UNESP Universidade Estadual Paulista

Faculdade de Engenharia de Ilha Solteira

Departamento de Engenharia Elétrica

Programa de Pós-Graduação em Engenharia Elétrica

Tese de Doutorado:

Desenvolvimento e Implementação de um Sintetizador de Frequência CMOS utilizando Sistema Digital

Adriano dos Santos Cardoso

Orientador: Prof. Dr. Nobuo Oki

Ilha Solteira
Novembro/2009

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Desenvolvimento e Implementação de um Sintetizador de Freqüência CMOS utilizando Sistema Digital

ADRIANO DOS SANTOS CARDOSO

Orientador: Prof. Dr. Nobuo Oki

Tese apresentada à Faculdade de Engenharia - UNESP – Campus de Ilha Solteira, para obtenção do título de Doutor em Engenharia Elétrica. Área de Concentração: Automação.

Ilha Solteira – SP
Novembro/2009

FICHA CATALOGRÁFICA

Elaborada pela Seção Técnica de Aquisição e Tratamento da Informação
Serviço Técnico de Biblioteca e Documentação da UNESP - Ilha Solteira.

Cardoso, Adriano dos Santos.
C179d Desenvolvimento e implementação de um sintetizador de frequência CMOS utilizando sistema digital / Adriano dos Santos Cardoso. -- Ilha Solteira : [s.n.], 2009.
77 f. : il.

Tese (doutorado) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira. Área de conhecimento: Automação, 2009

Orientador: Nobuo Oki
Bibliografia: p. 75-77

l. Sintetizador de frequência. 2. PLL. 3. DLL. 4. CMOS.

unesp **UNIVERSIDADE ESTADUAL PAULISTA**

CAMPUS DE ILHA SOLTEIRA

FACULDADE DE ENGENHARIA DE ILHA SOLTEIRA

CERTIFICADO DE APROVAÇÃO

TÍTULO: Desenvolvimento e Implementação de um Sintetizador de Frequência CMOS utilizando Sistema Digital

AUTOR: ADRIANO DOS SANTOS CARDOSO
ORIENTADOR: Prof. Dr. NOBUO OKI

Aprovado como parte das exigências para obtenção do Título de DOUTOR em ENGENHARIA ELÉTRICA , Área: AUTOMAÇÃO, pela Comissão Examinadora:

Prof. Dr. NOBUO OKI
Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira

Prof. Dr. CARLOS ANTONIO ALVES
Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira

Prof. Dr. AILTON AKIRA SHINODA
Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira

Prof. Dr. JOSÉ RAIMUNDO DE OLIVEIRA
Departamento de Engenharia de Computação e Automação Industrial / Universidade Estadual de Campinas

Prof. Dr. JOSÉ RICARDO DESCARDECI
Departamento das Engenharias / Universidade Federal do Tocantins

Data da realização: 25 de novembro de 2009.

A minha querida esposa Simone
e minha filha Nicole
por todo apoio e dedicação
Incondicional.

AGRADECIMENTOS

Gostaria de agradecer todos que me apoiaram neste árduo trabalho.

Ao CNPq pelo apoio financeiro.

Ao professor Nobuo Oki que sempre depositou confiança e me ajudou nos momentos de incerteza.

Aos colegas do departamento pelas momentos de ajuda e discussão de idéias que me ajudaram no desenvolvimento do trabalho.

Aos professores que de alguma forma contribuíram com minha formação.

Aos meus pais Manoel e Agnalda pelo apoio em todos os momentos.

A Simone e minha filha Nicole que sempre estiveram ao meu lado.

A Deus.

RESUMO

Sintetizadores de frequência são circuitos críticos usados largamente em muitas aplicações de temporização. Circuitos PLL apresentam uma boa solução para temporização, mas utilizam geralmente blocos analógicos que são facilmente influenciados em desempenho devidos a instabilidades inerentes aos processos de fabricação e ruídos. Com a evolução dos circuitos e ferramentas para sistemas digitais foi possível a implementação de circuitos que utilizem somente recursos digitais tais como os DLL. Um dos papéis dos sintetizadores é equalizar a fase de um sinal de clock em relação a uma segunda referência adicionando fase entre os sinais. Este trabalho tem como objetivo o desenvolvimento de um circuito DLL com arquitetura flexível e programável para utilização no ajuste de fase e recuperação de sinais. Os blocos digitais foram implementados utilizando ferramentas de alto nível de abstração para avaliação do comportamento funcional. O objetivo final é a implementação do circuito validado em tecnologia CMOS 350 nm da AMS.

Palavras Chave: PLL. DLL. CMOS.

ABSTRACT

Frequency Synthesizers are critical circuits widely used in timing applications. PLLs devices had showed a good solution for timing, but they normally because the use analog building blocks that are often influenced by the subtract building process and noises. Nevertheless, after the evolution of complex circuits and development tools it had been possible the implementation of systems that implement only digital resource such as DLL. One of major goals of synthesizers is to equalize the phase between a clock signal and a second reference. This work aims to develop DLL devices that are built in a flexible and reprogrammable architecture for using in decrements or increments in the phase and clock recovery. Digital blocks were implemented using high level abstraction tools for analysis of functional behavior. The main objective is the circuit implementation and validations in CMOS .35 AMS process.

Keywords: CMOS. PLL. DLL.

LISTA DE SÍMBOLOS E SIGLAS

AMS	Austria Micro Systems
CP	Charge Pump – (Bomba de Carga)
CPLD	Complex Programmable Logic Device
CMOS	Complementary Metal Oxide Semiconductor (Semicondutor de Metal Oxido Complementar)
D/A	Conversor Digital Analógico
DLL	Delay Locked Loop (Laço de Sincronismo de Atraso)
Divider	Bloco lógico contador para aumentar a frequência do PLL
Edge Combiner	Bloco lógico que implementa multiplos de frequência em um VCDL
EDIF	Eletronic Design Interchange Format – Arquivo de transferência de projetos digitais
F_{out}	Frequência de Saída do Sistema
FPGA	Field Programmable Gate Array
F_{ref}	Frequência de Referência
Jitter	Ruído de oscilação no domínio do tempo
Loop Filter	Filtro do Laço
LUT	Look Up Table –Forma de Onda implementada em dados de memória
MULTIPLIER	Multiplicador escalar de contagem
PLL	Phase Locked Loop
PFD	Phase Frequency Detector – Detector de Fase que usa máquina de estados finitos
VCDL	Voltage Controled Delay Line (Tensão controlada por linha de atraso)

VCO	Voltage Controlled Oscillator
VHDL	Very High Speed Integrated Circuits Hardware Description Language
W / L	Largura e comprimento das dimensões dos transistores
XOR	Porta Lógica Exclusive Or

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

Circuitos sintetizadores apresentam grande empregabilidade em sistemas de comunicação, sistemas de dados e reconhecimento e recuperação de sinais de clock. Para tantas aplicações, diversas são as arquiteturas que têm sido exploradas no intuito do desenvolvimento de circuitos eficientes em termos de baixo ruído, rápido tempo de resposta e estabilização. Circuitos digitais CMOS podem apresentar boas respostas a estes problemas devido a sua grande capacidade de integração e utilização de blocos lógicos e conseguir, mesmo utilizando que um grande número de componentes, um reduzido consumo de energia.

Este trabalho tem como objetivo o estudo, desenvolvimento e validação de circuitos DLL para recuperação de sinais de clock.

No capítulo 2 são apresentadas classificações e especificações das principais famílias de sintetizadores de frequência. No capítulo 3 tem-se algumas das características do modelamento dos sintetizadores PLL e DLL. No capítulo 4 são apresentadas algumas das principais características de desempenho dos sintetizadores. No capítulo 5 a descrição dos principais blocos lógicos dos sintetizadores de frequência. O capítulo 6 trata da descrição do funcionamento e principais aspectos da montagem do DLL. No capítulo 7 é mostrado os resultados da implementação do DLL proposto. E no capítulo 8 tem-se as conclusões e discussões e a seguir os trabalhos futuros.

2 SINTETIZADORES DE FREQUÊNCIA E APLICAÇÕES

Os objetivos deste trabalho foram o estudo dos sintetizadores de frequência e a implementação um circuito DLL de topologia simples para poder ser incorporado à sistemas diversos, tais como recuperação de sinais de clock ou incorporado a memórias de forma a auxiliar a recuperação mais rápida da informação. O DLL é usado como ferramenta de auxilio na recuperação e estabilização de sinais de clock ou estabelecimento de sinais, fornecendo quando necessário acréscimo da fase para que as interfaces dos sistemas possam trabalhar mais rapidamente.

Neste capítulo será feito uma introdução em relação às principais topologias de sintetizadores de frequência e suas principais características construtivas e de desempenho.

2.1 USO DE TECNOLOGIA CMOS

Os projetos de circuitos necessitam da implementação de centenas de milhares de elementos como transistores, resistores e capacitores, recebendo a denominação de circuitos VLSI (*Very Large Scale Integrated)*. Para que ocorra a implementação destes projetos então é necessário que todos os elementos possuam as características de grande capacidade de integração, ou seja, cada elemento possa ser construído em dimensões bem reduzidas, baixo consumo de energia, grande imunidade ao ruído. Agrupando todas essas características, a tecnologia disponível atualmente que melhor se adequada a isto é a CMOS (*Complementary Metal Oxide Semiconductor*). A partir de um substrato dopado tipo N, são adicionados os elementos que formam transistores, resistores por meio de polisilicio, células de guarda, capacitores por meio de metais (algumas implementações multi-camadas) e até mesmo indutores. Essa flexibilidade também agrega a habilidade do circuito de possuir poucos

elementos externos, o que implica menor custo de implementação e maior integração de todo o sistema. As vantagens de implementação de projetos em circuito CMOS possibilitam que esta seja uma opção bastante viável e interessante na implementação de sintetizadores de frequência (LEE, 1988).

Com o desenvolvimento de novas tecnologias digitais e analógicas e a implantação de novos protocolos de comunicação surge à necessidade de pesquisa de sistemas que possam gerar sinais ou criar meios de propagação desses sinais de forma a estar adequado aos parâmetros de desempenho das novas tecnologias.

Sintetizadores de frequência são circuitos capazes de gerar um ou mais sinais a partir de uma ou várias fontes de referência. A saída de um sintetizador é caracterizada por sua frequência de operação, resolução e sua pureza (ou relação sinal ruído). Diversas são as formas de se implementar os sintetizadores utilizando recursos digitais, analógicos, métodos diretos e indiretos e uma combinação de vários destes. A Figura 2.1 ilustra os principais tipos de sintetizadores utilizados e sua nomenclatura atual (JIA, 2005).

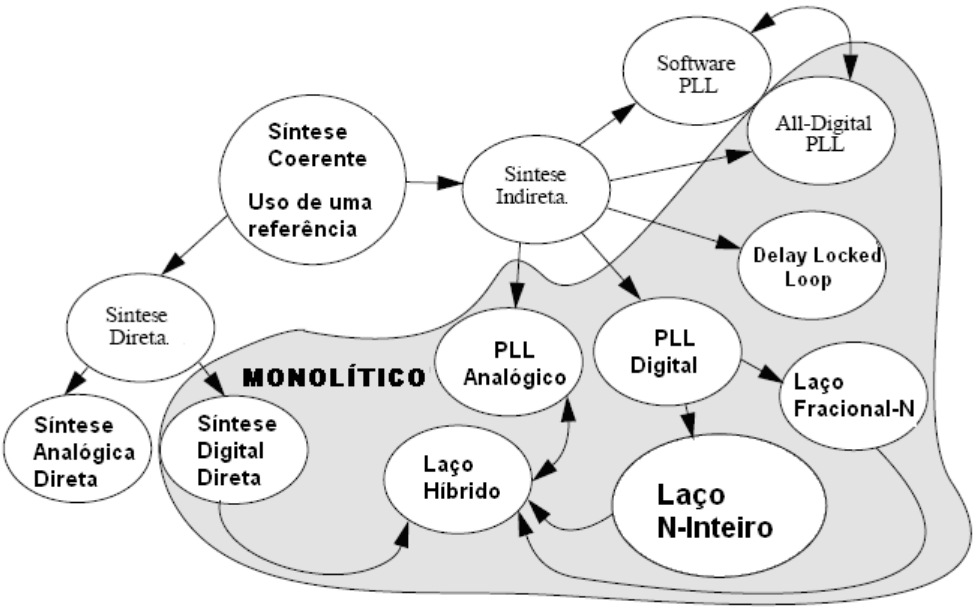

Figura 2.1. Famílias de Sintetizadores.

2.2 SINTETIZADOR DE FREQUÊNCIA USANDO MÉTODO DIRETO

O método direto é um método bastante simples de ser implementado por não necessitar de retroalimentação. O princípio da sintetização de forma direta é a teoria de amostragem de Nyquist, que garante que um sinal limitado em largura de banda pode ser reconstituído fazendo-se uma amostragem de pelo menos duas vezes a maior componente de frequência do sinal a ser reconstituído.

Um sinal de referência modulado por uma palavra de controle passa pelo acumulador de fase, que determina uma posição a ser localizada no LUT (*Look Up Table)*. Para a reconstrução do sinal é utilizado um conversor D/A e finalmente havendo necessidade, um filtro tipo passa baixa para fazer a seleção do sinal desejado e remover alguns sinais espúrios. Por fim, utilizando quatro principais blocos é possível fazer a reconstituição do sinal conforme a Figura 2.2.

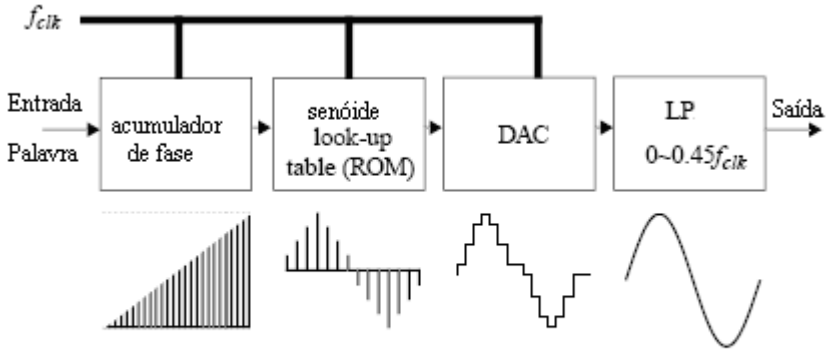

Figura 2.2. Sintetizador de Frequência Direto.

Apesar do circuito conversor D/A ser bastante simples, a dificuldade de implementação está na habilidade e conhecimento do aperfeiçoamento das LUT porque quando existe a necessidade de precisão do sinal, ou ainda uma resolução em um número grande de bits, por exemplo, maior que oito, a tabela fica grande e o circuito se torna complexo (grande número de elementos). Isto pode exigir a necessidade de uma compressão desta LUT. Devido a este problema, equações são implementadas de forma que esta LUT possa ser sintetizada com menor número de posições. Muitas técnicas matemáticas têm sido então empregadas de forma a gerar circuitos de sintetização direta (SILVA, 2001).

2.3 SINTETIZADORES DE FREQUÊNCIA EM LAÇO PLL

Uma grande variedade dos sintetizadores é desenvolvida em sistema com retroalimentação de forma a criar uma correção do sinal de saída em relação ao sinal original como apresentado na Figura 2.3. Também são mostrados os principais blocos de um sintetizador de sincronismo que de modo geral são chamados sintetizadores de Phase Locked Loop (sintetizadores de sincronismo de fase). Os principais blocos deste sintetizador são detector de fase, *charge pump, Loop Filter*, VCO e opcionalmente incluso um divisor.

Estes dispositivos são chamados de sistemas em laço (*Loop*) que podem ser digitais ou analógicos. Uma característica comum a estes dispositivos é o estado em sincronismo (*in lock*) que ocorre quando o sinal de referência é tal que o sinal gerado não produz mais diferença em relação à referência. Então, o sinal do sistema entra em uma nova condição oscilando em torno do próprio sinal de referência.

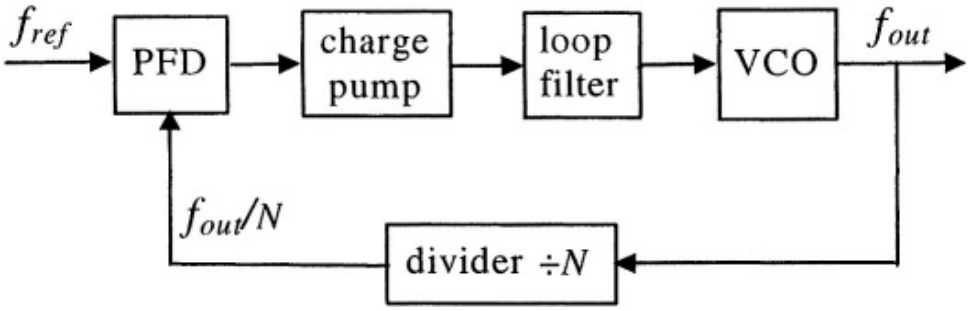

Figura 2.3. Blocos Lógicos PLL.

Várias arquiteturas baseadas nesses blocos principais são permitidas o que possibilita a empregabilidade em varias aplicações tais como comunicação de dados, sintonizadores e sintetizadores de frequência, recuperação de sinal, estabilização de sinal e diminuição de sinais espúrios e ou redução de *jitter* [KIM 90] (CRANINCKX, 1998). O problema *jitter* será tratado mais adiante.

O funcionamento em linhas gerais do circuito PLL consiste em: comparação do sinal de referência com a saída do divisor (ou da frequência de saída F_{ref}); aplicação da saída do comparador (detector de frequência) de um sistema *charge pump* que converte a saída do detector, que é tipicamente diferencial, em um sinal de saída única; na seqüência aplica-se um

filtro que adequa o sinal de acordo com os parâmetros de banda de frequência em que se quer trabalhar; e finalmente o VCO oscila de acordo com o sinal aplicado na entrada, gerando uma frequência de saída proporcional ao sinal gerado no *Loop Filter*. O bloco opcional *Divider* proporciona uma seleção de frequência programável fazendo com que o sinal de saída seja um múltiplo do sinal de referência de entrada e inversamente proporcional ao número do *Divider*. O *Loop Filter* faz o papel de seleção de faixa de frequência e em circuitos mais simples os componentes são capacitores e resistores formando um filtro passa baixa ou passa faixa de segunda ou terceira ordem. Circuitos auxiliares ou opcionais podem substituir o capacitor por um amplificador operacional associado a um capacitor de valor e área menor no intuído de diminuir a área envolvida. Outra solução que pode substituir o capacitor é a utilização de um circuito contador crescente / decrescente.

O circuito VCO (*Voltage Controlled Oscillator)* é um oscilador que responde de acordo com um sinal de tensão de entrada e na saída produz uma variação da frequência de oscilação proporcional a este sinal. Desta forma o VCO é o cerne dos sintetizadores de frequência baseados em sistema em laço. Nos próximos capítulos serão discutidos as vantagens e problemas relacionados à arquitetura que utiliza os circuitos VCO. Recentes pesquisas têm desenvolvido blocos em substituição ao VCO implementando um VCO digital chamado de DCO (STASZWSKI, 2005). Estas implementações têm por objetivo a utilização em um sistema que não utilize circuitos analógicos. O que também pode ser chamado de ADPLL (*All Digital PLL*).

2.4 SINTETIZADOR DE FREQUÊNCIA DLL

Uma família importante de sintetizadores são os circuitos DLL – *delay locked loop*. Muito parecidos com os PLL, este é constituído por um VCDL em substituição ao circuito VCO de um típico PLL. As principais vantagens de um DLL em relação ao PLL são a de ter estabilidade incondicional por não possuir zeros em seu modelamento linearizado e a de não acumular *jitter* em seus estágios do VCDL. Esses aspectos serão abordados no capítulo 3. A Figura 2.4 ilustra os principais blocos de um DLL.

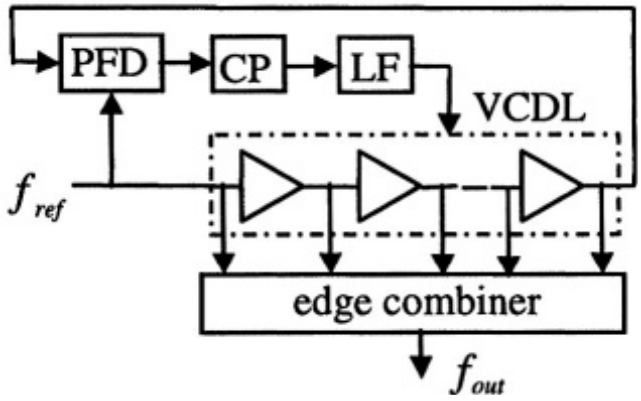

Figura 2.4. Blocos Básicos de um DLL.

O princípio inerente de um DLL é a utilização de um sistema baseado em atrasadores. Muitas vezes são utilizados um grande arranjo de atrasadores chamados de VCDL (*Voltage Controlled Delay Line*), que são controlados de forma parecida ao VCO de um PLL. Apesar da diferença ser apenas de um componente, as vantagens do DLL é de permitir que muitos blocos ou mesmo todos sejam feitos de forma digital ou com elementos puramente digitais pois não são necessários ajustes no *Loop Filter* como é feito no PLL. Isto decorre do princípio de estabilidade que ocorre devido à características próprias do componente VCDL que não possui pólos (KELIU, 2005). Por fim o DLL também possui como característica a utilização do circuito *Edge Combiner* que tem por objetivo gerar funções ou sinais que são combinação dos sinais gerados entre os atrasos dos atrasadores. Assim uma infinidade de combinações pode ser implementada de acordo com a intenção do sinal a ser gerado. Outros blocos lógicos podem ser acrescentados a arquitetura geral do DLL tais como: Controle de Ciclo Ativo e Detector de Sincronismo. O Controle de Ciclo Ativo recebe os sinais do *Edge Ccombiner* e faz a restituição do sinal de forma que o sinal de saída tenha então 50% de ciclo ativo. Isto é necessário porque os sinais do *Edge Combiner* geralmente têm menos de 25% de ciclo ativo. O detector de sincronismo permite que o DLL entenda que o sistema entrou em regime, evitando que o sistema entre em sincronismo em uma frequência sub-múltipla da frequência desejada.

De forma geral, muitas são as maneiras de se construir sintetizadores de frequência de acordo com as necessidades de aplicação. Os principais métodos de sintetizadores são descritos entre sintetizadores de forma direta analógico e digital, PLL-N inteiro, PLL-N fracional, DLL e sistemas híbridos.

Fazendo um resumo das principais famílias de sintetizadores tem-se que: os sintetizadores direto analógicos apresentam rápido chaveamento, baixo ruído e poucos sinais espúrios, mas apresentam grande complexidade e consumo (SHU, 2005). Os sintetizadores

Direto Digital apresentam também baixo ruído e grande consumo e grande complexidade (APEEL, 2000), os PLL do tipo N-inteiro apresentam demorado tempo de chaveamento (KELIU; SINENCIO, 2003) e os PLL-N fracionais geram sinais espúrios (ZARKESHVARI, 2000). Os circuitos sintetizadores DLL apresentam baixos sinais espúrios, mas podem ser complexos e geralmente tem o consumo alto. Sistemas híbridos são os sistemas que aproveitam duas ou mais arquiteturas para que em conjunto possam apresentar um sistema de sintetizador de frequência que possa unir as melhores características de cada um (OLIVEIRA 2008).

No próximo capítulo serão apresentadas as principais características dos modelamentos próprios dos circuitos dos sintetizadores de frequência PLL e DLL bem como seus principais problemas construtivos.

3 MODELAMENTO MATEMÁTICO DE SINTETIZADORES DE FREQUÊNCIA

Para se estudar as principais influências de ruídos e desenvolver a sistematização do desenvolvimento dos sintetizadores são feitos os modelamentos lineares dos blocos lógicos dos sintetizadores. Isto facilita a determinação dos parâmetros dos circuitos e também permite entender melhor o funcionamento dos blocos levando em consideração que a linearização também não resolve todos os problemas de modelamento.

3.1 MODELO LINEAR

O método de linearização dos sintetizadores de frequência permite obter os parâmetros de projeto bem como na implementação de novas propostas de solução. A Figura 3.5 ilustra um modelo linearizado para um PLL.

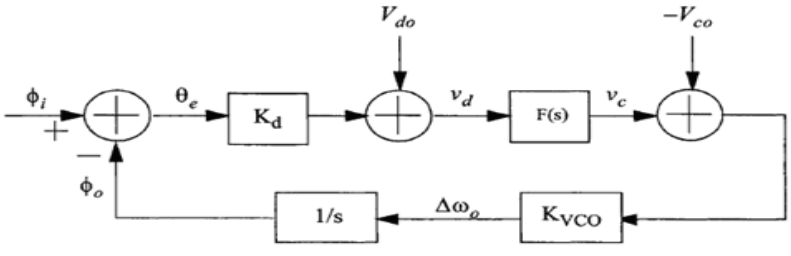

Figura 3.5. PLL Linearizado.

Como pode ser visto na Figura 3.5 a diferença de fase medida pelo Detector de Fase ($\Phi_i - \Phi_o = \theta_e$) é aplicado ao ganho do detector ($K_{d)}$ e também ao ganho do *Loop Filter* (*F(s)*).

Fontes de ruído são modeladas como Vn no *Loop Filter* e Vco para o VCO. Devido ao fato do ganho do VCO (*Kvco/s*) possuir pólos na origem o sistema como um todo é classificado como instável o que implica que o *Loop Filter* precisa compensar esse pólo. Além de compensar o pólo do VCO o *Loop Filter* faz o papel de estabelecer a largura de banda que o sistema deve trabalhar.

O ganho do K_d do detector de fase depende naturalmente do tipo de construção. Quando circuito é feito de uma porta lógica o ganho é menor do que os que são feitos utilizando máquinas de estados finitos por fazerem a detecção em todo o ciclo como será demonstrado no próximo capítulo.

Em outros termos, o sinal detectado pelo Detector de Fase é submetido a uma função de transferência que é o *loop filter*. Desta forma estabelece o sinal que o VCO atuará para gerar a oscilação.

Existem diversos tipos de detectores que podem ser desenvolvidos em arquiteturas complexas para contornarem os principais problemas que são o *Dead Zone*, baixo ganho, ganho não linear e erros de detecção. O detector de fase será estudado no próximo capítulo.

Da mesma maneira que o detector de fase possibilita inúmeras construções diferentes, o *Loop Filter* e mesmo o VCO permitem alternativas de construção que podem priorizar redução da área, menor tempo de sincronismo, facilidade de se atingir estabilidade do sistema e redução de *Jitter* e de sinais espúrios.

A implementação mais comum do *Loop Filter* é um filtro passa baixa implementado com resistores e capacitores. A partir deste filtro são feitos os equacionamentos e diagramas e análises. Na Figura 3.6 tem-se o diagrama esquemático do *Loop Filte*r. Como entrada tem-se o I_{cp} que é a corrente que é gerada no *charge pump* e como saída o V_c que é a tensão de referência que controla o VCO.

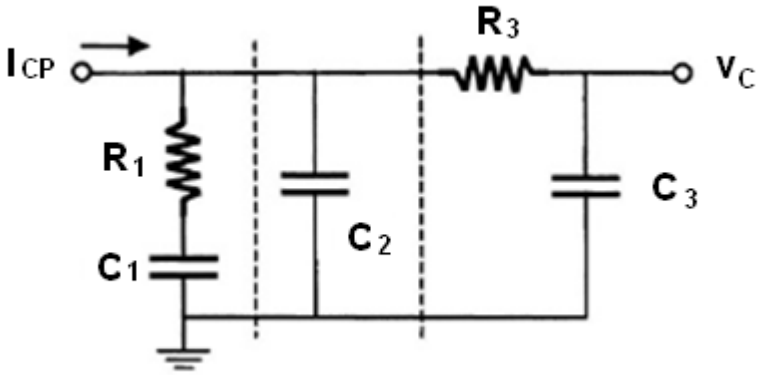

Figura 3.6. *Loop filter* implementado com resistores e capacitores.

No modelamento do DLL têm-se praticamente as mesmas considerações e configurações usadas no PLL, ou seja o *Loop filter* , o detector de fase e o *charge pump.* No caso do *loop filter* é utilizado apenas um capacitor. O VCDL não acumula ganhos no tempo o que implica que seu modelo não possui zeros ou pólos, introduzindo apenas um ganho e uma variação de fase ao sistema. Assim, em uma análise geral em que não seja incluída as não linearidades, considera-se que o DLL seja estável para sua faixa de operação de frequência.

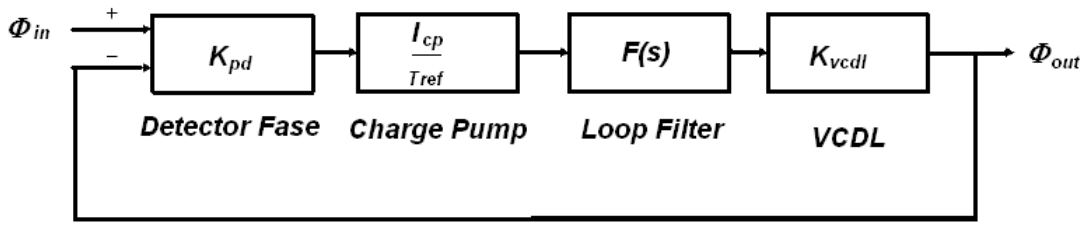

Figura 3.7. Modelamento Linear para DLL.

3.2 RUÍDO DE FASE DE JITTER

Aplicações que requerem precisão no sincronismo de fase tornam as pesquisas por osciladores e geradores de sinais com baixo ruído muito atraentes e com grau de exigência alto. O ruído de fase (*phase noise*) é caracterizado por variações aleatórias na frequência, fase e deformações na forma de onda de saída de osciladores, sendo quantificado pela magnitude do ruído de fase ou *jitter*. O *jitter* pode ser interpretado como a variação da fase de um sinal periódico no tempo (RAZAVI, 2000). O período da n-ésima oscilação chama-se T_n e Δt como sendo o intervalo entre dois intervalos de tempo.

Existem varias definições de *jitter* na literatura (MAGIEROWSKI, 2000; RAZAVI, 1996; RAZAVI 1998). A mais comum é a que se define o *jitter* Absoluto ou também chamado de *jitter* de Longo Termo que é definido como:

$$\Delta T_{abs}(N) = \sum_{n=1}^{N} \Delta T_n \qquad (1)$$

O *jitter* absoluto representa a acumulação do *jitter* nos primeiros N ciclos. A segunda definição de *jitter* é chamada de *jitter* de longa duração em termos RMS, ou *jitter* cíclico. Assim,. A definição de *jitter* cíclico é:

$$\Delta T_c = \lim_{N \longrightarrow \infty} \sqrt{\frac{1}{N} \sum_{n=1}^{N} \Delta T_n^2} \tag{2}$$

O *jitter* cíclico representa a media de longa duração do efeito de flutuação do sinal de referência (F_{ref}).

Outra definição para o *jitter* absoluto é o valor medido em fontes de ruído branco relacionado com o *jitter* cíclico sendo f_o a frequência do sinal, então se tem:

$$\Delta T_{ABS}(\Delta t) = \sqrt{\frac{f_o}{2}} \Delta T_c \sqrt{\Delta t} \tag{3}$$

As equações de medição do *jitter* são importante pois o valor do *jitter* é um dos principais parâmetros de desempenho dos sintetizadores.

A Figura 3.8 ilustra o efeito da oscilação *jitter* acumulada ao longo dos períodos.

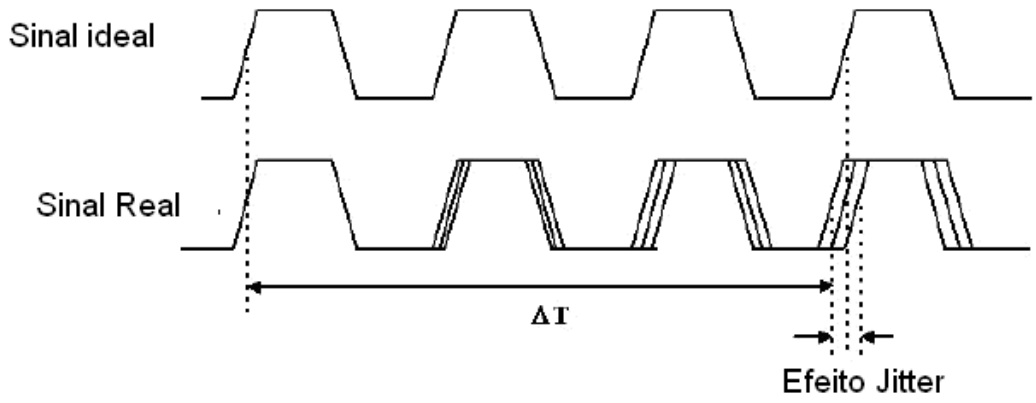

Figura 3.8. Efeito *jitter* na geração de sinal.

Para melhor avaliação das características de desempenho bem como ajustes dos parâmetros do circuito é necessário o uso de ferramentas matemáticas para estimar os parâmetros

como o MatLab ou fazer simulações dos desempenhos de velocidade e estabilização dos circuitos. A ferramenta MatLab auxilia no desenvolvimento dos parâmetros dos modelos lineares e ferramentas como CADENCE, Mentor e Max + Plus II da Altera facilitam a implementação dos circuitos analógicos ou digitais. Alguns parâmetros porém, somente podem ser avaliados após a confecção do leiaute devido ao fato de existir dificuldade ou inexistência de ferramentas para avaliação de *jitter,* consumo de energia e até mesmo o tamanho do circuito. No próximo capítulo são apresentadas algumas características importantes na avaliação e concepção dos circuitos sintetizadores de frequência.

4 CARACTERÍSTICAS DE DESEMPENHO DE SINTETIZADORES

O desempenho dos sintetizadores deve estar de acordo com os objetivos do sinal em relação à amplitude da portadora, sinais espúrios, ruídos de fase, rejeição de ruído e tempo de estabelecimento do sinal. Portanto, uma revisão dos principais conceitos de sintetizadores é desenvolvida neste capítulo.

4.1 RUÍDO DE FASE

O ruído de fase é a diferença que se pode medir entre o sinal ideal e o sinal real obtido em relação ao desempenho em frequência, ou espalhamento da frequência do sinal. Em outros termos é o efeito *jitter* no domínio da frequência. Um dos objetivos de um sintetizador de frequência é que tanto o *jitter* como o ruído de fase sejam os menores possíveis visando criar um sinal o mais puro e próximo do sinal ideal.

A Figura 4.9 mostra o efeito de ruídos de fase aplicados sobre um sinal ideal e o seu novo espectro.

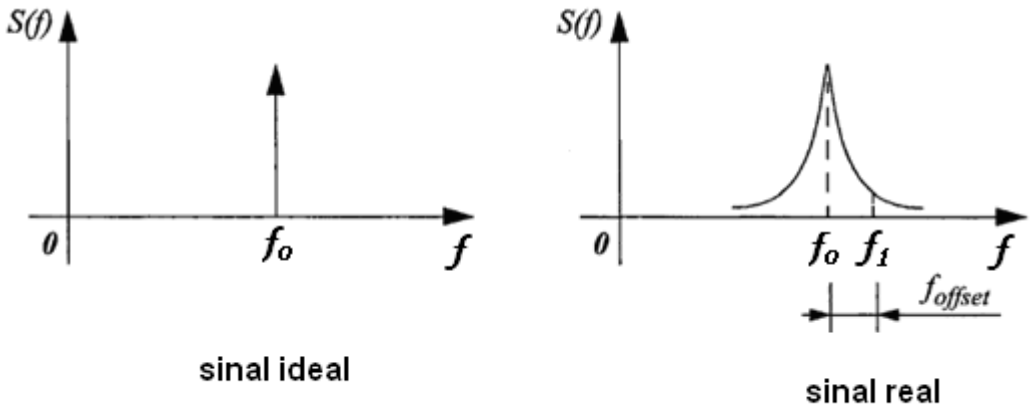

Figura 4.9. Ruído de Fase aplicado em um Sinal Ideal.

4.2 TONS ESPÚRIOS

Os tons chamados espúrios são aqueles que não são desejados. Aparecem principalmente em ambos os lados da portadora e tem amplitude significativa. Estes tons são gerados de forma não intencional devido a não linearidades dos blocos dos sistemas e também por problemas nos chaveamentos e na estabilização do sinal. Outra fonte importante de sinais espúrios é o próprio PLL quando o mesmo utiliza o bloco *Divider* para selecionar os canais de operação. Aa Figura 4.10 mostra um sinal real com frequência estabelecida em torno de f_o e seus tons espúrios em ambos os lados da banda.

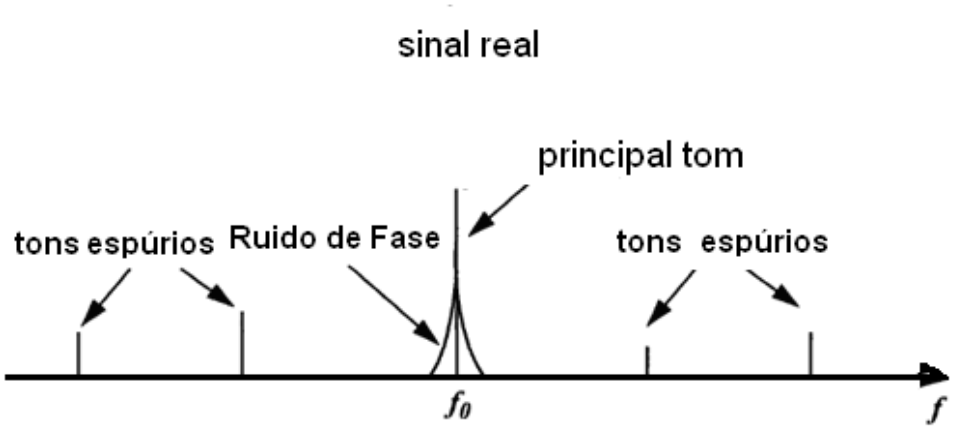

Figura 4.10. Geração involuntária de sinais espúrios.

Na implementação de sintetizadores para utilização em comunicação de dados é sempre colocado como referência qual é o limite de sinais espúrios permitido. Assim, se torna um importante parâmetro a relação entre o sinal ideal e o sinal espúrio bem como as especificações de relação sinal / ruído.

4.3 SINTETIZADOR DE FREQUÊNCIA: ESTREITAMENTO DE ESPECTRO E LARGURA DE BANDA

Quando o objetivo do sintetizador é de recuperação de sinal, ou melhora no desempenho do sinal de referência, tem-se que, o uso do sintetizador ocasiona um estreitamento da banda do sinal e por consequência também uma diminuição dos sinais espúrios. Este efeito é bastante importante para geração de sinais puros e é ilustrado na Figura 4.11.

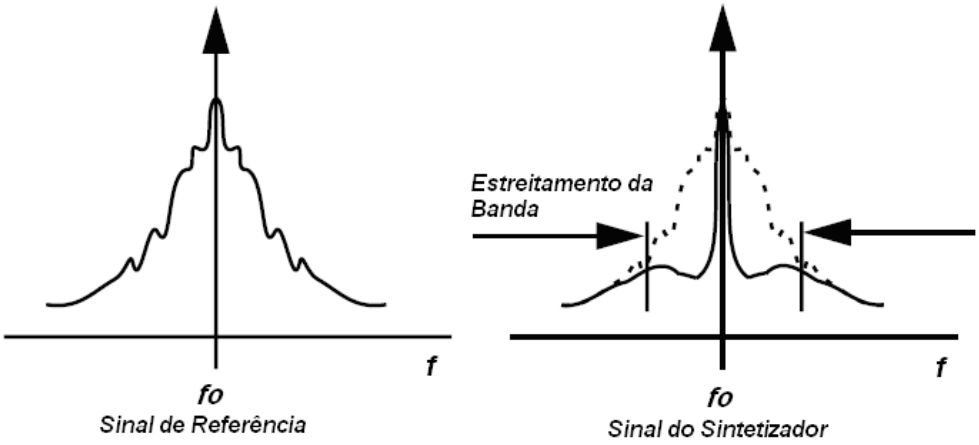

Figura 4.11. Redução de largura de banda.

Com a redução da largura de banda os sinais se tornam mais puros. Mas um fator que limita essa redução é a geração de ruídos oriundos do próprio circuito como ruído branco, ruído térmico, ruído *jitter* e todos os ruídos gerados pelos descasamentos dos componentes. Neste aspecto, a implementação de circuitos tipo DLL se torna importante devido à baixa geração de *jitter* porque o VCDL ou o bloco que contém os atrasadores não possuem retroalimentação, não havendo assim a introdução do *jitter* cíclico.

4.4 SELEÇÃO DE CANAIS

A freqüência de operação do circuito é escolhida pelos parâmetros do *Loop Filter*. Para geração de frequências de saída diferente do sinal F_{ref} têm-se duas alternativas: Usar PLL com os blocos contadores *Multiplier* e *Divider*; ou usar o DLL com *Edge Combiner* (CHIEN, 2000). Utilizando o PLL pode-se acrescentar entre o detector de fase e o VCO blocos tipo *Multiplier* e *Divider*. No caso do número do *Divider* ser de 3 e o *Multiplier* ser 2 tem-se então, que o sistema entrará em sincronismo quando $3/2F_{ref}= F_{out.}$ A seleção de canais causa como consequência indesejada a introdução de sinais espúrios no espectro do sinal de saída, o que pode atrapalhar o desempenho nas características de sinal / ruído.

4.5 ESTABILIDADE

Os circuitos PLL possuem problemas de estabilidade. Este fato pode ser explicado usando o modelo linearizado. No PLL existe um pólo na origem (K_{vco}/s) e também ocorrem pólos no *Loop Filter*. Então, para evitar maiores problemas, os parâmetros do *Loop Filter* precisam ser analisados através do diagrama de Bode e seus pólos serem alocados de forma a diminuir os problemas de oscilação e de instabilidade.

Para o circuito DLL, que não possui VCO, o ganho linearizado do VCDL é K_{vcdl} que é uma constante com uma introdução de fase. Desta forma não existe o zero na origem (K_{vco}/s). O modelo do capacitor ou acumulador, por trabalhar com acumulação de informações, possui um pólo na origem mas não introduz instabilidade por ser facilmente reiniciado ou criado algum circuito para reiniciar automaticamente este acumulador ou capacitor. Por estas características o DLL é considerado estável.

4.6 TEMPO DE ESTABELECIMENTO (*SETTLING TIME*)

Os sintetizadores de frequências usam uma frequência de referência para gerar as frequências seletoras dos canais. A diferença entre a frequência do canal desejada e a frequência de saída do sintetizador de frequência é a especificação denominada precisão (*accuracy*). A dinâmica do sistema sintetizador de frequência implica que o chaveamento dos canais é não-instantâneo. O tempo de estabelecimento (*settling time*) é definido como o tempo despendido para o sintetizador varrer toda a faixa de freqüência, o que significa o máximo tempo necessário para chaveamento dos canais. É esperado que esse tempo seja de vários ciclos do sinal de referência.

4.7 FREQUÊNCIA DE TRABALHO, CONSUMO DE ENERGIA E TENSÃO DE ALIMENTAÇÃO

Tanto em circuitos DLL como PLL, a tensão de trabalho esta ligada diretamente a tecnologia empregada. No caso deste trabalho, a tecnologia disponível é a 350nm da AMS (Áustria Micro Systems), o que permite níveis de tensão de 3,3V ou um pouco abaixo disso, sendo que não foram encontradas publicações em que o valor de alimentação fosse menor que 3,0V para essa tecnologia. Assim as portas lógicas e demais circuitos foram desenvolvidos a partir desde valor. Uma consequência direta do aumento da tensão de alimentação é que o circuito vai consumir mais energia. Caso o objetivo do sistema fosse economia de energia, todos os transistores deveriam ser remodelados com valores de W e L maiores, mas com o limitante de que com área maior, os transistores apresentam maiores valores de capacitâncias e capacitâncias parasitas, o que também compromete o desempenho do circuito em termos de frequência.

Blocos lógicos desenvolvidos em circuitos analógicos ou digitais implementam as equações necessárias para construção do sintetizador de freqüência. No próximo capítulo são apresentadas as principais características dos blocos lógicos dos sintetizadores de frequência tanto do PLL como do DLL.

5 BLOCOS LÓGICOS DE SINTETIZADORES DE FREQUÊNCIA

Os blocos lógicos dos sintetizadores de frequência podem ser desenvolvidos utilizando uma abordagem de circuitos analógicos ou digitais. Atualmente existem diversos tipos de circuitos do tipo PLL que utilizam pelo menos um bloco digital. E também é comum encontrar DLLs que utilizam pelo menos um bloco analógico. Muitas vezes os blocos lógicos são implementados de acordo com as familiaridades dos projetistas. Os primeiros PLL e DLL eram constituídos primordialmente por blocos analógicos implementados segundo a concepção do controle clássico (LINDSEY, 1981). Porém com a evolução dos circuitos e aumento da complexidade e também devido à maior facilidade de projetos em escala mais ampla foram desenvolvidos projetos cuja arquitetura utiliza ferramentas computacionais mais sofisticadas para melhor avaliação dos problemas como ruídos e interferências. Quando implementados em circuitos CMOS também são avaliados o consumo de energia e tamanho do circuito para poder garantir características como portabilidade do sistema e grande capacidade de integração.

Apesar de construção relativamente simples, todos os blocos dos sintetizadores trabalham em frequências bastante elevadas o que implica em problemas de não linearidades e de efeitos de ruídos, de capacitâncias parasitas e demais efeitos indesejados que sempre afetam o desempenho do circuito. Assim é necessário o estudo do princípio de funcionamento de cada bloco e também seu desempenho individual.

5.1 DETECTORES DE FASE

O principal elemento de um circuito PLL / DLL é o detector de fase. Este faz o levantamento do erro entre o sinal de referência e o sinal gerado. A importância destes

dispositivos se deve ao fato de que erros na detecção de fase fazem com que o sistema ou demore muito a atuar, ou em outras vezes, não consiga sair quando necessário do estado de sincronismo. Diversas topologias tem sido implementadas ao longo dos anos de acordo com as conveniências e problemas resultantes de implementação.

A primeira implementação mais simples de um detector de fase é a uma porta lógica Exclusive Or (XOR). Assim, as duas entradas fazem a comparação entre o sinal Up e Dn.

Figura 5.12. Detector de Fase XOR Diferencial.

Como o ganho do detector de fase XOR é bastante limitado, o detector do tipo XOR consegue detectar a diferença de fase mas não ter um ganho proporcional ao deslocamento após a π/8 rad, após isso até 7π/8 o ganho é uma constante como visto na Figura 5.13 (SHU, 2005).

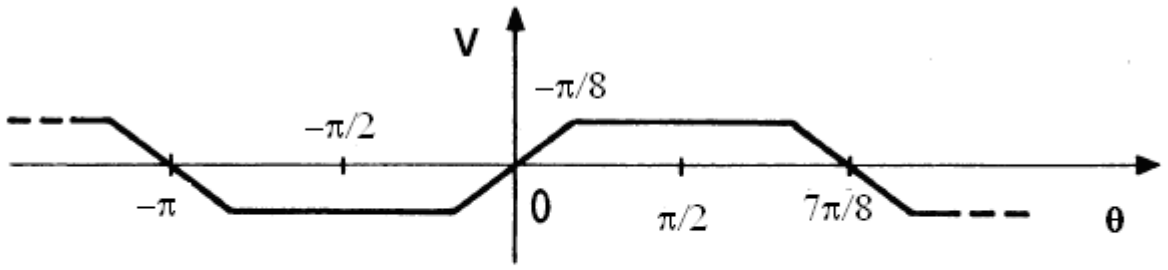

Figura 5.13. Ganho do Detector de fase XOR.

Para melhorar o ganho de fase no intervalo entre π/8e 7π/8 foi proposto à implementação de máquinas de estado finito usando Flip Flop tipo D que conseguem melhorar o desempenho do ganho de fase. Este tipo de solução é bastante comum. Existem variações de detectores de fase

que associado com um detector de sincronismo faz uma parada de leitura para economia de energia (COCUBURN, 2006). No funcionamento PFD (*Phase Frequency Detector*), compara-se as duas entradas **ref** com **div**. Se **div** e **ref** forem iguais então as saídas Up e Dn permaneceram estáveis e iguais a zero. Quando **ref** é maior que **div** então Up =1 e Dn=0 senão Up=0 e Dn=1. Em outros termos temos 4 sinais a serem analisados: ref div Up e Dn. A Figura 5.14 ilustra o processo de temporização, ou formas de onda do Detector de Fase implementado com máquina de estados.

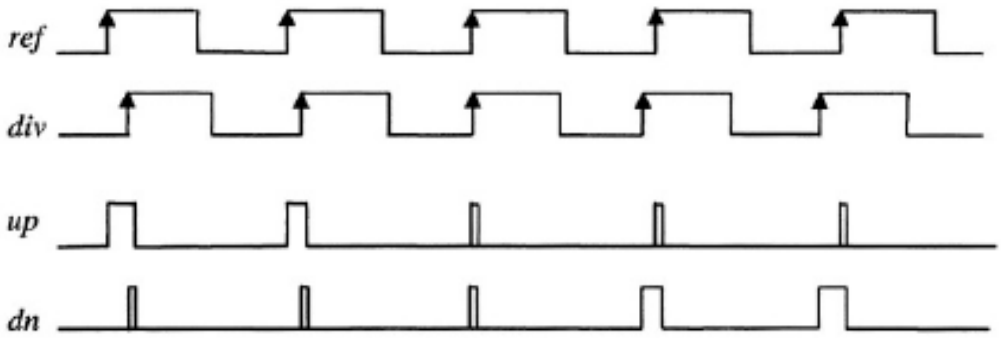

Figura 5.14. Detector de Fase – Formas de onda.

A Figura 5.15 ilustra o ganho de fase obtido quando é usado para detecção de fase um PFD.

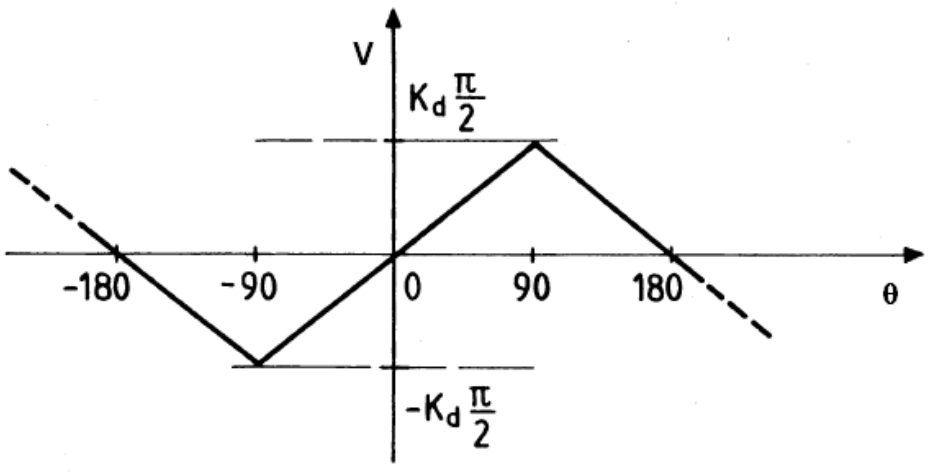

Figura 5.15. Função de Transferência do Detector de Fase com Flip Flops.

5.1.1 Dead Zone

Um efeito importante que é analisado nos detectores de fase é o problema do *Dead Zone* (erro estático do modelo do detector de fase). Com a implementação do circuito detector de fase é possível verificar pequenas imperfeições no seu modelo físico em relação ao ideal. Pequenas variações na diferença de fase geralmente não são detectadas em um detector de fase simples. Isto ocorre devido ao fato de que pequenos atrasos nos elementos lógicos (portas lógicas) precisam ser compensados para que possa haver a detecção. A Figura 5.16 ilustra o problema de *Dead Zone*.

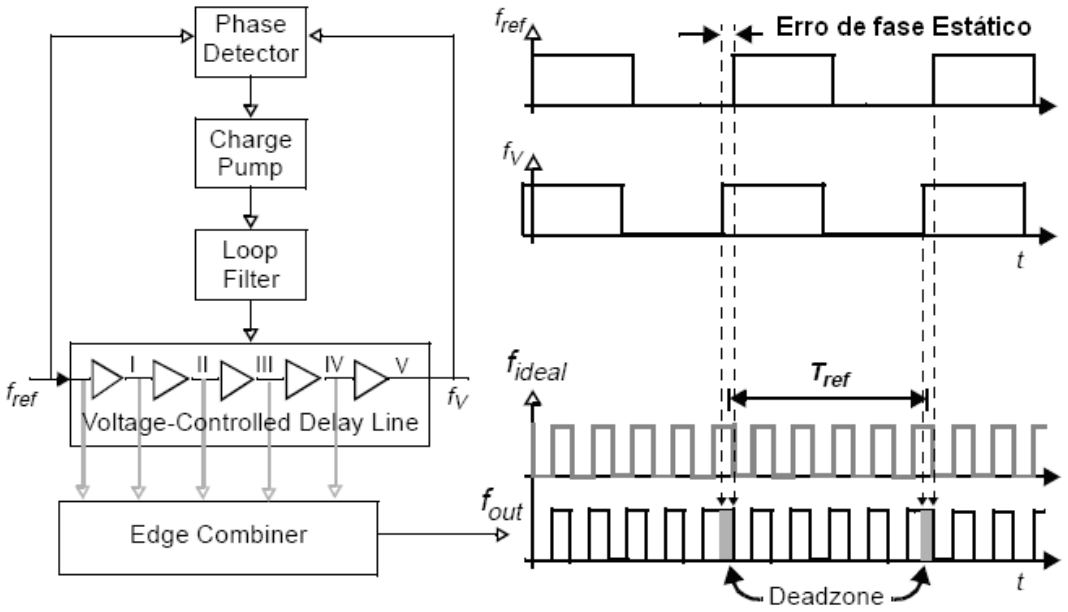

Figura 5.16. Implementação do detector de fase: problema *Dead Zone*.

A solução mais comum para este problema é a inserção de atrasadores (implementados com 2*N inversores) de forma a não se alterar a lógica principal do circuito e de introduzir 2*N atrasos inerentes ao atraso de propagação das células inversoras. Outra solução é o emprego de portas lógicas *NAND* em que as duas entradas são curto-circuitadas para produzir uma porta inversora equivalente. A Figura 5.6 apresenta o diagrama simplificado de um detector de fase com uma célula de atraso (*delay*).

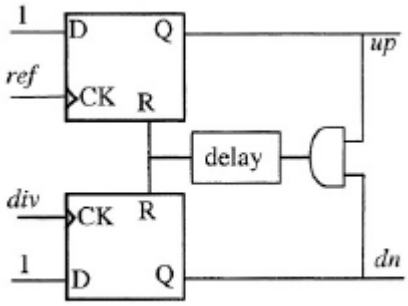

Figura 5.17. Detector de fase com redução do problema de Dead Zone.

5.1.2 Charge Pump

A maioria dos detectores de fase possui saídas diferencias e cada saída representa meio ciclo de detecção. Os sinais dos detectores são em formas de pulsos bastante estreitos. A forma encontrada de se converter esses sinais em sinal de tensão é transformar estes pulsos na forma de corrente e então ativar chaves que convertem estes sinais em um único sinal de controle. A Figura 5.18 mostra o esquema de construção do *charge pump*. Cada uma das saídas do detector de fase (*Up* e *Dn*) controlam as fontes de corrente I_{cp} (corrente do *charge pump*) que é transformado novamente em valor de tensão de acordo com os parâmetros da impedância do *Charge pump* (Z_{lf}). Na saída temos o valor de tensão V_c que é a tensão enviada ao *Loop Filter*.

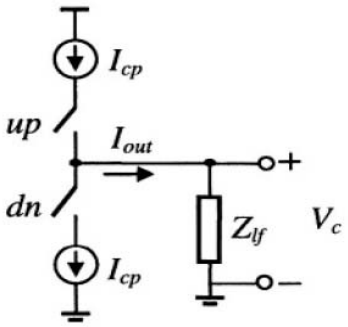

Figura 5.18. *Charge pump*.

Várias arquiteturas são usadas para a construção dos *charge pump*. Na Figura 5.19 são apresentados três tipos usando transistores CMOS.

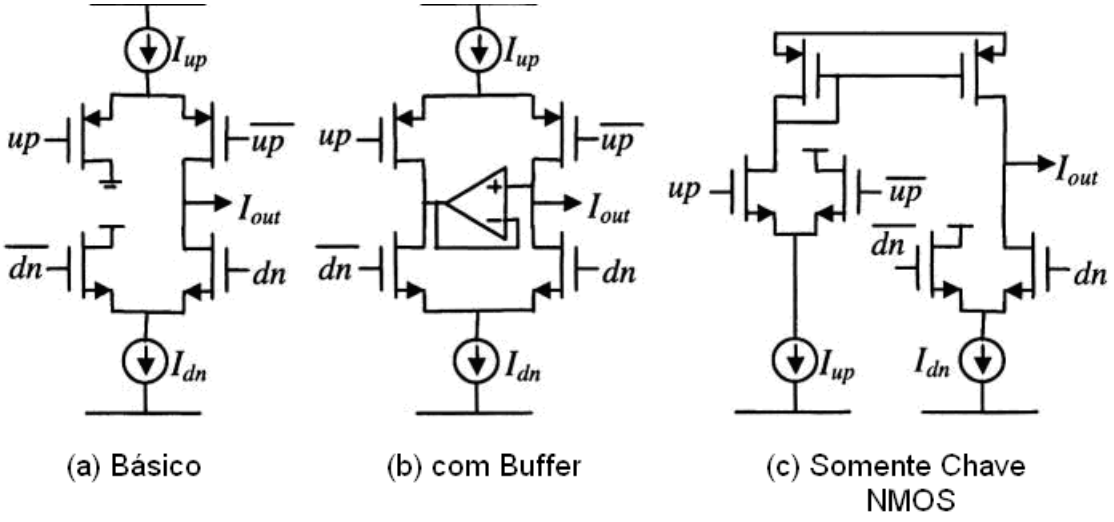

Figura 5.19. Exemplos de *charge pump*.

Para implementações mais complexas pode-se alterar ou produzir controles nos valores de corrente dos *charge pump* com o intuito de melhorar o desempenho do sistema para facilitar e acelerar o processo de chaveamento e estabelecimento de sincronismo (LIP, 2007).

5.1.3 Acumulador de Fase

Depois de obtido o valor da detecção de fase e o sinal convertido em um único pulso faz se necessário um sistema que atue como acumulador de fase. Circuitos analógicos implementam este acumulador de fase utilizando um capacitor com sistema de inicialização e *reset*. Conforme o tempo transcorre o ângulo de fase se altera e o valor do erro acumulado aumenta. O aumento do valor de acumulação depende do valor absoluto do erro. Então um erro duas vezes maior implicará em um aumento duas vezes mais rápido no valor de fase acumulada. Na Figura 5.9 é apresentado o processo de acumulação de fase quando o sistema esta em malha aberta.

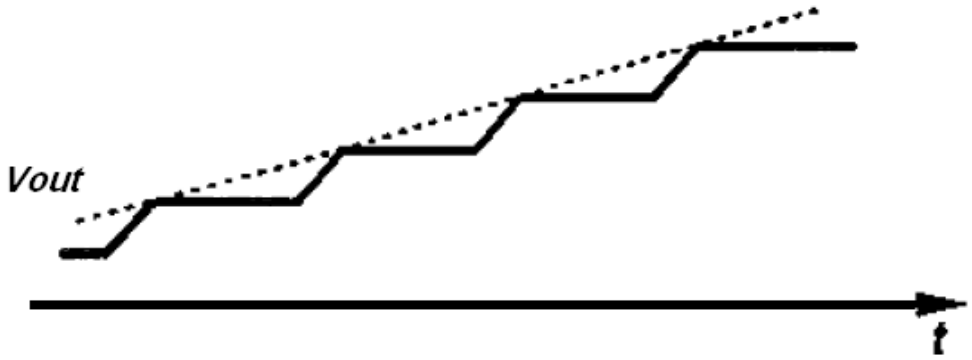

Figura 5.20. Processo de Acumulação de Fase.

Em circuitos digitais é utilizado um contador do tipo *Up* / *Down* que dispensa a utilização no sistema do bloco *charge pump*. Porém a resolução do acúmulo de fase fica limitado à resolução do contador. O valor total acumulado depende da diferença de fase entre os sinais e a figura 5.10 ilustra esse processo.

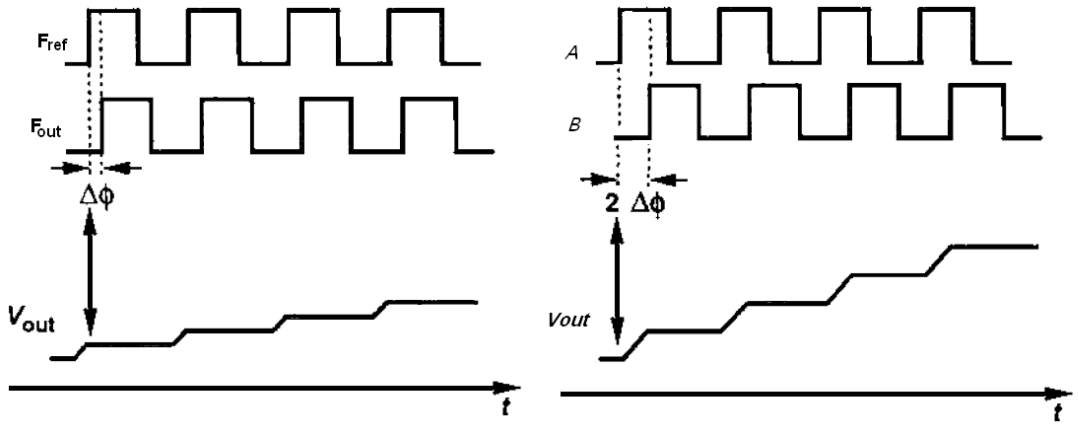

Figura 5.21. Variação do acúmulo de fase em relação ao erro de fase.

5.1.4 VCO

Quando se cria uma cadeia de elementos de inversores de numero ímpar e se retroalimenta a saída em relação à entrada, gera-se um sistema naturalmente instável.

Uma vez acionado o primeiro elemento, o segundo elemento então estará desabilitado, o que fará com que o terceiro elemento fique então ativo. Como o último elemento dessa cadeia

terá que, nessa seqüência, necessariamente estar ativo, desabilitará então o primeiro elemento da cadeia, e então invertendo todos estados dos elementos sucessivamente. Este princípio é utilizado no desenvolvimento de alguns osciladores chamados em anel que de forma resumida podem ser interpretados como sendo um VCO.

No exemplo típico de um VCO um sinal de referência de entrada DC, controla e regula o tempo de atraso dos elementos inversores de forma a controlar a frequência de oscilação natural do conjunto. Existem diversas topologias de elementos VCO, muitos com elementos passivos como indutores e capacitores, consumindo uma grande área de implementação quando implementado em tecnologia CMOS. Existem outras propostas de implementação do VCO que utiliza tecnologia CMOS e topologias digitais.

Basicamente existem dois principais métodos de implementação de um circuito VCO que são os osciladores em anel e os que se utilizam de um tanque LC. Em um esquema de atraso ou multi-fase, o oscilador utiliza os atrasos defasados para conseguir maior velocidade de oscilação. Muitas implementações de osciladores controlados já ultrapassam unidades de GHz de frequência central, utilizando multi-fases (OLIVEIRA, 2008). A figura 5.11 ilustra uma célula usada no circuito de osciladores em anel.

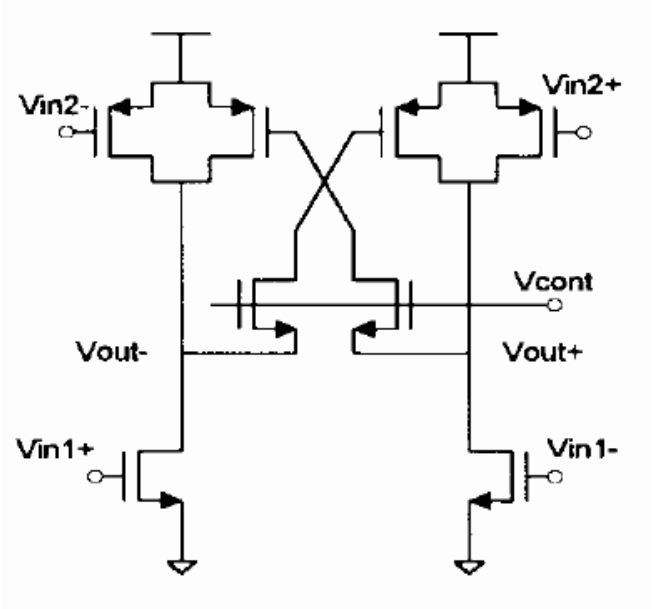

Figura 5.22. Célula de Atraso VCO.

O VCO tipo tanque em muitos casos ocupa uma área considerável levando-se em conta o tamanho do circuito CMOS pelo fato de que a implementação de capacitores e indutores ocupam

grande área do circuito. No exemplo da Figura 5.23 tem-se um VCO em que os capacitores são emulados usando os transistores M5 e M6.

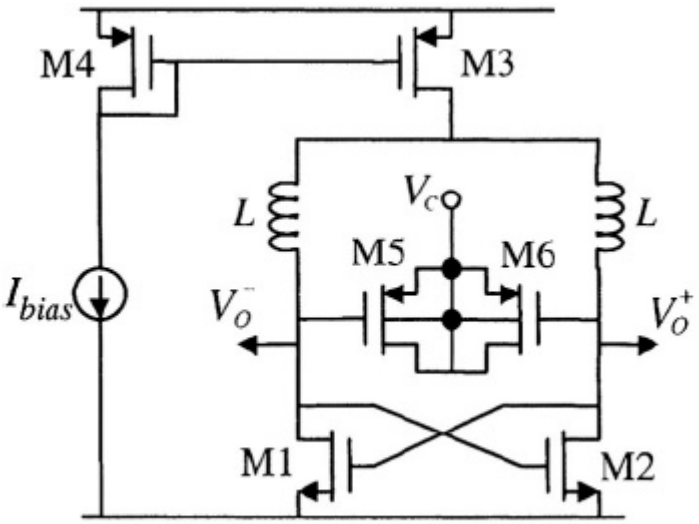

Figura 5.23. VCO tipo LC.

5.1.5 VCDL

Um circuito característico de um DLL é o VCDL (*Voltage Controlled Delay Line*). O VCDL representa o que em um PLL seria o VCO. As principais características de um VCDL é ser um arranjo de células atrasadoras controladas por tensão ou alguma outra referência. Outra característica importante é que não existe a propagação de *jitter* entre as células o que facilita a implementação de circuitos quando a geração deste tipo de ruído é criticamente indesejada. O circuito deste trabalho propõe um arranjo de células atrasadoras que possam ser programadas de forma a termos um controle total do atraso gerado. O atraso é criado de forma ponderada a cada bit o que contribui com um número menor de células atrasadoras e também que um número menor no tempo de sincronismo do DLL.

A diferença então em relação aos VCO feitos em cadeia é que o VCDL não possui sinal de realimentação o que implica a não propagação dos sinais (ruídos) tipo *jitter* (BAE, 2005). O sinal de tensão de entrada controla o atraso de propagação, assim, conforme se aumenta o valor de tensão aumenta-se o atraso e a frequência de trabalho é diminuída (CHANG, 2002).

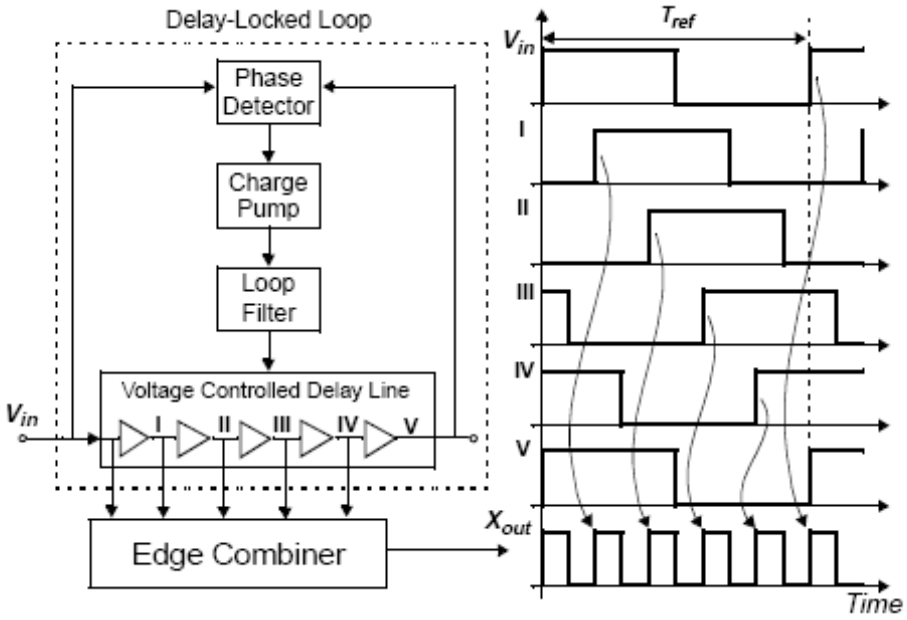

Figura 5.24. VCDL.

Existem muitas arquiteturas de VCDL que trabalham com um pequeno número de células (XILINX, 2001) cujo objetivo do circuito é a geração de sinais múltiplos da frequência de entrada pelo número 2 e utiliza-se 4 elementos atrasadores para gerar uma defasagem de 90 graus em relação a cada elemento. Já a proposta de Cockburn (2006) utiliza-se de 256 elementos para obter uma grande variação de frequência de trabalho variando de 14 a 250MHz. Os objetivos do DLL são fundamentais para a escolha do tipo de cadeia de VCDL utilizada.

Neste trabalho opta-se por trabalhar com não exatamente uma VCDL (que por definição deveria variar de acordo com tensão) mas sim com uma cadeia de atrasadores programáveis (no sentido de configuráveis) controlado pelo valor acumulado no contador Up/ Down.

Neste capítulo foram descritos os blocos lógicos necessários para implementação tanto de um PLL como DLL. No próximo capítulo será descrito a escolha confecção e desenvolvimento do DLL para recuperação de sinais.

6 DESCRIÇÃO E IMPLEMENTAÇÃO DO DLL DIGITAL PROPOSTO

Um circuito DLL típico é composto por três componentes: detector de fase, *charge pump* e VCDL. Convencionalmente pode-se categorizar os DLL em dois tipos: analógicos e digitais. Neste trabalho adota-se a utilização de tecnologia digital para substituição do *charge pump* e do capacitor do *Loop Filter* e implementou-se blocos lógicos compatíveis. Os blocos lógicos digitais implementados como substituto do *charge pump* e capacitor foi o contador Up / Down de 8 bits com entrada diferencial.

A contribuição mais importante é a implementação de linha de atraso programável com cada bit com peso diferente sendo que cada bit produz o atraso proporcional a 2^n ζ em que n é o número do bit e ‘.

6.1 PROJETO DE DLL DIGITAL

Ferramentas CAD (*Computer Aided Design)* baseadas em captura de esquemático apresentam uma dificuldade enorme de implementar circuitos com centenas de elementos. Deste fato, surge a necessidade do estudo e desenvolvimento de sistemas capazes de implementar circuitos e sistemas digitais com um maior número de elementos. O uso de ferramentas computacionais é indispensável na fase de projeto quando o sistema se utiliza de muitos elementos. Várias ferramentas computacionais foram desenvolvidas nos últimos anos para auxiliar no processo de síntese de circuitos digitais, tais como software Mentor, Cadence, Altera (ALTERA, 2001) e Xilinx que permitem o desenvolvimento, simulação, compilação e minimização dos circuitos de forma bastante otimizada e automatizada. Arquivos tipo EDIF (*Eletronic Design Interchange Format*) permitem que uma vez compilado o circuito em uma destas ferramentas, possa-se fazer o intercâmbio de

informações entre as ferramentas e recompilar os processos sem a necessidade de novo desenvolvimento dos circuitos.

As vantagens de se adotar uma tecnologia baseada em conceitos digitais são a portabilidade (que é o fato de se poder migrar de tecnologia usando-se a mesma arquitetura básica) e robustez (que é a baixa interferência dos efeitos do processo sobre os parâmetros de desempenho do circuito). Também são importantes a baixa tensão de alimentação e também a facilidade de desenvolvimento do projeto devido ao uso de ferramentas digitais. Dentre as principais desvantagens pode-se incluir a maior área do circuito em relação a circuitos puramente analógicos e provavelmente, maior consumo de energia. Outra desvantagem na construção de um DLL é a impossibilidade de fazer seleção de canais como o PLL. Isto ocorre porque o DLL não trabalha com oscilação de frequências, mas sim sempre em torno do mesmo tom. Em termos de período de referência T_{ref} tem-se que (JIA, 2005):

$$\max\{(D_{vcdl}(\min), \frac{2}{3} D_{vcdl}(\max))\} < T_{ref} < \{(\min 2D_{vcdl}(\min), D_{vcdl}(\max))\} \quad (4)$$

Sendo que:

D_{vcdl} - Atraso de programação do arranjo de Células atrasadoras;

$D_{vcdl}(\text{Max})$ - Atraso na condição de valor máximo;

$D_{vcdl}(\min)$ - Atraso na condição de valor mínimo.

O papel do circuito detector de fase é comparar o sinal F_{ref} com F_{out}. Quanto maior o ganho melhor será seu desempenho porque diminuirá o tempo de sincronismo. O detector de fase mede a diferença entre sinais de freqüências inicialmente diferentes, ou seja, no sentido mais formal o que o detector de fase mede é o deslocamento dos sinais. Porém conforme os sinais vão se estabelecendo e ficando com a mesma freqüência, o detector de fase mede a diferença de fase entre os sinais. Ao valor do detector de fase é aplicado um ganho que depende dos aspectos construtivos do Detector de Fase e por conseqüência o valor deste ganho precisa ser convertido em um sinal não diferencial e acumulado ao longo do tempo. Para sistemas mistos analógico e digitais às vezes é empregado para essa função um simples capacitor. Mas para tanto, deve se levar em consideração que em circuitos CMOS, às vezes capacitores ocupam área relativamente grande e proibitiva e também é necessário um circuito auxiliar para fazer o reinício do processo de sincronismo.

Outro processo adotado é a utilização de circuitos tipo contadores *Up* / *Down*. Desta forma, cada uma das saídas do detector de fase tipo PFD aciona uma das entradas do contador.

Por consequência, conforme se muda o tempo de atraso, D_{vcdl} é modificada a frequência de trabalho do sistema. A resolução do sinal é definida como sendo a menor variação possível criada para se mudar a frequência de trabalho.

6.2 CONTADOR, AMOSTRAGEM E ACUMULAÇÃO

Os sinais *Up* / *Down* representam a direção que um acumulador precisa ir pra que se faça o correto controle do sinal que controla o VCO ou o VCDL. Assim precisa-se converter a saída diferencial em uma saída única e fazer a acumulação destes valores de forma a encontrar o valor correto da acumulação de fase. Nos sintetizadores tradicionais é utilizado um circuito *charge pump* em que cada um dos sinais é colocado como fonte de corrente pra controlar o ganho do *charge pump*. Parâmetros de ganho dessa corrente podem ser controlados para melhorar o desempenho de sincronismo do sistema. Na implementação proposta por este trabalho é implementado um contador síncrono que faz a amostragem dos dois sinais ao mesmo tempo e também faz a acumulação do sinal. Quando o valor extrapola o valor máximo da contagem, automaticamente o contador é zerado e computado o ganho de fase. Desta forma o ganho de fase é contínuo independente do começo ou final da contagem do contador. A Figura 6.25 ilustra o processo de amostragem para um período de amostragem = T_s.

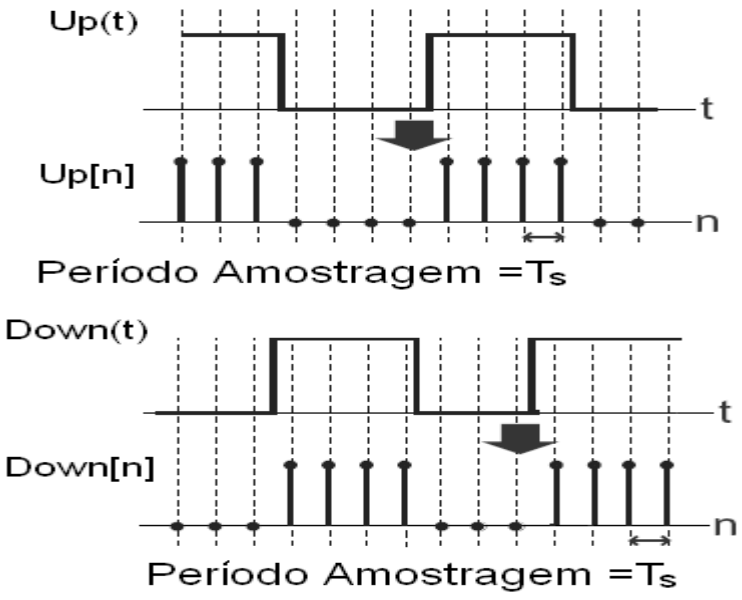

Figura 6.25. Processo de Amostragem Up e Down.

O contador precisa ser acionado por um clock mais rápido que o sinal Down(t) e Up(t). Durante o transcorrer do tempo é feita amostragens quando o clock do contador estiver ativo (metade do período). Nesta amostragem várias partes do sinal original não são computadas. Este problema causa vários efeitos de ruído e *jitter* mas no acumulado da fase praticamente são anulados esses problemas de ruídos. O valor final a ser computado leva em consideração vários períodos de amostragem. O contador digital implementado tem duas entradas e faz a soma de Up(t) e Down(t). Para que não houvesse conflitos de prioridade foi estabelecido que quando os dois sinais fossem ambos "zero" lógico ou "um" lógico, o contador não iria incrementar ou decrementar o valor da contagem.

6.3 GERAÇÃO DO SINAL CONVERGÊNCIA E O ESTABELECIMENTO

O objetivo principal do DLL é a geração do sinal que seja a reprodução do sinal de referência de entrada e adicionar alguma fase quando necessário, sendo que o sinal de saída possua menor *jitter*. Ainda, espera-se que o sinal do DLL possa ser estabelecido o mais rápido possível e com o menor ruído. A comparação entre o sinal gerado e o sinal de referência pelo detector de fase a cada ciclo de clock faz com que o sistema esteja completamente instável no início do processo de detecção e, neste momento, o sinal de saída apresenta bastante sinais espúrios. Espera-se que a cada ciclo de clock o DLL possa monotonicamente alterar o tempo de atraso do VCDL (ou linha de atraso programável) de forma que em alguns ciclos de clock o sistema possa convergir. Na literatura é mostrado que esse processo de estabelecimento pode variar de alguns ciclos de clock até mais do que uma centena deles. Esse número é bastante influenciado por fatores tais como tecnologia empregada, arquitetura empregada e frequência de operação.

Problemas de convergência podem ocorrer quando não existe limitação da frequência de trabalho do DLL. Se a variação de atraso que o DLL pode gerar for muito maior que o intervalo de período de trabalho de frequência do sistema o DLL pode entrar em sincronismo em falha (*false-lock*). Este problema deve ser resolvido para evitar problemas na geração final do sinal. Uma das maneiras tratadas para corrigir o problema de *false-lock* é o desenvolvimento de um

bloco chamado detector de sincronismo que atua de forma a detectar a lógica de sincronismo e atuar então no detector de fase para alterar o procedimento de contagem do detector.

Dependendo do emprego do DLL pode ser necessário ainda o emprego de um corretor de ciclo ativo, porque algumas arquiteturas não garantem que o DLL gere um sinal com 50% de ciclo ativo ao final do processo de geração de sinal.

Outro sistema que pode ser adotado é o de *Edge Combiner* que é a geração de sinal múltiplos da frequência de saída F_{out}, utilizando a combinação dos sinais sucessivos obtidos de diferentes pontos da cadeia de atraso.

6.4 DIMENSÕES E AJUSTES DAS RELAÇÕES DOS TRANSISTORES

As dimensões dos transistores W/ L (largura e comprimento) empregados nos circuitos devem sempre ser de dimensões mínimas, ou seja, tem que ser projetados de tal forma que não exista combinação possível de W/L de valor menor. Isto ocorre devido ao custo de implementação que aumenta devido às dimensões dos transistores. Outro fator é que as capacitâncias parasitas são proporcionais as dimensões dos elementos (RAZAVI, 1998). Então a forma de minimizar esses efeitos indesejáveis é projetar os transistores em dimensões mínimas.

Os de W/ L também são função da tensão de alimentação sendo que quanto menor os valores de tensão, maiores são as dimensões de W/ L para que haja a polarização do transistor. As capacitâncias parasitas também têm influencia na frequência máxima de operação dos transistores. Quanto maior a frequência de operação do circuito mais fácil é verificado os efeitos das distorções das formas de onda de tensão de saída de forma até que seja então inviável a operação do circuito numa frequência alta.

A tecnologia disponível para desenvolvimento é a 350nm do kit Mentor – AMS (Austria Micro Systems) que apesar de não estar atualizada, pois atualmente já estão disponíveis no mercado Kits em que a tecnologia é 130 nm, é adequada para as avaliações sistemáticas do nosso problema.

Outro aspecto a ser observado é a utilização de blocos lógicos desenvolvidos a partir de circuitos digitais. Esta escolha foi feita para priorizar as simulações em qualquer tipo de ferramenta de simulação que comporta avaliações de ferramentas de simulação de alto nível.

Uma desvantagem própria deste método é que comparado com circuitos analógicos, existe quase sempre um aumento considerável de área do circuito total e também do consumo de energia.

6.5 CONSTRUÇÃO DA CADEIA DE ELEMENTOS DE ATRASO PROGRAMÁVEL

A frequência de trabalho do DLL e sua resolução são definidas pela cadeia de atrasadores. Assim, o primeiro passo é a definição da resolução de fase sistema, que é a menor unidade de atraso controlada pela cadeia ζ. Desta forma tem-se que resolução é igual ao atraso mínimo possível por esta cadeia. E que o número total da cadeia multiplicado pelos atrasos individuais geram o tempo mais longo da cadeia.

Nos principais projetos de linha de atraso o tempo de atraso de cada célula são iguais entre si. Desta forma cada célula contribui com um tempo de atraso de T_{ref}/N, em que N é o numero de células da cadeia. Num exemplo, fazendo-se uma cadeia com 8 atrasadores e que cada célula atrasadora possa introduzir um atraso de 10 ηs tem-se o seguinte: o total de atraso de um VCDL precisa ser igual a um período de clock T_{Ref} =80ηs, frequência máxima de 50 MHz e frequência mínima de 6,25 MHz. Um maior número de células atrasadoras aumenta a resolução de freqüência, mas faz com que o sistema tenha que operar em frequências maiores devido ao maior atraso da linha de atraso.

Para se conseguir boas resoluções muitas vezes são utilizadas grandes cadeias de atrasadores com centenas de elementos. Um número que aparece frequentemente na literatura é uma cadeia com 256 elementos (COCKBURN, 2006). Para poder manipular e controlar adequadamente cadeias tão longas muitas arquiteturas foram propostas de forma a manter a resolução de frequência mas fazendo com que células não tenham mais o mesmo tempo de atraso. Outra evolução das VCDL foi a divisão desta linha em duas etapas: denominadas de *Fine Delay Line* (Resolução Fina) e *Course Delay Line* (Resolução Grossa). Além de permitir melhor controle do andamento da seleção da resolução, permite-se que haja menor geração de *jitter* (GALLUP, 1999).

6.5.1 Obtenção do Atraso fixo

Uma porta lógica possui um atraso de propagação fixo, este valor de atraso é função da tensão de alimentação e principalmente das dimensões dos transistores envolvidos. Desta forma pode-se implementar e controlar atrasos em portas lógicas como portas NOT, NOT Tri-State, NAND, de forma a fazer com que atrasos possam ser controlados de acordo com especificações desejadas.

Diferentes dimensões permitem que os transistores das portas lógicas possuam mais ou menos efeitos capacitivos de acordo com as dimensões. E esses efeitos capacitivos basicamente definem o tempo de respostas dos componentes. A tecnologia adota de 350 nm permite que se faça uma resolução de 50ps quando se utiliza a relação entre atrasos de diferentes portas lógicas.

6.5.2 Portas Logicas para geração de atrasos

Para o desenvolvimento de lógicas digitais é necessário primeiramente o desenvolvimento de cada porta lógica digital de acordo com as características do sistema. Foram desenvolvidas as portas lógicas priorizando o desempenho em frequência e por conseqüência a diminuição de área de forma a obtermos circuitos rápidos e com área reduzida.

6.5.3 Inversor e Tri-State Inversor

Para o desenvolvimento de células de atraso formam utilizados como unidades o inversor comum e o inversor *tri-state*. A escolha foi para ter disponível atrasos que possam ser múltiplos um do outro. Foram escolhidos como atraso de 50ps para o inversor como padrão e o tri state foi dimensionado para que o atraso fosse o dobro do valor 100 ps. Temos na Figura 6.26 e na Figura 6.27 a representação destas montagens. Nestas figuras os números representam as dimensões de W e L em micrometro.

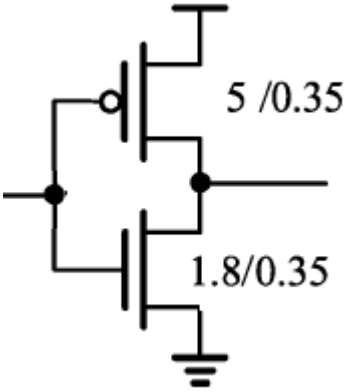

Figura 6.26. Inversor.

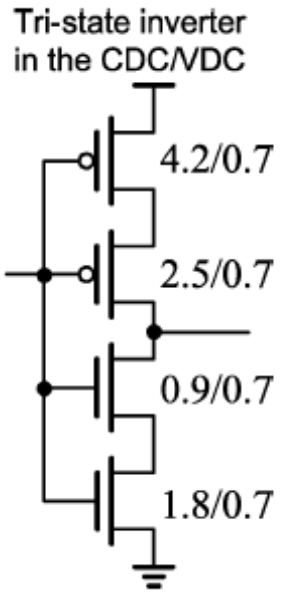

Figura 6.27. Tri State Inversor.

Todas as configurações das dimensões dos transistores (W e L) foram projetadas para que o circuito seja alimentado com tensão de 3,3V. Este fato é pelo comprometimento entre frequência de trabalho e área dos circuitos.

6.5.4 Modelo de Simulação e uso de Tecnologia Digital

Para facilitar o processo de simulação, validação e verificação foram utilizadas ferramentas que trabalham somente com circuitos digitais porque diversas variáveis poderiam ser a princípio não consideradas levando-se em consideração apenas o desempenho da arquitetura do sistema. Desta forma valida-se a arquitetura, posteriormente, fazer as adaptações e configurar os parâmetros para implementação em circuitos CMOS.

As ferramentas utilizadas foram Max + Plus II v 10.1 da Altera que implementa, configura e simula circuitos digitais principalmente das famílias de FPGAs e CPLDs MAX e FLEX. e Quartus v 7.2 e 9.0. E que permitem ainda a implementação, simulação e gravação da família de componentes Cyclone II.

Outro recurso da ferramenta utilizada é a entrada em VHDL que permite que parte do circuito ou mesmo ele todo possa ser programado e configurado em alto nível de abstração, o que em termos práticos facilita todo o processo de portabilidade tão importante em circuitos digitais. Esta prática de utilização de VHDL e outras linguagens vem sendo muito utilizado em circuitos tipo DLL (XILINX, 2000).

A validação da arquitetura foi realizada com a placa de desenvolvimento Cyclone II FPGA Starter Development Board que utiliza como FPGA o componente Cyclone II EP2C20F484C7. Este componente tem como restrições aplicações em que a frequência interna de trabalho sejam maiores de 400 MHz (ALTERA, 2006). Nas simulações utilizando as ferramentas Quartus apresentou frequências externas até o limite de 100 MHz.

6.6 SIMULAÇÕES E IMPLEMENTAÇÃO DE ARQUITETURAS DIGITAIS

Para verificar o funcionamento do DLL testou-se a implementação do modelo digital usando a ferramenta de desenvolvimento de projeto da Altera Max + Plus II e também da ferramenta Quartus II v. 9.0. Apesar de não apresentar todas as ferramentas para cálculos de

potência, *jitter* e análise de histograma de frequências por exemplo permite uma verificação das condições de funcionamento e sincronismo do sistema. Nestas ferramentas de desenvolvimento de circuitos digitais são realizadas as seguintes etapas de projeto: Entrada de projeto, síntese, roteamento, simulação, análise temporal, programação e gravação. A Figura 6.28 apresenta o diagrama do desenvolvimento dos projetos digitais nas ferramentas de projeto.

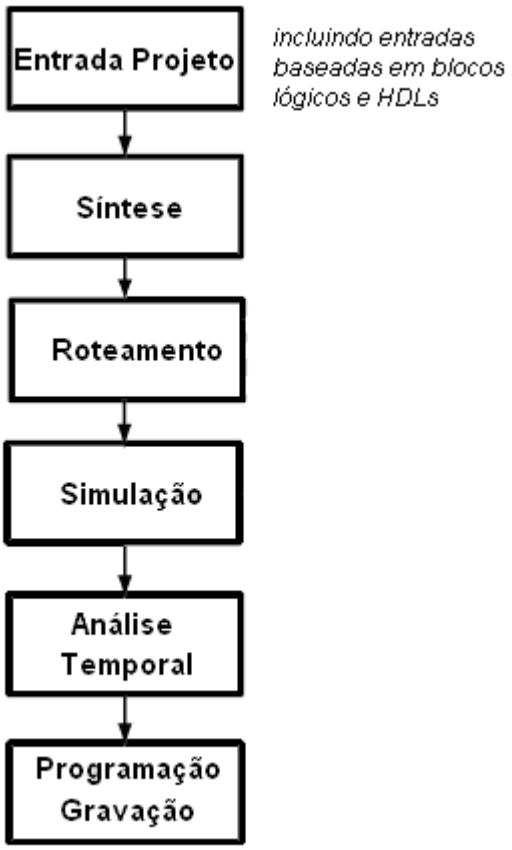

Figura 6.28. Diagrama de Síntese Digital.

A entrada de projeto é o tipo de arquivo a ser usado na ferramenta. Nesta etapa de projeto são permitidas diversas formas de arquivo sendo as principais os arquivos gráficos, arquivos tipo Netlist, arquivo EDIF e os arquivos de HDL (linguagens de descrição de hardware). Os arquivos de HDL e EDIF são padronizados para que possam funcionar em todas as plataformas e não ser dependente de fabricantes de ferramentas ou associados à alguma tecnologia. O processo de síntese faz a verificação se é possível a implementação física do sistema levando-se em consideração a do número de componentes do sistema. O processo de roteamento também faz a verificação das possibilidades de ligação física entre os blocos lógicos implementados no sistema e traz a verificação das possibilidades de roteamento de acordo com a pinagem final do sistema.

As etapas de simulação e análise temporal fazem a verificação de uma implementação física do sistema. Apesar de o nome ser simulação, geralmente estas ferramentas apresentam dados bastante confiáveis em termos de validação das condições de especificação do circuito. A geração do arquivo de roteamento significa na prática que o circuito é factível do ponto de vista de implementação e consequentemente é gerado o arquivo de programação e gravação do dispositivo FPGA.

6.7 ESCOLHA, MONTAGEM E IMPLEMENTAÇÃO DOS BLOCOS LÓGICOS DA CONFIGURAÇÃO PROPOSTA

6.7.1 Detector de Fase

Como o DLL precisa trabalhar em uma frequência a mais alta possível e com geração bastante baixa de *jitter*, o detector de fase necessita ter as seguintes características: alto ganho, saída diferencial, redução dos problemas de *dead zone* e rapidez de chaveamento. Para obter estas características optou-se pela implementação de um PFD (*Phase Frequency Detector*). A Figura 6.29 apresenta um modelo de detector de fase e na Figura 6.30 o detector de fase implementado para o DLL.

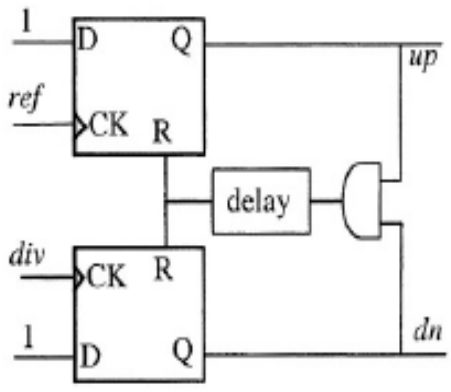

Figura 6.29. PFD para DLL.

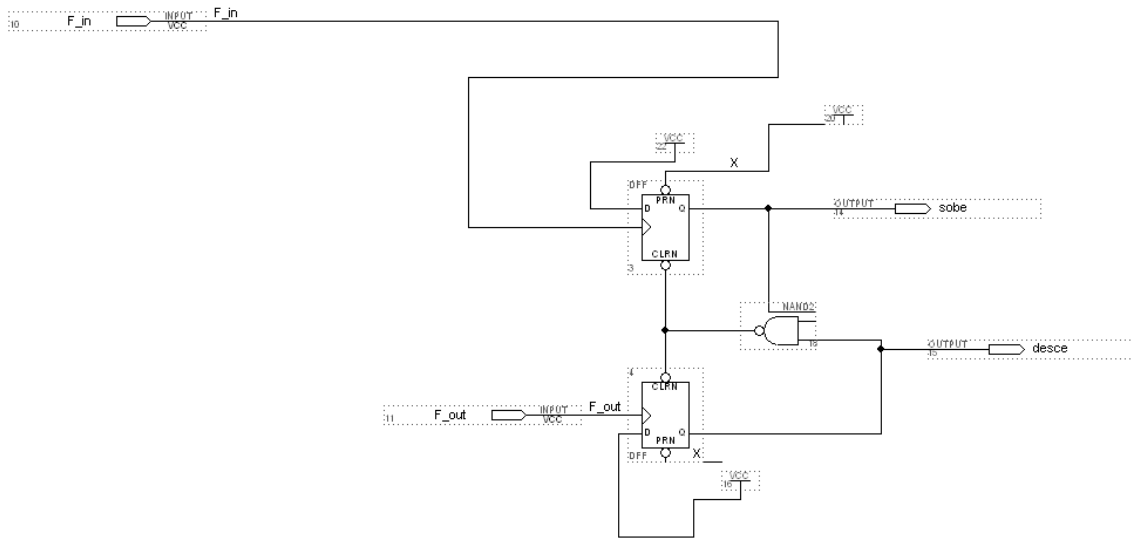

Figura 6.30. Detector de Fase Implementado.

Este detector de fase é composto por dois Flip Flops tipo D e uma porta AND. A célula de atraso é constituída por dois inversores colocados em seqüência de forma a não alterar a lógica do circuito. O detector de fase foi simulado em malha aberta, ou seja, foi isolado para verificação do funcionamento sem interferência dos demais blocos. Na Figura 6.31 o sistema está em fase, ou que significa que os sinais F_out e F_in são iguais. As saídas do detector de fase são apenas pulsos bastante curtos e para esta situação os valores de **sobe** e **desce** entram com sinais opostos no contador e acabam se anulando não provocando alteração na contagem do contador.

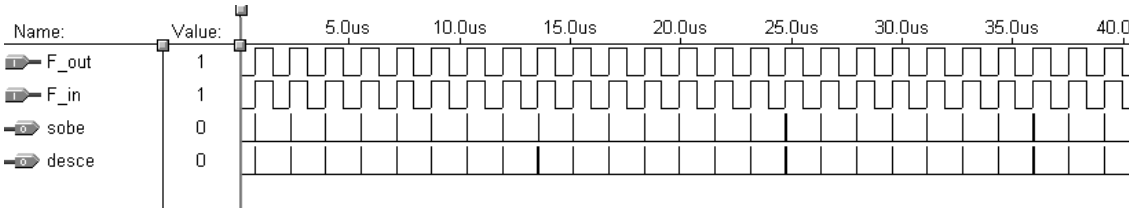

Figura 6.31. Fout = F_in sistema em fase.

Na Figura 6.32 tem-se o caso de F_out e F_in com mesma freqüência, mas com F_in adiantado. Neste caso é verificado que no sinal **desce** prevalece o positivo, fazendo com que o contador diminua o valor de contagem atual.

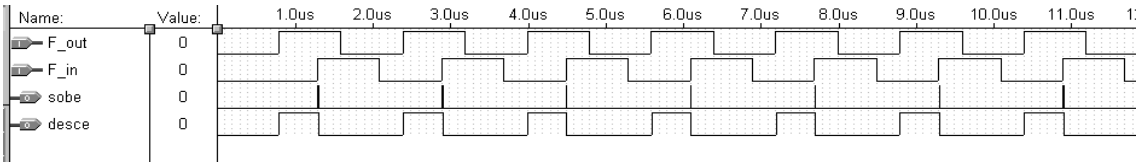

Figura 6.32. F_out e F_in com mesma frequência mas F_in atrasado.

O exemplo oposto da Figura 6.32 é apresentado em Figura 6.33, caso em que o contador é solicitado a aumentar a contagem.

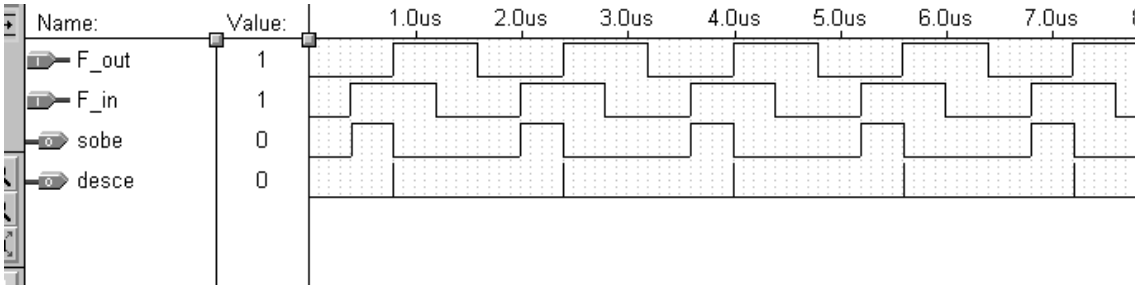

Figura 6.33. F_out e F_in com mesma frequência mas F_in adiantado.

A Figura 6.34 ilustra a situação em que F_out é mais rápido que F_in e então o detector ativa o **desce** de forma proporcional ao atraso de fase detectado.

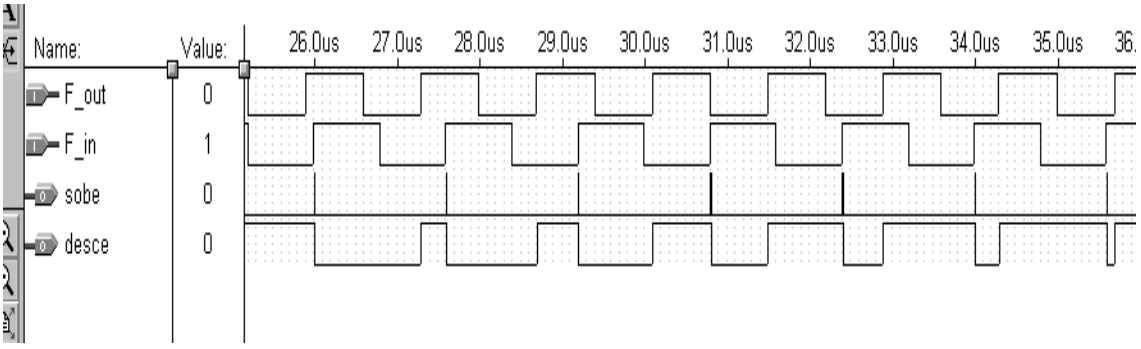

Figura 6.34. F_out mais rápido que F_in.

6.7.2 Contador

O contador foi desenvolvido utilzando-se a linguagem de descrição de hardware VHDL. A utilização de VHDL se deve a rápida prototipagem e também pelo fato de não estar bem definido o número de bits, então este contador apresenta 8 bits podendo este valor ser alterado até a implementação final do projeto.

O contador faz o papel de acumulador de fase e *charge pump*, desta forma, o clock acionado faz com que haja a leitura das variáveis de entrada *Up* e *Down*. E estes valores são somados e determinam se o contador deve ser incrementado, decrementado ou permanecer estável.

6.7.3 Detecção e Acumulação de Fase

Para verificar o funcionamento do sistema Detector de Fase e Contador temos na Figura 6.35 um sistema para verificar se realmente o detector de fase somado ao contador consegue operar corretamente.

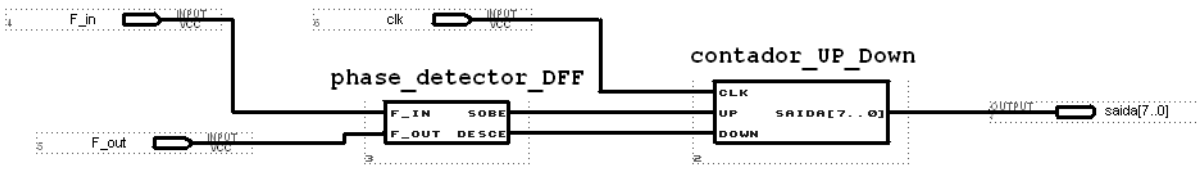

Figura 6.35. Detector de Fase e Contador.

A Figura 6.36 apresenta a simulação quando F_in esta adiantado de F_out e na seqüência a Figura 6.37 apresenta o caso em que F_in esta atrasado e a contagem é decrementada.

Nas simulações do detector de fase foram verificados sinais espúrios, ou em outras palavras oscilações inerentes aos atrasos dos circuitos digitais nas transições de seleção da largura do período de frequência. Como este tipo de oscilação representa uma geração indesejada

de *jitter*, foi implementado um sistema baseado em clock para que não mais fosse gerado esse tipo de ruído.

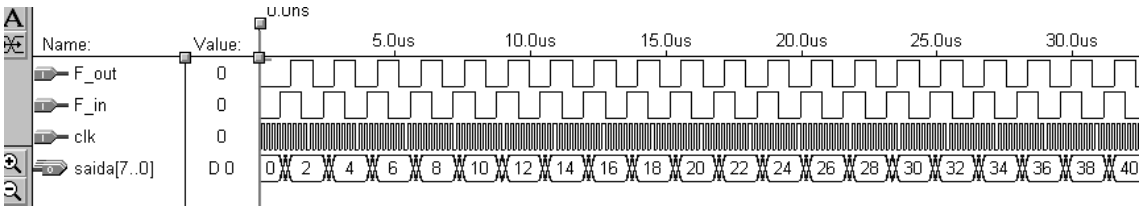

Figura 6.36. O sinal F_in adiantado: Contagem crescente.

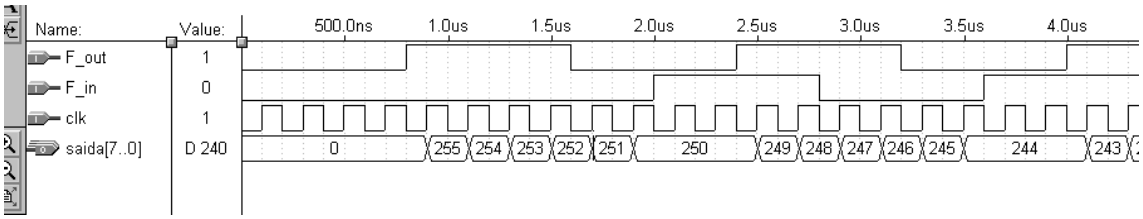

Figura 6.37. O sinal F_in atrasado: Contagem decrescente.

6.7.4 Linha de Atraso Programável Digital

Para criar agora um sistema DLL é necessário criar um sistema que produza um atraso proporcional ao número do contador. Desta forma, em algum momento depois de alguns ciclos do período de F_in (sinal de referência de entrada) o sistema entrará em sincronismo. Mas para que isso aconteça é necessário que a frequência do F_in seja limitada porque também é limitada a capacidade de seleção de fase de qualquer VCDL ou neste caso, um Arranjo Programável de Atraso.

A primeira idéia de um tipo de Arranjo Programável de Atraso foi estabelecer um sistema semelhante ao apresentado em Cock burn, 2006. Ele implementou uma Linha de Atraso Programável através de um multiplexador (Mux) que controlava 256 linhas as quais eram selecionadas pelo seletor de tamanho da linha de atraso. Apesar de conceitualmente ser simples a implementação na prática é dificultada pelo grande número de elementos. Uma grande cadeia implica em interferências de ruídos parasitas e problemas de descasamento. A Figura 6.38 ilustra o sistema de linha programável.

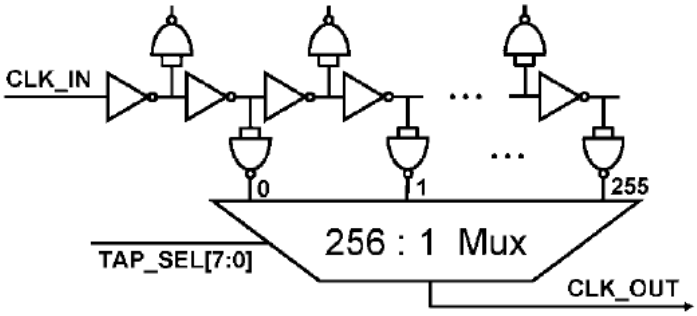

Figura 6.38. Linha de Atraso Programável proposta por Cockburn.

Outras arquiteturas propõem a diminuição do tamanho da linha por meio de Sistemas Grosso e Fino (*course, fine*), de forma que o Grosso faz a seleção a passos mais largos e o Fino faz os ajustes finais do sistema. Essa separação pode permitir a facilidade de estabelecimento do sistema bem como uma melhor escolha do ponto em que é necessário o emprego de um ou outro tipo de célula de atraso levando em consideração as características dos circuitos CMOS, tais como o tamanho, o consumo e a geração de ruídos (CARDOSO, 2009; OH, 2008; WANG, 2006).

Devido a limitação da tecnologia será implementado a princípio uma resolução de 10ηs usando arranjos de portas NOT e *inversor Tri-State*. Estes blocos lógicos são selecionados por cada uma das linhas do contador usando para isso um Mux 2 x 1. Assim, pode-se implementar um sistema em que cada bit do contador implementasse o dobro de atraso do bit seguinte, desta forma, o primeiro bit S0 implementaria atraso de 20ηs, o segundo (S1) de 40 ηs e assim sucessivamente até o bit S7 que introduz atraso de 2,56 us. A seleção em cada bit é feita através de um multiplexador que permite ou não a introdução do atraso. A Figura 6.39 demonstra o circuito do processo de seleção de atraso.

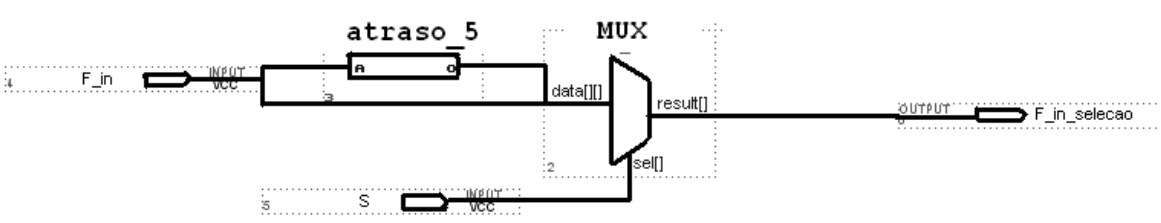

Figura 6.39. Sistema de Seleção de Atraso usando Mux.

Os tempos de atraso das portas lógicas da ferramenta da Altera Max Plus II apresentam muitas diferenças em relação à tecnologia CMOS 350nm. Para usar tempos de atraso semelhante foi utilizado do recurso de Lcell que são células atrasadoras disponibilizadas pela ferramenta que introduzem atraso próximo a 4 ηs. Para gerar um atraso de 20 nano segundos foi implementado uma cadeia de 5 atrasadores para efeitos de simulação de comportamento do sistema.

Colocando os Mux e os atrasadores em cadeia, pode-se assim gerar a Linha Programável de Atraso com o peso ponderado em cada bit.

Assim quando o contador apresentar por exemplo, o número D ‘125’ que corresponde a B’01111101’ tem-se atraso total de 2,5us. Os atrasos neste caso não estão levando em consideração os atrasos inerentes as portas lógicas do Mux e demais circuitos tais como o próprio contador. Na implementação de um sistema de 8 bits precisa-se implementar 8 células com atrasos selecionáveis por chaves Mux e isso em cascata sendo que para cada mudança no bit mais significativo o tempo de atraso selecionável dobra. A Figura 6.40 apresenta uma linha de atraso programável de 8 bits. Nas simulações foram encontrados problemas de linearidade na somatória dos atrasos principalmente nos primeiros bits de forma que os bits de menor valor apresentassem valores de atraso mais significativos do que o ideal e esse erro era reduzido conforme era a relação entre os bits mais significativos. Esse tipo de erro provavelmente se deve a não idealidades dos roteamentos implementados pelas ferramentas de sínteses.

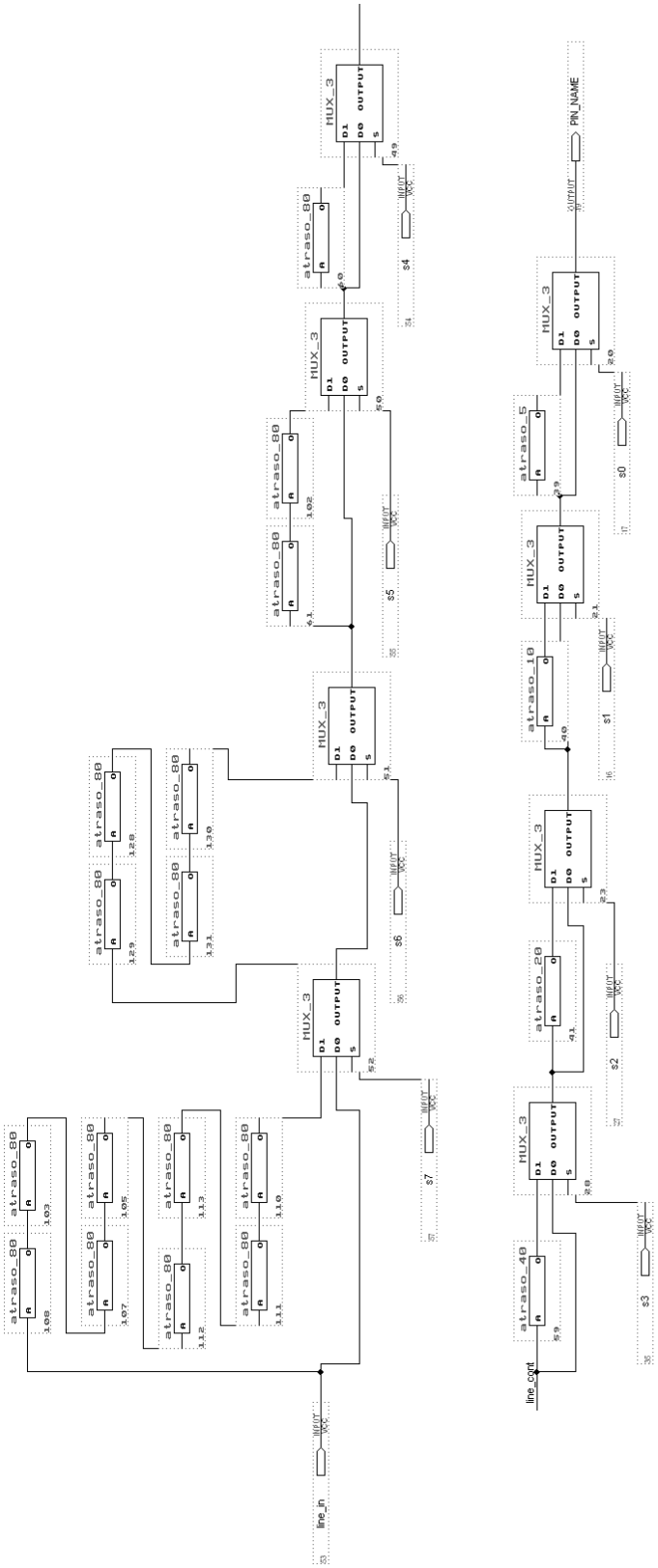

Figura 6.40. Montagem da Linha de Atraso Programável de 8 bits.

A simulação da Linha de Atraso Programável é bastante complexa tendo em vista a análise que é a soma de atrasos dos oito bits. Mas para ilustrar o comportamento, a Figura 6.17

mostra a simulação para quando os bits s7, s5, s3, s1 estão em nível alto (1) e introduzem atraso. Pela simulação o total de atraso introduzido pela linha foi de 2,1171us.

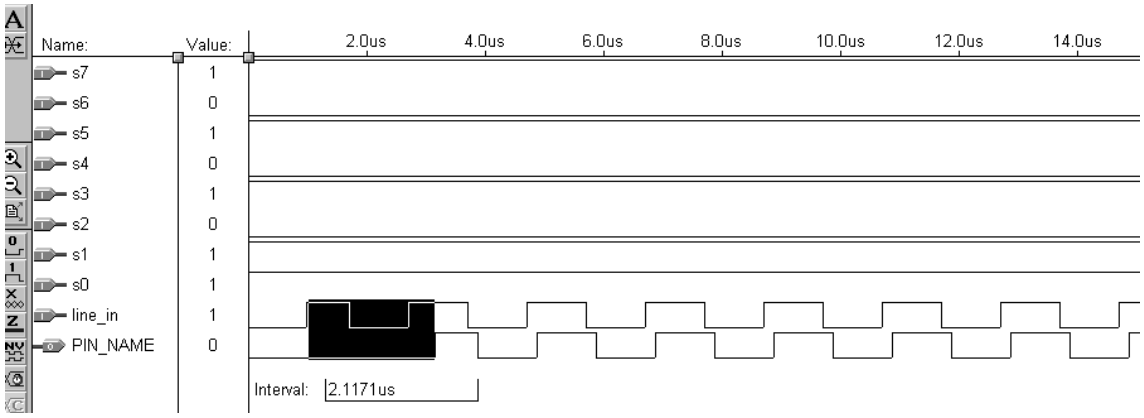

Figura 6.41. Simulação da Linha de Atraso Programável.

Para gerar o sistema completo do DLL cada subprograma precisa ser representado por um símbolo. A Figura 6.42 ilustra o símbolo gerado para a Linha de Atraso Programável.

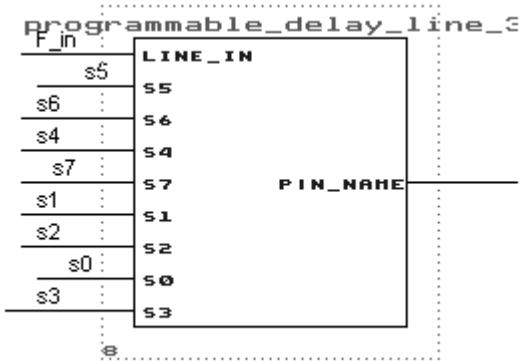

Figura 6.42. Símbolo Criado para a Linha de Atraso Programável.

Então a Figura 6.43 apresenta o Sistema DLL digital CMOS.

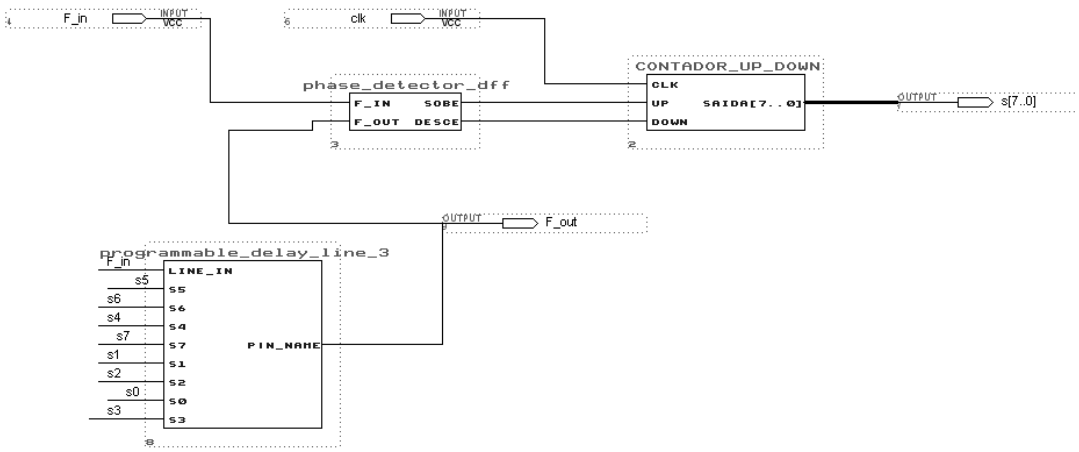

Figura 6.43. Arquitetura Implementada Modelo Digital DLL.

No próximo capítulo serão detalhados os resultados do DLL proposto.

7 RESULTADOS SIMULADOS E EXPERIMENTAIS

Para introdução ou equalização de fase entre dispositivos periféricos ou memórias são utilizados circuitos DLL ou PLL na faixa de frequência de 100 a 500 MHz (RAMBUS, 2004). A faixa de frequência do DLL proposto foi limitada pela tecnologia do dispositivo FPGA. Desta forma o sistema implementado opera entre 1 e 100 MHz.

Neste capítulo serão abordadas as etapas de simulação implementação, validação do circuito DLL e os resultados do DLL proposto.

7.1 SIMULAÇÃO, IMPLEMENTAÇÃO E VALIDAÇÃO DO SISTEMA DLL

Para realizar a simulação foi determinado que o clock do sistema (define a frequência de amostragem do detector de fase) atuasse com múltiplos de 4ηs porque não foi possível simulações que funcionassem em valores menores que este.

Também como parâmetro de simulação foi definido que a frequência F_ref não poderia ser maior que 2/3 da frequência do clock.

Vários casos merecem destaque para a simulação do DLL. Como dito anteriormente a ferramenta de trabalho não parece funcionar com clock menor que 4ηs então foi adotado **clk =** 4ηs e verificado se o sistema consegue entrar em sincronismo.

Na Figura 7.44 (denominado de Caso 1) tem-se que F_out inicia inativo e após alguns períodos o sistema entra em sincronismo. O contador contou de forma crescente até o numero 2 e estabilizou neste valor. F_in e F_out não estão alinhados exatamente em fase mas, isso deve ser devido aos atrasos de propagação das demais células e também da impossibilidade do sistema em introduzir um menor deslocamento de fase. Neste caso existe um sincronismo fraco porque

em alguns ciclos o sistema acaba acumulando erro de fase o que implica que F_out vai ficar oscilando ente um e outro valor de frequência.

As simulações funcionaram bem com um período de 10 ns para F_in o que implica que tem-se uma frequência máxima de operação de 100 MHz. Muito provavelmente o sistema poderia alcançar maiores frequências, mas esta ferramenta impossibilitou esta análise. A família de componentes utilizada foi a Cyclone II, Por ser o componente da placa de teste disponível.

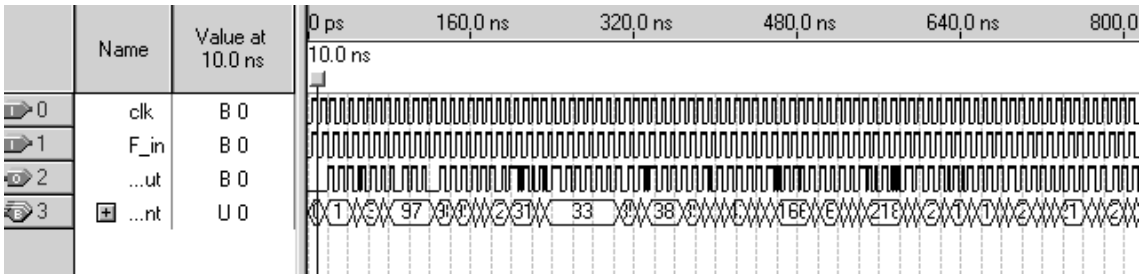

Figura 7.44. Caso 1. Simulação na Máxima Frequência Possível.

Por outro lado, em baixas frequências é esperado que o sistema demore mais para se estabelecer devido à procura mais demorada do tamanho do atraso que deve ser introduzido para gerar a frequência F_out correspondente. E também é previsível que a partir de certa freqüência, mesmo que todas as células de atraso estejam sendo ativadas não seriam suficientes para gerar o atraso (BYUNG, 2005. O Caso 2 mostrado na Figura 7.45 tem-se os seguintes dados: F_in tem período de ativo de 630ηs o que implica em um período total de 1260ηs e **clk** de período de 400 ηs. Assim tem-se a frequência de 793.65kHz. Fazendo um cálculo aproximado do número de ciclos que foram necessários para o sistema entrar em sincronismo tem-se: 105μs/ 1260μs = 83 ciclos.

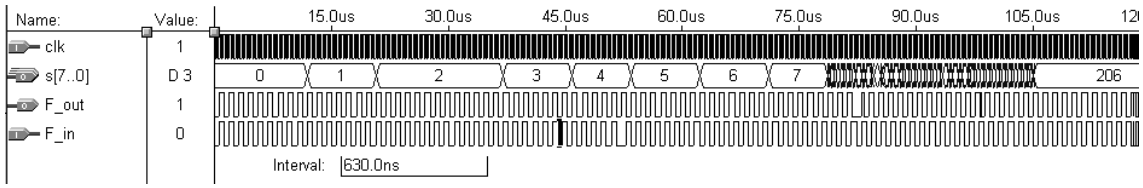

Figura 7.45. Caso 2. Mínima frequência que o DLL entra em sincronismo.

Todas as simulações com valor do período de F_in maiores que 630 ηs o sistema apresentou dificuldade em entrar em sincronismo fazendo-se uma busca por todas as possibilidades de Linha de Atraso por várias vezes.

Existem outras situações que o sistema entra em um fraco sincronismo e depois de algumas dezenas de ciclos tem-se o restabelecimento do sincronismo de forma definitiva. Este fato se deve a existência de algum pequeno atraso entre os dois sinais e que após vários ciclos são acumulados de forma a acionar o detector de fase novamente o que implica na alteração da contagem e busca por uma nova posição de sincronismo. No caso mostrado na Figura 7.46 tem-se F_in com período de 1300 ηs e **clk** com período de 540 ηs.

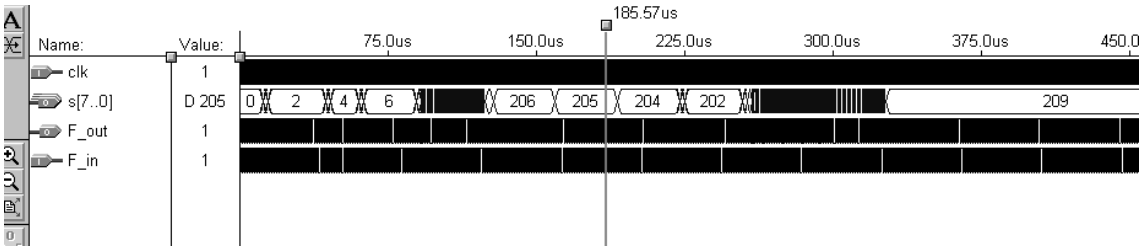

Figura 7.46. Falso Sincronismo e Restabelecimento do Sincronismo.

Em outro exemplo uma simulação típica em 25 MHz em que foram necessários poucos ciclos de clock para o estabelecimento do sinal. Na frequência de operação de 2 MHz, é possível verificar que o sistema converge em menos de 1.28 μs e que a partir de então a frequência de saída acompanha o sinal de entrada com uma defasagem de fase.

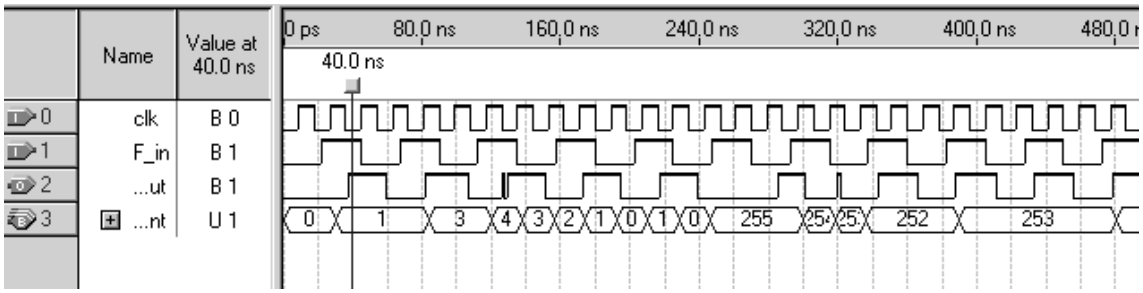

Figura 7.47. Simulação típica DLL em 25 MHz.

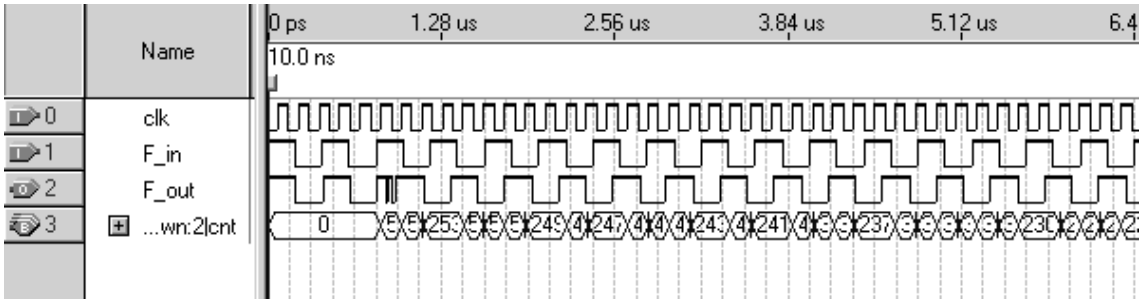

Figura 7.48. Simulação típica DLL em 2 MHz.

7.2 IMPLEMENTAÇÃO SISTEMA DLL NO KIT DE DESENVOLVIMENTO

Utilizando o kit de desenvolvimento Cyclone II FPGA Starter Development pode-se implementar o circuito DLL em duas configurações. O circuito DLL_2 é um DLL constituído de uma linha de atraso programável com 8 elementos e a menor unidade de atraso duas Lcell. E o circuito DLL_1 foi implementado com menor unidade de atraso possível, uma unidade Lcell. Os circuitos apresentaram praticamente as mesmas respostas e também os equipamentos de medição e geração de sinal impossibilitaram verificar todo o espectro de trabalho dos circuitos DLL. Como fonte de sinal foram utilizados um gerador de sinais Sony - Tektronics AFG310 que gera sinais retangulares até 15 MHz e os sinais de clock interno disponíveis no próprio Kit de desenvolvimento de 50 MHz, 27 MHz e 24 MHz. A Figura 7.49 mostra como foi o processo de aquisição de sinais da Placa Cyclone II FPGA Starter Board.

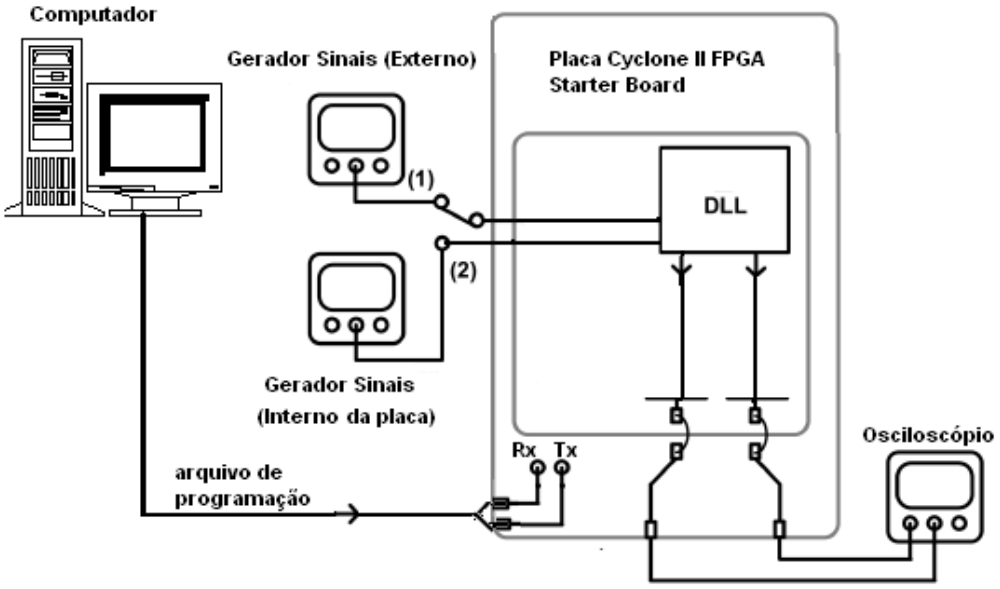

Figura 7.49. Esquema do processo de aquisição de sinais.

A Figura 7.50 mostra como foi realizado o de teste do DLL. O computador envia os arquivos de configuração através de um cabo USB. A placa do Kit utiliza de dois sinais, sendo o

mais rápido utilizado como referência. O próprio clock interno podendo ser de 50 MHz, 27 MHz ou 24 MHz. O segundo sinal pode ser oriundo de um gerador de sinais externo ou mesmo um clock interno desde que seja mais lento do que o primeiro sinal. A placa do kit também apresenta um barramento em se selecione qual pino de saída do sistema. E por fim, são feitas as medições através do osciloscópio Tektronics.

Figura 7.50. Montagem sistema.

A Figura 7.51 mostra o canal 1 o sinal gerado pelo gerador de sinais e o canal 2 a saída do DLL. A Figura 7.52 mostra o sinal F_in em 16 MHz e com os sinais sobrepostos para facilitar a visualização do efeito de deslocamento de fase.

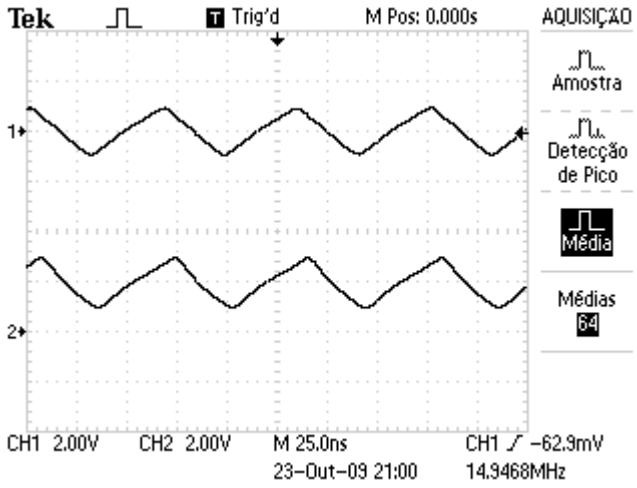

Figura 7.51. Implementação DLL_1 a 15 MHz.

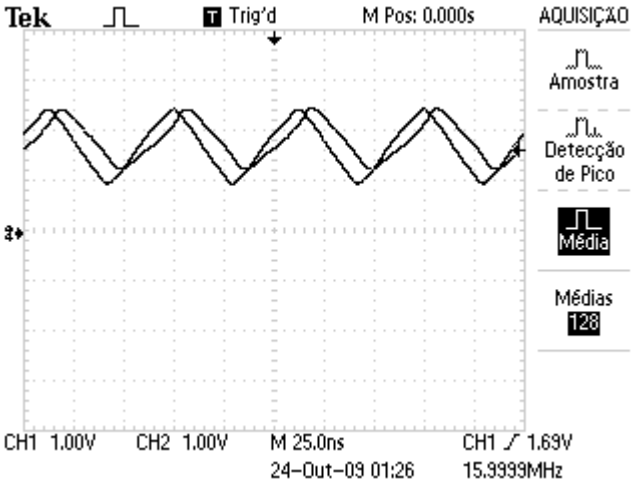

Figura 7.52. Implementação DLL_1 Efeito do deslocamento de fase.

O sistema também utilizou do clock interno da placa para fazer a implementação do DLL. Um sinal de 50 MHz foi utilizado com clock do sistema e o sinal de 21.1 MHz como sinal de referência como mostrado na Figura 7.53.

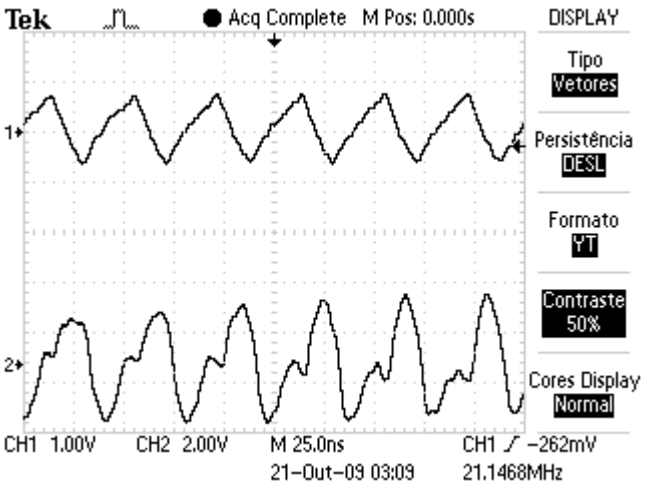

Figura 7.53. Implementação DLL_1 a 21.1 MHz.

A Figura 7.54 mostra a implementação do DLL em uma frequência de 2 MHz. Esta implementação valida o resultado da simulação em 2 MHz da Figura 7.48. Espera-se por esta comparação que as demais simulações sejam validadas por analogia porque em frequências mais

altas as implementações mostraram imagens bastante senoidais possivelmente por problemas na aquisição de dados (o osciloscópio trabalha no máximo a 100 MHz).

Nota-se que para sinais com frequência menor apresentam pouca distorção em relação a uma onda quadrada.

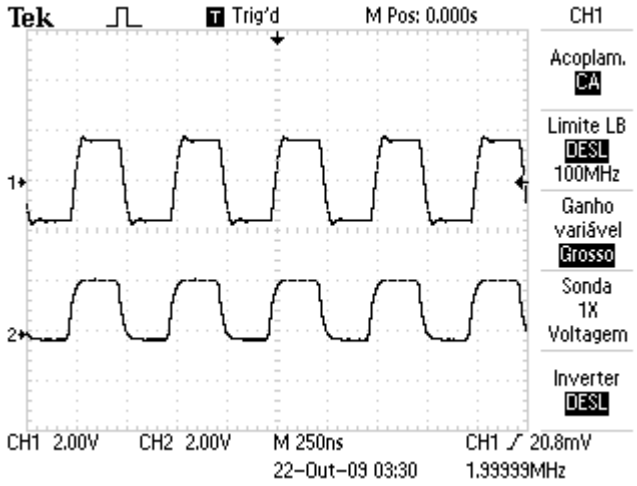

Figura 7.54. Implementação DLL_1 a 2 MHz.

7.3 CONCLUSÕES

O circuito DLL proposto apresenta as seguintes características:

- Portabilidade; por ser um circuito digital pode ser implementado em diversas tecnologias adaptando somente o tempo da célula de atraso.
- Sistema de detecção baseado em amostragem que facilita o processo de convergência e diminui o tempo de estabelecimento.
- Ciclo ativo próximo a 50% o que facilita o interfaceamento com outros sistemas.
- Número de bits, ou resolução pode ser facilmente modificado para trabalhar em outras faixas de frequência.
- Nova arquitetura da Linha de Atraso Programável que facilita a interface com o contador (acumulador).

Frequência de trabalho entre 1 MHz e 100 MHz para a família de FPGA proposta.

Tempo máximo de sincronismo 83 ciclos a 1 MHz (aproximadamente).

A arquitetura proposta mostrou-se validada por apresentar características semelhantes a trabalhos implementados fisicamente em tecnologia CMOS mesmo sendo que o trabalho ficou limitado as características do dispositivo FPGA.

O dispositivo DLL apresentou os requisitos para a partir de um sinal de clock fazer o reconhecimento e estabelecimento do sinal e gerar a fase necessário para que os sinais entrem em sincronismo numa ampla faixa de frequência.

Na Tabela 7.1 tem-se uma comparação de desempenho dos artigos com arquitetura semelhante. Sistemas implementados com tecnologia CMOS analógica apresentaram maior tempo de sincronismo em relação aos sistemas digitais.

Tabela 7.1 – Comparação de Desempenho.

	Este Trabalho	**Lip Kai 2007**	**Hsiang Hui 2002**	**Cockburn 2006**
Tecnologia e Processo	350nm CMOS Digital (Simulado)	180nm CMOS Analógico	350nm CMOS Analógico	180nm CMOS Digital
Tensão	3.3V	1.8V	3.3V	2.0V
Frequência de Operação	1-100MHz	50 – 150 MHz	100 MHz	14 -250 MHz
Tempo de Sincronismo	<2µs	2µs em 100 MHz	1130 ciclos	963 ciclos em 14 MHz E 89 em 166 MHz
Ciclo ativo 50%	Sim	Não	Não	Sim

Devido às restrições a equipamentos, não foi possível fazer testes em frequências próximas aos limites do FPGA. O gerador de sinais com frequência mais alta gera sinais de 27 MHz e o osciloscópio pode fazer aquisições de até 100 MHz.

8 CONCLUSÕES E DISCUSSÕES

O sistema DLL proposto propõe uma nova arquitetura baseado em uma nova Linha de Atraso Programável. A principal inovação foi a utilização de um contador associado diretamente e proporcionalmente a cada bit fazendo que o tempo de sincronismo fosse bastante reduzido.

A arquitetura da linha de atraso programável provou-se eficiente na geração rápida do atraso a ser implementado. O sistema entrou em sincronismo em muito poucos ciclos apresentando desempenho semelhante ao proposto por outros autores que utilizam diferentes tecnologias.

O dispositivo DLL apresentou os requisitos para a partir de um sinal de clock fazer o reconhecimento e estabelecimento do sinal e gerar a fase necessária para que os sinais entrem em sincronismo numa ampla faixa de frequência.

Com os dispositivos FPGA da família Cyclone foi possível a implementação e ajustes para o sistema operar em pelo menos 100 MHz. A faixa de frequência apresentou-se bastante ampla devido ao número de bits utilizado na arquitetura da linha de atraso programável.

O sistema DLL proposto apresentou ciclo ativo de 50% o que implica na não necessidade de circuitos auxiliares para se fazer o interfaceamento com outros dispositivos.

A utilização de sistema baseado em HDL (linguagens de descrição de hardware) permite uma rápida reconfiguração dos parâmetros do circuito para novas avaliações do sistema.

A utilização de arquiteturas digitais se mostrou eficiente de forma a apresentar melhor tempo de sincronismo em relação aos dispositivos analógicos conforme é mostrado pela Tabela 7.1

Para avaliar alguns parâmetros de desempenho é necessário a implementação física do dispositivo CMOS e utilização de equipamentos sofisticados para avaliação *jitter* e outros dados.

Outro ponto a ser avaliado no sistema é fazer implementações de outras arquiteturas de detectores de fase para avaliação de desempenho de ganho e tempo de chaveamento.

Blocos podem também ser adicionados para a detecção do sincronismo e para a facilitação e seleção da faixa de trabalho e assim estabelecimento do sinal.

De modo geral o circuito DLL proposto satisfaz as condições de operação de pelo menos 100 MHz e não foi possível medir o *jitter* mas espera-se valores baixos devido ao bom comportamento das amostragens do sinal no tempo.

9 TRABALHOS FUTUROS

Este trabalho pode ser complementado com a implementação física do dispositivo em tecnologia CMOS. O sistema não foi implementado devido as dificuldades de implementação e de compatibilidade entre os arquivos de projetos por algum problema de configuração das ferramentas de síntese (MENTOR e suas entradas para arquivos EDIF e VHDL).

Espera-se que com novas tecnologias possa-se reduzir drasticamente a resolução de fase do sistema de forma a se alcançar frequências de operação em torno de GHz.

Sistemas Híbridos PLL e DLL ou Dual DLL abordam características de duas arquiteturas de forma a desenvolver um sistema mais robusto e mais completo. Uma futura aplicação dos DLL são sistemas dual DLL ou híbridos com o PLL de forma a operar em sistemas de alta frequência e baixíssimo *jitter*. E sistemas PLL ou DLL quando podem operar em frequências maiores que 300 MHz podem trabalhar com recuperação de clock para memórias de computadores.

REFERÊNCIAS BIBLIOGRÁFICAS

ALTERA CORPORATION. **Cyclone II FPGA starter development board:** reference manual. Version 1.0. 2006. Disponível em: <http://www.altera.com/literature/manual/mnl_cii_starter_board_rm.pdf>. Acesso em: 01 jan. 2009.

ALTERA CORPORATION. **MAX.+PLUS II getting started.** Version 8.1. 1997. Disponível em: <www.altera.com/literature/manual/81_gs.pdf>. Acesso em: 01 jan. 2004.

APPEL, G. Fractional N synthesizers: analyzing fractional N synthesizers and their ability to reduce phase noise, improve loop speed and reduce reference spur levels. **RF Design magazine, RF Signal Processing,** p. 34–50, nov. 2000.

BAE, S.-J. et al. A VCDL-based 60-760Mhz Dual Loop DLL with Infinite Phase-shift Capability and Adaptive Bandwidth Scheme. **IEEE Solid State Circuits**, v. 40, n. 5, p. 1119-1129, may 2005.

CHANG, H.-H. et al. A wide-Range Delay-Locked Loop with a Fixed Latecne of One Cock Cycle. **IEEE Journal of Solid State Circuits,** v. 37, n. 8, p. 1021-1027, ago. 2002.

CHIEN, G.; GRAY, P. R. A 900 MHz local oscillator using a DLL-based frequency multiplier technique for PCS applications. **IEEE Journal of Solid State Circuits,** v. 35, n. 12, p. 1996-1999, dez. 2000.

COCKBURN, B. F.; BOYLE, K. Design and characterization of a Digital Delay Locked Loop Synthesized from Black Box Standard Cells. **IEEE Canadian Conference on Electrical And Computer Engineering,** Ontawa, p. 1214-1217, may 2006.

CRANINCKX, J.; STEYAERT, M. A Fully Integrated CMOS DCS-1800 Frequency Synthesizer. **IEEE Journal of Solid-State Circuits,** v. 33, n. 12, dec. 1998.

GALLUP, W. B. et al A portable Digital DLL for High-speed CMOS interface circuits. **IEEE Journal of Solid State Circuits,** v. 34, n. 5, p. 632-644, may 1999.

JIA, C. **A Delay locked loop for multiple clock phases delay generation.** 2005. 83 f. Tese (Doutroado) - Georgia Institute of Techology, 2005.

KIM, B. **High speed clock recovery in VLSI using hybrid analog/digital techniques.** 1990, Tese (Doutorado) - University of California, Berkeley, 1990.

KIM, B.-G.; KIM, L.-S. A 250 MHz – 2G Hz wide range delay locked loop. **IEEE Journal of Solid State Circuits,** v. 40, n. 6, p. 1310-1321, jun. 2005.

LEE, T. H. **The Design of CMOS radio-frequency integrated circuits.** Cambridge: Cambridge University, 1988.

LIP, K.; SULAIMAN, M.-S.; YUSOFF, Z. Fast-Lock Dual Charge Pump Analog DLL using Improved Phase Frequency DetectorVLSI Design, Automation and Test, 2007. In: INTERNATIONAL SYMPOSIUM ON VLSI DESIGN, AUTOMATION, AND TEST, 2007, Hsin-chu shih, Taiwan. **2007 International…** Hsin-chu: IEEE, 2007. p. 25-27.

MAGIEROWSKI, S. K.; ZUKOTYNSKI, S. CMOS LC-Oscillator Phase-Noise Analysis Using Nonlinear Models. **IEEE Trans. Circuits Syst. I:** analog and digital signal processing, v. 51, n. 4, p. 664–667, apr 2004.

OH, K. I. et al. Low-Jitter multi-phase digital DLL with closest edge selection scheme. **Eletronic Letters,** v. 44, n. 19, sep. 2008. Disponível em: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04625175>. Acesso em: 01 jan. 2009.

OLIVEIRA, V. J. S; CARDOSO, A. S.; OKI, N. A Dual Path Frequency Synthesizer using a Hybrid Analog Digital Loop filter. In: CONFERENCE ON DESIGN OF CIRCUITS AND INTEGRATED SYSTEMS, 23., Grenoble, France, 2008. **DCIS 2008…** Grenoble: France: DCIS, 2008.

RAMBUS CORPORATION. **DLL/ PLL on a DRAM**. 2009. Disponível em: < http://www.rambus.com/us/patents/innovations/detail/dll_pll.html >. Acesso em: 01 mar. 2005.

RAZAVI, B. A study of Phase Noise in CMOS Oscillators. **IEEE Journal of Solid-State Circuits**, v. 31, p. 331–343, mar. 1996.

RAZAVI, B. **Design of analog CMOS integrated circuits**. New York: McGraw-Hill, 2000.

RAZAVI, B. **RF Microelectronics.** Upper Saddle River, NJ: Prentice-Hall, 1998.

SHU, K. et al. A 2.4-GHz Monolithic Fractional-N Frequency Synthesizer With Robust Phase-Switching Prescaler and Loop Capacitance Multiplier. **IEEE Journal of Solid-State Circuits**, v. 38, n. 6, p. 866-874, jun. 2003.

SHU, K.; SÁNCHEZ-SINENCIO, E. **CMOS PLL Synthesizers**: analysis and design. New York: Kluwer, 2005.

SILVA, A. C. R., CARDOSO, A. S. An environment to aid the synthesis of three phase analogue waveform using AHDL. In: SYMPOSIUM ON INTEGRATED CIRCUITS AND SYSTEMS DESIGN, 14., Pirenopolis, 2001. **SBCCI 2001…** Pirenopolis: SBCCI, 2001. p.142-147.

STASZWSKI, R. B. et al. A Digitally Controlled Oscillator in a 90 nm Digital CMOS Process for Mobile Phones. **IEEE Journal of Solid State Circuits**, v. 40, n. 11, p. 2203-2211, nov. 2005.

XILINX INCORPORATED. **Virtex Delay-Locked Loop (DLL):** VTT003 (v1.1) 2000. Disponível em: <http://www.ece.umd.edu/courses/enee759h.S2003/references/vtt003.pdf>. Acesso em: 01 jan. 2009.

ZARKESHVARI, F.; NOEL, P.; KWASNIEWSKI, T. PLL-Based Fractional-N Frequency Synthesizers. In: INTERNATIONAL WORKSHOP ON SYSTEM-ON-CHIP FOR REAL-TIME APPLICATIONS, 5., Banf, Alberta, Canada, 2005. **Proceedings of the…** Washington: IEEE, 2005, p. 85-91.

WILLIAM, C.; L.; CHIE, C. M. A Survay of digital phase-locked loop. **Proceedings of IEEE,** v. 69, n. 4, apr. 1981.