



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

ANNE GABRIELE ARCANJO DA SILVA

**CONSTRUÇÃO E VALIDAÇÃO DE MANUAL PARA APLICAÇÃO
DA METODOLOGIA MDV DE VERIFICAÇÃO FUNCIONAL**

CAMPINA GRANDE - PB

2021

ANNE GABRIELE ARCANJO DA SILVA

**CONSTRUÇÃO E VALIDAÇÃO DE MANUAL PARA APLICAÇÃO
DA METODOLOGIA MDV DE VERIFICAÇÃO FUNCIONAL**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharela em
Ciência da Computação.**

Orientador: Professor Dr. Elmar Melcher.

CAMPINA GRANDE - PB

2021

ANNE GABRIELE ARCANJO DA SILVA

**CONSTRUÇÃO E VALIDAÇÃO DE MANUAL PARA APLICAÇÃO
DA METODOLOGIA MDV DE VERIFICAÇÃO FUNCIONAL**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharela em
Ciência da Computação.**

BANCA EXAMINADORA:

**Professor Dr. Elmar Uwe Kurt Melcher
Orientador – UASC/CEEI/UFCG**

**Professor PhD. Kyller Costa Gorgônio
Examinador – UASC/CEEI/UFCG**

**Professor Tiago Lima Massoni
Professor da Disciplina TCC – UASC/CEEI/UFCG**

Trabalho aprovado em: 30 de Março de 2021.

CAMPINA GRANDE - PB

ABSTRACT

Objective: The article describes the process of construction, content validation and semantic validation of a manual on Metric-Driven Verification (MDV) methodology application. **Method:** Methodological and descriptive survey conducted in 2021, following the steps: bibliographical survey, manual construction, validation of content by experts along with adequacy of the manual, semantic validation by verification engineers through a GoogleForms composed with 18 items of Likert scale variation levels: totally disagree, disagree, neither agree nor disagree, agree and totally agree. The study participants in content validation were 2 judges and in the semantic validation 19 experts, all of them verification engineers from Idea! Electronic Systems. **Results:** The manual was considered valid by the judges and, among the experts, there was an average total agreement of 67% in the semantic evaluation. **Conclusion:** A possible obstacle in using the manual is the need to adopt the Verification Plan framework provided therein. It is expected that despite the possible obstacles in its application, it will be used by professionals of hardware verification as an auxiliary tool to know and adopt the methodologies driven by metrics, as well as to enable them to enjoy the benefits offered by them.

Descriptors: Metric-driven verification, functional verification, verification plan, manual, validation.

Construção e Validação de Manual para Aplicação da Metodologia MDV de Verificação de Hardware

Anne Gabriele Arcanjo da Silva
Universidade Federal de Campina Grande
Departamento de Sistemas e Computação
Campina Grande - PB, Brasil
anne.silva@ccc.ufcg.edu.br

Elmar Uwe Kurt Melcher
Universidade Federal de Campina Grande
Departamento de Sistemas e Computação
Campina Grande - PB, Brasil
elmar.melcher@gmail.com

RESUMO

Este trabalho tem como objetivo descrever o processo de construção, validação de conteúdo e validação semântica de um manual para aplicação da metodologia *Metric-Driven Verification* (MDV) de verificação de *hardware*. Trata-se de uma pesquisa descritiva e metodológica, realizada entre os anos de 2021 e 2022, seguindo as etapas: levantamento bibliográfico; construção do manual em paralela validação de conteúdo pelos juízes e adequação; validação semântica por engenheiros de verificação primeiros interessados no produto final através de formulários com 18 itens compostos por níveis de variação da escala Likert: discordo totalmente, discordo, não concordo e nem discordo, concordo e concordo totalmente. Participaram da validação de conteúdo, 2 juízes e na validação semântica 19 peritos, sendo todos engenheiros de verificação da empresa Idea Electronic Systems. O manual foi considerado válido pelos juízes e, entre os peritos, houve concordância total média de 67% na avaliação semântica. Um possível obstáculo no uso do manual é a necessidade de adotar a estrutura do plano de verificação nele oferecida. Espera-se que apesar dos possíveis obstáculos na sua aplicação, este seja utilizado pelos profissionais de verificação de *hardware* como ferramenta auxiliar para conhecer e adotar as metodologias dirigidas por métricas, assim como possibilitá-los usufruir dos benefícios por elas ofertados.

Palavras-chave

Metric-driven verification, verificação funcional, plano de verificação, manual, validação

1. INTRODUÇÃO

Dentro do fluxo de desenvolvimento na microeletrônica, os processos de verificação de um *design* de *hardware* são essenciais para garantir sua qualidade, pois são feitos antes do envio do *design* para a fabricação em larga escala. Existem dois fatores que deixam claro o quão relevante é a verificação e como é importante realizá-la com êxito na primeira vez. [1]

O primeiro fator é que as atividades de verificação geralmente exigem mais desenvolvedores e um tempo de desenvolvimento maior, por isso é um dos processos mais caros na microeletrônica. Dessa forma, “quanto mais tempo gasto no ciclo de verificação, maior se torna o custo final” [2] do projeto.

O segundo fator é ilustrado na Figura 1, retirada de [1]. À medida que os projetos aumentam em complexidade e as janelas de mercado ficam menores, aumenta o custo da falha e, assim, “a verificação torna-se crucial” [1].

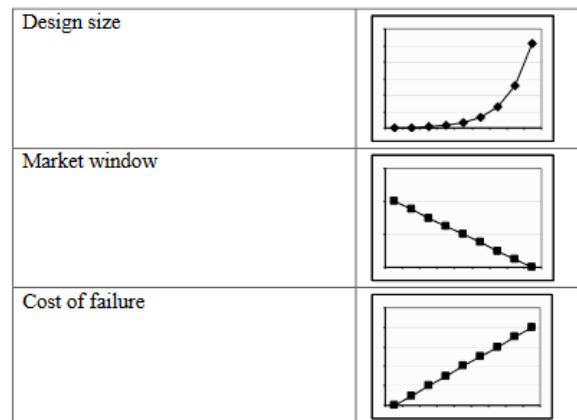


Figura 1 - Tamanho do design, janela de mercado e custo de falha.

Atualmente, os dispositivos entram diretamente no mercado consumidor. “Ao obter a verificação eficaz na primeira vez, as empresas podem economizar milhões de dólares apenas em possíveis *respins*” [1]. Além disso, ao atingir a janela de mercado no prazo do lançamento de um dispositivo, pode-se ganhar milhões a mais. [2]

Dentre as mais diversas formas de verificação de um *design* de *hardware*, a verificação funcional é a forma que, por meio da construção de um *testbench* implementado em SystemVerilog, tem como finalidade “garantir que o *design* implementa a funcionalidade pretendida”¹ [3].

Se a funcionalidade está totalmente expressa na especificação ou nos requisitos do dispositivo, ou se a intenção do produto vai além desses documentos, não é uma convenção unificada no mundo da engenharia de verificação funcional.¹

Neste trabalho, a definição de funcionalidade será considerada como “o que o dispositivo foi destinado a fazer, não necessariamente o que foi documentado para fazer” [1]. Os autores [1] afirmam que assim definem o conceito de funcionalidade porque “os primeiros defeitos encontrados são questões de especificação, não defeitos de *design*” [1].

Há aproximadamente duas décadas, processos de verificação dirigidos por métricas (*metric-driven*) foram desenvolvidos para evitar os possíveis defeitos de especificação das funcionalidades dos *designs*, assim como contornar os dois fatores anteriormente

¹ Versão original do inglês: “ensuring that the device does what it was intended to do”. (Capítulo 1: *The Verification Crisis*, página 3).

mencionados. Esses processos *metric-driven* são apresentados em formas de fluxos com etapas e objetivos bem definidos para possibilitar que os engenheiros de verificação funcional entreguem resultados com eficiência.

Usando processos baseados por métricas, critérios claros de verificação podem ser elicitados, possibilitando definir um fechamento para o ciclo de verificação, além de uma maior integração entre *designer* e verificador. Dessa forma, a qualidade da verificação é garantida através de resultados rastreáveis que são definidos em comum acordo com os projetistas do *design* a ser verificado.

É observado também que mesmo com a aplicação dos processos *metric-driven*, utilizando somente a estrita interpretação dos documentos de especificação do *design*, essas metodologias podem não alcançar seu potencial esperado. [1]

Contudo, [1] destaca que ao invés de processos dirigidos por métricas, ainda diversas empresas realizam as atividades de verificação funcional somente a partir das experiências anteriores, que determinam o fluxo de desenvolvimento adotado pelas equipes de verificação.

Atualmente, grande parte dos conteúdos das metodologias de verificação *metric-driven* estão disponibilizados em livros extensos, ou sob a compra de licença das empresas de software direcionados para o desenvolvimento de *hardware*, como por exemplo a Cadence® Design Systems [4], criadora do MDV [5].

1.1 Objetivo

Tomando como hipótese que um dos fatores que desincentivam a adesão às metodologias *metric-driven* por essas empresas seja a baixa disposição de conteúdos abertos que proponham fluxos claros e completos de trabalho, ao contrário da diversidade encontrada nos fluxos de gestão de projetos de software (como Scrum, Kanban e vários outros), este trabalho tem como objetivos:

- Construir um manual, com estrutura clara e concisa, para equipes ou engenheiros de verificação que desejam experimentar ou adotar em seu fluxo de trabalho os processos orientados por métricas definidos na metodologia *Metric-Driven Verification* (MDV) [4];
- Realizar a validação de conteúdo do manual por juízes e a validação semântica através de peritos.

2. FUNDAMENTAÇÃO TEÓRICA

O *Metric-Driven Verification* (MDV) é uma metodologia para verificação de *hardware*, fornecida pela Cadence® [4], que descreve processos dirigidos por métricas que permitem ao verificador planejar melhor seu trabalho, definindo exatamente o que precisa ser feito de maneira mensurável. [5]

A Cadence® [4] divide o processo MDV em 4 fases, como o apresentado na Figura 2, retirada de [1].

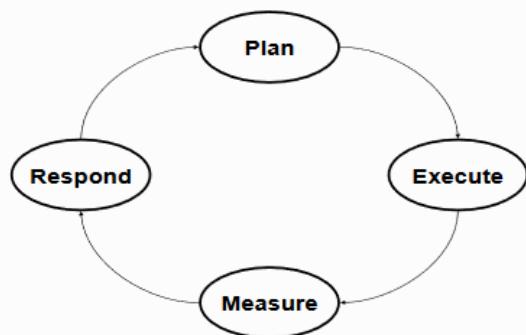


Figura 2 - As 4 fases do ciclo MDV

A primeira fase do MDV é o *Plan*, momento em que é escrito o plano de verificação (*Verification Plan - VP*), onde é determinado o que precisa ser feito e como mensurar que de fato foi feito aquilo que foi determinado [5]. O conteúdo de um VP descreve informações como: as *features* do *design*, a lista das classes de teste a serem implementadas, a cobertura de sinais de entrada, a checagem dos sinais de saída, entre outras informações do que o *testbench* deverá conter.

A fase de execução *Execute* é a segunda, após o planejamento, em que deve ser implementado o *testbench* de verificação de forma a captar as métricas definidas na fase anterior [5].

Na terceira fase, *Measure*, é o momento em que são executadas as regressões do *testbench* implementado e a coleta das métricas por meio das ferramentas MPA, tipicamente usadas nas atividades de verificação. Essas métricas incluem coberturas funcionais, de *RTL* e de *assertions*. [5]

A última e quarta fase, chamada de *Respond*, é a fase em que o verificador, tendo o resultado das regressões executadas na fase anterior, analisa e age sobre esses resultados, por exemplo; alterando a extensão de pontos da cobertura, removendo ou adicionando pontos a serem cobertos. Esta análise é usada para ajustar os planos da fase *Plan* e corrigir possíveis erros detectados no *testbench* ou no *RTL*. [5]

O Cadence® vManager™, é uma ferramenta MPA usada em grande escala pelas empresas de *design* de *hardware* (*design houses*) para executar as simulações das atividades de verificação, e “é totalmente baseada na metodologia MDV”[6].

Essas métricas captadas de forma automatizada, facilitadas pelo vManager [6], permitem responder com eficiência aos problemas da verificação à medida que eles surgem devido à estrutura de etapas definidas pelo MDV.

3. METODOLOGIA

Trata-se de um estudo descritivo com foco na construção e avaliação de estratégias metodológicas.

Foi desenvolvido de Novembro de 2021 a Março de 2022, com as seguintes etapas: levantamento bibliográfico; construção do manual em paralela validação do conteúdo pelos juízes e adequação do manual; validação semântica por engenheiros de verificação primeiros interessados no conteúdo.

3.1 Construção e Validação do Conteúdo

Inicialmente foi realizado o levantamento bibliográfico e a leitura dos conteúdos de [1], [2], [3], [5] e [6] referenciados neste trabalho, que explicam ou instruem sobre a aplicação da metodologia MDV.

A validação de conteúdo ou validação pelos juízes, é baseada no julgamento realizado por um grupo de juízes experientes na área ou interessados diretos no produto final, aos quais cabe analisar se o conteúdo está adequado ao que se propõe.

Todas as etapas descritas a seguir foram realizadas em paralela validação dos juízes, com frequência semanal de acompanhamento. Via chamada de vídeo, os produtos em desenvolvimento e as novas propostas eram apresentadas e então coletados os pareceres.

Para esta etapa foram convidados: o gerente técnico do time de verificação e o membro do time mais requisitado pelos outros integrantes para solução de problemas na área, ambos da empresa Idea Electronic Systems [7].

Foi analisada a estrutura do *Verification Plan - VP* utilizado pela equipe de verificação da empresa [7], que disponibilizou de forma aberta o template utilizado, em comparação com as requisições para aplicação do MDV na fase de planejamento.

Foram identificadas e corrigidas possíveis falhas, removidas ou acrescentadas partes na então estrutura vigente do *VP* de acordo com os requisitos da metodologia para aplicação da fase *Plan*. Em [8] é apresentado um comparativo das versões do *VP* antes e depois das modificações.

Finalizada a reestruturação do *VP*, o manual, intitulado “*HOW TO: Do a Verification Planning With MDV*”, disponível no Apêndice A, foi elaborado ligando os conteúdos bibliográficos levantados às habilidades e aos comportamentos esperados do engenheiro de verificação da empresa para o preenchimento de cada tópico do plano de verificação, assim, instruindo sobre a fase *Plan* da metodologia.

Dois outros documentos foram criados e anexados ao manual. O documento “*HOW TO: Start Verification Activities [Guideline]*” designado para fornecer instruções sobre uma possível sequência de atividades para a aplicação da fase *Execute*, e o documento “*HOW TO: Write a Coverage Report*” para a fase *Measure*, ambos construídos segundo o fluxo e padrão de trabalho da empresa [7]. A fase *Respond* não foi coberta pelo manual.

O manual e os documentos nele referenciados foram escritos em inglês, idioma oficial para registro e documentação na Idea Electronic Systems.

A elaboração das ilustrações foi baseada no referencial bibliográfico e nos principais assuntos abordados no manual. Foram desenhadas através de adaptações de imagens retiradas da internet em um editor de imagem.

A formatação e configuração das páginas foram realizadas por meio da ferramenta Confluence de documentação, por ser a ferramenta usada por [7] para atividades deste âmbito.

3.2 Validação Semântica

Para a validação semântica, foram determinados como peritos todos os membros do time de verificação da Idea Electronic Systems [7] e os engenheiros de verificação das empresas parceiras que também trabalham em união ao time de verificação, totalizando 19 peritos.

Os dados da avaliação semântica foram coletados através do envio de um *Google Form* [8] contendo 18 itens obrigatórios, separados em 5 grupos de avaliação: conteúdo, linguagem, *layout*, motivação e utilidade; composto por níveis de variação da escala Likert⁹: discordo totalmente, discordo, não concordo e nem discordo, concordo e concordo totalmente.

Os itens avaliados em cada grupo foram retirados a partir do método usado por [10] e estão listados na Tabela 1.

| Conteúdo | |
|-----------|------------------------------------|
| 1 | Apropriado ao público. |
| 2 | Atende às necessidades do público. |
| 3 | A sequência do texto é lógica. |
| 4 | Títulos e subtítulos pertinentes. |
| 5 | Trechos importantes em destaques. |
| Linguagem | |
| 6 | A escrita utilizada é atrativa. |
| 7 | A linguagem é clara e objetiva. |

| 8 | A redação é compatível com o público. |
|-----------|---|
| Layout | |
| 9 | A disposição do texto está bem organizada. |
| 10 | A extensão do texto é adequada. |
| 11 | As cores aplicadas auxiliam a leitura. |
| 12 | O tipo de letra facilita a leitura. |
| 13 | O tamanho das letras é adequado. |
| Motivação | |
| 14 | O conteúdo é motivador. |
| 15 | O conteúdo despertou interesse do leitor. |
| 16 | Atende diferentes perfis de profissionais. |
| Utilidade | |
| 17 | O conteúdo do manual é útil para compreender a estrutura do plano de verificação. |
| 18 | O conteúdo do manual facilita implementar o que é proposto no plano de verificação. |

Tabela 1 - Itens da avaliação semântica enviados aos peritos.

Os peritos selecionados receberam via e-mail o formulário, com o intuito de avaliar de que forma o material seria visto e entendido por seu público alvo.

No formulário, foi informado que a coleta das respostas é voluntária e de modo anônimo. Foi solicitado que lessem o manual, percorressem pelos demais documentos nele referenciados e registrassem suas sugestões no sentido de aperfeiçoá-lo. Uma caixa optativa de sugestões foi adicionada ao final, para possíveis recomendações de melhorias.

4. RESULTADOS E DISCUSSÕES

Do total de 19 peritos que receberam a solicitação de preenchimento do formulário via e-mail, foram obtidas 12 respostas e 7 ausências, não havendo nenhum preenchimento incompleto das questões.

Não houveram avaliações classificadas como *discordo totalmente* e nem como *discordo*.

Do grupo Conteúdo, o item 3 (“A sequência do texto é lógica.”) teve 100% de avaliação como *concordo totalmente*. Os itens 1, 4, e 5 obtiveram 67% das avaliações como *concordo totalmente*, enquanto o item 2 foi avaliado em 50% como *concordo totalmente* e 33% como *concordo*.

Para os itens do grupo Linguagem, o item 7 (“A linguagem é clara e objetiva.”) foi avaliado com 83% de total concordância. Os outros dois itens do grupo receberam uma avaliação distribuída aproximadamente em 50% entre *concordo totalmente* e *concordo*.

Para o Layout, os itens 9 e 13 também receberam uma avaliação distribuída aproximadamente em 50% entre *concordo totalmente* e *concordo*. Ambos os itens 11 e 12 foram avaliados em 66% como concordância total e o item 10 (“A extensão do texto é adequada.”) foram avaliados 58% como apenas *concordo*.

Dos itens do grupo Motivação, o item 16 (“Atende diferentes perfis de profissionais.”) teve 41% de concordância total e 33% de imparcialidade (nem concordo e nem discordo). Os demais itens do grupo foram avaliados em 66% de concordância total.

O grupo Utilidade teve em ambos os itens 33% de avaliação como concordo totalmente e uma avaliação entre 50% e 60% de apenas concordo.

Em todos os itens avaliados, manteve-se a proporção média de 67% de concordância total e 33% de concordância parcial.

Somente um dos peritos respondeu o item aberto opinativo, que encontra-se transcrito na Tabela 2.

| Sugestões |
|--|
| <i>“Integrar um material existente na base de dados da Cadence (MDV Foundation Planning Workshop.pdf) que sugere a utilização do vManager para fazer anotações das specs (e também a ligação dos mesmos com os parâmetros, os tests cases e a cobertura).”</i> |

Tabela 2 - Alterações sugeridas pelos peritos após avaliação semântica do manual.

Dados os resultados, pode-se considerar o manual como válido semanticamente e que pode ser utilizado como ferramenta para a aplicação dos requisitos propostos pela metodologia MDV [5] de verificação, tendo como concordância total média de 67%.

Apesar de válido, dado que o item 16 (“Atende diferentes perfis de profissionais.”) foi avaliado com 33% de imparcialidade, pode-se considerar que este é um quesito a ser revisto e melhorado dentro da estrutura do manual, já que trata-se de um item do grupo Motivação e pode interferir diretamente na sua aceitação por um público amplo.

5. CONCLUSÃO

Após o cumprimento das etapas metodológicas para a validação do manual sobre aplicação do *Metric-Driven Verification* [5], pode-se concluir que este tem um conteúdo válido sobre o assunto que aborda. Conclui-se também que o manual é considerado válido semanticamente, podendo ser utilizado como ferramenta para a aplicação dos requisitos propostos por metodologia MDV de verificação, tendo como concordância total média de 67%.

O fato de que o conteúdo foi desenvolvido tendo como base um fluxo de trabalho, até então não disponibilizado ao público, e criado a partir das experiências de uma empresa específica, ameaça a sua universalidade.

Os possíveis obstáculos são a necessidade de que para adotá-lo em completude, é preciso também adotar a estrutura do plano de verificação oferecida em conjunto ao manual, ou realizar a interpretação e as adaptações que venham a ser necessárias para a utilização do manual em uma gestão das atividades de verificação que seja diferente da gestão propostas.

Apesar dos possíveis impactos, de modo geral, o manual e a sua aplicação podem contribuir para que os engenheiros ou equipes de verificação, que não possuem uma estrutura bem definida para elaboração do plano de verificação ou que buscam aprimorar as estruturas já utilizadas, possam conhecer e adotar as metodologias dirigidas por métricas, em especial o MDV.

O maior benefício oferecido pelo uso do manual é a possibilidade de pôr em prática a mentalidade e a forma de planejamento esperada dos engenheiros de verificação que utilizam a ferramenta vManager [6], dado que é uma das ferramentas MPA mais utilizadas na área.

Uma sugestão de pesquisa proposta, que é também a sugestão coletada na avaliação dos peritos, é o estudo continuado das extensões da ferramenta vManager para a integração da metodologia MDV e a complementação deste conteúdo à estrutura do plano de verificação e ao manual.

6. REFERENCES

- [1] CARTER, Hamilton. B. HEMMADY, Shankar G. *Metric Driven Design Verification: an engineer's and executive's guide to first pass success*. New York, NY : Springer, 2007.
- [2] BRENNAN, John. *Cadence MDV Highlights with vManager*. Cadence. https://www.cadence.com/en_US/home/tools/system-design-and-verification/system-design-and-verification-flows/metric-driven-verification-signoff.html#
- [3] BERGERON, Janick. *Writing Testbenches: Functional Verification of HDL Models*. 2nd. ed. New York, NY: Springer, 2003.
- [4] Cadence Design Systems. https://www.cadence.com/ko_KR/home.html
- [5] Metric-Driven Verification Signoff. Cadence Design Systems. https://www.cadence.com/en_US/home/tools/system-design-and-verification/system-design-and-verification-flows/metric-driven-verification-signoff.html
- [6] vManager Verification Management. https://www.cadence.com/ko_KR/home/tools/system-design-and-verification/planning-and-management/vmanager-solution.html
- [7] Idea Electronic Systems. <https://www.idea-ip.com/>
- [8] Site repositório para o trabalho. <https://sites.google.com/ccc.ufcg.edu.br/mdv-idea/manual>
- [9] Jean Pierre LUDWIG; Ederson Benetti FAIZ; Rafael Bazzei PALOSCHI; José de SOUZA. Planejamento estratégico: análise de eficácia da metodologia aplicada por meio da Escala Likert. *Espacios*. Vol. 36 (Nº 16) Ano 2015. Pág. 9
- [10] Ribeiro, L.C.C., Oliveira, T.C., Moreira, S., de Paula. F.A. *Construção e Validação de Manual sobre Burnout em Professores*. *Revista de Enfermagem do Centro-Oeste Mineiro*. 2017.

About the authors:

Anne Silva é bacharelanda em Ciências da Computação na Universidade Federal de Campina Grande (UFCG). Há dois anos é integrante do time de verificação da Idea Electronic Systems como analista júnior, realizando a verificação de *IP designs* e captando bibliografias para a melhoria do processo de verificação do time. Anteriormente, foi aluna de capacitação em microeletrônica na UFCG, onde trabalhou pelo Virtus em parceria com a Idea na liderança da equipe de verificação por 1 ano.

Elmar Melcher Possui doutorado pela Ecole Nationale Supérieure des Telecommunications em Eletrônica e Comunicações e graduação em Elektrotechnik pela Technische Universitaet Stuttgart. No Brasil desde 1993, atualmente desenvolve pesquisas no Laboratório de Arquitetura Dedicadas - LAD da UFCG, tendo coordenado o projeto do decodificador de vídeo MPEG-4, no âmbito da Rede Brazil IP, vencedor, em 2006, do prêmio Best Design IP/SoC, em conferência internacional realizada na França. É professor associado da Universidade Federal de Campina Grande.

APÊNDICE A - Manual para Aplicação da Metodologia MDV

HOW TO: Do a Verification Planning With MDV

Introduction

i This document will guide you in the *elaboration of your verification plan* using the **guidelines** available on Cadence®'s various platforms about the **Metric-Driven Verification (MDV) methodology**.

Table of Contents



- What is the Metric-Driven Verification
 - ▲ The Verification Process
- Verification Planning
 - 1. Features
 - 2. Parameters
 - 3. Tests
- Scheduling a VP Review
 - References

What is the Metric-Driven Verification



The Cadence® **vManager™**, which is the MPA tool used in our verification activities, “**is completely based on the Metric Driven Verification methodology**”[1].

The MDV is a metric-driven process that ables the verifier to better plan his work defining exactly what needs to be done in a manner that’s measurable. [2]

This automated measurements that vManager facilitates, will allow to efficiently respond to issues as they arise.

Cadence® divides the MDV process into 4 phases, as shown in the figure below:

Figure 1 - The four phases of the MDV cycle



Source: Authors' compilation¹.

Based on each feature of the verification plan, incentives are applied, and finally the metric of each feature is collected, back-annotated to the verification plan, and the visualized result is obtained to determine whether the verification is complete. [3]

In the next section is presented a summary of MDV 4 phases cycle.

▲ The Verification Process

Figure 2 - First phase of MDV



Source: Authors' compilation¹.

The first step of the verification is PLANNING. [3]

This is where you determine:

- *What needs to be done and;*
- *How to measure that it was in fact done.*

 **That is the main goal of this guide.**

Figure 3 - Second phase of MDV



Source: Authors' compilation¹.

Once the verification is planned, you need to make it happen.

That brings you to the **EXECUTION** phase of the verification. [3]

During this phase you will:

- *Generate the verification environment and;*
- *Create on it the objective metrics outputs you specified during planning.*

Check here [HOW TO: Start Verification Activities \[Guideline\]](#) .

Figure 4 - Third phase of MDV



Source: Authors' compilation¹.

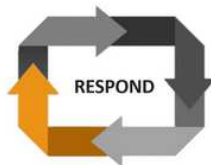
As you engines operate you need to **MEASURE the effectiveness of your efforts.** [3]

During the planning you specified the metrics that will be used to gauge this effectiveness, now, you need to gather this metrics from our execution engines (like vManager). Some typical metrics include:

- *RTL code coverage;*
- *Functional coverage;*
- *Assertion coverage.*

Check here [HOW TO: Write a Coverage Report](#) .

Figure 5 - Last phase of MDV



Source: Authors' compilation¹.

In the **RESPONSE phase, the vericator acts on the results** of the data analysis performed during the measure phase. [3]

This analysis will be used to adjust existing plans and to facilitate other verification processes.

¹ Assembly from an image collected on the internet.

Verification Planning



The Verification Planning is where the MDV cycle begins and it is the **main content of this guide**.



This chapter was written by adapting Cadence's guidelines for the filling of our workflow format verification plan .

“Verification concerns that are raised late in the project are one of the main causes of schedule slips.” [3]

To avoid this, the **planning technique** used in MDV focuses on determine what needs to be verified and what metrics will be measured and has as **result a Verification Plan** [2], which is explained in this sections.

“Verification planning is a **solo brainstorming** performed by the verifier” [3]. The process begins with the **reading** and **studing** of the device's under verification block guide (or any documentation that provides the device's functionalities explanation).



This chapter will contain the following sections:

- **1. Features:** *Identifying device features;*
- **2. Parameters:** *Identifying device features attributes;*
- **3. Tests:** *Identify what features exercise in each configuration;*

1. Features



The main output of a planning session will be a **feature-oriented Verification Plan (VP)** [4]. To this end it is the job of the vericator to understand the spectations of each feature of the given device.

i What is a feature ?

A feature is something that the device under verification does. A feature is a **VERB** or **ACTIVITY**. [3]

Examples of features are:

- A device **provides** data to a read request.
- A device **consumes** data from a write request.
- A device **routes** a transaction from one port to another.
- A device **decodes** a MP3 stream and **provides** audio output.

*This section **goal** is to list all the device's features.*

► Questions during this stage might be:

The vericator must aks him self questions about the features of the design, regarding:

- "How the device is **accessed**?" [3]
- "How the device **outputs data**?" [3]

► Step by step

| 1. IDENTIFY THE FEATURES | 2. DESCRIBE THE FEATURES | 3. IDENTIFY THE METRICS |
|---|---|--|
| Gather all information/references about the device and divide it in sections based on what the design does ; | Write a feature description using verbs or activities ; | Describe the metrics that will guarantee that the feature was objectively and automatically measured. |

► Practical example:

RS Encoder Features

► Hands on:

[Guideline] Features Table

2. Parameters



Now that you have the devices **features** listed, you can create the definition of the stimuli that your testbench will need. [4]

What is a parameter ?

A parameter can be a **configuration signal**, a **interface input signal**, or a **group of them**, that need to be generated following a set of **constraints** respective to a feature.

Examples of parameters are:

- A device **receives** a **reset signal** in every each two hundred messages processed.
- A device **consumes** **input data singal** of 2480 bits lenth.

This section **goal** is to determine what interface fuctionality need to be exercised.

► Questions during this stage might be:

- "What protocol is used to **configure the device**?" [3]
- "Is the entire protocol used, or are there **transactions types** that the device legally ignores such as **burst transactions**?" [3]

► Step by step

| 1. IDENTIFY THE PARAMETERS | 2. DESCRIBE THE COSNTRAINTS | 3. DETERMINE THE COVERAGE |
|---|---|--|
| Looking at each feature, gather all information about the device's interface input signals and divide them in parameters based on the constraints that each one may have; | For each parameter identified, describe the behavior of the signal(s) and the restrictions for its generation . Aspects of the signal(s) such as: size, rate and external influences must be well described. | Describe the measures that will be needed to capture those constraints, ansewring: <ul style="list-style-type: none">• "What is important to cover about this parameter?"• "How do you know when this parameter has been utilized so that can be covered?"• "How do you cover this parameter?" |

► Practical example:

RS Encoder Parameters

► Hands on:

[Guideline] Parameters Table

3. Tests



Once the basic device **features** and the **parameters** to exercise them are established, it is time to describe the verification environment's tests.

The verifier will focus on **what the design does** in each feature, especially observing its **corner cases**. [4]

i What a good verification test description must have?

Your tests must be written in a way that generates specific contexts of device functioning, and this context must exercise one or several features.

The number of tests to be implemented is the one that covers all the features described. [3]

This section **goal** is to determine how the featured will be simulated and checked.

► Questions such as the following must arise:

- "What is the correct behavior for the device if the serial input port is driven before the device is configured?" [3]
- "What happens to an output transaction if a reset is asserted before the transaction is complete?" [3]

As this process moves from feature to feature it is important to weigh each Test according to its significance, because it is that weight distribution that guides the priorities of each verification activity. [4]

► Step by step

| 1. IDENTIFY THE TEST'S GOAL | 2. IDENTIFY THE TEST'S TARGET & STIMULI |
|--|---|
| Once again, moving from feature to feature, describe the execution context that the environment must simulate for the feature to be exercised in an ideal or corner case. That context will be your Test Goal description. | For each test goal, identify which output signals will be the target of this test context and thus determine which parameters will be used as input stimuli |

► Practical example:

[RS Encoder Tests](#)

► Hands on:

[\[Guideline\] Tests Table](#)

Scheduling a VP Review



Now that you have the first version of your Verification Plan, **it is time to schedule its review!**

As mentioned above, the planning phase is the core of MDV methodology. The value offered by the resulting plan is directly proportional to the number of stakeholders that actually attend the review session.

So, when you have finished planning, you need to communicate your team manager and send an email to:

- ✉ **Verification project manager;**
- ✉ **RTL designer;**
- ✉ **A third stakeholder that can give impartial opinion on the review.**

With the meeting invite sent, in the review you will present your verification plan and collect all possible feedback to further improve your verification.

References

-
- [1] vManager Verification Management. Cadence Design Systems. [vManager Verification Management](#)
 - [2] Metric-Driven Verification Signoff. Cadence Design Systems. [Metric-Driven Verification Signoff](#)
 - [3] CARTER, Hamilton. B. HEMMADY, Shankar G. Metric Driven Design Verification: an engineer's and executive's guide to first pass success. New York, NY : Springer, 2007.
 - [4] BRENNAN, John. Cadence MDV Highlights with vManager. Cadence. [Metric-Driven Verification Signoff](#)