**Cobertura utilizando Assertion: um breve resumo de sua aplicação e benefícios.**

Erica Clementino de Carvalho [1], Jackson Platiny Soares Leite [2]

erica.clementino@academico.ifpb.edu.br. [1] [jackson.platiny@academico.ifpb.edu.br](mailto:pacheco.franklin9@gmail.com) [2]

**Resumo**

É notável que a utilização das tecnologias de VLSI e o aumento de demanda por produtos eletrônicos em todo o mundo, faz com que a complexidade dos circuitos neles se torne cada vez mais densa, e em contrapartida, o tempo de mercado desses produtos assim como o tempo de projeto para o desenvolvimento, produção e venda dos mesmos se torne cada vez mais escasso. A verificação funcional é a etapa do projeto desses circuitos onde é testado se o design dele está bem estruturado, e se sabe que um bom design deve apresentar as coberturas de código, funcional e de asserção. A cobertura de asserção é de extrema importância nesse processo, pois a asserção auxilia o programador no desenvolvimento, depuração do código e identificação de erros que não deveriam acontecer, sendo a primeira linha de defesa contra bugs. Nas linguagens de descrição de Hardware, uma asserção é uma declaração condicional que verifica um comportamento específico, e caso ele ocorra, é exibido uma mensagem. A cobertura utilizando asserções traz como benefícios para o projeto uma maior segurança no seu desenvolvimento devido a sua função, assim como ele atende as demandas geradas pelo mundo moderno, mas principalmente, ele auxilia na fase de verificação funcional contínua, permitindo que os produtos desenvolvidos possam continuar sendo verificados e analisados mesmo que já estejam no mercado, pois abre espaço para que a solução de possíveis problemas encontrados após seu lançamento no mercado possam ser solucionados via *software*, de forma a evitar grandes prejuízos financeiros as empresas destes produtos.

**Palavras-chave**: Cobertura de Asserção. *Very Large Scale Integration*. Circuitos Eletrônicos. Verificação Funcional. Sistemas Digitais.

***Abstract***

*It is remarkable that the use of VLSI technologies and the increasing demand for electronic products around the world, makes the complexity of the circuits in them become increasingly dense, and in return, the time to market of these products as well as the Project time for the development, production and sale of them becomes increasingly scarce. Functional verification is the stage in the design of these circuits where it is tested that its design is well structured, and it is known that a good design must have code, functional, and assertion coverage. Assertion coverage is extremely important in this process, as the assertion assists the programmer in development, code debugging, and identifying bugs that shouldn't happen, being the first line of defense against bugs. In Hardware description languages, an assertion is a conditional statement that checks for a specific behavior, and if it occurs, a message is displayed. The coverage using assertions brings as benefits to the project a greater security in its development due to its function, as well as it meets the demands generated by the modern world, but mainly, it helps in the continuous functional verification phase, allowing the developed products to be able to continue to be checked and analyzed even if they are already on the market, as it makes room for the solution of possible problems found after their launch on the market to be solved via software, in order to avoid major financial losses for the companies who have these products.*

***Keywords:*** *Assertion Coverage. Very Large Scale Integration. Electronic Circuits. Functional Verification. Digital Systems.*

**1 Introdução**

É notável que a utilização das tecnologias de *Very Large Scale Integration* ou VLSI e o crescimento constante de demanda por produtos eletrônicos em todo o mundo, cada vez mais faz com que os circuitos eletrônicos que as utilizam exijam níveis de complexidade cada vez maiores, e como consequência direta disso, o tempo de mercado dos produtos, assim como o tempo de projeto para o desenvolvimento, produção e venda deles, estão cada vez menores.

Tendo em vista o apresentado, é necessário que existam meios de garantir que o produto chegue ao mercado em perfeitas condições, de forma a garantir a integridade de seu funcionamento e evitar falhas e erros de desenvolvimento ou produção que possam gerar prejuízos exorbitantes as empresas fabricantes e ao “poder” de seus nomes no mercado. Um desses meios de testar o produto antes de manda-lo ao mercado e evitar que tais problemas ocorram é a etapa de Verificação Funcional, onde nesta, é possível e recomendado que se faça o uso de asserções para detectar problemas no design, a fim de evitar os problemas comentados.

Um bom design deve conter a tríade de cobertura: a cobertura de código, a cobertura funcional e a cobertura de asserção. A correta utilização delas introduz estabilidade ao design. Neste artigo, iremos abordar, com base na bibliografia pesquisada, a cobertura de asserção e sua importância para a verificação funcional e para que o produto seja lançado da melhor maneira possível no mercado.

**2 Referencial teórico**

Segundo Camara (2011), o projeto de hardware necessita seguir algumas métricas para que o tempo de verificação funcional dele seja minimizado, como por exemplo, agilidade, cobertura e o uso de *Assertions*. Agilidade é a métrica que irá mostrar o quão rápido e eficiente é o desenvolvimento do ambiente de verificação funcional, já a métrica de cobertura é utilizada para conferir se todos os prováveis valores de um conjunto foram corretamente estimulados, e finalizando a simulação quando objetivo de cobertura enfim for alcançado. O uso de *Assertions*, que é o foco principal deste artigo, tem como objetivo inserir condições dadas como verdadeiras pelo engenheiro de verificação responsável em um determinado ponto do *testbench*, para que com o auxílio destes, as falhas de protocolo e controle possam ser devidamente diagnósticas com mais facilidade e agilidade.

Para Silva (2007), *Assertions* são a descrição do comportamento esperado quando o projeto é estimulado com uma entrada, e normalmente são chamados de *checker* ou monitor no projeto. Para verificar uma característica do DUV (*Design Under Verification*), a sua funcionalidade deve ser primeiramente entendida, após ela precisa ser explicitamente descrita, e finalmente, as condições sob as quais o comportamento é colocado precisam ser estabelecidas, e isso é implementado através de uma *Assertion*. A cobertura funcional é baseada na funcionalidade do projeto, e ao utilizar *Assertions* como ponto inicial da cobertura funcional, pode-se descrever cenários funcionais mais elaborados, para capturar todas as combinações de dados e controle.

De acordo com Damasceno (2005), apesar da adição de *Assertions* no circuito durante a fase de verificação a tornar mais eficiente, ainda não há como garantir que todos os erros de projeto sejam encontrados antes de sua comercialização, o que principalmente se dá pelo fato de ser praticamente impossível que consiga prever todas as possibilidades de uso do circuito durante esta fase. Como não há uma forma de garantir que um circuito seja comercializado totalmente sem erros de projeto, é possível fazer um monitoramento na fase posterior de sua comercialização, utilizando justamente conceitos relacionados a *Assertions*, fazendo com que a fase de verificação e validação se torne algo permanente e contínuo, o que facilita a identificação de erros devido ao maior número de possibilidades de uso do circuito estar sendo utilizada, onde está fase posterior a comercialização pode ser chamada de fase de verificação contínua.

**3 Definição e Aplicação**

Podemos dizer que *Assertions* auxiliam o programador no desenvolvimento, depuração do código e identificação de erros que não deveriam acontecer, sendo a primeira linha de defesa contra bugs. Nas linguagens de descrição de Hardware (HDL), um *assertion* é uma declaração condicional que verifica um comportamento específico, e caso ele ocorra, é exibido uma mensagem. Geralmente é utilizado como monitores em busca de um comportamento não esperado, mas também pode ser utilizado para criar alertas de um comportamento desejado, ou seja, um *assertion* é uma característica ou propriedade funcional específica que se espera que seja válida para um design.

No método de verificação baseada em *assertion* (ABV), os engenheiros de projeto usam *assertion* para capturar a intenção de um projeto específico e verificar se o projeto implementa corretamente a intenção por meio de simulação, verificação ou simulação dessas assertivas.

Com a linguagem de declaração amplamente disponível e as ferramentas de suporte, os projetistas e engenheiros de verificação adotaram o método ABV para melhorar a qualidade do projeto e a eficiência da verificação. Alguns dos benefícios de sua utilização são:

* Garantir o comportamento formal correto do projeto;
* Aumentar a detecção das fontes de erros;
* Reduzir o tempo de observação e depuração;
* Simular e verificar formalmente, visando garantir um alto retorno sobre o investimento (ROI).

**4 Conclusão/Considerações**

Com base no absorvido após a análise da bibliografia pesquisada, é possível concluir que a utilização de *Assertions* em um projeto, especialmente seu uso na cobertura, traz diversos benefícios, tais como uma maior segurança no desenvolvimento do projeto devido a sua função, assim como também sua utilização atende as demandas geradas pelo mundo moderno, onde cada vez mais os produtos precisam ser pensados, desenvolvidos e postos no mercado para melhor aproveitar o seu tempo hábil de mercado.

Além dos benefícios apontados acima, ele também auxilia na chamada fase de verificação funcional contínua, que se dá após a inserção do produto no mercado, permitindo que através das técnicas e tecnologias que a sua utilização propõe, os produtos desenvolvidos possam continuar sendo verificados e analisados mesmo que já estejam no mercado, e em caso de detecção de problemas, falhas ou erros, facilita a criação de possíveis resoluções para aquela complicação, podendo ela ser feita até mesmo na forma de *Software*, por meio de uma atualização ou *patch* novo lançado para aquele produto.

**Referências**

[1] CAMARA, RÔMULO C P. **OVM-tpi:** **Uma Metodologia de Verificação Funcional Para Circuitos Digitais**. Tese de mestrado, UFPE, 2011. Disponível em: <https://repositorio.ufpe.br/bitstream/123456789/2448/1/arquivo3452_1.pdf>. Acessado em: 03/12/21.

[2] SILVA, KARINA R G. **Uma Metodologia de verificação Funcional para Circuitos Digitais**. Tese de doutorado, UFCG, 2007. Disponível em: <https://lad.dsc.ufcg.edu.br/lad/uploads/Lad/tese_karina.pdf>. Acessado em: 05/12/21.

[3] DAMASCENO, FABRÍCIO O. **Utilizando SNMP para Asserções em Hardware**. Tese de mestrado, UFMG, 2005. Disponível em: <https://repositorio.ufmg.br/bitstream/1843/SLBS-6GVF6Y/1/fabricio_damasceno.pdf>. Acessado em: 04/12/21.