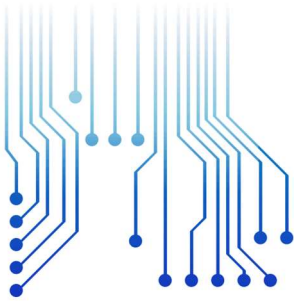


CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

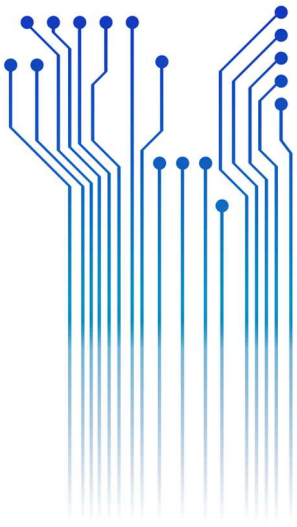
Universidade Federal
de Campina Grande

JOSÉ SAMUEL MENDES

Centro de Engenharia
Elétrica e Informática

RELATÓRIO DE ESTÁGIO
NXP SEMICONDUCTORS

Departamento de
Engenharia Elétrica



Campina Grande
2018

JOSÉ SAMUEL MENDES

NXP SEMICONDUCTORS

Relatório de Estágio Integrado submetido à Unidade Acadêmica de Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Microeletrônica

Orientador:
Professor Gutemberg Gonçalves dos Santos Junior, D. Sc.

Campina Grande
2018

JOSÉ SAMUEL MENDES

NXP SEMICONDUCTORS

Relatório de Estágio Integrado submetido à Unidade Acadêmica de Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Microeletrônica

Aprovado em / /

Professor Gutemberg Gonçalves dos Santos Junior, D. Sc.
Universidade Federal de Campina Grande
Avaliador, UFCG

Professor Marcos Ricardo Alcântara Morais, D. Sc.
Universidade Federal de Campina Grande
Orientador, UFCG

Dedico este trabalho aos meus avós, pela educação e toda experiência de vida compartilhada.

AGRADECIMENTOS

Aos meus pais, pelo apoio incondicional e por fazerem todos os esforços possíveis para realização deste sonho.

A toda a minha família, pelo suporte, incentivo e por se alegrarem comigo a cada vitória, em especial aos meus irmãos pelos anos de convivência em Campina Grande, durante a graduação.

A minha namorada Mariane, pelo companheirismo, afeto, paciência e ajuda durante a elaboração deste trabalho.

Aos Pseudomitos, grupo de estudo no qual participei durante toda a graduação, realizando várias reuniões para discutir resoluções de questões, tirar dúvidas e conversas sobre os diversos assuntos da universidade, principalmente nas vésperas de avaliações.

Aos amigos de outrora, aos que vieram só agora, mas principalmente aos que ficarão para sempre.

A todos que fazem parte do X-MEN Lab (PEM) que durante o fim da graduação se tornou minha segunda casa.

Aos professores Marcos Morais e Gutemberg Júnior, por todos os ensinamentos, orientação e oportunidades concedidas, permitindo que eu descobrisse áreas de interesse da minha formação e por acreditarem na microeletrônica no Brasil.

A todos os colegas da NXP, em especial ao meu tutor Guilherme Coraucci, pelo conhecimento e ensinamentos compartilhado durante o período de estágio.

“Don’t be sorry, be better”

Kratos-*God of War*

.

RESUMO

Neste trabalho estão descritas as atividades desenvolvidas durante o período de estágio realizado na equipe de Integração de SoC, na empresa NXP Semiconductors – BSTC (Brazil Semiconductor Technology Center), juntamente com o grupo de *front-end*. Ao longo do ano de 2018, foram realizados trabalhos na área de integração, análise de potenciais problemas com domínio de relógio diferentes, verificação de codificação de RTL e análise de redução de consumo. Além das atividades práticas, foram realizados treinamentos sobre assuntos relacionados a empresa, bem como, as metodologias utilizadas para realizar as atividades propostas.

Palavras-chave: NXP Semiconductors, Integração de SoC, Estágio.

ABSTRACT

In this work are described all activities developed during the period of internship at the NXP Semiconductors SoC integration team within the front-end group. Over the internship, in 2018, was possible to participate in the following activities: integration, Cross domain clock analysis, Lint analysis and power reduction analysis. Beside these activities, training about topics related to the company, as well as the methodologies used to do the proposed activities.

Keywords: NXP Semiconductors, SoC Integration, Internship.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

BSTC - Brazil Semiconductor Technological Center

CAD – Computer Aided Design

CDC - Clock Domain Cross

CPF/UPF - Common/Unify Power Format

CTS - Clock Tree Synthesis

DFT - Design for testability

ECO - Engineering Change Order

IoT- Internet of Things

IP - Intellectual Properties

LEC - Logical Equivalence Checker

ODC - Observability Don’t Care

QoS - Quality of Silicon

RTL - Register Transfer Level

SDC - Synopsys Design Constraints

SoC - System-on-a-Chip

STA - Static Timing Analysis

STC - Stability Condition

SUMÁRIO

1 INTRODUÇÃO

Neste relatório de estágio é descrito meu período como estagiário no setor de Integração de SoC (*System-on-a-Chip*) durante o período de 16/01/2018 a 17/12/2018 na empresa NXP Semiconductors. O estágio integrado é um componente curricular obrigatório, no curso de Engenharia Elétrica da Universidade Federal de Campina Grande – UFCG. E deve satisfazer uma carga horária mínima de 660 horas, seguindo os requisitos previstos na Resolução 01/2012 do Colegiado do Curso em consonância com a Lei do Estágio (Lei 11.788/2008).

1.1 OBJETIVO

Este trabalho tem como objetivo descrever as atividades realizadas durante o estágio na NXP, relacionando as experiências vividas no mercado de trabalho com os ensinamentos obtidos na Universidade. Será visto neste relatório uma descrição sobre a empresa, seu histórico e atuação no mercado, além das atividades realizadas durante o período supracitado que se concentraram na área de Integração de SoC.

2 EMPRESA

A NXP Semiconductors é uma empresa de semicondutores com sede nos Países Baixos, fundada originalmente em 1975 pela Philips, com nome Philips Semiconductors. Em agosto de 2006, a Philips anunciou a venda da sua divisão de semicondutores e a empresa passou a se chamar NXP Semiconductors. Atualmente, está presente em 33 países, possui 31.000 funcionários e cerca de 130 filiais pelo mundo (“About NXP ”). Em dezembro de 2015, a NXP comprou a multinacional de semicondutores americana Freescale Semiconductors.

Figura 1- NXP Semiconductors no mundo

Fonte: About NXP.

A principal missão da empresa é criar uma infraestrutura e conexões seguras para um mundo mais inteligente e criar soluções que tornem a vida das pessoas mais fáceis, melhores e seguras.

2.1 ATUAÇÃO NO MERCADO

A empresa tem várias linhas de produtos, dentre os quais pode-se destacar os chips e microcontroladores para as áreas automotiva, IoT, segurança, saúde e telefonia. As principais indústrias que desenvolvem e fabricam produtos nestes segmentos no Brasil são clientes da NXP. Ainda, por tratar-se de uma multinacional, detém clientes de grande renome, como Amazon, Apple, Bosch, Continental, Ericsson, Facebook, Gemalto, Giesecke & Devrient, Huawei, Hyundai, Kona, Nokia Networks, Panasonic, Samsung e ZTE.

2.2 BRAZIL SEMICONDUCTOR TECHNOLOGICAL CENTER

Figura 2- Sede da NXP no Brasil (BSTC)

Fonte: NXP no Brasil

O estágio foi realizado no centro de pesquisa da NXP localizado em Campinas – SP e conhecido como Brazil Semiconductor Technological Center – BSTC. A equipe de funcionários é dividida em subgrupos, entre eles, os times de verificação e integração de SoC, verificação e design de IP, validação e desenvolvimento analógico.

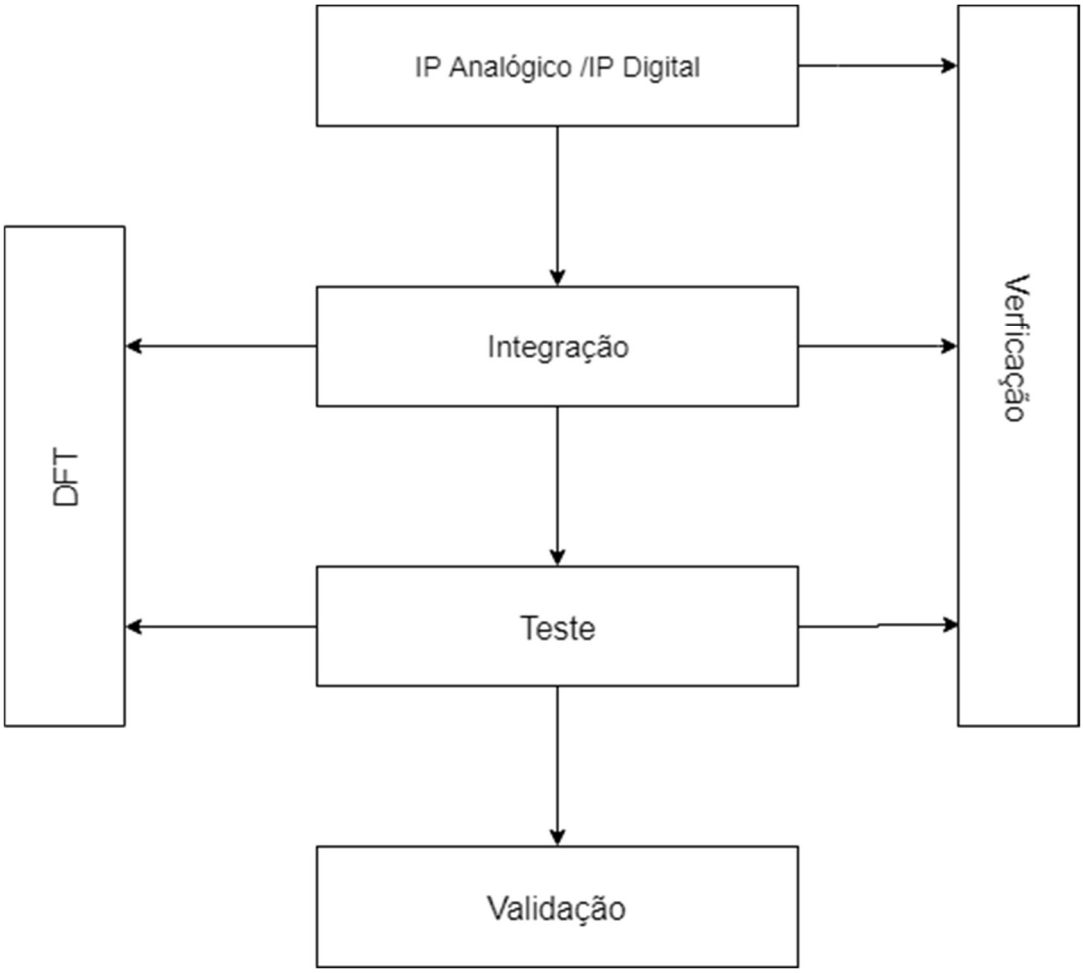
O BSTC iniciou as operações em 1997 quando ainda atuava sobre a direção da Motorola Inc. que apostou na capacidade intelectual e de inovação de engenheiros locais. Apesar de ter começado com apenas 8 engenheiros, o centro conta atualmente com 140 funcionários, dos quais 95% são engenheiros (“NXP no Brasil”).

Durante os seus 20 anos de história, o BSTC produziu mais de 100 projetos, incluindo microcontroladores e IPs em áreas como gerenciamento de potência, redes automotivas, processamento digital de sinais e aceleradores criptográficos e de temporização.

3 FLUXO DE PROJETO

O processo de criação de um chip é longo e complexo necessitando de uma grande quantidade de profissionais, cada um especializado numa diferente etapa do processo. Dentro deste, vários grupos contendo etapas acabam sendo classificados sob identificadores, que auxiliam os gerentes e líderes a identificar o estado atual do projeto e poder estimar as ações necessárias para otimizar o processo. Na Figura 3 abaixo pode-se verificar as etapas de um processo de criação de um chip.

Figura 3 – Fluxograma



Fonte: Autor (2018)

A primeira etapa no fluxo envolve a criação dos IPs, ou *Intellectual Properties*, componentes que possuem funcionalidades específicas desejáveis no chip. Esta etapa envolve a criação tanto de IPs digitais quanto analógicos, podendo existir ainda, os IPs que possuem ambos funcionando em sincronia.

Após discutida as especificações desejadas do chip é criada ou reutilizada a lista dos componentes específicos necessários. Em seguida, é possível prosseguir para a etapa de integração. Nesta, os IPs são conectados entre si e suas funcionalidades são aplicadas de forma que as especificações do SoC seja atendida. Dependendo dos IPs conectados, e da maneira que os são, é possível gerar funcionalidades mais complexas e resultar em um produto com características finais diferentes, como potência, consumo ou *leakage.* De maneira paralela a essa conexão, é necessário distribuir os componentes no espaço físico destinado ao chip, levando sempre em conta a relação entre área, potência e performance, buscando sempre manter estes parâmetros o mais otimizado possível. Tomando sempre cuidado para não prejudicar o QoS (*Quality of Silicon*).

É ao final da etapa de integração que o projeto sai do ambiente de simulações e passa para a etapa real, com produção em silício contendo o chip desenvolvido. A partir deste ponto, é possível fazer o controle de qualidade e realizar as análises que vão de fato caracterizar o comportamento do produto real, e não mais simulado.

Paralelamente a todo esse fluxo de desenvolvimento do produto, é necessário garantir que não haja erros ao longo do projeto. Quaisquer erros que possam se propagar sem controle estariam indesejadamente presentes no resultado final. Desta maneira, existem divisões responsáveis por verificar cada uma das etapas mencionadas anteriormente.

De maneira mais abstrata, é possível classificar essas divisões em dois grandes grupos: o de verificação de IPs e o de verificação de SoC. Tendo o foco de suas atenções em detectar erros nas funcionalidades e na integração dos IPs, respectivamente. Dessa maneira, o grupo de verificação de um projeto começa seu trabalho já no início do fluxo e continua trabalhando ativamente até o momento em que não houver mais nenhuma alteração física ou lógico no produto.

Já com o chip fisicamente disponível, a etapa de testes consiste em verificar que os padrões e rotinas, geradas na integração, retornem os resultados esperados e definidos em simulação quando exercitados. Esses testes são realizados através de máquinas especialmente desenvolvidas, chamadas de testadores. Elas possuem agulhas ou soquetes desenvolvidos especificamente para cada projeto, e são programadas para gerar valores

iguais aos padrões previamente definidos, e comparam com os resultados esperados, testando assim, a funcionalidade da parte lógica do chip e verificando se houve algum erro de fabricação no chip.

Por fim, ocorre a etapa de validação, onde diversas análises realizadas no chip confirmam suas características e especificações determinadas no início do projeto. Realizando ainda testes com determinadas condições de funcionamento que simulam o envelhecimento do chip, o que permite especificar como será o seu comportamento ao longo dos anos.

4 INTEGRAÇÃO DE SOC

A fim de explanar as atividades realizadas durante o estágio se faz necessário um conhecimento mais aprofundado das etapas de Integração de SoC.

Na NXP a equipe de Integração de SoC divide-se em três equipes: *front-end, back-end e DFT* onde cada uma dessas são compostas por subequipes que realizam diferentes atividades especificadas por seus respectivos gerentes.

4.1 *FRONT-END*

De uma forma ampla a equipe de front-end é responsável de integrar os diversos componentes do SoC. Alguns destes componentes são: IP digitais e analógicos, memórias, microprocessadores, periféricos e entre outros diversos componentes existentes no SoC. A equipe de *front-end*, é dividida nos seguintes grupos: Integração, Verificação de Equivalência Lógica entre RTL (*Register Transfer Level*) e base de dados estrutural, Verificação de Codificação de RTL, Síntese, e Análise de Potenciais Problemas com Domínio de Relógios Diferentes e Análise de Consumo (*Power Analysis*).

4.1.1 INTEGRAÇÃO

O grupo de Integração é responsável pela integração destes diversos componentes, através da formulação de RTLs, elaborados a partir de uma linguagem descritiva de

hardware, capazes de integrar todos os componentes que serão utilizados no chip, além de serem responsáveis pela criação do topo do projeto. Em alguns projetos se faz necessário o uso de ferramentas que possam automatizar este processo, visando mais agilidade e organização.

Devido ao enorme tamanho de um projeto de SoC é bem comum a reutilização de vários componentes. Porém em alguns casos, é preciso criar lógicas intermediárias capazes de adaptar os componentes para determinada especificação de projeto. Estas lógicas intermediárias são conhecidas como *glue logic* e também ficam sob responsabilidade do grupo de Integração.

4.1.2 VERIFICAÇÃO DE CODIFICAÇÃO DE RTL

Devido ao enorme tamanho de um projeto de SoC, é bastante comum ocorrer alguns erros de sintaxe e até mesmo erros de lógicas durante a integração do projeto. Portanto se faz, necessário verificar de uma forma rápida estes erros. Essa função é estabelecida ao grupo de verificação de codificação de RTL, onde as principais tarefas são averiguar possíveis erros de conexões entre componentes e erros de lógica, principalmente nas *glue logics*.

Além disso o grupo ajuda a otimizar o processo de síntese, visando adiantar lógicas não-sintetizável. Devido ao processo de sintetização ser bastante longo, antecipar este tipo de lógica, acarreta em menos tempo gasto durante o processo de sintetização.

4.1.3 SÍNTESE

A síntese é considerada o processo de conversão do RTL, uma linguagem de nível intermediário, para portas lógicas primitivas e registradores. Esta conversão resulta em uma base de dados estrutural, também chamada de *netlist*. Esta base de dados é usada posteriormente pela equipe de *back-end* para o processamento físico do projeto.

É função da equipe de síntese realizar esta conversão através de ferramentas **CAD**, bem como criar scripts visando otimizar o fluxo e verificar erros nos RTLs que proporcionam a não sintetização do projeto.

Além disso, é função da equipe de síntese escrever as *constraints* de tempo ou *timing constraints*, que são um conjunto de regras que limitam a solução de determinadas operações. Geralmente essas *constraints* são escritas no formato SDC (*Synopsys Design*

Constraints) e são de grande importância, não só para a síntese, mas para diversas outras fases do projeto. Entre algumas de suas funções são: definição de pinos de *clock*, bem como algumas de suas características, definição de caminhos falsos (*false paths*), definição de caminhos multi cíclicos (*multicycle paths*), caminhos síncronos e assíncronos, definição de modos de funcionamento do chip e entre outros conjuntos de regras.

O grupo também, neste momento, inclui estruturas lógicas de teste, juntamente com a equipe de DFT. Os domínios de *power* também são adicionados nesta etapa através dos arquivos de CPF/UPF (*Common/Unified Power Format).*

4.1.4 VERIFICAÇÃO DE EQUIVALÊNCIA LÓGICA ENTRE RTL E BASE DE DADOS ESTRUTURAL

Devido a capacidade das ferramentas de síntese de otimizar determinadas lógicas e da adição das já citadas *glue logics*, existe a possibilidade de a lógica existente nas *netlists* divergirem da lógica existente no RTL.

Além disso, durante determinada fase do projeto existe um momento conhecido como RTL *freeze,* onde a partir deste momento, a alteração em códigos de RTL não é permitida. Esta implementação ocorre devido ao tempo que se leva para rodar ferramentas de síntese ou até mesmo ferramentas utilizadas no *back-end.* Portanto alteração em lógicas a partir desta fase são feitas por ECO (*Engineering Change Order).* Estas alterações, no geral, são feitas de forma direta na *netlist.*

Portanto é função do grupo de LEC (*Logical Equivalence Checker*) verificar possíveis diferenças lógicas entre a *netlist* e os RTLs, bem como propor, soluções para corrigir esta divergência. Em determinada fase do projeto o LEC se é utilizado para comparar a síntese com a base de dados do *postlayout*.

4.1.5 ANÁLISE DE POTENCIAIS PROBLEMAS COM DOMÍNIO DE RELÓGIOS DIFERENTES

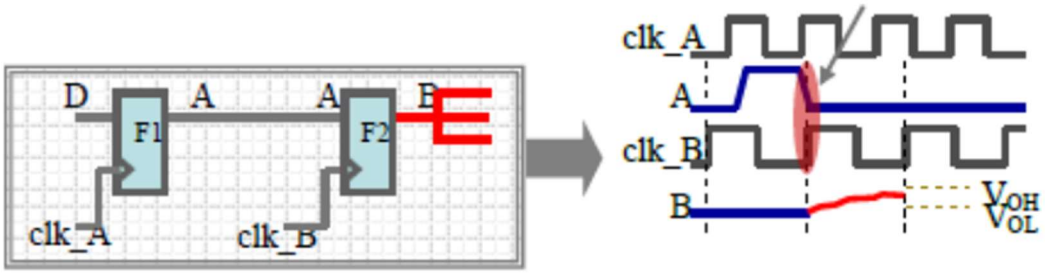
Como já citado anteriormente, devido ao enorme tamanho de um projeto de um SoC e também devido utilização/reutilização de vários IPs, se espera que alguns caminhos, entre interfaces de IPs, sejam assíncronos. Isto pode acarretar em problemas de metaestabilidade e *glitch*.

Metaestabilidade ocorre quando um dado está sendo propagado serialmente por um caminho assíncrono entre dois registradores. O problema ocorre quando o *clock* do segundo *flop,* comumente chamado de *destination flop*, ativa o processo de captura enquanto o primeiro, conhecido como *source flop*, está carregando o fio comum aos dois, fazendo com que a informação capturada esteja entre os valores de tensão característicos para 0 ou 1, sendo impossível de determinar o seu estado atual. Este sinal é chamado de sinal **metaestável**. O grande problema da metaestabilidade é que ela é um processo estocástico, podendo ocorrer em um determinando momento de funcionamento do *chip*. Logo não é possível encontrar tais erros através de simulações seja de RTL ou *gate level* e até mesmo em validação.

Pode-se perceber então que metaestabilidade ocorre apenas em caminhos assíncronos, pois em caminhos síncronos é possível estimar o tempo e assim ajustar este tempo evitando que ocorra metaestabilidade. Este tipo de análise é feito pelas equipes de *back-end* e *front-end* através de *Static Timing Analysis* (STA).

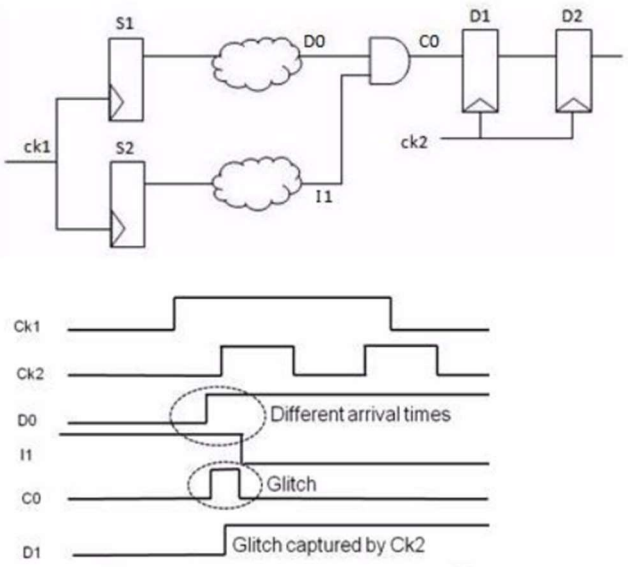
Existem vários modelos de *glitch,* tanto para sinais de *clock* quanto para de dados. Porém, o modelo de *glitch* que é verificado pela equipe de CDC (*clock domain cross*), ocorre quando em um determinado caminho, geralmente assíncrono, existe mais de um registrador emitindo o dado, nos quais, através de alguma lógica combinacional, converge em um único *flop*. Nas Figura 4 e Figura 5 abaixo pode-se ver, respectivamente, exemplos de metaestabilidade e *glitch.*

Figura 4 - Exemplo de Metaestabilidade



Fonte: CDC Rules Reference Guide (2018)

Figura 5 – Exemplo de *glitch*



Fonte: CDC *Rules Reference Guide* (2018)

Portanto é função da equipe de CDC verificar, através de ferramentas, violações de caminhos assíncronos sem sincronizadores e violações de *glitch,* bem como propor soluções para resolver estes tipos de violações. Além disso é comum, a equipe de CDC realizar ajustes nos arquivos de *timing constraints* (SDC), adaptando-os para verificações de CDC, tendo em vista que, no geral, as *timings constraints* são elaboradas visando apenas as ferramentas de *timing*, na qual são utilizadas para encontrar violação de caminhos síncronos (*setup* e *hold*).

4.1.6 *POWER*

O grupo de *power* tem como objetivo procurar soluções que possam reduzir o consumo de energia no *chip* através de medidas que possam ser implementadas a um nível intermediário de abstração.

Seja na formulação dos arquivos de CPF e UPF (*Common Power Format* e *Unify Power Format*) ou na implementação de lógicas que visam reduzir o consumo de energia do chip, seja essa estática ou dinâmica, este processo é conhecido como *Power Reduce.* E também fazendo-se uma estimativa inicial do consumo de potência do *chip*, conhecido como *Power Estimate.*

4.2 BACKEND

A equipe de *backend*, tem como função desenvolver o design físico do SoC, bem como verificar as grandezas físicas do mesmo. A equipe de *backend* é subdividida nas seguintes equipes: *Floorplan*, *Placement*, *Clock tree synthesis* e *Routing*.

4.2.1 FLOORPLAN

Durante esta fase a equipe precisa definir a área que o circuito irá ocupar. Esta define-se levando em conta as especificações do chip, área de atuação, quantidade e complexidade de circuitos lógicos existentes e entre outras características.

4.2.2 PLACEMENT

Cabe a equipe de placement definir a posição de cada componente do SoC. O *placement* é feito utilizando a *netlist*, gerada pela síntese. Além das características citadas anteriormente a equipe leva em conta as interfaces mais próximas com os sinais de entrada e saída do chip (periféricos).

4.2.3 CLOCK TREE SYNTHESIS (CTS)

O objetivo da equipe de CTS é implementar as árvores de relógio para o devido funcionamento dos circuitos, levando sempre em conta, as grandezas físicas, tais como atraso, potência e entre outras, que possam influenciar de forma direta ou indireta os caminhos realizados pelos sinais de *clock.*

4.2.4 ROUTING

Nesta etapa, é realizada as conexões físicas entre as células. Logo a equipe de *Routing* fica a cabo de realizar estas conexões levado em consideração alguns fatores, tais como: largura dos fios, regras de área mínima, espaçamento entre fios, integridade de sinal e entre outros.

4.2.5 DFT

A equipe de DFT (*Design for testability*) tem como função a elaboração de circuitos lógicos, com intuito de detectar possíveis erros de fabricação. As tarefas da equipe de DFT podem ser divididas em: inserção de cadeia de *scans* no RTL, bem como elaboração de *constraints* únicas para teste, síntese do modo de teste do *chip*, e por fim geração de padrões que possam estimular o SoC e detectar possíveis erros de fabricação.

5 ATIVIDADES REALIZADAS

O estágio foi realizado na equipe de Integração de SoC, em conjunto com a equipe de *front-end*. Trabalhando juntamente com as equipes encarregadas de integração, onde utilizou-se a ferramenta de integração conhecida com *Rabbit*®; *Lint*; CDC e *Power Reduction*, tendo em vista que ambos processos utilizam a mesma ferramenta CAD, conhecida como Spyglass®;

Além dessas atividades práticas, foram oferecidos alguns treinamentos no início do estágio, principalmente sobre ferramentas da própria empresa, embora tenham sido ofertados também cursos para maior familiarização com as linguagens básicas para a realização das atividades práticas.

5.1 TREINAMENTOS

No início do processo, foi necessário um estudo aprofundado e treinamento com os funcionários mais experientes, a fim de que todos os estagiários ficassem a par de diferentes ferramentas, sistemas e protocolos utilizados na empresa, para que todos conhecessem as diferentes áreas e etapas do processo de desenvolvimento ao qual estariam diretamente envolvidos. Inicialmente, os treinamentos foram focados nos conceitos de qualidade e metodologia aplicados na empresa, mas ainda tiveram abordados diversos outros, focados nos fluxos de projeto, sistemas de segurança, utilização e funcionamento de um sistema de controle de versão específico da empresa, linguagens descritivas e desenvolvimento de algoritmos.

5.2 ATIVIDADES PRÁTICAS

Nesta etapa, grande parte das atividades realizadas durante o período de estágio foram com a equipe de *front-end* nas etapas de Integração, Lint, *Timing Constraints,* CDC e *Power Reduction,* realizando as mais diversas atividades.

Além disto, na resolução de problemas ou até mesmo para sanar dúvidas, teve-se o contato com ferramentas de rastreamento de problemas como o *Design* PDM®. O contato com suporte de terceiros que disponibilizam as ferramentas utilizadas dentro da empresa, também se fez necessário, sempre buscando o melhor conhecimento destas ferramentas ou propondo melhorias e correções, com intuito sempre de otimizar o fluxo do projeto existente na empresa.

Estas atividades serão explanadas de forma mais detalhada nos capítulos a seguir.

5.2.1 INTEGRAÇÃO

Nesta fase do projeto, o estagiário teve o primeiro contato com ferramentas utilizadas para automatizar o processo de integração, conhecida como *Rabbit*®, na qual se baseia na construção de tabelas onde se instancia as entradas e saídas dos IPs que vão se conectar e em seguida a ferramenta gera arquivos em HDL, neste caso em Verilog, que representam estas conexões. Estas tabelas eram divididas por partições afim de dividir e organizar melhor as equipes que trabalham no processo. Em seguida conectava-se as interfaces das partições, gerando assim, o topo do projeto.

5.2.2 LINT

Nesta etapa, inicia-se logo quando os componentes do SoC são conectados e podendo assim dizer, que já tem-se um topo sendo instanciando. Assim durante esse processo, o estagiário precisou reunir uma lista de todos os arquivos bem como os diretórios de todos os IPs digitais. Além disso, foi necessário também reunir alguns arquivos no formato *liberty*, no qual descreviam as características de tempo, dos IPs analógicos, memórias e *macrocells*. A criação desta lista foi realizada com ajuda de *scripts*, elaborados pelo próprio estagiário.

Em seguida, com esta lista pronta, a ferramenta lê estes arquivos, compila e em seguida elabora. Nesta fase de compilação o estagiário deparou-se com problemas de

compilação em alguns arquivos VHDL e *liberty*, ao perceber que estes problemas ocorriam unicamente na ferramenta, fez-se necessário alterações locais nos arquivos, visando não interferir o fluxo do projeto. Estas alterações ficaram conhecidas como *hacks*. Após estes *hacks* terem sido realizados, viu-se que o SoC foi compilado e elaborado. Portanto, entrou-se em contato com o suporte da ferramenta, para explicar os problemas encontrados e buscar alguma solução.

Em seguida verificou-se possíveis erros de Lint. Dentre estes, os principais foram: entradas de circuitos não instanciadas, erros no tamanho dos sinais, geração de *latches*, pinos de *clocks* recebendo valores constantes, *loops* combinacionais e entre outros.

5.2.3 CDC

Neste processo, o estagiário inicialmente precisou rodar um *setup*, pois para iniciar a verificação era necessário que as *constraints* estivessem consolidadas. Nesta fase inicial do CDC era necessário fazer alguns ajustes para que a ferramenta reconhecesse as *constraints* em formato SDC, alterando-se alguns parâmetros da ferramenta. Além disso precisou-se criar algumas *constraints* com intuito de evitar caminhos de testes, tendo em vista que o CDC é verificado apenas para o modo funcional do chip.

Em seguida verificou-se possíveis caminhos assíncronos, sem sincronizadores, entre interfaces do SoC, onde as violações eram divididas em dois tipos: violações para um único sinal e violações em vetores. Neste processo era bastante comum encontrar violações dentro de alguns IPs, porém existiam protocolos de sincronização, mas a ferramenta não reconhecia estes tipos de violações. Isto era esperado, tendo em vista que a mesma faz uma análise apenas estrutural. Nestes tipos de violações fazia-se necessário entrar em contato com o responsável por determinado IP, explicando o problema relatado pela ferramenta, e assim aguardava-se o retorno do responsável comunicando se existia ou não uma violação.

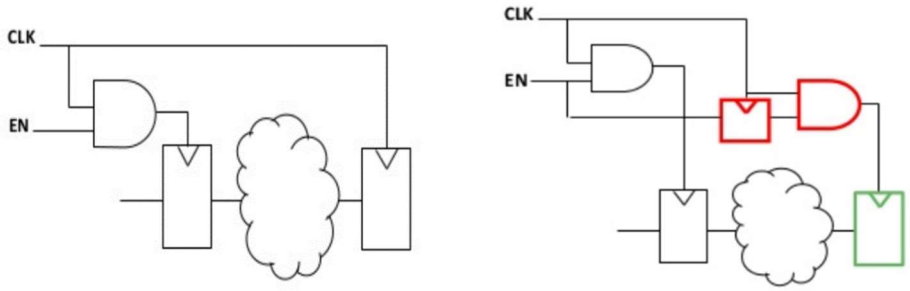
5.2.4 *POWER REDUCTION*

Ainda utilizando a ferramenta *Spyglass*®, nesta fase, buscou-se formas de implementar lógicas que visava reduzir o consumo de energia do SoC. No geral essas lógicas se baseiam na implementação de *clock gates*.

As implementações destas condições se baseavam em duas técnicas para realizar a implementação de um *clock gate*, nas quais são STC (*Stability Condition*) e ODC (*Observability Don't Care)*.

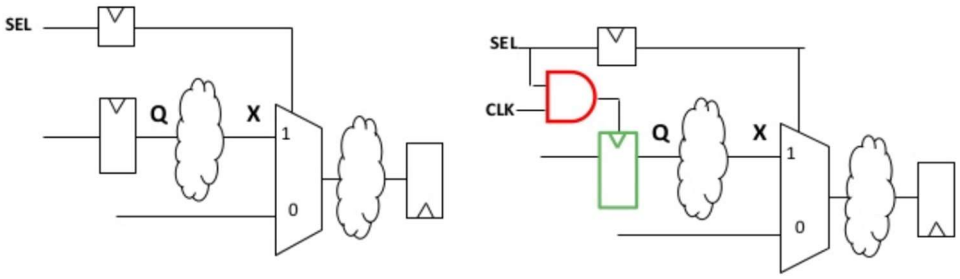
STC ocorre quando a entrada de um registrador não irá mudar, ou seja, ela está uma condição estável, por uma determinada quantidade de tempo. ODC se é implementado quando uma determinada saída de um registrador não importa. Nas Figura 6 e Figura 7 abaixo podemos ver exemplos de aplicações de ODC e STC.

Figura 6 – Exemplo de aplicação de *clock gate* com STC



Fonte: *Low Power VSLI Design* (2015)

Figura 7- Exemplo de aplicação de *clock gate* com ODC



Fonte: *Low Power VSLI Design* (2015).

Portanto nesta fase, se fazia necessário fazer uma análise de todo SoC e levando em conta onde era viável a implementação destes *clock gate*, levando em conta alguns parâmetros, tais como: quantidade de registradores que seriam afetados pelo *clock gate*, complexidade de implementação do mesmo, ou seja, quanto mais simples a

implementação, mais viável realizá-la. Por último, verificava-se o quão interna era essa alteração, no caso, dando prioridade sempre para lógicas entre interfaces das partições do projeto, lógicas internas de IPs. Devido ao risco de alterar alguma lógica e a necessidade de entrar em contato com o responsável pelo o IP, estava em último na prioridade.

6 CONCLUSÃO

A experiência de realizar um estágio integrado em uma empresa de relevância internacional como a NXP Semiconductors foi decerto construtiva. Foi possível solidificar alguns conhecimentos adquiridos durante a graduação, principalmente na área de microeletrônica, e adquirir novos, sempre com o auxílio dos engenheiros da NXP.

Esta envolveu constante pesquisa e estudo para a solução dos mais diversos problemas. Sendo sempre necessário a comunicação com colegas de diferentes setores para melhor entendimento de como sanar os problemas. Dessa maneira, a imersão no ambiente de trabalho se deu de maneira completa, com prazos a serem cumpridos e máximo desempenho objetivado.

Por fim, durante toda a extensão dessas atividades, foi necessário o emprego de várias habilidades comportamentais, denominadas *soft skills*. Essas habilidades envolvem capacidades interpessoais, e seu emprego é altamente desejado no ambiente de trabalho. Comportamentos de interesse geralmente envolvem proatividade, criatividade, gentileza, interesse e empenho ao se realizar atividades.

O êxito obtido na execução das tarefas propostas resulta dos conhecimentos adquiridos durante o curso de Engenharia Elétrica da UFCG e da constante disposição dos engenheiros da NXP na elucidação das dúvidas que surgiram durante esse período de estágio.

É relevante salientar a importância da experiência de estágio na formação de um engenheiro. Uma das raras oportunidades de lidar com problemas reais e procurar soluções eficientes. A experiência é uma das formas mais eficientes de aquisição de conhecimento. Sendo assim, é importante ressaltar as barreiras criadas pela UFCG/PRE durante esse período. Portanto, devido ao papel crucial dessa experiência na finalização da formação de um engenheiro, algumas práticas impostas pela UFCG deveriam ser revistas. Especialmente o limite de duração de estágio, atualmente de 6 meses. Limite esse que não está previsto em regimento interno ou em nenhuma lei que aborde o assunto, principalmente na lei Nº 11.788 de 25 de setembro de 2008, que regulariza a atividade de estágio.

REFERÊNCIAS

MANO, M. M. *Digital Design*. 3. ed. [S.l.]: Pearson.

Synopsys (2018). *CDC Rules Reference Guide.*

Synopsys (2018). *SpyGlass® Power Estimation and Exploration Rule Reference Guide.*.

Mahesh Dananjaya. (2015). *Low Power VSLI Design*. Acesso em 04 dez de 2018, disponível em Slide Share: <https://www.slideshare.net/MaheshDananjaya/low-power-vlsi-design-46842791>

About NXP. Disponível em: <https://www.nxp.com/about/about-nxp/aboutnxp:ABOUT-NXP> . Acesso em: 1 dez. 2018

NXP no Brasil. Disponível em: <https://www.nxp.com/about/about-nxp/about-nxp/worldwide-locations/nxp-no-brasil:BRAZIL> . Acesso em: 1 fev. 2018.