

µProcessador 0 Instalação dos Softwares em Casa

NOTA: nos labs da UTFPR eles *devem estar funcionando*, portanto pode ir direto para o laboratório 1.

É preciso instalar um compilador VHDL (*ghdl 0.35*) e um visualizador de formas de onda (*gtkwave 3.3.87*), no mínimo. As sugestões dadas são boas, acreditem.

Há versões para Linux, Windows e OS X.

Editor de Código

VHDL é texto puro, então qualquer porcaria serve, mas um bom editor ajuda muito.

Gostei do **SublimeText 3**, mas os acólitos do Bill Gates podem usar o **Notepad++** que é muito bom de forma geral. No Ubuntu, até o **gedit** (configurado adequadamente e com plugins¹) pode ser agradável.

Outra opção é usar o **Sigasi** na versão *standalone free* (Sigasi Studio Starter Edition), pra qualquer Sistema Operacional. Realmente excelente, específico para VHDL. E pro pessoal *hardcore*, dizem que o **emacs** e o **vim** configurados pra VHDL são os melhores, mas eu quero distância.

Linux

Abaixo as observações para a versão 0.33 no Ubuntu 14.04 LTS 64 bits, okay?

1. GHDL: Para instalar, baixe o pacote e abra: http://sourceforge.net/projects/ghdl-updates/files/Buils/ghdl-0.33/debian/ghdl_0.33-1ubuntu1_amd64.deb/download (ou digite “ghdl” no google). A central de programas faz o resto (ou use `sudo dpkg --install`).

Pode ser necessário instalar também *libgnat-4.8*² com `sudo apt-get install` (um `-f` pode ser necessário depois) e também *lib32z1-dev*³. Caso dê algum problema, confira as dicas de: <http://sourceforge.net/p/umhdl/wiki/Installation%20-%20Linux/#possible-problems-in-linux-64-bits> (as bibliotecas a instalar devem ser instaladas em ordem nesse caso, ok?)

2. GtkWave: com alguma sorte, `sudo apt-get install gtwave` resolve a parada.

Caso contrário, pode ser casca. Vai em <http://gtkwave.sourceforge.net/>, baixe e descompacte e leia o arquivo *README.txt*, que tem uma seção sobre instalação no Ubuntu.

3. Teste: veja a seção posterior.

Windows

Testado no meu Windows 10.

1. GHDL: deve ser fácil:

1. Vá no site oficial (ou vá por “ghdl” no google) e baixe o arquivo compactado⁴: <https://github.com/ghdl/ghdl/releases/download/v0.35/ghdl-0.35-mcode-windows.zip>.
2. Descompacte manualmente o arquivo para a pasta em que você vai trabalhar (digamos, pasta VHDL dentro da pasta Documentos).
3. Abra o *Windows PowerShell* (ache pela busca na barra); um terminal *shell* modo texto de linha de comando deve aparecer, com “*PS C:\Users\Juliano>*” ou algo similar na tela.
4. Entre na pasta que você descompactou com o comando de terminal “*cd*”. Por ex., se foi

1 *Snippets, word completion, code comment, smart spaces, quick open* e tamanho 4 para as tabulações, com *snippet* para arquivo vazio.

2 Se acusar falta de *libgnat* durante os testes.

3 Caso seja emitido um erro como “*/usr/bin/ld: cannot find -lz*”

4 Alternativamente, baixe o script (arquivo .ps1) e execute-o no *Windows Powershell*, se você estiver habituado com ele (a política de execução padrão tem que ser alterada, por exemplo).

descompactado na pasta padrão “Documentos”, é só digitar `cd Documentos` (teste o uso da tecla `Tab` para autocompletar o nome da pasta) ou, mais completo por exemplo, `cd c:\Users\Juliano\Documentos`.

5. Teste a instalação, digitando o comando `..\ghdl-0.35-mcode-windows\bin\ghdl.exe -v`

6. Informações de versão e copyright deverão aparecer na tela.

Se você tiver problemas ou achar difícil, por favor dê um *feedback* ao professor.

2. GtkWave: o site oficial é o <http://www.dspia.com/gtkwave.html>; mas pode ir baixar direto o arquivo em <https://sourceforge.net/projects/gtkwave/files/gtkwave-3.3.87-bin-win32/gtkwave-3.3.87-bin-win32.zip/download>.

Descompacte e execute; basta achar a subpasta `bin` e, dentro dela, dar duplo-clique no arquivo executável `gtkwave`. A janela principal do programa deverá abrir.

Testando a Instalação

Esta é uma versão do 1º laboratório sem as explicações.

Crie uma pasta de trabalho dentro da pasta do `ghdl` descompactado, por exemplo, pasta `lab_teste` dentro da pasta `\Downloads\ghdl-0.35-mcode-windows\`.

Se der problema no Linux, consulte <http://sourceforge.net/p/umhdl/wiki/Installation%20-%20Linux/> (é sobre outro aplicativo, o UMHDL, que usa ambos o `ghdl` e o `gtkwave`) e também garanta que `libgnat-4.8` e `lib32z1-dev` estejam instalados.

1. Fonte VHDL: digite o abaixo num editor de texto e grave como `porta_e.vhdl` dentro da pasta de trabalho `lab_teste`. As explicações do código estão no roteiro do microprocessador 1.

```
library ieee;
use ieee.std_logic_1164.all;

entity porta_e is
    port( in_a,in_b: in std_logic;
          a_e_b: out std_logic
    );
end entity;

architecture aporta_e of porta_e is
begin
    a_e_b <= in_a and in_b;
end architecture;
```

2. Testbench: digite o arquivo para testes abaixo e grave como `porta_e_tb.vhdl` dentro da pasta de trabalho `lab_teste`.

```
library ieee;
use ieee.std_logic_1164.all;

entity porta_e_tb is
end entity;

architecture aporta_e_tb of porta_e_tb is
    component porta_e is
        port( in_a,in_b: in std_logic;
              a_e_b: out std_logic
        );
    end component;
```

```

    );
end component;
signal a,b,e: std_logic;
begin
    uut: porta_e port map(in_a=>a,in_b=>b,a_e_b=>e);
    process
    begin
        a <= '0';
        b <= '0';
        wait for 50 ns;
        a <= '0';
        b <= '1';
        wait for 50 ns;
        a <= '1';
        b <= '0';
        wait for 50 ns;
        a <= '1';
        b <= '1';
        wait for 50 ns;
        a <= '0';
        b <= '0';
        wait;
    end process;
end architecture;

```

3. Compilação: a sequência de comandos abaixo deve ser executada num terminal, **um a um**, *sem mensagens de erro*. Caso haja mensagens, verifique erros de digitação ou de cópia.

Windows	Linux
cd Downloads\ghdl-0.35-mcode-windows\lab_teste	cd lab_teste
..\bin\ghdl.exe -a porta_e.vhdl	ghdl -a porta_e.vhdl
..\bin\ghdl.exe -a porta_e_tb.vhdl	ghdl -a porta_e_tb.vhdl
..\bin\ghdl.exe -r porta_e_tb --wave=ondas.ghw	ghdl -r porta_e_tb --wave=ondas.ghw

=> Note que não se inclui o “.vhdl” no último comando!

4. Simulação: para a visualização das formas de onda finais usamos o gtkwave. Passos:

1. Localize o executável gtkwave dentro da subpasta *bin* e execute-o com duplo clique.
2. Vá no menu *File* e selecione *Open new tab*.
3. Localize o arquivo *ondas.ghw* dentro da sua pasta de trabalho *lab_teste* e abra-o. Note que as ondas não aparecem a princípio.
4. Expanda a *treeview* que está no painel superior esquerdo (diz SST nele), clicando duas vezes na palavra *top*.
5. Clique na entidade *porta_e_tb* na árvore.
6. Selecione todos os sinais que aparecem ali (são *a*, *b* e *e*) e clique *Insert*.
7. Vá ao menu *Time => Zoom* e escolher *Zoom Full*.

As formas de onda devem aparecer no painel preto.

Outras Alternativas

Parece uma excelente ideia usar uma IDE que edite, compile e simule, tudo dentro do meio ambiente, não é mesmo?

Não é.

Tanto o **Xilinx ISE** quanto o **Altera Quartus II** são IDEs assim, mas são pesadíssimos e lentos. O Quartus em particular é uma fonte de problemas bastante desagradáveis para o usuário casual, como nós... Os editores integrados são, surpreendentemente, uma droga, e os simuladores nada amigáveis nas versões atuais destes programas.

No entanto, para a gravação de FPGAs destes fabricantes, eles são a única possibilidade. Se você já tem contato ou hábito de usar um destes, o projeto pode ser feito nele sem problemas.

Para interessados em instruções de utilização do Quartus II, há uma versão alternativa do 1º laboratório de microprocessador.