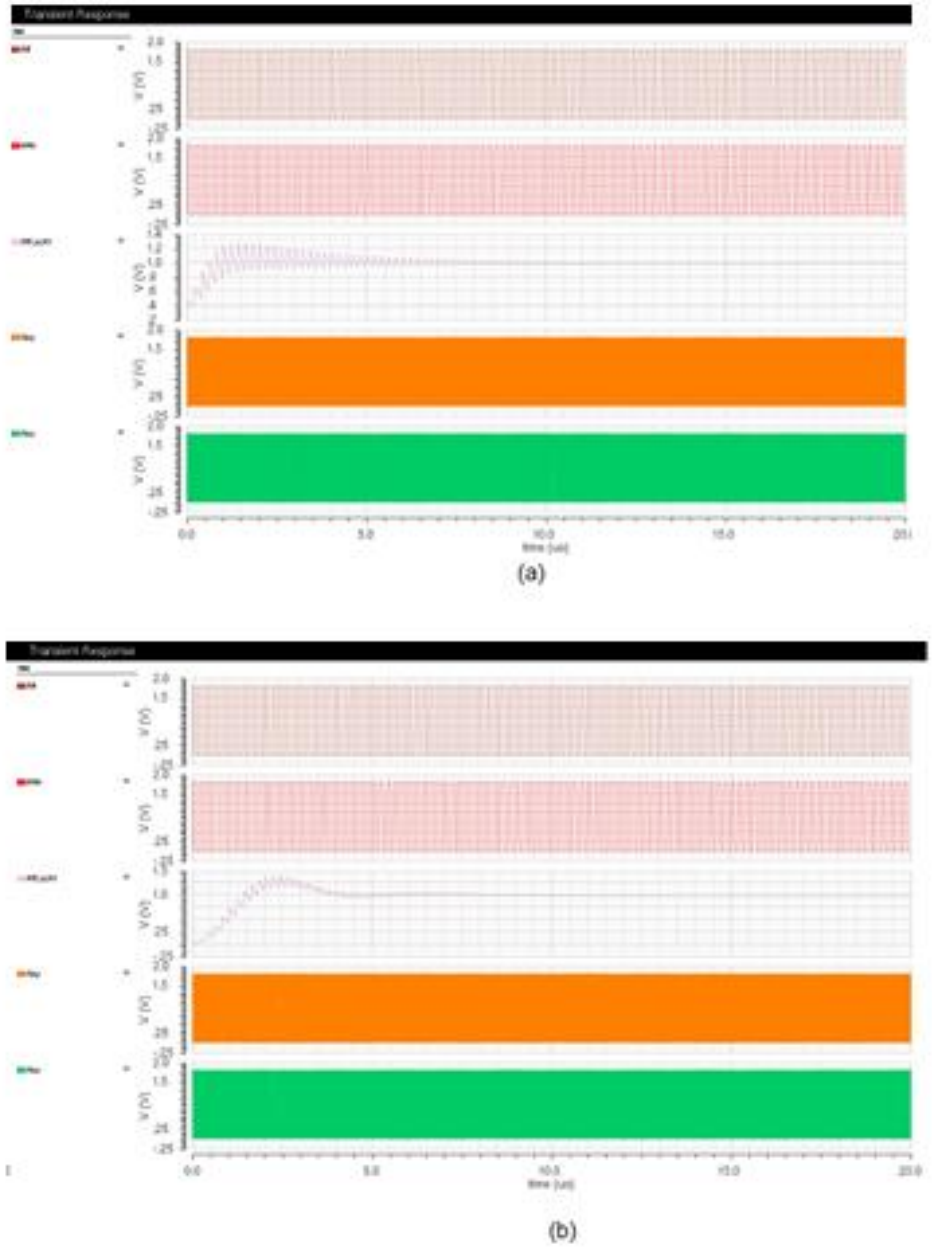

Figura 33 (a) Simulação mista CP (b) Simulação mista LF

Já a figura 34 mostra o tempo de acomodação dos dois blocos. Nota-se que nos dois blocos os tempos de acomodação foram de aproximadamente 10µs, o que está dentro dos parâmetros do projeto.

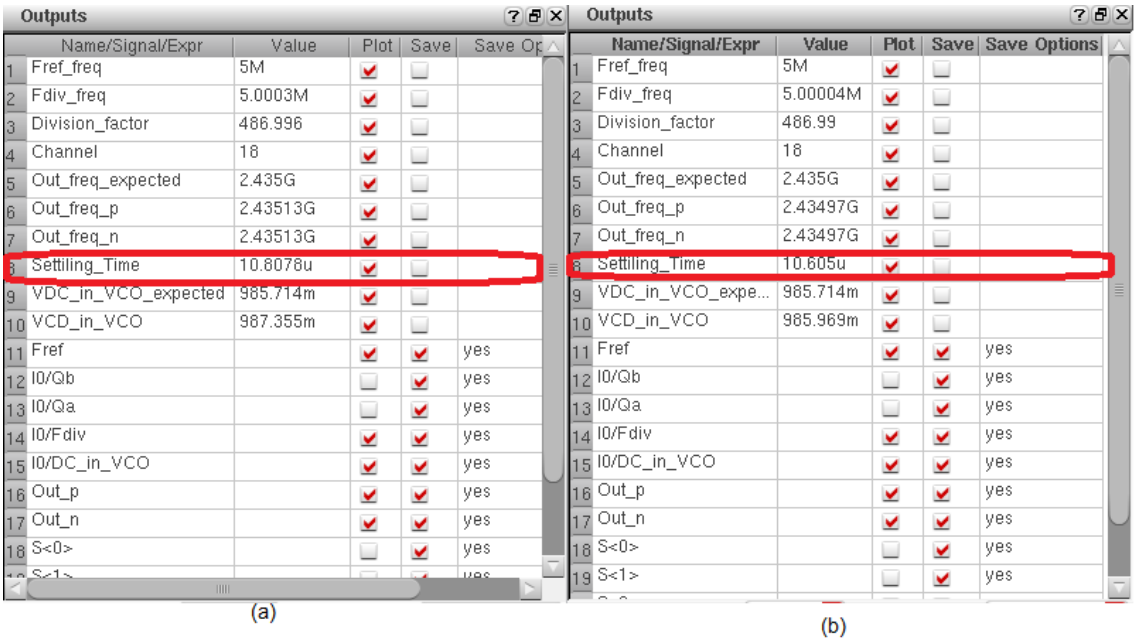

Figura 34 (a) Tempo de Acomodação LF (b) Tempo de Acomodação CP

Para finalizar a análise dos blocos, foi feita a simulação mista com os três blocos projetados. Essa simulação garante que os blocos estão cumprindo funcionando corretamente. Outra vez é necessário analisar se o tempo de acomodação do sistema está dentro dos parâmetros do projeto. Mais uma vez o tempo de acomodação ficou próximo de 10µs. Com essa análise foi possível comprovar a funcionalidade dos blocos. A Figura 35 mostra a simulação dos três blocos.

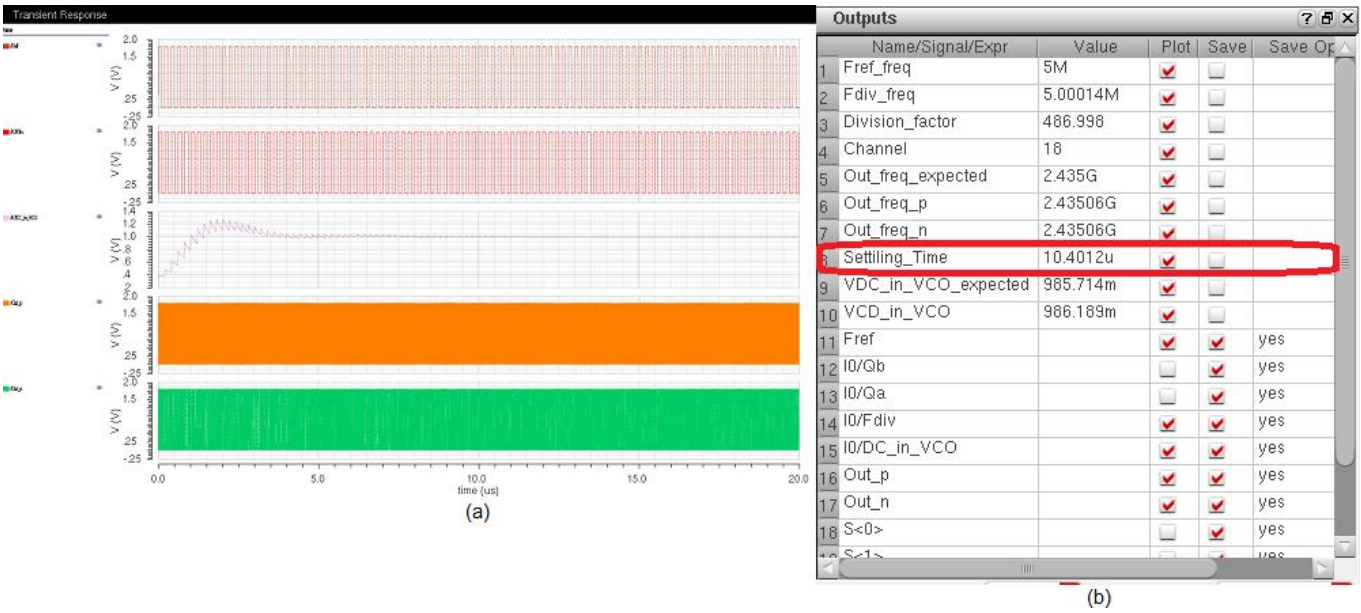

Figura 35 (a) Simulação blocos (b) Tempo de acomodação

Diante de todas as simulações feitas com cada bloco desde as transientes até a simulação de sinais mistos, pôde-se atestar a funcionalidade dos blocos. Todos os blocos obedeceram aos parâmetros de projeto definidos inicialmente. No decorrer do projeto foi necessário fazer algumas alterações em certas topologias para ter um ganho de desempenho do sistema. Foi o caso do CP, onde foi necessário mudar o sistema de chaveamento de transistores para chaves complementares, e no caso do PFD, onde foram testadas duas topologias para assim definir a mais indicada para a aplicação. O anexo A apresenta todas as simulações feitos no decorrer do trabalho, além das topologias usadas no projeto.

6. CONCLUSÃO

O desenvolvimento dos blocos do PLL abordados neste trabalho propiciou um aprofundamento nos conhecimentos de projetos de circuitos analógicos integrados, além de permitir a aprendizagem de uma metodologia eficiente de projeto e de conceitos de modelagem de circuitos analógicos utilizando Verilog-AMS. Diante disso o trabalho em um primeiro momento foi focado no estudo de temas relevantes para o desenvolvimento do projeto, e com isso ser possível sugerir propostas de topologia para os *blocos Charge pump*, *Loop Filter* e PFD do PLL, a ser utilizado em um transceptor *ZigBee*.

Na revisão bibliográfica, foram abordados conceitos sobre a construção dos blocos, e selecionadas as topologias mais adequadas e consolidadas na literatura, para que dessa forma o PLL funciona-se adequadamente. Assim, foram estudadas as características, funcionalidades e natureza física que regem os comportamentos dos blocos. Também foram feitos estudos a respeito da melhor metodologia de projeto, e realizada uma analise comparativa entre as duas mais utilizadas atualmente, sendo constatado que a metodologia *Top-Down* é a mais eficiente no desenvolvimento do projeto. Outra questão abordada foi referente à modelagem dos blocos utilizando a linguagem de descrição de *hardware* Verilog-AMS, visto que ela possibilita trabalhar com sistemas mistos e ser totalmente integrada às ferramentas de desenvolvimento CADENCE.

A simulação elétrica foi feita com software CADENCE, onde foram testados o PFD, CP e LF, através da realização de simulações transiente e DC dos blocos. Foi necessário modificar a topológica do CP, visto que as chaves com transistores não estavam funcionando adequadamente, sendo utilizadas desta forma chaves complementares CMOS. Também foram testadas duas topologias distintas de PFD, uma com o ffD TSPC e a outra com ffD biestável. Optou-se pela última, visto o seu melhor desempenho nos testes do circuito. Não se modificou a estrutura inicial do LF.

Não foi possível fazer o layout dos blocos neste trabalho, sendo recomendado para trabalhos futuros, pode-se também investigar uma topologia mais robusta de CP, assim diminuindo o ruído na saída do mesmo. Também não foi feita a integração de todos os blocos do PLL. Desta forma, será necessário no futuro integrar todos os blocos e fazer as simulações necessárias para validar o sistema tornando-se apto para uma futura prototipagem.

7. REFERÊNCIAS BIBLIOGRÁFICAS

Saleiro, Mario e Ey, Emanuel, “ZigBee uma abordagem prática 2010”;

Silva, A. T. D – “Módulos de Comunicação Wireless para Sensores”, 2007. Faculdade de Engenharia da Universidade do Porto.

Tose, Thobias – “Redes de Sensores sem Fio Aplicado a uma Estação de tratamento de esgoto”, 2010. Tese de Mestrado Universidade Federal do Espírito Santo.

Campos, Regina Silva – “Modelagem de um Transceptor ZigBee utilizando a Linguagem Verilog - AMS”, 2014. Trabalho de Conclusão de Curso Universidade de Brasília.

Cardoso, Adriano dos Santos – “Desenvolvimento e Impremetação de um Sintetizador de Frequência CMOS utilizando Sistema Digital”, 2009, Tese de Doutorado, Universidade Estadual Paulista.

Gomes, Pedro Henrique de Castro – “Análise e Síntese de um Algoritmo *Phase-Locked-Loop* Robusto para a Estimação de Amplitude, Fase e Frequência de Sinais Elétricos” Dissertação de Mestrado Universidade Federal de Juiz de Fora.

Neves, Leonardo Camargo – “Projeto de um Sintetizador de Frequência para Transceptor CMOS 920MHz Embarcado em um SoC”, 2010, Trabalho de Conclusão de Curso, Universidade de Brasília.

Giusti, Gustavo Buchweitz – “Projeto de um Circuito Divisor de Frequência de Ultra-Baixo Consumo de Potência”, 2007, Dissertação de mestrado, Universidade Federal de São Carlos.

Arguello, Angel María Gómez – “Estudo e Projeto de um sintetizador de Frequência para RF em Tecnologia CMOS de 0,35μm”, 2004. Dissentação de Mestrado Universidade de São Paulo.

Dabhi, Rajash A. e Nagpara, Bharat H. “*2 GHz PLL Frequency Synthesizer for ZigBee Applications*”. *International Journal of Innovative Reseach im=n Computer and Comunication Engineering.*

Rogers, John. Plett, Calvin, e Dai, Foster. “*Integrated Circuit Design for High-Speed Frequency Synthesis*”, 2006, livro. Editora: Copyrighted Material.

Kundert, k. S. – “The designer’s Guide To Verilog-AMS”, 2004. First edition, Consulting editor.

Kundert, K. S; Zinke, O. “The Designer’s Guide to Verilog-AMS”, 1. Ed. [S.1]: kluwer Academic Publishers, 2004.

Midorikama, E. T. “Uma Introdução às Linguagens de Descrição de Hardware”, Escola Politécinica da Universidade de São Paulo, 2001.

Nascimento, B. “Modelagem em Alto Nível da Seção de Recepção de um Transceptor RF”, 2010. Relatório de graduação. Universidade de Brasília.

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

Razavi, B. “RF Microeletronics.” 1 ed. [S.1.]: Prentice Hall PTR Upper Saddle River, NJ, 1998.

8. ANEXO

8.1 Topologia circuitos

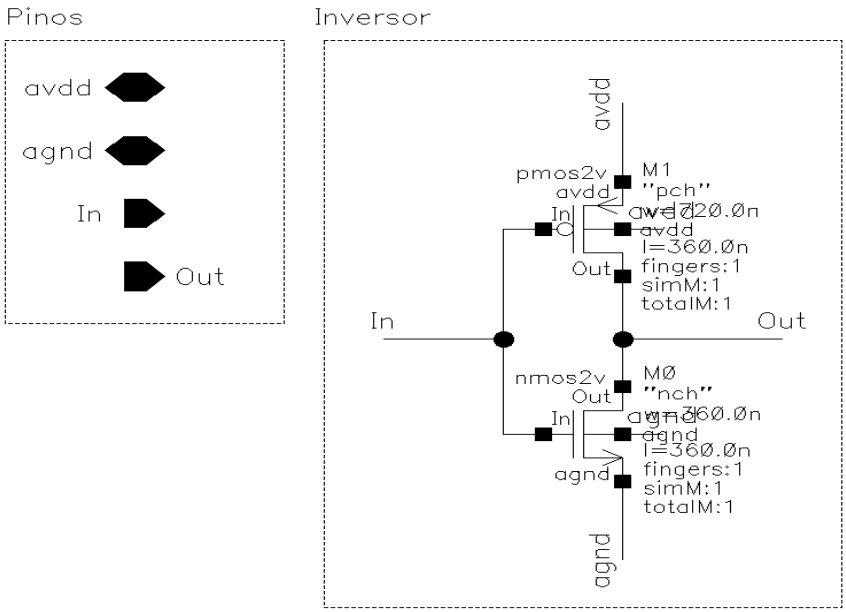

Figura 36 Topologia Inversor

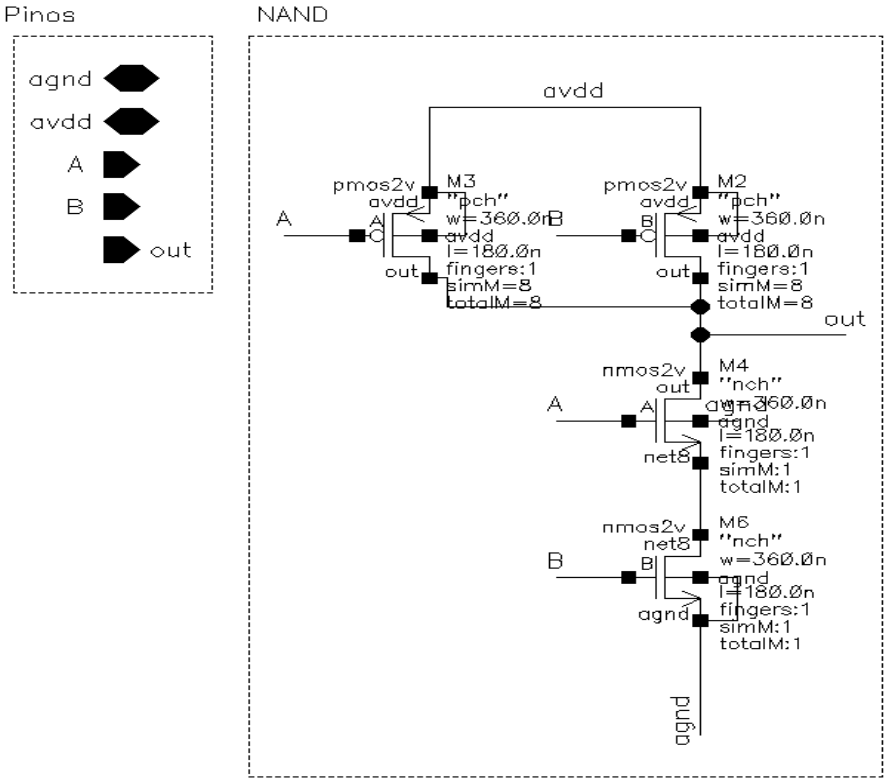

Figura 37 Topologia Nand

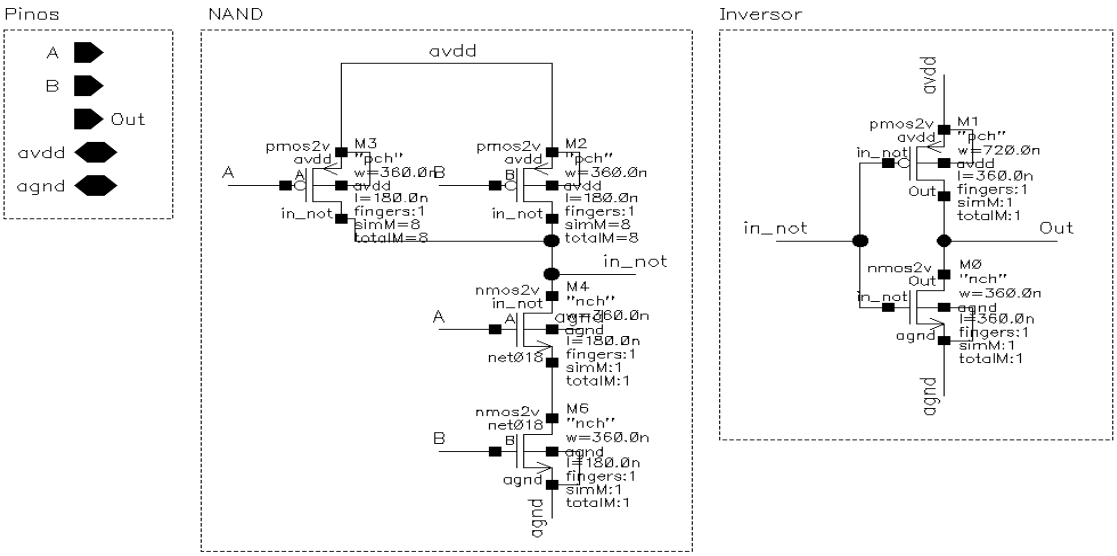

Figura 38 Topologia AND

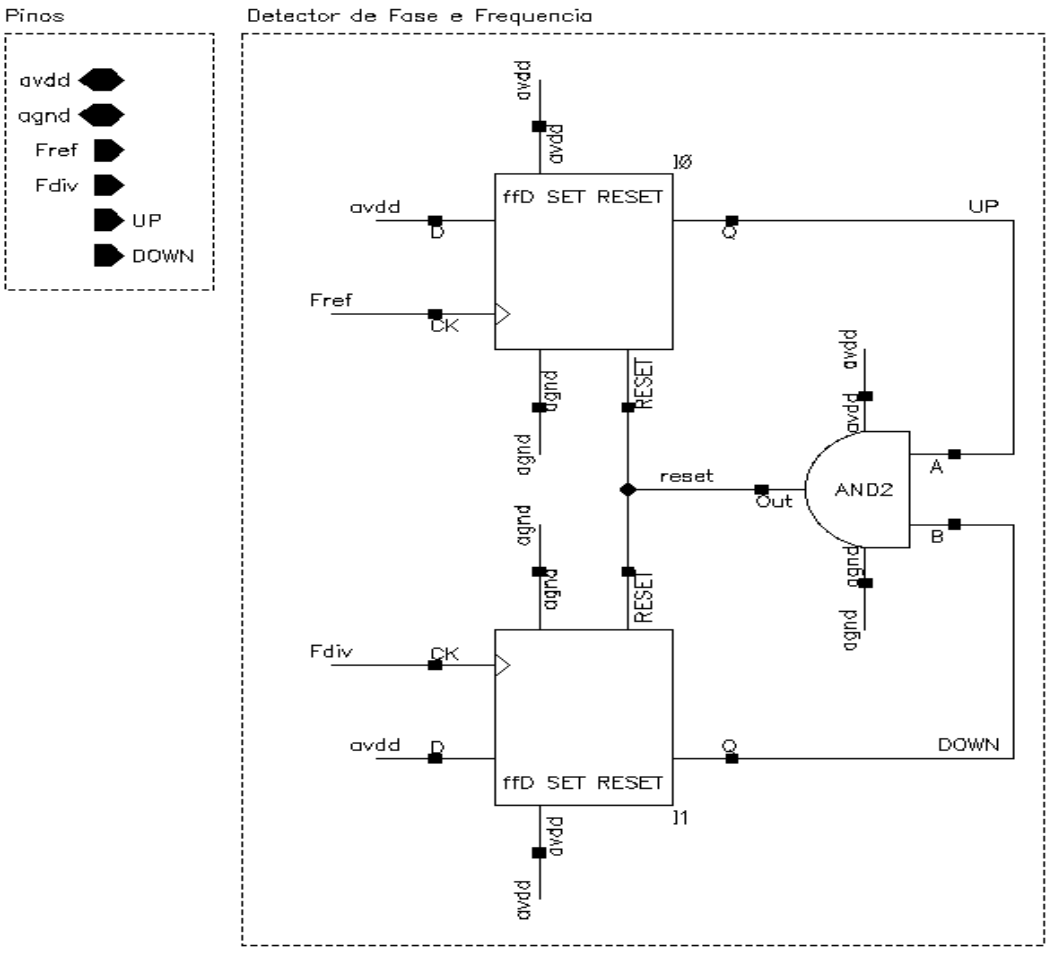

Figura 39 Topo do PFD

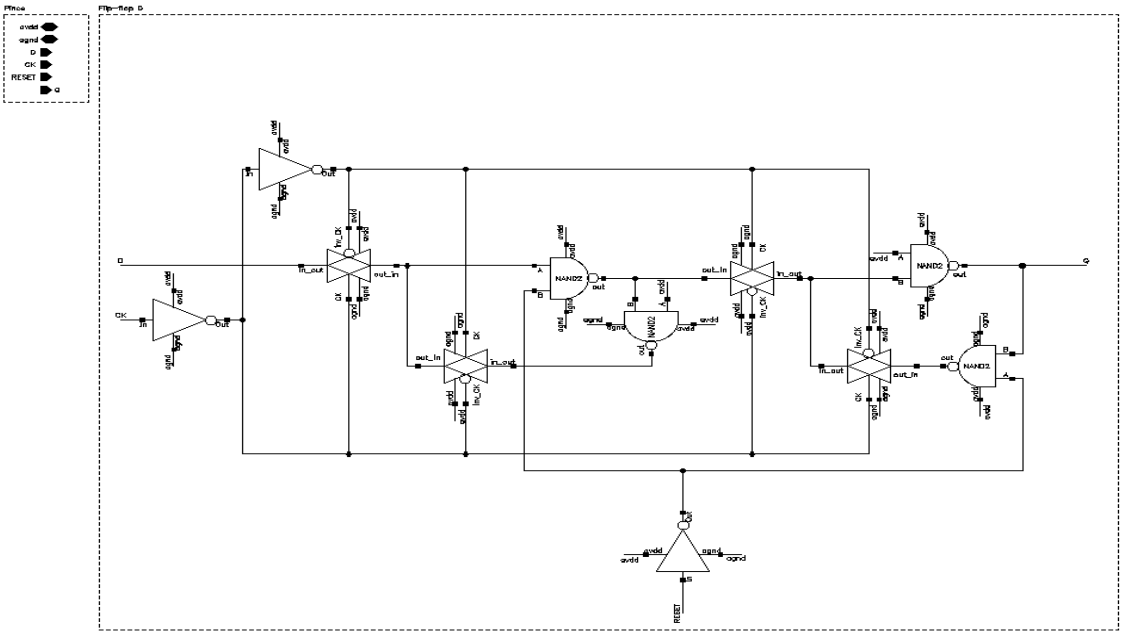

Figura 40 ffD biestável

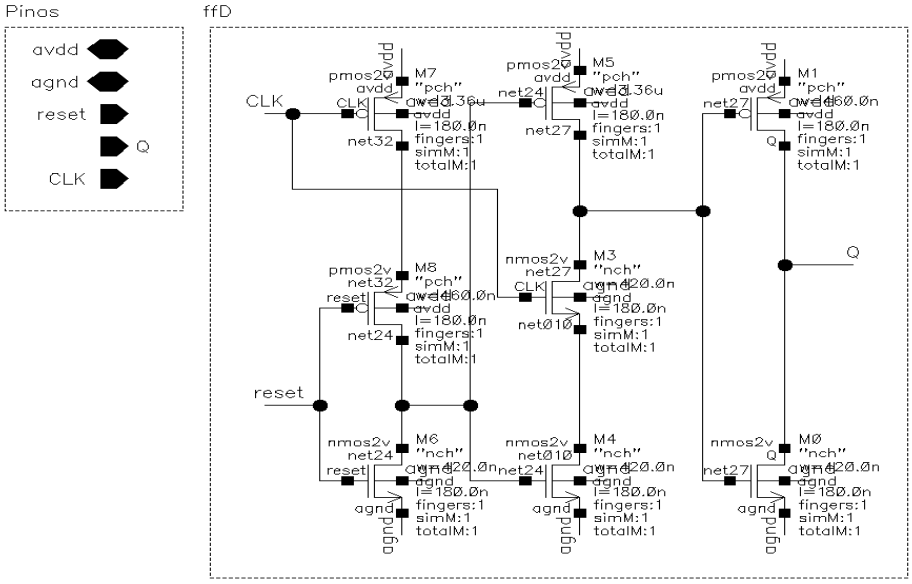

Figura 41 ffD TSPC

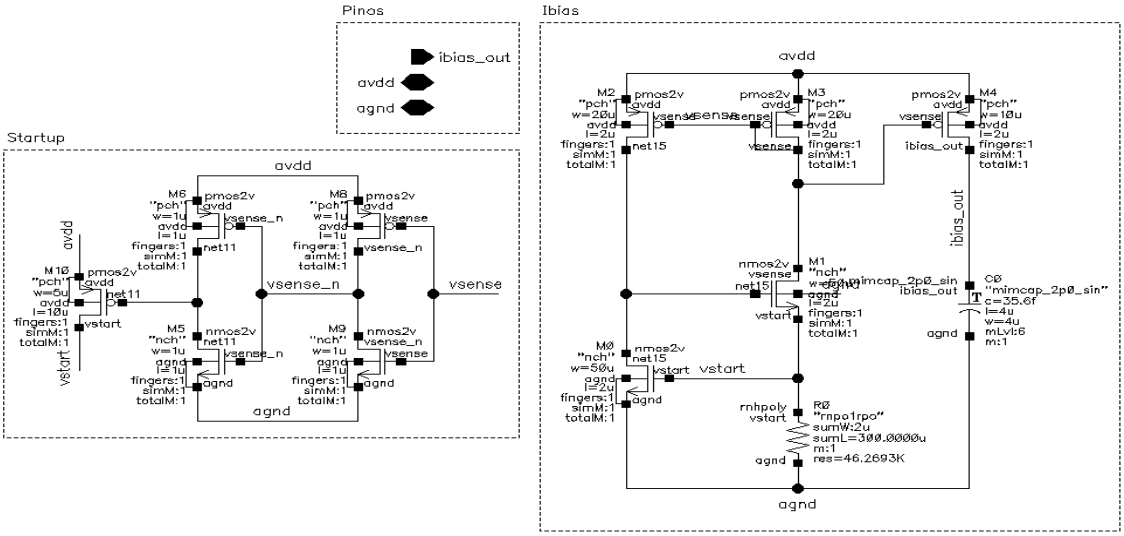

Figura 42 Fonte de Corrente

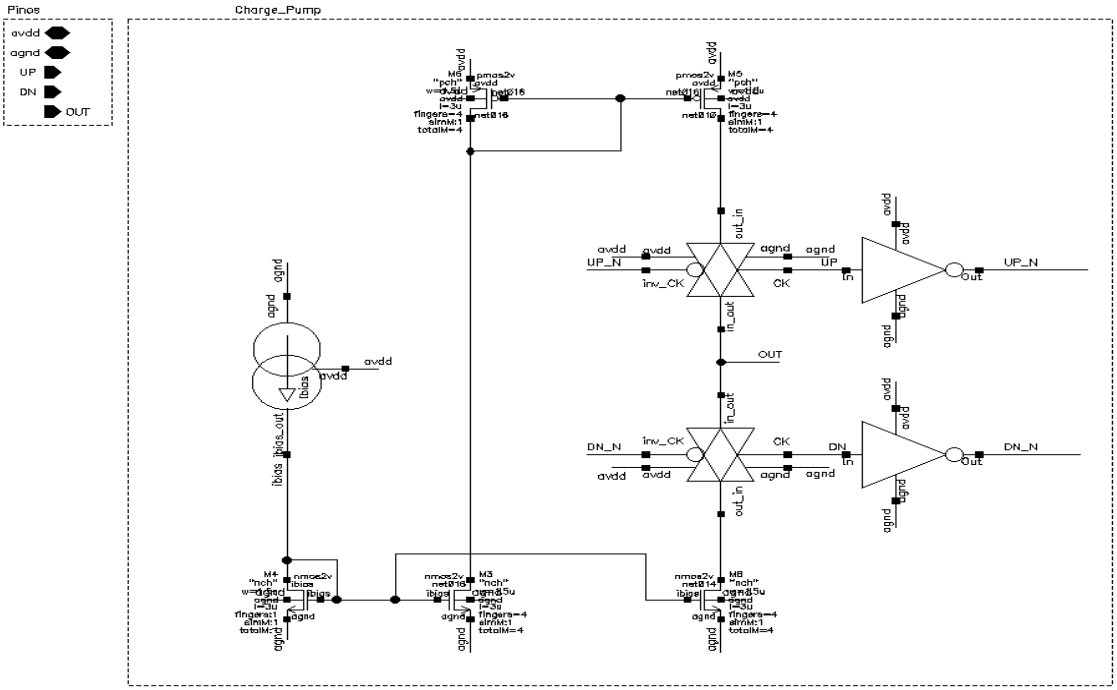

Figura 43 Charge Pump

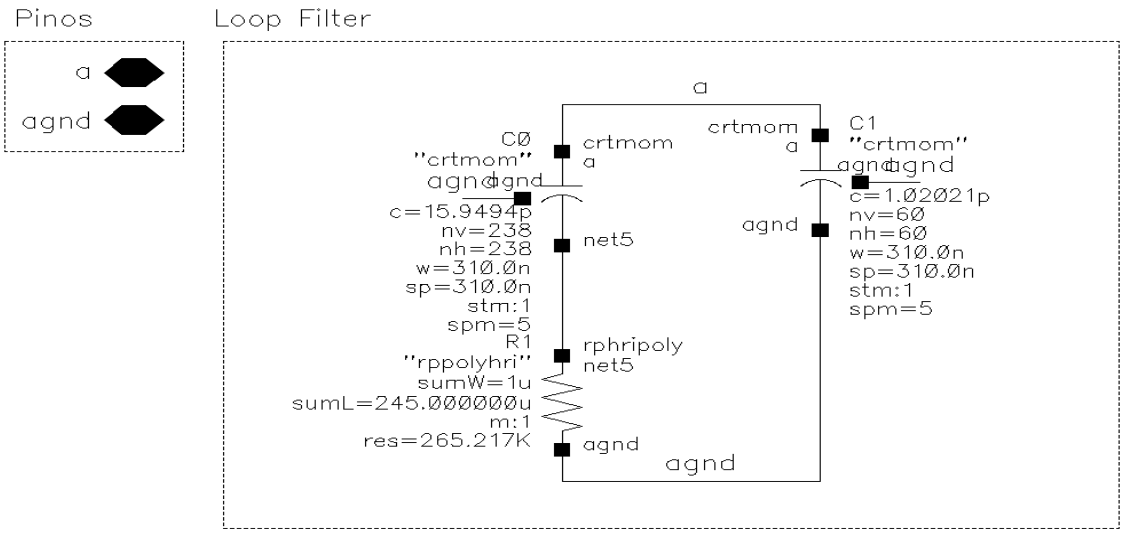

Figura 44 Loop Filter

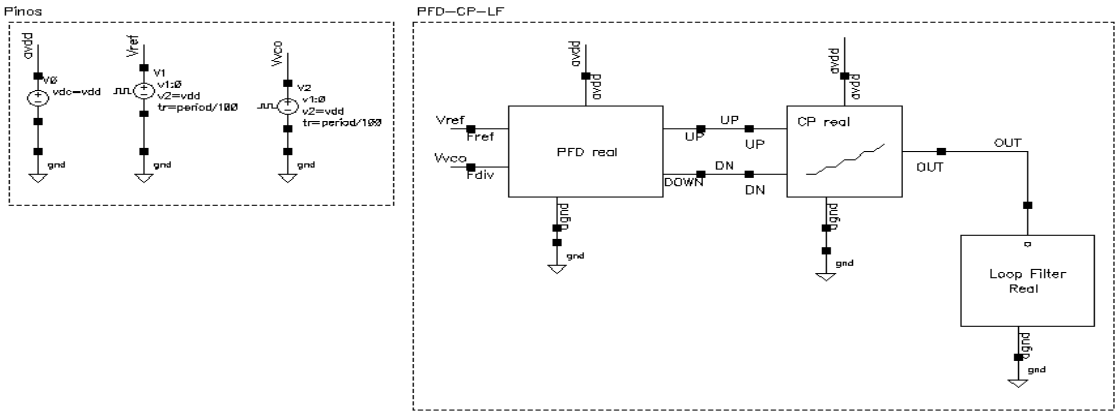

Figura 45 Blocos projetados

8.2 Simulações

Figura 46 Simulação Inversor

Figura 47 Simulação ffD TSPC

Figura 48 Simulação PFD com TSPC

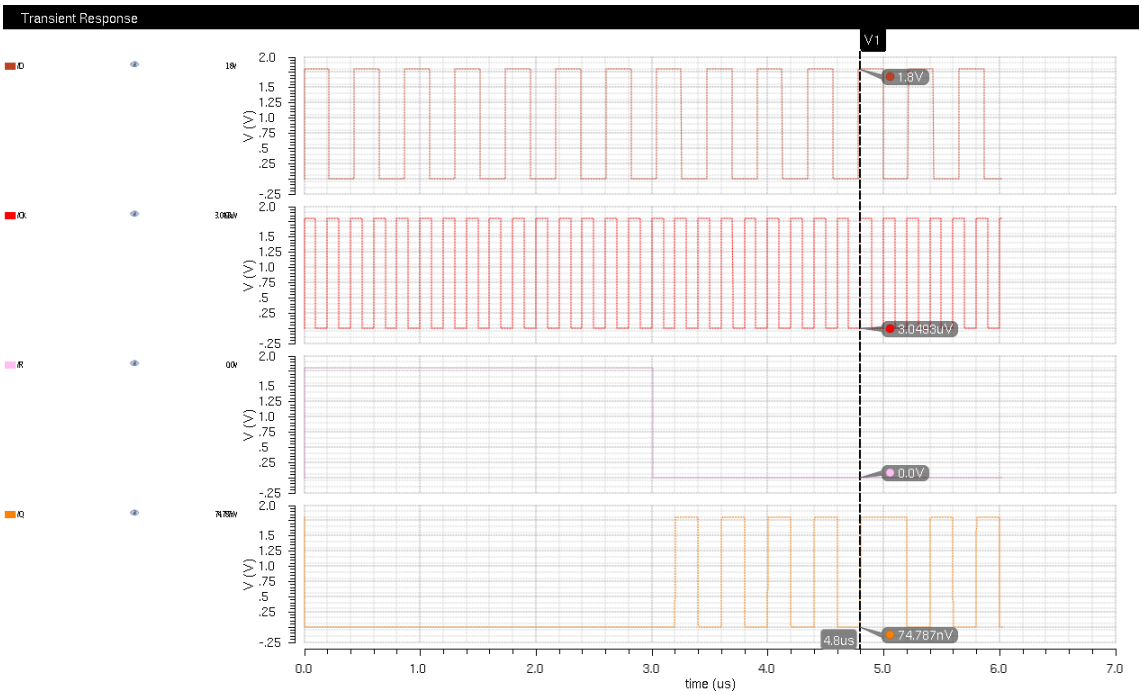

Figura 49 Simulação ffD biestável

Figura 50 Simulação PFD biestável

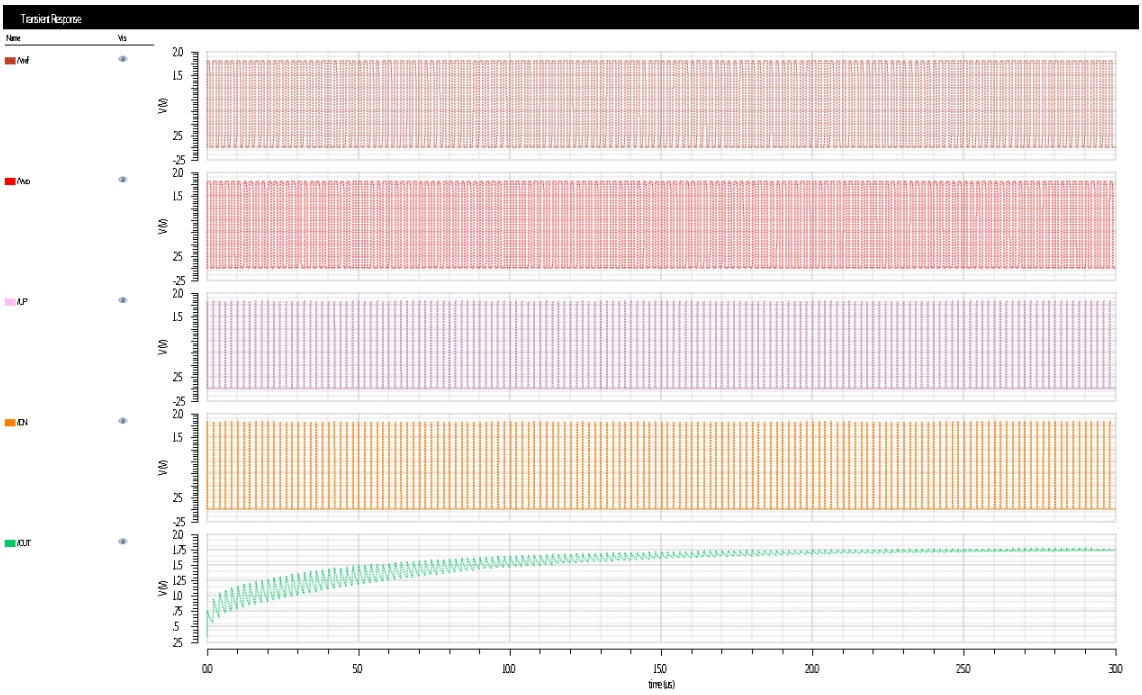

Figura 51 Simulação CP e LF

8.3 Esquemáticos das Simulações Mista

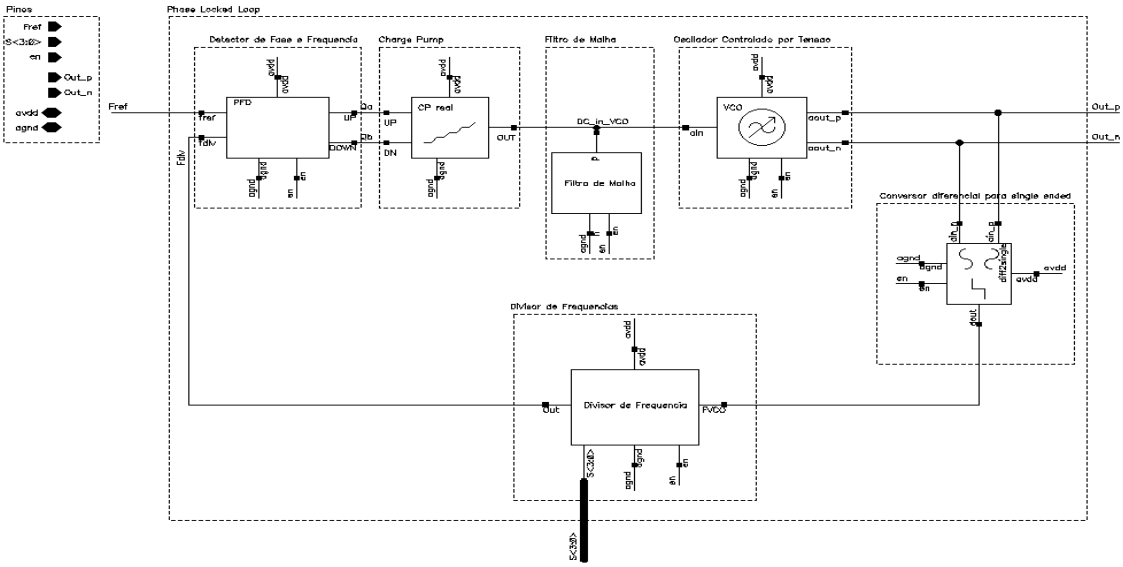

Figura 52 Esquemático CP real

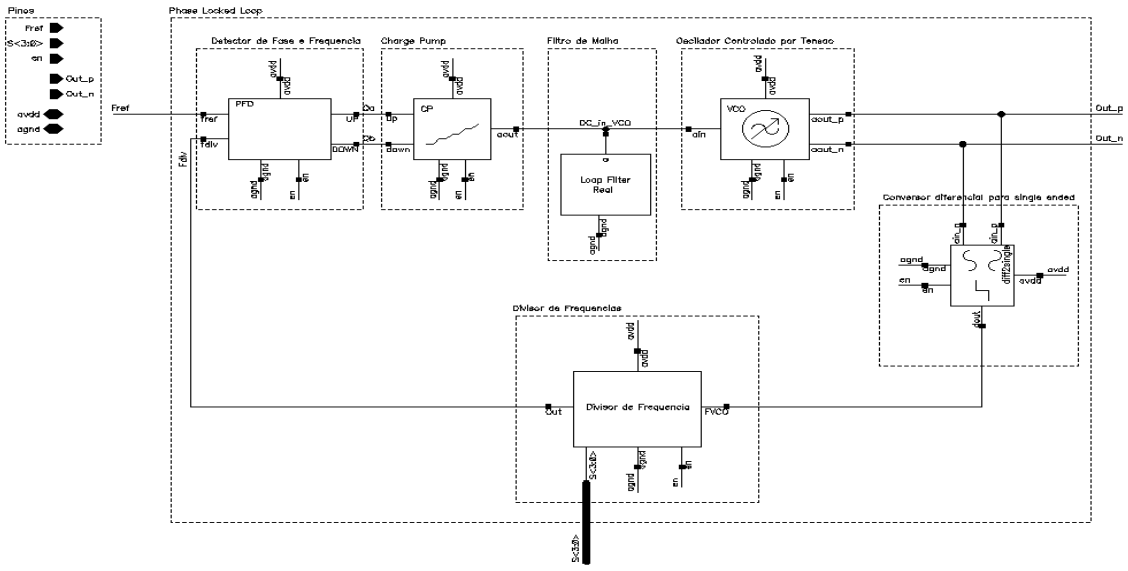

Figura 53 Esquematico LF real

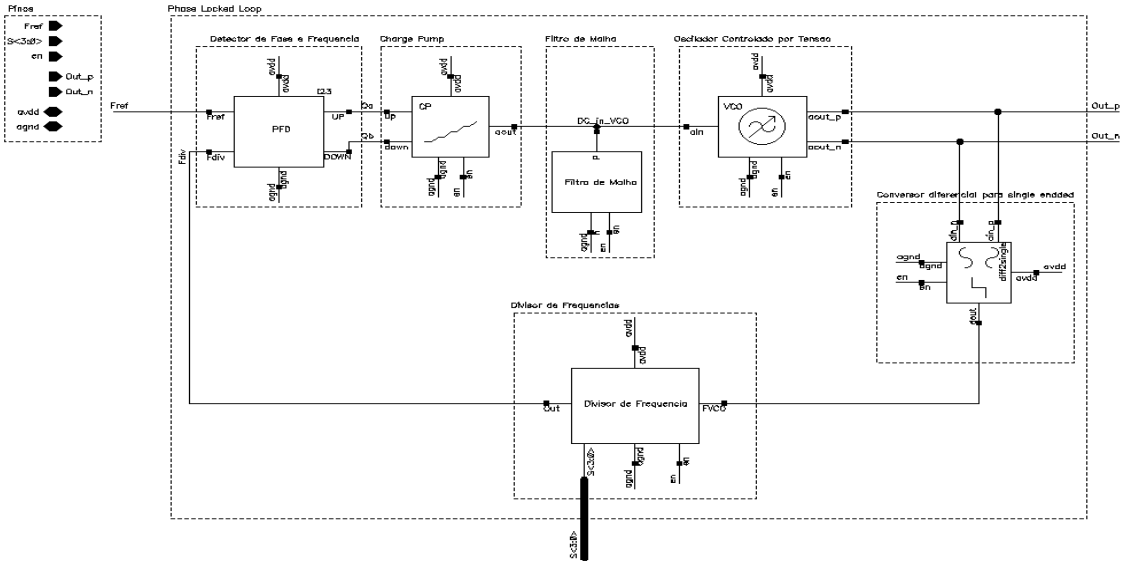

Figura 54 Esquemático PFD real

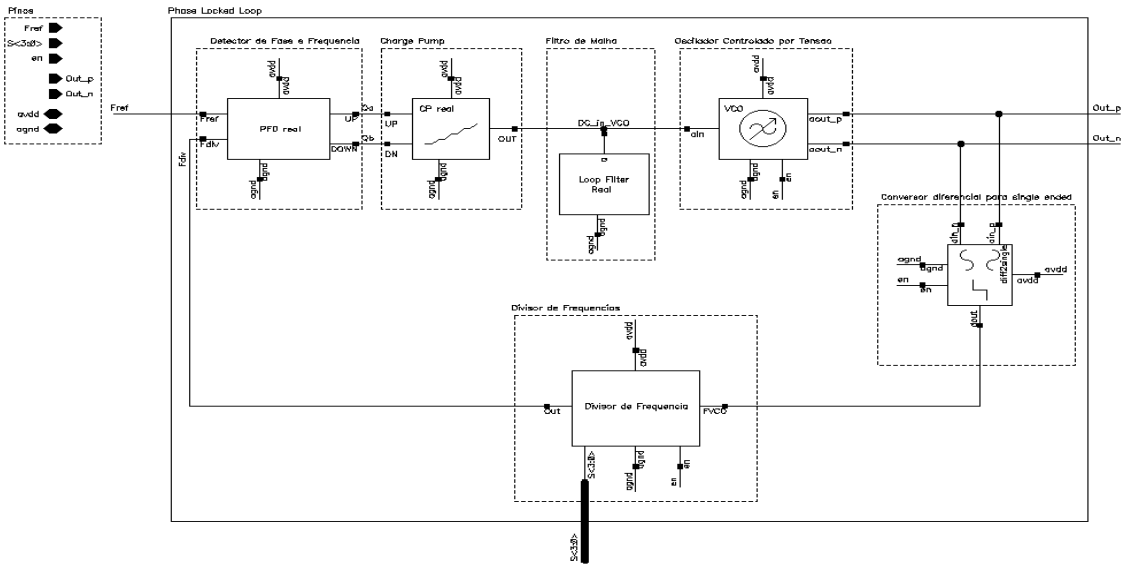

Figura 55 Esquemático Todos os blocos projetados