



Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Curso de Graduação em Engenharia Elétrica

DANILO BARRETO CAVALCANTI

RELATÓRIO DE ESTÁGIO INTEGRADO

Campina Grande, Paraíba
Março de 2017

DANILO BARRETO CAVALCANTI

RELATÓRIO DE ESTÁGIO INTEGRADO

*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: Eletrônica

Orientador:

Professor Edmar Candeia Gurjão, D. Sc.

Campina Grande, Paraíba
Março de 2017

DANILO BARRETO CAVALCANTI

RELATÓRIO DE ESTÁGIO INTEGRADO

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: Eletrônica

Aprovado em / /

Professor Avaliador
Universidade Federal de Campina Grande
Avaliador

Professor Edmar Candeia Gurjão, Dr. Sc.
Universidade Federal de Campina Grande
Orientador, UFCG

Dedico este trabalho à minha família e amigos que conheci durante o curso por todos os tipos de apoio.

AGRADECIMENTOS

Agradeço aos meus pais, por todos os tipos imagináveis e inimagináveis de apoio dados em todas as etapas do curso.

Aos amigos do curso, que compartilham o mesmo caminho acadêmico e cooperaram para trilhá-lo da forma mais agradável possível.

Aos membros do Grupo PET-Elétrica, que possuem uma ligação muito especial desde muito cedo, desde tutores, a membros veteranos, a novos membros que perpetuam o grupo.

À equipe de Engenharia Elétrica da UFCG que é sempre eficiente e bem-intencionada para resolver todos os tipos de problema.

Ao meu supervisor de estágio, o Professor Raimundo Carlos Silvério Freire, pelo acolhimento no laboratório e disponibilidade para resolver qualquer demanda.

Ao meu orientador, Professor Edmar, por ter me acompanhado desde o começo até o fim do curso e sempre me mostrar caminhos interessantes a seguir com muito pragmatismo e honestidade.

Finalmente, agradeço em especial à dupla Tchai e Adail por me acompanharem de perto em todas as fases do processo de formalização do estágio.

“Spiral out. Keep going.”

(Lateralus por Tool)

RESUMO

Neste relatório são apresentadas as atividades desenvolvidas pelo aluno Danilo Barreto Cavalcanti no Laboratório de Instrumentação e Metrologia Científicas (LIMC), da Universidade Federal de Campina Grande (UFCG), para a concepção e implementação de um fluxo de projeto para projetos de microeletrônica digital. O fluxo começa de uma descrição em System Verilog, em seguida realiza síntese RTL e gera um leiaute equivalente. Também possui um ambiente para verificação integrado.

Palavras-chave: System Verilog, EDA, microeletrônica

ABSTRACT

This report presents the activities of the student Danilo Barreto Cavalcanti in the *Laboratório de Instrumentação e Metrologias Científicas (LIMC)*, of the Federal University of Campina Grande (UFCG), regarding the conception and implementation of a design flow for digital microelectronics projects. The flow begins with a System Verilog description, followed by RTL synthesis and the generation of an equivalent layout. It also has an integrated verification environment.

Keywords: System Verilog, EDA, microelectronics.

LISTA DE ILUSTRAÇÕES

Figura 1 - Diagrama do Fluxo de Projeto Proposto 13

SUMÁRIO

Agradecimentos.....	v
Resumo.....	vii
Abstract	viii
Lista de Ilustrações.....	ix
Sumário	x
1 Introdução.....	11
2 Apresentação do Laboratório.....	12
3 Atividades Desenvolvidas	13
3.1 Instalação e Configuração do Ambiente Cadence®.....	14
3.2 Síntese Lógica.....	14
3.3 Geração de Leiaute	15
3.4 Integração de Ambiente de Verificação Funcional	17
4 Considerações Finais	19
Bibliografia.....	20

1 INTRODUÇÃO

Este trabalho relata o estágio realizado pelo aluno Danilo Barreto Cavalcanti no Laboratório de Instrumentação e Metrologia Científicas (LIMC) situado na Universidade Federal de Campina Grande (UFCG). O estágio integrado faz parte da carga horária obrigatória para obtenção do grau de Bacharel em Engenharia Elétrica da UFCG, e foi realizado durante o período de 29 de novembro de 2016 a 27 de março de 2017, totalizando 680 horas.

O LIMC, coordenado pelo professor Raimundo Carlos Silvério Freire, supervisor deste estágio, possui atividades em eletrônica e microeletrônica tanto em ensino quanto pesquisa. Os seus membros são essencialmente alunos de graduação ou pós-graduação da UFCG, mas também há colaborações com outros laboratórios e instituições.

Esse estágio teve como objetivo a elaboração de um fluxo de projeto digital usando ferramentas da empresa Cadence®. O resultado deve auxiliar membros atuais e futuros a elaborarem projetos de microeletrônica digital

2 APRESENTAÇÃO DO LABORATÓRIO

O Laboratório de Instrumentação e Metrologia Científicas da Universidade Federal de Campina Grande é coordenado pelo professor Raimundo Carlos Silvério Freire. Sob sua administração, alunos de graduação e pós-graduação desenvolvem atividades de ensino e pesquisa em eletrônica analógica.

O Laboratório possui salas equipadas com materiais para ensino e pesquisa em Eletrônica. A organização e distribuição dos materiais dependem dos projetos realizados. Dessa forma, a estrutura organizacional tende a mudar a cada período letivo dos cursos de graduação e pós-graduação em Engenharia Elétrica na UFCG, uma vez que novas oportunidades de trabalho são ofertadas.

Há, em especial, uma sala destinada ao ensino e pesquisa em microeletrônica. Essa sala é equipada com computadores com o pacote de ferramentas da empresa Cadence®. Essas ferramentas auxiliam o desenvolvimento de circuitos para microeletrônica fornecendo simulações e automações para a realização de projetos.

3 ATIVIDADES DESENVOLVIDAS

O estágio foi proposto para cumprir quatro passos essenciais a um projeto de microeletrônica digital no laboratório:

1. Instalação e configuração do ambiente Cadence®;
2. Síntese lógica;
3. Geração de leiaute;
4. Integração de ambiente de verificação funcional.

Esses tópicos foram considerados essenciais a partir de uma análise dos recursos disponíveis no laboratório e de fluxos de projeto propostos pelos fabricantes, por manuais e por trabalhos científicos (EMMERT, 2007; SILVA, 2010; CADENCE, 2014). Os tópicos abordados foram adaptados ao tempo de estágio e ao pessoal disponível e o projeto foi estruturado como na Figura 1.

Outra imposição foi de que o fluxo deveria poder ser refinado em quaisquer dos pontos de acordo com novas descobertas e necessidades de projeto, desde que pudesse ser uma base para esses trabalhos futuros. Foi estipulado que o fluxo de projeto deveria atender a uma especificação de um contador de quatro bits simples descrito em System Verilog.

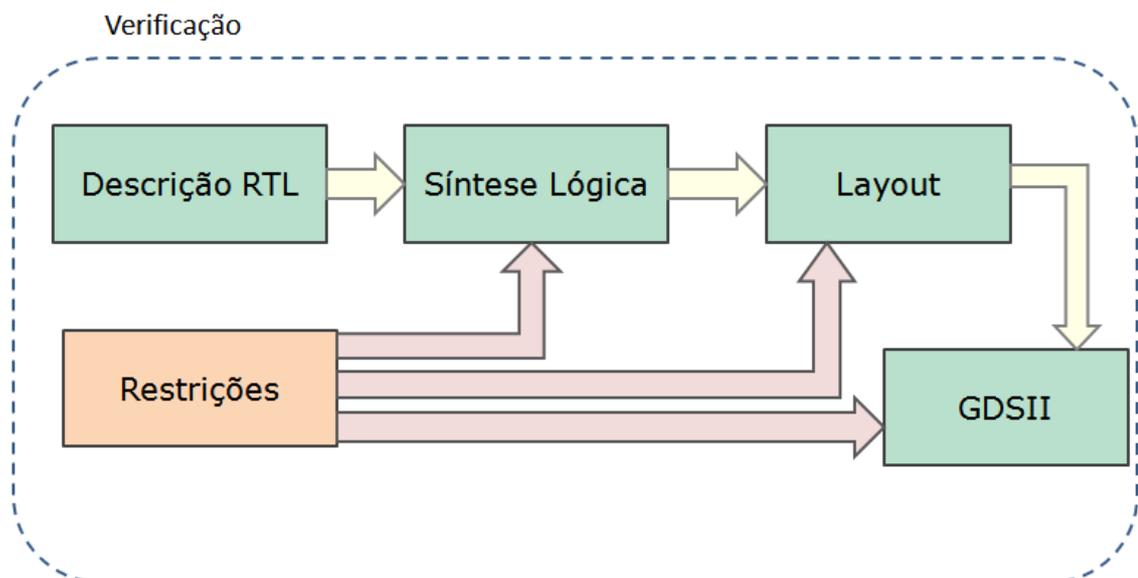


FIGURA 1 - DIAGRAMA DO FLUXO DE PROJETO PROPOSTO

3.1 INSTALAÇÃO E CONFIGURAÇÃO DO AMBIENTE CADENCE®

As máquinas de desenvolvimento já possuíam todas as ferramentas para desenvolvimento de projetos analógicos instaladas. Seguindo o padrão da instalação presente e os manuais fornecidos pela Cadence®, foi possível instalar as demais ferramentas necessárias.

A configuração do ambiente se deu pelo ajuste de variáveis de ambiente e pela criação de *scripts* do tipo *Makefile* e arquivos *tcl*. Essa tarefa foi concluída quando foi estabelecido um padrão para a inicialização de cada ferramenta via linha de comando.

3.2 SÍNTESE LÓGICA

Os projetos de microeletrônica digital começam por um modelo em linguagem de programação ou descrição de hardware (HDL) de alto nível (System C/System Verilog) que irão compor um modelo. Então, uma descrição um pouco menos abstrata é produzida em Verilog ou VHDL (duas HDL compatíveis com o ambiente). Esses dois arquivos que descrevem a função de um circuito a ser produzido precisam ser, eventualmente, comparáveis nas etapas de verificação.

A descrição menos abstrata, em Verilog ou VHDL, precisa ter código dito “sintetizável”. Um código é compreendido como sintetizável uma vez que possa ser traduzido em portas lógicas de forma automática por ferramentas de síntese. O que diferencia um código sintetizável de um não sintetizável é a presença de *macros* ou diretivas especiais que só podem ser entendidas por certos *hardwares* (como FPGA) ou certos programas (como em simulações), mas não podem ser traduzidos em expressões lógicas. A título de exemplo, uma estrutura *for* ou uma *macro* para geração de um bloco de memória não podem ser sintetizados.

Levando em consideração que a descrição Verilog fornecida é sintetizável, desconsiderou-se, a princípio, o passo de verificação com o modelo, uma vez que isso seria realizado posteriormente.

Elaborou-se um *script* para importação e síntese lógica dos módulos contidos em uma descrição Verilog sintetizável. O *script* executa a ferramenta adequada, ajusta o ambiente, carrega arquivos de restrição e retorna um outro arquivo Verilog da descrição sintetizada e relatórios sobre diversas considerações do circuito. Esse novo arquivo possui apenas instâncias de portas lógicas da biblioteca padrão Verilog e deve executar a mesma

função que a descrição de entrada. Essa verificação também foi adiada até a concepção da fase de verificação.

A partir desse ponto, a síntese lógica foi expandida para síntese RTL. O nível RTL indica *Register Transfer Level*, nível de transferência entre registradores, e é um dos níveis indicados para geração de leiaute no fluxo proposto. Ele abstrai conexões elétricas entre transistores enquanto mantém unidades com funções definidas. A síntese RTL consiste em traduzir o código sintetizável em unidades físicas de biblioteca. Na ausência de uma biblioteca de células, é possível obter um código RTL com células genéricas (AND, OR, XOR etc). Na presença, o programa utiliza algoritmos para traduzir os módulos descritos em células da biblioteca da tecnologia fornecida respeitando limitações fornecidas. É possível configurações limitações quanto a temporização, consumo de potência e área ocupada, por exemplo. Esse passo foi realizado, pois se viu necessário ter uma descrição em nível RTL para as fases posteriores.

Essa etapa foi desafiadora porque foi necessário estudar os arquivos de entrada para a ferramenta de síntese e seus formatos, diferenciando-os em essenciais, recomendáveis e opcionais. Para alcançar o estado final dessa etapa, foram consultados tutoriais *online* com versões anteriores da ferramenta e os manuais fornecidos digitalmente.

Finalmente, todas as linhas de código escritas nos *scripts* foram documentadas antes de prosseguir para a fase seguinte.

3.3 GERAÇÃO DE LEIAUTE

Um dos grandes desafios para um projeto de microeletrônica digital é a elaboração do leiaute. Como os projetos possuem uma grande quantidade de transistores, se fez necessária a adoção de ferramentas de automação dos passos desenvolvimento, em inglês *EDA (Electronics Design Automation) tools*. Essas ferramentas auxiliam principalmente no posicionamento (*placement*) e na inserção de conexões (*routing*) do circuito. Esse passo requereu uma quantidade considerável de tempo para que fosse concebido um fluxo de geração de leiaute.

Embora não existam tutoriais ou projetos-padrão a serem seguidos nos manuais da Cadence® ou do fornecedor dos arquivos de tecnologia, foi possível encontrar referências *online* para o fluxo de projeto proposto, embora com versões mais antigas da ferramenta (EMMERT, 2007; SILVA, 2010; MOREIRA & MORAES, 2013)

Ao analisar esse material, foi possível consultar os manuais disponíveis e conceber *scripts* para as diversas fases da geração de leiaute. Foi necessário estudar e entender cada sub-fase individualmente e observar se havia fragmentações ou aglutinações de passos a serem feitas. Isso gerou a seguinte sequência de etapas:

- 1 - *floorplanning, power planning*;
- 2 - *placement, clock tree synthesis (CTS), routing*;
- 3 - *filler cell addition, parasitic extraction, export*.

Os termos foram mantidos em inglês para condizer com a terminologia comumente utilizada na área. Traduzindo livremente, seriam: planejamento de ocupação, planejamento de alimentação, posicionamento, síntese de árvore de relógio, roteamento (ou “cabeamento”), adição de células de preenchimento, extração de componentes parasitas, exportação.

Esses passos costumam ser executados por um “*back-end designer*”, cujos conhecimentos devem direcionar a ferramenta a otimizar o chip em algum aspecto. Por exemplo, o chip pode ser utilizado em área ocupada, densidade, consumo e desempenho.

A seção mais complicada da sequência de passos foi a CTS seguida de *routing*, pois são consideradas bem importantes para a otimização de desempenho e consumo do chip. Para realizar esses passos, foi necessário entender um novo conceito de análise MMMC (*Multi-Mode, Multi-Corner*). O usuário deve mencionar ao programa algumas possibilidades de região de operação a partir de condições de funcionamento (tensão e temperatura) e de fabricação (minimizar resistência ou capacitância, por exemplo). Essas combinações são configuradas por meio da importação de arquivos de biblioteca de temporização (*timing libraries*). Quando as condições de análise MMMC são ajustadas, a ferramenta pode realizar a síntese da árvore de relógio e roteamento das interligações, pois também dispõe de um arquivo com restrições temporais a ser seguido.

Um desafio em especial dessa etapa foi a inclusão de “pads”. Pads são as partes metálicas que servem de alimentação e entrada e saída de sinais. O *Design Kit* utilizado possuía uma biblioteca específica para pads que precisava ser utilizada. Foi necessário buscar vários exemplos e referências externas quanto ao uso de tais pads, pois o manual do fornecedor não trazia direcionamentos quanto à instanciação ou à essencialidade de certos pads ao projeto. O redistribuidor da biblioteca foi contatado para fornecer esclarecimentos, mas a pessoa responsável não tinha informações adicionais além das presentes no manual.

Foram criadas duas versões do projeto: uma com a presença de pads (nomeada “chip”) e outra com a ausência (nomeada “IP”, de *intellectual property*). O projeto foi feito

totalmente sem os pads para só então passar para o passo seguinte. A diferença entre a biblioteca de células padrão e a biblioteca de pads fez com que algumas etapas tivessem que ser explicitamente diferenciadas no fluxo, o que foi observado como uma particularidade do *Design Kit* disponível.

Em seguida, era necessário verificar se a ferramenta tinha executado todas as etapas corretamente sem produzir falhas estruturais ou erros de conexão. Embora a ferramenta dispusesse de um kit de verificação próprio, havia uma ferramenta explicitamente feita para essa tarefa. Foi necessário estudar os formatos compatíveis a essa ferramenta e fluxos para a execução de DRC (*design rule check* - verificação de regras de desenho) e LVS (*layout versus schematic* - leiaute versus esquemático).

Essa última etapa foi uma das que mais demandou tempo, pois todos os exemplos e referências encontrados ou se referiam ao universo analógico (demonstrando a ferramenta com um inversor, por exemplo) ou a situações muito genéricas, sem informar arquivos ou formatos compatíveis. Esses problemas foram resolvidos compreendendo as entradas e saídas das ferramentas e como elas executavam internamente a partir de relatórios de execução e consultando especificidades nos manuais.

O trabalho foi considerado concluído quando era possível importar todos os arquivos de projeto, gerar o leiaute passo a passo com a possibilidade de modificação de parâmetros e ter como resultado um arquivo de formato GDS-II que passasse nos testes DRC e LVS. Esse formato é um dos principais para enviar para fabricação, efetuar certas verificações e criar bibliotecas de células.

Além disso, essa etapa também gera arquivos em Verilog que representam as instâncias e interconexões das células e módulos gerados pelo usuário, acompanhado de anotações de atraso de propagação de sinais. Esses arquivos podem ser carregados em outra ferramenta para simulação ou verificação.

Todas as linhas de código dos *scripts* gerados foram documentadas com referência ao manual da Cadence® e podem ser utilizados individualmente ou de forma sequencial.

3.4 INTEGRAÇÃO DE AMBIENTE DE VERIFICAÇÃO FUNCIONAL

Passos de verificação podem ser adicionados antes, depois ou durante quaisquer fases de um fluxo de projeto, em vários tipos de fluxos de projeto. No planejamento proposto,

foi requerida apenas a inserção do passo de verificação funcional, que pretende analisar se a funcionalidade do projeto está de acordo com o especificado por um modelo.

Os passos de verificação, de forma geral, são ditos os mais importantes durante o desenvolvimento de um projeto digital, pois podem garantir qualidade e minimizar falhas e efeitos indesejados durante a fase de prototipagem e fabricação (WEST & HARRIS, 2009).

Nesse estágio, foi criado um *script* que associa um conjunto de testes (*testbench*) a alguma das fases do projeto em comparação com o modelo de referência. É possível fazer uma análise comparativa entre o modelo com o arquivo em Verilog sintetizável; com o arquivo pós-síntese; e o arquivo pós-leiaute. As fases de síntese RTL e leiaute possuem anotações de atraso de propagação de sinal das portas instanciadas que são carregadas durante esses passos de verificação.

O arquivo *testbench* foi escrito em SystemVerilog e não possui padrão formal comercial (utilização de UVM ou OVM, por exemplo) nem complexidade, uma vez que necessita apenas testar os dezesseis estágios do contador de quatro bits e a funcionalidade do sinal de reinício. Isso está em consonância com as atividades propostas, uma vez que Verificação é uma área abrangente e que requer um investimento considerável para ser corretamente aplicada.

O desafio dessa etapa foi garantir que os arquivos gerados pelas outras ferramentas pudessem ser carregados corretamente para fins de simulação e verificação. Para isso foi necessário entender como configurar a ferramenta para os arquivos de entrada disponíveis e relacioná-los com referências ao *Design Kit*. Esse problema foi resolvido a partir da modificação de um exemplo e do estudo dos manuais dessa ferramenta que continham opções adicionais de uso.

4 CONSIDERAÇÕES FINAIS

As atividades desenvolvidas durante o estágio foram de suma importância para a complementação de minha formação em Eletrônica, apresentando conceitos e ferramentas nunca antes vistos durante o curso de graduação. Foi possível ter contato com ferramentas comerciais para desenvolvimento de projetos de microeletrônica digital ao mesmo tempo em que contribuí para a realização de atividades futuras do Laboratório.

É importante notar que o fluxo de projeto fruto deste trabalho de estágio serve de base para uma equipe desenvolver cada seção proposta individualmente e sugerir novos passos intermediários. A partir da experiência adquirida durante o processo, nota-se que a excelência na produção de um circuito digital depende da especialização de vários membros trabalhando em uma equipe de projeto para fornecer o melhor que as ferramentas têm a oferecer em determinado assunto.

Finalmente, acredita-se que, embora não tenha sido possível fabricar um circuito durante o tempo de realização do estágio, o fluxo proposto cobre boa parte dos requisitos para que um circuito seja aceito por uma fabricante.

Bibliografia

Cadence. (novembro de 2012). *Common Power Format User Guide. 2.0.*

Cadence. (agosto de 2013). *Common Power Format Language. 2.0.* Estados Unidos da América.

Cadence. (setembro de 2014). *Assura Physical Verification Developers.*

Cadence. (setembro de 2014). *Assura Physical Verification User Guide.*

Cadence. (junho de 2014). *EDI System Text Command Reference.*

Cadence. (junho de 2014). *EDI System User Guide.*

Cadence. (junho de 2014). *LEF/DEF Language Reference.*

EMMERT, J. M. (22 de agosto de 2007). *Prof. J. M. Emmert (Marty)*. Acesso em novembro de 2016, disponível em <http://cecs.wright.edu/~emmert/tutorials/>

MITAL, P. (2016). *VLSI Concepts*. Acesso em dezembro de 2016, disponível em <http://www.vlsi-expert.com/>

MOREIRA, M., & MORAES, F. (2013). *Tutorial para Síntese Standard-Cells Utilizando Cadence.*

SILVA, L. M. (2010). *Tutorial: Fluxo de Projeto ASIC para Circuitos Integrados Digitais*. Porto Alegre.

WEST, N. H., & HARRIS, D. M. (2009). *CMOS VLSI Design: A Circuits and Systems Perspective* (Fourth ed.).