

Instalação do Simulador GHDL

1 O que é o GHDL?

Uma das ferramentas sugeridas para fazer os exercícios em VHDL é o software *open source* GHDL, que é um analisador, compilador e simulador de *designs* de circuitos digitais descritos em VHDL. Com o GHDL você poderá simular e verificar se o circuito projetado funciona conforme o esperado.

O site do GHDL está no seguinte endereço:

<http://ghdl.free.fr/>

A documentação está disponível em

https://ghdl.readthedocs.io/en/latest/examples/quick_start/README.html

O código fonte está disponível em

<https://github.com/ghdl/ghdl>

O GHDL é um simulador de circuitos digitais, mas não tem incorporada nele uma interface para apresentar o resultado da simulação. Ao invés disso, o GHDL permite gravar o resultado num arquivo no formato VCD (Value Change Dump). Por sua vez, podemos usar o programa *open source* GtkWave para visualizar os arquivos no formato VCD.

O GtkWave está disponível no seguinte endereço:

<http://gtkwave.sourceforge.net/>

A documentação está disponível em

<http://gtkwave.sourceforge.net/gtkwave.pdf>

O código fonte está disponível em

<https://github.com/gtkwave/gtkwave>

Nas seções a seguir são apresentados instruções para a instalação do GHDL e do GtkWave nos sistemas operacionais Windows 10, Ubuntu 18.4 e Mac OS X Catalina.

2 Instalação no Windows 10

Apresentamos 2 opções de instalação do GHDL para o sistema operacional Windows 10:

1. Versão compilada em 32 bits para execução nativa no Windows 10

- (a) Instale o arquivo do *script* InstalarGHDL.ps1 na pasta c:/programs/ghdl;
- (b) Usando o terminal PowerShell no modo Administrador execute os comandos

```
cd c:\programs\ghdl
./InstalarGHDL.ps1 -download
```

O script de instalação fará o download e a instalação do GHDL, incluindo a configuração do caminho (PATH) para a execução via linha de comando.

Se preferir, faça o download do arquivo ghdl-0.37-mingw32-mcode.zip disponível em

<https://github.com/ghdl/ghdl/releases/download/v0.37/>

Copie o arquivo baixado para a pasta c:/programs/ghdl e execute os seguintes comandos no Power Shell (no modo Administrador):

```
cd c:\programs\ghdl
./InstalarGHDL.ps1
```

2. Versão compilada em 64 bits para execução via Container Docker (opção recomendada)

- (a) Instale o Docker Desktop para Windows, seguindo as instruções disponíveis em

<https://docs.docker.com/docker-for-windows/install/>

- (b) Instale o container contendo o GHDL usando o seguinte comando no terminal Power Shell do Windows 10

```
docker pull ghdl/ghdl:buster-mcode
```

- (c) Crie a pasta “mydata” em c:/Users/<seu usuário>, executando os seguintes comandos no Power Shell:

```
cd c:\Users\<seu usuário>
mkdir mydata
```

- (d) Para iniciar o container, execute o seguinte comando no Power Shell:

```
docker run --rm -v C:/Users/<seu usuário>/mydata:/workspace
-w /workspace --name ghdl -it ghdl/ghdl:buster-mcode /bin/bash
```

Os arquivos do *design* VHDL deverão estar na pasta mydata no sistema de arquivos do Windows 10.

Adicionalmente a essas 2 alternativas de instalação, você precisará instalar o visualizador GtkWave e um editor para as descrições em VHDL. As sugestões de editor são o Microsoft Visual Studio Code e o GNU Emacs. As instruções para instalação estão disponíveis em

- GtkWave: <http://gtkwave.sourceforge.net/>
Faça o download do arquivo gtkwave-3.3.100-bin-win64.zip.
Copie o arquivo baixado para a pasta c:/programs/gtkwave e execute os comandos no Power Shell (no modo Administrador):

```
cd c:\programs\gtkwave
./InstalarGtkWave.ps1
```
- Visual Studio Code: <https://code.visualstudio.com/download>
- GNU Emacs: <https://www.gnu.org/software/emacs/download.html>

3 Instalação no Ubuntu 18.04

Inicie instalando o Docker, seguindo as instruções disponíveis em

<https://www.digitalocean.com/community/tutorials/como-instalar-e-usar-o-docker-no-ubuntu-18-04-pt>

Instale o container contendo o GHDL executando o seguinte comando num terminal:

```
docker pull ghdl/ghdl:ubuntu18-mcode
```

Crie a pasta “mydata” na pasta raiz do seu usuário, executando os seguintes comandos no terminal:

```
cd ~/
mkdir mydata
```

Para iniciar o container, execute o seguinte comando terminal:

```
docker run --rm --name ghdl -u $(id -u):$(id -g) -v ~/mydata:/workspace
-w /workspace -it ghdl/ghdl:ubuntu18-mcode /bin/bash
```

Os arquivos do *design* VHDL deverão estar na pasta mydata no sistema de arquivos do Ubuntu.

Adicionalmente você precisará instalar o visualizador GtkWave e um editor para as descrições em VHDL. As sugestões de editor são o Microsoft Visual Studio Code e o GNU Emacs.

Para instalar o GtkWave, execute os seguintes comandos no terminal:

```
sudo apt-get update
sudo apt install gtkwave
```

Para instalar o GNU Emacs, execute os seguintes comando no terminal:

```
sudo apt-get update
sudo apt-get install emacs
```

As instruções para instalação do Visual Studio Code estão disponíveis em

<https://code.visualstudio.com/download>

4 Instalação no Mac OS X

Inicie instalando o Docker, seguindo as instruções disponíveis em

<https://docs.docker.com/docker-for-mac/install/>

Instale o container com executando o seguinte comando num terminal:

```
docker pull ghdl/ghdl:ubuntu18-mcode
```

Crie a pasta “mydata” na pasta raiz do seu usuário, executando os seguintes comandos no terminal:

```
cd ~/
mkdir mydata
```

Para iniciar o container, execute o seguinte comando terminal:

```
docker run --rm --name ghdl -u $(id -u):$(id -g) -v ~/mydata:/workspace
-w /workspace -it ghdl/ghdl:ubuntu18-mcode /bin/bash
```

Os arquivos do *design* VHDL deverão estar na pasta mydata no sistema de arquivos do Mac OS X.

Adicionalmente você precisará instalar o visualizador GtkWave e um editor para as descrições em VHDL. As sugestões de editor são o Microsoft Visual Studio Code e o Aquamacs (port do Emacs para Mac OS X).

O Gtkwave para Mac OS X está disponível em

<https://sourceforge.net/projects/gtkwave/files/>

Faça o download do arquivo gtkwave-3.3.103-osx-app, descomprima e copie o arquivo gtkwave-app para a pasta de Aplicativos do Mac OS X.

As instruções para instalação do Aquamacs estão disponíveis em

<https://aquamacs.org/>

As instruções para instalação do Visual Studio Code estão disponíveis em

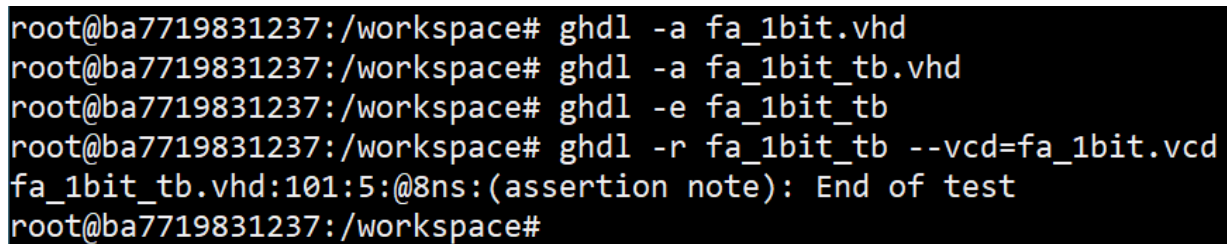
<https://code.visualstudio.com/download>

5 Testando a instalação do GHDL

Para testar a instalação do GHDL, utilize os arquivos `fa_1bit.vhd` e `fa_1bit_tb.vhd`, que contém o projeto do somador completo de 1 bit e do testbench correspondente.

Para as instalações no Windows 10, execute os seguintes comandos no Power Shell ou no terminal do container (ver Figura 1):

```
ghdl -a fa_1bit.vhd
ghdl -a fa_1bit_tb.vhd
ghdl -e fa_1bit_tb
ghdl -r fa_1bit_tb --vcd=fa_1bit.vcd
```

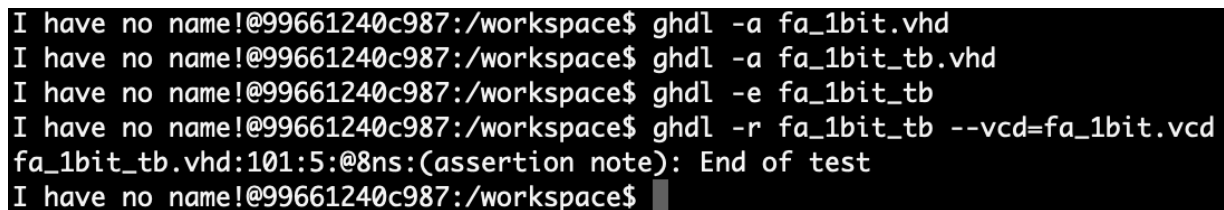


```
root@ba7719831237:/workspace# ghdl -a fa_1bit.vhd
root@ba7719831237:/workspace# ghdl -a fa_1bit_tb.vhd
root@ba7719831237:/workspace# ghdl -e fa_1bit_tb
root@ba7719831237:/workspace# ghdl -r fa_1bit_tb --vcd=fa_1bit.vcd
fa_1bit_tb.vhd:101:5:@8ns:(assertion note): End of test
root@ba7719831237:/workspace#
```

Figura 1: Compilação e simulação com o GHDL

Para as instalações em container GHDL no Ubuntu 18 e no Mac OS X, execute os seguintes comandos no terminal do container (ver Figura 2):

```
ghdl -a fa_1bit.vhd
ghdl -a fa_1bit_tb.vhd
ghdl -e fa_1bit_tb
ghdl -r fa_1bit_tb --vcd=fa_1bit.vcd
```



```
I have no name!@99661240c987:/workspace$ ghdl -a fa_1bit.vhd
I have no name!@99661240c987:/workspace$ ghdl -a fa_1bit_tb.vhd
I have no name!@99661240c987:/workspace$ ghdl -e fa_1bit_tb
I have no name!@99661240c987:/workspace$ ghdl -r fa_1bit_tb --vcd=fa_1bit.vcd
fa_1bit_tb.vhd:101:5:@8ns:(assertion note): End of test
I have no name!@99661240c987:/workspace$ █
```

Figura 2: Compilação e simulação com o GHDL

Para visualizar o resultado da simulação, abra o arquivo `fa_1bit.vcd` no GtkWave. Você verá o resultado conforme é mostrado na Figura 3.

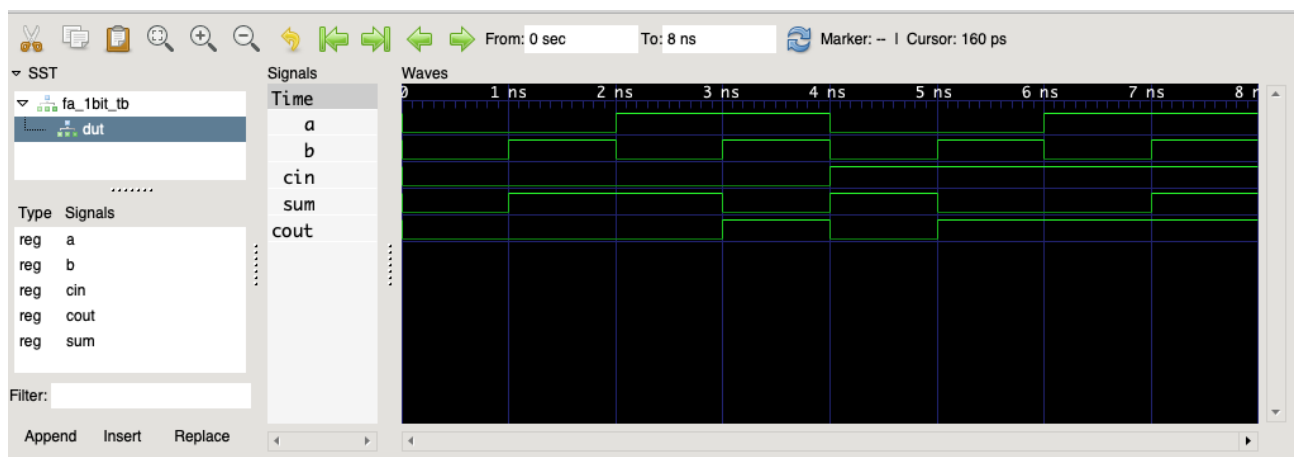


Figura 3: Visualização do resultado com o GtkWave