

Bruno Fernandes dos Santos

**Projeto e Verificação de um Temporizador
Digital para Medição de Parâmetros de
Processo de Sistemas com Baixo Consumo de
Energia**

João Pessoa, PB - Brasil

Novembro de 2018

Bruno Fernandes dos Santos

Projeto e Verificação de um Temporizador Digital para Medição de Parâmetros de Processo de Sistemas com Baixo Consumo de Energia

Relatório de Estágio Supervisionado submetido ao Departamento de Engenharia Elétrica da Universidade Federal da Paraíba como parte dos requisitos para obtenção do título de Engenheiro Eletricista.

Universidade Federal da Paraíba – UFPB

Centro de Energias Alternativas e Renováveis – CEAR

Departamento de Engenharia Elétrica

Graduação em Engenharia Elétrica

Orientador: Prof. Dr. Fabrício Braga Soares de Carvalho

João Pessoa, PB - Brasil

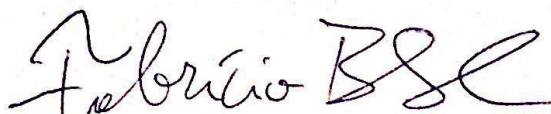
Novembro de 2018

Bruno Fernandes dos Santos

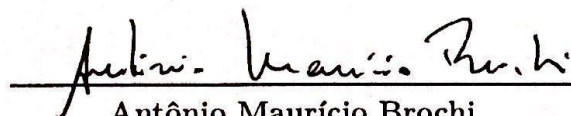
Projeto e Verificação de um Temporizador Digital para Medição de Parâmetros de Processo de Sistemas com Baixo Consumo de Energia

Relatório de Estágio Supervisionado submetido ao Departamento de Engenharia Elétrica da Universidade Federal da Paraíba como parte dos requisitos para obtenção do título de Engenheiro Eletricista.

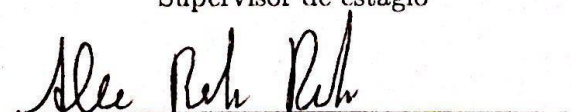
Relatório aprovado. João Pessoa, PB - Brasil, 01 de novembro de 2018:



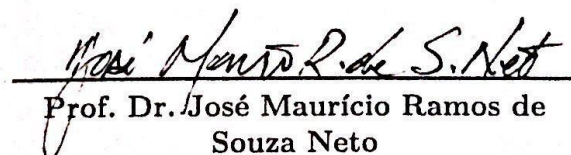
Prof. Dr. Fabrício Braga Soares de
Carvalho
Orientador



Antônio Maurício Brochi
Supervisor de estágio



Alex Rocha Prado
Mentor de estágio



Prof. Dr. José Maurício Ramos de
Souza Neto
Professor Avaliador Interno

João Pessoa, PB - Brasil
Novembro de 2018

IDENTIFICAÇÃO DO ESTÁGIO

EMPRESA

NOME: NXP Semiconductors

ENDEREÇO: Rua James Clerk Maxwell, 400

LOGRADOURO: Condomínio Technopark

CEP: 13069-380

CIDADE/ESTADO: Campinas, SP

TEL: (19) 3783-8638

ESTÁGIO

ÁREA DA EMPRESA: Digital IP Design

DATA DE INÍCIO: 09/01/2018

DATA DE TÉRMINO: 31/12/2018

CARGA HORÁRIA SEMANAL: 30 horas

SUPERVISOR DE ESTÁGIO: Antônio Maurício Brochi

Lista de ilustrações

Figura 1 – Fluxograma simplificado para a criação de um produto.	18
Figura 2 – Fluxograma para o projeto de um circuito integrado.	20
Figura 3 – Metodologia para projeto de um IP digital.	22
Figura 4 – Diagrama de blocos do temporizador digital implementado.	27
Figura 5 – Diagrama de tempo para uma operação de temporização.	28
Figura 6 – Estrutura de um sincronizador.	29
Figura 7 – Temporizador Digital sintetizado com a tecnologia FDSOI-28nm. . . .	30
Figura 8 – Simulação funcional para o modo de operação de interrupção periódica.	31

Lista de tabelas

Tabela 1 – Parâmetros físicos e tecnológicos do temporizador após sintetização. . .	32
---	----

Lista de abreviaturas e siglas

IoT	<i>Internet of Things</i>
RSSF	Redes de Sensores Sem Fio
IP	<i>Intellectual Property</i>
BSTC	<i>Brazil Semiconductor Technology Center</i>
SPS	<i>Semiconductor Product Sector</i>
CPqD	Centro de Pesquisa e Desenvolvimento
CAN	<i>Controller Area Network</i>
SoC	<i>System on Chip</i>
ULA	Unidade Lógico-Aritmética
ADD	<i>Architecture Definition Document</i>
SRS	<i>Semiconductor Reuse Standard</i>
HDL	<i>Hardware Description Language</i>
RTL	<i>Register Transfer Level</i>
STA	<i>Static Timing Analysis</i>
ATPG	<i>Automatic Test Pattern Generation</i>

Lista de símbolos

C_{pd}	Capacitância de dissipação de potência dinâmica
V_{CC}	Tensão de alimentação
f_I	Frequência do sinal de entrada
N_{SW}	Número de bits transicionando
P_T	Potência dinâmica
L	Comprimento do canal do transistor

Sumário

1	INTRODUÇÃO	13
1.1	OBJETIVOS DO ESTÁGIO	14
1.1.1	OBJETIVOS GERAIS	14
1.1.2	OBJETIVOS ESPECÍFICOS	14
1.2	A EMPRESA	14
2	FLUXOGRAMA DE PROJETOS NA NXP SEMICONDUTORES	17
2.1	O PROCESSO DE ESPECIFICAÇÃO DE UM PRODUTO	17
2.2	FLUXO DE PROJETO DE UM CIRCUITO INTEGRADO	19
2.3	PROCESSO DE IMPLEMENTAÇÃO DE UM IP DIGITAL	22
3	IMPLEMENTAÇÃO DO TEMPORIZADOR DIGITAL	25
3.1	DESCRIÇÃO DAS SUAS FUNCIONALIDADES	26
3.2	IMPLEMENTAÇÃO E RESULTADOS	29
4	CONCLUSÕES	33
	REFERÊNCIAS	35

1 INTRODUÇÃO

As revoluções tecnológicas vislumbradas nos séculos XX e XXI tiveram, em grande parte, o desenvolvimento do mercado de semicondutores como um dos principais viabilizadores das suas conquistas. A grande escalabilidade de circuitos integrados motivou o segmento a investir em pesquisa e desenvolvimento de alta tecnologia com o intuito de produzir transistores cada vez menores. Desta forma, foram desenvolvidos chips com uma funcionalidade crescente, tornando os sistemas embarcados, por exemplo, soluções comercialmente e tecnologicamente viáveis para todos os segmentos da indústria, aumentando assim a produtividade e competitividade dos setores que aderiram aos sistemas embarcados como solução tecnológica, resultando assim em um aumento da qualidade geral do serviço e/ou produto.

Mais especificamente, um dos segmentos modernos que vem se destacando como tema de pesquisas acadêmicas e objeto de interesse de investidores é a Internet das Coisas (*Internet of Things* (IoT), em inglês). A Internet das Coisas representa um grande salto para a escalabilidade e abrangência de sistemas embarcados, tanto residenciais, comerciais e industriais, pois possibilita a integração via rede de grandes sistemas eletrônicos. Sistemas IoT encontram uma aplicabilidade generosa, encontrando funcionalidade em sistemas de monitoramento militar, sensoriamento de funções vitais, gerenciamento de sistemas ambientais, entre outros (CHEN, 2010). Devido ao grande número de sensores utilizados em aplicações industriais, por exemplo, torna-se necessária a integração destes e de seus dados através de uma rede de comunicação, estruturada por meio de uma Rede de Sensores Sem Fio (RSSF).

Além dos desafios naturais das topologias de RSSF e de coordenação de dados, existem os requisitos tecnológicos para que tais sistemas sejam viabilizados comercialmente. Por exemplo, os nós de sistemas IoT costumam operar alimentados por baterias, o que restringe fortemente o consumo de energia dos sensores e sistemas embarcados de processamento (ANTONIO; COSTA, 2017).

Esta restrição de projeto de circuitos integrados relativa ao consumo de energia introduz uma série de desafios e redirecionamentos quanto às metodologias e processos de fabricação de circuitos integrados, além de ser necessário reavaliar a tecnologia utilizada, de acordo com seus parâmetros de consumo de energia, como o *leakage* (traduzido aqui para desperdício/vazamento de corrente elétrica) dos transistores.

O desperdício (i.e., *leakage*) de energia varia exponencialmente tanto com parâmetros da tecnologia, como com o comprimento do canal dos transistores CMOS (*Complementary Metal-Oxide Semiconductor*), assim como com parâmetros operacionais, como a

tensão de alimentação (RABAHEY; CHANDRAKASAN; NIKOLIC, 2001). Vale ressaltar que estas limitações trazem decisões conflitantes quanto ao projeto de circuitos digitais, pois ao aumentar-se a tensão de operação para priorizar o desempenho do sistema, reduz-se a eficiência do mesmo através do aumento do desperdício de energia dos transistores.

Tais desafios constituem os principais motores de inovação nas empresas de semicondutores atuais, principalmente na NXP Semiconductors, em seu centro de Campinas, São Paulo. Sendo assim, é imprescindível que os projetistas de IP (*Intellectual Property*, do inglês) Digital estejam inteirados sobre tais assuntos.

1.1 OBJETIVOS DO ESTÁGIO

1.1.1 OBJETIVOS GERAIS

Adquirir, desenvolver e aprimorar habilidades específicas de projetistas de circuitos integrados digitais através de treinamentos especializados e participação em projetos comerciais.

1.1.2 OBJETIVOS ESPECÍFICOS

- Desenvolver competências técnicas necessárias para executar o fluxo de projeto padrão da empresa no segmento de projeto de circuitos digitais
- Aprender as funcionalidades das ferramentas de projeto, verificação e documentação utilizadas pelo grupo de Digital IP Design
- Conhecer e participar do processo formal de especificação de produtos e sistemas
- Projetar, através de linguagem de descrição de hardware, os módulos digitais requisitados, assim como verificar sua funcionalidade junto à equipe de verificação de IP Digital

1.2 A EMPRESA

Neste documento será apresentada a história, assim como o desenvolvimento, do *Brazil Semiconductor Technology Center* (BSTC), ou Centro Brasileiro de Tecnologia em Semicondutores, onde as atividades do estágio foram realizadas. Por conseguinte, serão descritas as histórias de diversas empresas, entre elas a atual NXP Semiconductors N.V., que participaram na criação do BSTC.

O nascimento das atividades do que se tornaria o BSTC ocorreu em 1987, com o investimento da Motorola Inc. no potencial dos engenheiros brasileiros. A então Motorola

Inc., que mais tarde em 2011, iria se segmentar em duas empresas distintas, deu autonomia para a seção brasileira, possibilitando assim o surgimento do BSTC em 1997, com um total de 8 engenheiros na época. Localizado em Campinas, SP, devido à proximidade estratégica com as melhores universidades do país, o BSTC hoje conta com mais de 130 funcionários, sendo 95% destes engenheiros (NXP.COM, 2018).

No entanto, para que o BSTC construísse a atual reputação internacional supracitada, foram necessários anos de esforço e desempenho que resultaram em projetos bem sucedidos. Portanto, o investimento inicial feito na mão-de-obra brasileira surgiu do setor de semicondutores da então Motorola Inc., conhecido na época como *Motorola's Semiconductor Product Sector*, ou Motorola SPS, em 1997. Dentre os muitos setores da Motorola, o SPS era o que apresentava maior desempenho no mercado internacional na época, sendo um dos maiores responsáveis pelo sucesso comercial da empresa. Tendo isto em vista, aliado ao fato de que a Motorola pretendia instalar uma fábrica de celulares em Jaguariúna, SP, um dos seus maiores investimentos em território brasileiro, o SPS buscou profissionais brasileiros para compor seu corpo de funcionários.

Uma considerável parcela destes funcionários locais eram provenientes do Centro de Pesquisa e Desenvolvimento (CPqD), devido à sua história prévia em pesquisa em microeletrônica, e da *design house* Vértice Microeletrônica, que então se unificaram para criar um centro de desenvolvimento de produtos em semicondutores no Brasil, que pouco depois seria denominado como o BSTC.

Já em 1998, a equipe brasileira no BSTC já estava desenvolvendo projetos relevantes, como microcontroladores da Motorola para geração de caracteres utilizados em sistemas de *closed-captions*. Logo em seguida, em 1999, os microcontroladores de 8 bits da linha S08 já eram de responsabilidade dos engenheiros brasileiros, em conjunto com outras equipes internacionais, o que resultou em um grande reconhecimento da equipe e do centro nacional por parte do mercado internacional e dos principais diretores da empresa.

Ainda na mesma época, vários outros projetos eram tocados nacionalmente com uma metodologia diferenciada, baseada principalmente em síntese lógica, ao invés dos esquemáticos tradicionais, que levou a uma produtividade bastante satisfatória.

Em consequência disto, em 2000 aconteceu um grande investimento da Motorola em mão-de-obra brasileira, que promoveu um grande crescimento do corpo de funcionários do BSTC, em sua maior parte engenheiros. Uma grande quantidade de projetos de IPs foram alocados para o BSTC, em grande parte os IPs dedicados para controladores de motores à explosão, a exemplo da Unidade de Controle Eletrônico utilizada em carros de competição da McLaren, o que solidificou ainda mais a confiabilidade da competência dos engenheiros brasileiros. Estes IPs compuseram uma estrutura de microcontrolador com alto poder de processamento, uma das maiores da época, conhecida como Cobra, utilizada por exemplo no MPC5777M da NXP.

Ainda nesta linha, uma parcela considerável dos projetos da equipe eram destinados ao controle do *powertrain* de diversos automóveis, tendo destaque principalmente os controladores CAN (*Controller Area Network*) associados. Por exemplo, neste segmento de IPs para sistemas embarcados automotivos, um dos maiores projetos coordenados e implementados no BSTC é o módulo FlexCAN, utilizado atualmente em uma grande variedade de controladores CAN da NXP.

Em 2006, com o surgimento da Freescale Inc. a partir do SPS, a equipe brasileira transicionou suas atividades para a mesma, mantendo ainda seus projetos para sistemas embarcados automotivos, assim como projetos paralelos em sistemas analógicos e microcontroladores de 8 bits.

Em 2015, com a aquisição da Freescale pela NXP, o BSTC passou a atuar na linha de microcontroladores, que são utilizados em aplicações diversas, como dispositivos eletrônicos portáteis de monitoramento para praticantes de atividades físicas, existindo uma forte tendência para direcionar os projetos para sistemas IoT.

Atualmente, enfatiza-se o projeto voltado para sistemas com baixo consumo de energia, que representa uma propriedade de dispositivos utilizados em sistemas IoT, onde a disponibilidade de energia é limitada por ser proveniente, na maioria das vezes, de baterias. O projeto aqui apresentado está diretamente relacionado com tais especificações de baixo consumo, sendo voltado para arquiteturas com um controle *on-chip* de consumo de potência.

2 FLUXOGRAMA DE PROJETOS NA NXP SEMICONDUCTORES

Conforme apresentado anteriormente, o principal produto da NXP são os circuitos integrados, mais tecnicamente conhecidos como *System on Chip (SoC)*. Um SoC consiste em um conjunto de módulos digitais (i.e., IPs), como processadores (*cores*), Unidades Lógico-Aritmética (ULA), interfaces de memória, temporizadores, medidores de temperatura, conversores analógico-digital e digital-analógico, entre outros, integrados em silício. Estes módulos estão estruturados em um único chip e constituem um verdadeiro sistema embarcado completo, com blocos analógicos e digitais próprios, daí a sua denominação.

Devido à complexidade destes dispositivos, aliado ao grande número de equipes envolvidas em um projeto de um SoC, é imprescindível que a metodologia de projeto adotada seja clara, sólida e, principalmente, colaborativa, o que insere inúmeros desafios quanto à necessidade de comunicação e troca de resultados entre os times. Com isso posto, neste capítulo é apresentado de forma sucinta, porém didática, a metodologia/fluxograma seguida durante um projeto de circuito integrado, destacando as equipes envolvidas durante cada processo e o que é esperado ao final de cada etapa.

2.1 O PROCESSO DE ESPECIFICAÇÃO DE UM PRODUTO

Para que um produto, neste caso um SoC, saia de sua concepção inicial para sua concretização no mercado, são necessárias diversas etapas de especificação, comunicação, negociação e avaliação técnica, que se darão entre as equipes responsáveis representativas da NXP e a empresa cliente. Desta forma, podemos sintetizar estas etapas em um fluxograma tradicional, conforme apresentado na Figura 1, onde enfatiza-se a hierarquia entre as equipes desde a negociação com o cliente até a especificação do produto.

Uma das primeiras etapas que dará início a um projeto de circuito integrado, a ser utilizado comercialmente, é a especificação de um produto desejado por parte de um cliente. É o cliente ou o mercado que irão definir quais funcionalidades são desejadas para seu produto e quais serão suas aplicações, delimitando assim o ambiente em que o sistema estará imerso, que resultará nas devidas restrições. Esta descrição é feita em alto nível, sem particularidades técnicas, o que servirá como orientação básica para a especificação mais detalhada do produto e de sua arquitetura.

Durante esta etapa, existe uma constante interação entre a equipe de *Marketing* da NXP com o cliente, pois as necessidades dos clientes quanto à funcionalidade do produto são avaliadas no quesito viabilidade de execução e recursos disponíveis, para assegurar que

Figura 1 – Fluxograma simplificado para a criação de um produto.



Fonte: Autoria própria

o chip tenha características de consumo e operação realistas, como por exemplo tensão de alimentação e frequência. Além disto, estas reuniões são excelentes oportunidades para que a NXP possa divulgar suas inovações tecnológicas, oferecendo assim aditivos ao produto original do cliente, o que pode ser interpretado como uma atuação direta da NXP na inovação de suas empresas clientes.

Ao término das discussões sobre as funcionalidades do produto, a equipe de *Marketing* e Sistemas irá converter os requisitos do produto provenientes do cliente para descrições mais técnicas junto aos arquitetos de sistemas. São estes arquitetos de sistemas que materializam a proposta do cliente em uma arquitetura lógica, que detalha as funcionalidades do produto, como estas são implementadas com a tecnologia disponível, quais são as interfaces necessárias entre os módulos do circuito integrado, assim como será a comunicação deste com o ambiente externo.

O resultado deste trabalho dos arquitetos de sistema é um documento contendo a definição dos principais módulos (i.e., IPs) presentes no SoC, assim como sua integração com os demais módulos, destacando quais IPs terão de ser projetados e quais poderão ser reutilizados, o que caracteriza uma boa prática para compatibilidade entre tecnologias. Este documento é denominado como Documento de Definição de Arquitetura (ou em inglês, *ADD - Architecture Definition Document*).

Por fim, na base do fluxograma de projeto de um SoC, estão os engenheiros de projetos. Com o ADD, os engenheiros da NXP são capazes de traçar um planejamento relativo à implementação, verificação e documentação do SoC pretendido. É durante esta fase que as restrições técnicas, como máxima frequência de operação, são delineadas. As

metas quanto ao consumo de energia, área necessária e conseqüente número de portas lógicas são definidas também nesta fase. Estas especificações são orientadas pela tecnologia e processo de fabricação que serão utilizados, de acordo com seus parâmetros de processo, como por exemplo *leakage* de corrente, faixa de temperatura suportada, capacitâncias parasitas, entre outros.

O passo seguinte consiste na implementação do produto de fato, onde seu fluxograma próprio é apresentado e discutido na seção seguinte.

2.2 FLUXO DE PROJETO DE UM CIRCUITO INTEGRADO

Nesta seção descrevem-se os pontos principais de uma implementação de um circuito integrado, destacando quais ferramentas são utilizadas em cada parte e quais são os resultados esperados ao fim destas.

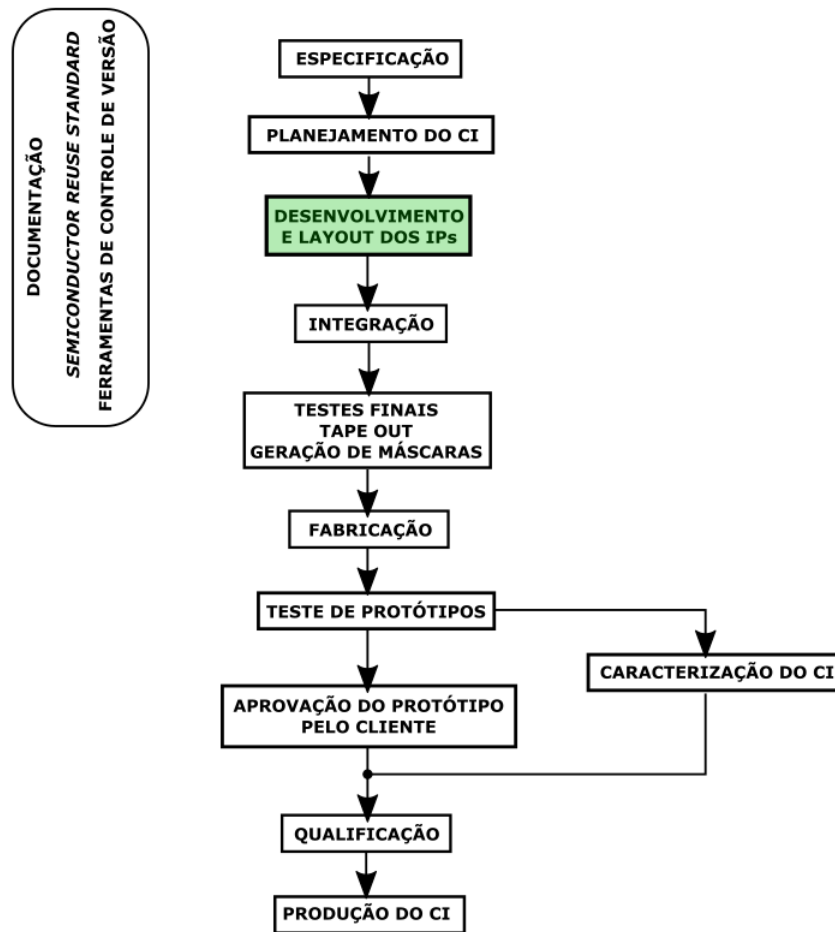
O ponto de partida para a implementação de um SoC é sua especificação. Esta, por sua vez, tem sua versão inicial oriunda do ADD previamente comentado, disponibilizado pelos arquitetos de sistemas. Esta documentação é constantemente atualizada ao longo do projeto do SoC, de acordo com as modificações que vão se fazendo necessárias com o tempo. Para que exista uma compatibilidade entre as contribuições dos diversos colaboradores de diferentes times, são necessárias camadas de protocolos e ferramentas relacionadas ao controle de versão destas contribuições.

O próprio ambiente de projeto em que estas atividades ocorrem é bastante estruturado e hierarquizado, contendo desde instruções e regras de como um diretório de trabalho deve ser organizado de acordo com sua funcionalidade (ou seja, como as pastas que contém determinado tipo de arquivo devem ser nomeadas e organizadas), a ambientes virtuais de controle de versão e sincronização, a exemplo das ferramentas de encapsulamento de comandos.

Com isso, existem padrões que determinam a metodologia que cada engenheiro deve se basear para realizar suas atividades, estabelecendo regras de como um código de descrição de hardware deve ser estruturado, comentado e implementado, para evitar problemas que já foram relatados por outras equipes previamente e através de estratégias padronizadas podem ser prevenidos.

Para o ambiente da NXP Semicondutores, existe o *SRS - Semiconductor Reuse Standard*, que padroniza todas estas atividades, com as devidas motivações. Esta metodologia, ao ser seguida, garante que os projetos realizados serão facilmente reutilizados por outras equipes, incluindo equipes remotamente localizadas, sem necessidade de suporte do autor. Estas etapas, assim como as ferramentas e metodologias envolvidas nas mesmas, estão ilustradas na Figura 2.

Figura 2 – Fluxograma para o projeto de um circuito integrado.



Fonte: Autoria própria

Já com uma especificação mais técnica e padronizada disponível, o planejamento do SoC é iniciado. Este planejamento tem por objetivo particionar o chip em diversos módulos, que serão projetados pelas equipes alocadas, caso estes já não existam previamente. Uma importante diferença desta estruturação de módulos com a definição de blocos no ADD é a existência de uma estratégia de *floorplanning*. Na especificação utilizada pelos engenheiros de projeto, um *floorplanning* inicial é realizado para orientação do time de integração.

Em seguida, as interfaces internas são definidas. Estas interfaces são as responsáveis por fazer a comunicação coerente entre módulos cujos protocolos de comunicação divergem. Por exemplo, para que um periférico se comunique com o processador serão necessárias algumas interfaces, ou *gaskets*, para fazer a conversão de protocolos, além de arbitrar as operações desejadas entre eles.

Por fim, as estratégias e arquitetura de teste, práticas estas conhecidas como *DFT* - *Design for Testability*, são delineadas pela equipe de DFT. Sendo assim, os registros, as cadeias de escaneamento (*scan chains*), os pinos e a lógica necessária para a execução de

testes são determinados.

Com todos os módulos definidos, assim como a interface entre eles e o *floorplanning* determinados, o passo seguinte consiste em implementar estes módulos. Tais módulos podem ser digitais, analógicos, ou ambos. As atividades do estágio, relatadas mais detalhadamente nas seções posteriores, se deram exclusivamente na implementação de módulos digitais. Todavia, devido à natureza dos projetos da empresa, a colaboração direta com outras equipes sempre se faz necessária.

Nesta etapa de desenvolvimento e *layout* de IPs, os engenheiros implementam e verificam cada módulo utilizando as ferramentas necessárias. Relativamente ao projeto digital, durante esta fase, a especificação dos módulos individuais é elaborada em paralelo com o projeto destes, pois assim as mudanças requisitadas são incorporadas tanto na documentação quanto na implementação. Tal metodologia reforça a necessidade de um sistema com controle de versão rígido, como comentado anteriormente.

Na seção seguinte, será explanada em mais detalhes esta etapa de implementação de módulos funcionais, os IPs.

Com todos os módulos, ou sua maior parte, projetados e verificados, pode-se prosseguir para a integração destes no SoC. Neste procedimento, os módulos são conectados com suas interfaces e posicionados, de uma maneira iterativa, de acordo com seu *floorplanning*. A equipe de integração é a responsável para esta atividade, embora exista uma comunicação constante com as equipes de projeto de IPs analógicos e digitais para obter informações sobre o que está sendo integrado.

Em paralelo à própria integração, similarmente ao que acontece durante o projeto de IPs, a verificação desta é efetuada pelo time de verificação de SoC. Este procedimento verifica se os módulos estão se comunicando corretamente e se suas funções básicas estão operando como esperando a nível de SoC.

Ao final da etapa de integração, os testes finais são efetuados. Estes testes consistem, em sua maioria, de rotinas de programas para aplicações típicas, garantido que o circuito integrado está funcionando corretamente, ainda em uma camada de simulação computacional.

Com todos os testes executados e os consequentes resultados positivos, o *tape out* pode acontecer, que consiste na finalização do projeto e entrega dos arquivos que serão utilizados para gerar as máscaras. A partir destas máscaras, a fabricação pode ser efetuada em silício por uma fábrica de semicondutores escolhida.

Os *waffers* resultantes da fabricação são então obtidos, que mais tarde irão ser encapsulados apropriadamente, provendo assim os primeiros protótipos. Estes protótipos retornarão para a NXP a fim de se executar a validação, onde serão executados testes já no circuito integrado físico. Em paralelo, o circuito é caracterizado quanto às suas faixas de

temperatura e umidade admissíveis, limites de tensão suportados, nível de ruído aceitável, entre outras propriedades físicas.

Com a validação concluída, as primeiras amostras são disponibilizadas para o cliente para a sua averiguação. Caso as amostras estejam satisfatórias, a produção comercial do CI pode ser efetuada após apropriada qualificação.

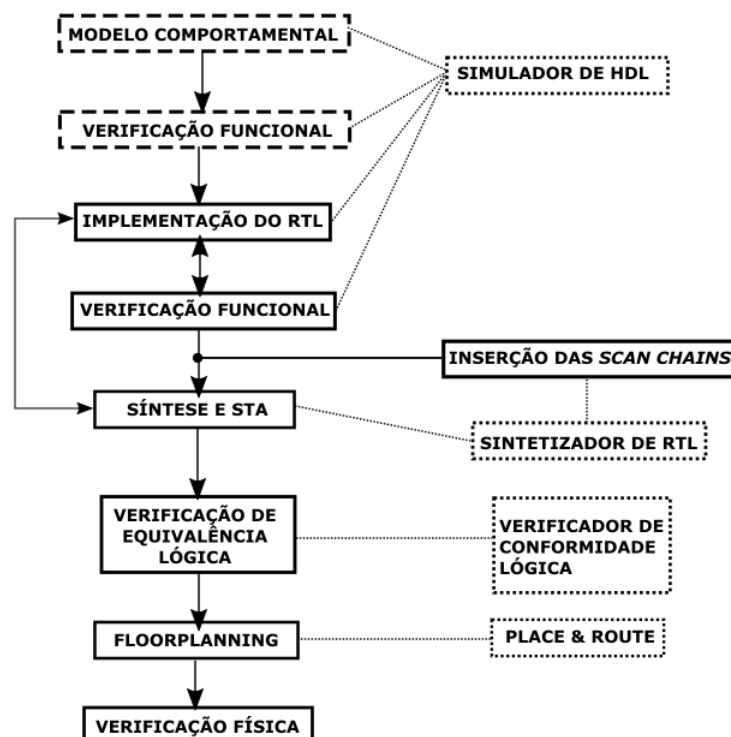
2.3 PROCESSO DE IMPLEMENTAÇÃO DE UM IP DIGITAL

As atividades do estágio aqui documentado ocorreram na equipe de design de IPs digitais, composta por 9 funcionários ao todo, com um gerente alocado, que é o mesmo supervisor deste estágio.

Nesta seção discutir-se-ão, em determinado nível de detalhe, as etapas necessárias para a confecção de um IP, incluindo a natureza das ferramentas utilizadas, sem especificar quais softwares de fato foram utilizados.

No fluxograma da Figura 3 sintetiza-se a estrutura da metodologia de projeto de um IP Digital, incluindo a funcionalidade das ferramentas utilizadas (em tracejado), conforme apresentado na Figura 3.

Figura 3 – Metodologia para projeto de um IP digital.



Fonte: Autoria própria

Ao iniciar o projeto de qualquer IP, deve-se ter bem claro qual é a funcionalidade

esperada, definindo de maneira concisa qual o comportamento desejado em uma variedade de situações, pois a implementação seguirá estas definições como referência. A estratégia básica para implementação de hardware é através da sua concepção em uma linguagem de descrição de hardware, como SystemVerilog, por exemplo.

Em alguns projetos mais complexos, é interessante implementar, ou seja, descrever em HDL, *modelos comportamentais* ao invés de iniciar com a implementação do código sintetizável, ou similarmente, a descrição em *Register Transfer Level* (RTL). Isso se dá pela necessidade de verificar que a funcionalidade planejada para o módulo realmente resulta nas operações esperadas, produzindo os resultados adequados em um sistema maior. Naturalmente, uma etapa de verificação funcional segue após término da descrição do modelo.

As etapas de verificação funcional, executadas por engenheiros da equipe de Verificação de IP, é fundamental para garantir que o módulo se comporta como esperado. Este procedimento representa uma grande parcela do esforço do projeto total, demandando uma parte considerável do tempo programado. Isso se deve ao fato de que um grande número de testes são realizados com o RTL do IP, submentendo-o a situações planejadas pelos verificadores, assim como por situações completamente aleatórias, mas que podem comprometer o circuito em atuação caso aconteçam devido a ruídos elétricos, por exemplo. Sendo assim, o módulo digital deve estar sempre preparado para retornar para um estado conhecido a partir de um estado inesperado. Portanto, o projeto de um IP envolve sérios *trade offs* para determinar qual o nível apropriado de segurança e redundância para aquele projeto.

Em seguida, tem-se a etapa de implementação do RTL, onde o código do IP é criado. Para o contexto deste relatório, utilizou-se primordialmente SystemVerilog para implementar os IPs requisitados. Durante esta fase, conforme discutido anteriormente, é necessário seguir regras de boas condutas para escritas de RTL, como deixar o código bem organizado e comentado, posicionando as declarações de variáveis, entradas e saídas nos lugares especificados, além de atentar para boas práticas de codificação, como comentar os ramos condicionais para posterior fácil identificação.

A verificação funcional do RTL é feita em paralelo com a própria implementação do mesmo, através de uma interação constante entre o responsável pelo RTL e o engenheiro de verificação. Assim que defeitos são detectados pelo verificador, o engenheiro de projetos é notificado via um sistema *online* de documentação de projetos, onde, ao final da correção das falhas detectadas, uma nova versão com apropriada identificação será disponibilizada através desta mesma plataforma. Esta metodologia permite o registro sólido e detalhado do histórico de modificações do código assim como suas causas.

Uma das principais ferramentas utilizadas neste estágio do projeto é o simulador de HDL. É esta ferramenta que compila o código produzido e verifica quaisquer erros de

sintaxe eventuais, além de simular seu funcionamento caso seja um modelo comportamental. Ferramentas adicionais permitem verificar as formas de ondas a partir dos estímulos gerados pelo *testbench*, fornecido pelo verificador, para averiguar o comportamento funcional do RTL através de suas formas de onda.

Ao término da verificação funcional, com todos os testes executados e confirmados, a síntese do RTL pode ser efetuada. Através de uma ferramenta de síntese, o código HDL será interpretado e irá gerar uma *netlist* que corresponderá ao mapeamento dos blocos constituintes do IP para a tecnologia utilizada. Durante este procedimento, técnicas de otimização são realizadas, a fim de reduzir algum aspecto do circuito. Pode-se configurar a ferramenta de síntese para priorizar a velocidade do circuito, o que poderá gerar um bloco com um número de portas lógicas maior. Do contrário, pode-se também determinar para a ferramenta que a área deve ser minimizada, o que pode resultar em um circuito mais lento.

Além disso, em paralelo, uma análise das restrições temporais é gerada, conhecida como *Static Timing Analysis* (STA). É durante este procedimento que possíveis condições de violação de *setup* são apresentadas. Caso as condições de STA não sejam satisfeitas, o RTL deve ser modificado para tratar os possíveis caminhos críticos associados, o que irá requerer uma nova verificação funcional para garantir o comportamento correto do circuito após as alterações.

Ainda assim, através de ferramentas e *scripts*, as *scan chains* são inseridas para complementar o DFT. Parte das ferramentas de síntese se encarregam dessa tarefa.

Com a finalização da síntese e a consequente geração da *netlist*, prossegue-se com a etapa de verificação de equivalência lógica. A *netlist* obtida pela síntese é comparada com o RTL que a gerou, para verificar, matematicamente, se estes são logicamente equivalentes.

Por fim, um *floorplanning* experimental pode ser efetuada para auxiliar o engenheiro de projetos a determinar qual a melhor estratégia para o IP. Através de modelos de interconexão obtidos pelo *floorplanning*, o IP pode ser reposicionado ou reprojetoado para diminuir o atraso de determinados caminhos críticos.

A última etapa consiste na verificação física do IP, com o *layout* do IP sendo verificado pela equipe de integração. Este trata-se do processo mais completo de verificação de um IP antes de sua fabricação em um SoC.

3 IMPLEMENTAÇÃO DO TEMPORIZADOR DIGITAL

Foi discutida na introdução deste relatório a tendência de se priorizar sistemas embarcados com baixo consumo de energia, principalmente em aplicações IoT, devido às suas restrições de fornecimento de energia e manutenção. Além disso, ao reduzir o consumo de energia, reduz-se, por conseguinte, a temperatura de operação do chip, que irá então aumentar sua durabilidade, confiabilidade e diminuir custos de refrigeração (TIWARI; SINGH; RAJGOPAL, 1998).

Nesta linha de projeto, surgem diversas estratégias para reduzir o consumo de energia, propostas tanto para o software como para o hardware. Uma das propostas mais inovadoras, discutidas por Esteve (2018), consiste em utilizar controladores de potência específicos para reduzir o consumo de energia, através de algoritmos de escalonamento de tensão e frequência. Ao controlar a tensão de operação do processador em conjunto com sua frequência, pode-se otimizar o consumo de energia geral do SoC.

Sabe-se que a potência dinâmica P_T pode ser expressa conforme Equação 3.1 (ESTEVE, 2018).

$$P_T = C_{pd}^2 V_{CC}^2 f_I N_{SW} \quad (3.1)$$

Onde C_{pd} representa a capacitância de dissipação de potência dinâmica, V_{CC} é a tensão de alimentação, f_I é a frequência do sinal de entrada, e N_{SW} é o número de bits transicionando de estado.

Sendo assim, ao reduzir a tensão de acordo com a frequência de operação, reduz-se o consumo de energia através de uma relação quadrática.

O temporizador descrito na seção seguinte atua em sistemas desta natureza, auxiliando o processador a obter medidas de corrente de acordo com a frequência de um sinal de entrada. É importante ressaltar que não serão apresentados detalhes quanto à implementação do temporizador, tampouco do sistema em que este está contido, por questões de confidencialidade. Todavia, tal decisão não prejudica o propósito deste trabalho, que consiste na descrição das atividades efetuadas na NXP Semicondutores.

3.1 DESCRIÇÃO DAS SUAS FUNCIONALIDADES

Uma das atividades que pode representar na prática as etapas de desenvolvimento e verificação de um IP, conforme seção anterior, foi o projeto de um temporizador digital para medição de parâmetros de processo de um subsistema. Este subsistema, por sua vez, é o responsável por controlar a tensão e a frequência do processador do SoC para otimizar o consumo de energia. Para tanto, são necessárias algumas informações, como a corrente de consumo do chip e o atraso do caminho crítico. A primeira, para ser obtida pelo temporizador, é convertida para uma relação de frequência conhecida proveniente de um módulo analógico. A segunda consiste em medir o período de um determinado sinal.

Estas atribuições de medição de tempo são comumente efetuadas por um temporizador digital. Neste trabalho, será apresentado um temporizador com as seguintes características:

- 4 canais de entrada para medição do sinal, em domínio assíncrono
- 1 canal para sinais de altíssima frequência
- 4 contadores de 16-bits independentes
- 1 *prescaler* de 16-bits para escalonamento da frequência de medição
- 5 modos de operação para medições de diferentes naturezas
- 4 saídas de interrupção independentes
- capacidade de concatenação de contadores para incrementar seu tamanho até 64 bits, mantendo a precisão de medição

Em suma, a função principal do temporizador consiste em realizar as seguintes medições e temporizações:

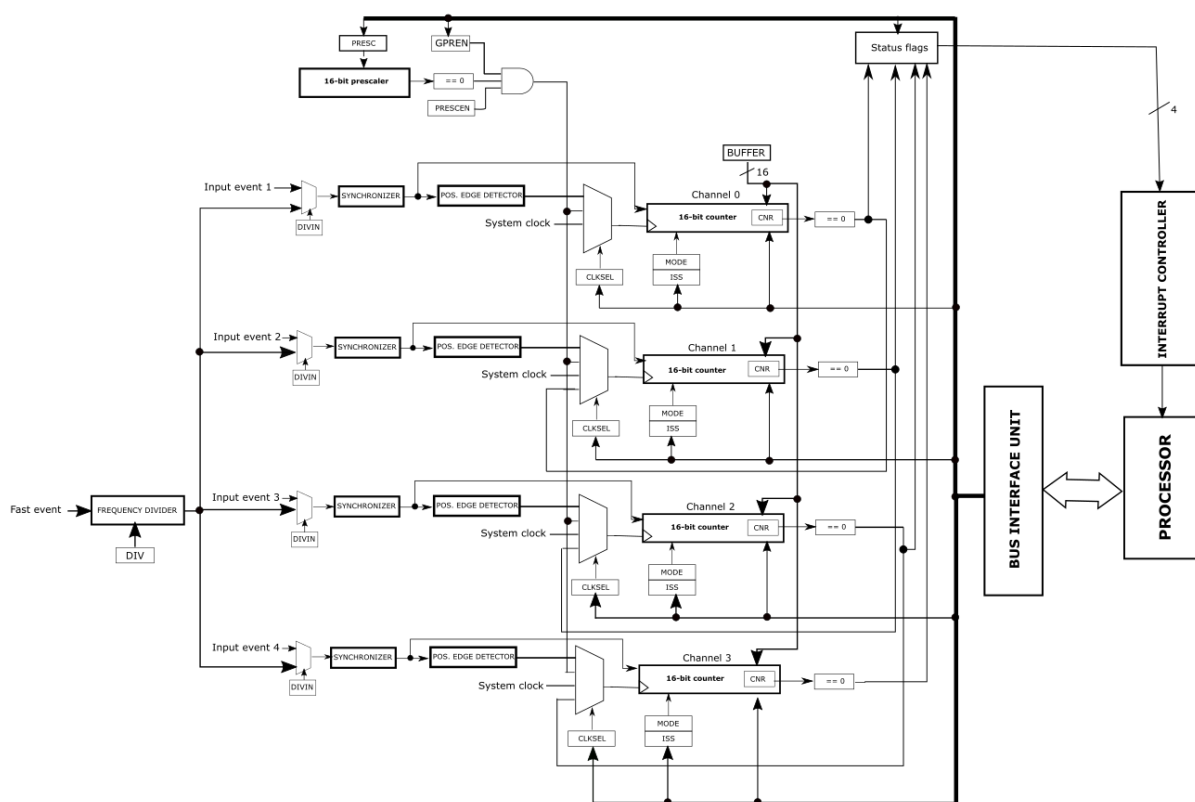
- Medir o período de um determinado pulso (tempo entre duas bordas de subida)
- Medir o *duty cycle* de um pulso (tempo entre uma borda de subida e a próxima borda de descida)
- Medir o complemento do *duty cycle* (tempo entre uma borda de descida e a próxima borda de subida)
- Medir o tempo entre duas bordas de descida consecutivas
- Inserir um intervalo preconfigurado para temporização
- Gerar interrupções periódicas com período programado

- Contar quantas bordas de subida ou descida ocorreram em determinado intervalo de tempo

Com essas funcionalidades, conseguimos mensurar a frequência de um sinal, seu ciclo de trabalho, sua potência consumida, além de gerar interrupções periódicas ou únicas para propósitos diversos.

O diagrama de blocos do temporizador que relaciona a interface entre os principais módulos é apresentado na Figura 4.

Figura 4 – Diagrama de blocos do temporizador digital implementado.



Fonte: Autoria própria

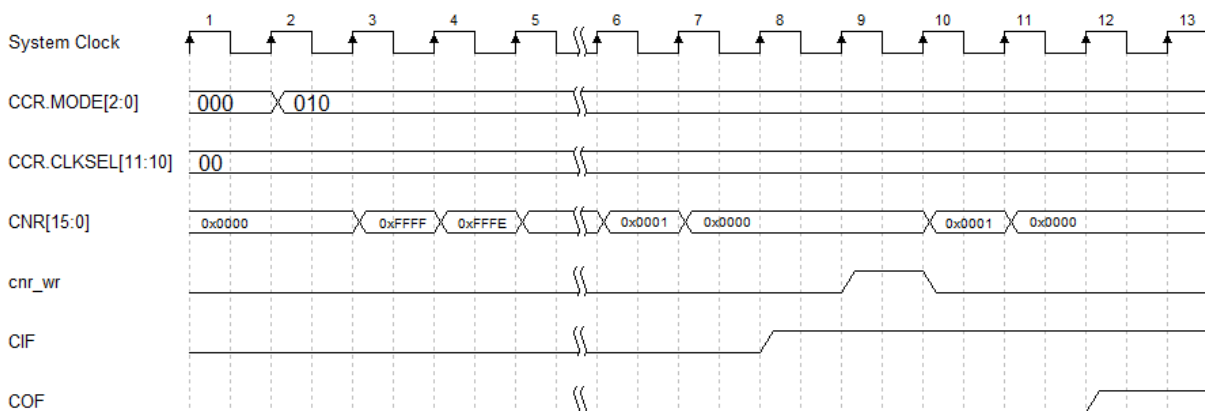
Na figura anterior pode-se ver a interface entre o temporizador digital e o sistema externo, assim como sua estrutura básica. A comunicação com o processador é feita via um barramento de dados cujo protocolo de comunicação estabelece como os dados podem ser escritos nos registros ou lidos destes. Os diversos registros do temporizador estão representados como blocos retangulares cuja sigla indica a *bitfield* correspondente ao mapa da memória. A interface entre o processador e os registros do temporizador, pelos quais este é configurado e sua operação é definida, é implementada pela *Bus Interface Unit*, que consiste em um módulo que converte os dados do barramento do processador para comandos de escrita/leitura para os registros internos do temporizador.

De acordo com o apresentado nos parágrafos anteriores, é possível notar que os 4 canais possuem estruturas idênticas, porém independentes, contendo seus contadores próprios e entradas externas apropriadas.

Cada canal pode receber uma entre um conjunto de 4 entradas de relógio possíveis, externas ou internas, além da entrada para sinais de alta frequência, que é única para todos os 4 canais. Esta entrada de relógio, que determinará a frequência com que o contador irá operar, sendo assim diretamente relacionada com sua precisão, é configurada pelo registro CLKSEL. Basicamente, o usuário pode configurar um determinado contador para operar na mesma frequência do relógio de sistema do temporizador ou operar em uma frequência inferior de acordo com o *prescaler*, que irá reduzir a frequência em um valor desejado. Desta maneira, pode-se efetuar medições de pulsos mais longos, porém com uma precisão inferior. Todavia, caso queira-se medir pulsos com longa duração, mas sem comprometer a precisão, é possível concatenar dois ou mais canais utilizando a saída do canal anterior como entrada de relógio para o canal posterior. Assim, quando o canal escravo entrar em uma condição de estouro, conforme indicado por sua *flag* de estouro, o próximo canal incrementa neste momento, realizando assim a concatenação desejada, dobrando o número de bits do contador, embora sejam necessários dois canais para isso.

Além da medição de pulsos, o usuário é capaz de fornecer informações de temporização para o processador, onde, após um determinado intervalo de tempo, o processador receberá uma interrupção. O diagrama de tempo fornecido na Figura 5 ilustra uma operação de temporização.

Figura 5 – Diagrama de tempo para uma operação de temporização.



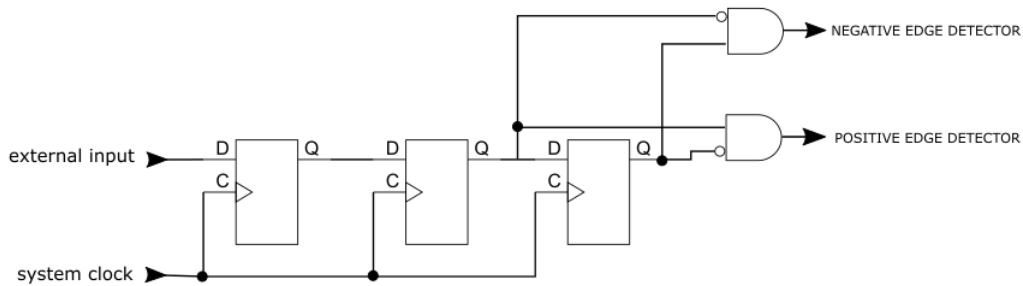
Fonte: Autoria própria

Na Figura 5, os sinais de CIF e COF são as *flags* de interrupção e estouro (*overflow*), respectivamente, o registro CNR armazena o valor do contador, enquanto que os registros CCR.MODE e CCR.CLKSEL determinam o modo de operação e a fonte de relógio, respectivamente. Já o sinal cnr_wr indica o momento em que uma escrita é realizada no

registro do contador.

Referenciando novamente para a Figura 4, ressaltamos a importância dos sincronizadores. Estes sincronizadores são responsáveis por sincronizar os sinais externos, nesse caso Input event 1 a 4 e Fast event, com o domínio do relógio do sistema. Isto é necessário pois os sinais externos são oriundos de módulos cujos sinais de relógio, ou seja, domínio de *clock*, são diferentes do *clock* do temporizador. Desta forma, como estes sinais externos são assíncronos em relação ao temporizador, podem ocorrer violações de *setup* e *hold*, daí a necessidade de sincronização. Uma estrutura simples para um sincronizador está apresentada na Figura 6. É interessante perceber que apenas dois *flip-flops* são necessários para a sincronização, sendo o terceiro utilizado para detecção de bordas de subida ou descida.

Figura 6 – Estrutura de um sincronizador.



Fonte: Autoria própria

Por fim, destaca-se o *Frequency Divider*, cuja atribuição é a de reduzir a frequência de sinais de entrada com alta frequência, também assíncrona do *clock* do sistema.

É importante destacar que o temporizador aqui implementado foi realizado de maneira parametrizável, isto é, o número de canais, a capacidade de divisão do divisor de frequência, assim como algumas funcionalidades específicas, são configuráveis na etapa de sintetização.

3.2 IMPLEMENTAÇÃO E RESULTADOS

Nesta seção os resultados obtidos são apresentados e discutidos, ou seja, quantidade de portas lógicas necessárias para implementar o temporizador digital, estimativa do consumo de potência, área correspondente e outras informações de temporização. Apresenta-se também um exemplo de simulação funcional, dentre as muitas necessárias, obtidas durante a etapa de verificação em colaboração com o verificador.

Na Figura 7 disponibiliza-se a visão geral do temporizador já mapeado para a tecnologia utilizada para a sintetização, que neste caso foi a FDSOI 28nm (*Fully Depleted*

Silicon On Insulation, com $L = 28$ nanômetros).

Figura 7 – Temporizador Digital sintetizado com a tecnologia FDSOI-28nm.



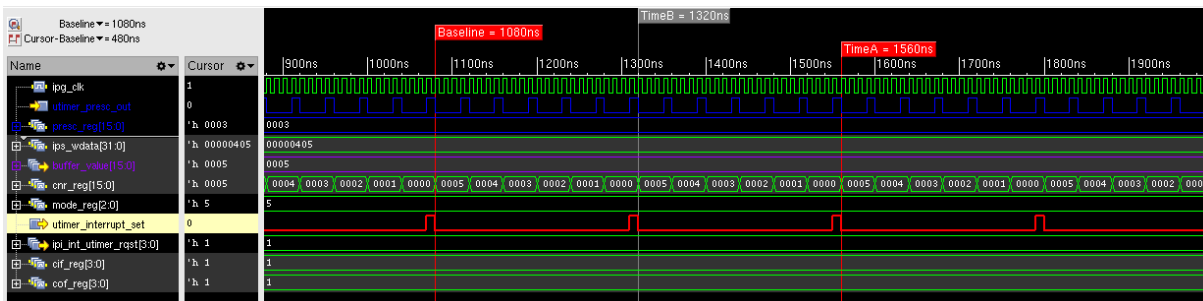
Fonte: Autoria própria

Este IP, na sua representação mapeada, reforça o conceito de que as interconexões e o roteamento destas em um módulo digital são parâmetros cruciais para um bom projeto de circuito integrado.

Através da análise das simulações funcionais efetuadas, pode-se demonstrar o comportamento da lógica do circuito diante de certos estímulos, o que deve ser verificado como coerente pelo verificador. Na Figura 8 encontram-se as formas de onda para uma

operação de interrupção periódica, um dos modos de operação do temporizador, que pode ter diversas aplicações conforme desejo do usuário.

Figura 8 – Simulação funcional para o modo de operação de interrupção periódica.



Fonte: Autoria própria

É através deste tipo de inspeção de forma de onda que realiza-se a verificação funcional do IP. A ferramenta de simulação de HDL permite identificar os valores dos sinais e variáveis do código em todos os pontos da simulação, assim pode-se determinar se a lógica, a descrição em HDL, está se comportando conforme esperado, caso contrário é possível também identificar o que está causando os erros e como consertá-los.

Com a efetuação da síntese, diversos procedimentos são efetuados, conforme descrito na seção 2.3. Primeiro, um conjunto de regras sobre padrões de escrita de RTL são verificadas. Este conjunto de regras tem por objetivo verificar se o código escrito é realmente sintetizável (e.g. não contém *initial blocks* no caso de SystemVerilog), se os sinais de relógio foram definidos corretamente, em conjunto com suas restrições de frequência, entre outras. É nesta etapa também que o esforço de otimização é configurado. Com um esforço muito alto, o tempo de execução da ferramenta pode ser bastante longo, podendo levar dias mesmo com computadores poderosos. Em contextos onde deseja-se apenas verificar resultados preliminares, sem necessariamente precisar de uma otimização máxima, este longo tempo de execução pode ser proibitivo devido aos prazos para entrega do projeto. Sendo assim, esforços de otimização mais baixos podem ser escolhidos para sínteses preliminares.

Após a síntese, diversos relatórios são gerados, entre os principais tem-se o relatório de área que determina qual a área ocupada por aquele módulo, incluindo o número de portas lógicas que o constituem. É gerado também o relatório de consumo de potência, onde uma estimativa estática, e portanto não muito precisa, das atividades das portas lógicas é efetuada e o consequente consumo de potência é calculado então.

Por fim, a cobertura de DFT é determinada. Já foi explanado que a inserção de *scan chains* ocorre durante a síntese e é efetuada pela ferramenta de síntese de acordo com a estratégia de teste desejada. Portanto, ao final da síntese, a própria ferramenta

verifica qual foi a cobertura alcançada por aquela estratégia, ou seja, qual a porcentagem de situações, que são as combinações de entradas e saídas, que podem ser verificadas pelas *scan chains*. Tudo isto é sintetizado em um relatório de DFT.

Na Tabela 1, apresentam-se, resumidamente, os resultados destes relatórios, destacando apenas as informações mais importantes.

Tabela 1 – Parâmetros físicos e tecnológicos do temporizador após sintetização.

Parâmetros Físicos do Temporizador					
Área					
Total	1405 μm^2		Portas Lógicas	4318	
Consumo de Potência					
Total	40,8 uW	Leakage	2,8 uW	Dinâmica	38 uW
Cobertura para Testes					
Teste	99,56 %	Falhas	98,42 %	ATPG	100 %
Frequência			92 MHz		

É possível constatar que o temporizador, devido às estratégias de otimização direcionadas para a redução de área, possui uma área de apenas $1405\mu m^2$, sendo esta uma área bem reduzida para parâmetros atuais. O número de portas lógicas é uma estimativa baseada na área de uma porta NAND, ou seja, divide-se a área total pela área da NAND gate de 2 portas para obter esse número. Sendo assim, 4318 portas representa um tamanho adequado para um temporizador com 5 modos de operação e 4 canais.

Outro parâmetro importante é o consumo de energia. Mesmo sendo apenas uma análise estatística, pois o consumo real irá depender da aplicação e sua demanda para o temporizador, obtiveram-se valores satisfatórios que não representam uma parcela significativa do consumo total esperado para o sistema completo. Estes resultados de área e consumo de energia foram obtidos para uma frequência de 92 MHz. Frequência mais baixas permitem estratégias mais agressivas de otimização e posterior redução de área e consumo.

Por último, tem-se os números de testabilidade. Nota-se que as coberturas estão todas acima de 98%, o que representa uma confiabilidade muito elevada. Destacamos os resultados para o *Automatic Test Pattern Generation* (ATPG), onde todas as combinações possíveis podem ser efetuadas de acordo com a estratégia de DFT adotada, que foram as *scan chains*.

Este números foram bastante satisfatórios para a equipe de projeto de IP Digital, o que representa um sucesso na execução das atividades de estágio, já com resultados práticos a serem utilizados comercialmente a nível mundial, por diversos tipos de mercado, destacando-se o mercado de *consumer electronics* e microcontroladores.

4 CONCLUSÕES

A natural divergência metodológica entre os projetos acadêmicos e projetos comerciais, oriunda de suas necessidades e contextos distintos, é especialmente perceptível durante as atividades de estágio supervisionado. É nesta oportunidade em que o estudante pode desenvolver suas competências profissionais, tanto no âmbito técnico como comportamental, constituindo assim uma transição de realidades desafiadora, porém demasiadamente enriquecedora. Durante o período inicial do estágio, tive que reavaliar minhas estratégias de abordagem de problemas, redescobrimo e aprimorando percepções de questões práticas requisitadas por aquele ambiente até então nunca vislumbrado.

A motivação e empenho alimentados pelo fato de que os resultados ali desenvolvidos iriam ser utilizados por consumidores em diversas partes no mundo, afetando assim suas qualidades de vida, criaram e lapidaram minha postura diante dos problemas e frustrações devido à falta de conhecimento técnico, naturais de profissionais ingressantes no mercado de trabalho. Aliado a isso, o constante apoio e suporte técnico provido pelos colegas de equipe, sendo esta constituída por engenheiros com décadas de experiência profissional e participação em projetos de importância global, nutriu um sentimento de colaboração que tornou insignificante a inexperiência e amadureceu minha visão sobre as competências de um engenheiro.

Para que o sucesso alcançado neste período de experiência profissional fosse possível, foram necessárias habilidades e conhecimentos desenvolvidos ao longo do curso de graduação inteiro. Sendo assim, não somente o conhecimento técnico sobre projeto de circuitos digitais, adquiridos nas disciplinas de Circuitos Lógicos, Arquitetura de Sistemas Digitais, VLSI System Design e Digital Design with VHDL using PLDs and FPGAs, durante a graduação e o intercâmbio nos Estados Unidos, mas como também a postura profissional e de trabalho em equipe, desenvolvida durante as atividades no Ramo Estudantil IEEE da UFPB, além do pensamento crítico reforçado nos projetos de iniciação científica, foram imprescindíveis para este feito e reconhecidos pelos supervisores de estágio.

Em contrapartida, são ricas as possibilidades de trabalhos futuros decorrentes das atividades aqui relatadas, projetadas para o longo prazo, onde outros módulos digitais podem ser projetados, assim como a análise das propriedades físicas do temporizador, resultantes do seu *layout*, poderia ser efetuada.

Em suma, conclui-se que a complementaridade do ambiente acadêmico com o ambiente profissional apresenta interessantes possibilidades de desenvolvimento para o estagiário, criando assim uma geração de profissionais flexíveis e inovadores.

Referências

- ANTONIO, R. A.; COSTA, R. M. de la. Implementation of dynamic voltage and frequency scaling on a processor for wireless sensing applications. *Proc. of the 2017 IEEE Region 10 Conference*, v. 1, p. 2955–2960, 2017. Citado na página 13.
- CHEN, G. Circuit design advance for wireless sensing applications. *Proceedings of the IEEE*, v. 1, n. 98, p. 1808–1827, 2010. Citado na página 13.
- ESTEVE, E. *New Power Management IP Solution Can Dramatically Increase SoC Energy Efficiency*. 2018. Website da Design & Reuse. Disponível em: <<https://www.design-reuse.com/articles/44834/new-power-management-ip-solution-can-dramatically-increase-soc-energy-efficiency.html>>. Acesso em: 09 out 2018. Citado na página 25.
- NXP.COM. *NXP no Brasil*. 2018. Website da NXP Semiconductors. Disponível em: <<https://www.nxp.com/about/about-nxp/about-nxp/worldwide-locations/nxp-no-brasil:BRAZIL>>. Acesso em: 01 out 2018. Citado na página 15.
- RABAEY, J. M.; CHANDRAKASAN, A.; NIKOLIC, B. *Digital Integrated Circuits: A Design Perspective*. 2. ed. Cambridge: Springer LLC, 2001. Citado na página 14.
- TIWARI, V.; SINGH, D.; RAJGOPAL, S. Reducing power in high performance microprocessors. *Proc. of the 35th Conference on Design Automation*, v. 1, 1998. Citado na página 25.