FEDERAL UNIVERSITY
OF SANTA CATARINA

Laboratório 1:
Introdução ao Quartus II e DE1-SOC

EEL5105 – Circuitos e Técnicas Digitais

Objetivos

- Introduzir o laboratório de **EEL5105**.
- Introduzir a **plataforma** utilizada nas aulas de laboratório:
  - **Kit DE-1 SOC;**
  - **FPGA Altera Cyclone V 5CSEMA5F31C6N;**
  - **Software Quartus II;**
  - **Software ModelSim.**
- Conhecer as entradas e saídas digitais usadas na placa DE-1 SoC.

EEL5105 – Circuitos e Técnicas Digitais

Introdução ao Laboratório

Tarefa

Tarefa Adicional

EEL5105 – Circuitos e Técnicas Digitais

- Carga horária semanal
  - 2 horas-aula de **teoria**
  - 3 horas-aula de **laboratório**
- É oferecida para os seguintes cursos da **UFSC**:
  - Ciência da Computação
  - Engenharia Elétrica
  - Engenharia Eletrônica
  - Engenharia de Controle e Automação
  - Engenharia de Produção Elétrica

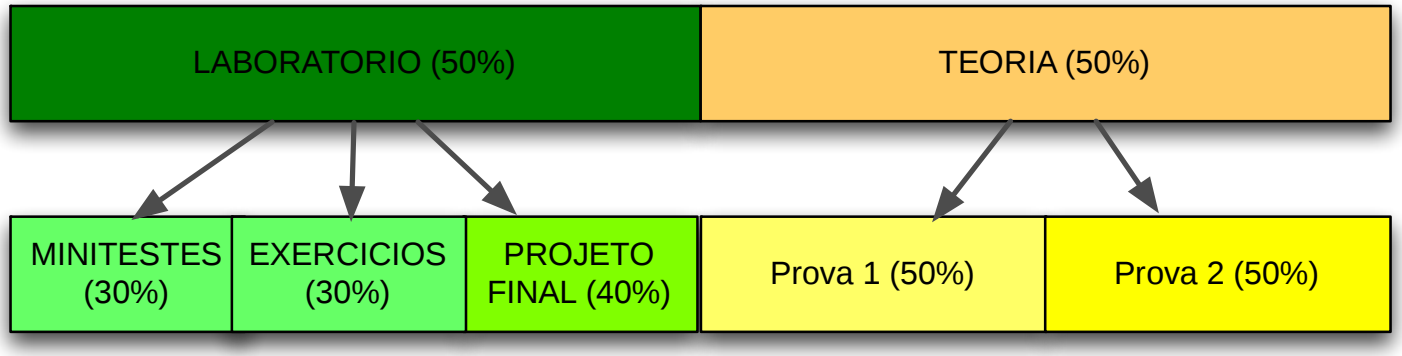
EEL5105 – Circuitos e Técnicas Digitais

- **Bibliografia Básica (disponíveis a partir do site da biblioteca da UFSC):**
  - ***Sistemas Digitais: Projeto, Otimização e HDLs***
    Frank Vahid, 1a Edição, Bookman, 2007
    - http://www.bu.ufsc.br/framebases.html opção **Minha Biblioteca**
    - Se, após o login na **Minha Biblioteca**, o livro não aparecer na lista de livros, use o seguinte link: http://integrada.minhabiblioteca.com.br/books/9788577802371

  - ***Sistemas Digitais: Princípios e Aplicações***
    Ronald J. Tocci, Neal S. Widmer e Gregory L. Moss
    10a Edição, Pearson Prentice Hall, 2007

EEL5105 – Circuitos e Técnicas Digitais

- **Estágiários de docência:** em breve no site.
- **Monitoria:** em breve no site.
- **Avaliação:**



EEL5105 – Circuitos e Técnicas Digitais

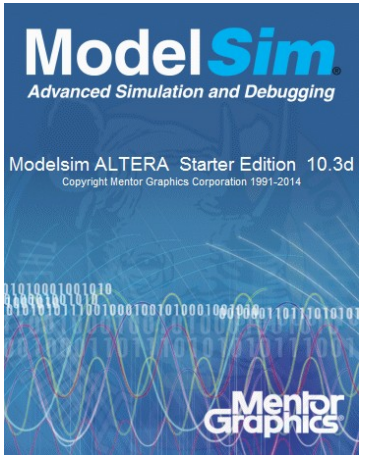
Introdução ao Laboratório

Tarefa

Tarefa Adicional

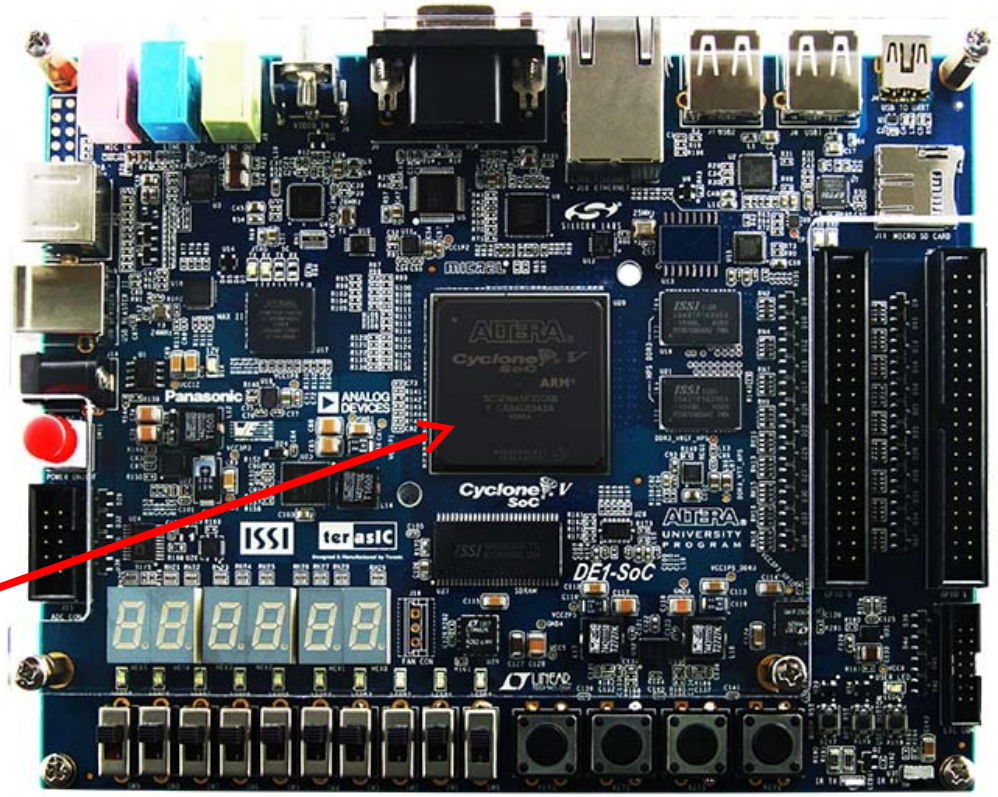
Introdução ao Laboratório

- **Ferramentas Utilizadas:**
  - **DE1-SOC**
    - Altera Cyclone V **5CSEMA5F31C6N**
  - Software **Quartus II** (Projeto)
  - Software **ModelSim** (Simulação)
  - Linguagem **VHDL**



Introdução ao Laboratório

- **DE1-SOC**

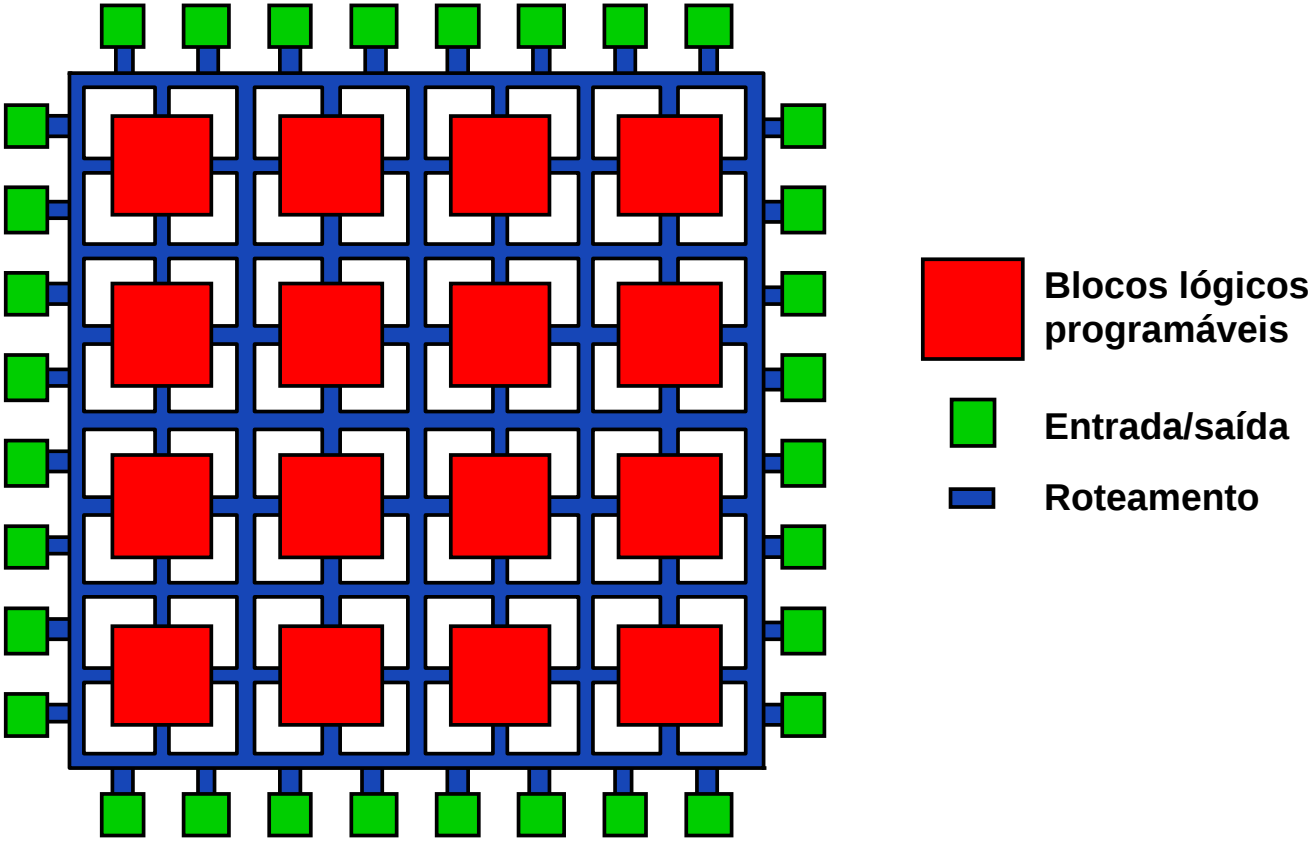


FPGA
Altera Cyclone V
5CSEMA5F31C6N

- Mais informações: **http://de1-soc.terasic.com**

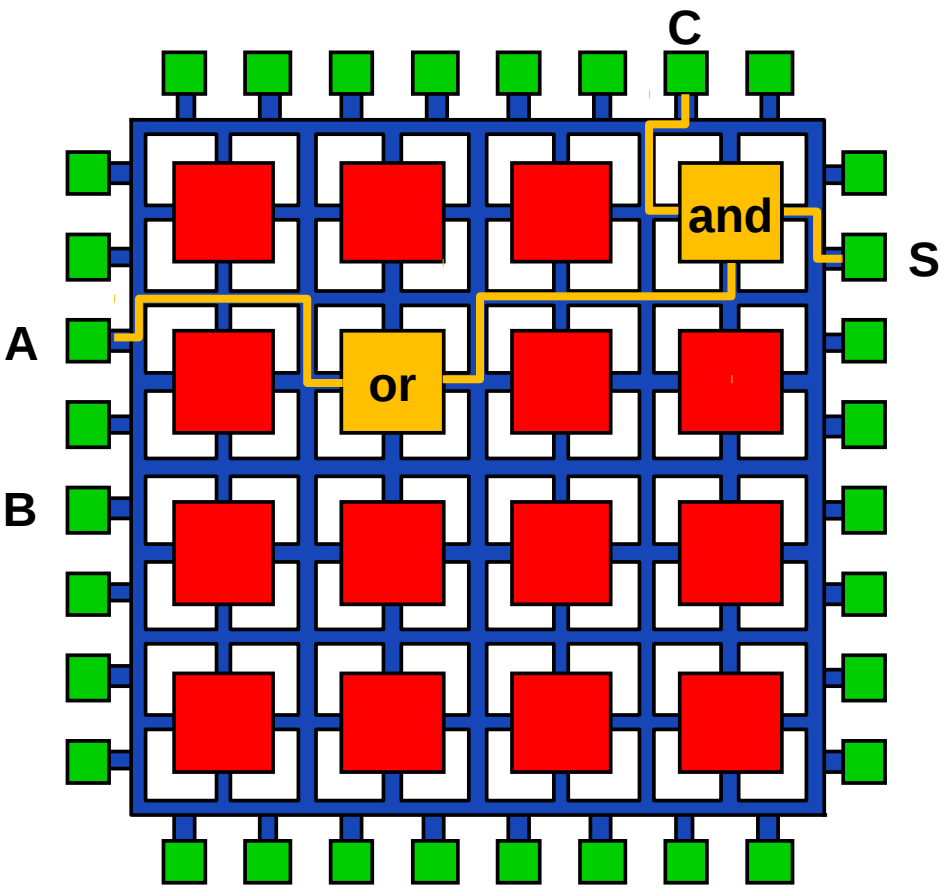
Introdução ao Laboratório

- Estrutura interna de um **FPGA**:



Introdução ao Laboratório

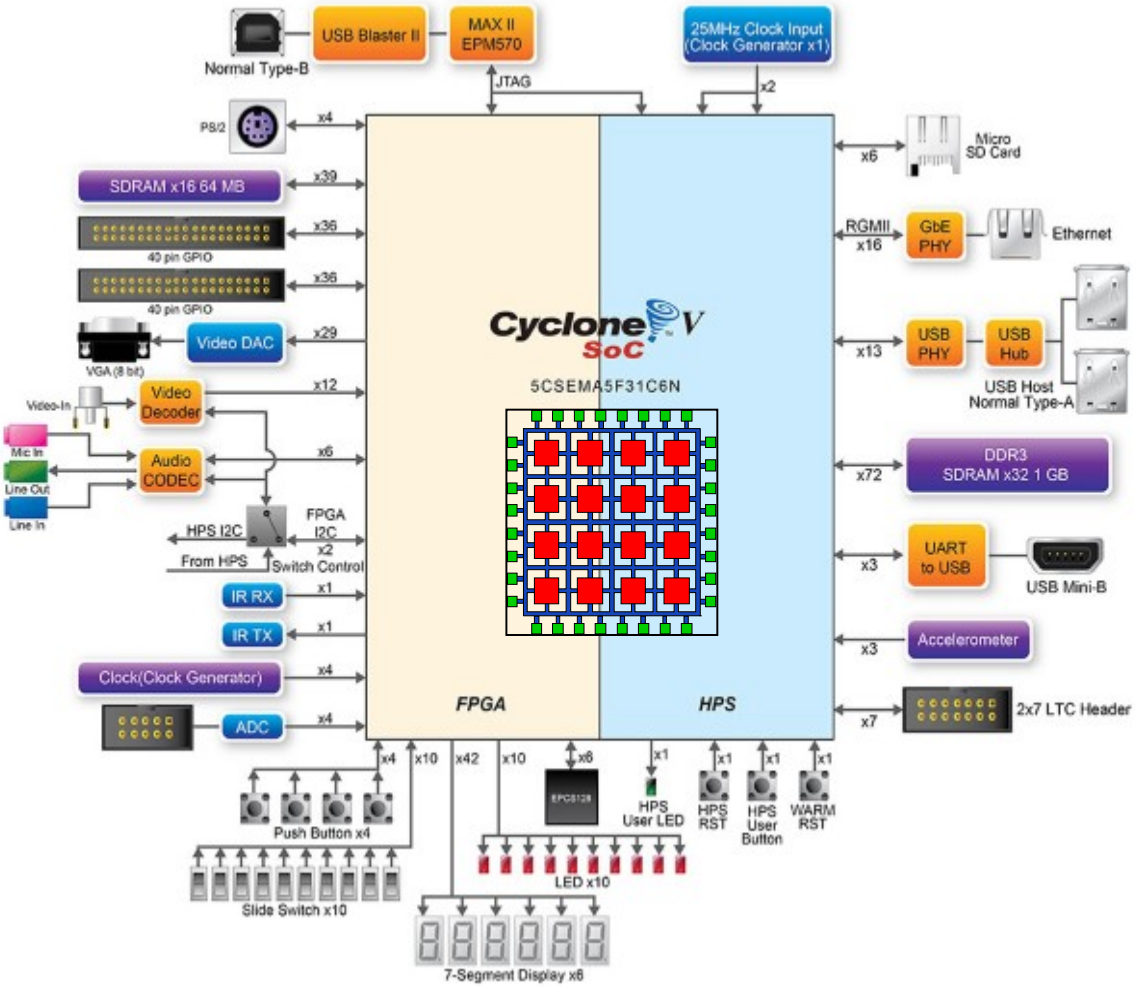
- Estrutura interna de um **FPGA**:



S = (A+B).C

Introdução ao Laboratório

- No **DE1-SoC**, os pinos do **FPGA** estão ligados em **chaves**, **botões**, **leds**, **displays de 7 segmentos**, **etc**, permitindo que estímulos sejam enviados ao **FPGA** e saídas sejam observadas de diferentes formas.



Atividade 1: LEDs

Enviar valores lógicos (‘0’ ou ‘1’) aos LEDs na placa.

Dicas

1. Note que o menú **File** tem opções diferentes para o Quartus project (**New Project**, **Open Project**, **Close Project**, **Save Project**) e para os arquivos (**New**, **Open**, **Close**, **Save**)

2. Quando definir o nomes de pastas, projeto ou arquivos, nunca use caracteres especiais como **espaços** ou **ç** ou **acentos**.

3. Seja organizado, salve seu “Quartus project” apenas na pasta da sua turma correspondente ao laboratório do dia.

Tarefa

- **Passo 1**: Criar projeto no **Quartus II**
  - Acessar **File -> New Project Wizard** e criar um projeto com as seguintes características:
    - **Pasta**: Escolher pasta com o nome da turma e com o nome da prática atual dentro do ***Desktop***
    - **Sugestão de nome do Projeto**: **Aula1**
    - **Tipo**: *Empty Project*
    - Em ***Add Files***, não adicionar nenhum arquivo
    - Na escolha do dispositivo:
      - Família: **Cyclone V**
      - Dispositivo: **5CSEMA5F31C6**
    - Demais opções deixar como padrão

Tarefa

- **Passo 2**: Criar arquivo do tipo **VHDL** dentro do projeto
  - No **Quartus II**, acessar
    File -> New -> Design Files -> VHDL File
    e o arquivo "***Vhdl1.vhd***" será criado

  - Em seguida, clique em **File -> Save As** e salve seu arquivo com algum nome desejado (sugestão: "***circuito1.vhd***").
  - Com seu arquivo aberto, clique em
    Project -> Set As Top Level Entity
    para que o arquivo criado se torne o principal do projeto.

Os detalhes sobre a linguagem de descrição de Hardware (VHDL - VHSIC Hardware Description Language) serão explicados na próxima aula.

Nesta aula usaremos um modelo de arquivo VHDL pronto para focar no uso de entradas e saídas da placa DE1-SoC.

Tarefa

- **Passo 3**: Escreva as seguintes líneas de código VHDL (pode copiar elas do arquivo “circuito1.vhd” disponível no moodle):

circuito1.vhd

```
1  library IEEE;
2  use IEEE.Std_Logic_1164.all;
3
4
5  entity circuito1 is
6  port (
7  LEDR: out std_logic_vector(9 downto 0)
8  );
9  end circuito1;
10
11
12 architecture arc_circuito1 of circuito1 is
13 begin
14
15 LEDR(9) <= '1';
16 LEDR(8) <= '0';
17 LEDR(7) <= '1';
18
19 end arc_circuito1;
20
21
```

Tarefa

- **Passo 4**: Fazer associação de **entradas/saídas** com **chaves/leds** do DE1-SOC usando arquivo externo com ***assignments***
  - Baixe o arquivo **Pinos.qsf** disponível no Moodle
    - Esse arquivo contém a descrição de relações entre os nomes que você usou no seu projeto [**SW, LEDR, etc**] e os **pinos** do **FPGA**.
  - No **Quartus II**, acesse **Assignments -> Import Assignments...** e faça a importação de **Pinos.qsf**.

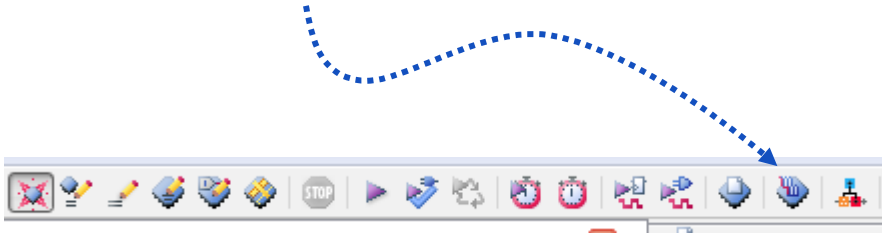
Tarefa

- **Passo 5**: Compilar o projeto.
  - Clique então em **Processing -> Start Compilation** para fazer a síntese do seu projeto

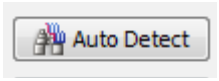
Tarefa

- **Passo 6**: Prototipação no **FPGA**

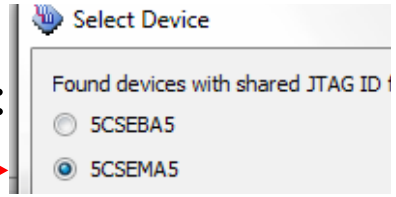
  - A seguir, com o kit **DE1-SOC** ligado à tomada e ao computador, acesse **Tools -> Programmer** no **Quartus** para gravar seu projeto



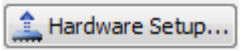
Tarefa

- **Passo 6**: Prototipação no **FPGA**
  - Com o **Programmer** aberto:
    1. Clique em 

e selecione a 2ª opção:

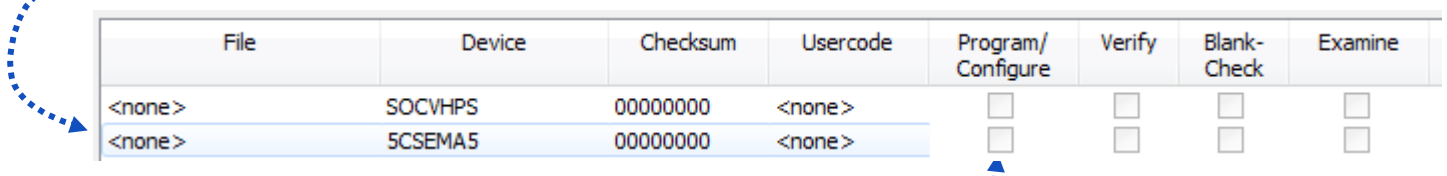


– Caso **Auto Detect** esteja desativado (

 ), clique em 

e selecione **DE1-SoC** como hardware a ser utilizado.

- A seguir, clique com o **botão da direita** do mouse no **segundo elemento** da lista de arquivos e escolha ***Change File***



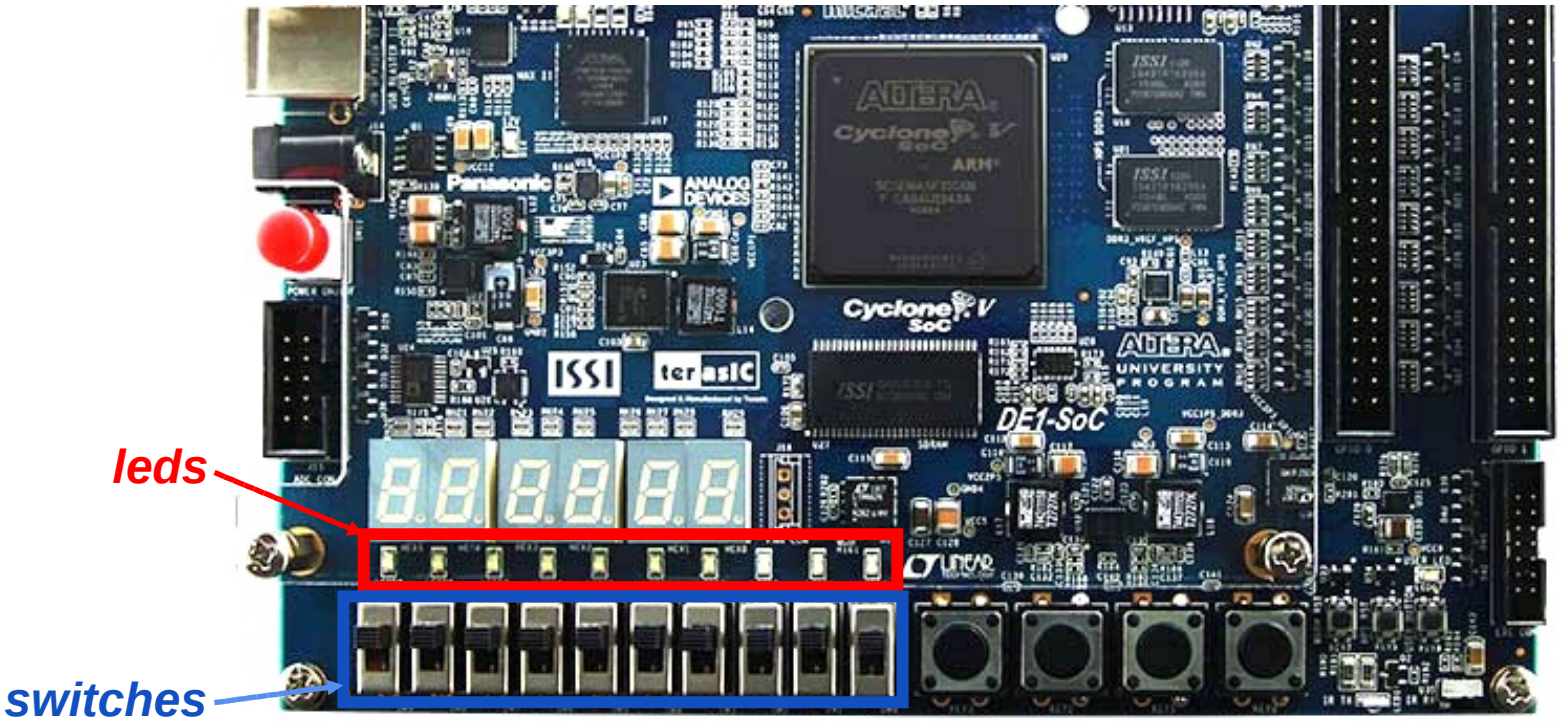
File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine
<none>	SOCVHPS	00000000	<none>	☐	☐	☐	☐
<none>	5CSEMA5	00000000	<none>	☐	☐	☐	☐

- Selecione o arquivo com extensão **.sof** da pasta **output_files**

- Marque a caixa **Program/Configure** e clique em

Tarefa

- **Passo 7**: Verificar o resultado na placa. Observar os ***leds*** que foram ativados. Cada componente na placa tem o nome impresso ao lado, por exemplo LEDR9, LEDR8 ...



Tarefa

- **Passo 8**: Modifique o código VHDL para que os LEDs mostrem os seguintes valores lógicos:

1 0 1 1 0 1 0 1 1 1

- Salve o arquivo, compile e prototipe novamente no FPGA.

Tarefa

Passo 9: Crie um novo arquivo com o nome “circuito2.vhd” seguindo os mesmos passos (passo 2 ao passo 7). Escreva as líneas de código VHDL mostradas abaixo (pode copiar elas do arquivo “circuito2.vhd” disponível no moodle). Complete para que os valores lógicos de todas as entradas (chaves) sejam transferidos a todas as saídas (LEDs).

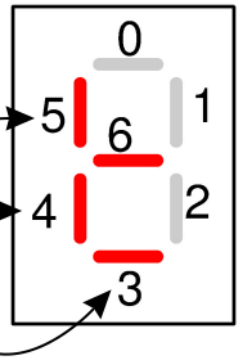
circuito2.vhd

```
1   library IEEE;
2   use IEEE.Std_Logic_1164.all;
3
4
5   entity circuito2 is
6   port (
7   SW:   in    std_logic_vector(9 downto 0);
8   LEDR: out   std_logic_vector(9 downto 0)
9   );
10  end circuito2;
11
12
13  architecture arc_circuito2 of circuito2 is
14  begin
15
16  LEDR(9) <= SW(9);
17  LEDR(8) <= SW(8);
18  LEDR(7) <= SW(7);
19
20  end arc_circuito2;
21
```

Tarefa Avançada

Tarefa Avançada

- Escreva seu nome usando os 6 displays de 7-segmentos

circuito3.vhd

```
1  library IEEE;
2  use IEEE.Std_Logic_1164.all;
3
4
5  entity circuito3 is
6  port (
7   HEX5: out   std_logic_vector(6 downto 0);
8   HEX4: out   std_logic_vector(6 downto 0);
9   HEX3: out   std_logic_vector(6 downto 0);
10  HEX2: out   std_logic_vector(6 downto 0);
11  HEX1: out   std_logic_vector(6 downto 0);
12  HEX0: out   std_logic_vector(6 downto 0)
13  );
14  end circuito3;
15
16
17 architecture arc_circuito3 of circuito3 is
18 begin
19
20  -- Mostra a letra t no display HEX0
21  HEX5(6) <='0';    --ligado
22  HEX5(5) <='0';    --ligado
23  HEX5(4) <='0';    --ligado
24  HEX5(3) <='0';    --ligado
25  HEX5(2) <='1';    --desligado
26  HEX5(1) <='1';    --desligado
27  HEX5(0) <='1';    --desligado
28
29  -- Completar para os outros displays HEX4, HEX3 ... HEX0
30
31  end arc_circuito3;
```