



Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Curso de Graduação em Engenharia Elétrica

CAIO LUIZ GOMES DE SOUSA ALVES

**ESTÁGIO NA GERÊNCIA DE MICROELETRÔNICA DE REDES
CONVERGENTES DO CPQD**

Campina Grande, Paraíba
Setembro de 2014

CAIO LUIZ GOMES DE SOUSA ALVES

ESTÁGIO NA GERÊNCIA DE MICROELETRÔNICA DE REDES
CONVERGENTES DO CPQD

*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 Marcos Ricardo Alcântara Morais, D. Sc.

Campina Grande, Paraíba
Setembro de 2014

CAIO LUIZ GOMES DE SOUSA ALVES

ESTÁGIO NA GERÊNCIA DE MICROELETRÔNICA DE REDES CONVERGENTES DO CPQD

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 Avaliador
Universidade Federal de Campina Grande
Avaliador

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

Dedico este trabalho aos meus pais, Guilherme e Fátima, por sempre estarem concedendo apoio e amor incondicional, incentivando a nunca desistir.

“É capaz quem pensa que é capaz.”

Buda

LISTA DE FIGURAS

Figura 1 - Representação geral de um testbench.....	12
Figura 2 - Representação detalhada de um testbench	12
Figura 3 - Representação da Sequência de Blocos	14
Figura 4 - Representação do bloco slicer	17
Figura 5 - Representação do esquema do Teste	17
Figura 6 - Representação da funcionalidade do driver	18
Figura 7 - Tela do programa Incisive Enterprise Manager	19
Figura 8 - Tela do Programa ICCR.....	20

LISTA DE ABREVIATURAS E SIGLAS

RTL – Registro de Transferência de Nível - Register Transfer Level

DSP – Processador Digital de Sinais - Digital Signal Processor

DUT – Dispositivo Sob Teste - Device Under Test

ASIC – Circuito Integrado de Aplicação Específica – Application Specific
Integrated Circuit

SUMÁRIO

1	Introdução.....	9
1.1	Fundação CPqD	9
1.2	Objetivos	10
1.2.1	Objetivos Específicos	10
2	Fundamentação Teórica.....	11
2.1	Verificação Funcional.....	11
2.1.1	UVM.....	13
2.2	Processamento Digital de Sinal.....	14
2.2.1	DSP para redes coerentes.....	14
3	Atividades Realizadas	16
3.1	Criação de testes	16
3.1.1	Bypass do Slicer	16
3.1.2	Simulação do bloco Mapper	17
3.2	Driver de comunicação	17
3.3	Regressão	18
3.4	Análise de cobertura	19
4	Conclusão	21
	Referências	22

1 INTRODUÇÃO

Neste capítulo será apresentado a história da empresa concedente do estágio, como também o objetivo do trabalho realizado no centro de pesquisa.

1.1 FUNDAÇÃO CPQD

De acordo com o site da empresa, o CPqD – Centro de Pesquisa e Desenvolvimento em Telecomunicação – é uma fundação de direito privado que atua na área de tecnologias da informação e comunicação, sendo o maior centro de pesquisa e desenvolvimento da América Latina em sua área de atuação. Os produtos do CPqD são voltados para os setores: telecomunicações, financeiro, energia elétrica, industrial, corporativo, administração pública e inclusão digital.

O CPqD iniciou suas atividades em 1976 como Centro de Pesquisa e Desenvolvimento em Telecomunicações da Telebrás, empresa estatal responsável pelos serviços públicos de telecomunicações do Brasil. O objetivo inicial foi suprir as necessidades tecnológicas do setor de telecomunicações brasileiro. De acordo com o site da empresa, o Centro passou a ter um papel fundamental no setor, desenvolvendo produtos e equipamentos para suprir a necessidade do sistema de comunicações, diminuindo a importação que antes era realizada e passando a exportar as soluções e equipamentos desenvolvidos.

Ao longo de sua trajetória, o CPqD sempre desempenhou um papel estratégico no setor telecomunicações. Nos primeiros anos de sua existência, voltou-se prioritariamente para as tecnologias emergentes, pesquisando, desenvolvendo e transferindo para a indústria diversos produtos de grande relevância para o mercado mundial.

O CPqD contribuiu de forma efetiva para a criação de uma “inteligência nacional” nas áreas de telecomunicações e tecnologia da informação. Em 1990, com a maior exposição nacional à competitividade internacional, o CPqD se responsabilizou pela evolução tecnológica do setor de telecomunicações. O Centro passou a enfatizar o crescimento da indústria nacional com suporte tecnológico e parcerias em pesquisas.

Em 1998, com a privatização das telecomunicações, o CPqD tornou-se uma fundação independente sem fins lucrativos, ampliando assim, o seu campo de atuação. Os produtos e serviços do CPqD foram reposicionados comercialmente no mercado de telecomunicações e de tecnologia da informação. Segundo o site da empresa, a carteira de clientes aumentou, incluindo todas as concessionárias de serviços de telecomunicações no Brasil, celulares, TV's a cabo e recentemente as empresas de distribuição de água e energia elétrica.

O principal produto do CPqD é a geração de conhecimentos que chegam na forma de tecnologias de produto, sistemas de softwares, serviços tecnológicos, consultorias e participação com capital intelectual em empresas emergentes de base tecnológica.

A fundação é parcialmente financiada por fundos governamentais de ciência e tecnologia. As tecnologias de produto geradas pela instituição são repassadas para outras empresas, que se responsabilizam pela sua produção e comercialização. Assim, o CPqD não comercializa os seus produtos para clientes finais, ele se responsabiliza em desenvolver e disseminar novas tecnologias, produtos e serviços. Empresas como: Padtec, Trópico, ClearTech, Instituto Atlântico, JÁ!, CPqD Technologies & Systems, Civcom e WxBR foram criadas a partir de iniciativas de parcerias tecnológicas criadas pelo CPqD.

1.2 OBJETIVOS

Desenvolver atividades com relação à verificação de circuitos de alta complexidade, referente à implementação de um sistema de processamento digital de sinal em um circuito integrado para altas taxas de transmissão.

1.2.1 OBJETIVOS ESPECÍFICOS

Implementar testes para garantir a funcionalidade do circuito que está sendo desenvolvido e analisar a quantidade de código RTL que foi excitado.

2 FUNDAMENTAÇÃO TEÓRICA

A microeletrônica surge como uma área que tenta solucionar grande parte dessas exigências impostas pelo mercado, integrando a maioria das funcionalidades específicas de um projeto em um circuito.

Segundo Deprá, et al. (2007), o desenvolvimento de circuitos integrados tem atingido níveis de complexidade e custos, tais que a adoção de técnicas, ferramentas e metodologias se tornaram imprescindíveis para prover a eficiência de um componente.

De acordo com, Costa, et al. (2012), houve a necessidade de verificar o funcionamento de circuitos integrados, com isso, a metodologia de verificação funcional vem se destacando no ambiente da arquitetura digital.

2.1 VERIFICAÇÃO FUNCIONAL

Segundo Bergeron (2003), a verificação funcional é um processo usado para demonstrar que o objetivo do projeto é preservado em sua implementação.

De acordo com Abdí (2005), mais de 70% do esforço para o desenvolvimento de um circuito integrado são gastos na verificação do circuito, sendo eles direcionados a verificação funcional e lógica do sistema. Ainda segundo o autor, o número de engenheiros de verificação aconselha ser duas vezes superior ao número de engenheiros de design, devido ao grande esforço exigido para realizar a verificação dos *RTL*.

Para a verificação ser provida, há a necessidade de se construir um ambiente, com o propósito de ser reusável para outros dispositivos. Este ambiente consiste na inserção de estímulos, randomizados ou cuidadosamente inseridos, na entrada do *DUT* e posteriormente na coleta dos dados de saída do dispositivo. Com o intuito de realizar a comparação desses valores com os valores especificados no início do projeto ou colhidos pelo modelo ao qual está sendo a referência. Este ambiente é conhecido como *Testbench* (Figura 1).

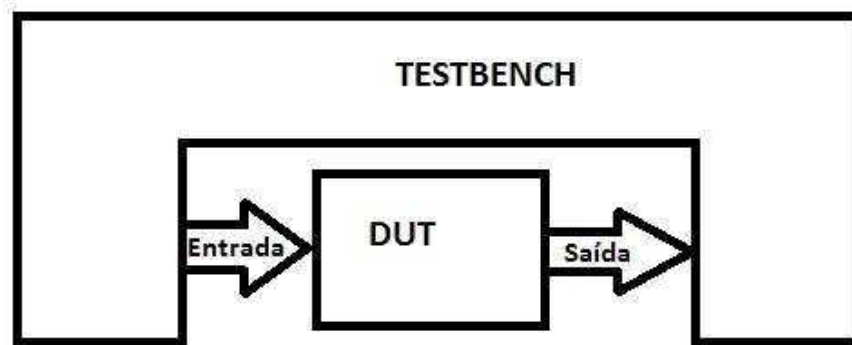


FIGURA 1 - REPRESENTAÇÃO GERAL DE UM TESTBENCH

Spears (2008) separa o *testbench* em camadas, o qual ajuda a fazer as tarefas mais fáceis de forma que os códigos estão divididos em pequenos pedaços. Estas camadas são divididas da seguinte forma: Sinal, Comandos, Funcional, Cenário e Testes. Dentre essas camadas, tem a que mede a cobertura funcional do código.

Ainda segundo Spears (2008), a camada Sinal é constituída pelos sinais de entrada e saída que excitam o dispositivo a ser testado. A camada de Comandos é composta pelos componentes que fazem a comunicação do dispositivo, que são: *Driver*, *Monitor* e Afirmações (*Assertions*). Na camada Funcional, os componentes são os que recebem as transações e fazem a comparação com o valor esperado, que são: *Agent*, *Scoreboard*, *Checker*. A camada Cenário tem a idéia de gerar as transações para serem passadas para o *Agent*, é nela que está contido o *Generator*. A camada de Testes é composta pelos testes ao qual farão a excitação do circuito. Na Figura 2 pode ser vista uma representação de como é composto um *testbench*.

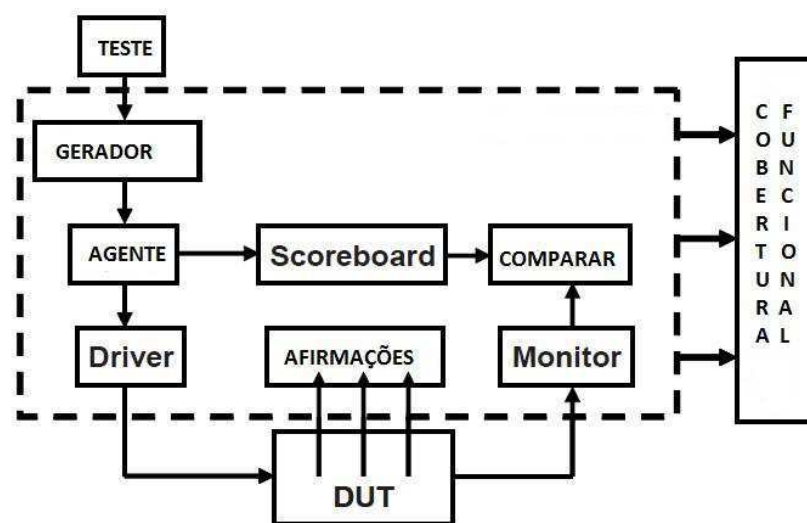


FIGURA 2 - REPRESENTAÇÃO DETALHADA DE UM TESTBENCH

A implementação do ambiente de verificação, *testbench*, é feita usando biblioteca do *System Verilog*. De acordo com Rosenberg e Meade (2010), desde 2000, as grandes indústrias do setor de microeletrônica, lançam metodologias de verificação para ajudar no desenvolvimento dos projetos, tais metodologias foram: *vAdvisor* (Cadence,2000), *eRM* (Cadence, 2002), *RVM* (Synopsys, 2003), *AVM* (Mentor, 2006), *VMM* (Synopsys, 2006), *URM* (Cadence, 2007), *OVM* (Cadence, Mentor, 2008), *UVM* (Accellera,2010).

A metodologia utilizada neste trabalho foi a mais atual, a metodologia UVM.

2.1.1 UVM

Introduzida pela Accellera em 2010, essa metodologia é baseada na sua metodologia antecessora, *OVM*.

Segundo Rosenberg e Meade (2010), as características que fizeram a *UVM* ser lançada ao mercado e amplamente utilizada pelas grandes design houses do mundo, foram:

- Maturidade, baseada em *OVM*, com modificações no topo que permitiram ser mais robusta;
- Apresenta formato *open-source*;
- Compatível com a maioria dos simuladores comerciais.

Ainda, de conformidade com Rosenberg e Meade (2010), olhando pelo lado técnico, a metodologia *UVM*, têm as seguintes fatores chave:

- Fornece a possibilidade tornar a verificação em arquivos mais sucintos e simples;
- Geração de estímulos;
- Cria um completo *testbench*, fornecendo classes que ajudam a automatizar o processo de verificação;
- Possibilita uma melhor prática para estimular a cobertura funcional do código RTL.

2.2 PROCESSAMENTO DIGITAL DE SINAL

Segundo Lopes (2012), o processamento de sinais é o mecanismo de realizar o tratamento de sinais, que podem ser analógico e digital.

Esse tratamento pode consistir na adaptação, modulação, compressão e extração de alguns parâmetros que está contido no sinal. No sinal digital, esse processamento é feito de forma discreta, ao qual é possível fazer abstrações matemáticas mais gerais e a implementação é mais confiável, flexível e tem o melhor custo-benefício em relação ao processamento analógico.

2.2.1 DSP PARA REDES COERENTES

De acordo com Ribeiro (2012), como os receptores coerentes intradinos, que são aqueles que transferem as características do sinal óptico para o domínio elétrico, realiza o batimento do laser do sinal com o laser do oscilador local, então sabe-se que nesse batimento a fase e a frequência dos lasers não são as mesmas.

Para isso, com a necessidade de haver a coerência para estimar fase e frequência, no domínio digital, empregaram o uso dos DSP's para altas velocidades, que são empregados na área óptica. Os quais, os algoritmos que os implementam são divididos em blocos de acordo com as suas principais funcionalidades.

Segundo Ribeiro, Diniz, et.al (2011), a sequência de algoritmos, conforme suas funcionalidades, que são divididos em blocos é apresentado na Figura 3.

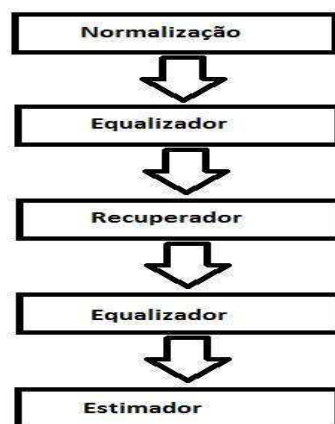


FIGURA 3 - REPRESENTAÇÃO DA SEQUÊNCIA DE BLOCOS

- Normalização: Compensar distorções e diferença de potência;
- Equalizadores: Compensar as dispersões, separar eixos de polarização e equalizar;
- Recuperador: Recuperar o tempo;
- Estimador: Estimação de desvio de frequência entre os lasers.

3 ATIVIDADES REALIZADAS

Neste capítulo, serão apresentadas as atividades realizadas durante o estágio na Fundação CPqD. Essas atividades serão divididas em 4 tópicos, que são eles: testes realizados para validação de blocos, driver de entrada, regressões executadas e análise de cobertura após as regressões serem executadas.

3.1 CRIAÇÃO DE TESTES

Como visto na fundamentação teórica, a verificação é constituída por testes que tem a finalidade de gerar estímulos de entrada para excitar determinadas funcionalidades do bloco a ser verificado, visando analisar a sua saída e a quantidade do código *RTL* que é coberto.

Ao decorrer do trabalho realizado no estágio, foram criados dois testes para dois blocos do *ASIC*, que são eles: *bypass* e simulação do bloco *mapper*.

3.1.1 BYPASS DO SLICER

Por se tratar de um DSP, a maioria dos blocos que o constitui apresentam a função de *bypass*, que se trata da verificação de conectividade do bloco.

O bloco apresenta uma entrada na forma complexa: uma fase e outra a amplitude. Normalmente, por erro de arredondamento, os sinais de entrada do bloco não possuem o mesmo valor que o de referência para a determinada modulação. Com isso, os pontos de entrada ficam dispersos na área do quadrante ao qual está contido.

Para que todos os pontos de entrada fiquem agrupados em um ponto que é tido como o valor de referência, então o bloco tem a função de verificar o quadrante que a entrada está e com isso, ajustá-lo para o ponto de referência do quadrante. Em outras palavras, o bloco consiste de um mapeador dos sinais.

Esta tarefa consistiu na comparação do sinal de entrada, do bloco, com o sinal de saída, com o intuito de ter a certeza que esses sinais são iguais, Figura 4. Então, é concluído que o bloco não exerceu nenhuma ação sobre esses sinais e que a conectividade interna do bloco está feita.



FIGURA 4 - REPRESENTAÇÃO DO BLOCO SLICER

3.1.2 SIMULAÇÃO DO BLOCO MAPPER

Como um circuito integrado é constituído por blocos que exercem diferentes funcionalidades, ao decorrer do fluxo do processo, o sinal de entrada é processado por esses blocos, e ao final do processo, se deseja recuperar o sinal de entrada.

Normalmente, os primeiros blocos são empregados a extrair as primeiras informações contidas nos sinais e então, passar esses dados com o primeiro tratamento feito para o próximo bloco, fazendo com que ao término do fluxo do processo, o último bloco exerça o inverso da função do primeiro bloco.

Então, para que se possa ter a certeza que o último bloco está realizando a tarefa ao qual foi implementado, cria-se um teste com as fórmulas que constituem o primeiro bloco, tendo o cuidado de verificar se a saída de um é compatível com a entrada do outro, gerando estímulos de entrada aleatórios para esse teste, pode-se fazer a comparação da entrada do teste com a saída do bloco e vê se os sinais são iguais. Uma representação é apresentado na Figura 4.

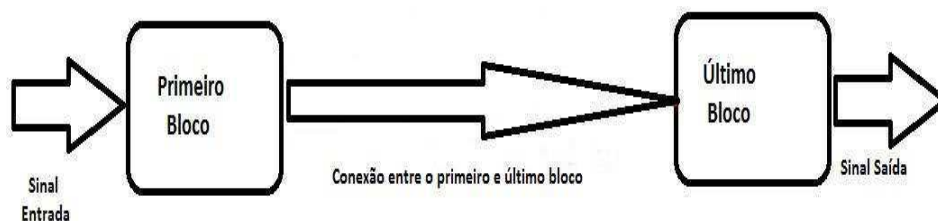


FIGURA 5 - REPRESENTAÇÃO DO ESQUEMA DO TESTE

3.2 DRIVER DE COMUNICAÇÃO

Com o intuito de realizar a comunicação com os dispositivos externos ao ASIC, tem-se a necessidade da implementação de um driver, Figura 6.

Normalmente escrito na linguagem System Verilog, esse componente apresenta as funções essenciais para estabelecer a comunicação, que são: escrita e leitura. De

acordo com as especificações, a função de escrita é feita quando há dados para serem gravados e quando o dispositivo está pronto para receber esses dados. Já a função de leitura, o dispositivo externo envia uma solicitação de leitura de um dado contido em um endereço do registrador de entrada.

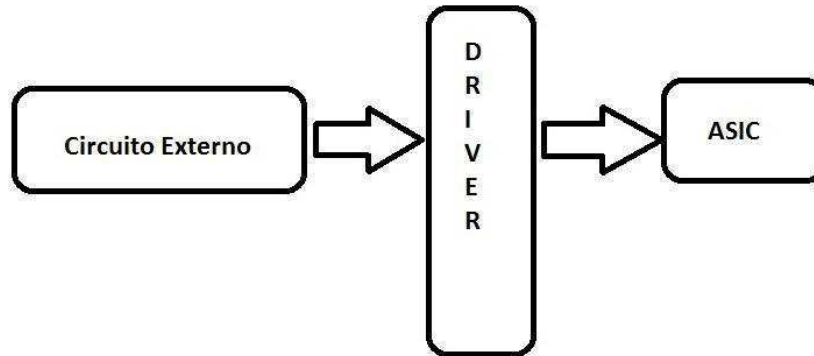


FIGURA 6 - REPRESENTAÇÃO DA FUNCIONALIDADE DO DRIVER

3.3 REGRESSÃO

Segundo Oliveira Júnior, a regressão tem o propósito de detectar a reintrodução de erros que levem o projeto a um nível de menor aperfeiçoamento, com isso, é tido como um dos passos que garante a não alteração do fluxo do circuito.

Então, quando todos os testes, implementados para um determinado bloco, estão realizando a verificação de forma correta e que as comparações implementadas estão satisfeitas. Nesse momento, todos os testes são colocados em execução com um determinado número de interações, no mínimo 1000, a serem feitas por cada padrão.

Inicialmente, em uma tabela são escritos os padrões que deverão ser executados, seguindo a regra estabelecida a seguir:

- Primeira coluna: Nome do teste;
- Segunda coluna: Nome do bloco;
- Terceira coluna: Nome do arquivo do log da simulação;
- Quarta coluna: Comando a ser passado para executar o teste, que deve conter os arquivos a serem lidos, a quantidade de interações, a modulação, dentre outras coisas.

Essa tabela é salva em formato .csv e posteriormente transformada em uma extensão específica. Então, a regressão é iniciada quando aberto o programa da *Cadence, Incisive Enterprise Manager*, Figura 7.

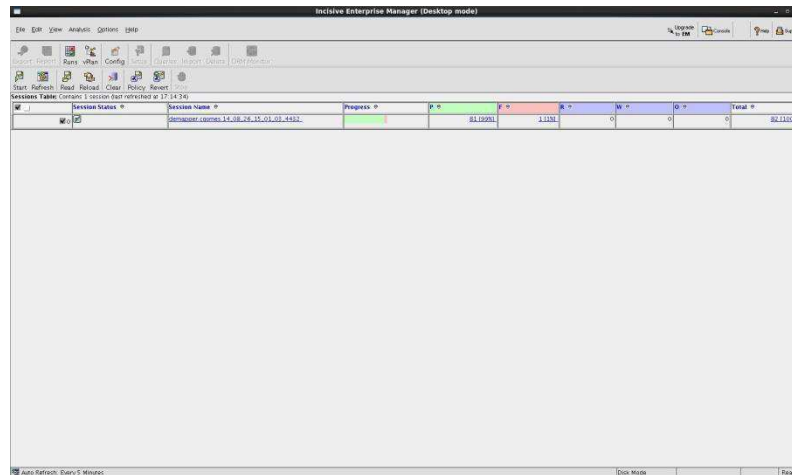


FIGURA 7 - TELA DO PROGRAMA INCISIVE ENTERPRISE MANAGER

3.4 ANÁLISE DE COBERTURA

Uma das partes mais importantes da verificação funcional é a análise de cobertura de código, pois é nesse momento em que podemos analisar se todo o código RTL foi excitado.

Utilizando o programa da Cadence, ICCR, esta tarefa é feita posterior a execução da regressão, essa parte consiste em verificar se houve transição de sinais, chamado de *toggle*, os módulos foram cobertos, e as expressões contidas foram todas executadas. Com isso, há uma porcentagem para cada um desses itens, como visto na Figura 8

O ideal é que todos esses itens tenham sido excitados em 100%, pois daria total confiabilidade ao sistema, sem a preocupação que, eventualmente, uma parte do circuito não coberta, chegasse a ser excitada no mundo real, pois poderíamos não saber o seu próximo comportamento. Porém, é muito difícil atingir a cobertura por completo.

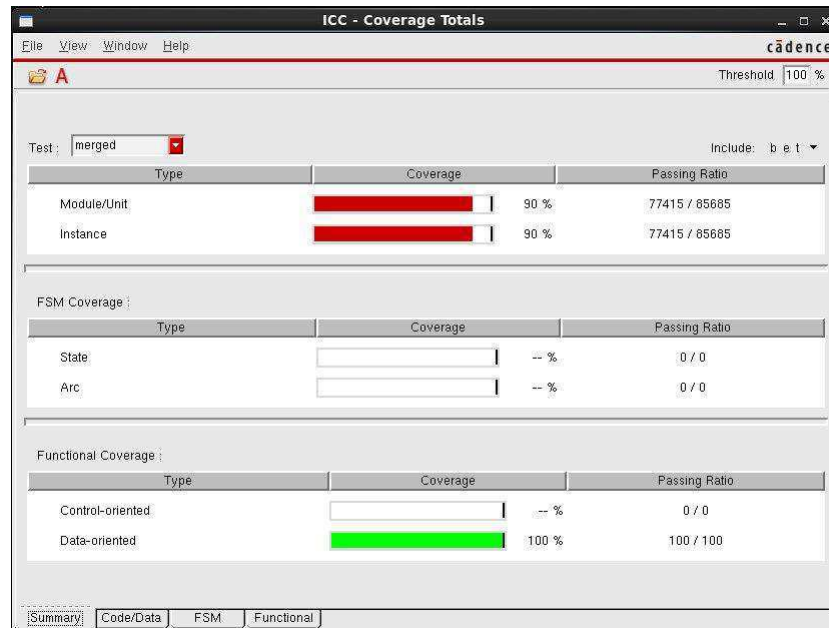


FIGURA 8 - TELA DO PROGRAMA ICCR

4 CONCLUSÃO

Ao término do período de estágio pode-se destacar os estudos sobre Redes Ópticas, Processamento Digital de Sinais, desenvolvimento de padrões para verificação de um SoC (System-on-Chip).

Outro fator foi a oportunidade de contribuir para um centro de pesquisa nacional que tem relevância internacional, possibilitando ter contato com a mais avançada tecnologia, está inserido em um projeto pioneiro à nível nacional, pois não é conhecido nenhum centro brasileiro que fez um sistema de grande complexidade e com essa tecnologia empregada. As reuniões de acompanhamento e o trabalho em equipe foi o que possibilitou o cumprimento das atividades descritas ao longo deste trabalho.

O ponto mais importante desta atividade foi o ganho profissional, por ter tido contato com pessoas de grande experiência na área e que sempre estiveram prontas para ajudar.

REFERÊNCIAS

Abdi, S., **Functional Verification of System Level Model Refinements**. Dissertation. University of California. Irvine, 2005. Disponível em: <

Bergeron, J., **Writing testbenches: Functional Verification of HDL Models**. Springer, 2003.

Costa, E. B. *et. al*, Verificação Funcional para Sistemas Digitais utilizando System Verilog. II Simpósio Brasileiro de Engenharia de Sistemas Computacionais. Natal, 2012. **Anais eletrônicos**. Disponível em: <http://sbesc.lisha.ufsc.br/sbesc2012/tiki-download_file.php?fileId=112>.

Deprá, D. A., Zatt, B., Santos, M. B. e Bampi, S., **Metodologia para Verificação Funcional de Hardware através de Co-simulação Paralela dentro de Sistemas de Software Complexos usando PLI: Decodificador H.264/AVC como Estudo de Caso**. UFRGS e Instituto Superior Técnico-Portugal. Hifen, Uruguaiana, V.31, nº 59/60, I/II semestre, 2007.

Lopes, R. R., **Aula sobre Processamento Digital de Sinais**. Departamento de Comunicação. Universidade Estadual de Campinas. Campinas, 2012.

Meyer, A. e Foster, H., **Metrics in SoC Verification**. Design and Verification Tecnology Division, Mentor Graphics Corporation. USA, 2010.

Oliveira Júnior, J. D., **Verificação Funcional de Modelos Transacionais de Processadores**. Universidade de Brasília. Brasília, 2014.

Ribeiro, V. B., **Filtros Digitais Para Recepção Coerente em 112Gb/s de Sinais Ópticos com Modulação QPSK e Multiplexação por Divisão em Polarização**. Dissertação. Universidade Estadual de Campinas. Campinas, 2012.

Ribeiro, V. B. *et. al*, Processamento de sinais para redes ópticas coerentes digitais de alta velocidade. Centro de Pesquisa e Desenvolvimento em Tecnologia-CPqD. **Cad.CPqD Tecnologia**, v.7, n.2, p7-30, Jul./dez. 2011. Campinas

Rosemberg, S. e Meade, K. A., **A Practical Guide to Adopting the Universal Verification Methodology (UVM)**. Cadence Design Systems. USA, 2010.

Site do CPqD, <www.cpqd.com.br> , acessado 07 de Julho de 2014

Spear, C., **SystemVerilog for verification, A guide to Learning the Testbench Language Features**. Second Edition, Marlboro, 2008.