



Plano de Verificação Funcional

MUSA

Fazemos Qualquer Negócio Inc.

Compilação 1.0

Histórico de Revisões

Data	Descrição	Autor(es)
23/10/2014	Criação do documento.	Terseu Hunter
24/11/2014	Refatoração do documento	Anderson Queiroz e Manuelle Macedo

CONTENTS

1. Introdução

1.1. Propósito do Documento

O objetivo deste documento é definir o plano de verificação da implementação MUSA. Este documento inclui o ambiente de verificação utilizado para realizar a verificação do processador, ao lado das principais características do design, a lista de testes, lista de *assertions* e outros.

1.2. Stakeholders

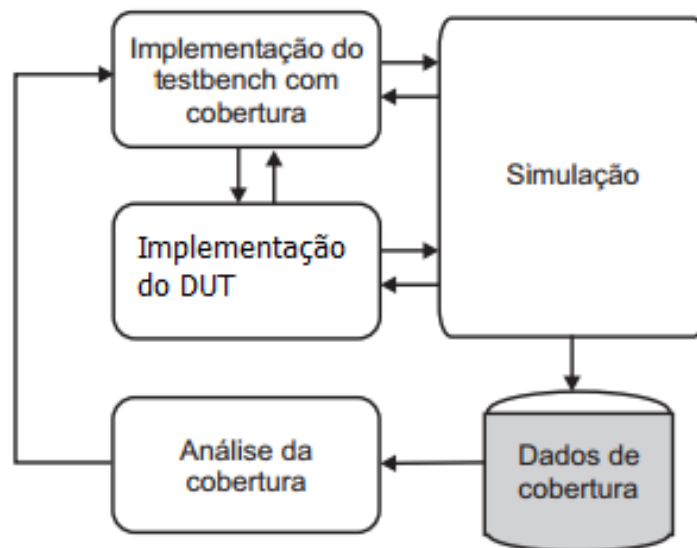
Nome	Papéis/Responsabilidades
Anderson, Manuelle e Weverson	Verificação Funcional
Gabriel, Lucas, Mirela, Patrick, Vinicius e Tarles	Implementação
Anderson e Manuelle	Analise

1.3. Siglas e Abreviações

Sigla	Descrição
DUT	Design Under Test
IF	Interface

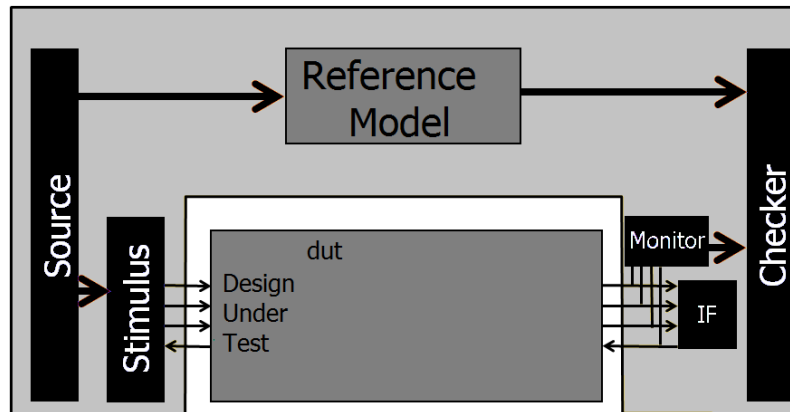
2. Visão Geral do DUT

- Implementação de um plano de verificação.
- Implementação de todos os testbenches necessários para o DUT completo e os DUTs resultantes de hierarquização.
- Implementação do DUT. Essa implementação é seguida pela simulação do DUT, juntamente com o testbench.
- Captação dos dados de simulação através da coleta dos itens de cobertura e dos logs da simulação.
- Análise da cobertura funcional, que pode levar a uma mudança de estímulos no testbench e a uma nova simulação.



3. Ambiente de Verificação

A metodologia de verificação adotada pelo projeto é baseada em *testbench*, compondo parte das análises por meio de verificação baseada em *waveform*. Situações especiais serão verificadas a partir de verificações baseadas em *assertions*. A interface do DUT será responsável por coletar os dados do MUSA e enviá-los para o *monitor*, no qual estarão declarados todos os *assertions*. A Figura abaixo apresenta um modelo conceitual do ambiente de verificação.



3.1. Design Under Test Interface

O DUT IF promove a interface entre o *monitor* e o DUT. Esta interface é responsável por controlar as informações trocadas entre o ambiente de verificação e o DUT. Dessa forma, ela deve conter instâncias de todos os sinais do DUT a serem utilizados ao longo do processo de verificação.

A interface do DUT possui também a implementação dos *assertions*. Estas estruturas têm como objetivo garantir que o comportamento dos sinais internos do DUT estão sendo produzidos e manipulados de maneira correta. Esta interface é instanciada na entidade *top level* do ambiente de verificação e seus sinais são conectados aos sinais provenientes do DUT.

3.2. Monitor e Checker

O *monitor* é responsável por observar o comportamento do DUT e coletar as suas saídas, de modo a verificar se as instruções estão funcionando da maneira desejada. O *monitor* observa o comportamento dos sinais de controle e, quando necessário, captura os dados armazenados na memória de instruções e no banco de registradores.

O *checker* é responsável por executar o modelo de referência com o mesmo programa usado pelo DUT e comparar os dados armazenados na memória de dados e no banco de registradores. Se qualquer mal funcionamento for identificado, o *checker* deve reportar uma mensagem de erro.

Quando a execução do programa chega ao fim, o monitor deve invocar o *checker*. O *monitor* identifica o final da execução do programa a partir da instrução HALT.

O teste que será executado no modelo de referência deve ser definido no arquivo `sim/tb/defines.sv`. Para executar o teste no DUT, o procedimento deve ser realizado no arquivo de memória de instruções `sim/tests/instrucoesMars.asm`, a partir da alteração do caminho especificado na função `read_memh`.

3.3. Modelo de Referência

Tendo em vista garantir que o processador executará as instruções corretamente, foi desenvolvido um modelo de referência, capaz de simular o comportamento do processador MUSA. Este modelo é capaz de executar todas as instruções suportadas pelo MUSA. O arquivo do modelo de referência está localizado no diretório `sim/model/main2.c`.

3.4. Especificações de Projeto do Ambiente de Verificação

Componente Descrição	
Nome do Documento	Plano de Verificação do MUSA
Versão e data do documento	Versão 1.0, 23 de outubro de 2014
Autor(es) / Proprietário(s)	Terseu Hunter
Metodologia de Verificação	Top-Down
Métodos de Verificação	Simulation and Formal Verification
Aplicação	ModelSim ALTERA Edition
Linguagens	System Verilog
Ambiente de verificação	Custom testbench
Arquivos de teste	No diretório: <code>sim/tests</code>
Tecnologias	FPGA Cyclone 3 Development Board

4. Lista de Funcionalidades

Feature Número	Feature Descrição	Prioridade
MUSA_F1	Signal are activated based on the instruction.	10
MUSA_F2	Communication with Instruction Memory	9
MUSA_F3	Read and write operation to the Data Memory.	9
MUSA_F4	Read and write operation to the Register File.	10
MUSA_F5	All interfaces protocols must work properly.	9

5. Lista de Testes

Número do Teste	Descrição	Método	Nível	Funcionalidade Verificadas	Prioridade	Proprietário	Situação
MUSA_T1	Execução de todas as instruções da categoria aritmética.	Sim	Unit	MUSA_F1, MUSA_F4	5	Anderson	70%
MUSA_T2	Execução de todas as instruções de transferência de dados.	Sim	Unit	MUSA_F1, MUSA_F4	5	Anderson	70%
MUSA_T3	Execução de todas as instruções da categoria lógica.	Sim	Unit	MUSA_F1, MUSA_F4	5	Anderson	70%
MUSA_T4	Execução de todas as instruções da categoria salto condicional.	Sim	Unit	MUSA_F1, MUSA_F4	5	Anderson	70%
MUSA_T5	Execução de todas as instruções da categoria salto incondicional.	Sim	Unit	MUSA_F1, MUSA_F2	5	Anderson	70%
MUSA_T6	Acesso à memória de declarados	Assertion	Unit	MUSA_F3	7	Manuelle e Weverson	80%
MUSA_T7	Acesso à memória de instruções	Assertion	Unit	MUSA_F4	9	Manuelle e Weverson	80%
continua na próxima página							

continuação da página anterior							
Número do Teste	Descrição	Método	Nível	Funcionalidade Verificadas	Prioridade	Proprietário	Situação
MUSA_T8	Execução de programas completos sob a arquitetura.	Sim	Unit	MUSA_F3, MUSA_F4	8	Anderson, Manuelle e Weverson	0%
MUSA_T9	Teste de todos os protocolos de interface.	Assertion	Unit	MUSA_F5	8	Anderson, Manuelle e Weverson	0%

6. Assertions

Número	Critério	Status
MUSA_A1	Assertion para a busca correta das instruções.	Em andamento
MUSA_A2	Assertion para verificar a operação de decodificação	Em andamento
MUSA_A3	Assertion para verificar a operação do bloco de execução.	Em andamento
MUSA_A4	Assertion para leitura da memória de dados e write back.	Em andamento
MUSA_A5	Assertion para branches e instruções de salto.	Em andamento
MUSA_A6	Assertion para verificar os protocolos de interface.	Em andamento

7. Recursos Requirements

Recursos	Quantidade	Descrição	Início	Duração
Recursos de Engenharia				
Engenheiro de Verificação	3	Alunos	10/11	15 dias
Recursos Computacionais				
Computador	3	Intel i5	N.A.	N.A.
Recursos de Software				
ALTERA Quartus	1	WEB Edition	10/11	21 dias
ALTERA ModelSIM	1	ALTERA WEB Edition	14/11	21 dias

8. Cronograma

Recursos	Início	Duração	Ação	Recursos
TBD	TBD	TBD dias	Definam as tarefas nesta tabela :)	N/A