

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

Modelagem e Projeto de um Divisor de Frequências para Utilização no PLL de um Transceptor ZigBee

Autor: José Alisson de Albuquerque Pinto
Orientador: Prof. Dr. Wellington Avelino do Amaral

Brasília, DF
2015



José Alisson de Albuquerque Pinto

Modelagem e Projeto de um Divisor de Frequências para Utilização no PLL de um Transceptor ZigBee

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

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof. Dr. Wellington Avelino do Amaral

Brasília, DF

2015

José Alisson de Albuquerque Pinto

Modelagem e Projeto de um Divisor de Frequências para Utilização no PLL de um Transceptor ZigBee/ José Alisson de Albuquerque Pinto. – Brasília, DF, 2015-

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

Orientador: Prof. Dr. Wellington Avelino do Amaral

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2015.

1. *Phase Locked Loop*. 2. Divisor de Frequências. I. Prof. Dr. Wellington Avelino do Amaral. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Modelagem e Projeto de um Divisor de Frequências para Utilização no PLL de um Transceptor ZigBee

CDU 02:141:005.6

José Alisson de Albuquerque Pinto

Modelagem e Projeto de um Divisor de Frequências para Utilização no PLL de um Transceptor ZigBee

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, Julho de 2015:

**Prof. Dr. Wellington Avelino do
Amaral**
Orientador

**Prof. Dr. Sandro Augusto Pavlik
Haddad**
Convidado 1

**Prof. Dr. Cristiano Jacques Miosso
Rodrigues Mendes**
Convidado 2

Brasília, DF
2015

Este trabalho é dedicado a minha família, especialmente para Meires Marques de Albuquerque, José Pinto e Susana Blanco Marques de Albuquerque, meus pais e irmã.

Agradecimentos

Agradeço primeiramente a Deus. Em segundo, gostaria de agradecer imensamente a todos que me ajudaram a passar pelas dificuldades que surgiram durante esses anos de graduação.

Expresso minha gratidão ao meu orientador Prof. Dr. Wellington Avelino do Amaral pelo conhecimento que eu adquiri no último ano na área de Rádio Frequência (RF), o que foi decisivo para descobrir minha vocação profissional, pela orientação neste trabalho, bem como em outros projetos, pelas dicas e encorajamento para me guiar profissionalmente e academicamente. Agradeço também ao Prof. Dr. Sandro Augusto Pavlik Haddad, pela introdução na área de Projeto de Circuitos Integrados (PCI). A matéria PCI teve um papel fundamental na minha formação e na forma que vejo a Engenharia Eletrônica. Tenho grande gratidão ao Prof. Dr. Ricardo Ramos Fragelli pelos anos de trabalho no Grupo de Pesquisa de Sistemas Inteligentes e Adaptativos (GPSIA) e pelas oportunidades que fui agraciado durante o tempo em que trabalhamos juntos. O senhor se tornou, durante esses anos, uma referência como profissional e como pessoa. Agradeço também ao Dr. Heider Marconi e ao Rafael Santos Ferreira pela ajuda e conhecimentos adquiridos durante o estágio.

Agradeço profundamente aos meus amigos que conheço desde o Ensino Médio, irmãos que a vida me proporcionou conhecer e que são modelos para mim. Em especial a Luana Santos e Hanna Rezende, pela ajuda na revisão deste trabalho, Vesna Simões, Eduardo Pereira, David Costa, Gustavo de Almeida, Leidyane Regina e Marianna Gouveia pelos anos de companheirismo, incentivo e carinho. Aos amigos que conheci durante a graduação: o amigo de TCC Thiago Nunes, José Alberto, Ana Luiza Nogueira, Marlon Portugal, Pedro Penaforte, Ítalo Alberto, Yan Watanabe, André Matheus, Arthur Jahn Sturzbecher, Aulus Diniz, Nicholas Gustavo. Agradeço também ao meu cunhado André Blanco Nunes Silva.

Por fim, tenho gratidão eterna a minha mãe Meires Marques de Albuquerque, meu pai José Pinto, a minha irmã Susana e toda a minha família, pois graças a eles eu pude me tornar a pessoa que sou hoje.

*"Que você se lembre sempre de que os obstáculos no caminho não são obstáculos, eles
SÃO o caminho." Jane Lotter*

Resumo

Existem diversas aplicações de sistemas *wireless* e de sensoriamento na indústria que apresentam como requisitos baterias com baixo consumo e longa duração, bem como baixa taxa de transferência de dados e maior simplicidade se comparados com outros padrões disponíveis. Neste contexto, o padrão ZigBee/IEEE 802.15.4 foi desenvolvido para atender este tipo de mercado que necessita de sistemas confiáveis e de baixo consumo. Este trabalho apresenta o projeto em nível de transistor de um divisor de frequências Inteiro-N a ser utilizado num *Phase Locked Loop* (PLL). Este PLL está inserido num transceptor ZigBee que opera na faixa de frequência entre 2400 - 2475 MHz e possui 16 canais de operação espaçados por um fator de 5 MHz. Durante todo o trabalho será utilizada a metodologia *top-down* que apresenta fluxos de projeto bem definidos. A modelagem do divisor terá como principal ferramenta a linguagem de descrição de *hardware* Verilog-AMS que dá suporte a metodologia *top-down*. Primeiramente foi realizada a pesquisa bibliográfica referente ao PLL, divisor de frequências e levantamento dos parâmetros iniciais deste divisor. Após isso realizou-se a modelagem do divisor, utilizando o Verilog-AMS, focando na análise comportamental e levantamento das funcionalidades do mesmo, bem como o projeto a nível de transistor de todo o divisor. O divisor utiliza a topologia *Pulse Swallow Divider* que é composta por um *prescaler* com dois módulos de divisão ($N/N + 1$), um *swallow counter*(S) e o contador principal (P). O projeto dos blocos foi realizado utilizando as topologias *True Single Phase Clock* (TSPC) e *Extended True Single Phase Clock* (E-TSPC) na tecnologia TSMC (*Taiwan Semiconductor Manufacturing Company*) 0,18 μ m. O objetivo geral foi projetar o divisor de frequências para o PLL, desenvolvido com especificações próprias para o protocolo ZigBee. Por fim, este trabalho terá como aplicação a agricultura, mais especificamente o sistema de irrigação onde utiliza-se uma rede de sensores para detectar pontos de baixa umidade, enviar esta informação para a central de processamento por meio de um transceptor ZigBee de baixo consumo, que tem o PLL como um dos principais blocos em sua construção, e tomar as decisões para melhorar a produtividade na lavoura.

Palavras-chaves: *Phase Locked Loop (PLL)*. Divisor de Frequências. *Pulse Swallow Divider*

Abstract

There are many applications of wireless and sensing systems in the present industry that requires battery with low power consumption and long lasting as well as low data transfer rate and greater simplicity when compared with other standards available. The ZigBee/IEEE 802.15.4 standard is designed to address this type of market that needs reliable systems and low consumption. This paper presents the transistor level design of a frequency divider to be used in a Phase Locked Loop (PLL). This PLL is housed in a ZigBee transceiver that operates in the frequency range between 2400 – 2475 MHz and has 16 channels spaced by a factor of 5 MHz. The top-down methodology shall be used, because it provides well defined project streams. The divider will be modeled using the hardware description language VerilogAMS which supports top-down methodology. Initially, it was carried out literature concerning the PLL frequency divider and collection of initial parameters of this divider. After that, the divider modeling was held, using Verilog-AMS, focusing on behavioral analysis and survey of the same features, as well as the transistor level design of the divider. The divider has the topology Pulse Swallow Divider which is comprised of a dual modulus prescaler ($N / N + 1$), a swallow counter (S) and the main counter (P). The design of the blocks was done using topologies True Single Phase Clock (TSPC) and Extended True Single Phase Clock (E-TSPC) at technology TSMC (Taiwan Semiconductor Manufacturing Company) 0.18 μm . The overall goal was to design a frequency divider for the PLL, developed with particular specifications for the ZigBee protocol. Finally, this work has application as agriculture, more specifically in the irrigation system where a network of sensors is used to detect low humidity points, sending this information to the central processing by means of a low-power ZigBee transceiver, having PLL as one of the main blocks in its construction, and make decisions to improve productivity in agriculture.

Key-words: Phase Locked Loop (PLL). Frequency Divider. Pulse Swallow Divider

Lista de ilustrações

Figura 1 – Topologia básica de um transceptor ZigBee [Li et al. (2011)].	28
Figura 2 – Implementação dos sensores de aquisição de dados por Patil S. S.; Davande (2014).	31
Figura 3 – Fluxo de projeto de circuitos integrados analógicos [Adaptado de FitzPatrick e Miller (1998)].	33
Figura 4 – Diagrama de blocos da metodologia <i>bottom-up</i> [McCorquodale et al. (2003)].	35
Figura 5 – Diagrama de blocos da metodologia <i>top-down</i> [McCorquodale et al. (2003)].	37
Figura 6 – Estrutura da linguagem Verilog-AMS [FitzPatrick e Miller (1998)]. . .	40
Figura 7 – Modelagem funcional em Verilog-AMS de um <i>flash</i> ADC [Kundert K.; Chang (2007)].	41
Figura 8 – Topologias de rede de comunicação WSN [Lewis (2004)].	44
Figura 9 – Camadas definidas pelos protocolos ZigBee e IEEE 802.15.4.	46
Figura 10 – Diagrama de blocos da topologia básica de um PLL.	48
Figura 11 – Modelo de um PLL tipo I.	49
Figura 12 – <i>Loop filter</i> utilizado no PLL tipo II.	50
Figura 13 – Exemplo de um PFD utilizando <i>flip-flops</i> tipo D em conjunto com o <i>charge pump</i> representado por duas chaves com fontes de corrente [Razavi (2012)].	52
Figura 14 – Gráfico da corrente média (média dos pulsos QA e QB da figura 13) pela diferença de fase do PFD e <i>charge pump</i> [Rogers, Dai e Olett (2009)].	52
Figura 15 – Função de transferência típica de um LPF [Golio e Golio (2010)]. . . .	53
Figura 16 – Curva característica do VCO [Rogers, Dai e Olett (2009)].	54
Figura 17 – Diagrama de blocos do divisor de frequências [Razavi (2012)].	54
Figura 18 – Diagrama de blocos de um PLL com modulador <i>Sigma-Delta</i> ($\Sigma - \Delta$) [Perrott, Trott e Sodini (2002)].	56
Figura 19 – Exemplos de divisores: (a) assíncrono e (b) síncrono [Lacaita, Levantino e Samori (2007)].	58
Figura 20 – Proposta do <i>prescaler</i> com fator de divisão 15/16 [Razavi (2012)]. . . .	59
Figura 21 – Modelo proposto para o <i>swallow counter</i> [Adaptado de Santos (2008)].	60
Figura 22 – (a) <i>Lach</i> D completo em CML; (b) <i>lach</i> D CML no modo regenerativo [Razavi (2012)].	61
Figura 23 – Exemplo de um divisor por dois na topologia CML [Razavi (2012)]. . .	62
Figura 24 – Exemplo de um par diferencial básico [Razavi (2002)].	63
Figura 25 – Relação entre as entradas e saídas do par diferencial [Razavi (2002)]. .	63

Figura 26 – <i>Flip-flops</i> tipo D dinâmico: (a) TSPC, (b) E-TSPC [Lacaita, Levantino e Samori (2007)].	65
Figura 27 – Exemplo de um <i>prescaler</i> divisor por 3/4 [Chen e Jung (2011)].	65
Figura 28 – Exemplo de um conversor de entrada diferencial para <i>single-ended</i> (CML para TSPC) [Perrott (2003)].	67
Figura 29 – Estrutura geral do divisor.	68
Figura 30 – Esquemático do <i>flip-flop</i> na topologia TSPC.	70
Figura 31 – <i>Test bench</i> do <i>flip-flop</i> na topologia TSPC.	71
Figura 32 – Resultados de simulação do <i>flip-flop</i> D na topologia TSPC.	72
Figura 33 – Consumo de energia do <i>flip-flop</i> D na topologia TSPC.	72
Figura 34 – Esquemático do <i>flip-flop</i> D na topologia E-TSPC.	73
Figura 35 – Resultados de simulação do <i>flip-flop</i> D na topologia E-TSPC.	74
Figura 36 – Consumo de energia do <i>flip-flop</i> D na topologia E-TSPC.	74
Figura 37 – Esquemático do <i>prescaler</i> com fator de divisão 15/16.	75
Figura 38 – <i>Prescaler</i> com fator de divisão 3/4.	76
Figura 39 – Esquemático do divisor por 3.	76
Figura 40 – Simulação do <i>prescaler</i> com fator de divisão 15/16 quando MC=0.	77
Figura 41 – Simulação do <i>prescaler</i> com fator de divisão 15/16 quando MC=1.	78
Figura 42 – Simulação do <i>prescaler</i> com fator de divisão 15/16 variando o valor de MC.	78
Figura 43 – Cálculo da MFO do <i>prescaler</i>	79
Figura 44 – Esquemático do contador P.	81
Figura 45 – Resultado de simulação do contador P possuindo frequência de entrada igual a 1 GHz.	81
Figura 46 – Resultado de simulação da MFO do contador P.	82
Figura 47 – <i>Test bench</i> do <i>swallow counter</i>	82
Figura 48 – Simulação do 12º canal do <i>swallow counter</i>	83
Figura 49 – Simulação do 2º canal do <i>swallow counter</i>	83
Figura 50 – Gráfico do consumo do <i>swallow counter</i> em todos os canais.	84
Figura 51 – Esquemático do divisor completo (<i>Pulse Swallow Counter</i>).	85
Figura 52 – Simulação do 12º canal do divisor completo.	86
Figura 53 – Simulação do 2º canal do divisor completo.	86
Figura 54 – Gráfico do consumo dos 16 canais do divisor completo.	87
Figura 55 – Simulação do consumo do divisor completo utilizando o <i>flip-flop</i> E-TSPC no divisor P.	89
Figura 56 – Esquemático do conversor diferencial para <i>single-ended</i>	90
Figura 57 – Simulação do conversor diferencial para <i>single-ended</i>	91
Figura 58 – <i>Test bench</i> do divisor completo em conjunto do conversor diferencial para <i>single-ended</i>	91

Figura 59 – Simulação do divisor completo em conjunto do conversor diferencial para <i>single-ended</i>	92
Figura 60 – <i>Test bench</i> da comparação entre o divisor completo a nível de transistor e a modelagem em Verilog-AMS.	93
Figura 61 – Simulação do divisor completo a nível de transistor e a modelagem em Verilog-AMS em conjunto do conversor diferencial para <i>single-ended</i>	94
Figura 62 – Gráfico da diferença entre a simulação divisor completo a nível de transistor e da modelagem em Verilog-AMS em conjunto do conversor diferencial para <i>single-ended</i>	94
Figura 63 – Esquemático de uma porta AND de duas entradas.	105
Figura 64 – Esquemático do multiplexados de 2 bits utilizado na lógica de controle do divisor completo.	105
Figura 65 – Esquemático de uma porta NAND de três entradas.	106
Figura 66 – Esquemático de uma porta NOT.	106
Figura 68 – Esquemático de uma porta OR de três entradas.	106
Figura 67 – Esquemático de uma porta OR de duas entradas.	107
Figura 69 – Esquemático de uma porta XNOR de duas entradas.	107
Figura 70 – Esquemático da fonte de corrente utilizada no conversor diferencial para <i>single-ended</i>	111

Lista de tabelas

Tabela 1 – Bandas de frequência e taxas de dados do ZigBee [Ergen (2004)]. . . .	45
Tabela 2 – Tabela da verdade de um divisor por dois utilizando um <i>flip-flop</i> D. .	66
Tabela 3 – Especificações da frequência de operação e fatores de divisão dos divi- sores.	67
Tabela 4 – Pinos do DFF na topologia TSPC.	73
Tabela 5 – Pinos do DFF na topologia E-TSPC.	75
Tabela 6 – Tabela da verdade de um divisor por dois utilizando um <i>flip-flop</i> D. .	76
Tabela 7 – Comparação entre <i>prescalers</i> por meio da FOM.	80
Tabela 8 – Consumo do <i>prescaler</i> para diferentes frequências de operação.	80
Tabela 9 – Pinos do <i>prescaler</i> 15/16.	80
Tabela 10 – Pinos do contador P com módulo igual a 32.	82
Tabela 11 – Pinos de entrada e saída do <i>swallow counter</i>	84
Tabela 12 – Fatores de divisão com DFF TSPC em todos os divisores.	87
Tabela 13 – Fatores de divisão com DFF TSPC em todos os divisores.	87
Tabela 14 – Fatores de divisão utilizando DFF E-TSPC apenas no contador P. . .	88
Tabela 15 – Pinos do divisor completo.	89
Tabela 16 – Pinos do conversor diferencial para <i>single-ended</i>	92
Tabela 17 – Resumo dos resultados obtidos para os 3 blocos principais do divisor completo.	95

Lista de abreviaturas e siglas

AC	<i>Alternating Current</i>
AND	Porta lógica AND
API	Camada de aplicação do protocolo ZigBee
DC	<i>Direct Current</i>
DF	Divisor de Frequências
CK	<i>Clock</i>
CML	<i>Current-Mode Logic</i>
CP	<i>Charge Pump</i>
FOM	<i>Figure of Merit</i> (Figura de Mérito)
GSM	<i>Groupe Special Mobile</i>
HDL	<i>Hardware Description Language</i>
IC	<i>Integrated Circuits</i>
E-TSPC	<i>Extended True Single-Phase Clocking</i>
FAO	Organização das Nações Unidas para a Alimentação e Agricultura
IEEE	(Institute of Electrical and Electronics Engineers)
IP	<i>Intellectual Property</i>
LPF	Filtro passa baixas ou <i>Loop Filter</i>
MAC	Camada de acesso do padrão IEEE 802.15.4
MFO	Máxima Frequência de Operação
MS	<i>Mixed Signal</i>
ONU	Organização das Nações Unidas
OVI	<i>Open Verilog International</i>
OR	Porta lógica OR

PFD	Detector de Fase e Frequência
PHY	Camada física do padrão IEEE 802.15.4
PLL	<i>Phase Locked Loop</i>
QAM	<i>Quadrature Amplitude Modulation</i>
RFID	<i>Radio-Frequency Identification</i>
SWK	Camada <i>Network</i> do protocolo ZigBee
TSPC	<i>True Single-Phase Clocking</i>
VCO	Oscilador controlado por tensão ou <i>Voltage Controlled Oscillator</i>
VHDL	<i>VHSIC Hardware Description Language</i>
VHSIC	<i>Very High Speed Integrated Circuit</i>
VLSI	<i>Very-large-scale integration</i>
WSN	<i>Wireless Sensor Network</i>
XNOR	Porta lógica XNOR
XOR	Porta lógica XOR

Lista de símbolos

$\sigma_{t_0}^2$	<i>Jitter</i>
θ_R	Fase do oscilador a cristal
θ_D	Fase da saída do divisor de frequências
θ_0	Fase da saída do PLL
FD	Cálculo do Fator de Divisão

Sumário

1	INTRODUÇÃO	27
1.1	Objetivos	28
1.1.1	Objetivos Gerais	28
1.1.2	Objetivos Específicos	28
1.2	Motivação: Aplicações em Agricultura	29
1.3	Organização do Trabalho	31
2	METODOLOGIA DE PROJETO	33
2.1	<i>Bottom-Up</i>	34
2.2	<i>Top-Down</i>	36
2.3	Escolha da Metodologia	37
3	LINGUAGEM DE DESCRIÇÃO DE HARDWARE	39
3.1	Verilog-AMS	39
3.1.1	Estrutura da Linguagem Verilog-AMS	41
4	REDES DE SENSORES SEM FIO	43
4.1	Introdução	43
4.2	Tipos de Rede WSN	44
4.3	Características da Rede ZigBee	45
5	FUNDAMENTOS DO PHASE-LOCKED LOOP (PLL)	47
5.1	<i>Phase-Locked Loop (PLL)</i>	47
5.2	Tipos de PLL	49
5.2.1	Tipo I	49
5.2.2	Tipo II	50
5.3	Blocos Básicos	51
5.3.1	Detector de Fase e Frequências (PFD)	51
5.3.2	<i>Charge Pump (CP)</i>	52
5.3.3	<i>Loop Filter (LPF)</i>	53
5.3.4	Oscilador Controlado por Tensão (VCO)	53
5.3.5	Divisor de Frequências	54
5.4	Tipos de Divisores para PLL	55
5.4.1	Inteiro-N	55
5.4.2	Fracionário-N	55
5.4.3	Escolha do tipo de divisor	56

6	DIVISOR DE FREQUÊNCIAS	57
6.1	Estrutura Geral	57
6.2	<i>Prescaler</i> com Dois Módulos de Divisão	58
6.3	<i>Swallow Counter</i> (S)	59
6.4	Contador Principal (P)	60
6.5	Projeto de Circuitos Lógicos para Altas Frequências	60
6.5.1	<i>Current-Mode Logic</i> (CML)	61
6.5.2	Considerações de Projeto do Amplificador Diferencial	62
6.5.3	<i>True Single-Phase Clocking</i> (TSPC)	64
6.5.4	Considerações de Projeto do TSPC	65
6.5.5	Escolha da Topologia dos Circuitos Lógicos	66
6.6	Análise do Ruído de Fase e <i>Jitter</i> no Divisor de Frequências	68
7	RESULTADOS	69
7.1	Circuitos Utilizados na Lógica de Controle	69
7.1.1	Portas Lógicas e Multiplexador de 2 bits	69
7.1.2	<i>Flip-flops</i> TSPC e E-TSPC	69
7.1.2.1	Topologia TSPC	70
7.1.2.2	Topologia E-TSPC	73
7.2	<i>Prescaler</i> (N/N+1) Divisor 15/16	75
7.3	Contador Principal (P) Divisor 32	80
7.4	<i>Swallow Counter</i> (S) Divisor 16	82
7.5	Divisor Completo e Lógica de Controle	85
7.6	Conversor de Entrada Diferencial para <i>Single-Ended</i>	89
7.7	Simulações Mistas	92
7.8	Resumo dos resultados	95
8	CONCLUSÃO	97
	Referências	99
	APÊNDICES	103
	APÊNDICE A – PORTAS LÓGICAS CMOS	105
	ANEXOS	109
	ANEXO A – FONTE DE CORRENTE	111

1 Introdução

Com o passar dos anos a sociedade vem demandando novos meios de comunicação e novas maneiras de se analisar um ambiente para coletar dados sobre o mesmo, objetivando por tecnologias precisas, confiáveis, de fácil acesso e que apresentem soluções factíveis para as necessidades do usuário. Neste contexto, surgiram sistemas de comunicação *wireless* que oferecem um método seguro de troca de informações sem fio. Dentre as aplicações se destacam redes de celulares, sensores *wireless*, *notebooks* e a própria *Internet* como um todo.

Uma das grandes aplicações de sistemas sem fio está no uso de *Wireless Sensor Network* (WSN) no monitoramento de sistemas de irrigação e *smart grid*, que tem como objetivo espalhar sensores em um dado ambiente para colher informações, analisá-las e a partir disso tomar decisões que possam aumentar a produtividade desse ambiente. Um exemplo disso é espalhar sensores numa plantação, armazenar informações acerca da umidade do solo, temperatura, previsão do tempo e com isso preveem onde é necessário adubar ou irrigar, por exemplo. As redes WSN que primam por baixo consumo, curto alcance e baixa taxa de transferências baseiam-se no protocolo ZigBee de 2,4 GHz, composto por 16 canais e padronizado pela IEEE 802.15.4 e pela *ZigBee Alliance*.

Este trabalho tem como objetivo modelar e projetar um divisor de frequências para o PLL de um transceptor ZigBee. Um dos dispositivos mais utilizados no gerenciamento dos sensores e na troca de informações é o transceptor de rádio frequência (RF), que possui a capacidade de enviar e receber dados, composto por um sistema de modulação e recepção. Estes dois sistemas necessitam de um sinal senoidal, denominado de portadora, para que a transmissão e recepção seja feita de forma correta. O *Phase-Locked Loop* (PLL) é uma das técnicas utilizadas para sintetizar frequências e proporcionar o sinal da portadora. A problemática que gira em torno do PLL é que embora ele seja bastante confiável, o seu circuito é difícil de ser projetado e apresenta alto consumo de energia, pois opera em altas frequências. Os blocos internos do PLL que proporcionam este consumo energético é o Oscilador Controlado por Tensão (VCO) e o divisor de frequências. Dessa forma, ao se projetar ambos os blocos visando baixo consumo, mas mantendo a integridade do funcionamento do PLL, é possível reduzir o custo energético do transceptor, o que acarretaria no uso de baterias pequenas e que duram meses, evitando a troca constante das baterias utilizadas nos sensores da rede WSN.

1.1 Objetivos

1.1.1 Objetivos Gerais

Diversas aplicações de comunicação sem fio utilizam o padrão IEEE 802.15.4/ZigBee por oferecer um produto barato que prioriza o baixo consumo de energia, com baixa taxa de transferência de dados e de curto alcance sendo utilizada no sensoriamento de sistemas WSN (*Wireless Sensor Network*). O transceptor (figura 1) é um dispositivo extremamente importante no sistema de comunicação do ZigBee, pois é responsável por enviar e receber a informação entre os nós e a central, sendo assim, composto por um sistema de transmissão e recepção (figura 1). O *Phase Locked Loop* (PLL) é um multiplicador de frequências responsável por fornecer um sinal de referência que é utilizado na modulação e demodulação do sinal pelo transceptor.

Este trabalho apresenta o projeto de um divisor de frequências para o PLL de um transceptor ZigBee. Este transceptor opera na frequência de 2,4 GHz e tem grande aplicação em redes WSN em geral, sendo o sistema de irrigação uma das possíveis aplicação deste dispositivo.

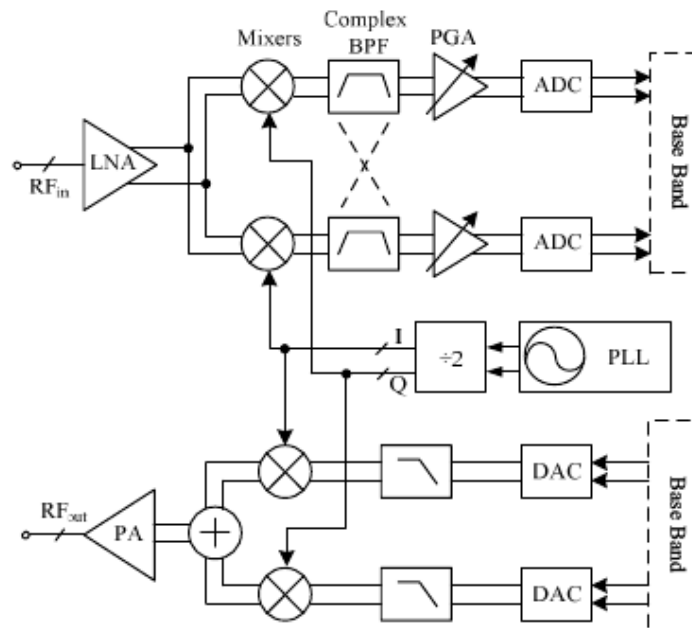


Figura 1 – Topologia básica de um transceptor ZigBee [Li et al. (2011)].

1.1.2 Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Estudar o funcionamento do PLL e, principalmente, do seu divisor;
- Estudar técnicas de construção de divisores de frequência para um PLL;

- Utilizar a linguagem de descrição de *hardware* Verilog-AMS na ferramenta Cadence para dar suporte a metodologia utilizada neste trabalho;
- Gerar as especificações de desempenho dos divisores para que seja possível criar modelos em Verilog-AMS que apresentem resultados mais próximos possíveis daqueles observados nos circuitos reais;
- Realizar o projeto dos circuitos básicos do divisor, *flip-flops* tipo D e portas lógicas, com o uso da ferramenta Cadence na tecnologia TSMC (*Taiwan Semiconductor Manufacturing Company*) 0,18 μ m;
- Projetar os principais blocos do divisor completo visando, primeiramente, baixo consumo e velocidade de operação que proporcione fatores de divisões precisos;
- Desenvolver um divisor de frequências para um PLL que apresente consumo de aproximadamente 0,7 mW, fatores de divisão que variem entre 480-495, opere de forma correta em frequências no mínimo 400 MHz maior que 2,40 GHz e apresente fatores de divisão precisos e estáveis;
- Atualizar a modelagem em Verilog-AMS com os parâmetros obtidos no término do projeto do divisor para o PLL, tendo como objetivo a verificação da funcionalidade do divisor ao ser integrado no PLL;
- Incluir o PLL no transceptor do sistema ZigBee a ser utilizado numa rede de monitoramento WSN de um processo de irrigação.

1.2 Motivação: Aplicações em Agricultura

Atualmente, nota-se um grande desenvolvimento de novas tecnologias voltadas para diversos setores da agricultura. O surgimento de novas técnicas de produção e produtividade estão diretamente ligadas à necessidade de criação de novos equipamentos, como máquinas e insumos, novas técnicas de produção e acompanhamento do plantio. Esta modernização da agricultura no Brasil veio com o intuito de acompanhar e se adaptar as exigências do mercado internacional tendo como objetivo final o aumento da produção, melhora na qualidade do produto, redução nos custos e na poluição [Teruel et al. (2012)].

Com o advento dessa modernidade, os sistemas de monitoramento remoto surgiram como uma solução para avaliar as principais características que influenciam no plantio, crescimento e colheita de produtos agrícolas. Estes sistemas utilizam sensores espalhados pelo campo e são utilizados para monitorar o crescimento de plantações [Ndzi et al. (2014)], irrigação [Romero et al. (2012)], temperatura, pH do solo [Kalaivani, Allirani e Priya (2011)].

Citando um dos exemplos mencionados acima, a irrigação tem como finalidade proporcionar o fornecimento de água em épocas de estiagem e aperfeiçoar o sistema natural de precipitação da água visando, juntamente com a adubação, o controle de pragas, doenças e aumento da produtividade na lavoura. A irrigação garante a boa qualidade do alimento, porém, é um insumo que desperdiça muita água. Um estudo realizado pela Organização das Nações Unidas (ONU) chegou à conclusão que 70% da água disponível no planeta é empregada na irrigação e que, no Brasil, este valor é de 72% [Walbert (2006)].

Segundo a Organização das Nações Unidas para a Alimentação e Agricultura (FAO, em inglês), em torno de 60% da água utilizada para irrigação é desperdiçada devido a fenômenos naturais, como a evaporação. Caso esse desperdício fosse reduzido em 10%, haveria água suficiente para abastecer o dobro da população mundial de 2006 [Walbert (2006)].

Fica evidente a necessidade de monitorar o processo de irrigação, avaliando quais partes da plantação necessitam de uma maior injeção de água e quais partes já estão bastante irrigadas e não precisam de água no momento. Por meio da identificação destes pontos estratégicos o sistema libera água somente quando necessário, ou seja, quando a umidade do solo está em níveis abaixo do mínimo tolerado. Com isso, o agricultor torna seu sistema mais eficiente, gasta apenas o necessário de água, diminui os gastos financeiros e contribui para a economia de água no planeta. Sistemas WSN apresentam uma opção de aplicação no contexto citado acima e o protocolo ZigBee se destaca ao utilizar sensores pequenos e por proporcionar um sistema confiável e de baixo consumo de energia. Isso motivou a formulação deste trabalho, que é desenvolver um divisor de frequências, com baixo consumo de energia, para um PLL a ser utilizado no transceptor do sistema ZigBee.

O protocolo ZigBee é um padrão que possui boa eficiência, como citado anteriormente, e é um forte candidato para ser utilizado em sistemas de irrigação. Um exemplo que pode ser mencionado é o trabalho feito por [Patil S. S.; Davande (2014)], no qual implementou um sistema de aquisição de dados WSN que monitorava o ambiente de uma plantação. Umidade do ar, do solo, temperatura e nível de água são os dados que foram estudados. A informação dos sensores é coletada por um nó coordenador, utilizando toda sua capacidade de processamento, que se comunica com o ZigBee de forma serial. Da mesma forma o nó coordenador pode trocar informações com um computador realizando um monitoramento em tempo real. Neste sistema foi utilizado um microcontrolador 89C52 que pode armazenar entre oito e dez leituras na memória RAM a cada meia hora, gerando um grande volume de informações a cerca do sistema. A figura 2 exemplifica a aplicação deste sistema.

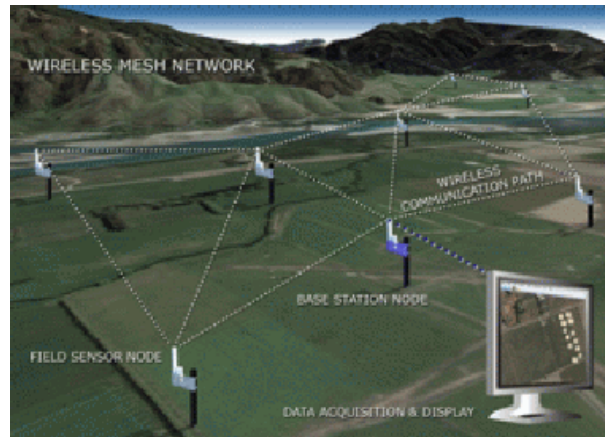


Figura 2 – Implementação dos sensores de aquisição de dados por Patil S. S.; Davande (2014).

1.3 Organização do Trabalho

Esta seção tem como objetivo explicar a organização desta monografia. Para facilitar o entendimento do leitor, este trabalho foi dividido em oito capítulos.

No capítulo 1 é feita uma introdução ao sistema a ser implementado e é apresentada a motivação, que também é uma aplicação deste sistema. Neste capítulo é discutido sobre o protocolo ZigBee o porquê de utilizá-lo e também é apresentado os objetivos gerais e específicos deste trabalho. Após esta apresentação, no capítulo 2 é abordado as metodologias utilizadas no projeto de circuitos integrados analógicos, bem como os prós e contras de cada uma e como a metodologia escolhida será adotada. Em seguida, no capítulo 3 é apresentada a importância que a linguagem de descrição de *hardware* Verilog-AMS têm para a metodologia escolhida no capítulo anterior, os motivos de sua criação, bem como a estrutura de um programa escrito nesta linguagem.

No capítulo 4 é exposto as principais características de uma rede de sensores WSN e do protocolo ZigBee. No capítulo 5 é feita uma introdução sobre os conceitos básicos do PLL, suas principais aplicações, diagrama de blocos, tipos de PLL, blocos internos e como eles funcionam para proporcionar uma boa performance ao PLL. No capítulo 6 são apresentadas as arquiteturas básicas e a implementação de circuitos divisores que operam em altas frequências. Também apresenta a topologia *Pulse Swallow Divider*, sua divisão em blocos e como eles se comportam para dar a funcionalidade desejada ao divisor do PLL. Subsequentemente, no capítulo 7 é apresentado os resultados do divisor projetado neste trabalho. E, por fim, no capítulo 8 é feita a conclusão, síntese de todo o projeto e possíveis trabalhos futuros.

Embora um sistema integrado que possua apenas circuitos digitais possa ser projetado em poucos meses e funcione corretamente logo na primeira fabricação do *chip*, um *chip* analógico demanda um tempo maior para ser projetado, aproximadamente 4 *tape-outs* para que funcione da forma esperada, embora isso não seja uma regra. Este cenário ocorre pelos seguintes motivos [Kundert e Zinke (2004)]:

- Grande incompatibilidade, risco envolvido e dificuldade ao integrar circuitos analógicos e digitais na simulação mista (simulações em conjunto de circuitos analógicos e digitais);
- O aumento nos riscos de projeto se tornaram um obstáculo para o planejamento do mesmo. Isso dificulta a previsão de quando o produto estará finalizado;
- Como, em alguns casos, o projeto precisa ser refeito algumas vezes, isso faz com que o tempo de projeto aumente consideravelmente e que o mesmo possa se tornar obsoleto quando finalizado. Isso se deve a alta rotatividade de produtos no mercado;
- Sistemas analógicos e de simulação mista necessitam de um grande investimento de tempo e dinheiro para serem realizados;
- Pouca disponibilidade de profissionais especializados e aptos para trabalhar na área;
- Alguns sistemas estão se tornando tão grandes, com tantos circuitos interligados que tornam-se difícil de serem projetados devido a baixa produtividade, mesmo com um time apresentando um número considerável de profissionais.

As empresas de semicondutores tem como principal estratégia construir produtos avançados, de alta qualidade, com base em uma arquitetura que incorpora *hardware* e *software* de ponta, bem como uma tecnologia de alta performance. Esta estratégia proporciona alta densidade de integração, bom desempenho e boas técnicas de encapsulamento, assim, oferecendo uma caracterização do produto em questão de aspecto, função, tamanho e custo. Para que isso seja possível, são implementadas técnicas de fluxo de projeto, no qual a *bottom-up* e o *top-down* são as principais metodologias utilizadas no fluxo de projeto de circuitos integrados, de forma a atender as deficiências citadas acima.

2.1 *Bottom-Up*

A forma mais tradicional de metodologia de projeto é a *bottom-up* no qual o projeto tem os seguintes passos: confecção dos blocos individuais a nível de transistor; verificação de cada bloco individualmente e, ao fim, todos os blocos são unidos e o sistema é simulado. Cada bloco possui uma especificação, a partir daí o projeto é iniciado e tem fim apenas na sua efetivação a nível de transistor (figura 4). Assim, os blocos são projetados e verificados

individualmente para se obter as especificações desejadas visando apenas este bloco e não o sistema como um todo. Isso pode gerar diversas complicações quando o sistema for montado e testado [Kundert e Zinke (2004)].

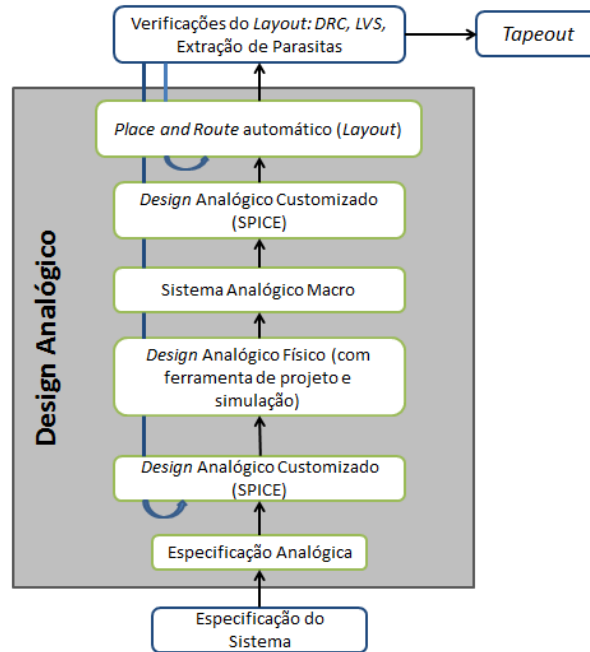


Figura 4 – Diagrama de blocos da metodologia *bottom-up* [McCorquodale et al. (2003)].

A abordagem do *bottom-up* é popularmente usada em sistemas autônomos, pequenos e adaptativos no qual a comunicação entre a equipe de projetos é mínima. Para projetos pequenos a metodologia *bottom-up* é eficiente, porém, para sistemas mais robustos surgem alguns problemas [Kundert e Zinke (2004)], como:

- A simulação do sistema montado é demorada e bastante complexa no qual, em alguns casos, é praticamente impossível de ser realizada;
- Em sistemas mais complexos a arquitetura do projeto tem grande impacto na performance, custo e funcionalidade. Como o *bottom-up* se preocupa pouco com a arquitetura, as especificações citadas acima são negligenciadas;
- Como os projetistas trabalham de forma separada, cada equipe se preocupa apenas com o projeto de um bloco específico, não há muita comunicação entre as equipes. As equipes discutem o projeto apenas na verificação do sistema completo, ou seja, no fim do projeto;
- Como existe pouca comunicação entre as equipes, podem aparecer alguns problemas na montagem do sistema. Se isso acontecer de fato, o sistema deverá ser reprojetoado, assim, perdendo mais tempo e dinheiro.

O diagrama de blocos da figura 4 mostra de maneira simplificada cada etapa do projeto utilizando a metodologia *bottom-up*.

2.2 Top-Down

Nesta metodologia o projeto tem início na parte mais abstrata e vai até o nível mais objetivo do sistema. Primeiramente é feita a escolha da arquitetura a ser utilizada com base nas especificações dadas à equipe, sem muito detalhamento. A partir disso, o processo de refinamento tem início, gerando subsistemas que são avaliados e projetados de forma a obter as características do projeto solicitado. Este refinamento é realizado até que o nível mais básico seja atingido, obtendo assim, blocos projetados a nível de transistor e bastante detalhados. Isso facilita o trabalho dos projetistas, pois no *top-down* a comunicação entre os projetistas é formalizada permitindo que o trabalho seja realizado de maneira mais produtiva e rápida se comparado com o *bottom-up* [Kundert e Zinke (2004)].

Uma grande vantagem desta metodologia é a redução dos impactos que possíveis mudanças no projeto causam no sistema final. Existem casos em que o circuito necessita ser reprojetoado e quando isto acontece, esta metodologia permite que a mudança e o levantamento dos impactos disso no sistema sejam levantados de forma rápida. Assim, minimizando os aspectos negativos do reprojeto.

A figura 5 exemplifica um diagrama de blocos da metodologia *top-down* que apresenta os seguintes princípios de projeto [Kundert e Zinke (2004)]:

- O *design* é feito de tal forma que os projetistas se comunicam durante todas as etapas do projeto. Como todos tem conhecimento do *design*, as descrições dos circuitos e *layout* podem ser co-simulados por qualquer membro da equipe;
- O projeto é feito de forma direta, ou seja, apenas quando um bloco está totalmente finalizado e funcionando que o próximo bloco é feito. Assim, cada mudança feita num bloco é analisada no contexto global do projeto e dos blocos projetados anteriormente;
- O processo de verificação é realizado com cautela e com um planejamento prévio que avalia os riscos de forma a diminuí-los;
- O processo de *design* se inicia no nível mais alto e abstrato do sistema e vai sendo refinado durante todo o processo com o objetivo de se obter a melhor estimativa e quantidade de dados sobre os blocos a serem projetados;

- Ao invés de utilizar documentos escritos, as especificações são geradas e entregues na forma de programa executáveis. Isso permite que as especificações sejam compreendidas de maneira mais rápida e holística;

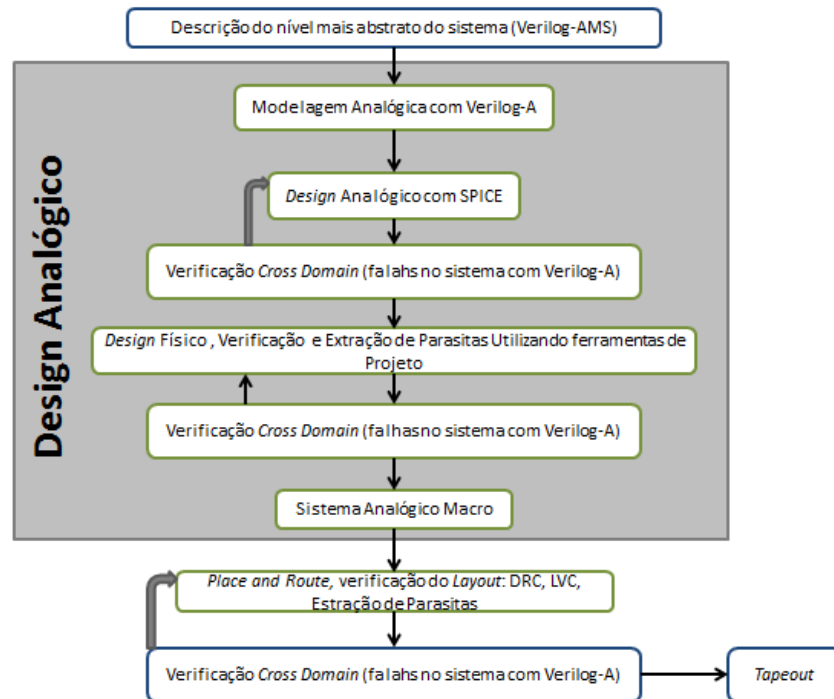


Figura 5 – Diagrama de blocos da metodologia *top-down* [McCorquodale et al. (2003)].

2.3 Escolha da Metodologia

Devido as características citadas anteriormente e as vantagens que o *top-down* possui em comparação ao *bottom-up*, neste trabalho será utilizada a metodologia de projetos *top-down*.

3 Linguagem de Descrição de *Hardware*

A complexidade de sistemas eletrônicos vêm aumentando de forma considerável, como foi profetizado por Gordon Moore dizendo que a quantidade de transistores presentes num circuito integrado dobra a cada dois anos aproximadamente. Com isso, novas metodologias de projeto, bem como linguagens de programação, foram criadas para dar suporte na construção e verificação de sistemas grandes e complexos como o VLSI (*Very-large-scale integration*). Segundo Palnitkar (2003), era impossível realizar a verificação manual de sistemas VLSI, pois as ferramentas computacionais, bem como as linguagens de programação disponíveis como FORTRAN, Pascal e C, não forneciam a estrutura necessária para tal finalidade. Através desta necessidade que surgiu a linguagem de descrição de hardware (HDL) no qual o Verilog HDL e VHDL (*VHSIC Hardware Description Language*) são as mais populares.

O HDL tem como objetivo dar suporte a modelagem temporal e projeto de *hardware*. Possui alto nível de abstração, proporciona a descrição das funcionalidades de um bloco, a aplicação e análise de erros antes que o mesmo seja fabricado. Esse suporte é fornecido através da caracterização do seu funcionamento e por meio de passos no qual todos eventos e processamento dos sinais de entrada são descritos em linguagem de programação. Esta ferramenta também dá suporte a simulações onde se cria um arquivo descrevendo todos os tipos de eventos possíveis de serem realizados no ambiente em que o *hardware* será aplicado.

3.1 Verilog-AMS

Circuitos eletrônicos integrados (IC) atuais apresentam blocos digitais e analógicos. Segundo FitzPatrick e Miller (1998), o aumento do nível de integração entre esses dois mundos têm levado o projeto de IC's para um novo mundo, o *mixed signal*. O Verilog-AMS surgiu como uma alternativa criada pelo IEEE (Instituto de Engenheiros Eletricistas e Eletrônicos) de forma a estender o Verilog HDL para a simulação e verificação de circuitos analógicos e sistemas *mixed signal*. Seu uso permite que os projetistas criem e utilizem módulos com alto nível de descrição, bem como uma estruturação de componentes que são descritos matematicamente de acordo com suas portas e parâmetros aplicados ao bloco.

A linguagem Verilog HDL se tornou padronizada pela IEEE desde 1995 com o IEEE 1364. Seu uso principal está relacionado com a verificação de sistemas digitais criada com o objetivo de ser simples, intuitiva e eficiente permitindo um alto nível de abstração do sistema e a criação de uma grande variedade de módulos de simulação. Verilog-A tem como base a definição de sistemas analógicos, tornando possível a criação de blocos,

sistemas e componentes com alto nível de descrição e detalhamento, sendo aplicável em sistemas elétricos e não elétricos. Segundo [FitzPatrick e Miller \(1998\)](#), esta linguagem é o resultado do desenvolvimento de uma padronização para a descrição e simulação de sistemas analógicos e *mixed signal* criada pelo Open Verilog International (OVI), agora continuada pelo IEEE. O Verilog-AMS dá base para que o projetista possa lidar com sistemas analógicos, digitais e *mixed signal* (MS). Isso é possível pois o Verilog-AMS incorpora elementos tanto do Verilog HDL como do Verilog-A (Figura 6).

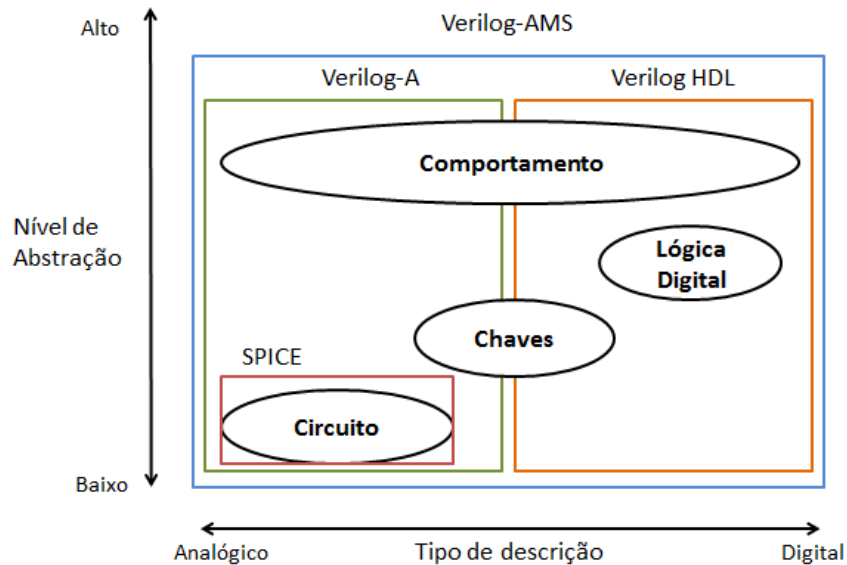


Figura 6 – Estrutura da linguagem Verilog-AMS [[FitzPatrick e Miller \(1998\)](#)].

Nível de abstração está relacionado com a obtenção de detalhes. É a habilidade de representar ou descrever algo de maneira objetiva, concisa e abstrata sem expressar muitos detalhes. Quanto maior o nível de abstração menor é a quantidade de detalhes disponíveis e apresenta uma descrição e compreensão mais simples. Em contrapartida, quanto menor o nível de abstração, a quantidade de informação disponível é maior, porém, com difícil descrição e complicada compreensão.

É possível utilizar o Verilog-AMS para modelar sistemas ou componentes eletrônicos, criar *test benches*, fazer verificação MS, realizar simulações no tempo discreto ou contínuo, assim, suas funcionalidades ajudam a acelerar o tempo de simulação. É uma linguagem bastante versátil, pois lida com diversos níveis de abstração, incluindo modelagem ideal e não ideal, modelagem funcional e estrutural [[Gang \(2009\)](#)].

Trata-se de uma ferramenta importante pois pode ser utilizada no início de um projeto para verificar o funcionamento completo do sistema, assim, unindo e gerando uma maior troca de informações entre os desenvolvedores do sistema e dos blocos individuais do mesmo. Isso faz com que a verificação seja feita e analisada de forma holística. Assim, esta linguagem é bastante utilizada no início do fluxo de projeto da metodologia *top-down* avaliando se a arquitetura escolhida atende as especificações do projeto, e obtendo-se

mais informações sobre o mesmo, para só assim dar início ao projeto a nível de transistor. Também é empregada na verificação em comparação as especificações originais e ao projeto em nível de transistor. Uma segunda opção de projeto é descrever parte do sistema em Verilog-AMS, projetar a outra parte a nível de transistor e realizar simulações mistas do sistema.

3.1.1 Estrutura da Linguagem Verilog-AMS

Todos os códigos em Verilog-AMS seguem um padrão de construção exemplificado na figura 7. Nesta figura é explicitado a modelagem funcional de um conversor *flash* analógico/digital.

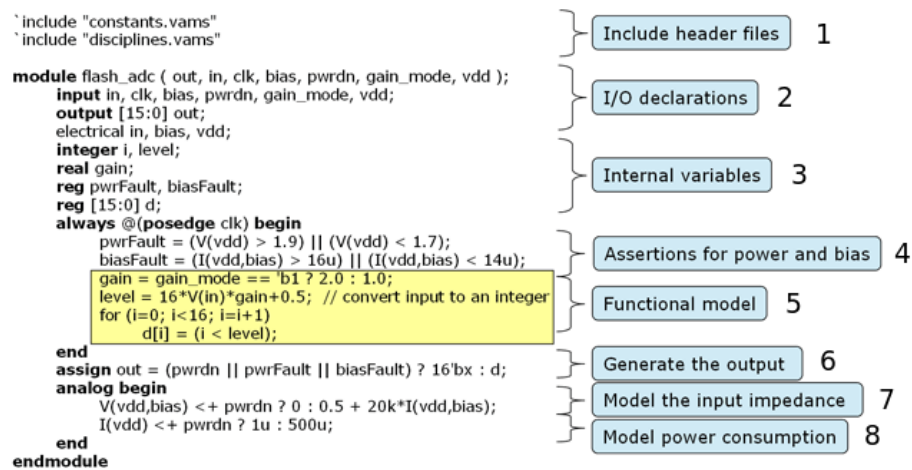


Figura 7 – Modelagem funcional em Verilog-AMS de um *flash* ADC [Kundert K.; Chang (2007)].

A parte mais básica da linguagem Verilog-AMS é a estrutura *module* onde são definidas as entradas (*input*) e saídas (*output*), os tipos de variáveis a serem utilizados, e descrição dos blocos individuais. Os números citados nos itens a seguir estão relacionados aos números apresentados na figura 7. Os principais elementos do Verilog-AMS são:

- disciplinas: é uma coleção de sinais físicos de diversas origens. Podem ser elétrica, mecânica, envolvendo dinâmica dos fluídos ou termodinâmica, por exemplo. Em (1) *disciplines.vams* define a coleção mais comum de disciplinas e naturezas;
- natureza: cada disciplina possui sinais de diversas naturezas. A disciplina elétrica, por exemplo, consiste nas naturezas corrente e tensão. Como descrito em (2), a disciplina *electrical* utiliza as naturezas tensão (vdc) e corrente (i);
- parâmetros: são denominações que especificam valores que são constantes durante a operação do bloco, embora possa ser alterado no decorrer do *teste bench*. Por exemplo, *parameter real r=0* representa um parâmetro r com valor inicial constante e igual a zero do tipo real. Este valor pode ser alterado durante qualquer instante;

- variáveis internas: representam sinais internos ou intermediários do sistema, assim, não são de entrada ou saída. São criados para se comportarem em função dos sinais de entrada e serem usados como referencia para os sinais de saída. Isso pode ser visto em (4) e (5);
- portas: são pontos onde as conexões podem ser feitas. Na figura 7, as portas e as direções são as entradas e saídas descritas em (2). Um sinal pode ter as direções entrada, saída ou ser bidirecional;
- corpo: é nesta parte que se encontra a essência do sistema, no qual é descrita todas as suas funcionalidades e a forma que os sinais são manipulados. Em Verilog, o corpo tem início com a descrição das funcionalidades básicas do sistema, já em Verilog-A e Verilog-AMS o mesmo tem início com *analog begin* ou *always begin*. Quando o funcionamento do circuito depende de um sinal de controle externo é utilizado um @ após a palavra *analog*. Na figura 7, isso é especificado com *@(posedge clk)* no qual o sistema é ativado com um sinal de clock;
- *Wreal*: trata-se de uma funcionalidade capaz de alterar o tipo de um sinal elétrico em uma variável real e discreta no tempo. Esse tipo de sinal é bastante utilizado pois torna possível diminuir o tempo de simulação uma vez que a parte analógica é simulada com o *software* digital.

4 Redes de Sensores sem Fio

Este capítulo tem como finalidade apresentar uma breve introdução sobre redes WSN e as principais características do protocolo ZigBee.

4.1 Introdução

Redes sem fio surgiram como uma alternativa de transmitir informação de forma simples e segura sem a necessidade de cabos, proporcionando assim, a mobilidade do usuário. Com o passar dos anos novas tecnologias e protocolos de comunicações foram criados para atender a demanda desse mercado. Um exemplo é a criação e aprovação da norma IEEE 802.11 que revolucionou o sistema de redes sem fio da época e que vem crescendo rapidamente desde então.

As redes *wireless* podem ser utilizadas em áreas como a internet, celulares, sistemas de identificação por radiofrequência (RFID), sensoriamento e monitoramento, entre outros. O sensoriamento e monitoramento de dados vêm se destacando nos últimos anos devido à necessidade de levantar informações de um sistema, processar estes dados e desenvolver métodos que aperfeiçoem o funcionamento do ambiente estudado. Um exemplo disso é a utilização de *smarth grid* na rede elétrica de uma empresa, possibilitando a obtenção de dados como tensão, corrente, temperatura, entre outros. O monitoramento é feito espalhando-se sensores no meio em que se deseja obter alguns dados do funcionamento interno ou de seus arredores [Lewis (2004)]. A essa rede de sensores é dado o nome de *Wireless Sensor Network* (WSN).

Baixo custo, baixo consumo de energia, sensores compactos e capacidade de integração com um grande número de sensores são algumas das principais características do WSN. Outra característica importantíssima é a forma como a informação é transmitida. Os sensores não enviam dados diretamente para a central de controle, dessa forma, os eles são transferidos de sensor a sensor, até que a central obtenha e processe esta informação [Akyildiz I. F.; Su e Cayirci (2008)]. Isso permite a integração de um sistema de baixo alcance, como o ZigBee, em um ambiente com grande área a ser monitorada.

O protocolo ZigBee é um exemplo de sistema WSN e se destaca por incorporar as características citadas acima, apresentar baixa taxa de transferência de dados e baixa latência. O padrão ZigBee foi desenvolvido para ser uma solução que apresentasse baixa complexidade e que fosse possível ser inserido em uma grande gama de aplicações nas áreas de monitoramento e controle. Assim, fica claro que o ZigBee é uma tecnologia simples que proporciona alta flexibilidade de dispositivos a serem usados no sistema.

Dentre as aplicações do Zigbee se destacam: automação residencial, como controle de iluminação, refrigeração; aplicações militares; sistemas de identificação de animais (RFID); monitoramento de dados biométricos; telemetria; agricultura como irrigação, fertilização, controle de doenças, controle de estufas, entre outros.

4.2 Tipos de Rede WSN

Uma rede de comunicação é composta por diversos nós, dotados de capacidade computacional, que recebem e transmitem informação em um sistema *wireless* ou por fio. Estas redes apresentam algumas topologias que são indicadas de acordo com a aplicação. Interconexão, malha, estrela e anel são algumas destas topologias (figura 8).

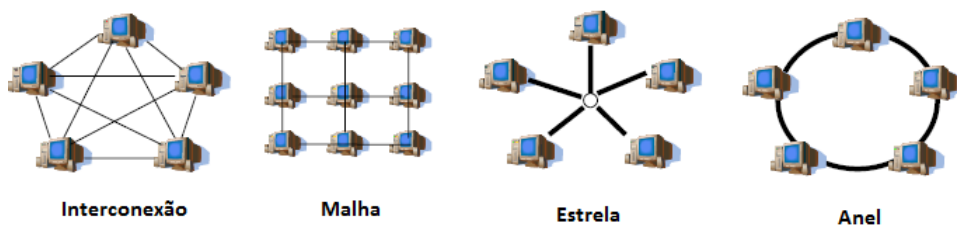


Figura 8 – Topologias de rede de comunicação WSN [Lewis (2004)].

Interconexão, como o nome já diz, todos os nós estão ligados. É aconselhada para redes com poucos nós, já que o número de conexões aumenta exponencialmente com o número de nós, exigindo grande capacidade computacional.

Redes em malha permitem transmissão de dados apenas para nós próximos. Esta rede é indicada para aplicações em larga escala, onde há a necessidade de monitoramento de um ambiente grande, pois, caso haja falha de algum nó o sistema pode funcionar sem grandes problemas devido aos diversos caminhos que podem ser percorridos entre os nós e pela escolha de um novo líder de grupo. Esses “líderes de grupo” são nós que recebem algumas funções adicionais. Como todos os nós são idênticos, caso haja falha em algum deles, outro nó pode receber a função de líder [Lewis (2004)].

Todos os nós da topologia estrela estão conectados, diretamente, com uma central. Dessa forma, a central é bastante exigida, pois lida com uma grande fluxo de mensagens. Por fim, todos os nós apresentam a mesma função na topologia anel e a informação é transmitida em uma única direção. Um problema desta topologia é que caso algum link seja cortado, todo sistema de comunicação falha.

4.3 Características da Rede ZigBee

A Zigbee Alliance é um grupo de empresas que, juntamente com o IEEE (*Institute of Electrical and Electronics Engineers*), gerencia e distribui o padrão ZigBee. Ambos trabalham juntos visando o desenvolvimento e correta aplicação do ZigBee de forma a proporcionar um sistema *wireless* seguro, com baixa potência e baixo custo.

A técnica *listen-before-talk* foi incorporada ao Zigbee e consiste em analisar o sistema de rádio, ou seja, recebe algum tipo de informação antes que a transmissão tenha início. O protocolo é projetado para permitir que os sensores possam aparecer ou desaparecer do WSN, ou seja, quando não ativo, é possível colocar o sensor em modo de baixo consumo de forma que apenas os sensores que realmente estão mandando informações estejam ligados [Ltd (2008)]. A tabela 1 apresenta as principais características do protocolo ZigBee.

Tabela 1 – Bandas de frequência e taxas de dados do ZigBee [Ergen (2004)].

PHY(MHz)	Faixa de Frequência (MHz)	Região	Taxa de transferência (kbps)	Quantidade de canais disponíveis	Modulação
868	868-868.8	Europa	20	1	BPSK
915	902-928	Estados Unidos	40	10	BPSK
2450	2400-2483.5	Global	250	16	BPSK

Existem três principais bandas de rádio frequências que são utilizadas pelo Zigbee e cada uma apresenta uma quantidade definida de canais de operação: 868 (1 canal), 915 (10 canais) e 2400 MHz (16 canais). Essas três bandas estão disponíveis para uso sem licença e são especificadas para determinadas áreas, com diferentes características, sendo elas: banda de 868 MHz possui taxa de transmissão de 20 Kbps, utilizada na Europa; banda de 915 MHz possui taxa de transmissão de 40 Kbps, utilizada nas Américas e Austrália e, por fim, a banda de 2,4 GHz apresenta taxa de transmissão de 250 Kbps e é utilizada mundialmente. As bandas de 868 e 915 MHz apresentam vantagens como poucos usuários, pouca absorção e reflexão de sinal. Entretanto, a banda de 2,4 GHz apresenta cobertura global, alta taxa de transmissão, baixo consumo, sendo assim, a mais utilizada [Ltd (2008)].

O padrão Zigbee tem como base o protocolo IEEE 802.15.4 que define duas camadas de transmissão, a camada física (PHY) e a camada de controle e acesso ao meio

(MAC) (figura 9). Já a Zigbee Alliance está encarregada a proporcionar a camada de *network* (SWK) e a camada de aplicação (API) [Wang, He e Wan (2011)]. A figura 9 apresenta uma melhor visualização dessas camadas.

A camada PHY caracteriza a parte física, ou seja, qual frequência o sistema irá trabalhar. A MAC controla o acesso ao canal de rádio. Já a camada SWK é responsável por registrar informação nos sensores vizinhos. Por ultimo, a camada API possui as aplicações que são executadas no nó da rede, dando, assim, funcionalidade aos dispositivos.

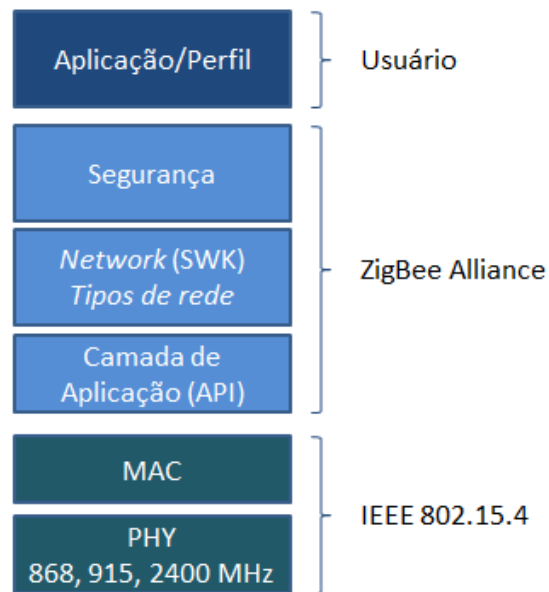


Figura 9 – Camadas definidas pelos protocolos ZigBee e IEEE 802.15.4.

5 Estrutura do *Phase-Locked Loop* (PLL)

Este capítulo tem como finalidade apresentar ao leitor os fundamentos essenciais para o completo entendimento deste trabalho e criar uma base teórica que possa ser associada à aplicação discutida anteriormente. Neste capítulo será apresentado os conceitos fundamentais de um *Phase Locked Loop* (PLL).

5.1 *Phase-Locked Loop* (PLL)

O século 20 ficou marcado no mundo de telecomunicações devido ao grande avanço tecnológico conquistado nesta época. Técnicas de modulação e demodulação de sinais foram desenvolvidas, bem como novos blocos como o *Phase-Locked Loop* (PLL). O PLL apresenta larga utilidade e grande importância em sistemas *wireless*, telecomunicações, rádio e surgiu como uma alternativa a necessidade de osciladores síncronos. Atualmente é utilizado em sistemas integrados de escala comercial, onde baixo consumo, aprimoramento da área do chip e redução de ruído são essenciais, sendo essa uma grande vantagem deste bloco [Kroupa (2003)]. PLL's de baixo consumo possuem aplicação em sistemas WSN (onde o gasto energético é crítico), modulação e demodulação, principalmente na *Quadrature amplitude modulation* (QAM), transceptores, bem como em sintetizadores de frequência que geraram um sinal estável múltiplo da frequência de entrada.

Em suma, o PLL é um sistema de realimentação negativa, sendo, neste trabalho, uma realimentação digital já que o PLL é digital, que combina uma série de estágios com o objetivo de entrar em sincronismo com a frequência de entrada estável (normalmente um oscilador de cristal), apresentando uma pequena, e constante, diferença de fase. A diferença de fase pode ser constante mas pequena ou igual a zero, isto depende do tipo do PLL que será utilizado. Este sincronismo é cometido por meio de repetidas comparações entre a diferença de fase de entrada e saída até que o sistema entre em equilíbrio (modo *locked*).

A realimentação citada acima utiliza um detector de fase e frequência (PFD, em inglês), *charge pump* (CP), filtro passa baixas (LPF), oscilador controlado por tensão (VCO) e um divisor de frequências (DF). O sinal de referência é conectado ao PFD que verifica a diferença entre as fases deste sinal com a saída do VCO que foi dividida pelo DF. Em seguida, o LPF atenua as altas frequências da componente AC gerada pelo CP, define a forma do ruído gerado pela diferença entre as fases e mantém a dinâmica e a estabilidade do PLL [Golio e Golio (2010)]. Através dos dados filtrados pelo LPF, o VCO produz uma oscilação que é proporcional ao sinal gerado pelo PFD. A variação na tensão de controle do VCO faz com que a diferença de fase decaia ou aumente gradualmente até o momento que

a mesma fique constante, indicando que ambos possuem mesma frequência e o sistema trava (*locked*), entrando em equilíbrio. Quando o sistema está travado a frequência de saída $F_{out} = M.F_{ref}$, onde M é o fator de divisão do DF [Plassche, Huijsing e Sansen (2000)]. A figura 10 exemplifica o diagrama de blocos de um PLL, onde F_{vco} e F_{out} representam o mesmo sinal.

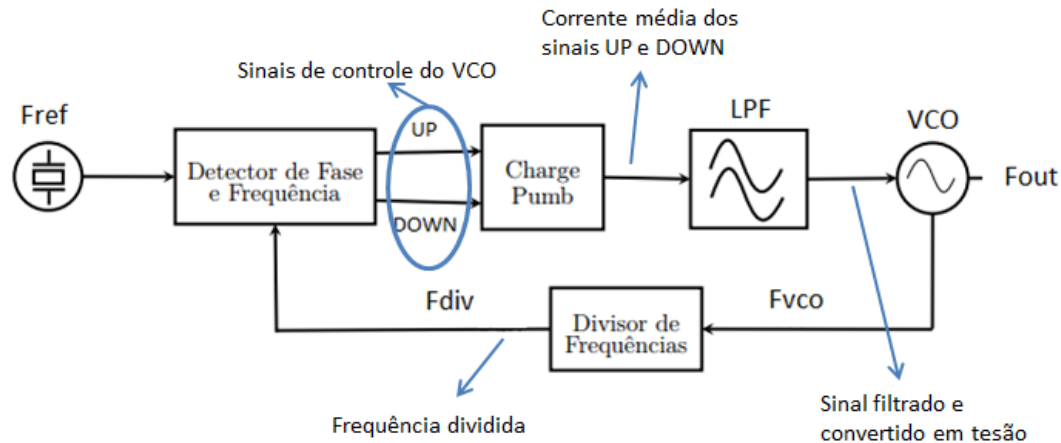


Figura 10 – Diagrama de blocos da topologia básica de um PLL.

Existem algumas especificações que devem ser avaliadas com cuidado no momento do projeto de um PLL. E são elas [Barrett (1999); Plassche, Huijsing e Sansen (2000)]:

- Supressão de componentes espectrais indesejadas: sinais de espúrio fazem parte do espectros de ruído e são caracterizados como qualquer parte discreta que não está diretamente ligada ao sinal. Têm como fonte o sinal de referência;
- Faixa de frequência: basicamente, a banda de frequência necessária para cada tipo de aplicação. Por exemplo, faixa predefinida para recebimento e transmissão de um transceptor. Neste trabalho, a faixa varia entre 2400 - 2475 MHz;
- Resolução da frequência: é o menor incremento de frequência possível, ou seja, o quanto a frequência do VCO pode aumentar a cada *loop*. É definido pelo espaçamento entre os canais de cada tipo de aplicação, ou seja, a frequência de referência que neste trabalho é de 5 MHz;
- Consumo: o consumo do sistema deve ser o menor possível, mas preservando as funções básicas e o bom desempenho. O VCO e o divisor possuem um grande papel no consumo de energia;
- Ruído da fase: indica a qualidade do sinal;
- Tempo de *switch* ou *settling time*: tempo que o PLL leva para estabilizar o VCO quando há uma mudança na frequência exigida na saída. Está diretamente ligado a resolução de frequência.

Utilizando a teoria de sistemas de controle e assumindo que o PLL seja contínuo no tempo cada bloco do PLL pode ser modelado no domínio de Laplace. Assumindo que o PLL está travado, é obtido a fórmula,

$$\frac{\theta_0(s)}{\theta_R(s)} = \frac{G(s)}{(1 + G(s)H(s))} = \frac{K_\theta F(s)(K_{VCO}/s)}{1 + K_\theta F(s)(K_{VCO}/s)(1/M)}, \quad (5.1)$$

para o ganho em malha fechada, onde $H(s)$ é a função de transferência do divisor e $G(s)$ é a multiplicação das funções de transferências dos demais blocos [Golio e Golio (2010)]. Neste exemplo, o *charge pump* foi excluído e o detector de fase é representado pelo ganho K_θ , $F(s)$ é a função do filtro e K_{VCO}/s a função do VCO no domínio da frequência. A fase do sinal de saída é $\theta_0(s)$ e a fase do oscilador a cristal conectado na entrada é $\theta_R(s)$.

5.2 Tipos de PLL

Nesta seção serão apresentados os tipos e as respectivas ordem do PLL. O tipo está relacionado com a quantidade de polos na origem da função de transferência do PLL em ganho aberto. Já a ordem do PLL se refere ao grau do denominador da função de transferência em malha fechada [Golio e Golio (2010)].

5.2.1 Tipo I

Neste tipo de PLL o detector de fase simplesmente verifica a diferença entre $\theta_0(s)$ e $\theta_R(s)$. A figura 11 apresenta o diagrama de blocos deste tipo de PLL, no qual o único integrador é o VCO e $L(s)$ representa a função de transferência do filtro que pode ser de primeira ou segunda ordem. De forma a melhor entender este sistema, será incluído um filtro de primeira ordem com $L(s) = \frac{1}{1+sRC}$. Assim, assumindo que o PLL esteja travado, a função de transferência em malha fechada é

$$\frac{\theta_0(s)}{\theta_R(s)} = \frac{K_\theta K_{VCO} L(s)}{s + K_\theta K_{VCO} L(s)/M} = \frac{K_\theta K_{VCO}}{RCs^2 + s + K_\theta K_{VCO}/M}. \quad (5.2)$$

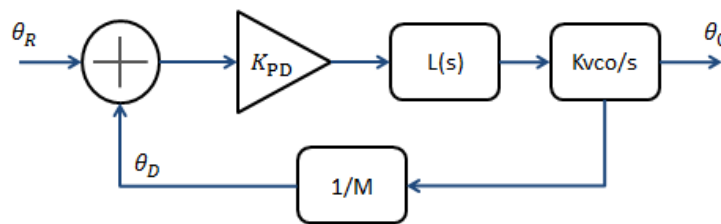


Figura 11 – Modelo de um PLL tipo I.

O ganho em malha aberta tende ao infinito quando $s \rightarrow 0$ devido ao polo na origem ocasionado pelo VCO. Filtros de ordem maior são necessários, mesmo com o deslocamento

da margem de fase, pois o filtro utilizado neste exemplo não suprime os sinais espúrios da referência e os ruídos intrínsecos aos blocos do PLL. A ordem do filtro tem que ser escolhida de forma a maximizar a atenuação de ruído e sinais espúrios mantendo o mínimo deslocamento da margem de fase possível [Golio e Golio (2010)].

Quando ocorre uma variação na entrada do PLL ou alteração no fator de divisão, já que ambos eventos apresentam a mesma reação no PLL, a diferença de fase entre $\theta_0(s)$ e $\theta_R(s)$ é uma constante. Como esta diferença é contante, então existe um erro de fase associado ao PLL tipo I, o que pode ser crítico para algumas aplicações. Neste contexto surge o PLL tipo II, que é utilizado em sistemas que exigem um erro de fase nulo.

5.2.2 Tipo II

A principal diferença se comparado com o tipo I é a inclusão do *charge pump*, que hora fornece e hora retira corrente do *loop filter* (figura 12), e a utilização de um detector de fase e frequência. Como consequência, o PLL não pode ser estudado no domínio da frequência s , pois trata-se não mais de um sistema contínuo mas sim discreto. Porém, se a largura de banda do PLL for menor que um décimo da frequência de referência é possível analisar o PLL tipo II no domínio de Laplace. A união do PFD com o *charge pump* é representada com um integrador. Como o VCO apresenta o mesmo comportamento na frequência, existem, assim, dois polos na origem o que caracteriza este PLL com tipo II [Manthena (2011)].

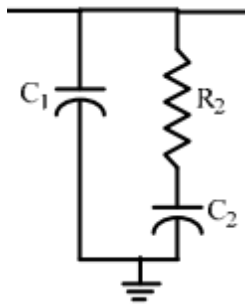


Figura 12 – *Loop filter* utilizado no PLL tipo II.

A função de transferência para o PLL tipo II é

$$H(s) = \frac{I_{CP}K_{VCO}}{2\pi MC_2} \frac{1 + sR_1C_1}{s^2 + s\frac{I_{CP}K_{VCO}R_2}{2\pi M} + \frac{I_{CP}K_{VCO}}{2\pi MC_2}}, \quad (5.3)$$

onde I_{CP} é o ganho de corrente do *charge pump*. Comparando a equação 5.3 com a função de transferência do segundo grau padrão utilizada na teoria de sistemas de controle, é

possível obter o fator de amortecimento e a frequência natural do PLL

$$H(s) = \frac{w_n^2}{s^2 + 2\xi w_n s + w_n^2}, \quad (5.4)$$

$$w_n = \sqrt{\frac{I_{CP} K_{VCO}}{2\pi M C_2}} e \quad (5.5)$$

$$\xi = \frac{R_2}{2} \sqrt{\frac{I_{CP} K_{VCO} C_2}{2\pi M}}, \quad (5.6)$$

onde w_n é a frequência natural e ξ é o fator de amortecimento do sistema. A resposta ao impulso apresenta uma exponencial com constante de tempo definida por

$$\tau = \frac{1}{2} \left(\frac{I_{CP} K_{VCO} R_2}{2\pi M} \right)^{-1}. \quad (5.7)$$

A equação 5.7 representa o *settling time*. Se os ganhos do *charge pump* e do VCO forem aumentados, então o *settling time* diminui. Porém, o aumento desses ganhos pode fazer com que a largura de banda seja maior que o um décimo da frequência de referência, impossibilitando toda a análise decorrida acima.

Por meio de pesquisa bibliográfica foi possível notar que o tipo II predominava nos PLL's utilizados para um transceptor ZigBee. Dessa forma, o divisor de frequências será aplicado neste tipo de PLL.

5.3 Blocos Básicos

O PLL utiliza fundamentalmente cinco blocos, ou estágios como mencionado anteriormente, em sua construção. Dependendo da tipo a ser usado é possível que mais alguns blocos sejam incluídos de forma a permitir que o PLL efetue diferentes funções. Cada estágio tem uma função única e contribui diretamente no correto funcionamento do PLL.

5.3.1 Detector de Fase e Frequências (PFD)

O detector de fase e frequência (PFD), como seu nome já diz, é um bloco que apresenta como saída um sinal proporcional a diferença de fase de duas frequências de entrada e tem como função ajustar o VCO para que estas frequências sejam iguais. No PLL, estas frequências são: frequência de referência (Fref) e a frequência de saída do VCO após a divisão pelo *prescaler* (F_{out}). A saída do PFD é representada por pulsos com algumas componentes AC (figura 13). A equação que rege o funcionamento do detector de fase é $v_e(s) = K_\theta[\theta_R(s) - \theta_0(s)]$, onde $v_e(s)$ é a tensão de erro, K_θ é o ganho do PFD, $\theta_R(s)$ e $\theta_0(s)$ representam a fase da entrada e da saída, respectivamente.

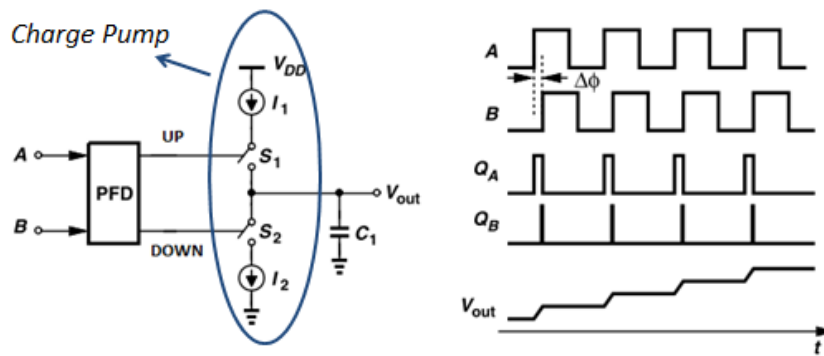


Figura 13 – Exemplo de um PFD utilizando *flip-flops* tipo D em conjunto com o *charge pump* representado por duas chaves com fontes de corrente [Razavi (2012)].

O PFD pode ser projetado de forma analógica ou com lógica digital, com porta XOR e XNOR, por exemplo. Mas a principal característica que deve ser levada em conta é o tipo de detector a ser utilizado. Embora exista o tipo sensível apenas à fase, é mais comum a utilização de um PFD sensível à fase e a frequência, pois este diminui consideravelmente a complexidade do PLL. As especificações de um PLL citadas anteriormente devem ser estudadas com cautela na hora de se escolher a topologia do PFD.

5.3.2 Charge Pump (CP)

Para integrar um PFD mais completo é necessário o uso de um *charge pump* (figura 13) que converte o sinal pulsante do PFD num sinal analógico. Uma das grandes vantagens do *charge pump* (CP) está na redução da complexidade do PLL. Este bloco é feito de duas fontes de corrente controladas pelos sinais *UP* e *DOWN*, que hora fornece hora retira corrente, respectivamente, e tem como saída um sinal que representa a média da diferença de fases do PFD (figura 14). Uma variável que pode injetar erro no sistema é a fuga de corrente presente no *charge pump*. Conforme esta fuga aumenta, será necessário um filtro passa-baixas mais robusto para suprimir o erro de ruído espúrio que irá resultar do VCO [Golio e Golio (2010)].

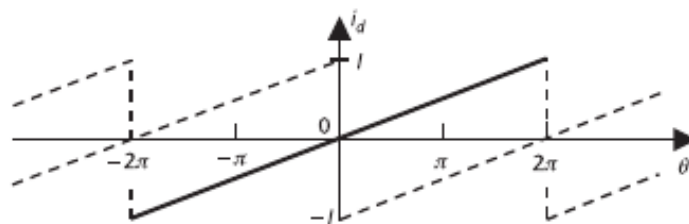


Figura 14 – Gráfico da corrente média (média dos pulsos QA e QB da figura 13) pela diferença de fase do PFD e *charge pump* [Rogers, Dai e Olett (2009)].

5.3.3 Loop Filter (LPF)

O sinal advindo do CP tem que ser tratado e transformado de corrente para tensão. Isto é feito conectando uma combinação de capacitores na saída do CP (figura 12). A este bloco é dado o nome de LPF e seu projeto é diretamente proporcional ao nível de sinais espúrios no PLL [Rogers, Dai e Olett (2009)]. A principal função do LPF é suprimir estes sinais espúrios, as componentes AC do PFD e garantir um sinal DC para a entrada do VCO. Quanto mais estreita for a frequência de corte deste filtro mais ruído será suprimido, porém, a estabilidade do *loop* e a dinâmica do PLL serão alterados, dificultando travamento do sistema, ou seja, entrando no modo *locked*. Assim, a frequência de corte deve ser definida com cuidado.

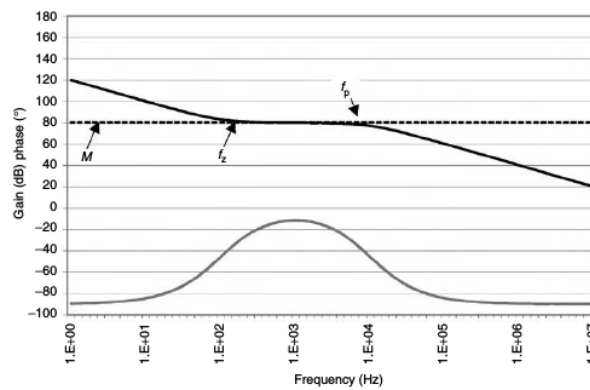


Figura 15 – Função de transferência típica de um LPF [Golio e Golio (2010)].

5.3.4 Oscilador Controlado por Tensão (VCO)

Por fim, o VCO tem como função gerar uma frequência diretamente proporcional ao sinal filtrado e tratado do LPF. É particularmente crítico para qualquer sintetizador de frequências, uma vez que este bloco define a frequência da oscilação na saída.

A frequência de saída do VCO pode ser modelada com

$$\omega_{VCO} = \omega_Q + K_{VCO}V_C(t), \quad (5.8)$$

onde ω_Q é a frequência quiescente ou frequência de repouso do VCO, K_{VCO} é ganho ou sensibilidade e $V_C(t)$ é a tensão de controle advinda do LPF. A figura 16 exemplifica o comportamento desta equação.

Para que haja um bom desempenho do VCO, visando a redução da diferença de fase, três passos devem ser seguidos. Primeiramente, utilizar um circuito ressonador com alto fator de qualidade. Diminuir a quantidade de componentes ativos no oscilador. E, por fim, minimizar o oscilador em anel presente no VCO. Isto pode ser feito escolhendo a melhor topologia para este bloco [Hsieh e Hung (1996)].

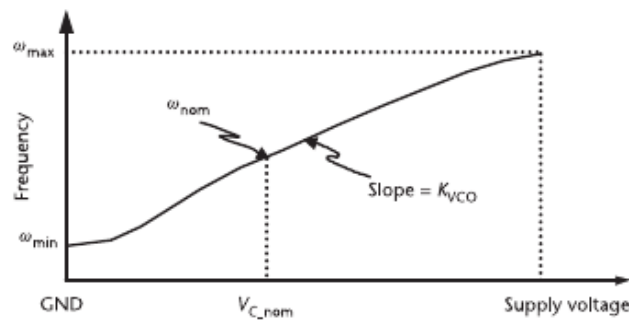


Figura 16 – Curva característica do VCO [Rogers, Dai e Olett (2009)].

5.3.5 Divisor de Frequências

O divisor de frequências é o bloco que multiplica a frequência de entrada com a saída do PLL por um fator M que neste trabalho é $M = NP + S$. Outra função deste bloco é dividir a frequência de saída do PLL para um nível da frequência de entrada de forma que quando o PLL esteja em modo *locked* a frequência dividida seja igual a referência. Dessa forma, o controle da frequência de saída depende diretamente da construção deste divisor (figura 17).

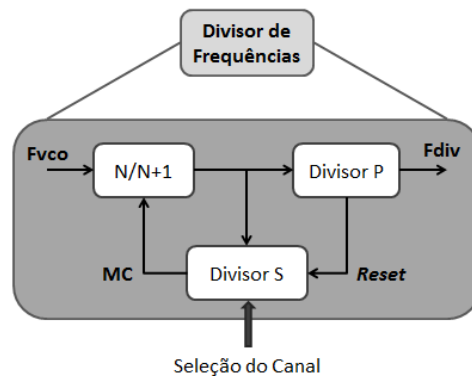


Figura 17 – Diagrama de blocos do divisor de frequências [Razavi (2012)].

Dependendo do contexto no qual o PLL será inserido, é possível que o sistema exija elevados fatores de divisão. A realização de um fator tão grande por um único divisor compromete o funcionamento do PLL e consome grandes quantidades de energia. Tipicamente, o divisor é projetado utilizando a técnica *Pulse Swallow Counter* por meio da integração de três divisores: *prescaler* com dois módulos de divisão; *swallow counter* (S) e contador principal (P). O *prescaler* é incluído na realimentação com o intuito de diminuir a frequência da entrada do divisor e minimizar o consumo total do sistema, pois os contadores auxiliares (P e S) trabalham em baixas frequências [Golio e Golio (2010)] e podem ser projetados utilizando técnicas que primam pelo consumo de energia em detrimento da velocidade de operação.

5.4 Tipos de Divisores para PLL

Nesta seção será explicado como funcionam e quais as maiores diferenças entre o PLL Inteiro-N e o Fracionário-N. As topologias apresentam estruturas semelhantes no sentido que apenas o divisor, que foi explicado na [subseção 5.3.5](#), é alterado.

5.4.1 Inteiro-N

Na topologia inteiro-N a frequência de saída do PLL é igual a frequência de entrada multiplicada por uma constante M, que apresenta valores inteiros como 2, 4, 8, por exemplo. A escolha da frequência de referência, bem como o fator de divisão M, são variáveis que são extremamente importantes e críticas no projeto de um PLL nesta topologia.

A frequência de saída pode ser modelada da seguinte forma: $f_{OUT} = f_0 + N \cdot f_{CH}$ onde f_0 é a menor valor da faixa de frequência, que neste trabalho é de 2400 MHz, e f_{CH} a frequência que separa dois canais consecutivos [Razavi e Behzad (1998)]. Nesta topologia, f_{CH} tem que ser exatamente igual a frequência de referência para que a seleção do canal seja realizada de forma eficaz, pois variando-se o módulo de M é possível escolher o canal de saída mais adequado. Esta limitação atinge diretamente a resolução de frequência e o ruído de fase do sistema, de forma que ambos ficam restringidos a determinados valores, o que pode atrapalhar o funcionamento do transceptor.

Em alguns casos, a implementação de um divisor com valor N se torna inviável. Em sistemas que utilizam a tecnologia GSM (*Groupe Special Mobile*) a resolução de frequência está em torno de 200 kHz, porém o espectro é de 1 GHz. Seria necessário um fator de divisão igual a $1 \text{ GHz} / 200 \text{ kHz} = 5000$ para que o sistema funcionasse. Um fator tão grande aumenta a amplificação de ruídos advindos do sinal de referência, o ruído de fase e eleva o consumo do divisor algo não desejado.

Dessa forma, o *prescaler* com dois módulos é utilizado de forma a realizar divisões por valores pequenos (N ou N+1), diversas vezes (definidas por S e P) com o intuito de, ao final, o módulo total da divisão ser alto como no exemplo acima. Esta topologia será tratado com mais detalhes nas seções seguintes.

5.4.2 Fracionário-N

Esta topologia possui uma vantagem em relação ao inteiro-N no que se relaciona com o fator de divisão que agora pode ser fracionário e a frequência de referência pode ser maior que o espaçamento entre os canais. Este valor fracionário é obtido variando-se o fator de divisão entre dois (*prescaler* de dois módulos) ou mais valores (divisor multi módulos). Assim, a média do fator de divisão é fracionário, o que permite alterar com liberdade a frequência de referência para valores maiores. Isso possibilita um PLL com

menor ruído de fase e aumento do *settling time* [Rogers, Dai e Olett (2009)].

Aqui, o *prescaler* de dois módulos também é utilizado com uma alteração na forma que o mesmo é controlado. O controle é feito por meio de um modulador sigma-delta que altera o fator de divisão de forma randômica com o objetivo de reduzir sinais espúrios gerados por não linearidades. Isso é feito alterando banda-base desses sinais para altas frequências que serão suprimidas pelo LPF. O fator de divisão passa a ser $M + N(K)$, onde $N(K)$ controla como o fator de divisão é alterado. A figura 18 exemplifica o modelo desta topologia.

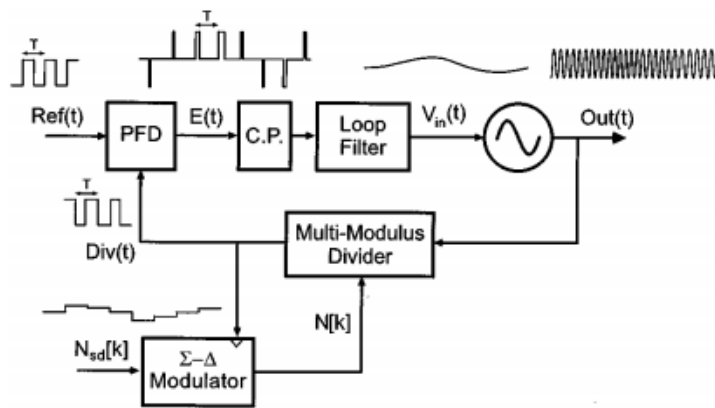


Figura 18 – Diagrama de blocos de um PLL com modulador *Sigma-Delta* ($\Sigma-\Delta$) [Perrott, Trott e Sodini (2002)].

Ao utilizar um divisor fracionário sinais espúrios também fracionários são gerados em consequência da alteração periódica do fator de divisão. Isso pode ser contornado ao se adotar um LPF de grau mais elevado, o que aumenta a complexidade e o custo do PLL. Normalmente o tratamento destes sinais espúrios é realizado no próprio modulador, apenas diminuindo seus efeitos. Embora a divisão por um fator fracionário possua grandes vantagens em comparação a topologia inteiro-N, algumas considerações devem ser avaliadas antes de definir qual delas usar.

5.4.3 Escolha do tipo de divisor

A topologia Inteiro-N foi escolhida com base na pesquisa bibliográfica realizada neste trabalho. Chegou-se a esta conclusão devido a grande aparição desta topologia em artigos científicos que tratavam sobre o projeto de um PLL para Zigbee. Outro fator importante é sua simplicidade de projeto e benefícios se comparada a topologia fracionária-N.

6 Divisor de Frequências

Nesta seção será discutido as arquiteturas básicas e a implementação de circuitos divisores que operam em altas frequências. Primeiramente é apresentado a estrutura do divisor, separando o mesmo em três blocos e caracterizando a função de cada um deles. Diante disso, é apresentado algumas topologias básicas de um *prescaler*, o projeto de portas lógicas e *flip-flops* utilizando a tecnologia CMOS em sistemas que trabalham em altas frequências. Por fim, será feito um estudo do ruído e distorção que são gerados pelo divisor.

6.1 Estrutura Geral

O divisor a ser projetado utiliza a técnica *pulse swallowing* e é composto de três partes: *prescaler* com dois módulos de divisão (N); *swallow counter* (S) e um contador programável (P). O *prescaler* pode dividir a frequência de entrada por dois valores, N ou $N+1$ dependendo do sinal MC. A saída do *prescaler* é dividida pelo módulo P do contador programável. Já o *swallow counter* divide a saída do *prescaler* por um fator S , seleciona um entre os 16 canais do ZigBee e controla o módulo da divisão do *prescaler* por meio do sinal de controle MC (figura 17).

Nas primeiras operações o *prescaler* divide o sinal de saída do VCO por um fator de $N + 1$. Este processo se repete até que a contagem do *swallow counter* tenha fim, o que habilita o sinal MC fazendo com que o módulo de divisão do *prescaler* seja alterado para N . Nesta primeira fase foram realizadas S divisões por $N + 1$, ou seja, se passaram $(N+1)S$ ciclos. Em seguida, o *prescaler* realiza divisões por um fator N até que o contador P tenha fim. Como se passaram S pulsos restam $P - S$ contagens e $N(P - S)$ ciclos a serem realizados antes que o contador programável e o *swallow counter* reiniciem, fazendo com que todo o processo seja repetido. Com esta lógica em mente, a cada $(N+1)S + N(P - S) = NP + S$ ciclos, o divisor de frequências gera um pulso em sua saída, ou seja, o fator de divisão é igual a $NP + S$. É de fundamental importância que o módulo de P seja maior que S , ou seja, *swallow counter* é reiniciado antes que o contador programável chegue ao seu valor máximo [Razavi (2012)]. O diagrama de blocos do divisor pode ser visualizado por meio da figura 17.

Como a faixa de operação do PLL varia entre 2400 MHz - 2475 MHz e possui 16 canais, então a frequência que separa dois canais é dada por $(2475 - 2400)/16$ que é aproximadamente igual a 5 MHz. Assim, neste trabalho o fator de divisão $NP + S$ varia entre 480 e 495. É possível verificar que fator de divisão mínimo é definido por NP e S seleciona os canais.

6.2 Prescaler com Dois Módulos de Divisão

O *prescaler* com dois módulos de divisão é o bloco que apresenta as maiores dificuldades de projeto num divisor de frequências. Trata-se basicamente da integração entre *flip-flops* e circuitos lógicos combinacionais de forma a se obter os valores de divisão especificados. Isto pode ser feito de forma assíncrona, caracterizada pela inclusão de *flip-flops* que invertem seu estado atual sempre que os mesmos forem habilitados pelo seu respectivo sinal de *clock* (figura 19 (a)). Ou de maneira síncrona no qual todos os elementos de memória são ativados de forma simultânea quando habilitados pelo *clock* (figura 19 (b)) [Lacaita, Levantino e Samori (2007)]. O divisor síncrono aqui apresentado é conhecido como contador Johnson.

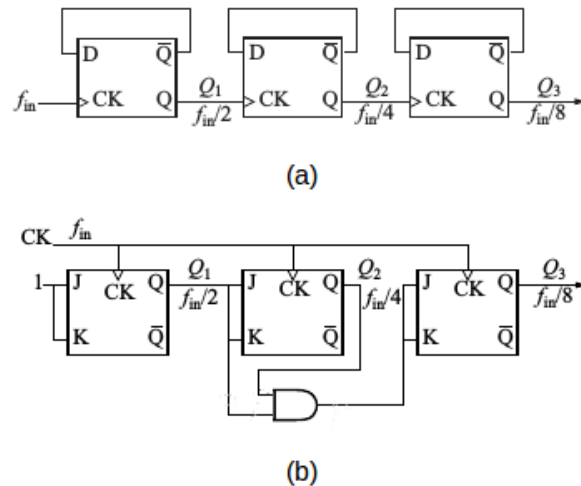


Figura 19 – Exemplos de divisores: (a) assíncrono e (b) síncrono [Lacaita, Levantino e Samori (2007)].

Contadores síncronos são rápidos e, devido a sua lógica combinacional para evitar estados indesejados, eles tornam possível o projeto de contadores com diferentes módulos de divisão. Em altas frequências, apresentam alta capacitância de carga em cada *flip-flop*, o que ocasiona o aumento do consumo de energia. Contadores assíncronos consomem menos energia, mas apresentam um atraso considerável entre a entrada do *clock* e a saída do divisor, o que ocasiona baixa velocidade de operação [Lacaita, Levantino e Samori (2007)]. Assim, mesmo com algumas desvantagens, os contadores síncronos são mais indicados para altas frequências. A divisão programável do *prescaler* é implementada utilizando-se uma parte síncrona e outra assíncrona de forma a se obter o melhor dos dois mundos, ou seja, baixo consumo e alta velocidade de operação em altas frequências.

Quando o fator de divisão do *prescaler* é citado, deve-se ter em mente que $N/N+1$ significa que este bloco realiza divisões por N ou por $N+1$ e não $\frac{N}{N+1}$. A figura 20 exemplifica o que foi dito acima por meio de um *prescaler* divisor por $15/16$, apresentando como parte síncrona o divisor por $3/4$ composto pelos *flip-flops* FF_1 e FF_2 e parte assíncrona

O *prescaler* divide o sinal de entrada por 15/16. Com isso, é possível projetar contadores que trabalham em baixas frequências. Estes contadores são o *swallow counter* (S) e o contador programável (P) que serão projetados utilizando a topologia TSPC, já que, neste caso, a velocidade de operação deixa de ser um fator limitante.

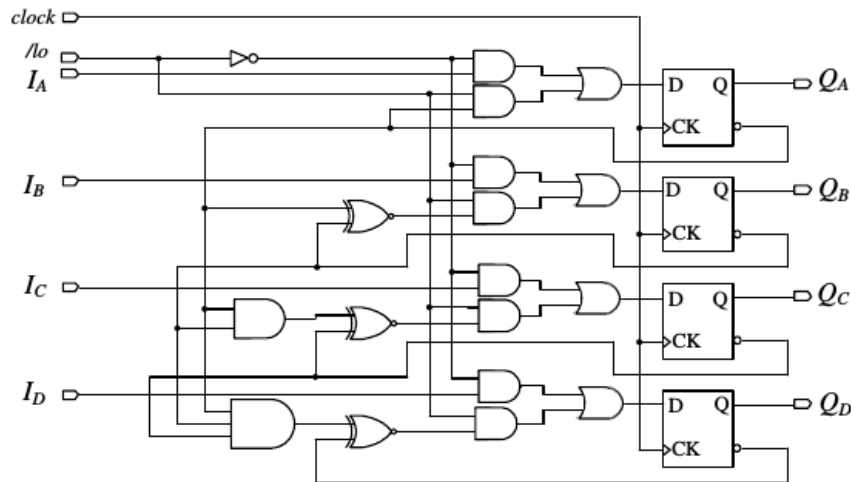


Figura 21 – Modelo proposto para o *swallow counter* [Adaptado de Santos (2008)].

6.4 Contador Principal (P)

Este divisor possui módulo de divisão igual a 32. Similarmente ao *prescaler*, seu fator de divisão apresenta uma relação linear com o consumo de corrente e como $N=15$, P tem que ser igual a 32 para que o fator de divisão total do PLL seja atingido, ou seja, de $N.P = 480$ até $N.P + S = 495$ (isso será explicado com mais detalhes no decorrer deste trabalho). Para se obter determinado valor, são necessários 5 *flip-flops* tipo D conectados em cascata como exemplificado por meio da figura 19 (a).

6.5 Projeto de Circuitos Lógicos para Altas Frequências

Os blocos do divisor podem ser projetados com diferentes técnicas, no qual cada uma apresenta vantagens e desvantagens que devem ser avaliadas com cautela. A escolha da topologia do divisor é administrada pelos seguintes fatores: excursão de entrada, capacitância de carga, velocidade máxima de processamento, excursão de saída, velocidade mínima para a correta operação dos circuitos e dissipação de potência. As técnicas *current-mode logic* (CML) e *True Single-Phase Clocking* (TSPC) são as mais utilizadas no projeto de divisores que trabalham em altas frequência.

6.5.1 Current-Mode Logic (CML)

Como a saída do VCO é diferencial e apresenta um sinal de alta frequência, o CML é o circuito mais adequado para o divisor por dois que é conectado na saída do VCO. Circuitos CML são baseados na arquitetura com entrada diferencial no qual uma corrente I_{SS} é chaveada pelo sinal de *clock* entre duas estruturas (figura 22) com diferentes modos de operação (modo sensitivo e modo *lach*) também habilitados pelo sinal de *clock*. Esta topologia apresenta alta velocidade de operação devido a facilidade com que a corrente pode variar entre os transistores deste par, sendo assim, utilizada em divisores que trabalham em altas frequências. A amplitude máxima da saída diferencial, normalmente projetada para o valor de 300 mV, é determinada com base na corrente I_{SS} e pela resistência de carga R_D [Razavi (2012)]. Outra vantagem do uso de entrada diferencial está no fato de evitar erros de modo comum devido as entradas e erros de variação na corrente e na referência de tensão (V_{DD}) [Lacaita, Levantino e Samori (2007)].

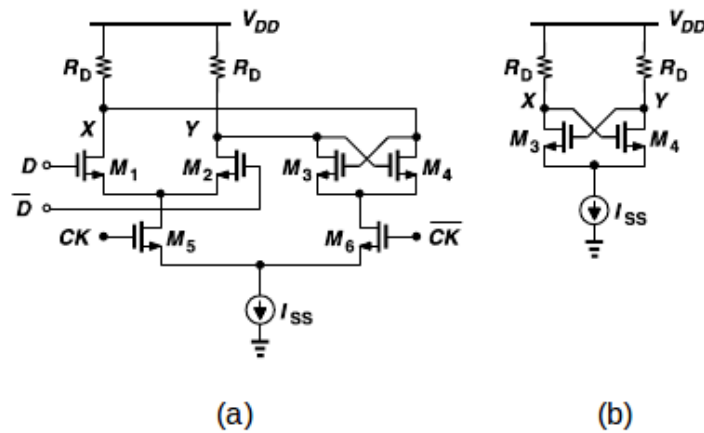


Figura 22 – (a) *Lach* D completo em CML; (b) *lach* D CML no modo regenerativo [Razavi (2012)].

Embora esta topologia apresente grandes vantagens, um parâmetro que deve ser avaliado com atenção é a razão de aspecto (W/L) dos transistores. Com a escolha correta é possível obter uma boa compensação entre velocidade e consumo de potência. Entretanto, ainda assim, o consumo de energia desta topologia pode ser considerado demasiado para algumas aplicações. Dessa forma, existe uma troca entre consumo de energia, velocidade de operação e razão de aspecto que normalmente é grande, o que exige uma maior área no *chip*.

Com estas propriedades, esta topologia é uma boa escolha quando se trabalha de circuitos operando em frequências de vários gigahertz. A característica de alta velocidade se torna mais evidente quando é analisado o funcionamento de um *lach* tipo D (figura 22). O circuito pode ser separado em parte regenerativa (transistores M3 e M4), par ativado pelo *clock* (M5 e M6) e par diferencial (M1 e M2). Quando o CK está no nível alto, M5 é ligado e os transistores M1 e M2 estão no modo sensitivo, no qual é possível “sentir” o

valor de D e amplificar a tensão diferencial de entrada, ou seja, X e Y segue o valor da entrada. Na transição para o modo regenerativo, ou modo *latch*, CK vai para o nível baixo, desligando $M1$ e $M2$, e tornando possível a operação de $M3$ e $M4$ (figura 22 (b)). Pode ser analisado como realimentação positiva e caso o ganho seja maior que uma unidade então o estado regenerativo permanece ativo até que CK mude e outro estado sensitivo tenha início.

A figura 23 mostra o exemplo de um divisor por dois utilizando dois *latches* mestre e escravo CML. Este bloco é composto por dois *latches* CML do tipo D conectados em realimentação negativa. Segundo [Razavi (2012)], o projeto tem início a partir dos parâmetros: máximo consumo de potência, excursão do sinal CK e capacitância de carga na saída. O seu projeto segue as etapas: calcular a corrente I_{SS} levando em conta o maior consumo de potência permitido; calcular R_D para que $R_D I_{SS} = 300mV$; selecionar $(W/L)_{1,2}$ de forma que o par diferencial mude completamente de estado para uma entrada de $300mV$; calcular $(W/L)_{3,4}$ para que o ganho da parte regenerativa seja maior que 1; calcular $(W/L)_{5,6}$ de forma que $M5$ e $M6$ da figura 23 suportem toda a corrente I_{SS} de acordo com a excursão de entrada do sinal CK .

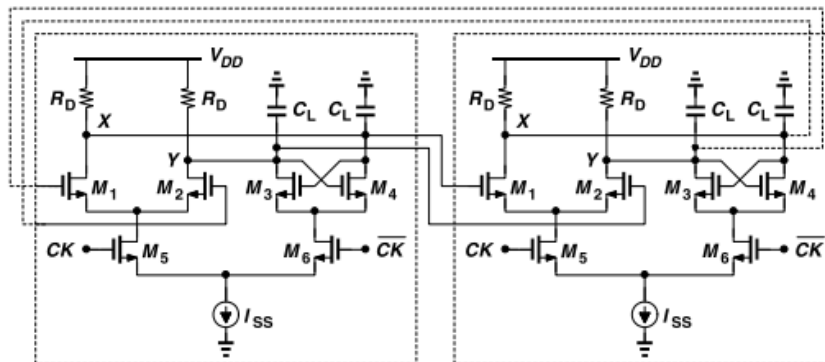


Figura 23 – Exemplo de um divisor por dois na topologia CML [Razavi (2012)].

6.5.2 Considerações de Projeto do Amplificador Diferencial

Um sinal diferencial é definido como um sinal medido entre dois pontos que apresentam sinais iguais, porém, opostos e variam em torno de um ponto fixo, ou seja, um modo comum. Um sinal não ideal sempre possui algum nível de ruído intrínseco associado a ele. Supondo que exista uma variação ΔV na referência de tensão V_{DD} , as saídas individuais do amplificador diferencial iriam sofrer uma alteração diretamente proporcional a ΔV . Porém a saída $V_{out1} - V_{out2}$ não é afetada pois o ruído possui mesmo valor em ambas, sendo assim, eliminado na diferença. Esta é uma propriedade importante no projeto do divisores . A figura 24 apresenta um par diferencial básico.

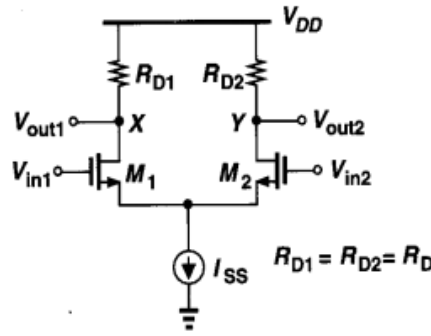


Figura 24 – Exemplo de um par diferencial básico [Razavi (2002)].

Onde I_{SS} é a corrente de controle e sempre se mantém constante. Segundo Razavi (2002), se V_{in1} é muito menor que V_{in2} , então M1 está desligado, fazendo com que toda a corrente I_{SS} passa pelo transistor M2 e como consequência disso: $V_{out1} = V_{DD}$ e $V_{out2} = V_{DD} - I_{SS}R_D$ (ponto (A) da figura 25). À medida que V_{in1} aumenta, V_{in2} diminui e uma pequena parcela de I_{SS} começa a passar pelo transistor M1 fazendo com que o mesmo ligue, o que diminui V_{out1} . Como a corrente I_{SS} é dividida entre M1 e M2, ou seja, $I_{SS} = I_{D1} + I_{D2}$, a corrente de M2 diminui enquanto I_{D1} aumenta fazendo com que V_{out2} se aproxime cada vez mais de V_{DD} . Quando $V_{out2} = V_{DD}$, M2 está desligado e toda a corrente I_{SS} passa pelo transistor M1, resultando em $V_{out2} = V_{DD}$ e $V_{out1} = V_{DD} - I_{SS}R_D$ (ponto (B) da figura 25). Este é o processo de troca de corrente entre o par diferencial.

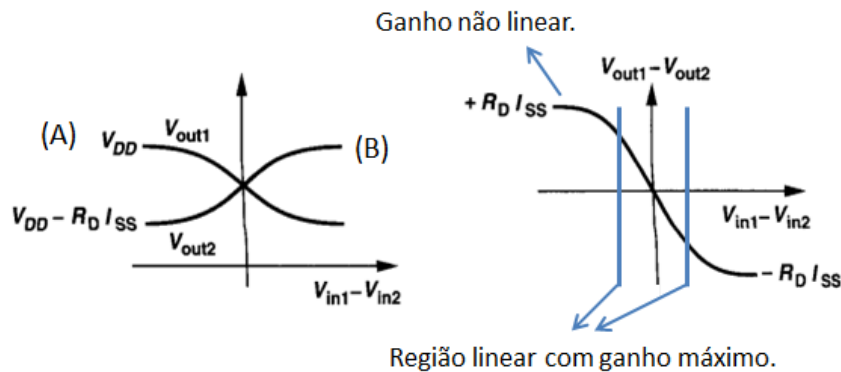


Figura 25 – Relação entre as entradas e saídas do par diferencial [Razavi (2002)].

Por meio da figura 25 é possível ver que máxima tensão de saída independe do nível de modo comum do sinal de entrada. O ganho máximo é obtido quando o circuito está em equilíbrio, ou seja, $V_{in1} = V_{in2}$.

A corrente que passa por um transistor NMOS em saturação, como é o caso de M1 e M2, é

$$I_{D1} = \frac{\mu_n C_{ox} W}{2L} (V_{GS} - V_{th})^2, \quad (6.1)$$

onde μ_n é a mobilidade das cargas no transistor, C_{ox} a capacitância entre o *gate* e o canal de elétrons e, por fim, W e L são as dimensões do *gate*. O objetivo é encontrar uma expressão que defina a corrente I_{D1} em função de I_{SS} e V_{in1} de forma que seja possível encontrar o valor máximo permitido para V_{in1} . Após vários passos,

$$I_{D1} = \frac{I_{SS}}{2} \left[1 \pm \sqrt{V_{in1}^2 \frac{\mu_n C_{ox} W}{I_{SS} L} - \frac{(\mu_n C_{ox})^2}{4I_{SS}^2} \left(\frac{W}{L}\right)^2 V_{in1}^4} \right]. \quad (6.2)$$

Quando V_{in1} for máximo, a equação dentro da raiz quadrada terá valor igual a 1 e I_{D1} será igual a I_{SS} ou zero. A equação 6.2 é derivada em relação a V_{in1} e igualada a zero, assim obtendo

$$V_{in1,max} = \sqrt{\frac{2I_{SS}}{\mu_n C_{ox} \left(\frac{W}{L}\right)}}. \quad (6.3)$$

Uma das vantagens do par diferencial é que a excursão de entrada, que é relacionada com a equação 6.3, é menor se comparada com circuitos TSPC, sendo assim mais fácil de satisfazer as necessidades de um sinal em altas frequências. W/L tem que aumentar para diminuir o valor da tensão necessária para que ocorra uma mudança de estado no circuito CML. Porém, é preciso ter um aumento significativo em W/L para que essa tensão diminua, tendo como consequência transistores maiores o que limita a velocidade de operação do circuito. Essa relação é de grande importância e deve ser levada em consideração no projeto.

6.5.3 True Single-Phase Clocking (TSPC)

Esta é uma lógica bastante usada no projeto de divisores que operam em baixas frequências ou que giram em torno de alguns *gigahertz*. Trata-se de uma topologia que adota uma lógica dinâmica no qual o sinal de *clock* é usado no pré-carregamento da saída para um estado desejado. Dessa forma, quando CK está em nível baixo (estado de pré-carregamento) o estado atual é pré-carregado para V_{DD} e quando CK está em nível alto (estado sensitivo) a saída apresenta um nível baixo ou alto caso o sinal de entrada seja '1' ou '0', respectivamente. Divisores que utilizam TSPC apresentam baixo consumo de energia se comparados com CML, mas são limitados para frequências não muito altas (alguns gigahertz) e exigem uma excursão *rail-to-rail* do *clock* para o correto funcionamento.

Se a quantidade de transistores empilhados em cada estágio da figura 26 (a) forem reduzidos, ou seja, com apenas dois transistores em cada estágio, temos a topologia E-TSPC (*Extended True Single-Phase Clocking*) que aumenta a velocidade de operação devido a retirada destes transistores. Isso é feito de forma a evitar efeito de corpo e

diminuir a capacitância de carga de cada estágio, o que diminui a propagação de atrasos, aumentando assim a velocidade de operação.

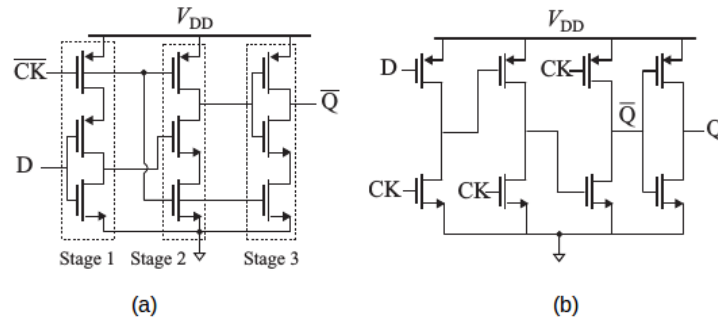


Figura 26 – *Flip-flops* tipo D dinâmico: (a) TSPC, (b) E-TSPC [Lacaita, Levantino e Samori (2007)].

A figura 27 apresenta o exemplo de um *prescaler* divisor por 3/4 na topologia TSPC que utiliza portas NOR para realizar a divisão por 3 e selecionar o fator de divisão. Construir *flip-flops* e portas lógicas em uma única célula tem como objetivo reduzir os atrasos que surgem devido a inclusão dessas portas, o que resulta em menos consumo de energia e aumento da velocidade de operação em comparação caso ambos fossem projetados separadamente [Lacaita, Levantino e Samori (2007)].

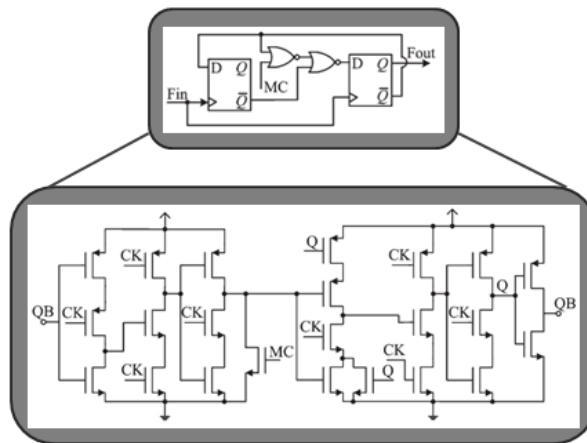


Figura 27 – Exemplo de um *prescaler* divisor por 3/4 [Chen e Jung (2011)].

6.5.4 Considerações de Projeto do TSPC

Segundo Deng e Niknejad (2010), máxima frequência de operação, equação (3.4), está diretamente relacionada com a propagação do atraso em cada mudança de estágio de um divisor por dois, como exemplificado na tabela 2. A propagação do atraso em cada estágio é apresentada como t_1, t_2, t_3, t_4 onde

$$f_{IN} = \frac{1}{2 \cdot \max(t_1 + t_2 + t_3 + t_4)}. \quad (6.4)$$

Tabela 2 – Tabela da verdade de um divisor por dois utilizando um *flip-flop* D.

Estágio	IN	$OUT(\bar{Q})$	Atraso
1	0	1 (Hold)	t_1
2	1	$1 \rightarrow 0$	t_2
3	0	0 (Hold)	t_3
4	1	$0 \rightarrow 1$	$t_4 = t_1 + t_2 + t_3$

Em todos os divisores existentes, o consumo de potência está diretamente relacionado com a frequência de operação e as capacitâncias de cada estágio do divisor. Assim, segundo Weste e Harris (2005), o consumo de energia está diretamente relacionado com a troca de estágios por

$$P_{switch} = C_L V_{DD}^2 f_{IN}, \quad (6.5)$$

onde C_L representa a soma das capacitâncias de todos os nós, V_{DD} é a tensão de alimentação e f_{IN} a frequência do sinal a ser dividido.

Existe um tempo limite no qual os transistores NMOS e PMOS possam ligar e desligar de forma que a operação do *flip-flop* seja realizada da maneira correta. Idealmente, apenas um dos transistores estão ligados no mesmo instante de tempo, mas existe um pequeno período de tempo no qual ambos estão ligados e uma corrente de fuga I_{short} irá encontrar um caminho do V_{DD} até o terra. Isso aumenta o consumo de energia, bem como a tempo de transição dos estados. A potência consumida por este efeito é

$$P_{short} = I_{short} V_{DD}. \quad (6.6)$$

O E-TSPC é mais rápido se comparado com o TSPC convencional e possui potência P_{switch} menor que o TSPC. Porém, segundo Guo et al. (2010), ao comparar o E-TSPC com TSPC, o consumo de potência devido a P_{short} é muito maior que P_{switch} . Assim, para algumas frequências de operação e dependendo valores de W/L , que influenciam na corrente de fuga, o TSPC apresenta características mais interessantes que o E-TSPC.

6.5.5 Escolha da Topologia dos Circuitos Lógicos

Como o VCO possui saída diferencial e trabalha com o dobro da frequência desejada para o transceptor ZigBee, será necessário incluir um divisor por dois no projeto do PLL. Este divisor é baseado na topologia CML que proporciona boa velocidade de operação com baixo consumo em sistemas que operam em altas frequências.

Para fazer a conversão do sinal diferencial do divisor por dois para a entrada *rail-to-rail* do TSPC é usado o circuito da figura 28. Este circuito utiliza um amplificador

com um estágio e um inversor para amplificar a saída *single-ended* do primeiro estágio, habilitando assim um sinal *rail-to-rail*.

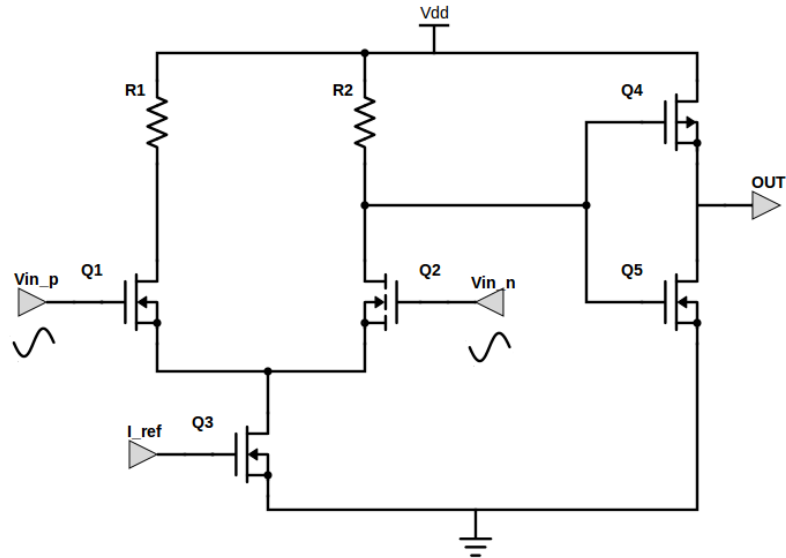


Figura 28 – Exemplo de um conversor de entrada diferencial para *single-ended* (CML para TSPC) [Perrott (2003)].

Embora circuitos CML operem em altas velocidades, sendo mais rápidos que o TSPC, *prescalers* projetados nesta topologia apresentam maior consumo de energia. Sendo assim, apenas *prescalers* que operam em frequências elevadas, normalmente acima de 20 GHz, utilizam circuitos CML pois outras topologias não operam de forma correta nessa faixa de frequência. *Prescalers* projetados em TSPC consomem menos energia ao custo de uma baixa velocidade de operação [Chen e Jung (2011)], uma escolha importante tendo em vista que o consumo do divisor é um fator crítico no projeto do PLL. Como o *swallow counter* e o contador programável operam em baixas frequências, o TSPC é a melhor escolha neste caso, pois a velocidade de operação deixa de ser um fator limitante e o consumo de energia é a principal variável em questão.

Com os argumentos acima e devido a grande maioria dos artigos pesquisados apresentarem *prescalers*, *swallow counter* e o contador principal projetados na topologia TSPC, o mesmo será feito neste trabalho. A tabela 3 apresenta os dados de projeto de cada bloco e a figura 29 apresenta a estrutura geral do divisor.

Tabela 3 – Especificações da frequência de operação e fatores de divisão dos divisores.

Divisores	Fator de Divisão	Topologia	Frequência de Operação
<i>Prescaler</i> (N/N+1)	15/16	TSPC (Fig. 20)	2400 - 2475 MHz
<i>Swallow Counter</i> (S)	16	TSPC (Fig. 21)	Depende do canal
Contador Principal (P)	32	TSPC	165 MHz
Divisor do PLL	480-495	TSPC	2400 - 2475 MHz

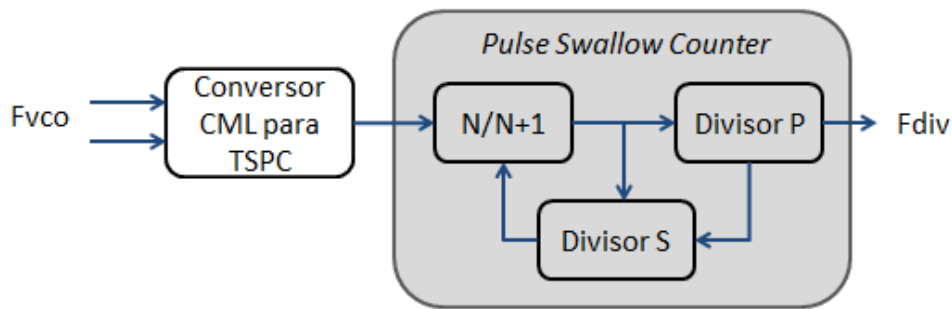


Figura 29 – Estrutura geral do divisor.

6.6 Análise do Ruído de Fase e *Jitter* no Divisor de Frequências

O projeto de divisores na faixa de Rádio Frequência (RF) é bastante crítico devido o compromisso entre a alta frequência de operação, redução do consumo energético e baixo ruído de fase. A influência do ruído de fase do divisor não se manifesta de forma pronunciante no PLL se comparado com a contribuição do PFD e do sinal de referência, que vem do oscilador a cristal [Bistue, Quemada e Adin (2009)]. Entretanto, em alguns casos essa influência pode afetar o PLL de forma mais significativa. O ruído de fase do divisor (θ_D) é somado com o ruído de fase do oscilador a cristal (θ_R). Porém, assim como o sinal do VCO é multiplicado pelo fator de divisão N , o mesmo ocorre com o ruído de fase do divisor do PLL. Para que a contribuição do divisor no ruído total seja mínima, ou seja, possa ser desconsiderada, então: $\theta_D \ll \theta_R$.

A inclusão indesejada de ruído no divisor afeta o ganho banda do PLL, ou seja, o ponto no qual o gráfico de ganho toca o eixo das frequências, o que altera a margem de fase e, como consequência, a estabilidade do sistema.

Dentro do divisor, o *prescaler* é o bloco mais crítico, pois opera em altas frequências e, juntamente com o VCO, apresenta o maior valor de ruído do PLL. A maneira mais fácil e simples de se avaliar o espectro do ruído de fase do divisor é por meio do cálculo do *jitter*. *Jitter* ($\sigma_{t_0}^2$) é a variação na periodicidade de um sinal, ou seja, uma variação estatística no recebimento de um sinal no instante t_0 . Esta variação ocorre devido: sinal de interferência em baixas frequências e diminuição no tempo de troca de nível lógico causado por flutuações na temperatura e outros fatores externos [Lacaita, Levantino e Samori (2007)].

7 Simulações e Análise dos Resultados

Este capítulo tem como objetivo apresentar os resultados das simulações em nível de transistor dos circuitos utilizados para a construção do divisor de frequências. Todos os blocos que serão descritos foram projetados utilizando a tecnologia TSMC 0,18 μm . Embora a frequência de operação do divisor completo varie de 2,40 GHz até 2,475 GHz, todas as simulações foram realizadas a 2,80 GHz, a não ser quando especificado o contrário, com o objetivo de oferecer uma margem de segurança relacionada a frequência de operação dos circuitos.

7.1 Circuitos Utilizados na Lógica de Controle

Neste tópico será abordado um pouco sobre o funcionamento e projeto das portas lógicas e dos *flip-flops* utilizados no projeto do divisor completo (*Pulse Swallow Divider*).

7.1.1 Portas Lógicas e Multiplexador de 2 bits

Ao todo foram utilizadas seis portas lógicas diferentes na lógica de controle do *prescaler*, do *swallow counter* e no divisor completo. Todas as portas lógicas foram projetadas utilizando a topologia dinâmica e a metodologia de um inversor, que é compensar a diferença da mobilidade dos elétrons dos transistores NMOS e PMOS com o balanceamento apropriado dos valores de W e L de ambos os transistores, ou seja, $(W/L)_P = 2(W/L)_N$. Como os valores de L foram fixados em 0,18 μm para reduzir as capacitâncias parasitas, $W_N = 1\mu m$ e $W_P = 2\mu m$. As figuras no apêndice apresentam o esquemático de todas as portas lógicas utilizadas, sendo elas: AND com duas entradas; NAND com 3 entradas; uma inversora; OR com duas e três entradas e, por fim, XNOR com duas entradas.

7.1.2 *Flip-flops* TSPC e E-TSPC

Os *flip-flops* tipo D (DFF) utilizados neste trabalho foram o TSPC primeiramente proposto por Yuan e Svensson (1989) e o E-TSPC proposto em Jr e Noije (1999). Ambos possuem vantagens e desvantagens que devem ser analisadas com cuidado ao utilizar uma dessas topologias, como por exemplo, consumo de energia, velocidade de operação, máxima frequência de operação, entre outros. Abaixo será discutido acerca dessas características e como foi realizado o projeto dos *flip-flops*.

7.1.2.1 Topologia TSPC

Como dito anteriormente, velocidade de operação e consumo são fatores chave no projeto de um *flip-flop* para altas frequências. Dessa forma, objetivando a redução da capacitância parasita, tanto nas portas lógicas citadas acima quanto nos dois *flip-flops* projetados utilizando transistores PMOS e NMOS, os valores de L foram fixados em $L = 0,18\mu\text{m}$ e os valores de W foram ajustados para se obter a melhor performance com a limitação imposta ao fixar o valor de L .

O TSPC é composto por dois modos ou estados de operação, o pré-carregamento e o sensitivo. Na fase de pré-carregamento $\text{CLK}=0$, tendo como consequências: o pré-carregamento de A1 para um valor que depende da entrada; pré-carregamento de A2 (figura 30) para V_{DD} e, por fim, $Q=1$. No modo sensitivo $\text{CLK}=1$ e as saídas Q e Q_+ irá depender do nível lógico que o nó A1 possuía na fase de pré-carregamento. Embora testes tenham sido feitos com o E-TSPC, é importante lembrar que foram utilizados *flip-flops* TSPC em todos os divisores deste trabalho.

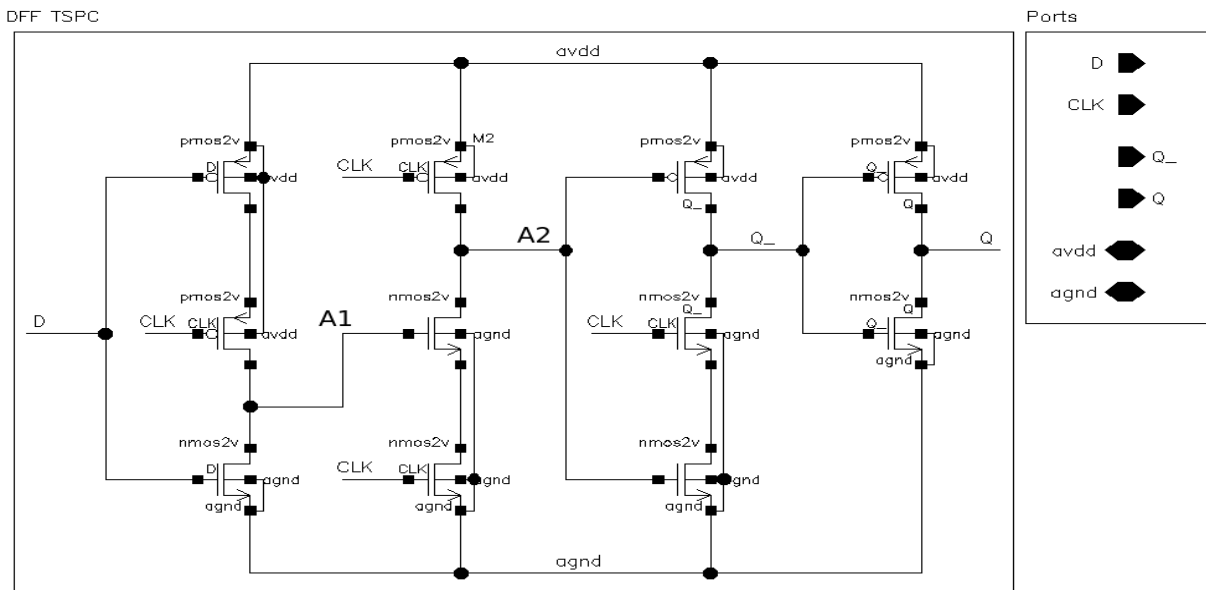


Figura 30 – Esquemático do *flip-flop* na topologia TSPC.

Ao projetar este *flip-flop*, o valor de $W_P = 2\mu\text{m}$ e $W_N = 1\mu\text{m}$ para os transistores NMOS e PMOS, respectivamente, foi o que apresentou melhor performance e atendeu melhor aos requisitos de velocidade, bem como uma melhor precisão na divisão por dois. Por meio da figura 31 é possível visualizar o *test bench* utilizado para fazer as simulações e verificar seu correto funcionamento. O primeiro *DFF* está configurado como um *flip-flop* D comum, já o segundo como um divisor por dois.

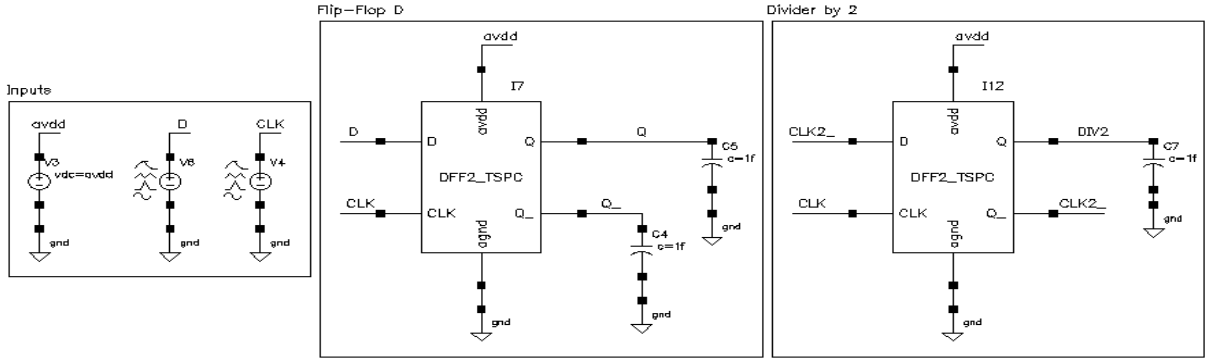


Figura 31 – Test bench do *flip-flop* na topologia TSPC.

Para calcular o fator de divisão foi necessário calcular o período do sinal de saída e dividi-lo pelo período do sinal de entrada. Este procedimento foi repetido para todos os divisores que compõem este trabalho utilizando a função *cross* da calculadora do Cadence, que permite selecionar dois instantes de tempo do gráfico de uma função e medir a diferença entre eles. Dessa forma, calculou-se a distância, no tempo, entre duas bordas de subida consecutivas tanto da onda de entrada quanto da onda de saída. O cálculo do fator de divisão é representado por

$$FD = \frac{\left[cross \left(F_{outN}, \frac{V_{DD}}{2}, 2, rising \right) - cross \left(F_{outN}, \frac{V_{DD}}{2}, 1, rising \right) \right]}{\left[cross \left(F_{inN}, \frac{V_{DD}}{2}, 2, rising \right) - cross \left(F_{inN}, \frac{V_{DD}}{2}, 1, rising \right) \right]}, \quad (7.1)$$

onde FD representa o verdadeiro Fator de Divisão de um bloco, o primeiro parâmetro da função *cross* representa o sinal de referência, o segundo representa o ponto no eixo Y no qual se quer medir o instante no tempo, o terceiro indica o ciclo do sinal de referência, com contagem da esquerda para a direita e, por fim, o quarto representa a borda de subida ou descida da onda.

A tensão de alimentação é de 1,8 V, a frequência de CLK é igual a 2,8 GHz e a frequência de D é de 2,1 GHz. Os resultados de simulação com estes valores de tensão e frequência de operação são apresentados por meio da figura 32. Nota-se um atraso da onda de saída em relação a entrada, o que é perfeitamente comum neste tipo de circuito e não fugiu aos resultados esperados.

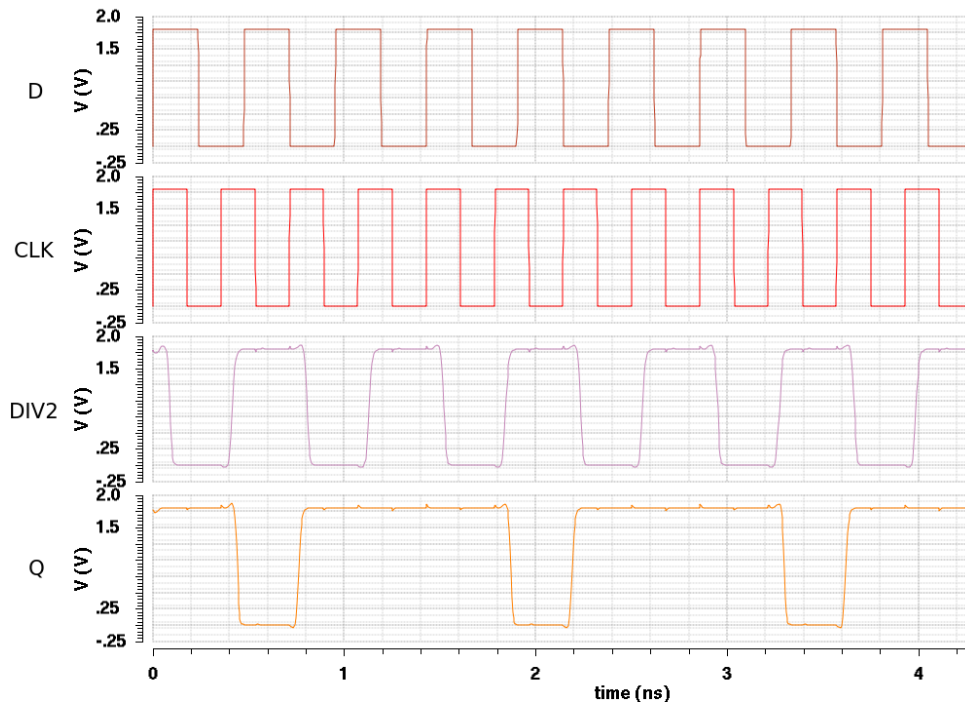


Figura 32 – Resultados de simulação do *flip-flop* D na topologia TSPC.

Para a verificação do consumo de corrente foram feitas simulações do consumo de corrente variando no tempo. O gráfico da corrente não apresentava um valor constante e sim um sinal oscilante. Para determinar a corrente média, utilizou-se a função *average* da calculadora do Cadence, sendo assim, todas as correntes calculadas neste trabalho foram calculadas com o procedimento supracitado. Esta topologia apresenta consumo médio de $200,81\mu W$ quando opera em 2,80 GHz e $172,148\mu W$ quando opera em 2,40 GHz (figura 33). A frequência máxima de operação de é 7 GHz. A tabela 4 apresenta a pinagem do DFF TSPC.

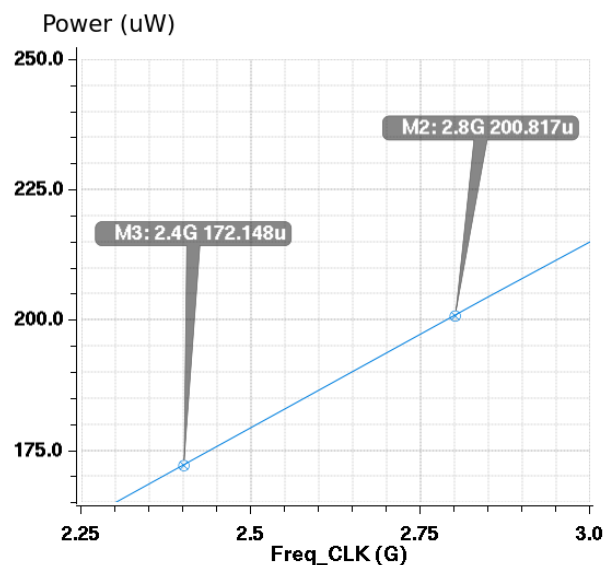


Figura 33 – Consumo de energia do *flip-flop* D na topologia TSPC.

A tabela 4 apresenta os pinos de entrada e saída do *flip-flop* TSPC.

Tabela 4 – Pinos do DFF na topologia TSPC.

Nome	Descrição	Tipo do Pino
D	Entrada do sinal	Entrada
CLK	<i>Clock</i>	Entrada
Q	Saída do <i>flip-flop</i>	Saída
Q ₋	Saída Invertida	Saída
avdd	Referência de Tensão	Entrada
agnd	Terra	Entrada

7.1.2.2 Topologia E-TSPC

Esta topologia possui certa semelhança com a topologia TSPC. Na realidade, a E-TSPC é uma adaptação da TSPC com o objetivo de aumentar a frequência de operação, possuindo o custo de aumentar o consumo de energia do *flip-flop*. Os mesmos princípios utilizados para o projeto na topologia TSPC se aplicam para o E-TSPC também, ou seja, $W_P = 2\mu m$ e $W_N = 1\mu m$. Sendo assim, as figuras abaixo mostram o esquemático do circuito e o resultado de simulação, onde: figura 34 apresenta o esquemático do *flip-flop*; a figura 35 mostra o resultado de simulação e a figura 36 caracteriza o gráfico de consumo pela frequência de operação.

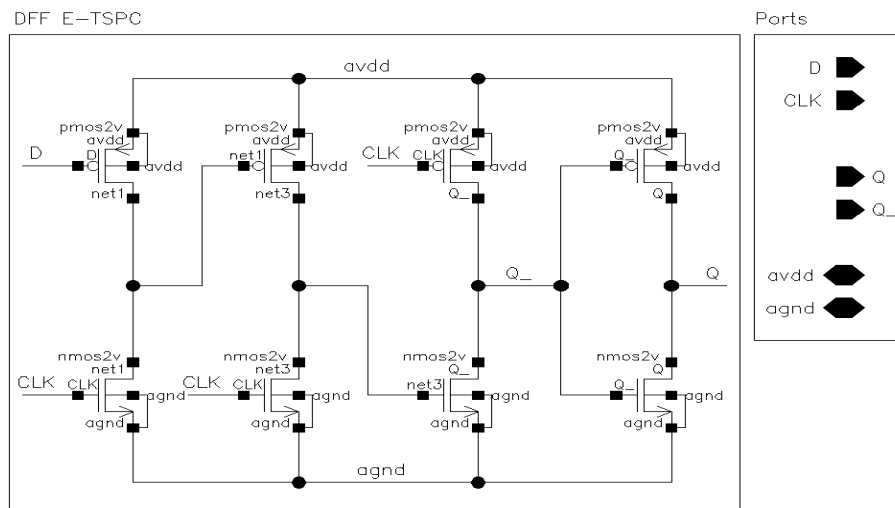


Figura 34 – Esquemático do *flip-flop* D na topologia E-TSPC.

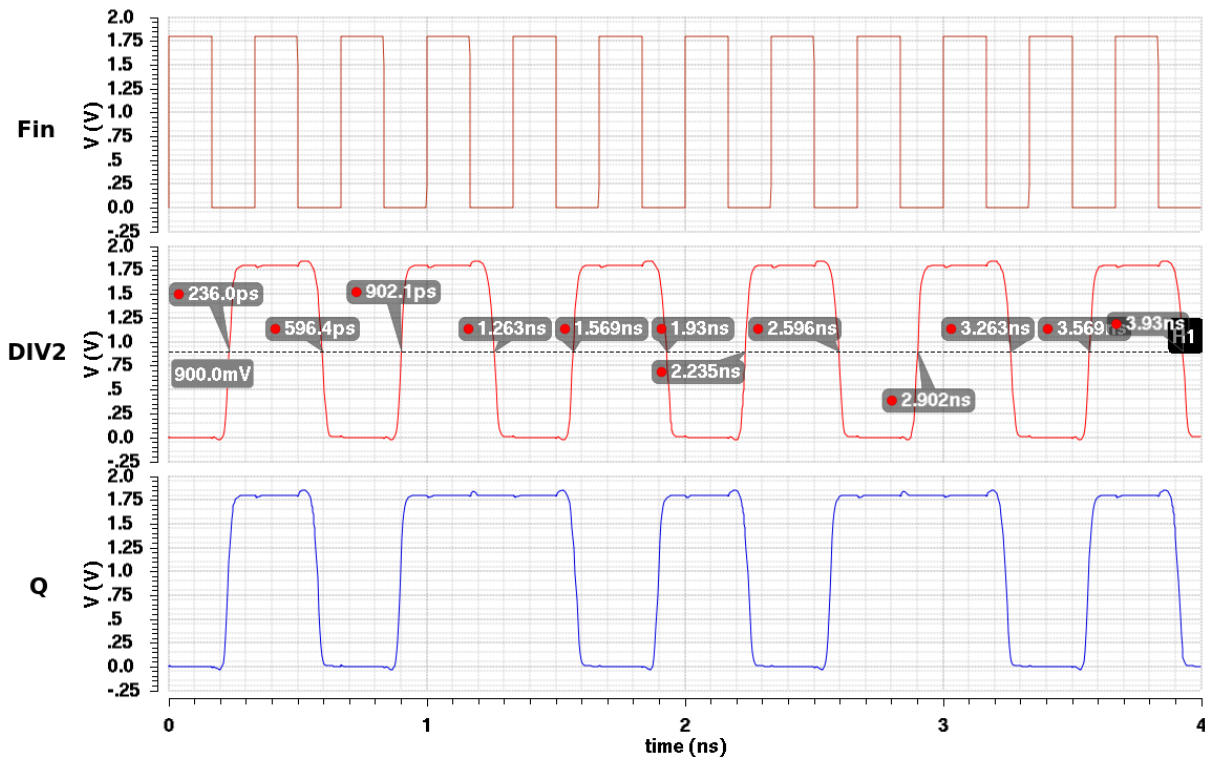


Figura 35 – Resultados de simulação do *flip-flop* D na topologia E-TSPC.

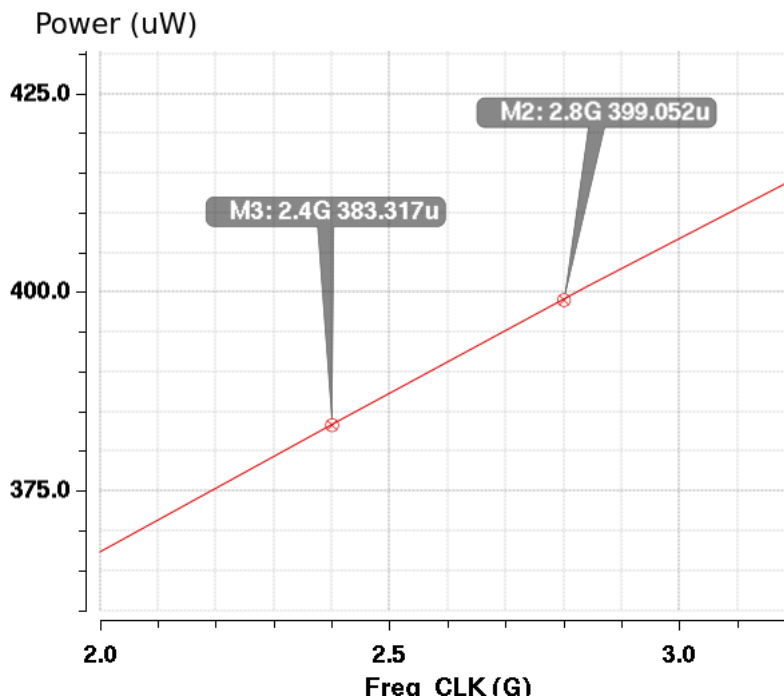


Figura 36 – Consumo de energia do *flip-flop* D na topologia E-TSPC.

Deve-se ter cuidado no momento de escolher entre qual das duas topologias usar, pois embora o E-TSPC funcione em altas frequências, ele consome muito mais corrente que o TSPC.

A tabela 5 apresenta os pinos de entrada e saída do *flip-flop* E-TSPC.

Tabela 5 – Pinos do DFF na topologia E-TSPC.

Nome	Descrição	Tipo de Pino
D	Entrada do dinal	Entrada
CLK	<i>Clock</i>	Entrada
Q	Saída do <i>flip-flop</i>	Saída
Q_	Saída Invertida	Saída
avdd	Referência de Tensão	Entrada
agnd	Terra	Entrada

7.2 Prescaler (N/N+1) Divisor 15/16

Segundo [Lacaita, Levantino e Samori \(2007\)](#), o primeiro passo para se projetar um divisor para o PLL é escolher o fator N e com base nele escolher o fator dos divisores P e S. O circuito proposto possui um fator de divisão 15/16 e é composto por uma parte síncrona representada por um *prescaler* com fator de divisão igual a 3/4 e uma parte assíncrona que possui dois *flip-flops* D configurados como divisor por 2 conectados em cascata, obtendo assim uma divisão por 4 (figura 37).

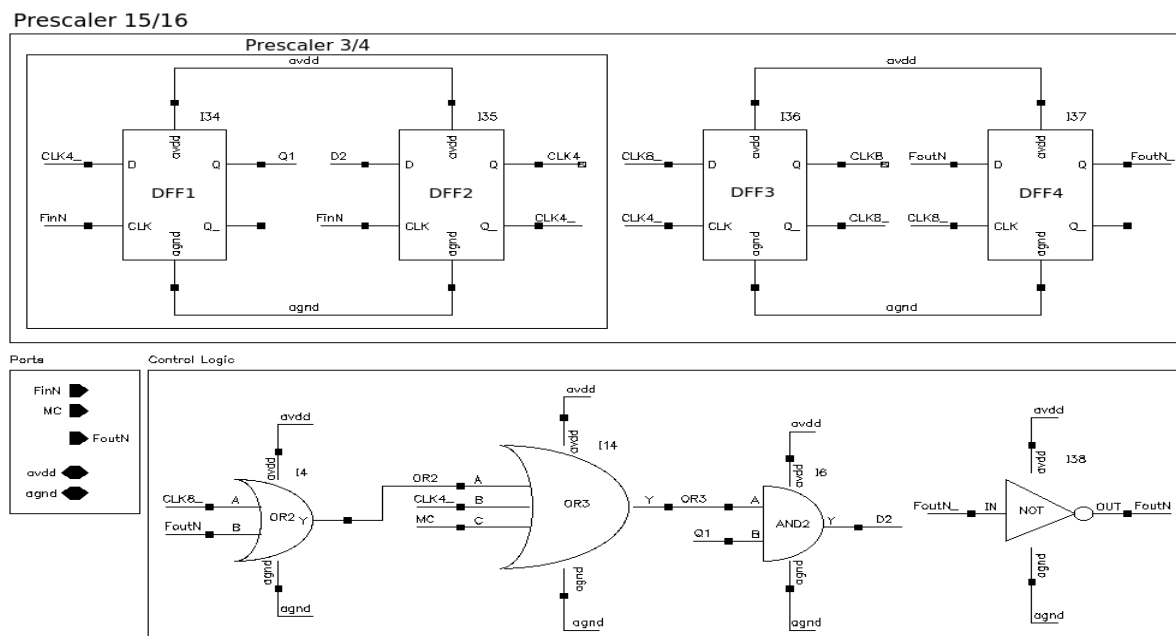


Figura 37 – Esquemático do *prescaler* com fator de divisão 15/16.

O divisor síncrono utilizado possui um fator de divisão 3/4 e o fator de divisão é definido pelas portas lógicas AND, OR e pelo sinal de entrada MC. Se MC é igual ao

nível lógico 1, a porta U3 da figura 38 também possuirá saída igual a 1, permitindo que a saída da porta U4 seja igual a sua entrada. Isso equivale a conectar a saída Q do DFF1 diretamente na entrada D do DFF2, permitindo que sistema realize uma divisão por 4 da frequência de entrada. Quando MC está em nível lógico baixo, a saída da porta U3 será igual a sua entrada e o circuito é reduzido igual ao apresentado na figura 39.

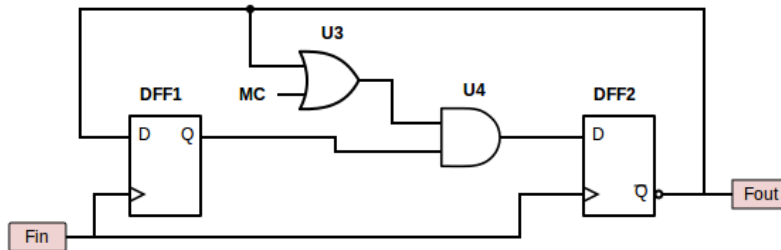


Figura 38 – Prescaler com fator de divisão 3/4.

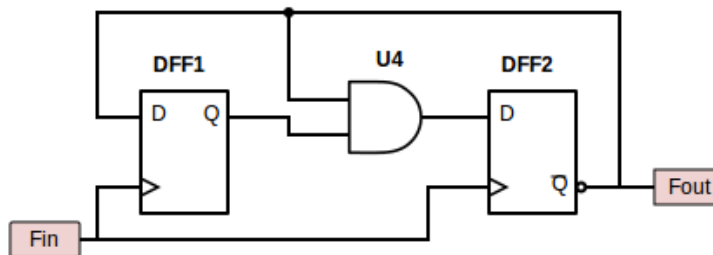


Figura 39 – Esquemático do divisor por 3.

Fazendo uma pequena análise do divisor por 3, suponha que o estado inicial das saídas seja $Q_1\overline{Q_2} = 00$. Após um pulso positivo de *clock*, $Q_1 = \overline{Q_2} = 0$ e $\overline{Q_2} = \overline{U4}$ pois $Q_1 = 1$. Como exemplificado na tabela 6, os próximos 3 estados de $Q_1 = \overline{Q_2}$ serão 10, 11 e 01. O estado 00 não irá se repetir devido a inclusão da porta U4 porque seria necessário que o estado anterior de $\overline{Q_2}$ e de U4 fossem, respectivamente, iguais a zero, ou seja, entrada e saída de um porta AND serem iguais a zero. Assim, esta implementação realiza uma divisão por 3 [Razavi (2012)].

Tabela 6 – Tabela da verdade de um divisor por dois utilizando um *flip-flop* D.

State (N)	Q_{1N} IN	$\overline{Q_{2N}}$ IN	U4 (X_N)Out	Q_{1N} OUT	$\overline{Q_{2N}}$ OUT
0	0	0	0	1	0
1	$\overline{Q_{20}} = 0$	$X_0 = 1$	0	1	1
2	$\overline{Q_{21}} = 1$	$X_1 = 1$	1	0	1
3	$\overline{Q_{22}} = 1$	$\overline{Q_{23}} = 0$	1	1	0
4	$\overline{Q_{23}} = 0$	$\overline{Q_{24}} = 1$	0	0	1

Com a adição da parte assíncrona, composta por dois *flip-flops* D (DFF3 e DFF4 da figura 37) obtendo uma divisão por 4, é possível obter um fator de divisão igual a 16 quando $MC=1$. Para incluir a divisão por 15 quando $MC=0$ é necessário manter a divisão por 3 do circuito da figura 38. Isso é feito incluindo uma porta OR que conecta as saídas dos *flip-flops* DFF3 e DFF4. A porta OR3 (figura 37) possui saída igual a 0 quando $F_{outN} = CLK8_ = CLK4_ = 0$, por um ciclo de *clock*, que pode ser visualizado nas figuras 40 e 42, forçando uma divisão por 3 no *prescaler* 3/4 e tornando possível a divisão por 15 do sistema completo.

Em resumo, quando $MC=1$ o *prescaler* divide a entrada por 16 e quando $MC=0$ o *prescaler* divide por 15. A figura 40 apresenta o resultado da simulação quando $MC=0$, a figura 41 apresenta o resultados da simulação quando $MC=1$ e, por fim, a figura 42 possui o resultado da simulação variando o valor de MC . Todas as figuras apresentam os instantes de tempo, no meio da transição do nível lógico, ou seja, em 0,9 V.

Os sinais $CLK8_$, OR2, $CLK4_ CLK4$, OR3, Q1 e D2 são sinais internos e correspondem aos sinais visualizados no esquemático do circuito da figura 37. Estes sinais foram ordenados nesta sequência com o objetivo de facilitar o entendimento do *prescaler* e saber como cada sinal muda no decorrer do tempo analisando cada porta lógica e os *flip-flops* mais críticos do circuito.

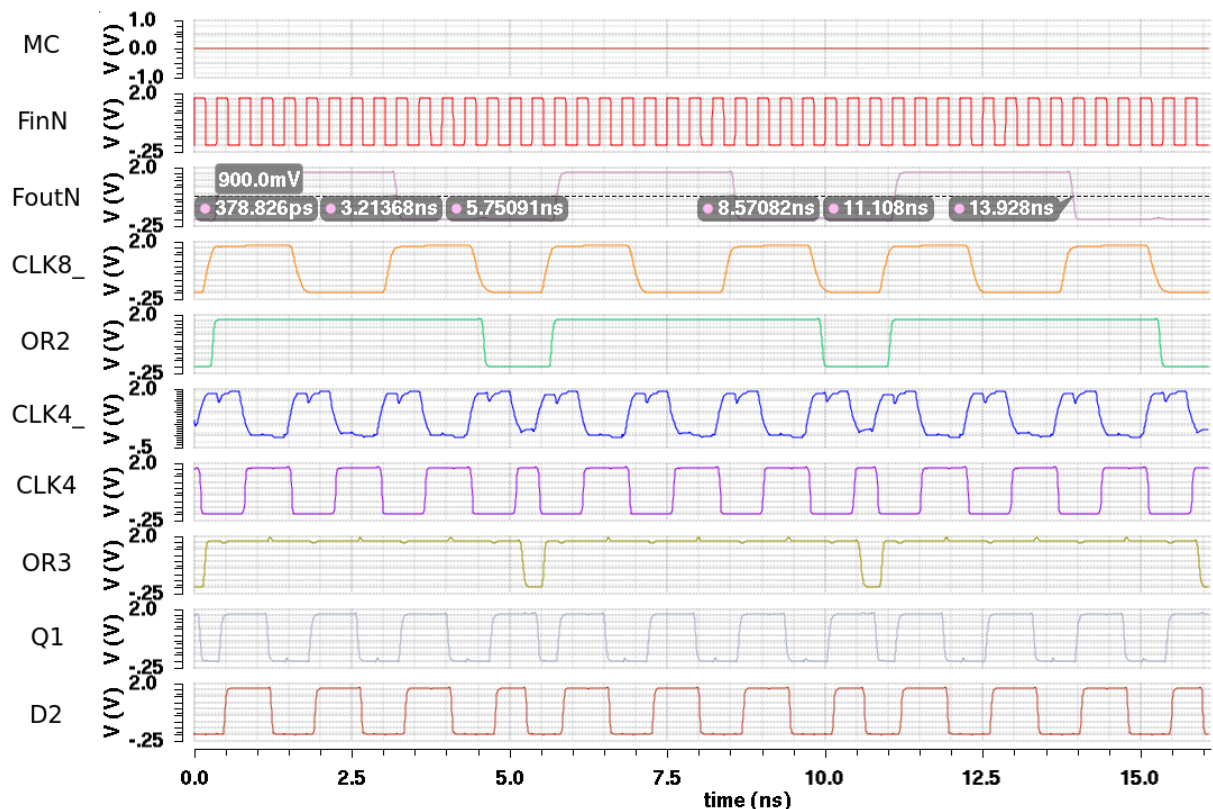


Figura 40 – Simulação do *prescaler* com fator de divisão 15/16 quando $MC=0$.

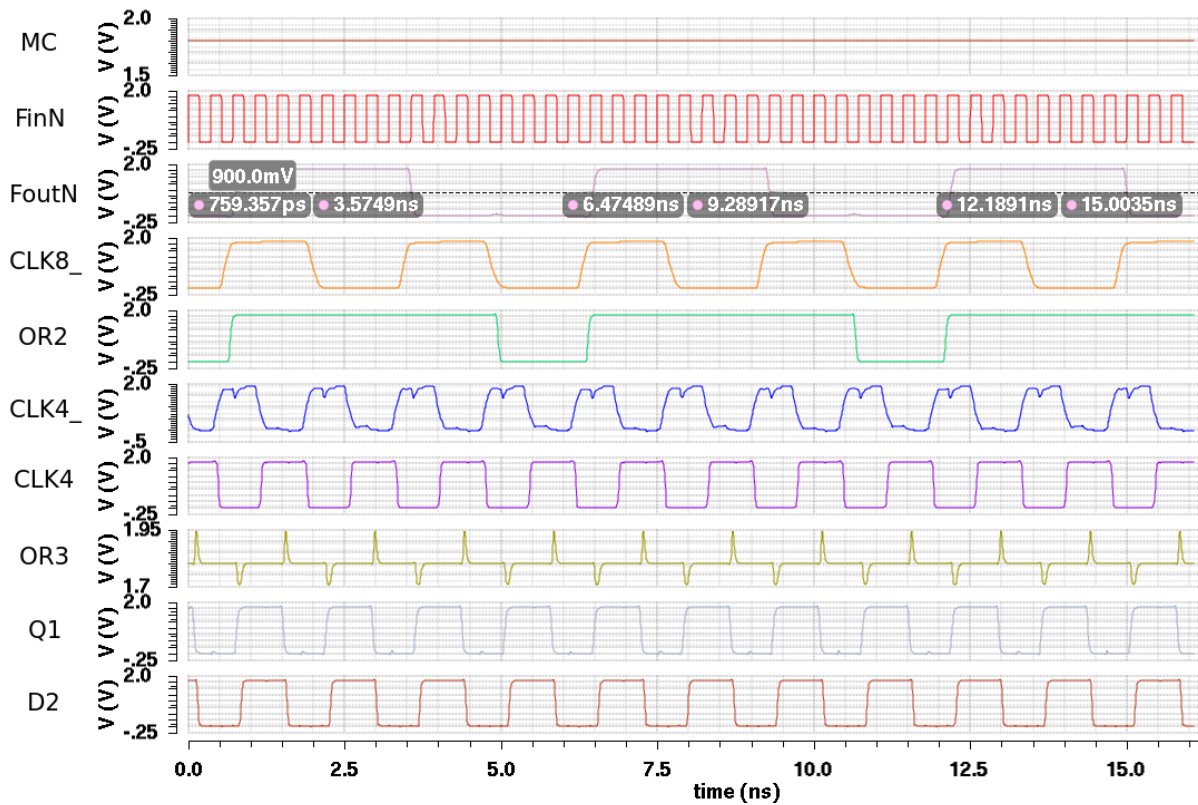


Figura 41 – Simulação do *prescaler* com fator de divisão 15/16 quando MC=1.

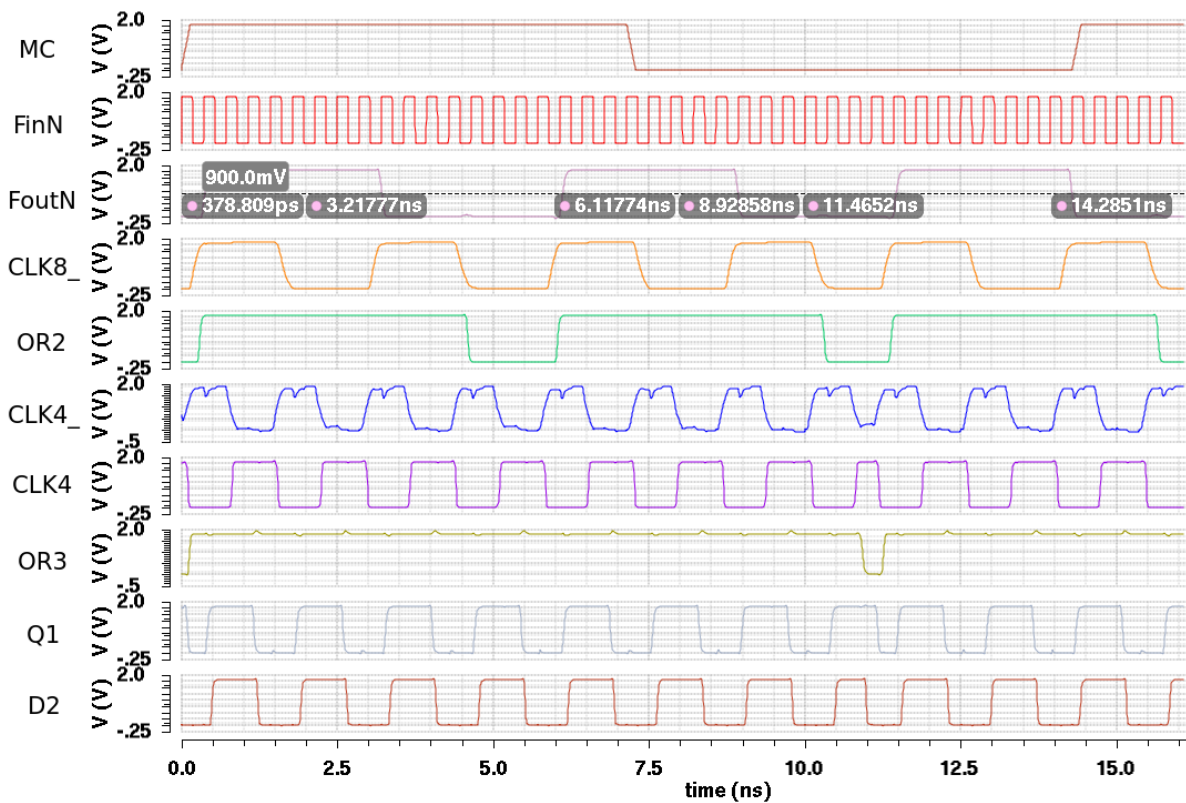


Figura 42 – Simulação do *prescaler* com fator de divisão 15/16 variando o valor de MC.

Tabela 7 – Comparação entre *prescalers* por meio da FOM.

	Tang et al. (2001)	Larsson (1996)	Liu et al. (2011)	Este Trabalho
Fator de Divisão	15/16	8/9	15/16	15/16
Tecnologia CMOS	0,5 μm	0,8 μm	0,18 μm	0,18 μm
Tensão de Alimentação	3,3 V	5 V	1,8 V	1,8 V
MFO	2,15 GHz	1,9 GHz	1,22 GHz	3 GHz
FOM (GHz/mW)	0,185	0,05	0,36	4,622
Consumo	11,6 mW	38 mW	3,36 mW	0,649 mW

Para a verificação do consumo de corrente foram feitas simulações do consumo de corrente variando no tempo. Esta simulação foi realizada quando MC=0 e MC=1, obtendo diferentes valores de corrente para as frequências de 2,40 GHz e 2,80 GHz. A tabela 8 apresenta estes dados de forma mais clara.

Tabela 8 – Consumo do *prescaler* para diferentes frequências de operação.

	FinN = 2,40 Ghz	FinN = 2,80 Ghz
MC=0	521,7 μW	607,1 μW
MC=1	479,7 μW	558,6 μW

Por fim, a tabela 9 apresenta os pinos do *prescaler* e suas características.

Tabela 9 – Pinos do *prescaler* 15/16.

Nome	Descrição	Tipo de Pino
FinN	Entrada do sinal	Entrada
MC	Controle do fator de divisão	Entrada
FoutN	Saída do sinal dividido	Saída
avdd	Fonte de alimentação	Entrada
agnd	Terra	Entrada

7.3 Contador Principal (P) Divisor 32

Como dito anteriormente, trata-se de um divisor com fator de divisão igual a 32 e a entrada deste bloco é conectada a saída do *prescaler*. Consiste em cinco *flip-flops* tipo D TSPC conectados em cascata sendo totalmente assíncrono (figura 44).

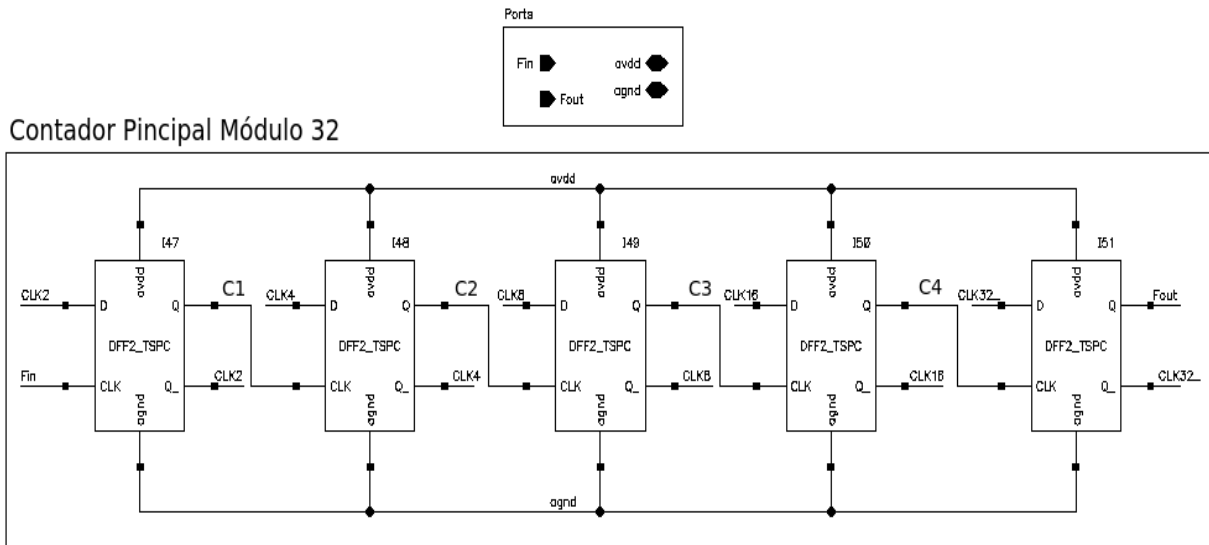


Figura 44 – Esquemático do contador P.

A simulação foi realizada com tensão de referência igual a 1,8 V e frequência de entrada igual a 1 GHz. A figura 45 apresenta o esquemático do contador P com os sinais internos do mesmo. O fator de divisão é de $(43,4496 - 11,4487) \cdot 10^{-6} / (1/1.10^6) = 32$.

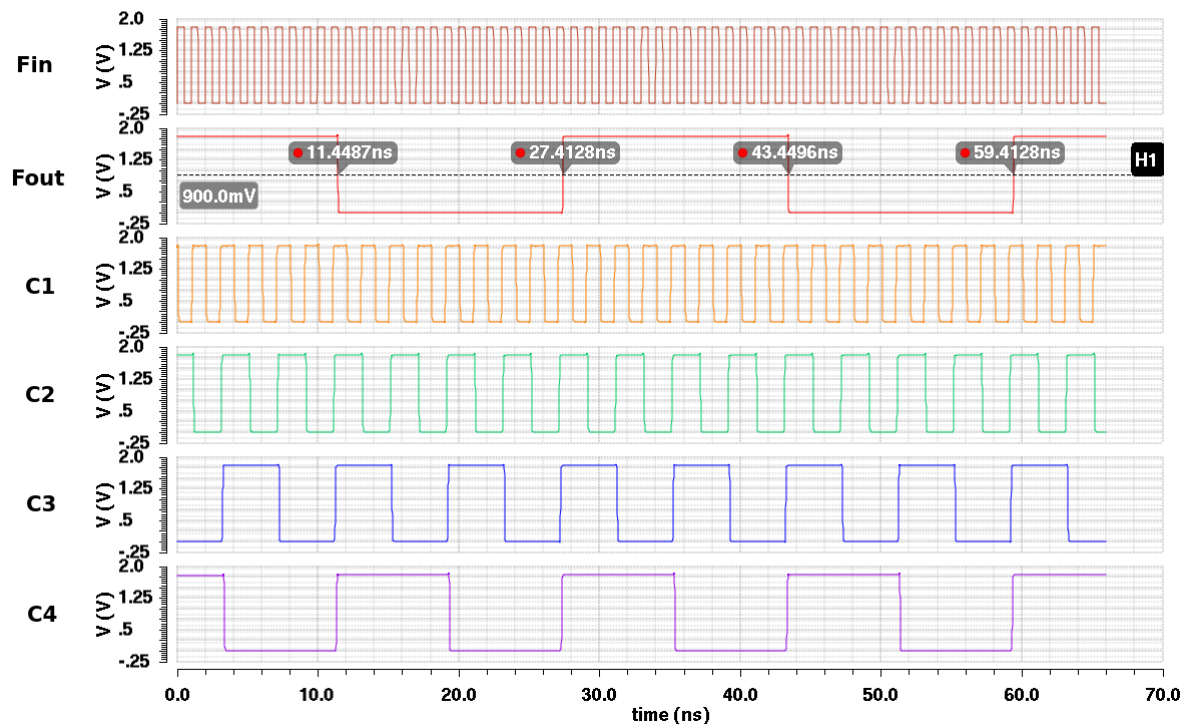


Figura 45 – Resultado de simulação do contador P possuindo frequência de entrada igual a 1 GHz.

Com frequência de entrada igual a 1 GHz, o consumo médio foi de $167,523 \mu W$. A MFO do contador P é, como pode ser visualizado na figura 46, de 8 GHz e, para esta frequência de entrada, o consumo foi de $1,077 mW$.

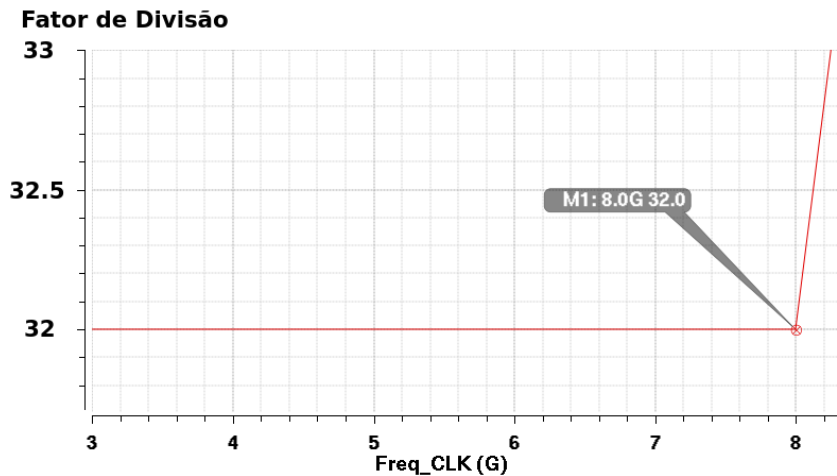


Figura 46 – Resultado de simulação da MFO do contador P.

A tabela 10 apresenta os pinos de entrada e saída do contador P.

Tabela 10 – Pinos do contador P com módulo igual a 32.

Nome	Descrição	Tipo de Pino
Fin	Entrada do sinal	Entrada
Fout	Saída do sinal dividido	Saída
avdd	Referência de Tensão	Entrada
agnd	Terra	Entrada

7.4 Swallow Counter (S) Divisor 16

O *swallow counter* é um contador síncrono de módulo 16, pois o ZigBee é composto por 16 canais. Sua principal função é selecionar um dos canais de acordo com a palavra binária formada por IA, IB, IC e ID, sendo ID e IA os bits mais e menos significativos, respectivamente. Para o projeto deste divisor utilizou-se apenas o *flip-flop* na topologia TSPC e algumas das portas lógicas citadas na [subseção 7.1.1](#). A figura 47 apresenta o *test bench* do *swallow counter*.

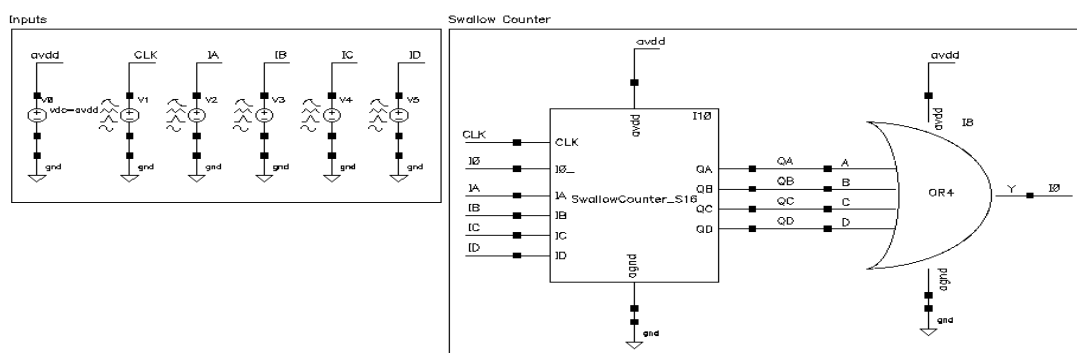


Figura 47 – Test bench do swallow counter.

Seu funcionamento ocorre da seguinte forma: a entrada $I0_$ age como um reset, ou seja, quando $I0_ = 0$ a contagem para no módulo de divisão indicado por ID-IA, o que explica a constância das ondas QA, QB, QC e QD neste estágio (figura 48), e quando $I0_ = 1$ o *swallow counter* continua a sua contagem, desde o módulo de divisão que representa a palavra binária de entrada ID-IA, até zero. Se $I0_ = 1$ e houver uma mudança em ID-IA, esta alteração no fator de divisão não irá se refletir na saída, pois só quando $I0_ = 0$ que o fator de divisão, e por consequência o canal selecionado deste bloco, pode ser alterado. Isso explica a necessidade de uma porta OR de quatro entradas realimentando o *swallow counter* em $I0_$ (figuras 48 e 49).

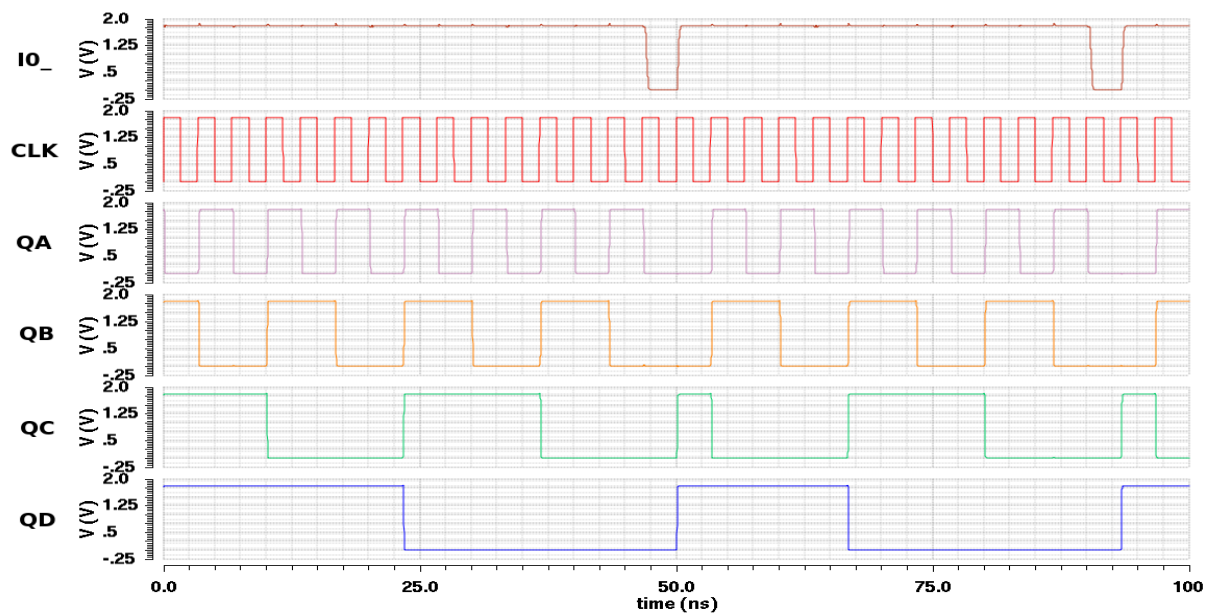


Figura 48 – Simulação do 12º canal do *swallow counter*.

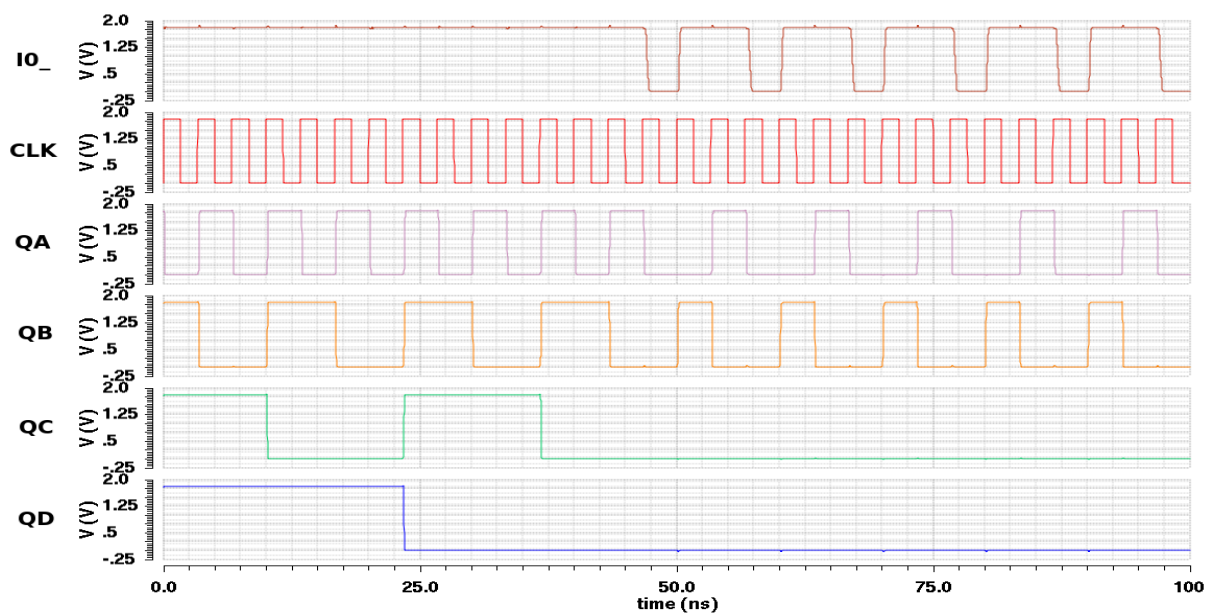


Figura 49 – Simulação do 2º canal do *swallow counter*.

Realizou-se simulações dos 16 canais com frequência de entrada de 300 MHz e os dados de consumo foram coletados, sendo estes apresentados na figura 50. Com isto, o consumo médio desde divisor é de $234,627\mu W$.

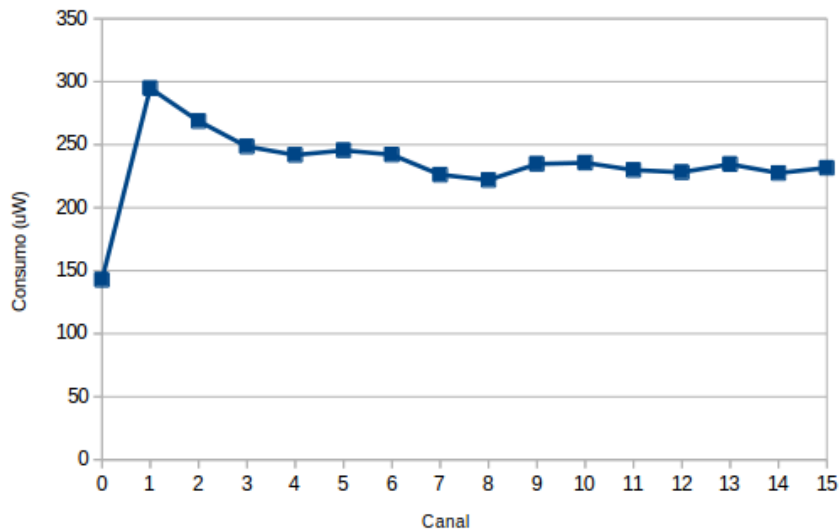


Figura 50 – Gráfico do consumo do *swallow counter* em todos os canais.

Por fim, a tabela 11 apresenta a pinagem do *swallow counter*. Nesta tabela, os pinos de saída QA, QB, QC e QD representam cada um dos bits do sinal dividido com QD e QA sendo os bits mais e menos significativos, respectivamente.

Tabela 11 – Pinos de entrada e saída do *swallow counter*.

Nome	Descrição	Tipo de Pino
CLK	Entrada do sinal	Entrada
I0_	Pino de reset	Entrada
IA	Entrada, bit 0	Entrada
IB	Entrada, bit 1	Entrada
IC	Entrada, bit 2	Entrada
ID	Entrada, bit 3	Entrada
QA	Saída, bit 0	Saída
QB	Saída, bit 1	Saída
QC	Saída, bit 2	Saída
QD	Saída, bit 3	Saída
avdd	Referência de tensão	Entrada
agnd	Terra	Entrada

7.5 Divisor Completo e Lógica de Controle

A lógica de controle do *pulse swallow counter* tem como objetivo unir os três divisores principais de forma que o sistema funcione conforme a teoria apresentada, sendo o multiplexador e a porta OR de quatro entradas os responsáveis por esta tarefa. A porta OR de quatro entradas é utilizada para alterar o fator de divisão do *prescaler*, bem como resetar o próprio *swallow counter* e alterar a entrada do multiplexador. A figura 51 apresenta o esquemático da completa implementação do divisor de frequências.

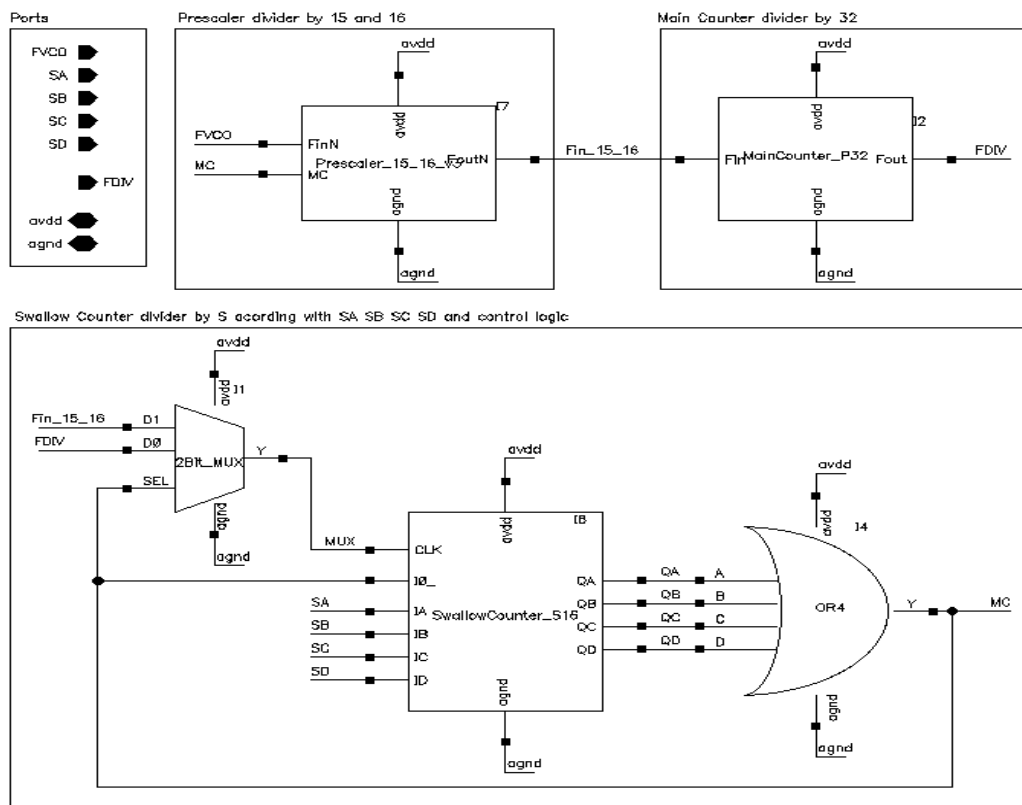


Figura 51 – Esquemático do divisor completo (*Pulse Swallow Counter*).

Os números em parênteses que serão escritos neste parágrafo representam pontos nas ondas da figura 52 para um melhor entendimento sobre o funcionamento do divisor completo. Inicialmente o *clock* de entrada do *swallow counter* é conectado a saída do *prescaler* (1). Devido a porta OR4, a entrada MC irá apresentar nível lógico 1 fazendo com que o *prescaler* realize a divisão por 16 (2). Quando o *swallow counter* contar até zero e terminar sua contagem, a saída da porta OR4 será zero (3) fazendo com que a fonte de *clock* do *swallow counter* seja alterada para a saída do contador P, que também é a saída do divisor completo. Neste momento o *prescaler* começa a dividir por 15, já que $MC=0$. A troca da fonte de *clock* do *swallow counter* tem como objetivo manter a saída da porta OR4 igual a 0 até que o contador P encerre sua contagem, tornando possível a divisão por 15 do *prescaler*, pois sem o multiplexador, a divisão por 15 iria acontecer apenas por um curto período de tempo. Ao fim da contagem do contador P, o *swallow*

counter recebe uma borda de subida (4) e é reiniciado com o valor S do canal selecionado. Assim, uma nova contagem tem início, pois a saída da OR4 é igual a 1, fazendo com que *swallow counter* receba o sinal de saída do *prescaler* como *clock* e um novo ciclo de divisão tem início. Por meio da comparação entre as figuras 52 e 53 é possível entender melhor as características explicadas acima.

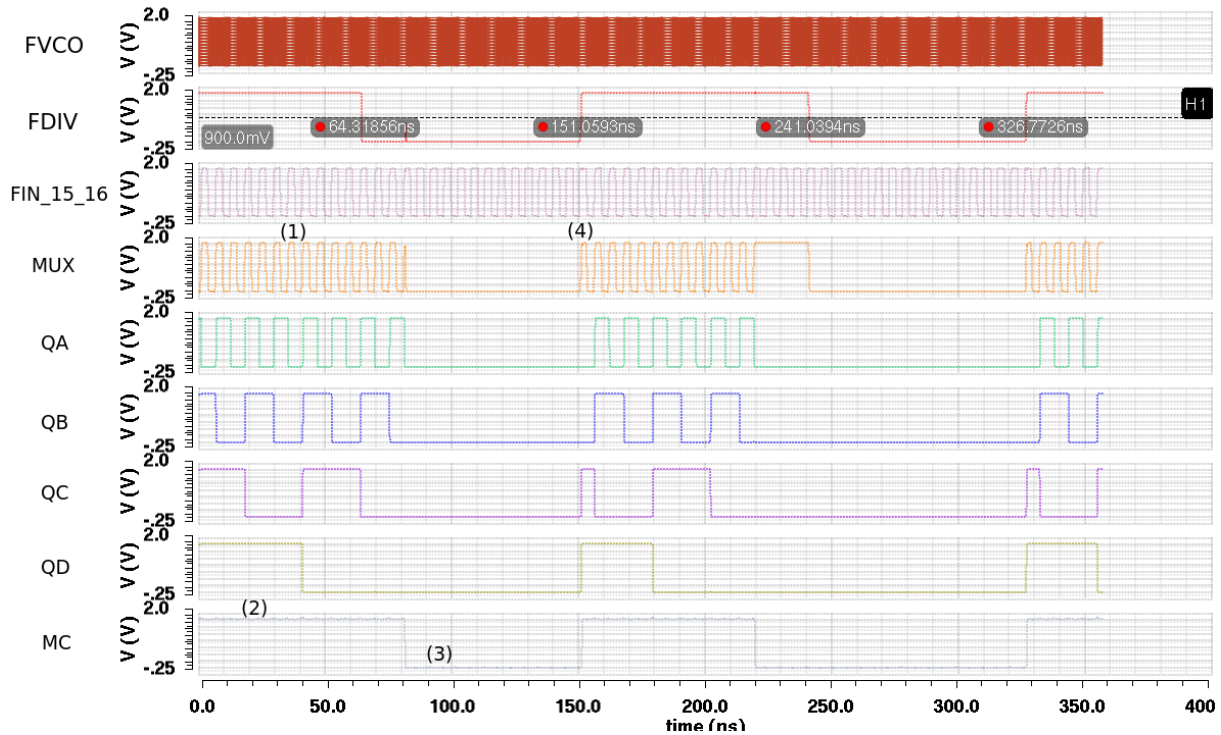


Figura 52 – Simulação do 12º canal do divisor completo.

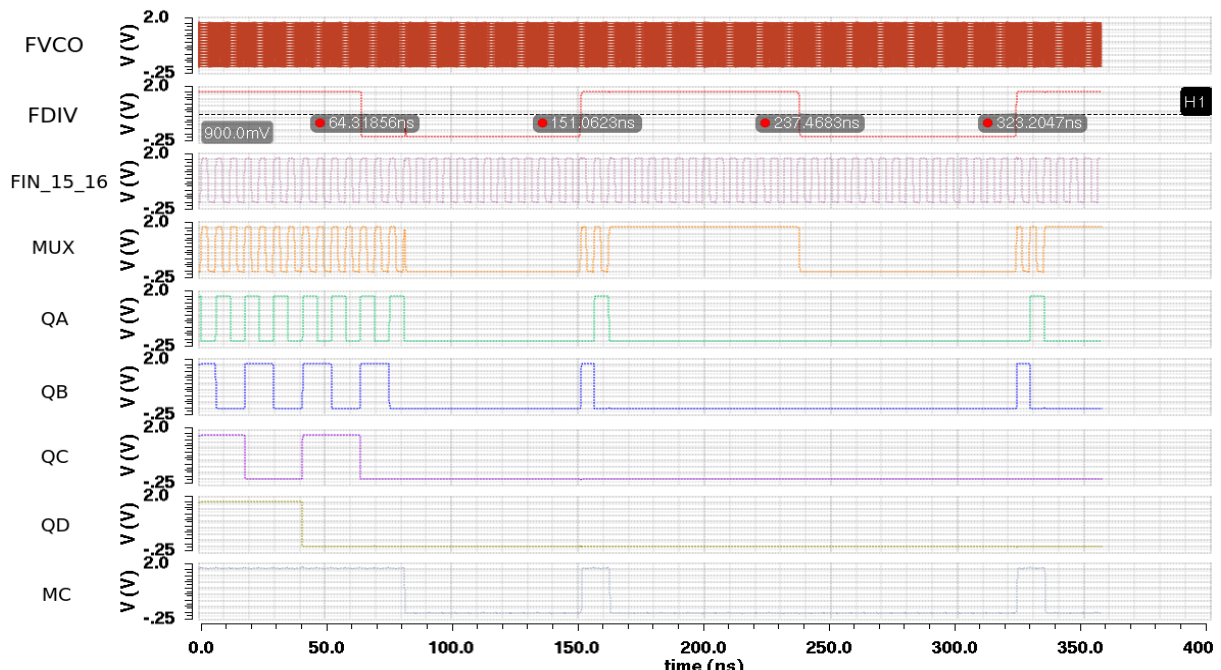


Figura 53 – Simulação do 2º canal do divisor completo.

Foram realizadas simulações para todos os 16 canais com frequência de entrada igual a 2,80 GHz e foi calculado o consumo para cada canal. O consumo médio nestas condições é em torno de $728,64\mu W$ e a figura 54 apresenta os dados de consumo coletados em cada um dos canais.

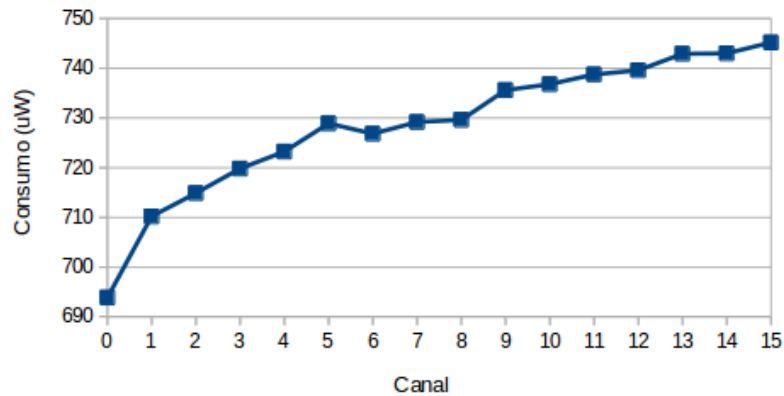


Figura 54 – Gráfico do consumo dos 16 canais do divisor completo.

O fator de divisão também foi calculado em cada simulação dos 16 canais e apresentou uma certa peculiaridade. O divisor completo realiza divisões com módulos que variam entre 480 e 495, do canal 0 ao canal 15, respectivamente. Porém, em todas as simulações o fator de divisão apresentou erro na terceira casa decimal, ou seja, ao invés de apresentar uma divisão por 490, por exemplo, o valor calculado foi 489,997. Isso ocorre pois o *flip-flop* utilizado na construção de todos os divisores foi o TSPC que apresenta certa limitação em sua velocidade de operação. Como a diferença é bastante pequena, isso não influenciou no correto funcionamento do PLL deste trabalho. As tabelas 12 e 13 apresentam todos os fatores de divisão.

Tabela 12 – Fatores de divisão com DFF TSPC em todos os divisores. Tabela 13 – Fatores de divisão com DFF TSPC em todos os divisores.

Canal	Fator de Divisão
0	480
1	480,998
2	481,998
3	482,997
4	483,998
5	484,997
6	485,997
7	486,996

Canal	Fator de Divisão
8	487,999
9	488,997
10	489,997
11	490,995
12	491,997
13	492,996
14	493,996
15	494,994

Os resultados acima são característicos dos divisores utilizando o *flip-flop* TSPC.

Uma outra alternativa seria utilizar o *flip-flop* E-TSPC. Entretanto, como o *prescaler* não funcionou corretamente com este ultimo *flip-flop* e utilizar esta topologia no *swallow counter* seria um desperdício de energia, pois este divisor opera em baixas frequências, trocou-se os *flip-flops* do contador P para a topologia E-TSPC e novas simulações do divisor completo foram realizadas.

Como resultados, foram obtidos fatores de divisão mais precisos se comparado a utilização do TSPC, porém o consumo de energia sofreu um grande aumento, pois embora o E-TSPC seja mais rápido, e por isso utilizado para projetos em altas frequências, ele consome mais energia, como mencionado na [subseção 7.1.2.2](#). Dessa forma, existe uma troca entre velocidade de operação e consumo. Como um dos objetivos deste trabalho é projetar um divisor de frequências para um PLL que possua o menor consumo possível, a utilização da topologia TSPC é a mais efetiva neste caso.

A tabela 14 apresenta todos os fatores de divisão utilizando DFF's E-TSPC no divisor P. É possível observar uma melhora na precisão do fator de divisão.

Tabela 14 – Fatores de divisão utilizando DFF E-TSPC apenas no contador P.

Canal	Fator de Divisão
0	480,003
1	481,002
2	482,002
3	483
4	484,002
5	485
6	486
7	486,998
8	488,002
9	489
10	490
11	490,998
12	492
13	492,998
14	493,998
15	494,996

A figura 55 mostra o consumo em cada um dos 16 canais, sendo o consumo médio igual a 2,143 mW, um valor bastante elevado se comparado com o uso do DFF TSPC.

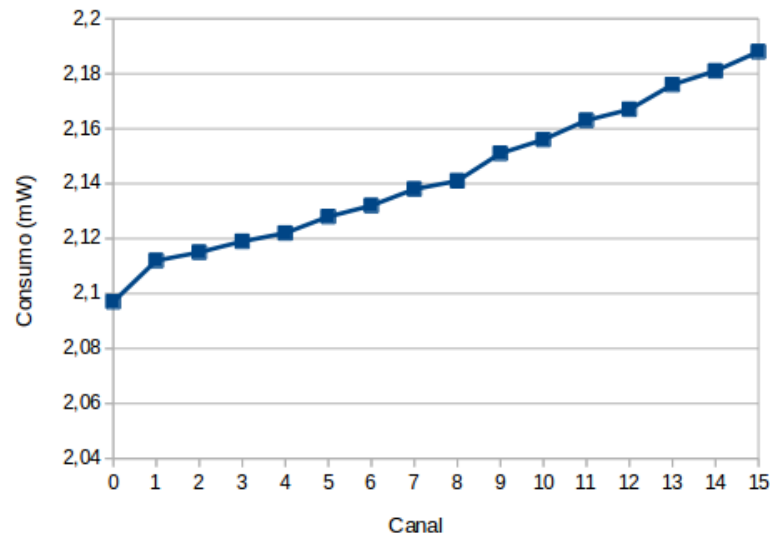


Figura 55 – Simulação do consumo do divisor completo utilizando o *flip-flop* E-TSPC no divisor P.

A tabela 15 apresenta os pinos do divisor completo. Onde SA, SB, SC e SD formam uma palavra de 4 bits responsável pela seleção do canal, nota-se que SA é o bit menos significativo e SD é o bit mais significativo.

Tabela 15 – Pinos do divisor completo.

Nome	Descrição	Tipo de Pino
FVCO	Sinal de entrada	Saída
SA	Bit 0	Entrada
SB	Bit 1	Entrada
SC	Bit 2	Entrada
SD	Bit 3	Entrada
FDIV	Sinal Dividido	Saída
avdd	Referência de Tensão	Entrada
agnd	Terra	Entrada

7.6 Conversor de Entrada Diferencial para *Single-Ended*

Este bloco, como o próprio nome diz, realiza a conversão de um sinal diferencial, neste caso a saída do VCO, para a entrada do divisor de frequências do PLL. Este conversor é necessário por dois motivos: o divisor do PLL possui apenas uma entrada para dividir o sinal do VCO e para o divisor funcionar corretamente, o sinal de entrada deve possuir máxima excursão de entrada, ou seja, o valor mínimo deve ser zero e valor máximo deve ser o mais próximo possível de V_{DD} .

Este bloco funciona como um amplificador diferencial com um estágio de ganho (figura 56). O primeiro estágio, formado pelos transistores M0, M1, M13 e resistores R0 e R1, tem como finalidade converter os sinais de entrada para uma saída *single-ended*. Como os sinais de entrada apresentam fase de 180° , o ganho deste estágio é máximo, pois $\Delta V_{IN} = V_{in_n} - V_{in_p} = 0$, ou seja, as correntes que passam em M0 e M1 são iguais.

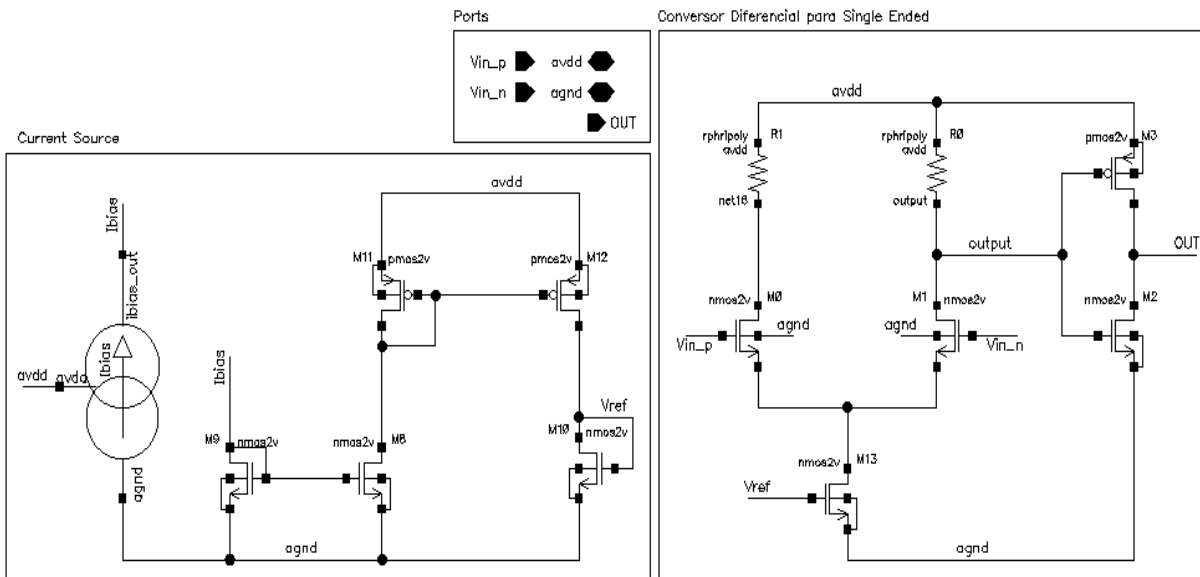


Figura 56 – Esquemático do conversor diferencial para *single-ended*.

É de grande importância que o nível DC de saída do inversor seja de 0,9 V, devido ao fato de todos os sinais de entrada e saída dos divisores apresentarem esta característica. Dessa forma, foi estipulado uma corrente de polarização de $I_{SS} = 200\mu A$ para o conversor e os resistores foram calculados a partir dessa corrente: $V_{OUT-DC} = V_{DD} - R_D \cdot I_{SS} \Rightarrow 1,8 = 0,9 - R_D \cdot 200 \cdot 10^{-6} \Rightarrow R_D = 9k\Omega$. O segundo estágio, composto por M2 e M3, trata-se de uma porta NOT com o objetivo de aumentar ao máximo a excursão de saída do conversor, já que a excursão de saída do VCO não é muito grande. O conversor conta ainda com uma fonte de corrente I_{bias} igual a $10\mu A$, sendo esta espelhada pelos transistores M8, M9, M10, M11 e M12 para a corrente de polarização do par diferencial $I_{SS} = 200\mu A$.

A faculdade UnB Gama trabalha atualmente na construção de uma biblioteca de IPs (*intellectual property*), contendo blocos funcionais para a reutilização em projetos de circuitos integrados complexos. Este bloco compõe esta biblioteca e adequou-se perfeitamente a este projeto. O esquemático dessa fonte pode ser encontrada em anexo.

A figura 57 contém o resultado de simulação com frequência de entrada igual a 2,80 GHz, senoide de entrada com nível DC igual a 0,9 V e excursão de entrada de $500mV$, bem como $V_{DD} = 1,8V$. O consumo neste tipo de simulação foi de $879,7\mu W$ e a frequência de saída foi de aproximadamente 2,80 GHz.

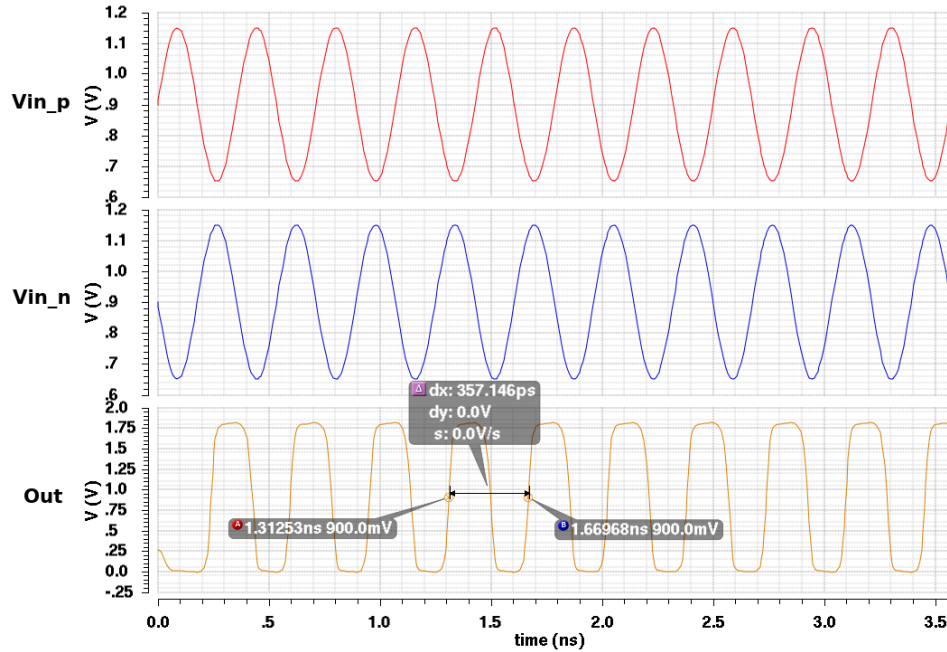


Figura 57 – Simulação do conversor diferencial para *single-ended*.

A validação do conversor se deu ao simular este bloco em conjunto ao divisor completo, no qual o *test bench* pode ser visualizado por meio da figura 58. A simulação (figura 59) foi realizada para o canal 8 do PLL e seu fator de divisão pode ser calculado como $(499,97058 - 325,68421) \cdot 10^{-9} / (1/2, 8 \cdot 10^9) = 488$.

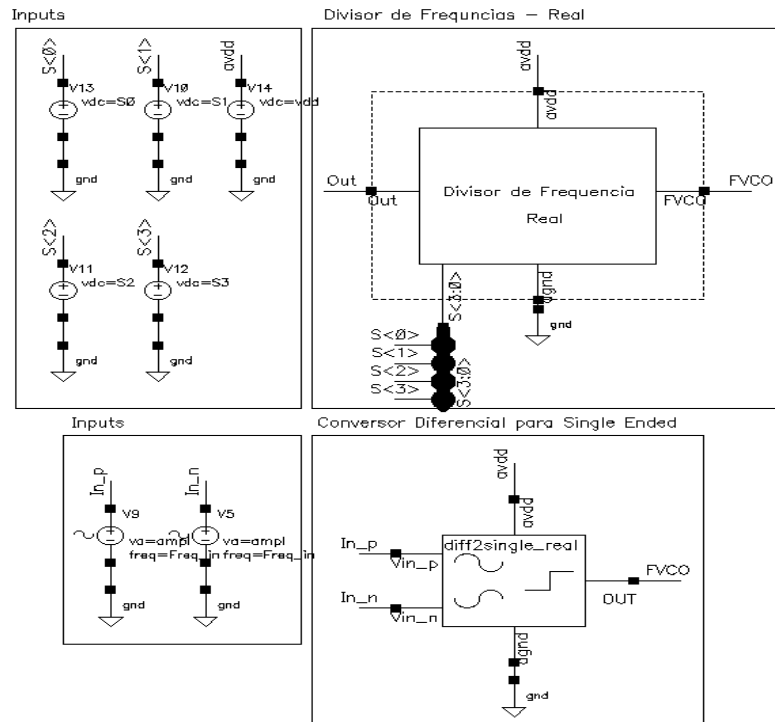


Figura 58 – *Test bench* do divisor completo em conjunto do conversor diferencial para *single-ended*.

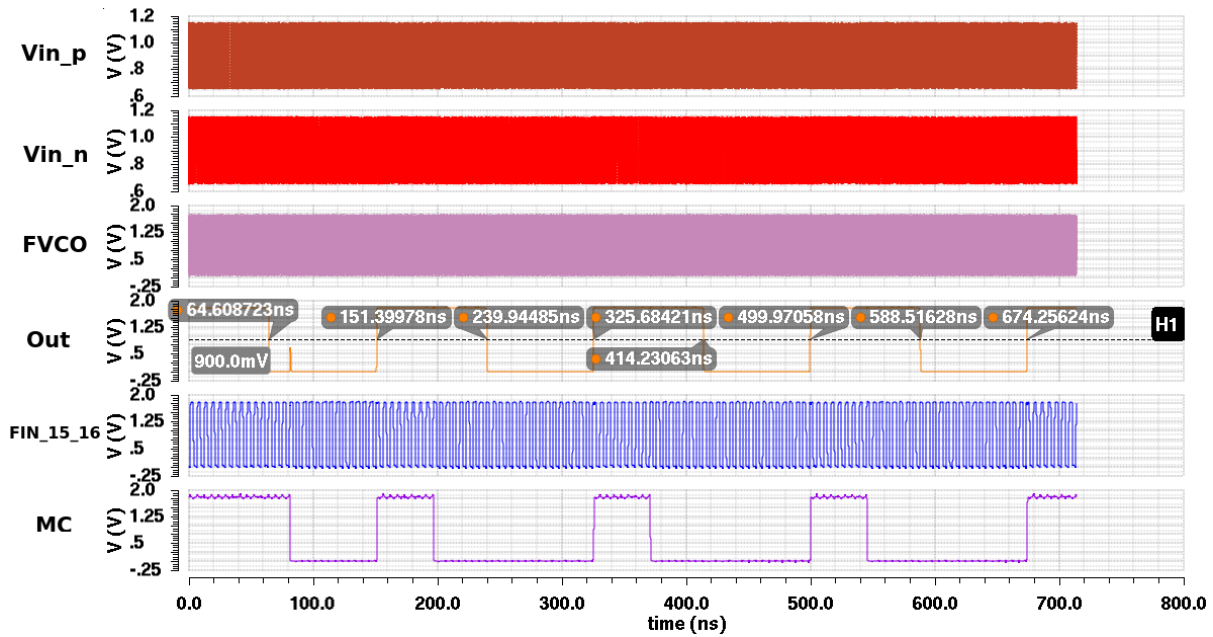


Figura 59 – Simulação do divisor completo em conjunto do conversor diferencial para *single-ended*.

Por fim, a tabela 16 mostra os pinos de entrada e saída do conversor.

Tabela 16 – Pinos do conversor diferencial para *single-ended*.

Nome	Descrição	Tipo de Pino
V_{inp}	Entrada 1	Entrada
V_{inn}	Entrada 2	Entrada
OUT	Saída do circuito	Saída
avdd	Referência de Tensão	Entrada
agnd	Terra	Entrada

7.7 Simulações Mistas

A simulação mista foi realizada comparando o divisor real (projetado neste trabalho a nível de transistor) e o divisor projetado utilizando a linguagem de descrição de *hardware* analógico Verilog-AMS [Guimarães (2015)] com o objetivo de verificar se o divisor proposto para o PLL era realizável. Ao fim do projeto do divisor a nível de transistor, a modelagem em alto nível foi atualizada com os mesmos parâmetros do divisor real (entre eles fator de divisão, velocidade de operação, entre outros), tendo em vista o objetivo de verificar se o PLL iria se comportar da forma esperada utilizando a versão atualizada da modelagem do divisor.

A figura 60 contém o *test bench* da simulação realizada com sinais de entrada, In_n e In_p , iguais a 2,525 GHz, com fase de 180° entre eles e excursão de entrada igual a $500mV$. Os sinais Out1, Out, FVCO representam, respectivamente, a saída do divisor real, a saída do divisor feito por meio da modelagem na linguagem Verilog-AMS e a saída do conversor diferencial para *single-ended* que possui como entradas o sinal de saída do VCO.

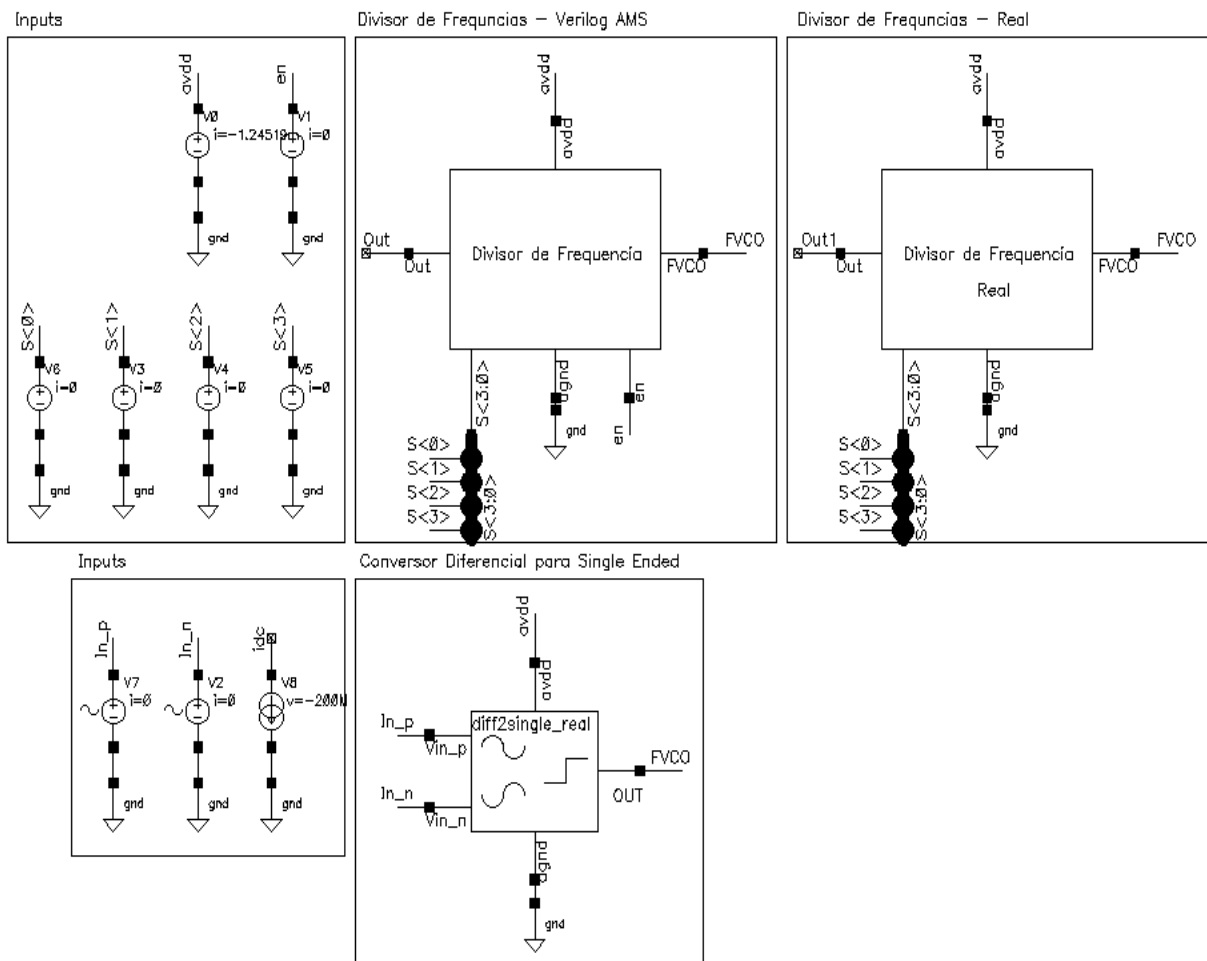


Figura 60 – *Test bench* da comparação entre o divisor completo a nível de transistor e a modelagem em Verilog-AMS.

É observado que ambos os sinais de saída são bastante semelhantes, diferindo apenas no atraso das portas lógicas e dos *flip-flops* utilizados no projeto de todo o divisor (figura 61). A diferença entre as ondas pode ser observada por meio do sinal Out_{diff} da figura 62.

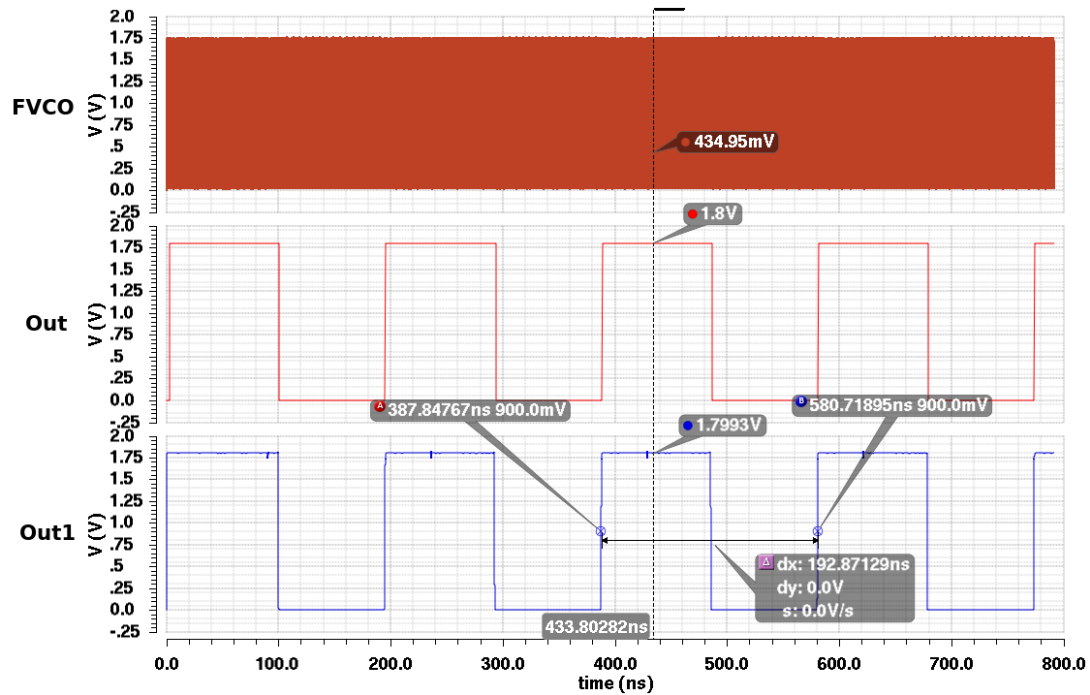


Figura 61 – Simulação do divisor completo a nível de transistor e a modelagem em Verilog-AMS em conjunto do conversor diferencial para *single-ended*.

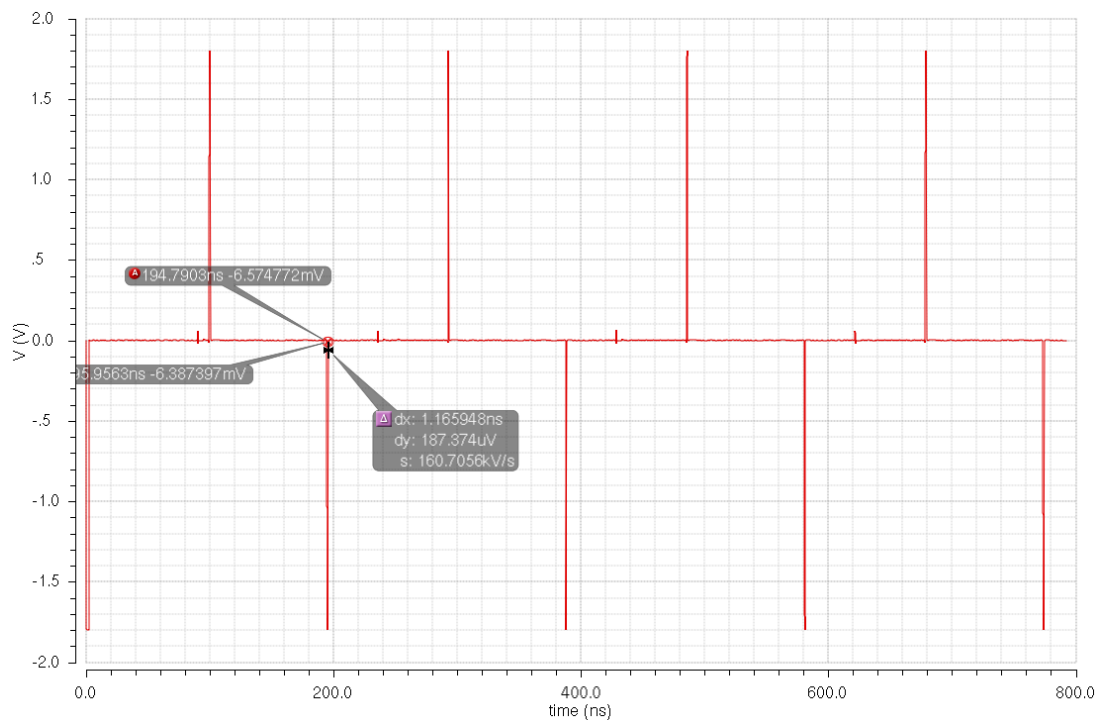


Figura 62 – Gráfico da diferença entre a simulação divisor completo a nível de transistor e da modelagem em Verilog-AMS em conjunto do conversor diferencial para *single-ended*.

7.8 Resumo dos resultados

Tendo em vista o grande volume de resultados obtidos, este tópico tem como finalidade resumir os resultados e apresentá-los de forma mais objetiva. Dessa forma, a tabela 17 apresenta os principais resultados citados nos tópicos anteriores de cada bloco, onde FD representa o fator de divisão.

Tabela 17 – Resumo dos resultados obtidos para os 3 blocos principais do divisor completo.

	Consumo Máximo	MFO	FD esperado	FD obtido
<i>Prescaler</i> (N/N+1)	607,1 μ W	3 GHz	15/16	15/16
<i>Swallow Counter</i> (S)	234,627 μ W	-	16	16
Contador Principal (P)	167,523 μ W	8 GHz	32	32
Divisor Completo	745,172 μ W	-	480-495	480-495

8 Conclusão

A modernidade e o desenvolvimento de novas tecnologias fizeram com que o sistema de produção atual fosse alterado de forma a atender as novas demandas do mercado. Técnicas de sensoriamento remoto e de identificação são exemplos práticos dessas novas tecnologias. Neste contexto, o conjunto de especificações ZigBee entra com o objetivo de garantir as especificações para a comunicação *wireless* entre sistemas eletrônicos. Isto é feito através da identificação por rádio frequência (RFID) de forma a aumentar a sua produtividade. O principal foco do ZigBee é a construção de sistemas com baixo consumo, baixa taxa de transferência de dados e com baixo custo de execução e efetivação. Dentre as três bandas principais do ZigBee a banda de 2,4 GHz é a mais utilizada, pois seu padrão possui especificação internacional, enquanto as outras duas não.

O sintetizador de frequências baseado na técnica *Phase Locked Loop* tem papel fundamental num transceptor de sistemas *wireless* em geral, principalmente os que funcionam no padrão ZigBee, e é um dos circuitos do transceptor que apresenta maior gasto energético. O divisor de frequências é um dos blocos que possui maior desafio no projeto de um PLL, pois apresenta consumo elevado. Dessa forma, projetar um divisor que apresente baixo consumo é um desafio necessário para o bom projeto do PLL.

Neste trabalho foi projetado um divisor de frequências na topologia *Pulse Swallow Counter*, do tipo Inteiro-N. Este divisor é composto por três blocos: *prescaler* com dois módulos de divisão ($N/N + 1$); *swallow counter* (S) e o contador principal (P). O projeto teve início com o projeto das portas lógicas e dos *flip-flops* tipo D nas topologias TSPC e E-TSPC. Foram realizadas simulações para verificar o correto funcionamento e o consumo energético delas.

Em seguida os três blocos que compõem o divisor completo foram projetados e testados separadamente com uma frequência de entrada maior que a especificada pela região de operação do ZigBee, de forma a aumentar a confiabilidade do sistema. O *prescaler* e o contador P foram avaliados com os *flip-flops* TSPC e E-TSPC para verificar qual dos dois apresenta melhor performance para o problema proposto. Verificou-se que ao usar o DFF TSPC, os dois divisores citados anteriormente funcionaram de maneira correta, apresentando fatores de divisão precisos e com baixo consumo. Já ao utilizar a topologia E-TSPC no *prescaler*, seu funcionamento era comprometido, apresentando ondas bastante deformadas, fator de divisão diferente do esperado, maior consumo se comparado a utilização do TSPC e grande variação no consumo com o aumento da frequência de operação. Já no contador P, quando foi utilizado o DFF E-TSPC, seu fator de divisão foi mais preciso e o consumo, como esperado, também aumentou.

Os três divisores supracitados, todos utilizando o DFF TSPC, foram unidos em conjunto com a lógica de controle para dar forma ao divisor completo e, novamente, foram realizadas simulações em cada um dos 16 canais de operação. O gasto energético foi satisfatório, ficando próximo do gasto proposto nos objetivos. Também apresentou bons fatores de divisão e funcionou perfeitamente com frequência de entrada igual a 2,8 GHz. Como o *prescaler* utilizando o DFF E-TSPC não funcionou corretamente, foram realizadas outras simulações do divisor completo utilizando o DFF E-TSPC no contador P. Levando-se em consideração a velocidade de operação e fatores de divisão precisos, os resultados foram satisfatórios. Porém, neste caso o consumo aumentou mais que o dobro quando se utilizava exclusivamente o DFF E-TSPC no contador P. Dessa forma, para se projetar o divisor para PLL de baixo consumo, é importantíssimo que se utilize *flip-flops* TSPC no projeto dos três divisores internos da topologia *Pulse Swallow Counter*. Mas se o principal objetivo for obter divisores com alta velocidade de operação, não tendo muitas restrições no consumo de energia, então a topologia E-TSPC é a mais indicada.

Como proposta de trabalhos futuros, tem-se a verificação do funcionamento do divisor completo nos *corners*, ou seja, nas condições de operação mais extremas, como faixa de variação do V_{DD} e da temperatura que garanta o correto funcionamento do divisor. Bem como o *layout* e verificação pós *layout*. Outra sugestão é o projeto e inclusão de um modulador *Sigma-Delta* ($\Sigma - \Delta$) no divisor projetado neste trabalho, obtendo assim um divisor fracionário, e posteriormente, realizar os testes desse novo divisor no PLL construído.

Referências

- AKYILDIZ I. F.; SU, W. S. Y.; CAYIRCI, E. A survey on sensor networks. *IEEE Communications Magazine*, v. 40, n. 8, p. 102–114, 2008. Citado na página 43.
- BARRETT, C. Fractional/integer-n pll basics. Texas Instruments, Wireless Communication Business Unit, Technical Brief SWRA029, 1999. Citado na página 48.
- BISTUE, G.; QUEMADA, C.; ADIN, I. *Design methodology for RF CMOS phase locked loops*. [S.l.]: Artech House, 2009. Citado na página 68.
- CHEN, W.-H.; JUNG, B. High-speed low-power true single-phase clock dual-modulus prescalers. *Circuits and Systems II: Express Briefs, IEEE Transactions on*, IEEE, v. 58, n. 3, p. 144–148, 2011. Citado 3 vezes nas páginas 16, 65 e 67.
- CHIEN, T.-H. et al. High-speed and low-power programmable frequency divider. In: IEEE. *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*. [S.l.], 2010. p. 4301–4304. Citado na página 59.
- DENG, Z.; NIKNEJAD, A. M. The speed–power trade-off in the design of cmos true-single-phase-clock dividers. *Solid-State Circuits, IEEE Journal of*, IEEE, v. 45, n. 11, p. 2457–2465, 2010. Citado na página 65.
- ERGEN, S. C. Zigbee/ieee 802.15. 4 summary. *UC Berkeley, September*, v. 10, 2004. Citado 2 vezes nas páginas 19 e 45.
- FITZPATRICK, D.; MILLER, I. *Analog behavioral modeling with the Verilog-A language*. [S.l.]: Springer, 1998. Citado 4 vezes nas páginas 15, 33, 39 e 40.
- GANG, P. Behavioral modeling and simulation of analog/mixed-signal systems using verilog-ams. In: IEEE. *Information, Computing and Telecommunication, 2009. YC-ICT'09. IEEE Youth Conference on*. [S.l.], 2009. p. 383–386. Citado na página 40.
- GOLIO, M.; GOLIO, J. *RF and microwave circuits, measurements, and modeling*. [S.l.]: CRC press, 2010. Citado 7 vezes nas páginas 15, 47, 49, 50, 52, 53 e 54.
- GUIMARÃES, T. A. N. Modelagem de um pll e projeto de vco para transceptor zigbee. 2015. Citado na página 92.
- GUO, C. et al. Design and optimization of dual modulus prescaler using the extended true-single-phase-clock. In: IEEE. *Microwave and Millimeter Wave Technology (ICMMT), 2010 International Conference on*. [S.l.], 2010. p. 636–638. Citado na página 66.
- HSIEH, G.-C.; HUNG, J. C. Phase-locked loop techniques. a survey. *Industrial Electronics, IEEE Transactions on*, IEEE, v. 43, n. 6, p. 609–615, 1996. Citado na página 53.
- ISMAIL, N. M.; OTHMAN, M. Cmos porgrammable divider for zigbee frequency synthesizer. *International Conference on Signals, Circuits and Systems*, 2009. Citado na página 59.

- ISMAIL, N. M.; OTHMAN, M. System analysis of 2.4 ghz ieee 802.15. 4 compliant frequency synthesizer. In: IEEE. *Microelectronics (ICM), 2009 International Conference on*. [S.l.], 2009. p. 197–200. Citado na página 59.
- JR, J. N. S.; NOIJE, W. A. V. A 1.6-ghz dual modulus prescaler using the extended true-single-phase-clock cmos circuit technique (e-tspc). *Solid-State Circuits, IEEE Journal of, IEEE*, v. 34, n. 1, p. 97–102, 1999. Citado na página 69.
- KALAIVANI, T.; ALLIRANI, A.; PRIYA, P. A survey on zigbee based wireless sensor networks in agriculture. In: IEEE. *Trendz in Information Sciences and Computing (TISC), 2011 3rd International Conference on*. [S.l.], 2011. p. 85–89. Citado na página 29.
- KRISHNA, M. V. et al. A 1.8-v 3.6-mw 2.4-ghz fully integrated cmos frequency synthesizer for the ieee 802.15. 4. In: *VLSI-SoC: Forward-Looking Trends in IC and Systems Design*. [S.l.]: Springer, 2012. p. 69–99. Citado na página 59.
- KROUPA, V. F. *Phase Lock Loops and Frequency Synthesis*. [S.l.]: John Wiley and Sons, Ltd ISBN: 0-470-84866-9, 2003. Citado na página 47.
- KUNDERT, K.; ZINKE, O. *The designer's guide to Verilog-AMS*. [S.l.]: Springer, 2004. Citado 3 vezes nas páginas 34, 35 e 36.
- KUNDERT K.; CHANG, H. Analog verification: Modeling analog for verification is easy. Disponível em <<http://www.designers-guide.com/newsletters/0709/>>. Acessado em 03 de novembro de 2014., 2007. Citado 2 vezes nas páginas 15 e 41.
- LACAITA, A.; LEVANTINO, S.; SAMORI, C. *Integrated frequency synthesizers for wireless systems*. [S.l.]: Cambridge University Press New York, NY, 2007. Citado 7 vezes nas páginas 15, 16, 58, 61, 65, 68 e 75.
- LARSSON, P. High-speed architecture for a programmable frequency divider and a dual-modulus prescaler. *Solid-State Circuits, IEEE Journal of, IEEE*, v. 31, n. 5, p. 744–748, 1996. Citado na página 80.
- LEWIS, F. L. Wireless sensor networks. *Smart environments: technologies, protocols, and applications*, New York: Wiley, p. 11–46, 2004. Citado 3 vezes nas páginas 15, 43 e 44.
- LI, Z. et al. A 5-ghz frequency synthesizer with afc for low if zigbee transceiver applications. In: IEEE. *New Circuits and Systems Conference (NEWCAS), 2011 IEEE 9th International*. [S.l.], 2011. p. 530–533. Citado 2 vezes nas páginas 15 e 28.
- LIU, H. et al. Low power cmos high speed dual-modulus 15/16 prescaler for wireless communications. In: IEEE. *Communications and Mobile Computing (CMC), 2011 Third International Conference on*. [S.l.], 2011. p. 397–400. Citado na página 80.
- LTD, J. Jn-ug-3017-zigbee stack user guide. Revision 1.6, 2008. Citado na página 45.
- MANTHENA, V. K. *Ultra Low Power CMOS Phase-Locked Loop Frequency Synthesizers*. Tese (Doutorado) — Nanyang Technological University, 2011. Citado na página 50.

- MCCORQUODALE, M. S. et al. A top-down microsystems design methodology and associated challenges. In: IEEE. *Design, Automation and Test in Europe Conference and Exhibition, 2003*. [S.l.], 2003. p. 292–296. Citado 3 vezes nas páginas 15, 35 e 37.
- NDZI, D. L. et al. Wireless sensor network coverage measurement and planning in mixed crop farming. *Computers and Electronics in Agriculture*, Elsevier, v. 105, p. 83–94, 2014. Citado na página 29.
- PALNITKAR, S. *Verilog HDL: a guide to digital design and synthesis*. [S.l.]: Prentice Hall Professional, 2003. Citado na página 39.
- PATIL S. S.; DAVANDE, V. M. M. J. J. Smart wireless sensor network for monitoring an agricultural environment. (*IJCSIT*) *International Journal of Computer Science and Information Technologies*, v. 5(3), p. 3487–3490, 2014. Citado 3 vezes nas páginas 15, 30 e 31.
- PERROTT, M. H. High speed frequency dividers. *6.976 High Speed Communication Circuits and Systems. Massachusetts Institute of Technology*. Disponível em <<http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-976-high-speed-communication-circuits-and-systems-spring-2003/lecture-notes/lec14.pdf>> Acessado em 17/11/2014, 2003. Citado 2 vezes nas páginas 16 e 67.
- PERROTT, M. H.; TROTT, M. D.; SODINI, C. G. A modeling approach for σ - δ fractional-n frequency synthesizers allowing straightforward noise analysis. *Solid-State Circuits, IEEE Journal of, IEEE*, v. 37, n. 8, p. 1028–1038, 2002. Citado 2 vezes nas páginas 15 e 56.
- PLASSCHE, R. J. van de; HUIJSING, J.; SANSEN, W. *Analog Circuit Design: High-Speed Analog-to-Digital Converters, Mixed Signal Design; PLLs and Synthesizers*. [S.l.]: Springer, 2000. Citado na página 48.
- RAZAVI, B. *Design of analog CMOS integrated circuits*. [S.l.]: Tata McGraw-Hill Education, 2002. Citado 2 vezes nas páginas 15 e 63.
- RAZAVI, B. *RF microelectronics*. [S.l.]: Prentice Hall New Jersey, 2012. Citado 8 vezes nas páginas 15, 52, 54, 57, 59, 61, 62 e 76.
- RAZAVI, B.; BEHZAD, R. *RF microelectronics*. [S.l.]: Prentice Hall New Jersey, 1998. Citado na página 55.
- ROGERS, J. W. M.; DAI, F. F.; OLETT, C. *Frequency Synthesis for Multiband Wireless Networks*. In: Iniewski, K. “Wireless Technologies: Circuits, Systems, and Devices”. [S.l.]: CRC Press: Taylor and Francis Group, 2009. Citado 5 vezes nas páginas 15, 52, 53, 54 e 56.
- ROMERO, R. et al. Research on automatic irrigation control: State of the art and recent results. *Agricultural Water Management*, Elsevier, v. 114, p. 59–66, 2012. Citado na página 29.
- SANTOS, S. d. A. *Sintetizador de frequências de 2, 4 GHz em CMOS, 0, 35 μ m para aplicações em ZigBee*. Tese (Doutorado) — Universidade de São Paulo, 2008. Citado 2 vezes nas páginas 15 e 60.

- TANG, Y. et al. A fully integrated dual-mode frequency synthesizer for gsm and wideband cdma in $0.5\mu\text{m}$ cmos. In: IEEE. *Circuits and Systems, 2001. MWSCAS 2001. Proceedings of the 44th IEEE 2001 Midwest Symposium on*. [S.l.], 2001. v. 2, p. 866–869. Citado na página 80.
- TERUEL, B. et al. Desempenho de rede de sensores sem fio em casa de vegetação. *R. Bras. Eng. Agríc. Ambiental*, SciELO Brasil, v. 16, n. 12, p. 1374–1380, 2012. Citado na página 29.
- WALBERT, A. Agricultura é quem mais gasta água no brasil e no mundo. Portal EBC. Disponível em: <http://www.ebc.com.br/noticias/internacional/2013/03/agricultura-e-quem-mais-gasta-agua-no-brasil-e-no-mundo>. Acessado em: setembro de 2014., 2006. Citado na página 30.
- WANG, W.; HE, G.; WAN, J. Research on zigbee wireless communication technology. In: IEEE. *Electrical and Control Engineering (ICECE), 2011 International Conference on*. [S.l.], 2011. p. 1245–1249. Citado na página 46.
- WESTE, N.; HARRIS, D. Cmos vlsi design. *A Circuits and Systems perspective*, Pearson Addison Wesley, 2005. Citado na página 66.
- YUAN, J.; SVENSSON, C. High-speed cmos circuit technique. *Solid-State Circuits, IEEE Journal of*, IEEE, v. 24, n. 1, p. 62–70, 1989. Citado na página 69.
- ZHENHUA, S.; ZHIQUN, L. Design of a low-power high-speed cmos frequency divider for wsn applications. In: IEEE. *Communication Technology (ICCT), 2011 IEEE 13th International Conference on*. [S.l.], 2011. p. 1091–1094. Citado na página 59.

Apêndices

APÊNDICE A – Portas Lógicas CMOS

Figuras das partas lógicas citadas na [subseção 7.1.1](#).

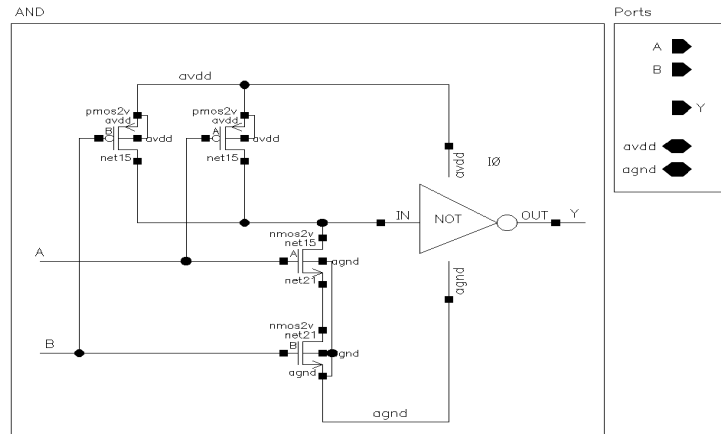


Figura 63 – Esquemático de uma porta AND de duas entradas.

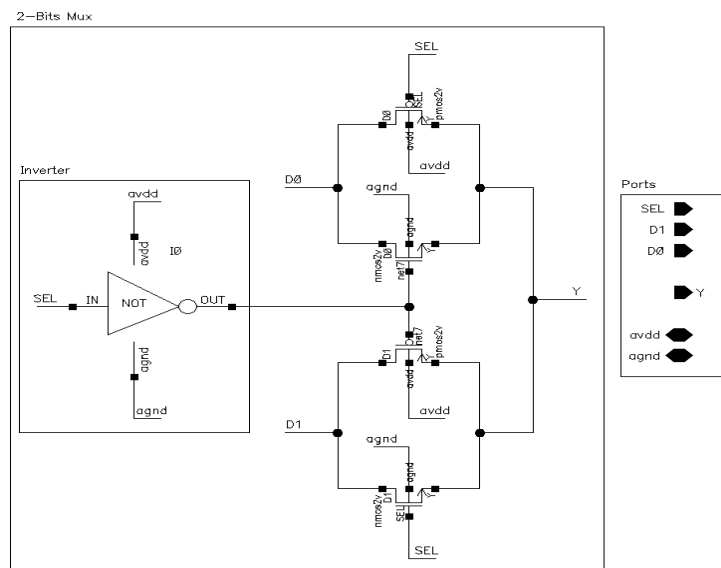


Figura 64 – Esquemático do multiplexados de 2 bits utilizado na lógica de controle do divisor completo.

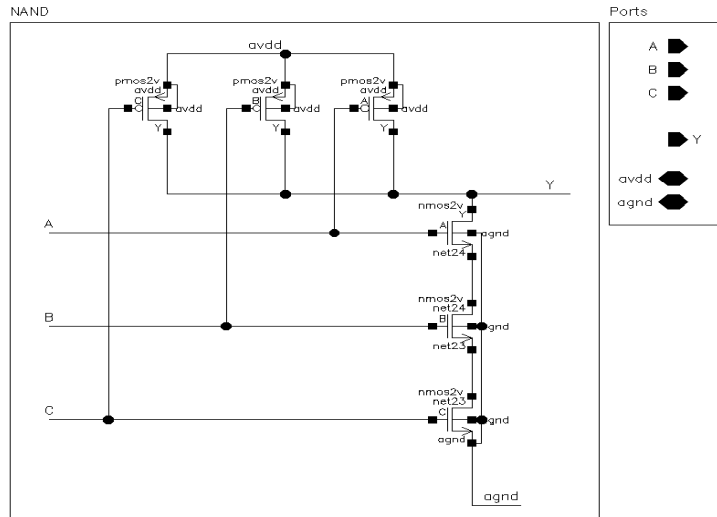


Figura 65 – Esquemático de uma porta NAND de três entradas.

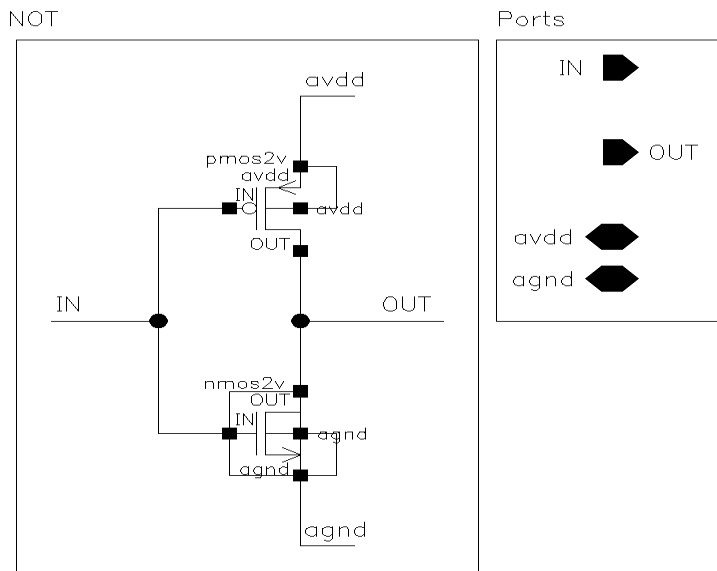


Figura 66 – Esquemático de uma porta NOT.

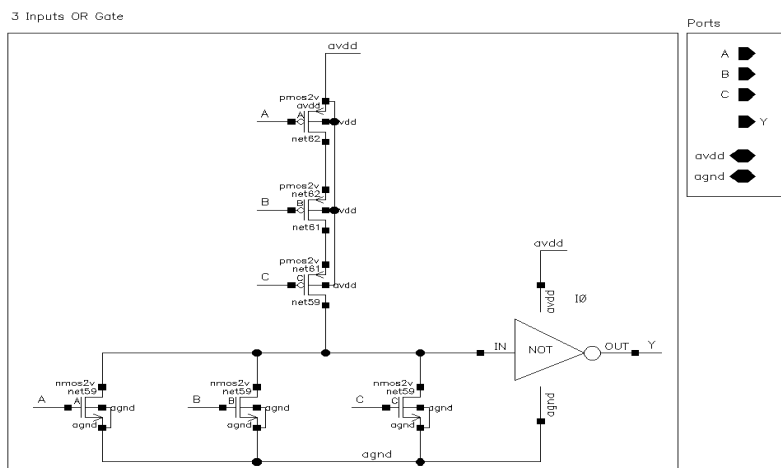


Figura 68 – Esquemático de uma porta OR de três entradas.

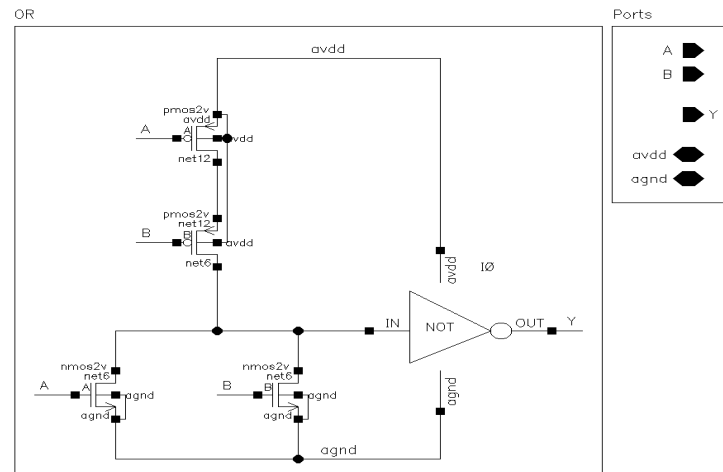


Figura 67 – Esquemático de uma porta OR de duas entradas.

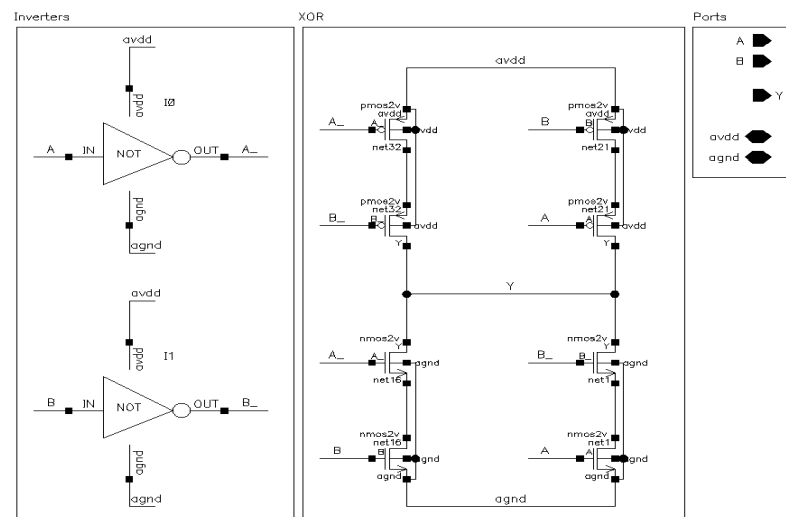


Figura 69 – Esquemático de uma porta XNOR de duas entradas.

Anexos

ANEXO A – Fonte de Corrente

O esquemático da fonte de corrente utilizada no projeto do conversor diferencial para *single-ended* pode ser visualizada por meio da figura 70. Esta fonte foi projetada para apresentar uma pequena variação na corrente de saída quando opera em um sistema com grandes alterações na temperatura.

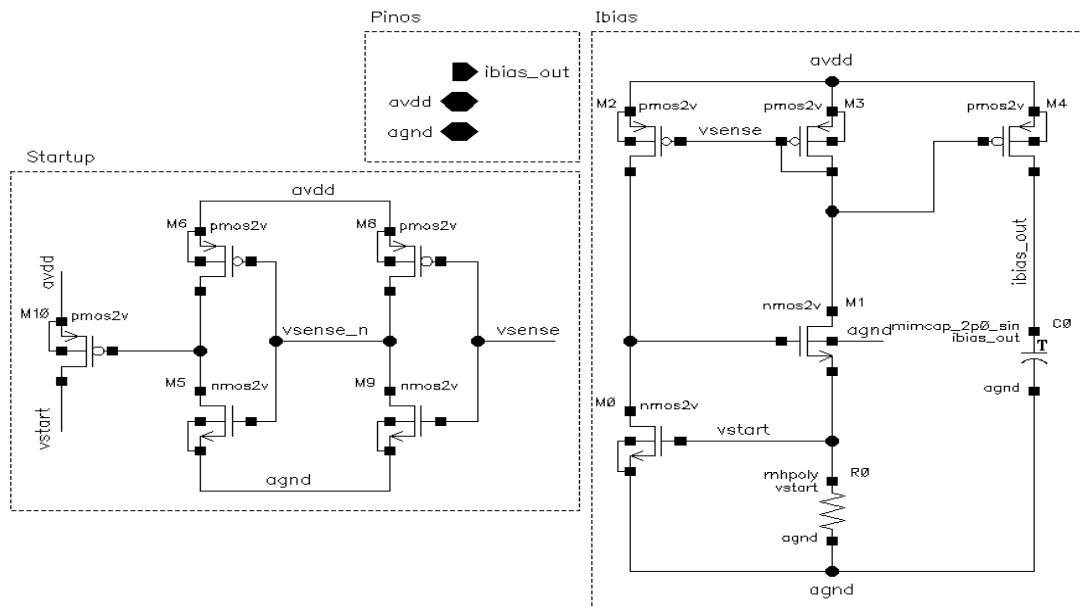


Figura 70 – Esquemático da fonte de corrente utilizada no conversor diferencial para *single-ended*.