



EEA-27 Microcontroladores e Sistemas Embarcados
Prof. Marcus Victor
Prof. Denis Loubach

Laboratório 1 - Preparação de ambiente
VERSÃO 1 - XX/XXXX

1 Objetivos

1.1 Objetivos específicos

- Preparar o ambiente para execução das atividades em laboratório;
- Testar configurações realizadas a partir de um programa "blink" ou "piscar".

2 Pré-requisitos

- Sistema Operacional Linux Ubuntu (mínimo: versão 20.4);
- Placa ICESugar-nano com conector USB;
- Drivers FTDI.

Caso não tenha os *drivers* instalados, use os seguintes comandos no terminal:

```
mkdir ftdi
cd ftdi
wget https://www.ftdichip.com/Drivers/D2XX/Linux/libftd2xx-x86_64-1.4.8.gz
tar xfvz libftd2xx-x86_64-1.4.8.gz
cd release/build/
sudo -s
cp libftd2xx.* /usr/local/lib
chmod 0755 /usr/local/lib/libftd2xx.so.1.4.8
ln -sf /usr/local/lib/libftd2xx.so.1.4.8 /usr/local/lib/libftd2xx.so
exit
```

3 Atividades

3.1 Instalação de *Drivers* e demais pacotes

Primeiro será realizada a instalação dos pacotes para Ubuntu, abra o terminal na pasta pessoal ou em outro destino que preferir, e use os seguintes comandos:

```
sudo apt-get install build-essential clang bison flex libreadline-dev \
    gawk tcl-dev libffi-dev git mercurial graphviz \
    xdot pkg-config python python3 libftdi-dev \
    qt5-default python3-dev libboost-all-dev cmake libeigen3-dev
```

3.2 Instalação das ferramentas

- **Icestorm Toolchain**

O conjunto de ferramentas icedstorm, empregue de acordo com o FPGA da placa ICESugar-nano, pode ser instalado com os comandos:

```
git clone https://github.com/YosysHQ/icedstorm.git icedstorm
cd icedstorm
make -j$(nproc)
sudo make install
```

- Retorne a pasta inicial por meio do comando:

```
cd ..
```

- **NextPNR**

O NextPNR é utilizado como ferramenta para roteamento e posicionamento. Utilize os seguintes comandos:

```
git clone https://github.com/YosysHQ/nextpnr nextpnr
cd nextpnr
cmake -DARCH=ice40 -DCMAKE_INSTALL_PREFIX=/usr/local .
make -j$(nproc)
sudo make install
```

Aguarde a instalação, poderá levar alguns minutos. Caso necessário o emprego da GUI (interface gráfica), adicione a seguinte informação na linha cmake:

```
-DBUILD_GUI=ON
```

Obs.: Sempre que alterar as ferramentas Icestorm, também efetue um *rebuild* do NextPNR.

- Retorne a pasta inicial da instalação.

- **Yosys**

O Yosys será aplicado na síntese do código em Verilog, para instalação use:

```
git clone https://github.com/YosysHQ/yosys.git yosys
cd yosys
make -j$(nproc)
sudo make install
```

- Lembre-se de voltar ao destino original.

- **Icarus Verilog + Gtkwave**

O Icarus mais o software Gtkwave irão servir como simulador, para instalação use:

```
sudo apt install iverilog gtkwave
```

- Lembre-se de voltar ao destino original.

4 Teste de ferramentas

Para verificar se o conjunto instalado está funcionando adequadamente, vamos realizar um teste com o LED laranja da placa. O código originalmente escrito por wuxx^[1] foi adaptado para uso em lab:

- Crie uma pasta localizada dentro do destino escolhido para instalação, use esta pasta para salvar os próximos arquivos solicitados;
- Implemente o código seguinte em um arquivo chamado "piscar" de extensão verilog .v:

```
module switch(    input CLK,
                  output LED
                );

    reg [25:0] counter;

    assign LED = ~counter[21];

    initial begin
        counter = 0;
    end

    always @(posedge CLK)
    begin
        counter <= counter + 1;
    end

endmodule
```

- Implemente o código a seguir em um arquivo chamado "io" de extensão .pcf:

```
#    BOARD PINS

set_io LED B6

set_io CLK D1

#    PMOD FULL (PMOD3)
set_io RX  A3
set_io TX  B3
set_io --warn-no-port PMOD1 B4
set_io --warn-no-port PMOD2 C6
set_io --warn-no-port PMOD3 B5
set_io --warn-no-port PMOD4 E3
set_io --warn-no-port PMOD5 E1
set_io --warn-no-port PMOD6 C2
set_io --warn-no-port PMOD7 B1
set_io --warn-no-port PMOD8 A1

#    PMOD LEFT (PMOD2)
set_io --warn-no-port PMODL1 B3
set_io --warn-no-port PMODL2 A3
set_io --warn-no-port PMODL3 B6
set_io --warn-no-port PMODL4 C5

#    PMOD RIGHT (PMOD1)
```

```
set_io --warn-no-port PMODR1 A1
set_io --warn-no-port PMODR2 B1
set_io --warn-no-port PMODR3 D1
set_io --warn-no-port PMODR4 E2
```

- Crie um arquivo *makefile* com os seguintes comandos:

```
filename = piscar
pcf_file = io.pcf

ICELINK_DIR=$(shell df | grep iCELink | awk '{print $6}')
${warning iCELink path: $(ICELINK_DIR)}

build:
    yosys -p "synth_ice40 -json $(filename).json -blif $(filename).blif" $(filename).v
    nextpnr-ice40 --lp1k --package cm36 --json $(filename).json --pcf $(pcf_file) --asc $(filename).asc
    icespack $(filename).asc $(filename).bin

prog_flash:
    @if [ -d '$(ICELINK_DIR)' ]; \
    then \
        cp $(filename).bin $(ICELINK_DIR); \
    else \
        echo "iCELink not found"; \
        exit 1; \
    fi

clean:
    rm -rf $(filename).blif $(filename).asc $(filename).bin
```

- Abra a pasta dentro do terminal de comando e execute o seguinte comando:

```
make build
```

- Verifique os resultados, caso nenhum erro seja identificado, utilize:

```
make prog_flash
```

- Com isso, a LED laranja deve piscar em uma determinada frequência.

5 Exercícios sugeridos

1. Abra o arquivo `piscar.v` e na linha 8, e substitua `"~counter[23]"` por `"~counter[19]"`, o que ocorre com a LED?
2. O arquivo `"io.pcf"` contém algumas informações relevantes utilizadas pelas ferramentas, sabe dizer o que `"io.pcf"` descreve?

Referencias

[1] <https://github.com/wuxx/icesugar-nano>