



Universidade Federal de Campina Grande  
Centro de Engenharia Elétrica e Informática  
Departamento de Engenharia Elétrica

Lucas Lacerda Paixão

## Relatório de estágio

Verificação de IP Digital - Freescale Semicondutores

Campina Grande, Paraíba  
Fevereiro, 2015

Lucas Lacerda Paixão

## Relatório de estágio

Verificação de IP Digital - Freescale Semicondutores

Relatório de Estágio Integrado submetido ao Departamento de Engenharia Elétrica da Universidade Federal de Campina Grande para obtenção do título do grau de Bacharel em Ciências no domínio da Engenharia Elétrica.

Orientador: Prof. Dr. Marcos Ricardo Alcântara Morais

Convidado: Prof. Dr. Antônio Marcus Nogueira Lima

Campina Grande, Paraíba  
Fevereiro, 2015

Lucas Lacerda Paixão

## Relatório de estágio

### Verificação de IP Digital - Freescale Semicondutores

Relatório de Estágio Integrado submetido ao Departamento de Engenharia Elétrica da Universidade Federal de Campina Grande para obtenção do título do grau de Bacharel em Ciências no domínio da Engenharia Elétrica.

---

Prof. Dr. Marcos Ricardo Alcântara Morais  
Orientador

---

Prof. Dr. Antônio Marcus Nogueira Lima  
Convidado

Campina Grande, Paraíba  
Fevereiro, 2015

# Resumo

Este relatório consiste em uma descrição das atividades de estágio realizadas no *Brazil Semiconductor Technological Center - BSTC*, centro de pesquisa, desenvolvimento e inovação da Freescale Semicondutores em Campinas - SP. Durante aproximadamente 8 meses, tarefas no segmento da microeletrônica (projeto de circuito integrado) foram desempenhadas, envolvendo o treinamento em linguagens e em procedimentos de verificação de IP digital, a elaboração de modelos de circuitos digitais, a implementação de testes funcionais e *testbench*, de modo geral. Todas relacionadas a eTPU, um co-processador capaz de realizar tarefas complexas de temporização e gerenciamento de Entrada/Saída, bastante utilizado na indústria automotiva para controle de motor.

**Palavras-chave:** Freescale, BSTC, verificação de IP digital, eTPU.

# Abstract

This work consists in a description of the internship activities held in the Brazil Semiconductor Technological Center – BSTC, a research, development and innovation center of Freescale Semiconductor at Campinas - SP. During approximately eight months, tasks in the microelectronics field (integrated circuit design) were done, involving training in languages and procedures of digital IP verification, elaboration of digital circuit models, implementation of functional tests and testbench in general. All of them were related with the eTPU effort, a co-processor able to perform complex timing tasks and input/output management, heavily used by the automotive industry in power train applications.

**Keywords:** Freescale, BSTC, digital IP verification, eTPU.

# Lista de figuras

Figura 1: presença da Freescale no mundo. Fonte: (Freescale, 2015).....	9
Figura 2: estrutura geral da eTPU. Fonte: (Freescale, 2004). .....	13
Figura 3: estrutura da engine da eTPU. Fonte: (Freescale, 2004).....	14

# Lista de siglas

<b>BSTC</b>	<i>Brazil Semiconductor Technological Center</i>
<b>CI</b>	Circuito integrado
<b>DUV</b>	<i>Device Under Verification</i> (dispositivo sob verificação)
<b>EAC</b>	<i>eTPU Angle Counter</i>
<b>eTPU</b>	<i>Enhanced Time Processing Unit</i>
<b>IP</b>	<i>Intellectual Property</i> (propriedade intelectual)
<b>PWM</b>	<i>Pulse Width Modulation</i> (modulação por largura de pulso)
<b>RTL</b>	<i>Register Transfer Level</i> (nível de transferência entre registros)
<b>SoC</b>	<i>System-on-chip</i> (sistema em chip)
<b>UFCG</b>	Universidade Federal de Campina Grande
<b>ULA</b>	Unidade Lógica e Aritmética

# Sumário

<b>1. Introdução</b> .....	<b>8</b>
<b>2. Apresentação da Freescale</b> .....	<b>9</b>
<b>3. Segmento de trabalho</b> .....	<b>11</b>
3.1. Área de atuação .....	11
3.2. eTPU2.....	12
3.2.1. Canal.....	13
3.2.2. EAC.....	14
3.3. eTPU3.....	15
<b>4. Atividades desenvolvidas</b> .....	<b>16</b>
4.1. Verificação funcional.....	16
4.1.1. Elaboração e manutenção de testes funcionais.....	16
4.1.2. Elaboração de modelo de referência.....	17
4.1.3. Cobertura funcional .....	18
4.1.4. Assertions .....	18
4.2. Verificação formal .....	19
4.3. Treinamento em linguagens e procedimentos de verificação .....	19
<b>5. Conclusão</b> .....	<b>20</b>
<b>6. Referências bibliográficas</b> .....	<b>21</b>



# 1 INTRODUÇÃO

Neste trabalho, são descritas as atividades desempenhadas como estagiário em tempo integral (40 horas semanais) de 05/05/2014 a 31/12/2014 na empresa Freescale Semicondutores LTDA. O estágio supervisionado ou integrado é um componente curricular obrigatório, no curso de Engenharia Elétrica da Universidade Federal de Campina Grande - UFCG. O segundo deve perfazer uma carga horária mínima de 600 horas, seguindo os requisitos previstos na Resolução 01/2012 do Colegiado do Curso em consonância com a Lei do Estágio (Lei 11.788/2008).

Inicialmente, na seção 2, é feita uma apresentação da Freescale enquanto empresa multinacional, no segmento de microeletrônica, com foco no seu centro de pesquisa, desenvolvimento e inovação no Brasil, em Campinas – SP, o *Brazil Semiconductor Technological Center - BSTC*. Na seção 3, discorre-se sobre o segmento de trabalho do estágio contemplando a sua área de atuação e os projetos envolvidos. Na seção 4, explicitam-se as atividades executadas. Por último, na seção 5, é dado destaque às considerações finais.

## 2 Apresentação da Freescale

A Freescale desenvolve soluções embarcadas para diversos segmentos da indústria. Seu portfólio inclui microcontroladores, microprocessadores, sensores, circuitos analógicos, entre outros. Com sede em Austin, Texas, nos Estados Unidos, a empresa dispõe de centros de pesquisa, desenvolvimento e inovação em mais de 20 países, totalizando aproximadamente dezoito mil funcionários. Foi fundada com esse nome em 2004 como *spin-off*, após o desmembramento do setor de semicondutores do restante da Motorola.



Figura 1: presença da Freescale no mundo. Fonte: (Freescale, 2015).

A Freescale Brasil iniciou suas operações em 1967, quando ainda atuava sobre a direção da Motorola, que apostou na capacidade intelectual e de inovação dos engenheiros locais. O *Brazil Semiconductor Technological Center - BSTC* iniciou suas operações com 8 experientes engenheiros e, ao final do ano 2000, esse número já ultrapassava 60. Atualmente o BSTC conta com mais de 150 funcionários (95% engenheiros) contabilizando um número superior a 180 projetos entregues. Desde 2007, o *design center* do Brasil passou a ter importância crucial na empresa adquirindo o status de *Premier Design Center* e passou a operar em uma sede própria localizada no condomínio Technopark em Campinas – São Paulo, uma

região estratégica pela sua localização, pela disponibilidade de mão de obra altamente especializada, além da proximidade das melhores universidades do país.

A área comercial da empresa também tem presença significativa na América do Sul, tendo o Brasil como base de operações. Apresentando uma estrutura enxuta e altamente competente, a equipe comercial conta com gerentes de vendas e engenheiros de aplicação que juntos oferecem aos seus clientes soluções completas de suporte e acompanhamento, iniciando no conceito do projeto, passando pelo desenvolvimento propriamente dito e culminando com o fornecimento dos dispositivos semicondutores via canais de distribuição. Empresas no Brasil como Magnetti Marelli, Visteon, Whirlpool, Motorola, Electrolux, Continental, Kostal, Delphi, entre outras, são clientes da Freescale Semicondutores.

## 3 Segmento de trabalho

### 3.1 Área de atuação

O fluxo completo de projeto de um circuito integrado envolve diversas etapas, simplificada: especificação, *design*, verificação, síntese, *layout*, fabricação, teste e validação. Na fase de *design*, geralmente as funcionalidades especificadas são modeladas em uma linguagem de descrição de hardware, como Verilog, resultando em um módulo de propriedade intelectual (*Intellectual Property core – IP core*), codificado a nível de transferência entre registros (em inglês, *Register-Transfer Level - RTL*) e geralmente conectado a diversos outros IP's compondo um sistema em *chip* ou SoC (*System on Chip*). O IP segue para as fases posteriores até se tornar um componente eletrônico no final da cadeia de desenvolvimento. A verificação, especificamente, tem um papel importantíssimo no fluxo de projeto, uma vez que é nessa etapa que se assegura que o RTL segue fielmente a especificação.

Na indústria de semicondutores, a concepção de um circuito integrado é uma tarefa complexa, e as falhas oriundas desse processo têm efeitos catastróficos. Depois de fabricado um lote de *chips*, muitas vezes com milhões de unidades, é inviável a retificação de cada circuito integrado - CI. Daí a necessidade de se investir grandes recursos e tempo na verificação desses componentes de maneira a produzi-los com a menor taxa de erros possível.

O estágio foi realizado no departamento de Verificação de IP Digital, que é responsável por desenvolver técnicas que viabilizem a identificação de falhas em IP's para em seguida reportá-las ao time de design, que deve proceder com as correções. Essas técnicas compreendem, por exemplo: verificação funcional, verificação formal, cobertura funcional, cobertura de código.

A verificação funcional, como o próprio nome sugere, concentra-se nas funcionalidades do IP. Para isso, implementa-se um ambiente de teste em *software*, denominado de *testbench*, que na maioria das vezes é composto por um conjunto de partes responsáveis por estimular o *Device Under Verification - DUV* e observar o

seu comportamento em relação a um modelo de referência. A verificação formal baseia-se na prova matemática da conformidade de um algoritmo a uma certa especificação formal ou propriedade, usando métodos formais. As técnicas de cobertura são importantes para mensurar a qualidade dos estímulos e revelar partes ou funcionalidades do IP não verificadas.

Um dos projetos em desenvolvimento no BSTC chama-se *Enhanced Time Processing Unit - eTPU*, um temporizador sofisticado bastante utilizado em microcontroladores automotivos para aplicações de controle de motor. Todas as atividades do estágio foram relacionadas a esse IP e contemplaram treinamento em diversas ferramentas computacionais, revisão de especificação, manutenção e codificação de estímulos e ambiente de teste, implementação de modelos de referência, planejamento de verificação funcional e formal.

## 3.2 eTPU2

A eTPU2 é um co-processador, em sua segunda versão, com capacidade de processamento independente da CPU, ou seja, é capaz de realizar tarefas complexas de temporização e gerenciamento de Entrada/Saída sem intervenção do *host*. Para isso, ela dispõe de memória e *core* próprios. Pode ser utilizada em aplicações que exigem baixa latência e elevada precisão como em controle de motor e comunicação serial. A estrutura geral da eTPU pode ser vista na Figura 2.

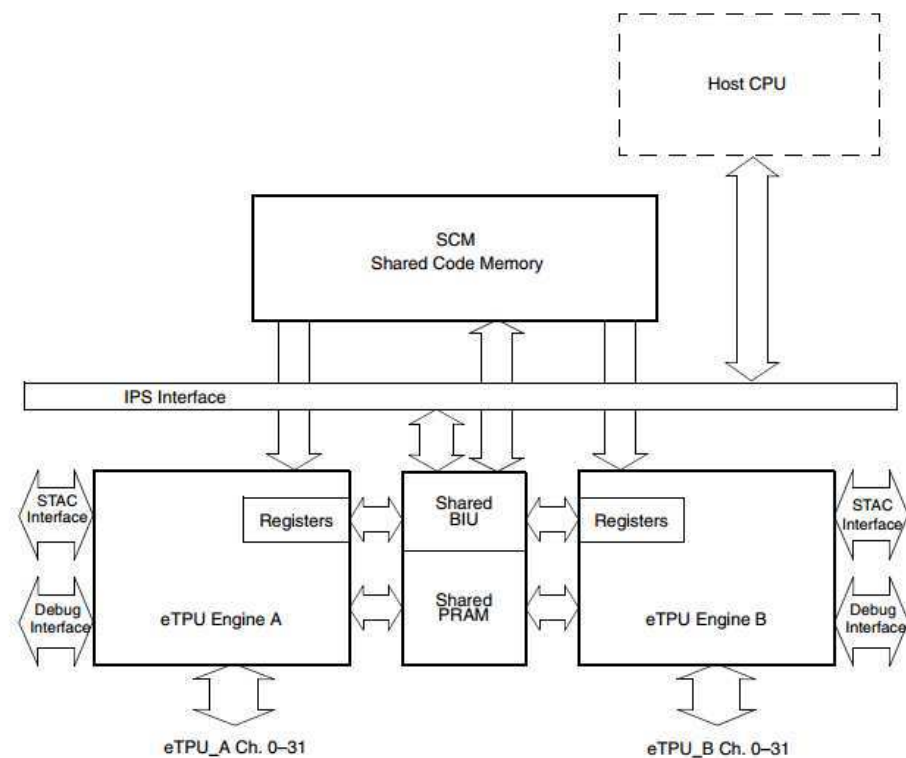


Figura 2: estrutura geral da eTPU. Fonte: (Freescale, 2004).

Cada *engine* da eTPU com seu microprocessador é responsável por processar as transições de sinal na entrada e gerar transições na saída de acordo com bases de tempo. As instruções a serem executadas são armazenadas na memória de código (*Shared Code Memory*). Dados e parâmetros de aplicação são armazenados na memória de dados (*Shared PRAM*). A CPU acessa os registros e essa memória da através de um barramento denominado de *Shared BIU*. Ainda existem 32 canais de entrada/saída, associados a cada *engine*, e que têm acesso a duas bases de tempo de 24 bits. Finalmente, um contador de ângulo, que essencialmente implementa um PLL digital, completa as funcionalidades da eTPU.

### 3.2.1 Canal

Associados aos 32 canais por *engine*, podem ocorrer eventos de *match* e/ou transição. As duas bases de tempo, TCR1 e TCR2, podem ser comparadas (igual e maior-ou-igual) a valores armazenados em registro e gerar eventos de *match*. Ademais, de acordo com uma transição pré-selecionada do sinal de entrada do

canal, um evento pode ser disparado resultando no armazenamento do valor das bases de tempo em registros de captura.

Os canais podem ser programados para responder automaticamente a variadas combinações de até dois eventos de transição e dois *matches*. Respostas podem incluir capturas dos contadores TCR1 e TCR2, ações sobre o sinal de saída e a geração de sinais de interrupção para a *microengine* da eTPU, conforme ilustrado a seguir.

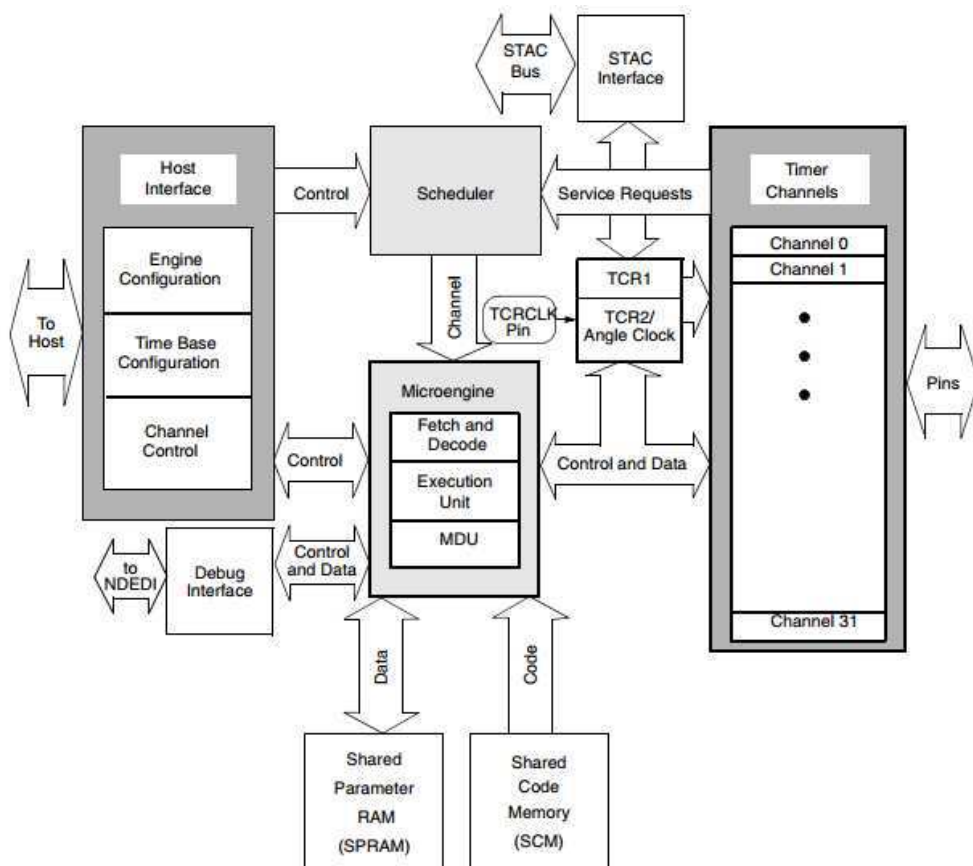


Figura 3: estrutura da engine da eTPU. Fonte: (Freescale, 2004).

### 3.2.2 EAC

O *eTPU Angle Counter* - EAC implementa uma lógica de medição estimada de ângulo (fase) de uma roda dentada. A taxa de incremento do contador é atualizada por software, enquanto um hardware dedicado implementa a lógica para incremento do contador com período, em média, fracionário, e para automaticamente acelerar ou parar a contagem adequadamente em casos de aceleração e desaceleração (ou

seja, quando os dentes são detectados antes ou depois do esperado). Diversos outros recursos são disponibilizados, como tratamento automático de dentes faltando.

### 3.3 eTPU3

A versão 3 da eTPU foi inicialmente especificada para implementar todas as funcionalidades da eTPU2 (mantendo a compatibilidade com as soluções já desenvolvidas pelos clientes) e mais alguns recursos novos, com melhoramentos na latência, na eficiência de código e nas funcionalidades de canal, por exemplo:

- Agendamento de tarefas com suporte a preempção envolvendo troca automática de contexto entre *threads* em execução;
- Novas instruções e operações de Unidade Lógica e Aritmética - ULA;
- Suporte a bases de tempo de 32 bits;
- Canais com capacidade de geração autônoma de PWM e recursos voltados para controle de motor com acesso a uma terceira base de tempo.



## 4 Atividades desenvolvidas

Os erros lógicos nos dispositivos são causados pelas discrepâncias ocorridas entre o comportamento pretendido e o comportamento observado, que podem ocorrer devido à especificação ambígua, interpretação errada da documentação ou devido a alguma inconsistência inserida durante o desenvolvimento do DUV. As falhas são captadas através da verificação, que pode ser realizada de forma estática (formal), dinâmica (funcional) ou híbrida (semi-formal). As atividades do estágio se concentraram nas duas primeiras formas, elucidadas a seguir.

### 4.1 Verificação funcional

A verificação funcional utiliza simulação para verificar o DUV. Para que isso seja possível, é necessário que haja um ambiente de verificação que possa receber o DUV, inserir estímulos e comparar suas respostas com as de um modelo de referência. Esse ambiente é denominado de *testbench*, que além do dispositivo a ser verificado e do modelo, é composto também por *drivers*, monitores, *response checkers*, entre outros.

#### 4.1.1 Elaboração e manutenção de testes funcionais

Os estímulos na eTPU2 se dividem em direcionados e randômicos. Os primeiros foram desenvolvidos para estimular partes específicas do DUV a partir da codificação, em linguagem C, de uma estratégia baseada em casos e situações bem definidas. Os últimos, por serem aleatórios, têm a capacidade de exercitar diversas funcionalidades diferentes, podem ser manipulados para contemplar áreas críticas e atingir uma maior cobertura. São normalmente ajustados através de tentativa e erro, trocando a semente de geração. Alguns dos testes fazem checagens por si só, outros dependem da presença de um *response checker*, que compara as saídas do DUV com as do modelo.

Durante o estágio, diversas regressões (um conjunto de estímulos) da eTPU2 foram simuladas, e certas falhas investigadas por inspeção de forma de onda e análise de código fonte. Para cada caso, apresentou-se uma solução relacionada a questões de projeto como interrupção, carregamento de contexto, manipulação de memória, configuração de canais, instruções, enfim. Os erros não estavam necessariamente no DUV, mas também em estímulos ou em outras partes do *testbench*.

### 4.1.2 Elaboração de modelo de referência

A eTPU2 é um IP já considerado verificado; inclusive ele segue compondo diversos SoC's em comercialização. As tarefas desempenhadas no *testbench* dele são apenas melhoramentos. Em contrapartida, a nova versão desse projeto, a eTPU3, envolveu novos procedimentos de verificação e foi necessário, por exemplo, a implementação de um novo modelo de referência em SystemVerilog, e não em C++, como na versão anterior.

No estágio, uma das tarefas foi codificar o modelo do canal da eTPU3 garantindo a plena compatibilidade com a versão 2 e implementando as novas funcionalidades como geração autônoma de PWM, sincronização de grupo de canais e outros recursos voltados para controle de motor baseado em uma especificação definida por um conjunto de engenheiros de hardware, de sistemas e de verificação. Para em seguida, proceder com a integração desse modelo com o restante do *testbench* da eTPU3.

O modelo foi implementado baseado em um conceito de modelamento eficiente, em que o objetivo básico é economizar na geração de eventos, minimizando o número de subrotinas chamadas ao longo de uma simulação. Isso é alcançado por duas vias essencialmente: a codificação, no modelo, não só de seu estado presente, mas também de seu estado futuro e uma representação não-convencional de sistemas síncronos, que evita a geração de eventos quando o sistema está estacionário. Esse conceito já havia sido implementado para o canal da eTPU2 a partir de uma biblioteca desenvolvida em *SystemC*. Durante o estágio, a tarefa foi

portar diversas partes dela para SystemVerilog, incluindo o modelo completo de canal da eTPU2, e implementar as suas novas funcionalidades.

### 4.1.3 Cobertura funcional

Em verificação, uma questão fundamental é a medida de quanto o DUV e suas funcionalidades foram exercitadas. Essa métrica pode revelar buracos de cobertura, ou seja, partes do design não testadas e, portanto, não verificadas, que podem comprometer o correto funcionamento do dispositivo. Nesse contexto, a cobertura funcional é uma técnica usada para medir o progresso da simulação e reportar quais funcionalidades deixaram de ser estimuladas. Os estímulos devem ser bons o suficiente para cobrir essas regiões e casos críticos (*corner cases*).

Foi tarefa do estágio pensar em um modelo de cobertura para as novas funcionalidades trazidas pela eTPU3 contemplando os novos recursos de canal, preempção e principalmente agendamento de tarefas. Baseado na especificação do IP, em seus atributos interessantes e propensos a erro, tendo em vista todas as restrições, foi feito um planejamento que servirá como base para a implementação do código de cobertura propriamente dito.

### 4.1.4 Assertions

Para verificar uma característica do DUV, primeiro a sua funcionalidade precisa ser entendida, depois ela precisa ser explicitamente descrita e finalmente as condições sob as quais o comportamento é aplicado precisam ser estabelecidas. Essa descrição pode ser implementada em uma *assertion*.

O contador de ângulo EAC é uma parte complexa da eTPU susceptível a inúmeros *corner cases*. Com o objetivo de melhorar a verificação funcional desse bloco, propôs-se como atividade de estágio um conjunto de SystemVerilog *Assertions (SVA)*, implementadas no monitor do EAC, seguindo cada detalhe da especificação. Comportamentos inusitados foram revelados e documentados. A

idéia é que, durante as simulações, caso alguma das assertivas falhe, uma mensagem de erro seja acusada e dispare um procedimento de investigação.

## 4.2 Verificação formal

Na classe de mecanismos de verificação estática, também chamada de verificação formal, de acordo com a definição de Bergeron (2003) tem-se: verificação de modelos, verificação de equivalência e prova de teoremas. A verificação de modelos demonstra que propriedades definidas pelo usuário nunca são violadas para todas as possíveis sequências de entradas. A verificação de equivalência, por sua vez, compara dois modelos para determinar se eles são logicamente equivalentes ou não. Finalmente, a prova de teoremas demonstra que o teorema está provado ou não pode ser provado.

Em se tratando da eTPU3, empregou-se a verificação de modelos. Fazer o planejamento da verificação formal do canal, da preempção e principalmente do agendamento de tarefas também foi tarefa realizada no estágio.

## 4.3 Treinamento em linguagens e procedimentos de verificação

Durante o período de trabalho na Freescale, um conjunto de cursos foi oferecido contemplando diversas ferramentas computacionais largamente utilizadas pela indústria de semicondutores, assim como fluxo de projeto, *Shell Script*, Linux, arquitetura de computadores, SystemVerilog, técnicas de modelamento, além de outros procedimentos de verificação empregados na empresa em diversos projetos.

## 5 Conclusão

A experiência do estágio em verificação de IP digital foi enriquecedora do ponto de vista pessoal e profissional. A oportunidade de aplicar a teoria aprendida durante a graduação em projetos reais de larga escala contribuiu para uma boa formação em Engenharia Elétrica. Conceitos técnicos de circuitos digitais, arquitetura de computadores, engenharia de software, programação e documentação foram exaustivamente exercitados, além dos fatores comportamentais como capacidade de trabalhar em equipe, comunicação eficaz, iniciativa e pró-atividade. Muitos conhecimentos novos também foram adquiridos com o *expertise* dos engenheiros da Freescale ao longo das diversas reuniões, discussões, treinamentos e estudos.

Finalizada esta primeira atividade profissional no Brasil, fica a certeza de que a qualidade do curso de Engenharia Elétrica na UFCG, a participação em projetos de pesquisa e desenvolvimento desde o início da graduação, como o Brazil-IP, somado às experiências internacionais e outras atividades de extensão foram cruciais no processo de rápida inserção no mercado de trabalho. Ainda mais em se tratando de uma empresa multinacional de grande porte focada em pesquisa, desenvolvimento e inovação tecnológica em uma área complexa e desafiadora por natureza, como a microeletrônica.

## 6 REFERÊNCIAS BIBLIOGRÁFICAS

FREESCALE SEMICONDUCTOR, INC. Enhanced Time Processing Unit (eTPU) Preliminary Reference Manual. [S.l.], 2004.

FREESCALE SEMICONDUCTOR, INC. Disponível em: [www.freescale.com](http://www.freescale.com). Acesso em: 05 de fevereiro de 2015.

Bergeron, Janick. Writing Testbenches: Functional Verification of HDL Models, Second Edition. Kluwer Academic Publishers, Norwell, MA, USA, 2003.