



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE ENERGIAS ALTERNATIVAS E RENOVÁVEIS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

BRUNO FERNANDES DOS SANTOS

**Proposta de Arquitetura de Geração de Relógio
Adaptativo Aplicada às Técnicas de
Escalonamento Dinâmico de Tensão e
Frequência na Tecnologia FDSOI-28 nm**

João Pessoa
Outubro de 2018

Bruno Fernandes dos Santos

**Proposta de Arquitetura de Geração de Relógio
Adaptativo Aplicada às Técnicas de Escalonamento
Dinâmico de Tensão e Frequência na Tecnologia
FDSOI-28 nm**

Monografia submetida à Coordenação do
Curso de Engenharia Elétrica da Universidade
Federal da Paraíba como parte dos requisi-
tos para obtenção do título de Engenheiro
Eletricista.

Universidade Federal da Paraíba
Centro de Energias Alternativas e Renováveis
Curso de Graduação em Engenharia Elétrica

Orientador: Prof. Dr. Cleonilson Protásio de Souza

João Pessoa
Outubro de 2018

Catálogo na publicação
Seção de Catalogação e Classificação

S237p Santos, Bruno Fernandes Dos.

Proposta de Arquitetura de Geração de Relógio Adaptativo Aplicada às Técnicas de Escalonamento de Tensão e Frequência na Tecnologia FDSOI-28nm / Bruno Fernandes Dos Santos. - João Pessoa, 2018.

71 f.

Orientação: Cleonilson Protásio de Souza.
Monografia (Graduação) - UFPB/CEAR.

1. Escalonamento Dinâmico de Tensão e Frequência. I. de Souza, Cleonilson Protásio. II. Título.

UFPB/BC

Bruno Fernandes dos Santos

**Proposta de Arquitetura de Geração de Relógio
Adaptativo Aplicada às Técnicas de Escalonamento
Dinâmico de Tensão e Frequência na Tecnologia
FDSOI-28 nm**

Monografia submetida à Coordenação do
Curso de Engenharia Elétrica da Universidade
Federal da Paraíba como parte dos requisitos
para obtenção do título de Engenheiro
Eletricista.

Trabalho aprovado em 29 de outubro de 2018:

Prof. Dr. Cleonilson Protásio de Souza
UFPB
Orientador

**Prof. Dr. José Maurício Ramos de Souza
Neto**
UFPB
Avaliador Interno 1

Prof. Ms.C. Rafael de Sousa Marinho
UFPB
Avaliador Interno 2

João Pessoa
Outubro de 2018

Dedico este trabalho aos meus pais, Maria Augusto e Orlando Benedito.

Agradecimentos

Aos meus pais, Maria Augusto e Orlando Benedito, pelo apoio incomensurável e constante durante toda a minha trajetória nessa breve existência.

À minha família, especialmente minha irmã, Kallina Fernandes, pelo carinho e paciência durante as intempéries intrínsecas da vida.

À todos os meus professores que fizeram parte da construção da minha personalidade e do meu conhecimento, em especial àqueles que moldaram minhas competências mais fundamentais, como a leitura e escrita.

Aos meus amigos que trilharam ao meu lado as diversas etapas da vida até aqui, os amigos são a família que podemos escolher.

Aos colegas de classe, em destaque à Henning, Filipe Vidal e Matheus Silveira, que compartilharam dos aprendizados e desafios de uma graduação em engenharia.

Aos meus orientadores das iniciações científicas e desta monografia, em especial o Prof. Juan Villanueva, primeiro orientador cujo suporte me possibilitou alcançar a primeira publicação em congresso nacional.

Aos meus avaliadores, Prof. José Maurício e Prof. Rafael Marinho, por aceitarem compor a banca avaliadora, colaborando consideravelmente assim com a confecção deste trabalho.

Aos meus colegas de equipe na NXP Semiconductors, pelo apoio técnico, compreensão, paciência em transmitir conhecimento durante suas atividades, assim como pela experiência de trabalhar em um time tão reconhecido e valorizado internacionalmente.

Aos demais amigos e todos que fizeram parte dessa caminhada.

*Há sempre alguma loucura no amor.
Mas há sempre um pouco de razão na loucura.
Friedrich Nietzsche, Assim Falava Zaratustra*

Resumo

Como a frequência de operação dos processadores atuais vem crescendo consideravelmente, tendo como consequência direta o aumento do consumo, atualmente, com o desenvolvimento e necessidade de circuitos eletrônicos de baixo consumo, particularmente no contexto de Internet das Coisas e Redes de sensores em fio, técnicas de Escalonamento de Tensão e Frequência (DVFS, *Dynamic Voltage and Frequency Scaling*) estão cada vez mais sendo utilizadas visando a redução do consumo de energia e aumento do tempo de vida dos circuitos que, normalmente, são alimentados por bateria. Este trabalho tem como objetivo o estudo e desenvolvimento de uma arquitetura de geração de relógio adaptativo no contexto de DVFS e sua demonstração de funcionamento por meio de simulação. A arquitetura proposta e desenvolvida contém três blocos principais: *Delay Locked Loop*, um controlador digital e uma réplica do caminho crítico do circuito integrado. O Delay Locked-Loop é o módulo analógico do sistema, enquanto que a geração e controle do sinal de relógio são realizados digitalmente. Sendo assim, utilizou-se duas ferramentas para a implementação e verificação da proposta, uma para análise de sistemas analógicos (Virtuoso da Cadence Inc.) e a outra para simulação e sintetização de circuitos digitais (Genus da Cadence Inc.). A tecnologia disponibilizada e empregada neste trabalho foi a FDSOI-28 nm. Simulações analógicas e funcionais foram efetuadas e a verificação das propriedades da arquitetura desenvolvida, tanto a nível funcional como a nível de implementação física, incluindo a análise de área e consumo de energia para o controlador digital, comprovam a viabilidade de proposta apresentada.

Palavras-chave: Redes de Sensores Sem Fio, Escalonamento Dinâmico de Tensão e Frequência, Relógio Adaptativo, Consumo de Energia.

Abstract

Due to the considerable increase of the operation frequency of the current processors, which results in an increase of the energy consumption, currently, with the development and necessity of low consumption electronic circuits, specially in the context of Internet of Things and Wireless Sensor Networks, Dynamic Voltage and Frequency Scaling techniques are becoming more popular with the goal of achieving power and energy consumption reduction, as well a increase of battery life. This work has as objective the study and development of a clock generation architecture applied to DVFS systems and its consequent demonstration through simulations. The proposed and developed architecture has three main blocks: Delay Locked Loop, a digital controller and a critical path replica of the integrated circuit. The *Delay Locked Loop* is the analog block, while the generation and control of the clock signal is made digitally. Therefore, it was used two software tools to implementation and verification of the proposal, one for the analysis of analog systems (Cadence Virtuoso) and another to simulation and synthesis of digital circuits (Cadence Genus). The available technology for this work was the FDSOI-28 nm. Functional and analog simulations were performed and the properties verification of the developed architecture, not only on the functional level as well on the physical implementation, including the area and power consumption evaluation for the digital controller, prove the feasibility of the presented architecture.

Keywords: Wireless Sensor Network, Dynamic Voltage and Frequency Scaling, Adaptive Clock, Energy Consumption.

Lista de ilustrações

| | |
|--|----|
| Figura 1.1 – Arquitetura geral de uma rede de comunicação oriunda da proposta da Internet das Coisas. | 26 |
| Figura 1.2 – Diagrama de blocos representativo de um nó sensor. | 27 |
| Figura 1.3 – Representação em diagrama de blocos de uma arquitetura para implementação de DVFS. | 30 |
| Figura 1.4 – Consumo de energia (em Joules) de cada componente para cada estratégia verificada. | 30 |
| Figura 2.1 – Esquemático para um inversor CMOS tradicional. | 34 |
| Figura 2.2 – Visão transversal da estrutura de um inversor CMOS em silício. | 35 |
| Figura 2.3 – Relação entre a potência dinâmica e a frequência de operação sem e com ajuste correlato da tensão de alimentação. | 36 |
| Figura 2.4 – Ilustração da operação de DVFS para a execução de uma tarefa. | 37 |
| Figura 2.5 – Modelo de primeira ordem para a descarga do capacitor de chaveamento durante uma transição de alto para baixo. | 38 |
| Figura 3.1 – Visão geral em diagrama de blocos da arquitetura de gerador de relógio adaptativo. | 43 |
| Figura 3.2 – Representação ilustrativa do sinal de relógio de acordo com a tensão de alimentação. | 44 |
| Figura 3.3 – Estrutura em diagrama de blocos do <i>Delay Locked-Loop</i> | 45 |
| Figura 3.4 – Esquemático do elemento de atraso utilizado no VCDL. | 46 |
| Figura 3.5 – Diagrama de blocos do sistema de primeira ordem do DLL. | 47 |
| Figura 3.6 – Circuito do Detector de Fase e Frequência (PFD). | 47 |
| Figura 3.7 – Comportamento funcional do PFD representado por seu diagrama de tempo. | 47 |
| Figura 3.8 – Esquemático do <i>charge pump</i> com o filtro passa-baixa incluído. | 48 |
| Figura 3.9 – Esquemático da réplica configurável do caminho crítico. | 49 |
| Figura 3.10 – Circuito do gerador de relógio do controlador. | 50 |
| Figura 3.11 – Diagrama de tempo para uma operação usual do controlador do gerador de relógio. | 51 |
| Figura 4.1 – Esquemático da implementação do caminho de atraso controlado por tensão. | 53 |
| Figura 4.2 – Variação do atraso da saída do VCDL com a tensão de controle. | 54 |
| Figura 4.3 – Esquemático da implementação da malha de realimentação do DLL. | 55 |
| Figura 4.4 – Geração dos sinais de controle UP e DN do detector de fase e frequência - saída atrasada em relação à referência. | 55 |

| | |
|--|----|
| Figura 4.5 – Geração dos sinais de controle UP e DN do detector de fase e frequência - saída adiantada em relação à referência. | 56 |
| Figura 4.6 – Obtenção da tensão de controle pelo <i>charge pump</i> e filtro passa-baixa. . . | 56 |
| Figura 4.7 – Esquemático da implementação do controlador do gerador de relógio adaptativo. | 57 |
| Figura 4.8 – Sinal de relógio adaptativo gerado pela arquitetura proposta. | 59 |

Lista de tabelas

| | |
|--|----|
| Tabela 4.1 – Resultados para os parâmetros físicos do controlador digital. | 57 |
|--|----|

Lista de abreviaturas e siglas

| | |
|--------|--|
| IoT | <i>Internet of Things</i> |
| RSSF | Redes de Sensores Sem Fio |
| M2M | <i>Machine-to-machine communication</i> |
| A/D | Analógico para Digital |
| IEEE | <i>Institute of Electrical and Electronics Engineers</i> |
| kbps | <i>kilobits per second</i> |
| DVFS | <i>Dynamic Voltage and Frequency Scaling</i> |
| CMOS | <i>Complementary Metal Oxide Semiconductor</i> |
| MIPS | <i>Million Instructions per second</i> |
| FDSOI | <i>Fully Depleted Silicon On Insulation</i> |
| PMOS | <i>P-type Complementary Metal Oxide Semiconductor</i> |
| NMOS | <i>N-type Complementary Metal Oxide Semiconductor</i> |
| DFS | <i>Dynamic Frequency Scaling</i> |
| MOSFET | <i>Metal Oxide Semiconductor Field Effect Transistor</i> |
| DLL | <i>Delay Locked-Loop</i> |
| PLL | <i>Phase Locked-Loop</i> |
| VCDL | <i>Voltage Controlled Delay Line</i> |
| PFD | <i>Phase and Frequency Detector</i> |

Lista de símbolos

| | |
|----------------------|---|
| V | Tensão de alimentação (Volts) |
| f_{op} | Frequência de operação (Hertz) |
| P_{total} | Potência total consumida pelo inversor CMOS (Watts) |
| P_{din} | Potência dinâmica consumida pelo inversor CMOS (Watts) |
| P_{est} | Potência estática consumida pelo inversor CMOS (Watts) |
| P_{cc} | Potência de curto circuito consumida pelo inversor CMOS (Watts) |
| C_L | Capacitância de chaveamento para um inversor CMOS (Faraday) |
| V_{TH} | Tensão limiar (Volts) |
| V_{DD} | Tensão de alimentação do inversor CMOS (Volts) |
| V_{in} | Tensão de entrada para o inversor CMOS (Volts) |
| $E_{0\rightarrow 1}$ | Energia envolvida em uma transição de nível lógico baixo para alto (Joules) |
| $p(t)$ | Potência instantânea do inversor CMOS (Watts) |
| $i_L(t)$ | Corrente instantânea do inversor CMOS (Amperes) |
| V_{out} | Tensão de saída do inversor CMOS (Volts) |
| T | Período do sinal de relógio (segundos) |
| $P_C(t)$ | Potência instantânea do capacitor de chaveamento (Watts) |
| $f_{0\rightarrow 1}$ | Frequência de transições de estado para o sinal de entrada (Hertz) |
| $P_{0\rightarrow 1}$ | Probabilidade de transição de estado para o sinal de entrada (adimensional) |
| τ | Constante de tempo de carga e descarga de um capacitor (segundos) |
| R_{eqn} | Resistência equivalente do NMOS na região de saturação (Ohms) |
| t_{pHL} | Tempo de propagação de nível lógico alto para baixo (segundos) |
| I_{DSAT} | Corrente no dreno do NMOS na região de saturação (Amperes) |

| | |
|------------|--|
| λ | Parâmetro de modulação do comprimento do canal (Amperes) |
| V_{ctrl} | Tensão de controle do elemento de atraso (Volts) |
| T_{OUT} | Período do oscilador de referência do DLL (segundos) |
| T_{in} | Atraso do sinal de referência (segundos) |
| T_{err} | Erro entre o atraso do sinal de referência com a saída do DLL (segundos) |
| K_{pd} | Ganho proporcional do detector de fase e frequência (adimensional) |
| K_i | Ganho integral do filtro passa-baixa (adimensional) |
| K_{vccl} | Ganho proporcional do caminho de atraso controlado por tensão (adimensional) |
| C_f | Capacitância do filtro passa-baixa de primeira ordem (Farads) |
| i_{UP} | Corrente de carga do capacitor do filtro passa-baixa do DLL (Amperes) |
| i_{DN} | Corrente de descarga do capacitor do filtro passa-baixa do DLL (Amperes) |
| V_{BN} | Tensão de polarização do espelho de corrente do <i>charge pump</i> (Volts) |

Sumário

| | | |
|----------|--|-----------|
| 1 | INTRODUÇÃO | 25 |
| 1.1 | Motivação | 28 |
| 1.2 | Delimitação do Trabalho | 31 |
| 1.3 | Objetivo Geral | 31 |
| 1.4 | Objetivos Específicos | 31 |
| 1.5 | Organização do Trabalho | 32 |
| 2 | FUNDAMENTOS DAS TÉCNICAS DE ESCALONAMENTO DINÂMICO DE TENSÃO E FREQUÊNCIA | 33 |
| 3 | IMPLEMENTAÇÃO DA ARQUITETURA PROPOSTA | 41 |
| 3.1 | <i>Delay Locked-Loop</i> | 44 |
| 3.2 | Réplica Configurável do Caminho Crítico | 49 |
| 3.3 | Controlador do Gerador de Relógio | 50 |
| 4 | RESULTADOS E DISCUSSÕES | 53 |
| 4.1 | <i>Delay Locked-Loop</i> | 53 |
| 4.2 | Controlador e Réplica Configurável | 56 |
| 5 | CONCLUSÕES | 61 |
| 5.1 | Sugestões para trabalhos futuros | 61 |
| | REFERÊNCIAS | 63 |
| | ANEXOS | 67 |
| | ANEXO A – CÓDIGO FONTE DO CONTROLADOR DO RELÓGIO ADAPTATIVO | 69 |

1 Introdução

A Quarta Revolução Industrial vem representando uma transição socioeconômica surpreendente para as gerações que estão a presenciando. Desde processos de baixa complexidade, como realizar pedidos para estabelecimentos, a atividades de alto risco, a exemplo de operações cirúrgicas (EINSTEIN.BR, 2018), estão sendo realizadas por intermédio de dispositivos eletrônicos conectados em rede. Neste sentido, os significados de certos termos, como mundo físico e mundo virtual, estão sendo reformados devido às transformações vivenciadas nos últimos vinte anos (SCHWAB; MIRANDA, 2016).

Através da imersão da sociedade no ambiente de dispositivos inteligentes interconectados, o entendimento do que realmente faz parte do ser humano, considerando também um contexto biológico, vem sendo expandido progressivamente (SCHWAB; MIRANDA, 2016).

Dentro desta perspectiva, um dos principais fenômenos que protagonizam estas transformações tecnológicas e sociais é a Internet das Coisas (*Internet of Things* (IoT), do inglês). Esta filosofia de comunicação entre dispositivos possui uma variedade considerável de aplicações e propósitos. Desde a comunicação entre máquinas (*Machine-to-machine communication* (M2M), do inglês), passando por integração via protocolo ZigBee, até Redes de Sensores Sem Fio (RSSF), vários são os protocolos e metodologias de comunicação que são abrangidos pela IoT.

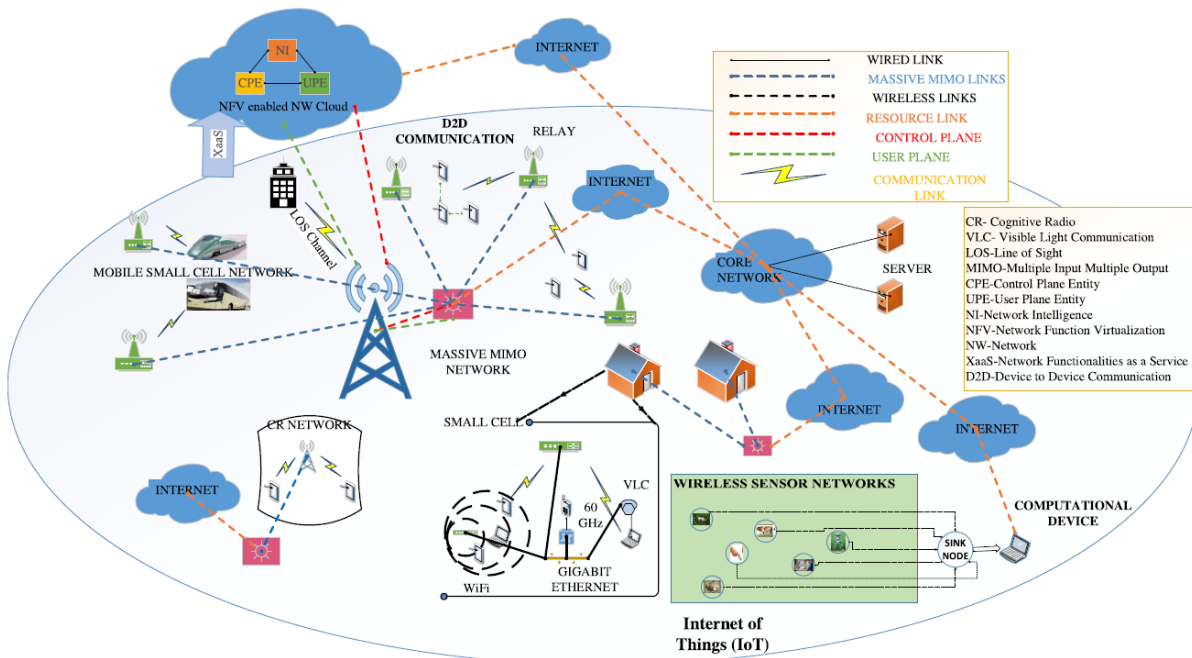
Um dos marcadores que ressaltam a rápida capacidade de expansão destas redes inteligentes, é o número de dispositivos eletrônicos conectados à Internet. De acordo com um estudo realizado pelo *Internet Business Solutions Group* da CISCO Coop. (EVANS, 2011), o número de dispositivos conectados à Internet superou o de usuários humanos na metade de 2008.

Tal proporção foi alcançada graças à grande aceitabilidade deste tipo de sistema por vários segmentos da sociedade (FEKI et al., 2013). Para citar alguns exemplos, pode-se mencionar os sistemas residenciais de gerenciamento ambiental, redes veiculares e de comunicação com infraestrutura, sistemas de monitoramento de sinais vitais em ambientes hospitalares, entre outros (MIRAZ; ALI; EXCELL, 2015).

Na Figura 1.1, apresenta-se uma visão sistêmica da dimensão das aplicações supracitadas, dentro de um contexto de sistemas IoT. É possível notar que uma série de desafios emergem dessa arquitetura, com propriedades, e conseqüentemente soluções, distintas. Tendo em mente que, de acordo com Gupta e Jha (2015), até 2020 teremos cerca de 50 bilhões de dispositivos conectados à nuvem, com a capacidade de acessar e compartilhar dados em qualquer lugar, é perceptível que deverá existir um crescimento

proporcional da atual capacidade de utilização do espectro, assim como da redução do custo e consumo de energia destes dispositivos.

Figura 1.1 – Arquitetura geral de uma rede de comunicação oriunda da proposta da Internet das Coisas.



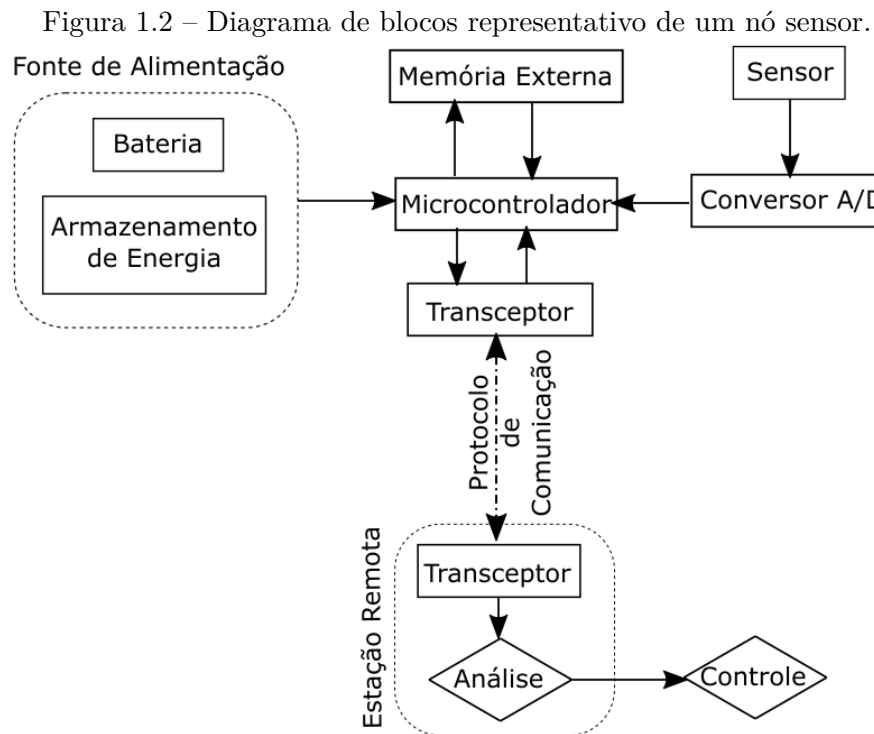
Fonte: (GUPTA; JHA, 2015)

Para tanto, dentro desta proposta, conforme Figura 1.1, existem as Redes de Sensores Sem Fio, que também podem ser utilizadas em vários ramos que necessitam de aquisição de dados em tempo real. De acordo com Sharma e Sharma (2014), uma rede de sensores sem fio pode ser definida como um sistema de aquisição de dados, onde os nós sensores, que adquirem e tratam os dados, não estão conectados via cabo, realizando a sua comunicação com a estação base através de algum protocolo de comunicação sem fio. Em um nó sensor, tradicionalmente, estão presentes os seguintes elementos:

- Um microprocessador para processamento local dos dados
- Uma memória externa para armazenamento temporário dos dados
- Um sensor para aquisição em tempo real das variáveis monitoradas
- Um conversor analógico-digital
- Um transceptor
- Uma fonte de alimentação

Para essa estrutura, representada esquematicamente pelo diagrama de blocos na Figura 1.2, o microprocessador realiza um tratamento local dos dados, antes de enviá-los

para a estação base, nesse caso remotamente localizada. Devido a quantidade limitada de memória interna dos microprocessadores utilizados em RSSF, uma memória externa é necessária para armazenar localmente os dados a serem processados, como seria o caso para uma análise de Fourier, onde os coeficientes da Transformada Rápida de Fourier poderiam ser disponibilizados pela memória externa. Os dados provenientes de uma determinada variável, por exemplo, a temperatura ambiente, são adquiridos por um sensor apropriado, que irá disponibilizar esta informação de maneira analógica. Por conseguinte, para que esta informação possa ser trabalhada pelo microprocessador, a mesma precisa ser convertida para o domínio digital previamente por meio de um conversor A/D. Vale ressaltar que toda essa arquitetura do nó sensor possui como fonte de alimentação comumente uma bateria. Todavia, alguns sistemas mais modernos utilizam mecanismos de armazenamento de energia, como painéis solares, para suprir essa necessidade (SHARMA; SHARMA, 2014).



Fonte: Adaptado de (SHARMA; SHARMA, 2014)

Desta maneira, após as operações locais serem efetuadas pelo microprocessador, os resultados podem ser enviados por meio de um protocolo de comunicação sem fio para a estação principal. Existem diversas alternativas para este procedimento, sendo as mais comuns: Zigbee, Bluetooth, *Z-wave*, *Insteon*, *Wi-Fi*, entre outros.

Devido a restrição de fornecimento de energia em um nó sensor, os protocolos que priorizam o baixo consumo de energia são os mais utilizados em aplicações reais (SHARMA; SHARMA, 2014). A exemplo do protocolo Zigbee, sendo estruturado pelo padrão IEEE 802.15.4, com sua taxa de transmissão na faixa de 20 a 250 *kilobits per*

second (kbps) e frequência de operação com valores de 868 *Mega Hertz* (MHz), 915 MHz e 2,4 *Giga Hertz* (GHz), o resultante baixo custo e baixo consumo de energia são atrativos para sua utilização em sistemas de monitoramento ambiental (FRITIAWAN et al., 2017).

Conforme discutido por Jhang, Sun e Cui (2010), os mecanismos de modulação do Zigbee, sendo estes fundamentados na modulação por deslocamento de fase (*Phase Shift Keying* (PSK), do inglês), em conjunto com o fato de que suas distâncias de alcance estão na faixa de 10 a 100 metros, permitem que os nós baseados em Zigbee possam ser alimentados com baterias AA, tendo uma durabilidade de 6 meses a 2 anos.

Por conseguinte, pode-se concluir desta discussão que os desenvolvimentos tecnológicos para as RSSF são fortemente direcionados para o baixo consumo de energia (SHARMA; SHARMA, 2014) (JHANG; SUN; CUI, 2010).

1.1 Motivação

Com o suporte da discussão anterior, é possível compreender a necessidade de se priorizar o baixo consumo de energia durante o projeto de nós sensores. As baterias utilizadas para fornecimento de energia, assim como o ambiente de difícil acesso ao qual diversas redes de sensores inteligentes se encontram, acrescentando assim a logística e custo para a manutenção destas, representam as restrições naturais deste tipo de sistema. Não apenas as RSSF, mas também diversos outros dispositivos móveis mais complexos que nós sensores, como celulares e *smart watches*, justificam os esforços dos profissionais da tecnologia em otimizar o consumo de energia de sistemas embarcados.

Diversas abordagens com o objetivo de reduzir o consumo de energia de um nó sensor, aumentando assim sua durabilidade e reduzindo o custo de manutenção, vem sendo discutidas pela academia. Uma destas estratégias, brevemente apresentada na introdução deste documento, é o armazenamento de energia, melhor conhecido como *energy harvesting* no meio científico. Esta técnica consiste em obter e armazenar a energia de alimentação do nó sensor através do próprio ambiente. Um exemplo clássico consiste do uso de sensores piezoelétricos para captar a energia mecânica das vibrações de máquinas, como motores elétricos, e convertê-la em energia elétrica para o nó sensor. Todavia, esta abordagem é claramente limitada pelo tipo de ambiente em que o nó se encontra, não sendo assim possível generalizá-la para qualquer contexto (ITOH, 2016).

Nesta perspectiva, uma metodologia passível de ser generalizada para diferentes contextos, sem uma dependência explícita do tipo de ambiente em que a RSSF ou o dispositivo móvel será aplicado, deve considerar aspectos relativos ao consumo de energia devido a arquitetura do sistema, ou seja, como os recursos de processamento e armazenamento de dados são gerenciados. Este tipo de estratégia pode ser implementada tanto a nível de *hardware* como a nível de *software*.

Um aspecto interessante das redes de sensores inteligentes sem fio é o fato destas possuírem um desempenho dependente da quantidade de dados a serem processados, variando assim seu consumo de energia de acordo com as tarefas agendadas para execução (AKYILDIZ, 2002). Sendo assim, é possível reduzir o gasto de energia desativando certos módulos do sensor quando este está em modo *sleep* (modo soneca, tradução livre do inglês). Por exemplo, um nó sensor responsável por medir a temperatura de um objeto possuirá uma taxa de medição relativamente lenta em comparação com sua frequência de operação, isto é, os componentes do sensor irão permanecer sem uso a maior parte do tempo. Apesar desta propriedade, o consumo de energia pela quantidade de informação a ser tratada ser relativa à aplicação, a metodologia para otimizar o consumo pode ser generalizada, conforme veremos no capítulo seguinte.

Uma das técnicas para otimizar o consumo de potência de acordo com as tarefas a serem executadas pelo microprocessador é o Escalonamento de Tensão e Frequência, ou *Dynamic Voltage and Frequency Scaling* (DVFS). Esta abordagem para economia de energia e aumento da durabilidade de sistemas embarcados a bateria é o principal conceito apresentado neste trabalho e será discutido em mais detalhes no capítulo 2.

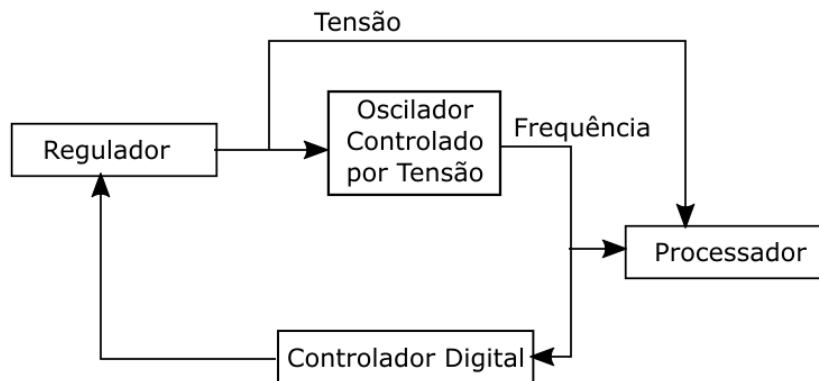
Todavia, vale apenas destacar que esta técnica baseia-se no fato de que, em circuitos com tecnologia de transistores *Complementary Metal Oxide Semiconductor* (CMOS), a potência e energia consumidas por um chip possui uma relação quadrática com a tensão de alimentação e linear com a frequência de operação do mesmo (SULEIMAN, 2005).

Consequentemente, ao se configurar os valores da tensão e frequência ideais para uma determinada tarefa, a economia de energia para finalizar aquele processamento será maximizada.

A título de exemplo, Antonio et al. (2017) propuseram uma arquitetura de *hardware*, voltada para um processador MIPS de 16 bits com 5 estágios de *pipeline*, cujo código fonte sintetizável está disponível gratuitamente (YA, 2013), para configurar a tensão (V) e frequência de operação (f_{op}) deste, sendo assim, efetuando um escalonamento dinâmico de tensão e frequência. Vale ressaltar que, tanto um algoritmo para a determinação de V e f_{op} , como um hardware para a geração destes foram propostos, constituindo assim uma abordagem mista (*hardware/software*) para a implementação de DVFS. Esta proposta é ilustrada esquematicamente na Figura 1.3.

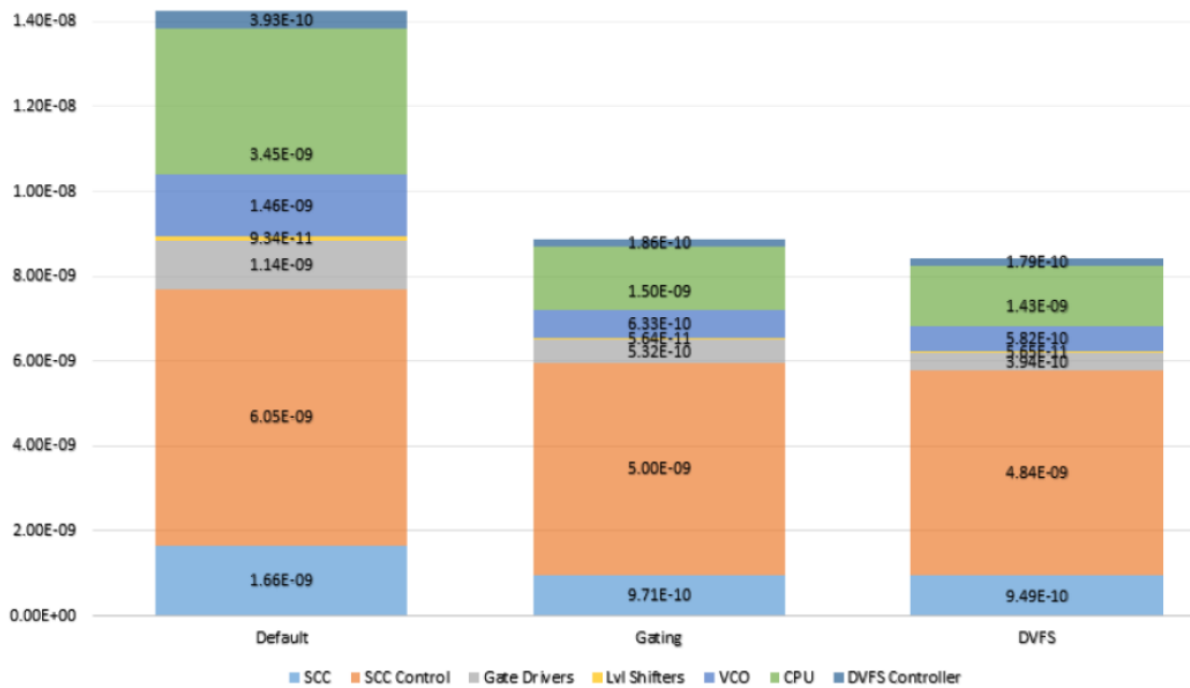
Tal proposta resultou em uma economia total de energia consumida, ou seja considerando o sistema completo, de 38,91%, em comparação com um sistema sem DVFS. É possível notar também, através da visualização gráfica dos resultados produzidos por Antonio et al. (2017), conforme Figura 1.4, que estratégias de *clock gating* (i.e., o relógio é desativado quando o processador está em modo *sleep*), possuem economia comparável, mas não superior ou igual às técnicas de DVFS.

Figura 1.3 – Representação em diagrama de blocos de uma arquitetura para implementação de DVFS.



Fonte: Adaptado de (ANTONIO et al., 2017)

Figura 1.4 – Consumo de energia (em Joules) de cada componente para cada estratégia verificada.



Fonte: (ANTONIO et al., 2017)

Entretanto, a aplicação do algoritmo de DVFS para decidir qual a melhor relação entre a tensão e a frequência do microprocessador introduz tarefas a mais para o sistema (i.e., introduz um *overhead*) (HONG, 1998). Dessa forma, uma possível incrementação no desempenho do sistema anterior pode ser alcançada caso o sinal de relógio, ou seja a frequência de operação, se adapte continuamente, em tempo real, às variações no valor da tensão de alimentação (KWAK; NIKOLIC, 2015).

Sendo assim, este trabalho é inspirado nessa necessidade dos circuitos de geração de relógio implementados em sistemas com DVFS. Ao propor uma arquitetura de geração de relógio adaptativo de acordo com a tensão, a demanda do controlador de potência será re-

duzida, assim como sua complexidade, pois a frequência estará se ajustando continuamente de acordo com a tensão de alimentação. Naturalmente, uma arquitetura orientada desta maneira irá resultar em uma redução de área e de consumo de energia para o controlador de potência, além de necessitar de um menor número de ciclos de relógio para controlar a tensão e frequência do processador.

1.2 Delimitação do Trabalho

Dentre as inúmeras arquiteturas e metodologias possíveis para a implementação de técnicas de escalonamento de tensão e frequência, este trabalho enfatizará a geração do sinal de relógio, ou seja a determinação da frequência apropriada para uma dada tensão de alimentação. Com isso, o nível de análise necessário para a avaliação da proposta está delimitado ao *hardware* do circuito gerador de *clock*. Sendo assim, as possíveis implicações a nível de software, isto é, como agendar as tarefas para otimizar o consumo, assim como a execução das mesmas pelo processador, não são analisadas aqui, sendo sugeridas como trabalhos futuros.

1.3 Objetivo Geral

Neste trabalho monográfico, o objetivo principal é o estudo e desenvolvimento de uma arquitetura de geração de relógio adaptativo no contexto de DVFS e sua demonstração de funcionamento.

1.4 Objetivos Específicos

Os objetivos deste trabalho são os seguintes:

- Demonstrar a redução da demanda para o controlador de potência ao utilizar a arquitetura proposta
- Validar a estratégia de controle de tensão e frequência via hardware para redução de consumo de energia
- Verificar os requisitos físicos para implementação, como área e número de portas lógicas, da arquitetura apresentada na tecnologia FDSOI (*Fully Depleted Silicon On Insulation*) 28-nm

1.5 Organização do Trabalho

Este trabalho está organizado em 5 capítulos, incluindo a introdução, estruturados da seguinte maneira:

O Capítulo 1 consiste na Introdução, onde uma contextualização inicial é apresentada, direcionando os propósitos do trabalho de acordo com a sua motivação, concluindo com a descrição dos objetivos geral e específicos.

No Capítulo 2 são apresentados os princípios básicos sobre técnicas de escalonamento dinâmico de tensão e frequência, discutindo as suas restrições e as origens destas.

No Capítulo 3, a implementação do circuito gerador de relógio adaptativo é fundamentada. As principais considerações, assim como os conceitos teóricos essenciais, para o desenvolvimento e projeto desta arquitetura são então discutidas.

Já no Capítulo 4, os resultados produzidos são justificados, assim como as ferramentas necessárias para sua obtenção são explanadas, delineando adequadamente o domínio de aplicabilidade destes.

Por fim, no Capítulo 5, as conclusões sintetizam as discussões e resultados apresentados neste trabalho.

2 Fundamentos das Técnicas de Escalonamento Dinâmico de Tensão e Frequência

Na introdução deste trabalho foi discutida uma das principais motivações para se reduzir o consumo de energia de um microprocessador, contextualizando a problemática para Redes de Sensores Sem Fio. Esta justificativa foi descrita como sendo o aumento na durabilidade dos nós sensores, pois estes são comumente alimentados por baterias (JHANG; SUN; CUI, 2010).

Todavia, existem outras razões para se reduzir o consumo de energia em qualquer sistema eletrônico, mesmo que este possua fonte de alimentação própria. Primeiramente, devido ao fato de que a frequência de operação dos microprocessadores atuais vem crescendo consideravelmente, alcançando a marca de dezenas de Megahertz para alguns Gigahertz, a depender da aplicação (BURD, 2001). Isto tem como consequência direta no aumento da potência dinâmica do dispositivo, pois esta depende da frequência de operação do circuito conforme demonstrado a seguir.

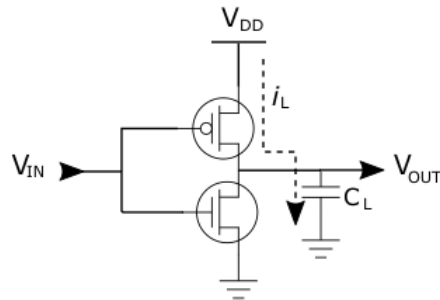
Além disso, alguns outros efeitos indiretos são potencializados pelo aumento do consumo de energia de um processador. Para mencionar os principais, pode-se ressaltar:

- A dissipação de calor, e portanto a temperatura do sistema, é diretamente proporcional ao aumento da potência consumida. Desta forma, a vida útil do componente tende a diminuir e sua taxa de falhas dobra a cada elevação de 10 °C na sua temperatura (ERGIN, 2003).
- A estrutura física de refrigeração e dissipação de calor torna-se mais complexa, e portanto custosa, para processadores com alto consumo de energia. De acordo com uma estimativa da Intel, realizada em 1998, os custos com refrigeração aumentam em \$ 1,00 por Watt (W) após o processador ter alcançado a marca de 40 W (TIWARI et al., 1998).
- A depender da aplicação, estes processadores poderão estar localizados em ambientes fechados, o que irá requisitar dispositivos de refrigeração térmica mais potentes quanto maior for o consumo de energia e dissipação de calor desses processadores.
- A vida útil da bateria é reduzida, assim como de diversos componentes do microprocessador. Um estudo realizado por Chandrakasan (1992) demonstrou que o consumo de potência dobra a cada quatro anos, o que não é compatível com o desenvolvimento de baterias e componentes mais robustos.

Essas desvantagens no use de microprocessadores com alto consumo de energia estimulam ainda mais o desenvolvimento de técnica de DVFS.

O princípio fundamental das técnicas de DVFS reside na dependência da potência dinâmica do circuito CMOS com a frequência de operação e com a tensão de alimentação. Para entender essa relação, pode-se utilizar o modelo clássico do inversor CMOS, conforme apresentado na Figura 2.1.

Figura 2.1 – Esquemático para um inversor CMOS tradicional.



Fonte: Autoria própria

A capacitância C_L do circuito da Figura 2.1 consiste na capacitância de chaveamento do inversor. A potência dissipada por um inversor CMOS pode ser dividida em três componentes principais, conforme Equação 2.1.

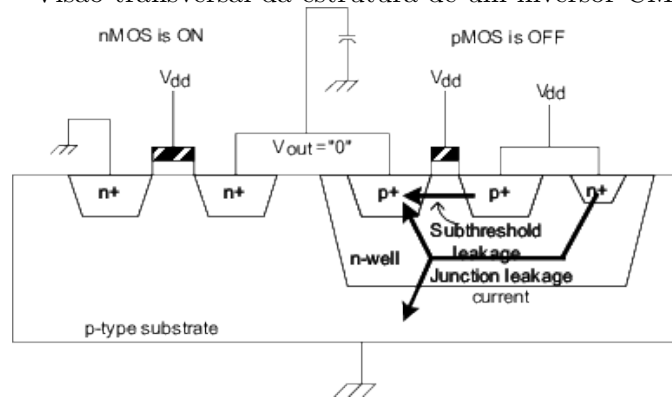
$$P_{total} = P_{din} + P_{est} + P_{cc} \quad (2.1)$$

Sendo assim, a potência total de um inversor CMOS é a soma da sua potência dinâmica (P_{din}) com sua potência estática (P_{est}) e a potência de curto circuito (P_{cc}). A potência estática está relacionada com a "fuga" (i.e., *leakage*, em inglês) de corrente através dos diodos reversamente polarizados presentes nas junções P-N de um inversor CMOS, conforme ilustrado na Figura 2.2. Já a potência de curto circuito é oriunda da dissipação de energia durante o momento em que tanto o transistor PMOS quanto o transistor NMOS estão na região de condução, realizando assim uma ligação direta entre a fonte de alimentação e o terminal de referência.

Por fim, a potência dinâmica está relacionada com a energia necessária para carregar e descarregar a capacitância de chaveamento durante uma transição de estado.

Devido ao foco deste trabalho ser em arquiteturas de *hardware* para aplicação de técnicas de DVFS, será enfatizado o estudo da potência dinâmica do inversor CMOS, assim como os resultados posteriormente apresentados são voltados para esta componente. Aliado à isso, a componente da potência dinâmica contribui em mais de 80% da potência total para microprocessadores cuja frequência de operação esteja na faixa de alguns MHz, com uma tensão de alimentação superior a tensão limiar (i.e., *Threshold Voltage* (V_{TH}), em inglês) (CHANDRAKASAN, 1992).

Figura 2.2 – Visão transversal da estrutura de um inversor CMOS em silício.



Fonte: (PAL, 2002)

Neste sentido, é possível determinar a potência dinâmica teórica de um inversor CMOS para uma transição do sinal de entrada, V_{in} , de nível lógico baixo para o nível lógico alto (V_{DD}) através da energia necessária para carregar a capacitância de chaveamento C_L , conforme Equação 2.2.

$$E_{0 \rightarrow 1} = \int_0^T P_{DD}(t) dt = V_{DD} \int_0^T i_{DD}(t) dt = V_{DD} \int_0^{V_{DD}} C_L dV_{out} = C_L V_{DD}^2 \quad (2.2)$$

Em relação a energia armazenada no capacitor após a transição, é possível deduzí-la através da Equação 2.3.

$$E_C = \int_0^T P_C(t) dt = \int_0^T V_{out}(t) i_L(t) dt = \int_0^{V_{DD}} C_L V_{out} dV_{out} = \frac{1}{2} C_L V_{DD}^2 \quad (2.3)$$

Com isso, conclui-se que metade da energia da transição de estado é dissipada pelo transistor PMOS. Considerando também a frequência do sinal de entrada V_{in} , determina-se a potência dinâmica P_{din} dissipada pelo inversor através da relação entre potência e energia dada pela Equação 2.4.

$$P_{din} = \frac{E_{0 \rightarrow 1}}{T} = E_{0 \rightarrow 1} f_{0 \rightarrow 1} = C_L V_{DD}^2 f_{op} P_{0 \rightarrow 1} \quad (2.4)$$

Desta forma, a potência dinâmica é diretamente proporcional com o número de transições de nível lógico do sinal de entrada, e conseqüentemente do sinal de saída para o caso de um inversor. O fator $P_{0 \rightarrow 1}$ é um indicativo da probabilidade de determinada porta lógica mudar o valor de sua saída, o que irá acarretar em um carregamento ou descarregamento das capacitâncias de chaveamento.

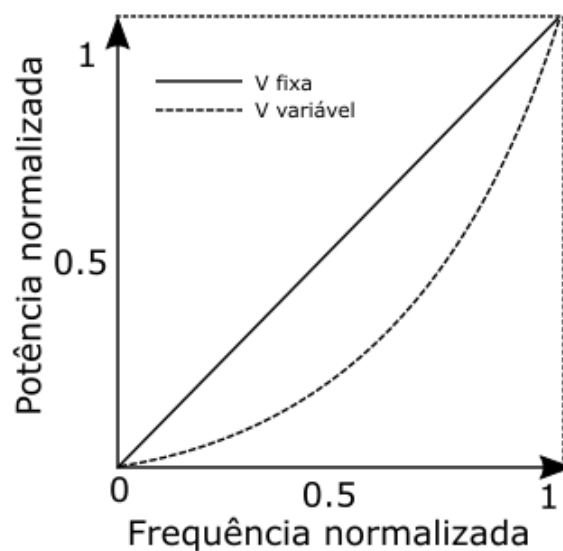
Portanto, é possível notar que a potência dinâmica do inversor, conseqüentemente também de microprocessadores complexos, possui uma dependência quadrática com a

tensão de alimentação V_{DD} e uma proporcionalidade linear com a frequência de operação f_{op} .

O conceito básico do escalonamento dinâmico de tensão e frequência reside no fato de que pode-se alcançar uma grande redução do consumo de potência encontrando a relação ideal entre V_{DD} e f_{op} para determinadas rotinas de execução de tarefas.

Na Figura 2.3, representa-se graficamente a relação dada pela Equação 2.4, em que compara-se uma situação com variação da tensão de alimentação e mantendo esta fixa.

Figura 2.3 – Relação entre a potência dinâmica e a frequência de operação sem e com ajuste correlato da tensão de alimentação.

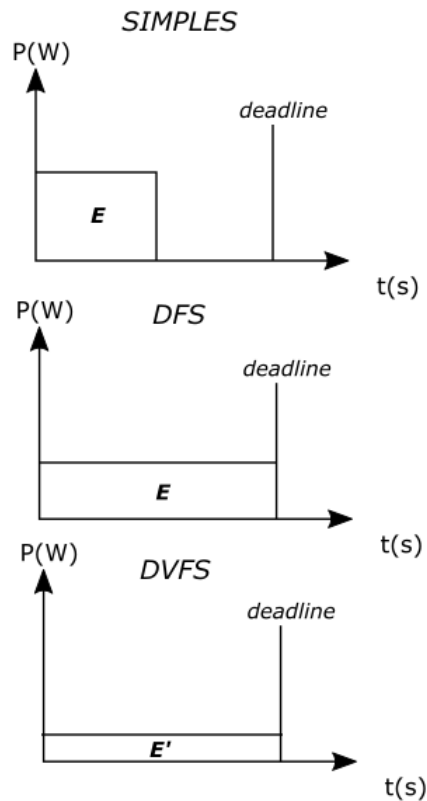


Fonte: Autoria própria

O principal resultado prático da relação dada na Equação 2 está no fato de que é possível otimizar o consumo de energia para executar uma rotina sem comprometer o desempenho geral do microprocessador. Conforme foi levantado na introdução deste trabalho, as aplicações que se beneficiam de técnicas de DVFS costumam ter um perfil variável de demanda de processamento com o tempo. Isto é, o processador permanece em modo *sleep* por longos períodos, até receber dados para tratar em momentos discretos (i.e., eventos). Desde que esses dados sejam processados dentro de um período adequado após o surgimento do evento, como os períodos seguintes serão de inatividade, pode-se reduzir o desempenho do processador, diminuindo sua frequência de operação e tensão, para efetuar aquela tarefa em um tempo adequado. Com isso, o desempenho geral do sistema é mantido, pois as rotinas estão sendo executadas nos prazos estabelecidos (i.e., *deadlines*), e uma economia real de energia é alcançada.

Na Figura 2.4, três situações de execução de uma tarefa são ilustradas. Na primeira, denominada como simples, nenhuma técnica de redução de consumo de energia é aplicada, servindo então como referência. Na segunda, apenas a frequência de operação é variada, técnica conhecida como Escalonamento Dinâmico de Frequência (*Dynamic Frequency*

Figura 2.4 – Ilustração da operação de DVFS para a execução de uma tarefa.



Fonte: Autoria própria

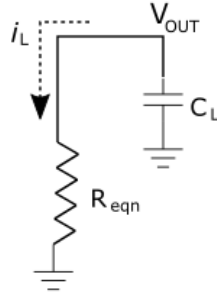
Scaling (DFS), do inglês). Nessa situação, o tempo de execução da operação dobra, indicando que a frequência foi reduzida pela metade. Desse modo, a potência dinâmica também é reduzida, o que pode ser verificado pela Equação 2.4. Entretanto, a energia consumida na operação permanece a mesma, o que também é confirmado pela Equação 2.2.

Para alcançar uma redução real de consumo de energia, deve-se recorrer à técnica de DVFS, em que tanto a tensão como a frequência são reduzidas. Sendo assim, a potência dinâmica é reduzida ainda mais em relação à segunda situação, resultando assim em uma economia significativa de energia para o microprocessador durante a execução daquela tarefa, o que é reforçado pela relação quadrática da tensão com a energia, conforme já apresentado na Equação 2.2.

Entretanto, não é possível reduzir a tensão e a frequência indiscriminadamente. Existe uma interdependência entre V_{DD} e f_{op} que restringe o quanto é possível reduzir V_{DD} para minimizar o consumo de potência de um microprocessador. No capítulo seguinte esta relação será melhor detalhada. Porém, cabe aqui destacar que, para qualquer circuito integrado, o tempo de propagação de um sinal pelos caminhos de lógica (i.e., as portas lógicas que executam as operações discretas) é dependente da tensão de alimentação destes caminhos. Para obter uma visualização de primeira ordem desta afirmação, através do

modelo da Figura 2.1, em uma situação de descarga do capacitor de chaveamento C_L , isto é, em uma transição do sinal de saída de alto para baixo, pode-se considerar o modelo de circuito RC apresentado na Figura 2.5.

Figura 2.5 – Modelo de primeira ordem para a descarga do capacitor de chaveamento durante uma transição de alto para baixo.



Fonte: Autoria própria

Sendo assim, a resposta transiente da tensão de saída para uma descarga completa, de V_{DD} para 0 V, é dada por uma relação exponencial decrescente conforme Equação 2.5.

$$V_{out}(t) = (1 - e^{-\frac{t}{\tau}})V_{DD} \quad (2.5)$$

Portanto, segue que o tempo t_{pHL} necessário para V_{out} atingir a metade do valor inicial, correspondendo assim ao tempo de atraso de propagação de uma transição de alto para baixo, é encontrado pela Equação 2.6.

$$t_{pHL} = \ln(2)\tau = 0,69R_{eqn}C_L \quad (2.6)$$

Através desta relação, para uma análise de primeira ordem, percebe-se que o atraso de propagação depende da capacitância de chaveamento C_L , conforme ressaltado previamente, mas também da resistência equivalente do transistor NMOS, para uma transição de alto para baixo, na região de saturação. A capacitância C_L , por sua vez, depende de parâmetros físicos como as dimensões de área dos transistores e das interconexões. Modelos mais completos desta capacitância incluem efeitos não-lineares, como os efeitos de encurtamento do canal, que também influenciam na capacitância equivalente de chaveamento (RABAEY; CHANDRAKASAN; NIKOLIC, 2003). Por outro lado, para um primeiro estudo, com o objetivo de averiguar a dependência de t_{pHL} com a tensão de alimentação V_{DD} , esses efeitos de segunda ordem podem ser desprezados sem comprometer significativamente a precisão (RABAEY; CHANDRAKASAN; NIKOLIC, 2003).

A restrição da redução da frequência e tensão durante uma operação de DVFS é melhor entendida quando se analisa a componente R_{eqn} da Equação 2.6. De acordo com Rabaey, Chandrakasan e Nikolic (2003), a resistência equivalente de um transistor de efeito

de campo (*Metal Oxide Semiconductor Field Effect Transistor* (MOSFET), do inglês) na região de saturação pode ser modelada conforme 2.7.

$$R_{eqn} = \frac{1}{V_{DD}/2} \int_{V_{DD}/2}^{V_{DD}} \frac{V_{out}}{I_{DSAT}(1 + \lambda V_{out})} dV_{out} \simeq \frac{3}{4} \frac{V_{DD}}{I_{DSAT}} \left(1 - \frac{7}{9} \lambda V_{DD}\right) \quad (2.7)$$

Na relação dada pela Equação 2.7, I_{DSAT} é a corrente de saturação no dreno do transistor NMOS, e λ é o parâmetro de modulação do comprimento do canal. Nesse sentido, essa equação mostra a dependência quadrática da resistência equivalente, e portanto do atraso de propagação, com a tensão de alimentação. Isto resulta em uma restrição na diminuição da tensão para reduzir o consumo de energia, pois o atraso de propagação irá aumentar proporcionalmente, o que exigirá que a frequência de operação diminua de maneira correspondente a fim de evitar problemas de *setup*, como será descrito no próximo capítulo.

É esta relação que faz com que, em certo ponto, a diminuição da tensão leve a uma diminuição da frequência ainda maior, o que poderá afetar o tempo para finalizar uma instrução, por exemplo diminuindo o desempenho e também aumentando o consumo de energia, pois o tempo de execução pode se tornar demasiadamente longo.

3 Implementação da Arquitetura Proposta

Neste capítulo são descritos os blocos digitais e analógicos que constituem a arquitetura proposta para o gerador de relógio adaptativo. Os conceitos teóricos necessários para seu entendimento são apresentados concorrentemente com a discussão da implementação do circuito, assim como as ferramentas utilizadas para a sua realização são mencionadas.

A principal metodologia de projeto que motivou e orientou a proposta de arquitetura para geração de relógio adaptativo, é a de integrar no mesmo chip o controle e geração do relógio e da tensão do microprocessador. Circuitos integrados que possuem todo o sistema de gerenciamento de potência *on-chip*, incluindo o controlador (*Power Management Controller* (PMC), do inglês), possuem diversas vantagens em relação à precisão do gerenciamento do consumo de energia. Com os reguladores, que proveem a tensão do processador, juntamente com o gerador de relógio, integrados no mesmo chip do microprocessador, a malha de controle resultante será menor em comparação com arquiteturas com reguladores *off-chip* (KIM et al., 2008).

Além da precisão e velocidade superiores em comparação com reguladores *off-chip*, SoCs com reguladores integrados facilitam a implementação de técnicas de DVFS, pois os reguladores podem ser controlados mais confiavelmente, devido à sua proximidade com o PMC, além de estar no mesmo domínio de potência deste. Em contrapartida, os sistemas com reguladores a parte irão requisitar um maior número de pinos do chip do microcontrolador, além de apresentar interconexões mais longas, o que resulta em transições de modo do PMC mais lentas (KELLER et al., 2016a). O custo de "encapsulamento" (tradução livre do termo *packaging*, do inglês) será também consequentemente maior.

Estas características de SoCs com reguladores e gerador de relógio integrados justificam o provável acréscimo de área necessário para sua implementação. Todavia, de acordo com trabalhos anteriores desta linha, como o SoC projetado por Keller et al. (2016b), que resultou em um acréscimo de apenas 2%, este requisito é confortavelmente aceitável.

Neste sentido, com os reguladores de tensão integrados no SoC, é possível alcançar um ajuste mais fino quanto ao controle da frequência de operação (EYERMAN; EACKHOUT, 2011), uma vez que esta depende da tensão de alimentação, conforme discutido anteriormente. Por outro lado, isso permite com que o controle de frequência seja efetuado indiretamente pelo regulador de tensão. Tornar o gerador de relógio adaptativo às variações de tensão do processador, incluindo o *ripple* desta, apresenta diversas vantagens, entre elas cabe destacar:

- A eficiência do sistema é aumentada, pois quaisquer variações na tensão do regulador geram uma resposta na frequência do gerador de relógio, e, por conseguinte, na frequência de operação do microprocessador (KELLER et al., 2016b)
- A necessidade de se configurar a frequência separadamente da tensão, de acordo com um algoritmo executado no processador (podendo ser efetuado no PMC ou no núcleo principal), não existe mais, o que irá tornar o PMC mais simples
- As transições entre os modos do PMC podem ser efetuadas mais rapidamente (KELLER et al., 2016a)
- A escalabilidade com a tensão de alimentação é ampliada, pois a frequência depende diretamente desta
- Um menor número de interconexões entre o processador e o PMC será necessário com um gerador de relógio adaptativo (KWAK; NIKOLIC, 2015)

Todavia, a estratégia adotada possui algumas desvantagens que merecem ser mencionadas. Uma das principais limitações da proposta aqui apresentada está no fato de que algoritmos mais sofisticados de DVFS, que exigem uma independência entre V_{DD} e f_{op} , não poderão ser programados no PMC ou no núcleo, pois a frequência irá variar proporcionalmente com a tensão. Desse modo, a determinação da frequência deverá ser feita diretamente pela tensão do regulador.

Outra restrição importante consiste no uso de deslocadores de nível (tradução livre para *level shifters*, do inglês) entre o gerador de relógio e o núcleo (KWAK; NIKOLIC, 2015).

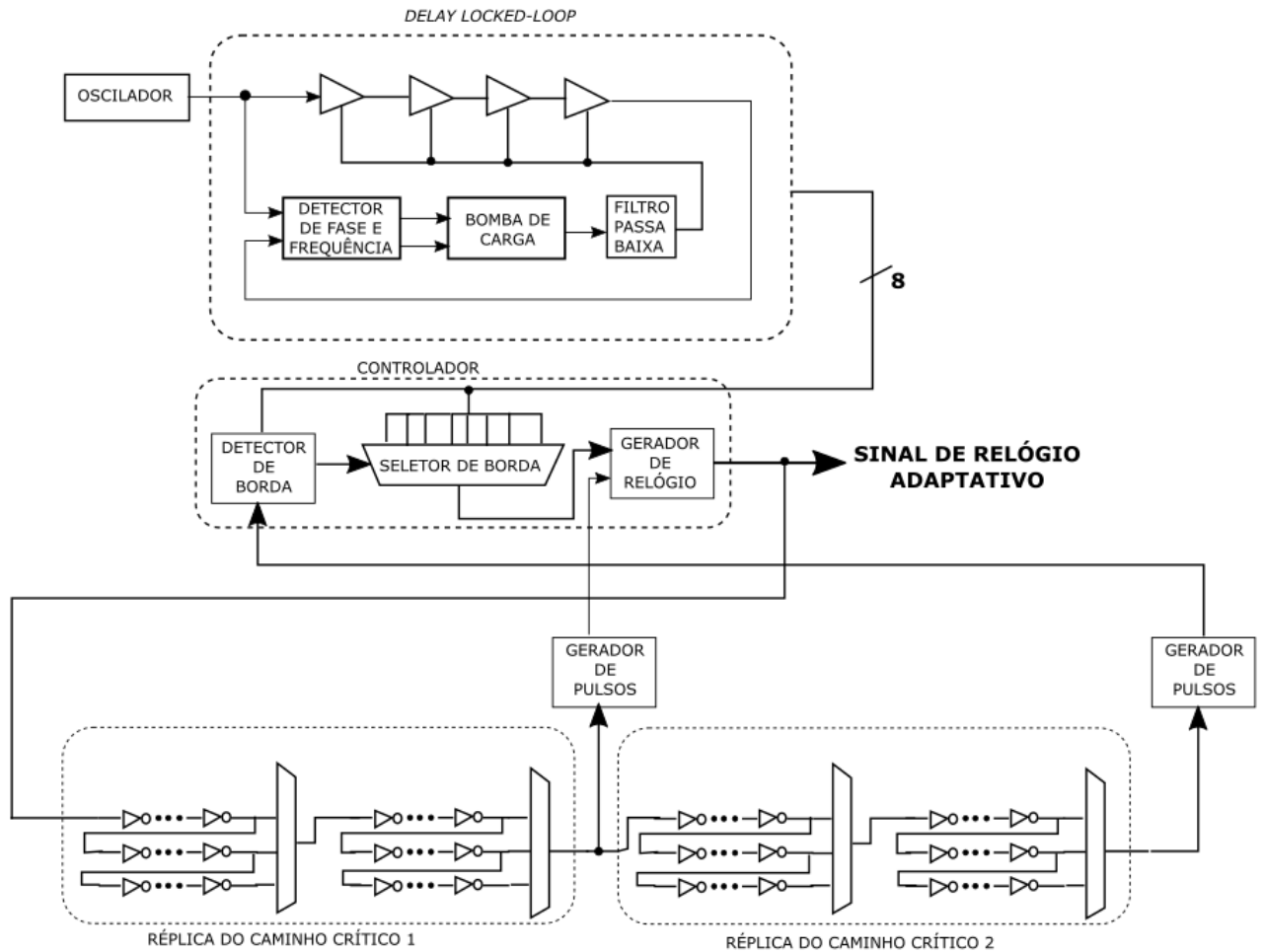
Entretanto, a eficiência, velocidade, e simplicidade do algoritmo de DVFS contra-balanceiam essas limitações em geral.

Para ilustrar as relações entre os diversos blocos da arquitetura, assim como apresentar uma visão geral dessa, na Figura 3.1 apresenta-se a proposta deste trabalho em um diagrama de blocos.

Através da Figura 3.1, podem-se destacar três blocos principais: O *Delay Locked-Loop*, o *controlador* e a réplica do caminho crítico, onde existem duas unidades desta. Estes blocos serão descritos em mais detalhes nas seções posteriores. Entretanto, para se obter uma visão geral do funcionamento do gerador de relógio adaptativo, além de entender as relações entre os módulos supramencionados, nos parágrafos seguintes será apresentada uma prévia do comportamento da proposta.

Em circuitos com uma frequência superior a algumas centenas de kHz, é fundamental que diferentes áreas do SoC estejam síncronas em relação ao sinal de relógio. Relógios assíncronos podem causar problemas de aquisição de dados nos *flip-flops* (i.e., os

Figura 3.1 – Visão geral em diagrama de blocos da arquitetura de gerador de relógio adaptativo.



Fonte: Autoria própria

registradores que armazenam informação), levando a um mal funcionamento do sistema (LUO; ZENG, 2011).

Para evitar esse problema, um *Delay Locked-Loop* (DLL) pode ser um mecanismo para sincronização. *Phase Locked-Loops* (PLL) também são alternativas viáveis. Porém, pelo fato de que um DLL utiliza um *caminho de atraso controlado por tensão* (tradução livre de *Voltage Controlled Delay Line*, do inglês), em oposição ao oscilador controlado por tensão dos PLLs, este apresenta um menor *jitter* em comparação com os PLLs (GHARIB; ABRISHAMIFAR, 2008). Além disso, por ser um sistema de primeira ordem, o DLL apresenta um projeto mais simples, resultando em uma área menor, o que está alinhado com os objetivos deste trabalho.

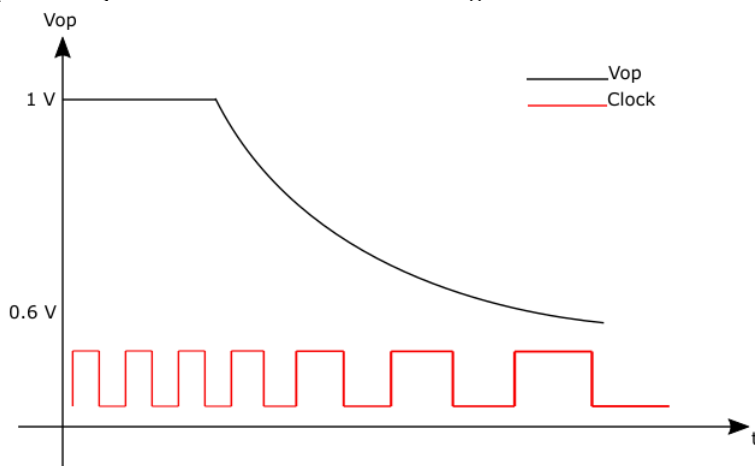
Desta forma, é o DLL que faz a sincronização do sinal de relógio adaptativo, tendo em vista que este tem seu período variável de acordo com a tensão de alimentação. Portanto, é essencial que o sinal fornecido ao processador seja síncrono com alguma referência conhecida. Para este fim, o DLL fornecerá 8 sinais de referência.

Para manter a propriedade de variabilidade da frequência com a tensão, sem comprometer o sincronismo do sinal de relógio com o DLL, um controlador digital é implementado. Este bloco tem por função realizar essa sincronização do sinal adaptativo com os sinais de referência do DLL, preservando ainda assim o período variável do sinal de relógio proveniente da réplica do caminho crítico.

A réplica do caminho crítico, sendo esta configurável, realiza a dependência da frequência com V_{DD} através do atraso de propagação do sinal de entrada para a saída. Neste bloco, existem duas réplicas de caminho crítico, a primeira gera um sinal de *reset* assíncrono para o relógio, produzindo assim o período em que o sinal se encontra em nível lógico baixo. Já a segunda réplica gera a borda de subida que será sincronizada com o DLL pelo controlador, realizando assim o *duty cycle*, tempo em que o sinal de relógio permanece em alto. Como o nome sugere, estes circuitos são réplicas do caminho crítico do processador que tem por objetivo simular o atraso de propagação dos caminhos críticos reais. Estas, por sua vez, são configuráveis, variando assim o atraso de propagação desejado através do número de inversores conectados em cadeia.

Com a implementação deste sistema, obtém-se um sinal de relógio cujo período é dependente da tensão. Esta propriedade é apresentada ilustrativamente, sem precisão matemática, na Figura 3.2.

Figura 3.2 – Representação ilustrativa do sinal de relógio de acordo com a tensão de alimentação.



Fonte: Autoria própria

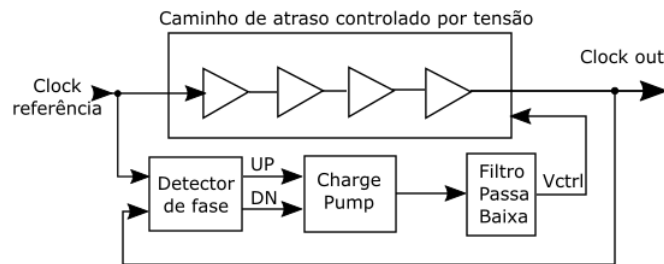
3.1 Delay Locked-Loop

Na discussão da introdução deste capítulo apresentaram-se as funcionalidades básicas dos blocos que constituem a proposta deste trabalho. Cabe então aqui um desenvolvimento teórico mais detalhado sobre o funcionamento de cada bloco, destacando as justificativas para a escolha de uma implementação particular.

O primeiro módulo estudado é o bloco analógico do *Delay Locked-Loop*. Tal bloco, conforme já explanou-se, tem por objetivo gerar 8 sinais de referência para sincronização do relógio adaptativo, a partir de uma referência fixa.

Na Figura 3.3 apresenta-se a estrutura do DLL, em diagrama de blocos, destacando assim seus componentes.

Figura 3.3 – Estrutura em diagrama de blocos do *Delay Locked-Loop*.



Fonte: Autoria própria

Em linhas gerais, a função do DLL é gerar, neste caso, 8 sinais de saída cujo defasamento é igual para todas as saídas. Para tanto, um sinal de relógio de referência, fornecido por um oscilador a cristal por exemplo, é utilizado na entrada para gerar esses 8 sinais de saída. Por exemplo, para um sinal de referência, proveniente do oscilador, com frequência de 1 GHz, 8 sinais de saída com frequência de 1 GHz, defasados de 125 picosegundos entre eles, serão fornecidos pelo DLL.

É notável pela descrição, assim como pela Figura 3.3, que uma malha de realimentação será necessária para alcançar a estabilidade. Deste modo, o DLL proposto possui 4 componentes principais:

- Um caminho de atraso controlado por tensão (*Voltage Controlled Delay Line (VCDL)*)
- Um detector de fase e frequência (*Phase and Frequency Detector (PFD)*)
- Um fornecedor de carga (traduzido livremente de *charge pump*, do inglês)
- Um filtro passa baixa

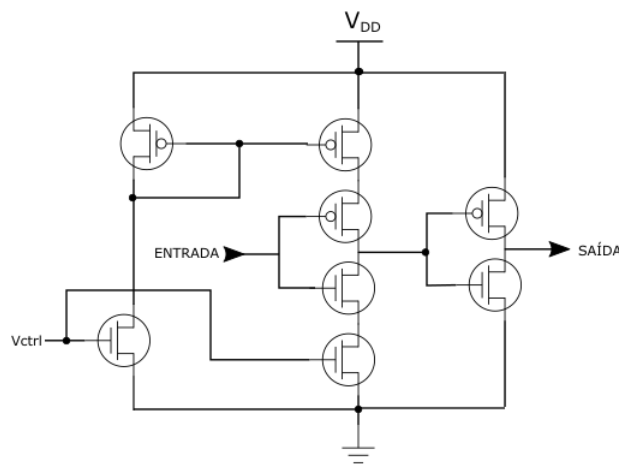
O sinal de *clock out* na Figura 3.3 representa a saída do último elemento de atraso do DLL.

O componente principal do DLL é o caminho de atraso controlado por tensão (VCDL), pois é este bloco que gera o defasamento necessário para as referências. Este bloco é constituído de 8 elementos de atraso idênticos, onde a saída de cada um desses elementos é uma saída de referência do DLL. A saída do último elemento é utilizada para efetuar o controle e ajuste das referências do DLL, e o objetivo da malha de realimentação é fazer com que esta seja igual ao sinal de referência, fornecido por algum oscilador.

Para o VCDL, existem, basicamente, duas estratégias para o circuito do elemento constituinte. A primeira, denominada como circuito de entrada simples ou absoluta, que foi utilizada neste trabalho, constitui de um inversor CMOS com dois transistores extras para controle do atraso de propagação. A segunda é a estratégia com entrada diferencial. Esta última possui uma faixa de frequência permissível relativamente maior do que a primeira alternativa. Todavia, o circuito a entrada simples pode ser limitado em corrente, o que reduz a sensibilidade do elemento de atraso ao ruído (PATEL et al., 2014), o que fez com que este fosse escolhido para a proposta deste documento.

O esquemático do elemento de atraso com entrada simples é apresentado na Figura 3.4, já com o segundo inversor na saída incluído, o que o caracteriza como um *buffer*.

Figura 3.4 – Esquemático do elemento de atraso utilizado no VCDL.



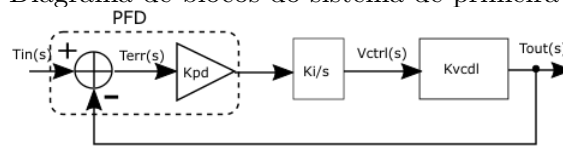
Fonte: Autoria própria

A tensão de controle, V_{ctrl} , determina o atraso das células inversoras constituintes do caminho de atraso controlado por tensão. Este circuito representa um inversor estendido, com dois transistores (um PMOS e um NMOS) adicionais, seguido de um outro inversor, na configuração de limitação de corrente (*current-starving*, do inglês). Nota-se que é o espelho de corrente na entrada, em conjunto com a tensão de controle, que realiza essa limitação de corrente que irá determinar a resistência equivalente do *buffer*, variando então o atraso. Essa configuração com o espelho de corrente torna o circuito menos sensível a variações na tensão de alimentação, V_{DD} (PATEL et al., 2014).

Em seguida, após 8 elementos de atrasos idênticos ao da Figura 3.4, o sinal de saída do último elemento é fornecido para o detector de fase (PFD). É nesta etapa de realimentação que ocorre o ajuste fino do DLL para fornecer as saídas igualmente defasadas, com defasagem de $\frac{T_{osc}}{8}$, sendo efetuado pela malha de realimentação do DLL. Com esta configuração, o DLL com realimentação pode ser representado por um sistema de primeira ordem, conforme diagrama de blocos da Figura 3.5 (WESTE; HARRIS, 2005).

Por ser um sistema de primeira ordem, o filtro passa-baixa pode ser constituído

Figura 3.5 – Diagrama de blocos do sistema de primeira ordem do DLL.

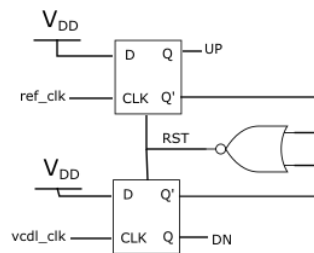


Fonte: Adaptado de (WESTE; HARRIS, 2005)

apenas de um capacitor simples.

Os dois elementos que realizam o controle de fato do DLL são o PFD e o *charge pump* (i.e., fornecedor de carga). O PFD detecta a diferença entre a fase do sinal de referência do oscilador e a fase do sinal de saída do último elemento de atraso. Esta diferença é convertida em um pulso com duração proporcional à defasagem associada. Tal funcionalidade pode ser implementada via o circuito da Figura 3.6, um detector de fase e frequência implementado com dois *flops* e uma porta lógica NOR.

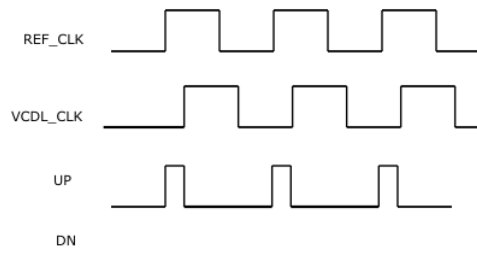
Figura 3.6 – Circuito do Detector de Fase e Frequência (PFD).



Fonte: Autoria própria

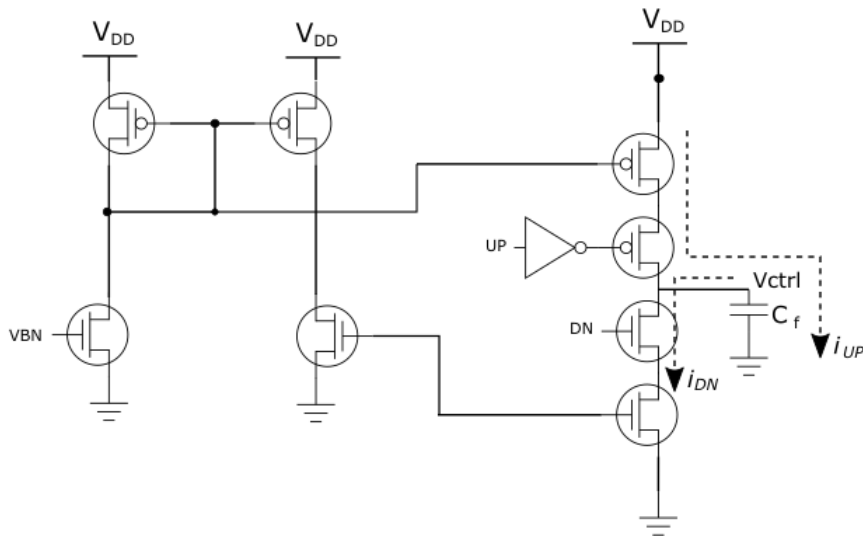
Na Figura 3.7, observa-se o diagrama de tempo para os sinais do PFD. Nesta representação, o sinal de referência (REF_CLK) está adiantado em relação ao sinal do VCDL (VCDL_CLK), o que faz com que o sinal UP possua um pulso cuja duração é proporcional a essa defasagem. Na situação inversa, com o sinal VCDL_CLK adiantado em relação a REF_CLK, seria o sinal de DN que teria esta largura de pulso, enquanto que o sinal de UP permaneceria em baixo.

Figura 3.7 – Comportamento funcional do PFD representado por seu diagrama de tempo.



Fonte: Autoria própria

O passo seguinte, efetuado pelo *charge pump* em conjunto com o filtro passa baixa, representado pelo capacitor C_f da Figura 3.8, consiste em converter esta largura de pulso em um valor de tensão, que será a tensão de controle V_{ctrl} .

Figura 3.8 – Esquemático do *charge pump* com o filtro passa-baixa incluído.

Fonte: Autoria própria

Com esse objetivo, utiliza-se um inversor expandido, cujos PMOS e NMOS extras tem seu tempo de ativação, ou seja, o tempo em que estes permanecem conduzindo, controlados pelos sinais de UP e DN do PFD. Quando o sinal de referência está adiantado em relação ao sinal do VCDL, o sinal de UP apresentará um pulso com largura proporcional a essa diferença. Sendo assim, o transistor PMOS passa a conduzir, fornecendo uma corrente i_{UP} para o capacitor, que causará uma elevação da tensão de saída V_{ctrl} . Isto resultará em uma diminuição dos atrasos dos elementos da VCDL, o que fará com que o sinal de VCDL se iguale ao sinal de referência. Na situação oposta, quando o sinal de VCDL está adiantado em relação a REF_CLK, o pulso do sinal DN irá permitir que o transistor NMOS conduza, retirando carga do capacitor e diminuindo a tensão V_{ctrl} . Desse modo, o sinal de VCDL_CLK é atrasado e se iguala ao sinal de referência.

Além do inversor expandido, tem-se também dois espelhos de corrente. Esses espelhos de correntes são necessários para igualar as correntes i_{UP} e i_{DN} , evitando assim prováveis defasagens indesejadas. Adicionalmente, esta estrutura é menos sensível ao ruído da rede de alimentação (SALMAN; FRIEDMAN, 2002).

Todavia, o circuito da Figura 3.8 ainda apresenta uma limitação importante, que consiste no compartilhamento de carga entre o filtro com as capacitâncias dos transistores adjacentes. Para eliminar esse problema, pode-se utilizar um *buffer* na saída do filtro, o que irá garantir que a tensão V_{ctrl} permaneça constante (SALMAN; FRIEDMAN, 2002). Por outro lado, essa solução acrescenta consideravelmente a área do *charge pump*, pois o *buffer* utilizado possui uma implementação complexa, devido a necessária alta impedância de entrada e ganho unitário.

A metodologia nessa situação consiste em averiguar se a diminuição na tensão de controle irá de fato afetar o ajuste fino do DLL. Para uma parte considerável dos casos, o que

inclui a proposta deste trabalho, esta redução não representa mais do que 5% do valor total da carga (SALMAN; FRIEDMAN, 2002), por isso não irá afetar significativamente o ajuste do DLL. Desta forma, optou-se por não utilizar o *buffer* para corrigir o compartilhamento de carga entre o filtro e os transistores.

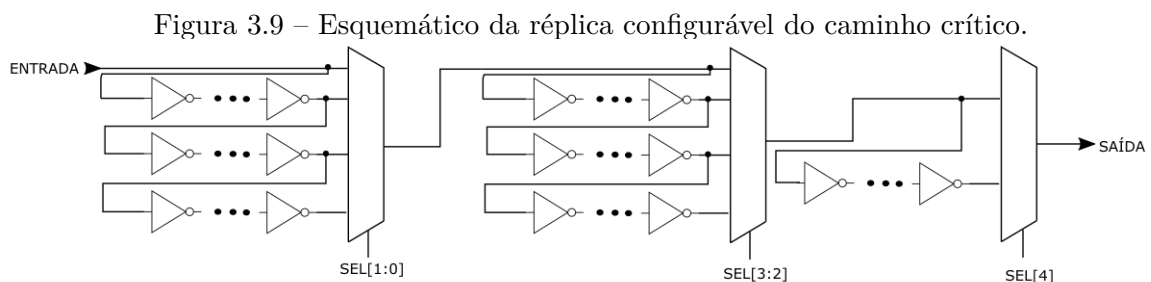
Por fim, com V_{ctrl} fornecida pelo filtro passa-baixa, a malha de controle é fechada e o sistema deve alcançar a estabilidade dentro das condições estabelecidas (i.e., referência do oscilador fixa).

3.2 Réplica Configurável do Caminho Crítico

A variabilidade da frequência do sinal de relógio, o que o confere a característica de ser adaptativo, é oriunda dos blocos de réplicas configuráveis do caminho crítico, conforme Figura 3.1.

Essa propriedade é derivada de um conceito simples apresentado no capítulo 2, que é o atraso de propagação de um circuito CMOS. Conforme foi derivado pela Equação 2.6, o atraso de propagação de um sinal em um circuito CMOS é dependente da tensão de alimentação, a partir de uma relação quadrática. Com isso, ao variar a tensão o atraso de propagação irá variar proporcionalmente, o que pode ser um contexto interessante para gerar o sinal de relógio adaptativo.

Ao utilizar o sinal atrasado após percorrer um caminho de lógica CMOS para gerar pulsos, é possível utilizar esse pulso para gerar as bordas de um sinal de relógio. Na Figura 3.9, a estrutura implementada é apresentada. O bloco de réplica do caminho crítico consiste em uma cadeia de inversores conectadas a um multiplexador, cuja saída depende de um registro configurável de 5 *bits*, denominado como SEL[5:0].



Fonte: Autoria própria

A saída da primeira réplica irá reiniciar assincronamente o sinal de relógio, o que significa que, assim que o sinal de relógio trafegar a primeira réplica, este será levado ao nível lógico zero, gerando assim o período de tempo em que o sinal de relógio permanece em nível lógico alto, isto é, o *duty cycle* do sinal de relógio.

A segunda réplica, em conjunto com o controlador, irá determinar a duração em que o relógio adaptativo permanece em baixo, pois a borda de subida irá ser gerada quando

o sinal de relógio percorrer a segunda réplica, gerando um pulso que irá determinar qual referência do DLL deverá ser selecionada, de acordo com o seletor de borda no controlador.

De tal maneira, o sinal de relógio resultante é síncrono com a referência do DLL, além de ter seu período configurável pelo número de inversores selecionados pelo registro SEL, que determina o tempo de atraso de propagação daquele caminho.

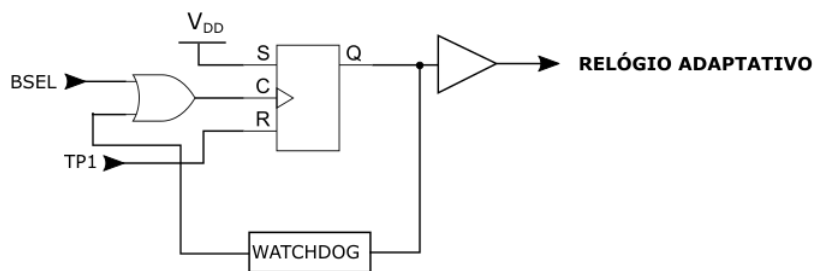
A denominação deste módulo como réplica do caminho crítico reside no fato de que tal circuito propõe simular o atraso do caminho crítico real do processador, através da quantidade de inversores que o sinal de entrada terá que percorrer.

3.3 Controlador do Gerador de Relógio

O bloco constituinte da arquitetura proposta com maior complexidade é o controlador. A justificativa para isso está na necessidade de se detectar a borda de subida entre as 8 saídas do DLL mais próxima no momento em que o pulso da segunda réplica chega ao controlador.

Conforme já explanado na introdução deste capítulo, o controlador do gerador de relógio adaptativo é o bloco responsável por sincronizar o relógio com as referências do DLL, além de monitorar a estabilidade do sinal. Relembrando a visão geral da arquitetura fornecida na Figura 3.1, destaca-se que o controlador é constituído por três componentes principais: o detector de borda, o seletor de borda e o gerador de relógio. O gerador de relógio está esquematizado na Figura 3.10, com BSEL como sendo a saída do seletor de borda.

Figura 3.10 – Circuito do gerador de relógio do controlador.

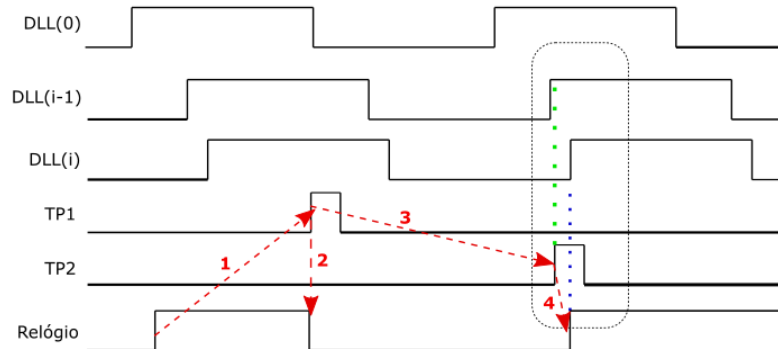


Fonte: Autoria própria

Para ilustrar o funcionamento geral do controlador, um diagrama de tempo para os principais sinais envolvidos no processo é fornecido na Figura 3.11. No diagrama, TP1 e TP2 são os pulsos gerados pelos geradores de pulsos após uma borda de subida ser detectada na saída das réplicas dos caminhos críticos.

A operação do controlador pode ser dividida em 4 etapas. Na primeira etapa, após o sinal de relógio ter se propagado pela primeira réplica de caminho crítico, um pulso é gerado, onde este irá ativar a entrada de *reset* assíncrono do gerador de relógio, fazendo

Figura 3.11 – Diagrama de tempo para uma operação usual do controlador do gerador de relógio.



Fonte: Autoria própria

com que o sinal de relógio (Relógio na Figura 3.11) vá para o nível lógico baixo (evento demarcado como 2 na Figura 3.11). Desta forma, o tempo de propagação para a réplica 1 define o período em que o sinal de relógio permanecia em nível alto, determinando assim o *duty cycle* do *clock*. Na etapa 3, após o sinal de relógio se propagar pela segunda réplica, a próxima borda de subida dos sinais do DLL deve ser selecionada. Para fazer essa detecção, basta apenas fazer uma operação lógica AND entre os sinais adjacentes do DLL. Caso o sinal anterior (DLL(i-1)) esteja em nível lógico e o sinal adjacente (DLL(i)) esteja em nível lógico baixo, então a próxima borda de subida ocorrerá no sinal DLL(i). Esta operação lógica é realizada pelo detector de borda, que envia esse sinal para o seletor de borda, que consiste em um multiplexador que irá selecionar a próxima borda de subida, neste caso para o sinal DLL(i). Esta identificação da borda de subida desejada do DLL e consequente sincronização está indicada pelo número 4 na Figura 3.11.

Por outro lado, como os sinais do DLL são gerados de maneira independente dos pulsos gerados nas saídas das réplicas, estes sinais (DLL[i] com TP1 e TP2) são completamente assíncronos, o que pode gerar uma situação de metaestabilidade. Uma metaestabilidade pode ocorrer quando o tempo de *setup* de um registrador não é respeitado ou quando o pulso de *clock* de um registro é muito curto (RABAEY; CHANDRAKASAN; NIKOLIC, 2003). Para evitar que uma possível metaestabilidade leve o sinal de relógio a ficar permanentemente em nível lógico baixo, uma unidade de *watchdog* monitora a saída do controlador para verificar se esta está oscilando. Caso o sinal de relógio permaneça em nível lógico baixo por mais de três ciclos de *clock* do sinal de referência do DLL, onde a última saída do DLL foi escolhida, um pulso é enviado ao gerador de relógio e este retoma sua oscilação.

Esta sensibilidade às situações de metaestabilidade é uma das principais desvantagens da arquitetura proposta, pois pode permitir com que o sinal de relógio adaptativo fique parado por até 3 períodos do sinal de referência do DLL. Entretanto, como a frequência de referência do DLL é a frequência do oscilador fixo, que para essa proposta foi considerada como sendo 1 GHz, em caso de metaestabilidade, o relógio iria voltar à operação em

pouco tempo. Além disso, em situações de metaestabilidade é natural que a saída do *flop* vá para nível lógico alto ou baixo, sem permanecer em um estado intermediário, o que devolveria a oscilação ao *clock* adaptativo, não necessitando da intervenção do *watchdog* (KWAK; NIKOLIC, 2015).

4 Resultados e Discussões

Neste capítulo apresentam-se e discorre-se sobre os resultados obtidos, além de esclarecer quais ferramentas foram necessárias para sua obtenção.

A arquitetura proposta pode ser dividida em dois domínios, o analógico e o digital. O *Delay Locked-Loop* é o módulo analógico do sistema, enquanto que a geração e monitoramento do sinal de relógio são feitos digitalmente. Sendo assim, utilizou-se duas ferramentas para a implementação e verificação da proposta, uma para análise de sistemas analógicos e a outra para simulação e sintetização de circuitos digitais. A ferramenta para análise analógica escolhida foi o ambiente de simulação conhecido como *Virtuoso* da Cadence Inc. A tecnologia disponibilizada e empregada neste trabalho foi a denominada como FDSOI-28 nm, de *Fully Depleted Silicon On Insulation*, com o comprimento do canal do transistor como sendo 28 nanômetros.

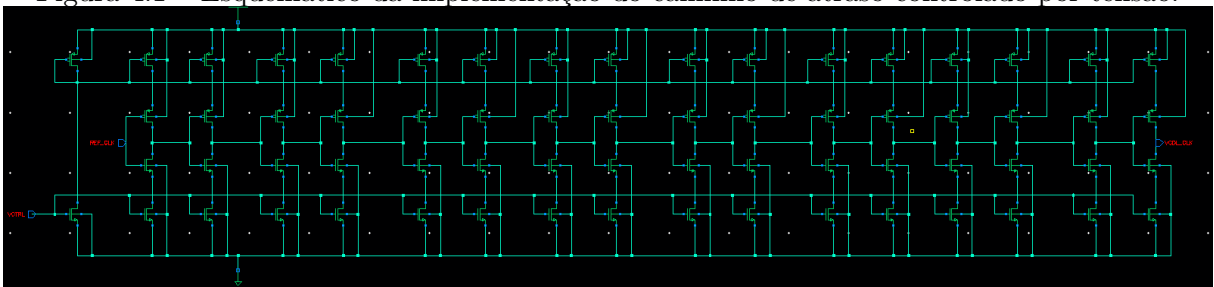
Já para a implementação do controlador, utilizou-se a metodologia de projeto de linguagem de descrição de hardware, em que a linguagem escolhida foi *SystemVerilog*, conforme especificação IEEE 1800-2009. Para sintetizar o código de descrição do controlador, utilizou-se o *software* Genus da Cadence Inc., na sua versão 14.7.

4.1 *Delay Locked-Loop*

O DLL, conforme apresentado no Capítulo 3, é composto de 4 blocos principais, sendo eles o caminho de atraso controlado por tensão (VCDL), o detector de fase e frequência (PFD), o fornecedor de carga (*charge pump*) e o filtro passa-baixa.

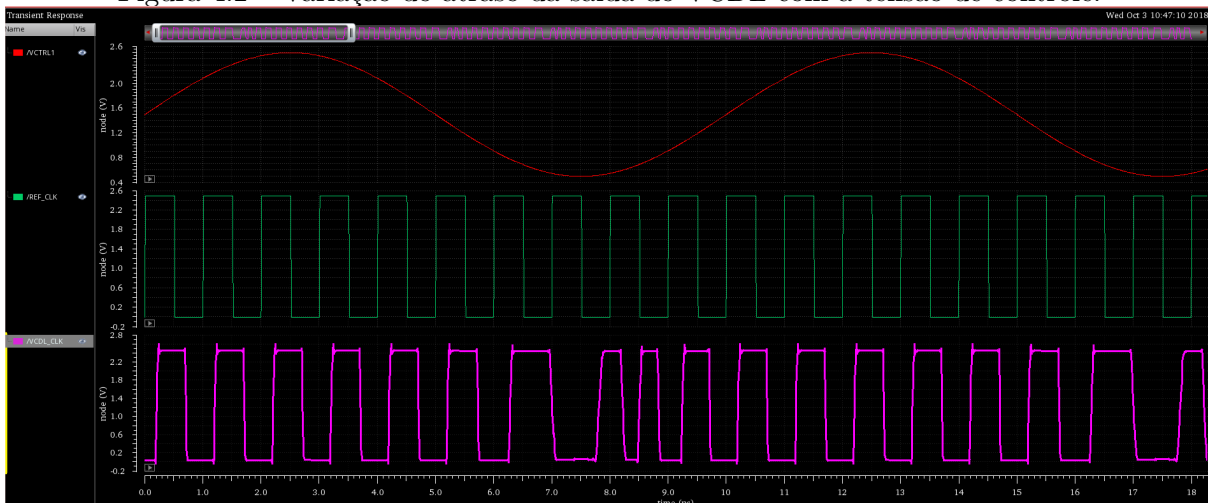
Os circuitos destes foram elaborados conforme as Figuras 3.4, 3.6 e 3.8 no Cadence Virtuoso. O esquemático do VCDL completo, com seus 8 elementos constituintes, é apresentado na Figura 4.1. Apenas os sinais de entrada, controle e a saída do último elemento de atraso são mostrados, para evitar a poluição da figura.

Figura 4.1 – Esquemático da implementação do caminho de atraso controlado por tensão.



Para verificar o atraso gerado pelos 8 elementos do VCDL de acordo com a tensão de controle, submeteu-se o circuito da Figura 4.1 à uma tensão de controle senoidal. Os resultados apresentados na Figura 4.2 demonstram que o atraso da VCDL é dependente da tensão de controle V_{ctrl} , conforme esperado. Nota-se também que, com o aumento de V_{ctrl} o atraso diminui, o que já foi discutido no capítulo anterior. Desse modo, ao detectar que o sinal de saída está atrasado em relação à referência, a tensão de controle deve aumentar para reduzir esse atraso, o que é feito através do sinal UP do detector de fase, conforme discutido a seguir.

Figura 4.2 – Variação do atraso da saída do VCDL com a tensão de controle.



Fonte: Autoria própria

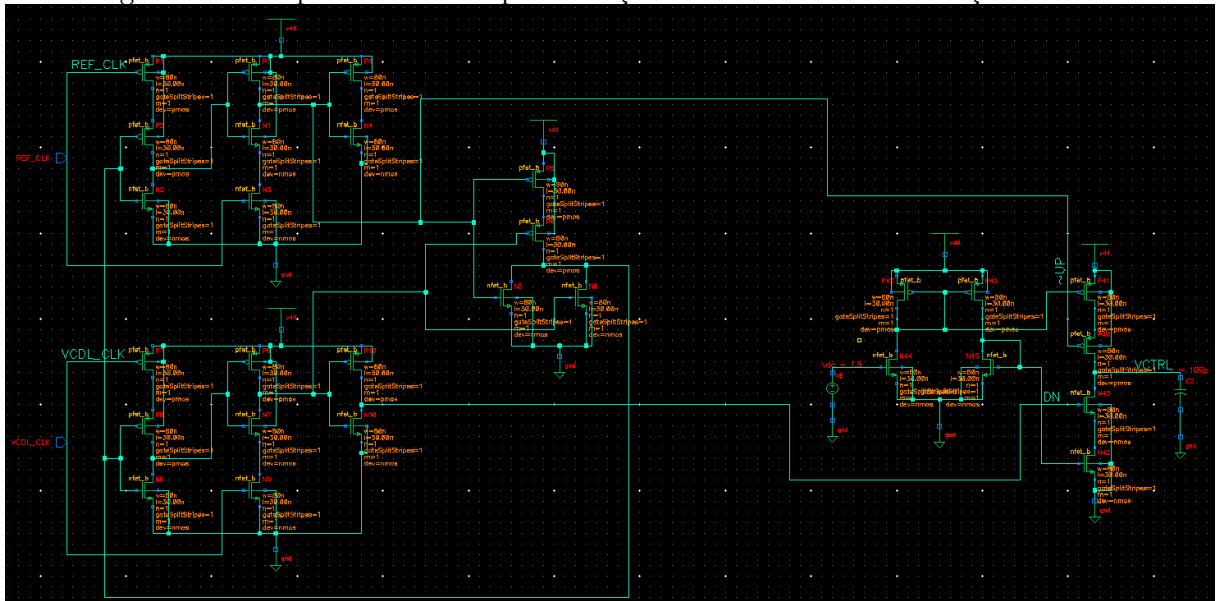
A implementação do detector de fase foi combinada com a implementação para o *charge pump* e para o filtro passa-baixa, o que resultou no circuito de realimentação completo da Figura 4.3.

Nota-se que o detector de fase foi implementado no nível de transistores, a fim de representar mais fielmente seu comportamento, ao invés de utilizar portas lógicas discretas. Os sinais resultantes do PFD são apresentados nas Figuras 4.4 e 4.5.

É perceptível que existem picos de tensão para o sinal que deveria permanecer inativo, o que irá gerar uma dissipação de potência entre a alimentação e o terra. Estes picos são oriundos dos efeitos das capacitâncias parasitas dos transistores da tecnologia FDSOI-28 nm. Entretanto, a duração destes corresponde a menos de 3% do período do sinal de UP ou DN, fazendo com que a dissipação de curto-circuito resultante seja negligível (RABAEY; CHANDRAKASAN; NIKOLIC, 2003)(SALMAN; FRIEDMAN, 2002).

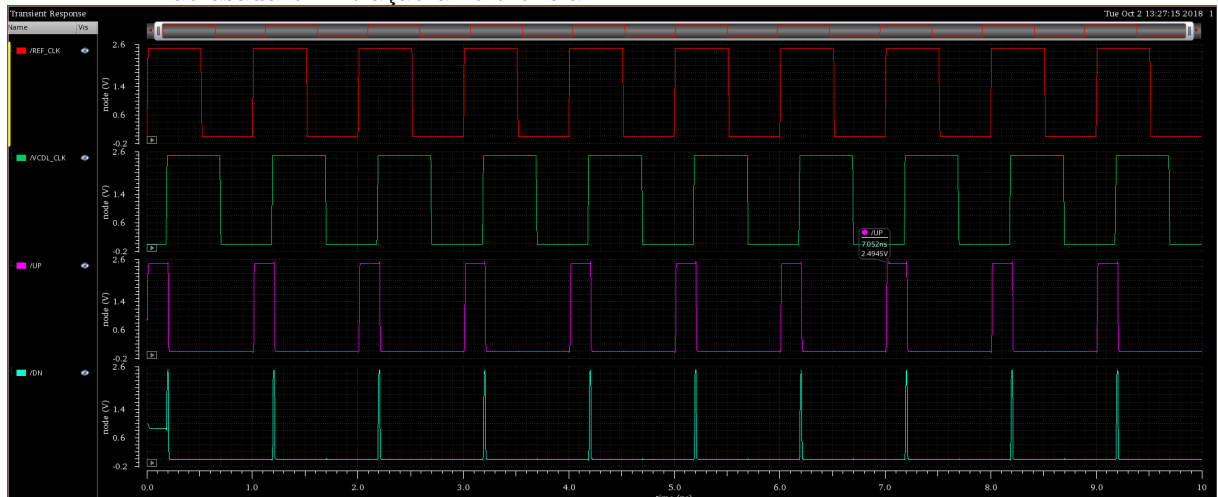
Estes sinais de controle, UP e DN, são responsáveis por determinar o valor da tensão de controle através do *charge pump* e do filtro de saída. Na Figura 4.6, mostra-se a tensão produzida para um determinado período de atraso do sinal de saída em relação à referência. Nota-se que existe uma queda de 0,54% no valor da tensão de controle. Esta

Figura 4.3 – Esquemático da implementação da malha de realimentação do DLL.



Fonte: Autoria própria

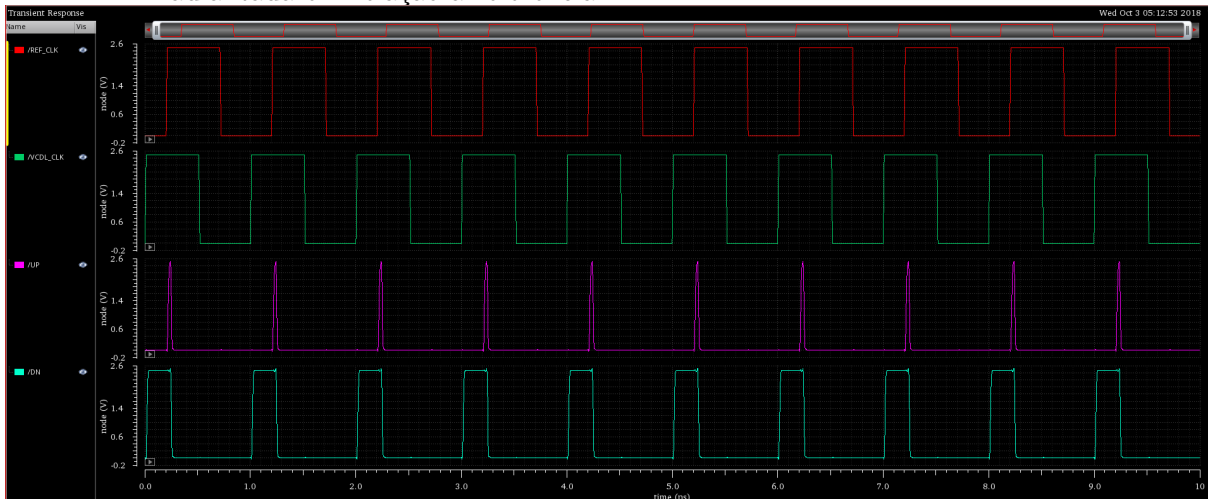
Figura 4.4 – Geração dos sinais de controle UP e DN do detector de fase e frequência - saída atrasada em relação à referência.



Fonte: Autoria própria

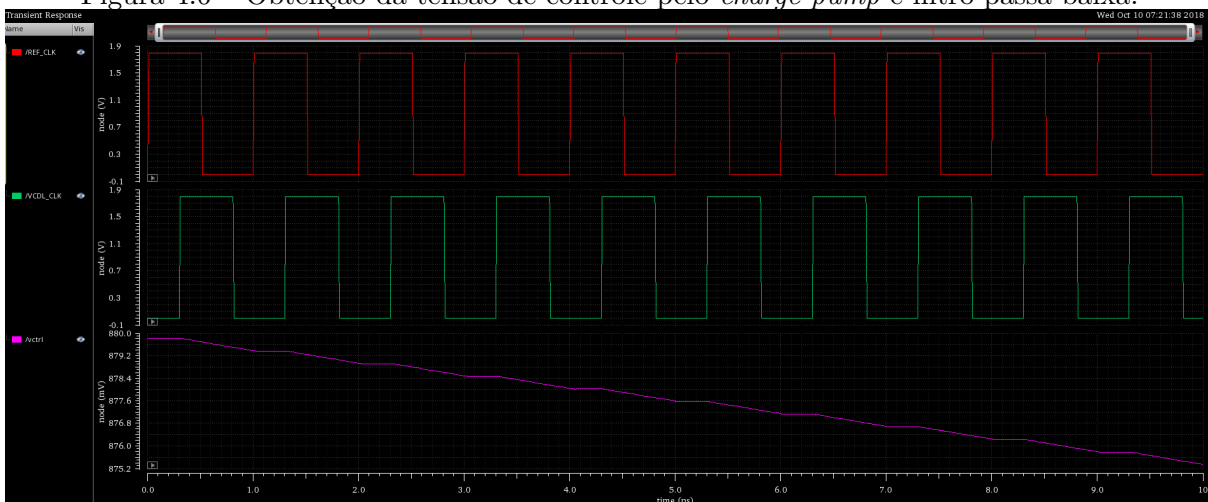
diminuição é ocasionada pelo efeito de compartilhamento de carga comentado no capítulo anterior, cuja solução consiste na utilização de um *buffer* com alta impedância de entrada. Seguindo a discussão do capítulo 3, esta alternativa não é interessante devido ao aumento de área associado e também que a redução de tensão foi inferior a 1%, o que não representa um efeito significativo para o DLL (SALMAN; FRIEDMAN, 2002).

Figura 4.5 – Geração dos sinais de controle UP e DN do detector de fase e frequência - saída adiantada em relação à referência.



Fonte: Autoria própria

Figura 4.6 – Obtenção da tensão de controle pelo *charge pump* e filtro passa-baixa.



Fonte: Autoria própria

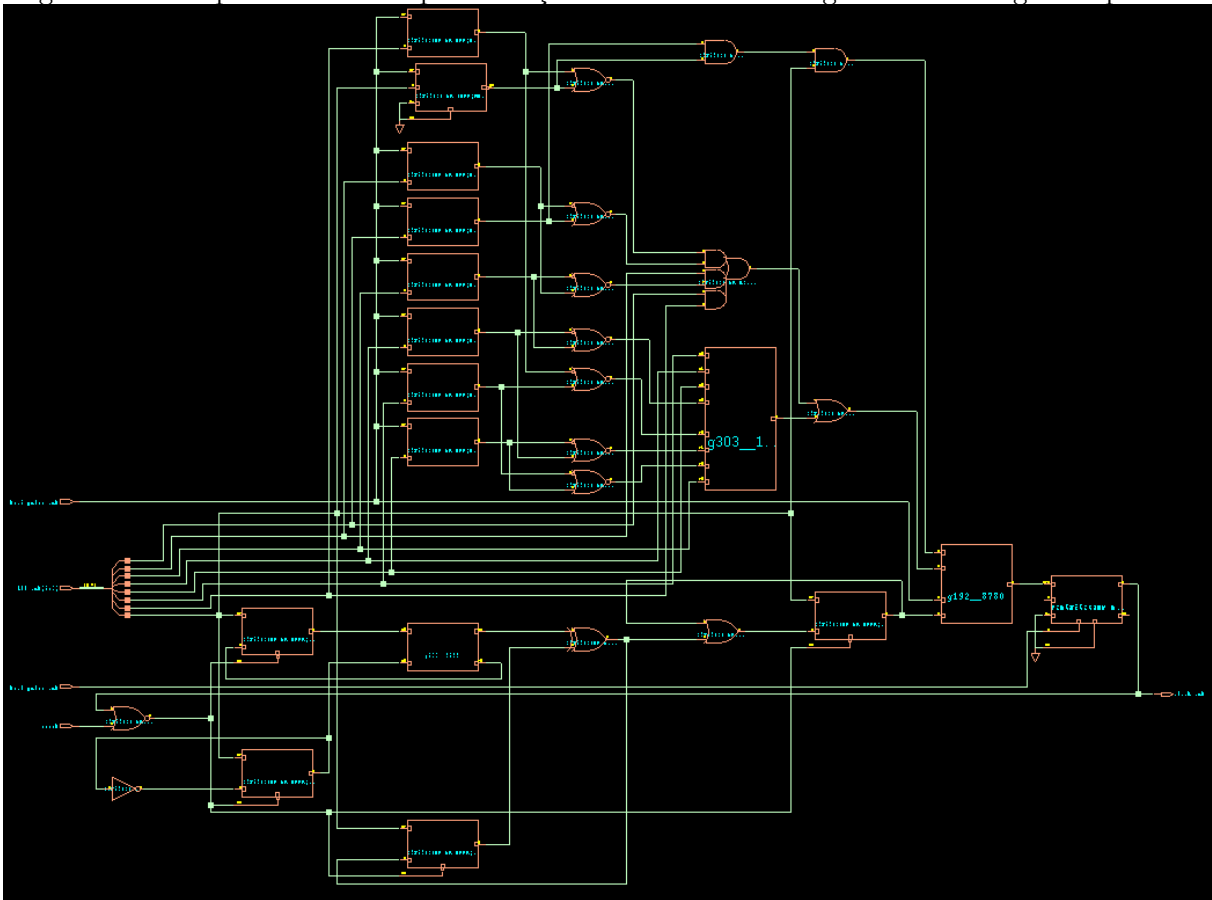
4.2 Controlador e Réplica Configurável

A implementação da seção digital do sistema foi realizada através da linguagem *SystemVerilog*. O código para a implementação do controlador está disponível na íntegra no Anexo A.

Após a sintetização, que consiste em transformar a descrição do *hardware* em *SystemVerilog* em um circuito digital, obteve-se o circuito apresentado na Figura 4.7.

Nota-se que este módulo apresenta a maior complexidade estrutural da proposta, contendo um número de portas lógicas e transistores significativamente maior do que os módulos analógicos anteriores. Desta forma, a área ocupada por este será maior. Os resultados para a área, número de portas lógicas, e estimativa de consumo de potência são

Figura 4.7 – Esquemático da implementação do controlador do gerador de relógio adaptativo.



Fonte: Autoria própria

Tabela 4.1 – Resultados para os parâmetros físicos do controlador digital.

| Parâmetros Físicos do Controlador | | | | | |
|-----------------------------------|------------------------|---------|----------------|----------|---------|
| Área | | | | | |
| Total | 44,173 μm^2 | | Portas Lógicas | 135 | |
| Consumo de Potência | | | | | |
| Total | 43,3 uW | Leakage | 1,48 uW | Dinâmica | 41,8 uW |
| Frequência | | | | | |
| 1 GHz | | | | | |

apresentados na Tabela 4.1.

É perceptível que mesmo com 135 portas lógicas, apenas uma área de 44,17 μm^2 foi utilizada. Além disso, a estimativa de consumo de energia, baseado em uma frequência de referência do DLL de 1 GHz, foi de apenas 43,3 μW , sendo 96,5% deste valor oriundo da potência dinâmica, o que reforça as estratégias de projeto discutidas nos capítulos 1 e 2 orientadas à redução de consumo de potência dinâmica.

Para comprovar a funcionalidade do controlador e do sistema como um todo, utilizou-se uma ferramenta de simulação, tendo como modelo o *netlist* gerado pela sintetização da

arquitetura proposta, para verificar a geração do relógio adaptativo. A simulação, que é limitada ao nível funcional, sem avaliar portanto assim efeitos de capacitâncias parasitas e queda de tensão nas interconexões entre as portas lógicas, tem como intuito estimular o módulo digital com as referências geradas pelo DLL e averiguar se, com uma variação da tensão de alimentação, a frequência do relógio varia correspondentemente.

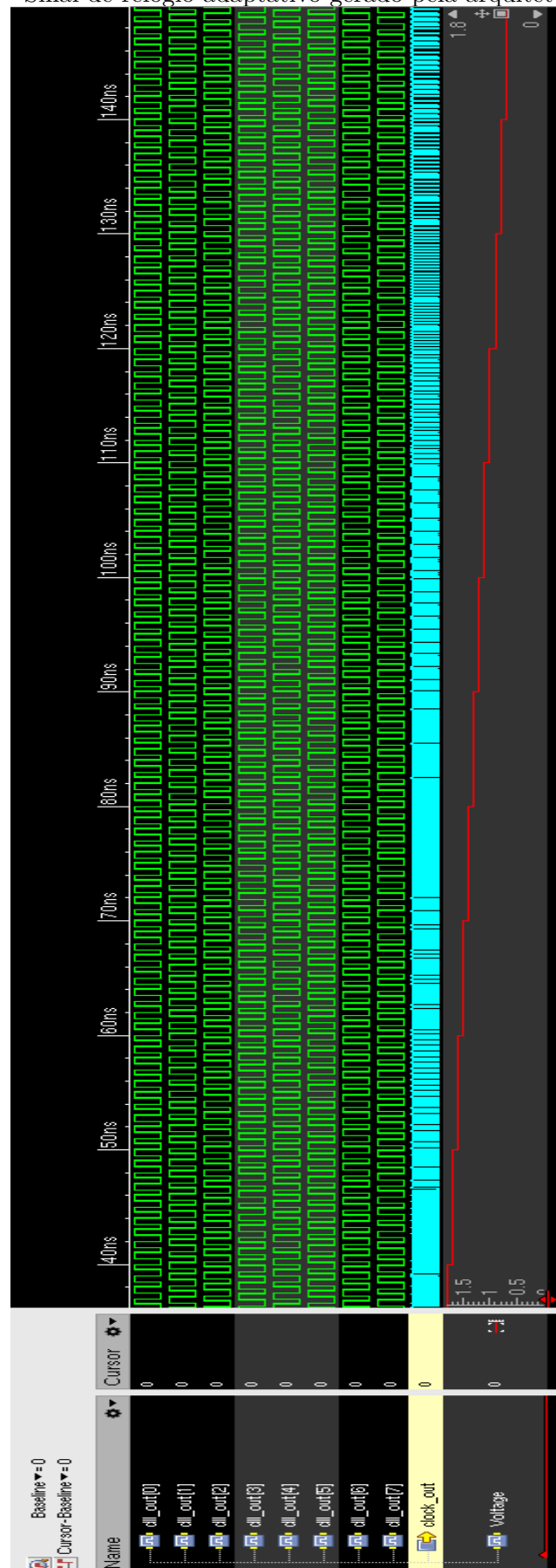
Os resultados para esta análise funcional seguem na Figura 4.8.

Para gerar a variação da tensão necessária, foi necessário configurar seu valor manualmente, onde este era utilizado como um argumento para configuração da simulação, assim como a frequência do sinal de referência (i.e., do oscilador fixo) para o DLL. Através destes resultados, é possível averiguar que com uma diminuição da tensão de alimentação, representada por *Voltage* na Figura 4.8, a frequência do sinal de relógio (i.e., *clock_out*) também diminui proporcionalmente.

Nesse sentido, é possível então concluir que a funcionalidade almejada foi implementada. A variação de frequência do relógio é instantânea com a alteração no valor da tensão de alimentação. Disto segue que, de acordo com o que foi discutido no capítulo 2, a demanda para o controlador de potência (PMC) é consideravelmente reduzida, pois será necessário apenas determinar a tensão que corresponde a frequência desejada. Tal estratégia pode ser feita por meio de reguladores programáveis, como foi feito por Keller et al. (2016a).

Por fim, ressalta-se que, devido a dependência do período do relógio com o atraso das réplicas dos caminhos críticos, o processador possui maior segurança para operar em tensões mais baixas, com um atraso maior, pois a frequência do seu sinal de *clock* está intrinsecamente ligada aos atrasos dos caminhos críticos.

Figura 4.8 – Sinal de relógio adaptativo gerado pela arquitetura proposta.



Fonte: Autoria própria

5 Conclusões

Neste trabalho, o principal fator que direcionou as decisões de metodologia e projeto tomadas foi a intenção de promover uma redução no consumo de energia de sistemas eletrônicos integrados. O desenvolvimento da proposta seguiu as diretrizes de sistemas voltados para a Internet das Coisas, especialmente no âmbito das Redes de Sensores Sem Fio. Para tanto, uma revisão bibliográfica sobre técnicas de redução de consumo em nós sensores foi realizada na introdução deste documento.

Deste modo, o contexto do projeto foi delimitado baseando-se na aplicabilidade de técnicas de escalonamento de tensão e frequência realizadas inteiramente em *hardware*. A proposta da arquitetura de gerador de relógio adaptativo teve como requisitos e parâmetros, que moldaram sua estrutura, as atuais capacidades dos circuitos integrados modernos, mais especificamente àqueles baseados na tecnologia FDSOI 28 nm. Entre eles, a possibilidade de integração de reguladores controlados, o que estimulou o uso de um gerador de relógio dependente da tensão de alimentação, destacou-se durante todo o trabalho.

Com a definição dos circuitos analógicos e digitais realizadores da funcionalidade da arquitetura proposta, as simulações analógicas e funcionais foram efetuadas, que comprovaram a viabilidade de proposta apresentada nesse documento.

Por fim, com a verificação das propriedades da proposta concluída, tanto a nível funcional como a nível de implementação física, o que incluiu a análise de área e consumo de energia para o controlador digital, o intuito de motivar a adoção de técnicas de controle de consumo de energia totalmente integradas e autônomas, sem a necessidade de intervenção do processador externo, foi alcançado.

5.1 Sugestões para trabalhos futuros

Uma das fortes possibilidades de incremento de resultados que emana ao fim deste trabalho, está na comprovação da proposta em mais alto nível. Nesse sentido, sugere-se para atividades futuras a simulação da utilização desta arquitetura em conjunto com um processador, onde este executará determinada rotina de tarefas. Por conseguinte, seria possível comparar o consumo de energia de um sistema microprocessado com e sem a arquitetura proposta, verificando assim a eficácia real do gerador de relógio adaptativo.

Referências

- AKYILDIZ, I. F. A survey on sensor networks. *IEEE Communications Magazine*, v. 40, n. 8, p. 102–114, 2002. Citado na página 29.
- ANTONIO, R. A. et al. Implementation of dynamic voltage and frequency scaling on a processor for wireless sensing applications. *Proceedings of 2017 Region 10 Conference*, p. 2955–2960, Novembro 2017. Citado 2 vezes nas páginas 29 e 30.
- BURD, T. D. *Energy-Efficient Processor System Design*. Tese (Doutorado) — Tese (Doutorado)—University of California at Berkeley, 2001. Citado na página 33.
- CHANDRAKASAN, A. P. *Low-Power CMOS Digital Design*. 4. ed. [S.l.]: IEEE Journal of Solid-State Circuits, 1992. v. 27. 473-484 p. Citado 2 vezes nas páginas 33 e 34.
- EINSTEIN.BR. *Cirurgia Robótica*. 2018. Website do Hospital Albert Einstein. Disponível em: <<https://www.einstein.br/especialidades/cirurgia/programa/cirurgia-robotica>>. Acesso em: 13 out 2018. Citado na página 25.
- ERGIN, O. *Circuit Techniques for Power-Aware Microprocessors*. Tese (Doutorado) — Dissertação (Mestrado)—The State University of New York, 2003. Citado na página 33.
- EVANS, D. *The Internet of Things: How the Next Evolution of the Internet Is Changing Everything*. 2011. Cisco Internet Business Solution Group, Cisco Systems Inc. Disponível em: <http://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf>. Acesso em: 13 out 2018. Citado na página 25.
- EYERMAN, S.; EACKHOUT, L. Fine-grained dvfs using on-chip regulators. *ACM Transactions on Architecture and Code Optimization*, v. 8, n. 1, p. 1–24, Abril 2011. Citado na página 41.
- FEKI, M. A. et al. The internet of things: The next technological revolution. *Computer*, v. 46, n. 2, p. 24–25, Fevereiro 2013. Citado na página 25.
- FRITIAWAN, H. et al. Zigbee based wireless sensor networks and performance analysis in various environments. *15th International Conference on Quality in Research: International Symposium on Electrical and Computer Engineering*, Dezembro 2017. Citado na página 28.
- GHARIB, M.; ABRISHAMIFAR, A. A novel low-power and high-performance dual-loop dll with linear delay element. *51st Midwest Symposium on Circuits and Systems*, p. 763–766, Agosto 2008. Citado na página 43.
- GUPTA, A.; JHA, R. K. A survey of 5g network: Architecture and emerging technologies. *IEEE Access*, v. 3, n. 1, p. 1206–1232, Julho 2015. Citado 2 vezes nas páginas 25 e 26.
- HONG, I. A survey on sensor networks. *In Real-Time Systems Symposium*, p. 178–187, 1998. Citado na página 30.
- ITOH, T. Ultra low power wireless sensor nodes for expanding application of the internet of things. *2016 IEEE CPMT Symposium Japan*, Dezembro 2016. Citado na página 28.

JHANG, Q.; SUN, Y.; CUI, Z. Application and analysis of zigbee technology for smart grid. *International Conference on Computer and Information Application*, p. 171–174, Junho 2010. Citado 2 vezes nas páginas 28 e 33.

KELLER, B. et al. A risc-v vector processing with simultaneous-switching switched-capacitor dc-dc converters in 28 nm fdsoi. *IEEE Journal of Solid-State Circuits*, v. 51, n. 4, p. 930–942, Abril 2016. Citado 3 vezes nas páginas 41, 42 e 58.

KELLER, B. et al. Sub-microsecond adaptative voltage scaling in a 28-nm fd-soi processor soc. *ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference*, p. 269–272, Setembro 2016. Citado 2 vezes nas páginas 41 e 42.

KIM, W. et al. System level analysis of fast, per-core dvfs using on-chip switching regulators. *Proceedings of the IEEE International Symposium on High Performance Computer Architectures*, p. 123–134, Fevereiro 2008. Citado na página 41.

KWAK, J.; NIKOLIC, B. A 550-2260 mhz self-adjustable clock generator in 28nm fdsoi. *IEEE Asian Solid-State Circuits Conference*, p. 18.1–18.4, Novembro 2015. Citado 3 vezes nas páginas 30, 42 e 52.

LUO, G.; ZENG, X. An improved voltage-controlled delay line for delay locked loops. *3rd International Conference on Computer Research and Development*, p. 237–240, Março 2011. Citado na página 43.

MIRAZ, M. H.; ALI, M.; EXCELL, P. A review on internet of things (iot), internet of everything (ioe) and internet of nano things (iont). *2015 Internet Technologies and Applications*, p. 219–224, Setembro 2015. Citado na página 25.

PAL, A. Optimal dual-vt assignment for low-voltage energy-constrained cmos circuits. *Design Automation Conference*, 2002. Citado na página 35.

PATEL, K. et al. Voltage controlled delay line with pfd for delay locked loop in cmos 90nm technology. *International Journal of Research in Electronics and Communication Technology*, v. 1, n. 1, p. 25–30, Janeiro 2014. ISSN 2348-9065. Citado na página 46.

RABAEY, J. M.; CHANDRAKASAN, A.; NIKOLIC, B. *Digital Integrated Circuits: A Design Perspective*. 2. ed. [S.l.]: Prentice Hall Electronics and VLSI Series, 2003. v. 1. 200-205 p. ISBN 0130909963. Citado 3 vezes nas páginas 38, 51 e 54.

SALMAN, E.; FRIEDMAN, E. G. *High Performance Integrated Circuit Design*. 1. ed. [S.l.]: McGraw-Hill, 2002. v. 1. 468-487 p. ISBN 9780071635769. Citado 4 vezes nas páginas 48, 49, 54 e 55.

SCHWAB, K.; MIRANDA, D. *A Quarta Revolução Industrial*. EDIPRO, 2016. ISBN 9788572839785. Disponível em: <<https://books.google.com.br/books?id=0wgcvgAACAAJ>>. Citado na página 25.

SHARMA, H.; SHARMA, S. A review of sensor networks: Technologies and applications. *Proceedings of 2014 RAECS*, p. 1–4, Março 2014. Citado 3 vezes nas páginas 26, 27 e 28.

SULEIMAN, D. Dynamic voltage and frequency scaling for microprocessor power and energy reduction. 2005. Citado na página 29.

TIWARI, V. et al. Reducing power in high-performance microprocessors. In: *Proceedings of the 35th Conference on Design Automation*. [S.l.: s.n.], 1998. Citado na página 33.

WESTE, N. H. E.; HARRIS, D. M. *CMOS VLSI Design: A circuits and systems perspective*. 4. ed. [S.l.]: Pearson Education, 2005. v. 1. ISBN 9789332559042. Citado 2 vezes nas páginas 46 e 47.

YA, F. Z. *16-bit Educational MIPS*. 2013. Disponível em: <http://opencores.org/project_mips16>. Acesso em: 14 out 2018. Citado na página 29.

—

Anexos

ANEXO A – Código fonte do controlador do relógio adaptativo

```

1 // *****
2 // CONTROLLER IMPLEMENTATION
3 // *****
4
5 module controller (
6
7     // Inputs
8
9     input logic [7:0]    dll_out,           // Outputs of the DLL
10
11     input logic         trc1_pulse_out,    // Pulse signal from TRC1
12
13     input logic         trc2_pulse_out,    // Pulse signal from TRC2
14
15     input logic         reset,            // Reset WatchDog counter
16
17     // Outputs
18
19     output reg          clock_out         // Adaptative Clock output
20 );
21
22 // -----
23 // Declaration of internal signals
24 // -----
25
26     logic [7:0]         dll_ref_sel;
27
28     logic               clock_stimuli;
29
30     logic               wdog_out;
31
32     logic               clock_in;
33
34     integer             i;
35
36 // -----

```

```

37 // Declaration of internal registers
38 // -----
39
40 reg [7:0]          dll_reg;
41
42 reg [2:0]          wdog_counter;
43
44 // -----
45 // Implementation of Edge detector for DLL reference select signals
46 // -----
47
48 // Array of registers with TRC2 pulse output as clock input
49
50 always_ff @(posedge trc2_pulse_out)
51 begin
52     dll_reg[7:0] <= dll_out[7:0];
53 end
54
55 // Edge detector and selector
56
57 always_comb
58 begin
59     // Edge detector
60     dll_ref_sel[0] = ~(dll_reg[0]) & dll_reg[7];
61     for(i=1;i<=7;i=i+1)
62     begin
63         dll_ref_sel[i] = ~(dll_reg[i]) & dll_reg[i-1];
64     end
65
66     // Edge selector
67     clock_stimuli = ((dll_ref_sel[0] & dll_out[0]) |
68                     (dll_ref_sel[1] & dll_out[1]) |
69                     (dll_ref_sel[2] & dll_out[2]) |
70                     (dll_ref_sel[3] & dll_out[3]) |
71                     (dll_ref_sel[4] & dll_out[4]) |
72                     (dll_ref_sel[5] & dll_out[5]) |
73                     (dll_ref_sel[6] & dll_out[6]) |
74                     (dll_ref_sel[7] & dll_out[7])) & trc2_pulse_out;
75 end
76
77 assign clock_in = clock_stimuli | wdog_out;
78

```



```
79 // Flop for generating the adaptative clock based on TRC1 and TRC2
80
81 always @(posedge clock_in, posedge trc1_pulse_out)
82 begin
83     if (trc1_pulse_out)
84         clock_out <= 1'b0;
85     else
86         begin
87             clock_out <= 1'b1;
88         end
89 end
90
91 // -----
92 // Implementation of the Watchdog circuit
93 // -----
94
95 // Watchdog logic
96
97 always @(posedge dll_out[0], posedge clock_out, posedge reset)
98 begin
99     // If there is a positive edge on Clock Out, then reset counter and
100     // the generator is working fine
101     if (clock_out || reset)
102     begin
103         wdog_counter = 3'd0;
104         wdog_out = 1'b0;
105     end
106     else
107     begin
108         wdog_counter = wdog_counter + 1;
109         if (wdog_counter > 3'd3)
110             begin
111                 wdog_out = 1'b1;
112             end
113     end
114 end
115
116 endmodule
```