Tutorial verificação comportamental

usando Quartus e ModelSim

Oscar Eduardo Anacona

7 de setembro de 2018

Problema proposto

VHDL

Escrever em VHDL a descrição de uma ULA (Unidade Lógica Aritmética) usando componentes como é apresentado na Figura 1.

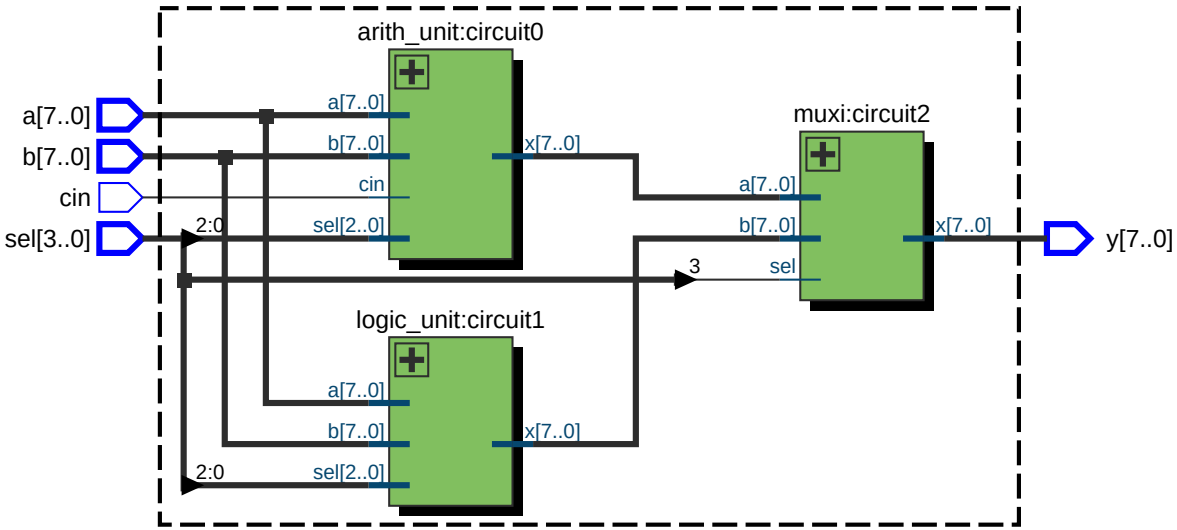

Figura 1: ULA

Problema proposto

VHDL

Escrever em VHDL a descrição de uma ULA (Unidade Lógica Aritmética) usando componentes como é apresentado na Figura 1.

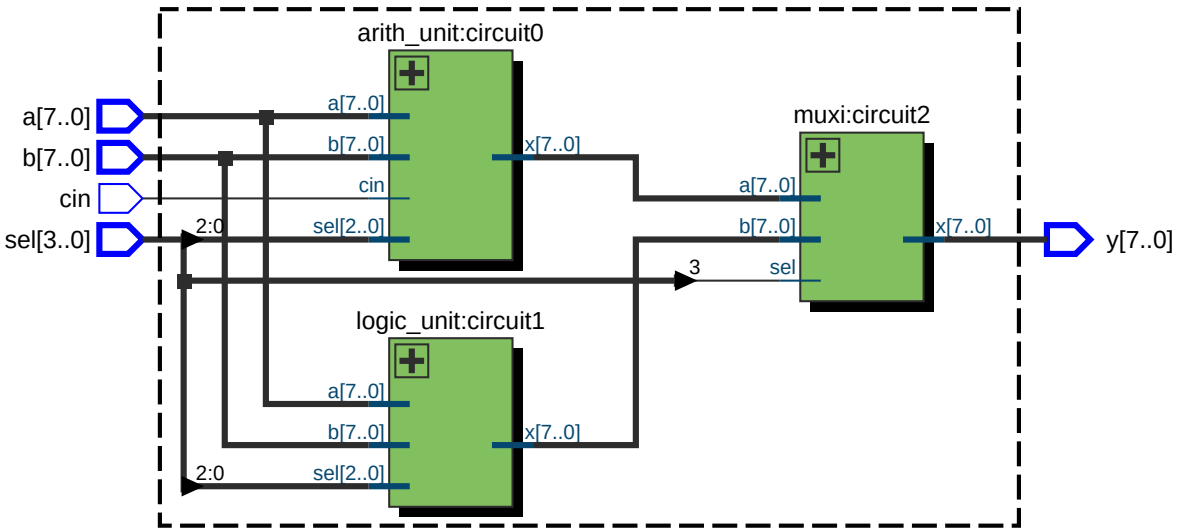

Figura 1: ULA

Verificação comportamental em ModelSim

Simular a arquitetura da ULA implementa em VHDL.

Problema proposto

Unidade aritmética

Especificações da unidade aritmética.

Sel	Operation	Function	Unit
0000	y<=a	Transfer a	Arithmetic
0001	y<=a+1	Increment a	
0010	y<=a-1	Decrement a	
0011	y<=b	Transfer b	
0100	y<=b+1	Increment b	
0101	y<=b-1	Decrement b	
0110	y<=a+b	Add a and b	
0111	y<=a+b+cin	Add a and b with carry	

Problema proposto

Unidade lógica

Especificações da unidade lógica.

Sel	Operation	Function	Unit
1000	y<=NOT a	Complement a	Logic
1001	y<=NOT b	Complement b	
1010	y<=a AND b	AND	
1011	y<=a OR b	OR	
1100	y<= a NAND b	NAND	
1101	y<=a NOR b	NOR	
1110	y<=a XOR b	XOR	
1111	y<=a XNOR b	XNOR	

Roteiro

Roteiro

Criação de um projeto no Quartus

Seguir os passos básicos:

- Criar uma nova pasta chamada ALU onde o projeto do Quartus será salvado.
- Criar um novo projeto dentro do Quartus.
- Colocar o endereço da pasta nomeada ALU.
- Em EDA Tool Settings, na opção simulation escolher ModelSim-Altera e o formato VHDL.
- Fim da criação do novo projeto.

Criação de um projeto no Quartus

Seguir os passos básicos:

- Criar uma nova pasta chamada ALU onde o projeto do Quartus será salvado.
- Criar um novo projeto dentro do Quartus.
- Colocar o endereço da pasta nomeada ALU.
- Em EDA Tool Settings, na opção simulation escolher ModelSim-Altera e o formato VHDL.
- Fim da criação do novo projeto.

Criação de um projeto no Quartus

Seguir os passos básicos:

- Criar uma nova pasta chamada ALU onde o projeto do Quartus será salvado.
- Criar um novo projeto dentro do Quartus.
- Colocar o endereço da pasta nomeada ALU.
- Em EDA Tool Settings, na opção simulation escolher ModelSim-Altera e o formato VHDL.
- Fim da criação do novo projeto.

Criação de um projeto no Quartus

Seguir os passos básicos:

- Criar uma nova pasta chamada ALU onde o projeto do Quartus será salvado.
- Criar um novo projeto dentro do Quartus.
- Colocar o endereço da pasta nomeada ALU.
- Em EDA Tool Settings, na opção simulation escolher ModelSim-Altera e o formato VHDL.
- Fim da criação do novo projeto.

Criação de um projeto no Quartus

Seguir os passos básicos:

- Criar uma nova pasta chamada ALU onde o projeto do Quartus será salvado.
- Criar um novo projeto dentro do Quartus.
- Colocar o endereço da pasta nomeada ALU.
- Em EDA Tool Settings, na opção simulation escolher ModelSim-Altera e o formato VHDL.
- Fim da criação do novo projeto.

Roteiro

Implementação da unidade aritmética, lógica e o MUX

Seguir os passos básicos:

- Criar um novo arquivo .VHD nomeado como **arith_unit.vhd**.
- Criar um novo arquivo .VHD nomeado como **logic_unit.vhd**.
- Criar um novo arquivo .VHD nomeado como **muxi.vhd**.
- Criar um novo arquivo .VHD nomeado como **alu.vhd**.
- Declaramos os componentes **arith_unit**, **logic_unit**, e **muxi** e depois os instanciamos.
- Para verificação da sintaxe, escolher o arquivo **alu.vhd** como **Top** e fazer a síntese do circuito.

Implementação da unidade aritmética, lógica e o MUX

Seguir os passos básicos:

- Criar um novo arquivo .VHD nomeado como **arith_unit.vhd**.
- Criar um novo arquivo .VHD nomeado como **logic_unit.vhd**.
- Criar um novo arquivo .VHD nomeado como **muxi.vhd**.
- Criar um novo arquivo .VHD nomeado como **alu.vhd**.
- Declaramos os componentes **arith_unit**, **logic_unit**, e **muxi** e depois os instanciamos.
- Para verificação da sintaxe, escolher o arquivo **alu.vhd** como **Top** e fazer a síntese do circuito.

Implementação da unidade aritmética, lógica e o MUX

Seguir os passos básicos:

- Criar um novo arquivo .VHD nomeado como **arith_unit.vhd**.
- Criar um novo arquivo .VHD nomeado como **logic_unit.vhd**.
- Criar um novo arquivo .VHD nomeado como **muxi.vhd**.
- Criar um novo arquivo .VHD nomeado como **alu.vhd**.
- Declaramos os componentes **arith_unit**, **logic_unit**, e **muxi** e depois os instanciamos.
- Para verificação da sintaxe, escolher o arquivo **alu.vhd** como **Top** e fazer a síntese do circuito.

Implementação da unidade aritmética, lógica e o MUX

Seguir os passos básicos:

- Criar um novo arquivo .VHD nomeado como **arith_unit.vhd**.
- Criar um novo arquivo .VHD nomeado como **logic_unit.vhd**.
- Criar um novo arquivo .VHD nomeado como **muxi.vhd**.
- Criar um novo arquivo .VHD nomeado como **alu.vhd**.
- Declaramos os componentes **arith_unit**, **logic_unit**, e **muxi** e depois os instanciamos.
- Para verificação da sintaxe, escolher o arquivo **alu.vhd** como **Top** e fazer a síntese do circuito.

Implementação da unidade aritmética, lógica e o MUX

Seguir os passos básicos:

- Criar um novo arquivo .VHD nomeado como **arith_unit.vhd**.
- Criar um novo arquivo .VHD nomeado como **logic_unit.vhd**.
- Criar um novo arquivo .VHD nomeado como **muxi.vhd**.
- Criar um novo arquivo .VHD nomeado como **alu.vhd**.
- Declaramos os componentes **arith_unit**, **logic_unit**, e **muxi** e depois os instanciamos.
- Para verificação da sintaxe, escolher o arquivo **alu.vhd** como **Top** e fazer a síntese do circuito.

Implementação da unidade aritmética, lógica e o MUX

Seguir os passos básicos:

- Criar um novo arquivo .VHD nomeado como **arith_unit.vhd**.
- Criar um novo arquivo .VHD nomeado como **logic_unit.vhd**.
- Criar um novo arquivo .VHD nomeado como **muxi.vhd**.
- Criar um novo arquivo .VHD nomeado como **alu.vhd**.
- Declaramos os componentes **arith_unit**, **logic_unit**, e **muxi** e depois os instanciamos.
- Para verificação da sintaxe, escolher o arquivo **alu.vhd** como **Top** e fazer a síntese do circuito.

Roteiro

RTL viewer

O **RTL Viewer** e **Technology Map Viewer** fornecem uma lista hierárquica do projeto e links para uma vista esquemática onde são exibidos os componentes dos elementos do projeto (vide Figura 2).

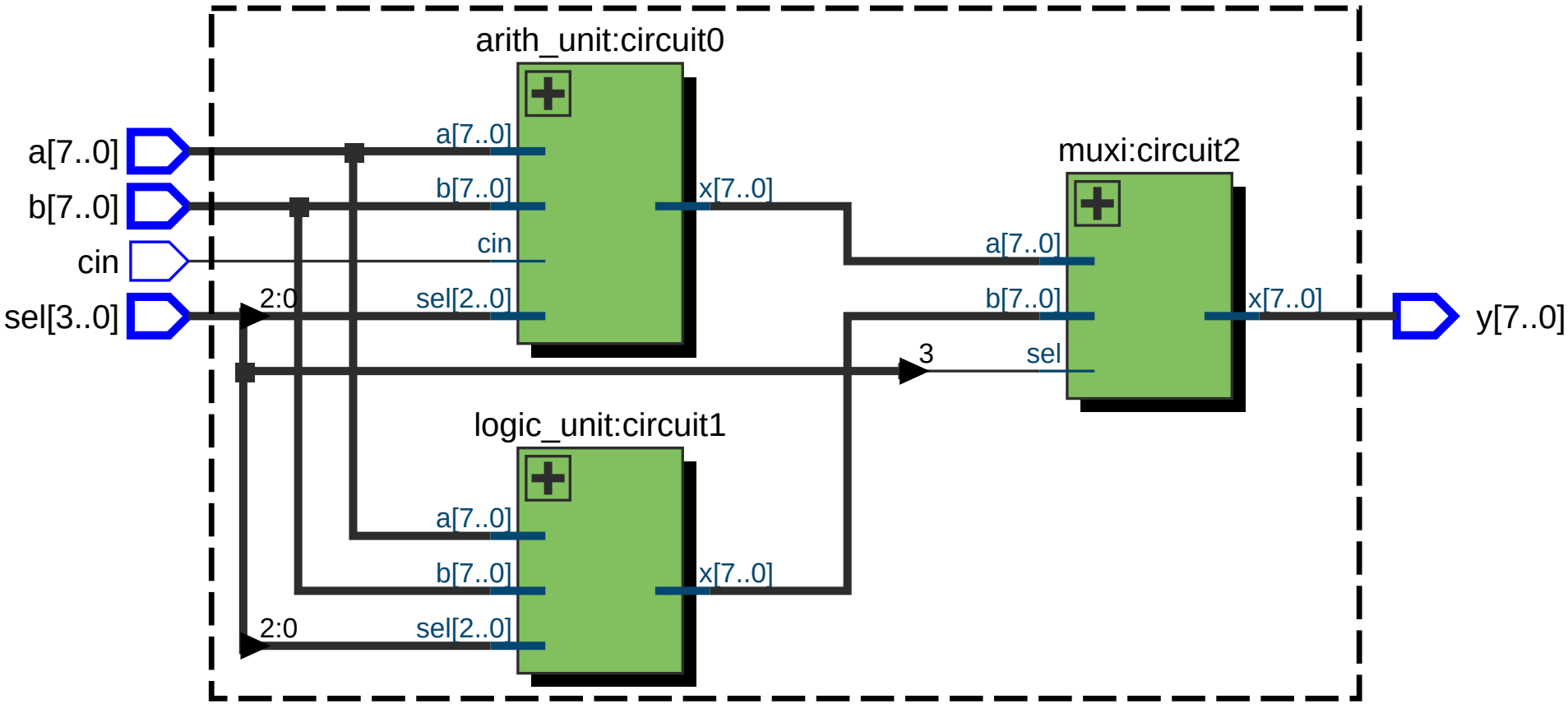

Figura 2: Esquemático

Roteiro

Verificação

O objetivo da verificação comportamental é a comprovação das especificações que foram propostas inicialmente, a representação é mostrada na Figura 3.

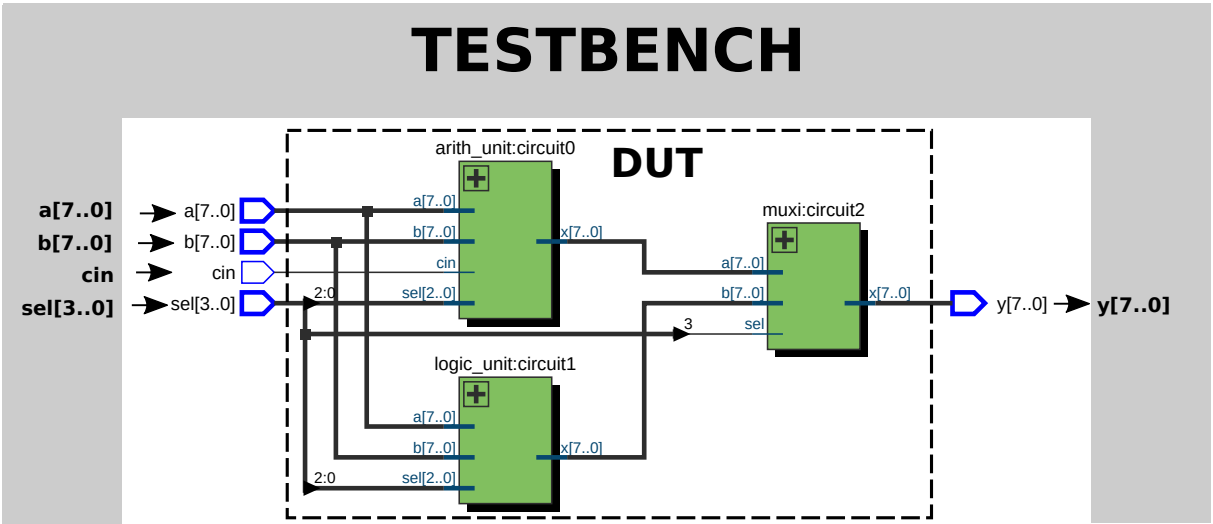

Figura 3: Verificação comportamental

Criação do TestBench

Seguir os passos básicos:

- Ir para **Processing->Start->Start Test Bench Template Writer**.
- Por defeito o Quartus gera um arquivo **.vht** dentro da pasta: **Simulation->ModelSim**.
- Mudar o nome da entidade do TestBench para **alu_tb** e salvar na pasta raiz do projeto como **alu_tb.vhd**.
- Deletar linhas dos estímulos que foram geradas pelo Quartus.
- Nomear a etiqueta da arquitetura que vai ser testada como **DUT (Design Under Test)**.
- Verificar o caminho onde está salvado o executável do ModelSim, assim: **Tool->EDA Tool Options->ModelSim-Altera**.

Criação do TestBench

Seguir os passos básicos:

- Ir para **Processing->Start->Start Test Bench Template Writer**.
- Por defeito o Quartus gera um arquivo **.vht** dentro da pasta: **Simulation->ModelSim**.
- Mudar o nome da entidade do TestBench para **alu_tb** e salvar na pasta raiz do projeto como **alu_tb.vhd**.
- Deletar linhas dos estímulos que foram geradas pelo Quartus.
- Nomear a etiqueta da arquitetura que vai ser testada como **DUT (Design Under Test)**.
- Verificar o caminho onde está salvado o executável do ModelSim, assim: **Tool->EDA Tool Options->ModelSim-Altera**.

Criação do TestBench

Seguir os passos básicos:

- Ir para **Processing->Start->Start Test Bench Template Writer**.
- Por defeito o Quartus gera um arquivo **.vht** dentro da pasta: **Simulation->ModelSim**.
- Mudar o nome da entidade do TestBench para **alu_tb** e salvar na pasta raiz do projeto como **alu_tb.vhd**.
- Deletar linhas dos estímulos que foram geradas pelo Quartus.
- Nomear a etiqueta da arquitetura que vai ser testada como **DUT (Design Under Test)**.
- Verificar o caminho onde está salvado o executável do ModelSim, assim: **Tool->EDA Tool Options->ModelSim-Altera**.

Criação do TestBench

Seguir os passos básicos:

- Ir para **Processing->Start->Start Test Bench Template Writer**.
- Por defeito o Quartus gera um arquivo **.vht** dentro da pasta: **Simulation->ModelSim**.
- Mudar o nome da entidade do TestBench para **alu_tb** e salvar na pasta raiz do projeto como **alu_tb.vhd**.
- Deletar linhas dos estímulos que foram geradas pelo Quartus.
- Nomear a etiqueta da arquitetura que vai ser testada como **DUT (Design Under Test)**.
- Verificar o caminho onde está salvado o executável do ModelSim, assim: **Tool->EDA Tool Options->ModelSim-Altera**.

Criação do TestBench

Seguir os passos básicos:

- Ir para **Processing->Start->Start Test Bench Template Writer**.
- Por defeito o Quartus gera um arquivo **.vht** dentro da pasta: **Simulation->ModelSim**.
- Mudar o nome da entidade do TestBench para **alu_tb** e salvar na pasta raiz do projeto como **alu_tb.vhd**.
- Deletar linhas dos estímulos que foram geradas pelo Quartus.
- Nomear a etiqueta da arquitetura que vai ser testada como **DUT (Design Under Test)**.
- Verificar o caminho onde está salvado o executável do ModelSim, assim: **Tool->EDA Tool Options->ModelSim-Altera**.

Criação do TestBench

Seguir os passos básicos:

- Ir para **Processing->Start->Start Test Bench Template Writer**.
- Por defeito o Quartus gera um arquivo **.vht** dentro da pasta: **Simulation->ModelSim**.
- Mudar o nome da entidade do TestBench para **alu_tb** e salvar na pasta raiz do projeto como **alu_tb.vhd**.
- Deletar linhas dos estímulos que foram geradas pelo Quartus.
- Nomear a etiqueta da arquitetura que vai ser testada como **DUT (Design Under Test)**.
- Verificar o caminho onde está salvado o executável do ModelSim, assim: **Tool->EDA Tool Options->ModelSim-Altera**.

Roteiro

Configuração da arquitetura de verificação

Seguir os passos básicos:

- Configurar o TestBench dentro do Quartus, assim: **Processing->Settings->EDA Tool Options->Simulation**.
- Verificar que as configurações estejam corretas.
- Adicionar e configurar o arquivo TestBench **alu_tb.vhd**.
- Observar que todos os arquivos .vhd, incluindo o TestBench, estão dentro da pasta raiz do projeto.
- Já configurado o TestBench, vamos lançar o ModelSim! **Tools->RTL simulation**.

Configuração da arquitetura de verificação

Seguir os passos básicos:

- Configurar o TestBench dentro do Quartus, assim: **Processing->Settings->EDA Tool Options->Simulation**.
- Verificar que as configurações estejam corretas.
- Adicionar e configurar o arquivo TestBench **alu_tb.vhd**.
- Observar que todos os arquivos .vhd, incluindo o TestBench, estão dentro da pasta raiz do projeto.
- Já configurado o TestBench, vamos lançar o ModelSim! **Tools->RTL simulation**.

Configuração da arquitetura de verificação

Seguir os passos básicos:

- Configurar o TestBench dentro do Quartus, assim: **Processing->Settings->EDA Tool Options->Simulation**.
- Verificar que as configurações estejam corretas.
- Adicionar e configurar o arquivo TestBench **alu_tb.vhd**.
- Observar que todos os arquivos .vhd, incluindo o TestBench, estão dentro da pasta raiz do projeto.
- Já configurado o TestBench, vamos lançar o ModelSim! **Tools->RTL simulation**.

Configuração da arquitetura de verificação

Seguir os passos básicos:

- Configurar o TestBench dentro do Quartus, assim: **Processing->Settings->EDA Tool Options->Simulation**.
- Verificar que as configurações estejam corretas.
- Adicionar e configurar o arquivo TestBench **alu_tb.vhd**.
- Observar que todos os arquivos .vhd, incluindo o TestBench, estão dentro da pasta raiz do projeto.
- Já configurado o TestBench, vamos lançar o ModelSim! **Tools->RTL simulation**.

Configuração da arquitetura de verificação

Seguir os passos básicos:

- Configurar o TestBench dentro do Quartus, assim: **Processing->Settings->EDA Tool Options->Simulation**.
- Verificar que as configurações estejam corretas.
- Adicionar e configurar o arquivo TestBench **alu_tb.vhd**.
- Observar que todos os arquivos .vhd, incluindo o TestBench, estão dentro da pasta raiz do projeto.
- Já configurado o TestBench, vamos lançar o ModelSim! **Tools->RTL simulation**.

Roteiro

Comandos básicos de uso do simulador

Seguir os passos básicos:

- Inicialmente a Janela que mostra as sinais de onda, chamada como **Waveforms** possui as extensões das sinais e apresenta uma pequena janela de tempo de simulação.
- No ícone nomeado como **Toggle leaf names <-> full names**, as extensões das sinais vão desaparecer e deixar só os nomes das sinais.
- Posicionar-se na janela **Waveforms**, pode ser habilitada a ferramenta **Zoom Full**.
- Mudança na representação numérica das sinais, selecionar as sinais e com o botão direito do mouse escolher: **Radix->Unsigned**.
- Posicionar-se na janela **Waveforms**, e armazenar as configurações das formas de onda modificadas.

Comandos básicos de uso do simulador

Seguir os passos básicos:

- Inicialmente a Janela que mostra as sinais de onda, chamada como **Waveforms** possui as extensões das sinais e apresenta uma pequena janela de tempo de simulação.
- No ícone nomeado como **Toggle leaf names <-> full names**, as extensões das sinais vão desaparecer e deixar só os nomes das sinais.
- Posicionar-se na janela **Waveforms**, pode ser habilitada a ferramenta **Zoom Full**.
- Mudança na representação numérica das sinais, selecionar as sinais e com o botão direito do mouse escolher: **Radix->Unsigned**.
- Posicionar-se na janela **Waveforms**, e armazenar as configurações das formas de onda modificadas.

Comandos básicos de uso do simulador

Seguir os passos básicos:

- Inicialmente a Janela que mostra as sinais de onda, chamada como **Waveforms** possui as extensões das sinais e apresenta uma pequena janela de tempo de simulação.
- No ícone nomeado como **Toggle leaf names <-> full names**, as extensões das sinais vão desaparecer e deixar só os nomes das sinais.
- Posicionar-se na janela **Waveforms**, pode ser habilitada a ferramenta **Zoom Full**.
- Mudança na representação numérica das sinais, selecionar as sinais e com o botão direito do mouse escolher: **Radix->Unsigned**.
- Posicionar-se na janela **Waveforms**, e armazenar as configurações das formas de onda modificadas.

Comandos básicos de uso do simulador

Seguir os passos básicos:

- Inicialmente a Janela que mostra as sinais de onda, chamada como **Waveforms** possui as extensões das sinais e apresenta uma pequena janela de tempo de simulação.
- No ícone nomeado como **Toggle leaf names <-> full names**, as extensões das sinais vão desaparecer e deixar só os nomes das sinais.
- Posicionar-se na janela **Waveforms**, pode ser habilitada a ferramenta **Zoom Full**.
- Mudança na representação numérica das sinais, selecionar as sinais e com o botão direito do mouse escolher: **Radix->Unsigned**.
- Posicionar-se na janela **Waveforms**, e armazenar as configurações das formas de onda modificadas.

Comandos básicos de uso do simulador

Seguir os passos básicos:

- Inicialmente a Janela que mostra as sinais de onda, chamada como **Waveforms** possui as extensões das sinais e apresenta uma pequena janela de tempo de simulação.
- No ícone nomeado como **Toggle leaf names <-> full names**, as extensões das sinais vão desaparecer e deixar só os nomes das sinais.
- Posicionar-se na janela **Waveforms**, pode ser habilitada a ferramenta **Zoom Full**.
- Mudança na representação numérica das sinais, selecionar as sinais e com o botão direito do mouse escolher: **Radix->Unsigned**.
- Posicionar-se na janela **Waveforms**, e armazenar as configurações das formas de onda modificadas.

Roteiro

Comandos básicos de uso do simulador

Seguir os passos básicos:

- Na pasta **Simulation->ModelSim** foi criado um arquivo com várias extensões, entre eles, os mais importantes têm extensão **<alguma-coisa>.d**.
- O arquivo **wave.do** armazena as configurações das sinais de onda.
- O arquivo **ULA_run_msim_rtl_vhdl**.do é o **script** que o ModelSim usa para fazer a simulação.
- Abrir o arquivo **ULA_run_msim_rtl_vhdl** num editor de texto e modificar a linha **add wave*** por:
  do wave.do
  view wave
- Testar as configurações criadas posicionando-se no prompt do ModelSim e procurar o comando:
  do ULA_run_msim_rtl_vhdl.do
- Programar os estímulos que vão entrar para a ALU

Comandos básicos de uso do simulador

Seguir os passos básicos:

- Na pasta **Simulation->ModelSim** foi criado um arquivo com várias extensões, entre eles, os mais importantes têm extensão **<alguma-coisa>.d**.
- O arquivo **wave.do** armazena as configurações das sinais de onda.
- O arquivo **ULA_run_msim_rtl_vhdl**.do é o **script** que o ModelSim usa para fazer a simulação.
- Abrir o arquivo **ULA_run_msim_rtl_vhdl** num editor de texto e modificar a linha **add wave*** por:
  do wave.do
  view wave
- Testar as configurações criadas posicionando-se no prompt do ModelSim e procurar o comando:
  do ULA_run_msim_rtl_vhdl.do
- Programar os estímulos que vão entrar para a ALU

Comandos básicos de uso do simulador

Seguir os passos básicos:

- Na pasta **Simulation->ModelSim** foi criado um arquivo com várias extensões, entre eles, os mais importantes têm extensão **<alguma-coisa>.d**.
- O arquivo **wave.do** armazena as configurações das sinais de onda.
- O arquivo **ULA_run_msim_rtl_vhdl**.do é o **script** que o ModelSim usa para fazer a simulação.
- Abrir o arquivo **ULA_run_msim_rtl_vhdl** num editor de texto e modificar a linha **add wave*** por:
  do wave.do
  view wave
- Testar as configurações criadas posicionando-se no prompt do ModelSim e procurar o comando:
  do ULA_run_msim_rtl_vhdl.do
- Programar os estímulos que vão entrar para a ALU

Comandos básicos de uso do simulador

Seguir os passos básicos:

- Na pasta **Simulation->ModelSim** foi criado um arquivo com várias extensões, entre eles, os mais importantes têm extensão **<alguma-coisa>.d**.
- O arquivo **wave.do** armazena as configurações das sinais de onda.
- O arquivo **ULA_run_msim_rtl_vhdl**.do é o **script** que o ModelSim usa para fazer a simulação.
- Abrir o arquivo **ULA_run_msim_rtl_vhdl** num editor de texto e modificar a linha **add wave*** por:
  do wave.do
  view wave
- Testar as configurações criadas posicionando-se no prompt do ModelSim e procurar o comando:
  do ULA_run_msim_rtl_vhdl.do
- Programar os estímulos que vão entrar para a ALU

Comandos básicos de uso do simulador

Seguir os passos básicos:

- Na pasta **Simulation->ModelSim** foi criado um arquivo com várias extensões, entre eles, os mais importantes têm extensão **<alguma-coisa>.d**.
- O arquivo **wave.do** armazena as configurações das sinais de onda.
- O arquivo **ULA_run_msim_rtl_vhdl**.do é o **script** que o ModelSim usa para fazer a simulação.
- Abrir o arquivo **ULA_run_msim_rtl_vhdl** num editor de texto e modificar a linha **add wave*** por:
  do wave.do
  view wave
- Testar as configurações criadas posicionando-se no prompt do ModelSim e procurar o comando:
  do ULA_run_msim_rtl_vhdl.do
- Programar os estímulos que vão entrar para a ALU

Comandos básicos de uso do simulador

Seguir os passos básicos:

- Na pasta **Simulation->ModelSim** foi criado um arquivo com várias extensões, entre eles, os mais importantes têm extensão **<alguma-coisa>.d**.
- O arquivo **wave.do** armazena as configurações das sinais de onda.
- O arquivo **ULA_run_msim_rtl_vhdl**.do é o **script** que o ModelSim usa para fazer a simulação.
- Abrir o arquivo **ULA_run_msim_rtl_vhdl** num editor de texto e modificar a linha **add wave*** por:
  do wave.do
  view wave
- Testar as configurações criadas posicionando-se no prompt do ModelSim e procurar o comando:
  do ULA_run_msim_rtl_vhdl.do
- Programar os estímulos que vão entrar para a ALU

Roteiro

Monitorando componentes internos

Seguir os passos básicos:

- Pode-se mudar a cor de cada sinal dentro da janela **Waveforms**.
- Se quisermos ver as sinais de cada componente da ALU, no **Workspace** do ModelSim aparecem as janelas nomeadas como **Instance (Hierarquia do circuito)** e **Objects (Entidade de cada circuito)**.
- Escolher as portas de cada instancia que vai ser monitorada e arrastrar para a janela **WaveForms**.
- Abrir o arquivo **ULA_run_msim_rtl_vhdl.do** num editor de texto e modificar o tempo de simulação.
- Testar as configurações criadas posicionando-se no prompt do ModelSim e procurar o comando:
  do ULA_run_msim_rtl_vhdl.do

Monitorando componentes internos

Seguir os passos básicos:

- Pode-se mudar a cor de cada sinal dentro da janela **Waveforms**.
- Se quisermos ver as sinais de cada componente da ALU, no **Workspace** do ModelSim aparecem as janelas nomeadas como **Instance (Hierarquia do circuito)** e **Objects (Entidade de cada circuito)**.
- Escolher as portas de cada instancia que vai ser monitorada e arrastrar para a janela **WaveForms**.
- Abrir o arquivo **ULA_run_msim_rtl_vhdl.do** num editor de texto e modificar o tempo de simulação.
- Testar as configurações criadas posicionando-se no prompt do ModelSim e procurar o comando:
  do ULA_run_msim_rtl_vhdl.do

Monitorando componentes internos

Seguir os passos básicos:

- Pode-se mudar a cor de cada sinal dentro da janela **Waveforms**.
- Se quisermos ver as sinais de cada componente da ALU, no **Workspace** do ModelSim aparecem as janelas nomeadas como **Instance (Hierarquia do circuito)** e **Objects (Entidade de cada circuito)**.
- Escolher as portas de cada instancia que vai ser monitorada e arrastrar para a janela **WaveForms**.
- Abrir o arquivo **ULA_run_msim_rtl_vhdl.do** num editor de texto e modificar o tempo de simulação.
- Testar as configurações criadas posicionando-se no prompt do ModelSim e procurar o comando:
  do ULA_run_msim_rtl_vhdl.do

Monitorando componentes internos

Seguir os passos básicos:

- Pode-se mudar a cor de cada sinal dentro da janela **Waveforms**.
- Se quisermos ver as sinais de cada componente da ALU, no **Workspace** do ModelSim aparecem as janelas nomeadas como **Instance (Hierarquia do circuito)** e **Objects (Entidade de cada circuito)**.
- Escolher as portas de cada instancia que vai ser monitorada e arrastrar para a janela **WaveForms**.
- Abrir o arquivo **ULA_run_msim_rtl_vhdl.do** num editor de texto e modificar o tempo de simulação.
- Testar as configurações criadas posicionando-se no prompt do ModelSim e procurar o comando:
  do ULA_run_msim_rtl_vhdl.do

Monitorando componentes internos

Seguir os passos básicos:

- Pode-se mudar a cor de cada sinal dentro da janela **Waveforms**.
- Se quisermos ver as sinais de cada componente da ALU, no **Workspace** do ModelSim aparecem as janelas nomeadas como **Instance (Hierarquia do circuito)** e **Objects (Entidade de cada circuito)**.
- Escolher as portas de cada instancia que vai ser monitorada e arrastrar para a janela **WaveForms**.
- Abrir o arquivo **ULA_run_msim_rtl_vhdl.do** num editor de texto e modificar o tempo de simulação.
- Testar as configurações criadas posicionando-se no prompt do ModelSim e procurar o comando:
  do ULA_run_msim_rtl_vhdl.do