Insper

Bits e Processadores

FPGA

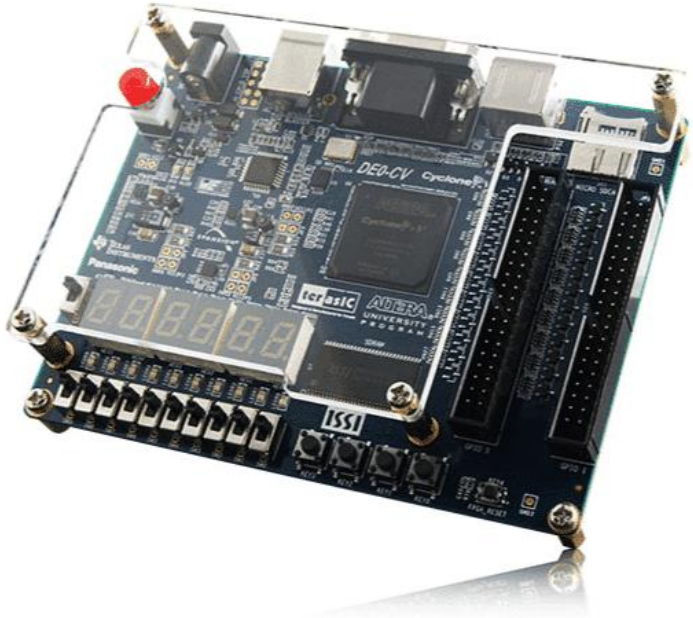
*

Ciência da Computação - Luciano Soares <lpsoares@insper.edu.br>

FPGA

Field Programmable Gate Array

Chip que pode ser programado ou reprogramado após a fabricação para executar qualquer tipo de circuito digital.



Insper

pytest

Framework de testes de código aberto para a linguagem Python que facilita a criação de testes simples, legíveis e escaláveis.

Ideal para TDD (**Test Driven Development**): prática de programação onde você escreve o teste automatizado de uma funcionalidade antes de escrever o código real que faz ela funcionar.

MyHDL

HDL significa Hardware Description Language (Linguagem de Descrição de Hardware). É um tipo de linguagem de computador usada para projetar, testar e descrever circuitos eletrônicos e sistemas digitais, como chips e FPGAs.

As linguagens HDL descrevem componentes físicos que funcionam ao mesmo tempo, ou seja, em paralelo. Os exemplos mais famosos de HDL tradicionais são o Verilog e o VHDL.

O MyHDL é um pacote de código aberto que permite usar a linguagem Python como uma linguagem de descrição e verificação de hardware.

Configurando o Sistema Localmente

PYTHON:

```
pip3 install -r requirements.txt
```

Praticando MyHDL

Edite o arquivo ***comb_modules.py*** com a lógica a seguir:

```
def exe1():
-    pass
+    q.next = a or (not b)
```

Execute: ***pytest -k exe1*** e verifique que o teste passa no teste.

Praticando MyHDL

Continue com o arquivo ***comb_modules.py***

Implemente os módulos **exe2** e **exe3**

Rode:

pytest -k exe2

pytest -k exe3

Rodando pelo Docker

Requisitos:

1. Docker Desktop instalado e aberto.
2. Git instalado.
3. Terminal funcionando.
4. Projeto clonado localmente.
5. Instale o fpgaloader

https://github.com/insper-bits/lab-myhdl-comb

https://github.com/Insper/fpgaloader/releases

Ajustando toplevel.py

```
@block
def toplevel(LEDR, SW, HEX0, HEX1):
    # Aux signals
    ledr_s = [Signal(bool(0)) for i in range(10)]

    # Instatiations
    ic1 = exe4(ledr_s, SW)
    # ic2 = exe5(ledr_s, SW)
    # ic2 = sw2hex(HEX0, SW)
    # ic3 = bin2hex(HEX1, SW)

    @always_comb
    def comb():
        for i in range(len(LEDR)):
            LEDR[i].next = ledr_s[i]

    return instances()
```

Rodando pelo Docker

Rodar o comando docker para gerar o RBF:

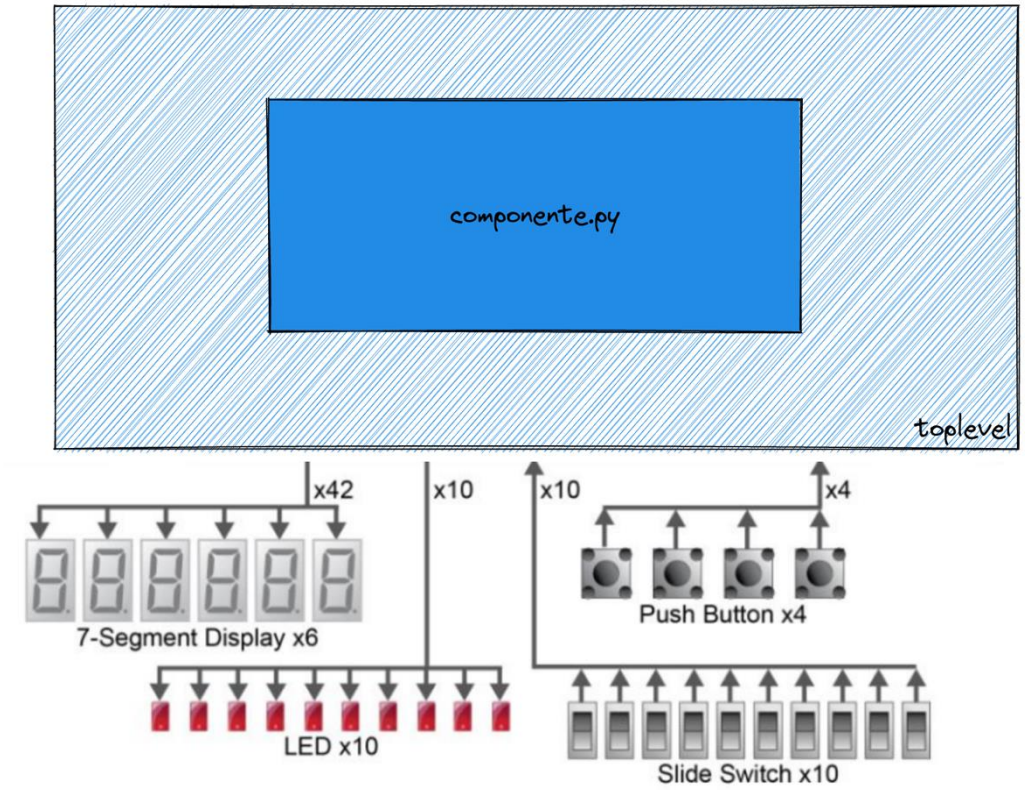
```
docker run --rm \
  -v "${PWD}:/work" \
  -w /work \
  rafaelcorsi/bits \
  bash -lc "make clean && make toplevel.rbf"
```

Programar o arquivo:

- Abra o fpgaloader
- Selecione o arquivo .rbf
- Programar a placa

Insper

Layout DE0-CV



- ***LED***: 10 leds que acendem com lógica ***1***
- ***Push Buttons***: 4 botões que quando apertados fornecem lógica ***0***
- ***Slide Switchs***: 10 Slides que quando acionados forcem lógica ***1***
- ***HEX Displays***: 6 displays de 7 segmentos (anodo comum)

Insper

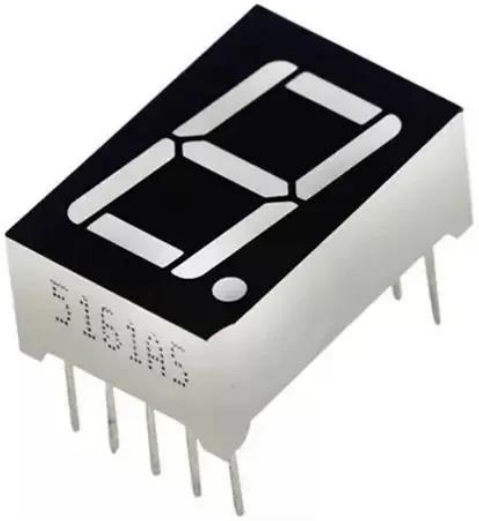
Praticando na FPGA

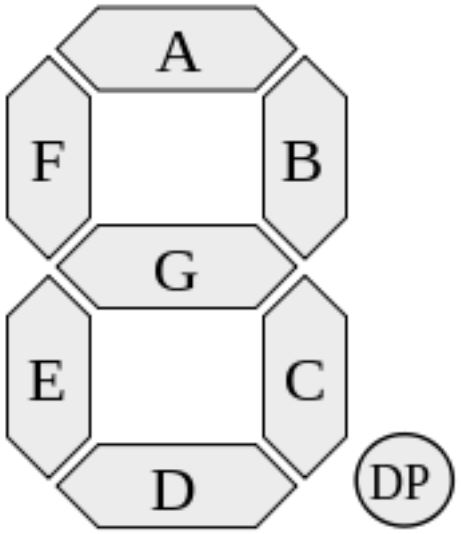
Continue com o arquivo ***comb_modules.py***

Desenvolva os códigos do **exe5** e **sw2hex**

Teste na FPGA

Display de 7 segmentos





Obs: Cuidado a ordem pode parecer invertida.

Insper

www.insper.edu.br