

# Elementos de Sistemas

## Aula 5 – Lógica Combinacional

"Conversamos como vivemos ao repetir, combinar e recombinar alguns elementos repetidamente, assim como a natureza faz quando, de partículas elementares, constrói um mundo."

"We converse as we live by repeating, by combining and recombining a few elements over and over again just as nature does when of elementary particles it builds a world."

William Howard Gass, Escritor Americano

Engenharia - Luciano Soares <[lpsoares@insper.edu.br](mailto:lpsoares@insper.edu.br)>



## **Problema :**

Um dos grupos está com 4  
integrantes os demais com 5 ou  
6 !

# Objetivo da Aula

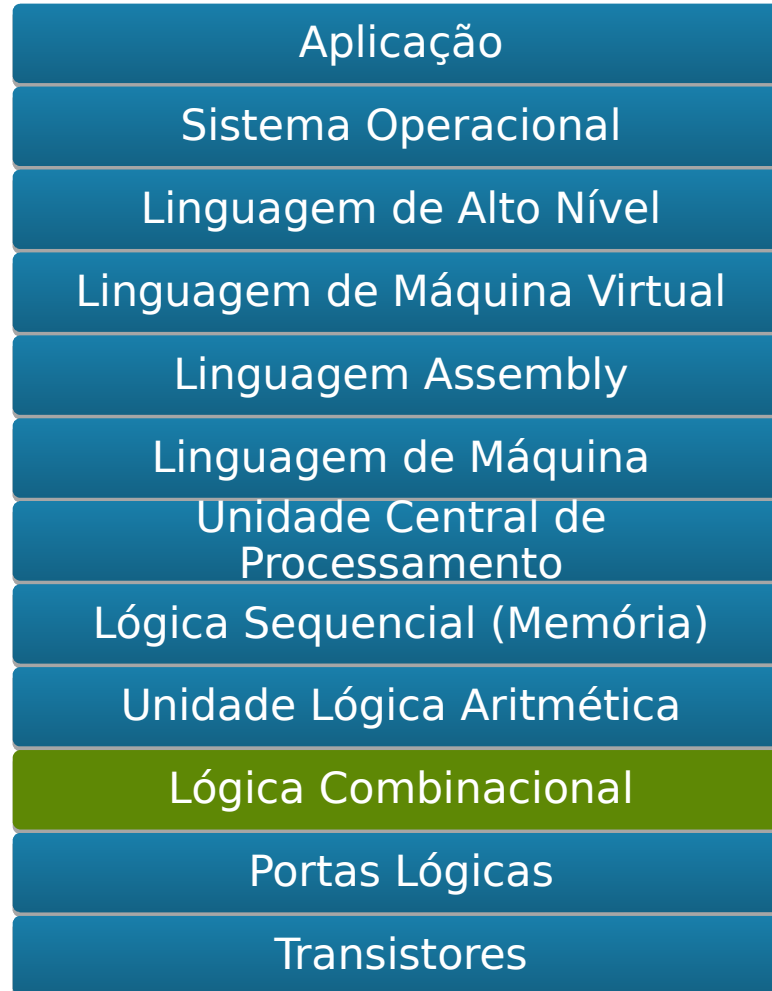
- Montar circuitos lógicos mais complexos;
- Programar em Linguagens de Descrição de Hardware;
- Simular lógicas para Dispositivos Lógicos Programáveis.
- Publicar e colaborar em repositórios distribuídos;

**Conteúdo(s):** Dispositivos Lógicos Programáveis;

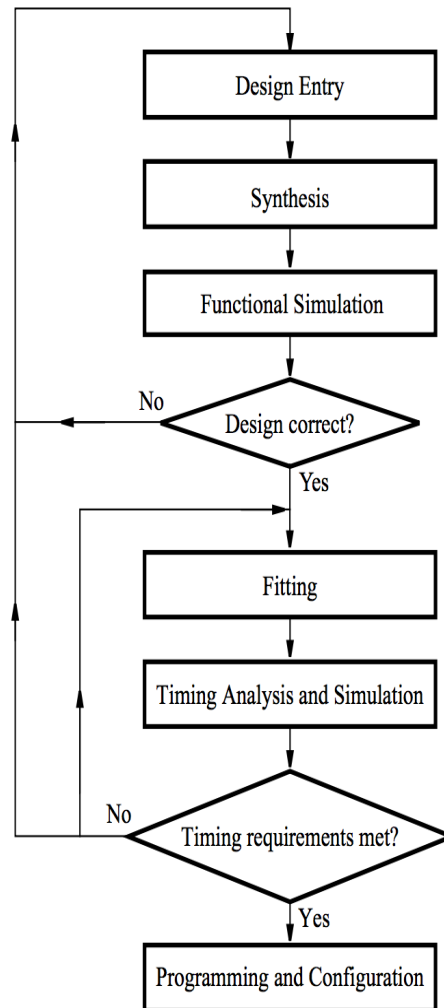
# Nessa aula você já deveria ter

- 1) o repositório 3s-Elementos-De-Sistemas no seu PC
  - 1) Ter ele atualizado a cada aula
- 2) O fork do repositório Z01 clonado no seu PC
  - 1) Referenciado o repositório para o upstream
  - 2) Apreendido a atualizar o repositório
- 3) Instalado a infra necessária para a continuação do curso
  - 1) Python + Quartus + Java
  - 2) Validado a instalação com o script *testeVHDL.py*

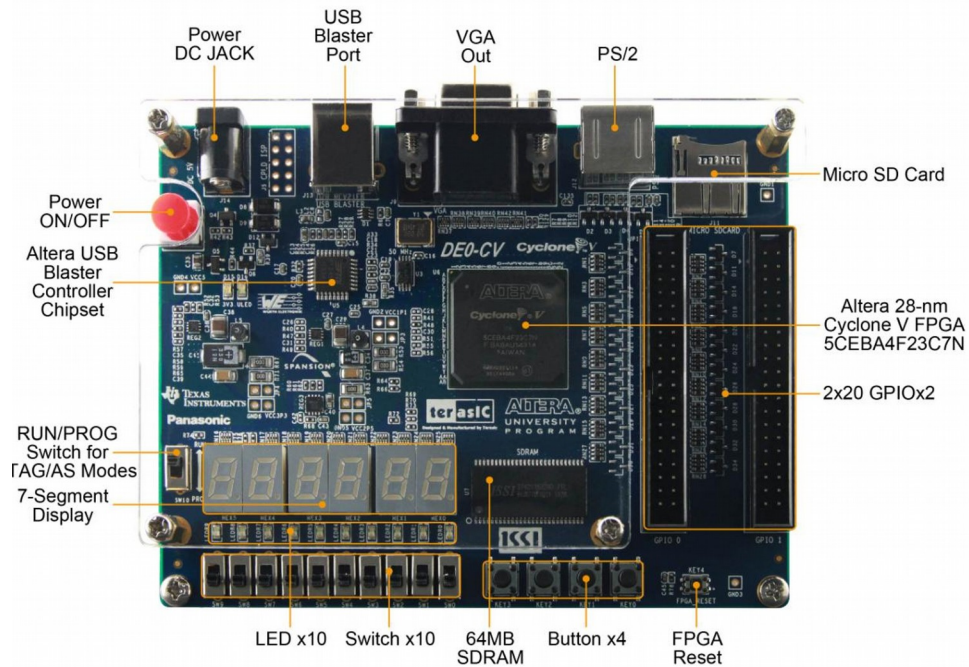
# Níveis de Abstração



# Processo de programação de FPGA



# 1 – FPGA



Handout 5 :  
treinar para o projeto !





# PROJETO C – Lógica Combinacional

# Projeto C

Definam quem vai ser o facilitador (Scrum Master) da Sprint.

Fazer planejamento da Sprint, com suas dependências e etc. **Façam Juntos.**

Definir claramente a dinâmica do grupo e o que cada um vai fazer, ou seja, aperfeiçoar o processo de trabalho em equipe aqui.

# Projeto C

Implemente as portas lógicas do projeto. Já estão presentes no repositório Git um esqueleto de cada entidade .vhd

## Multi-way

- Multiplexador de 2 entradas
- Multiplexador de 4 entradas
- Demultiplexador de 2 saídas
- Demultiplexador de 4 saídas
- Demultiplexador de 8 saídas
- Or8way
- Nor8way

## Barramento:

- Not16
- And16
- Or16
- Mux16
- BarrelShifter8
- BarrelShifter16

## Mistas

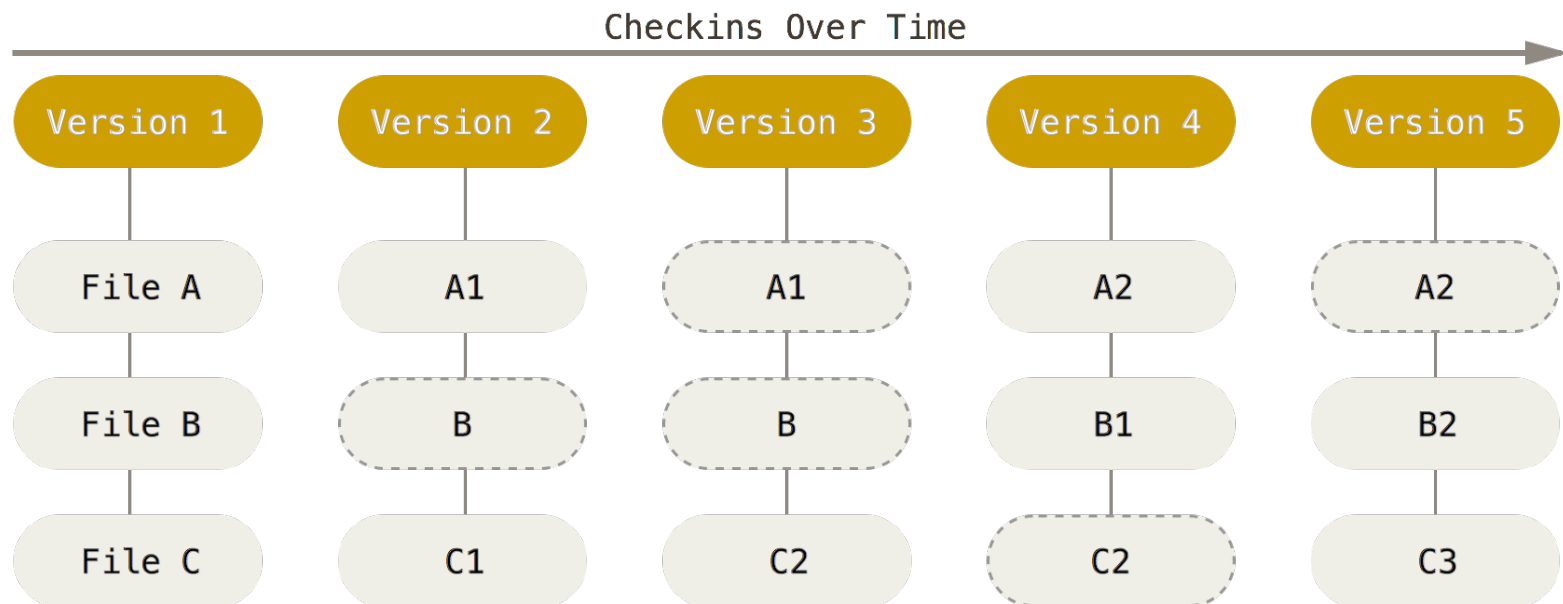
- Mux4Way16
- Mux8Way16

# Papeis nos grupos

- Facilitador (scrum master) :
  - Organizar o github + issues + project
  - Fazer a atualização do fork com o upstream
  - Gerenciar o grupo (atribuir tarefas)
  - Aceitar os pull-requests
  - Criar relatório da performance de cada um do grupo
  - Entregar o projeto no final
- Desenvolvedores :
  - Realizar as tarefas atribuidas pelo scrum-master
  - Realizar os pull-requests
  - Testar os códigos

# Git é o caminho

O Git é um repositório de códigos descentralizado gratuito. Git é capaz de armazenar todas as mudanças nos diretórios gerenciados de forma distribuída.

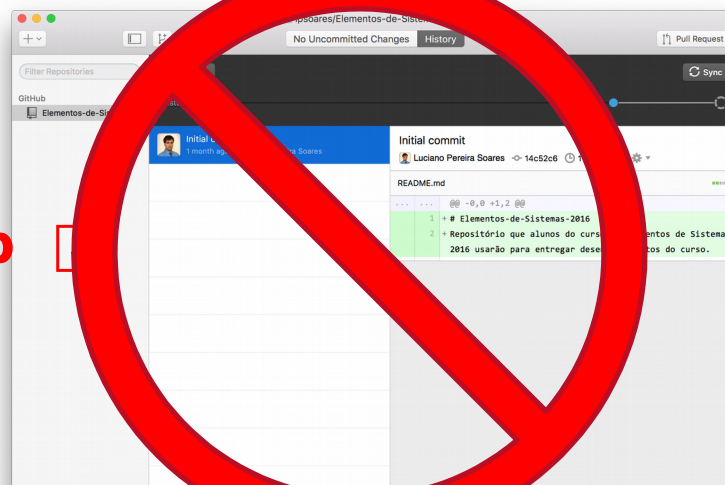


# Forma de trabalhado com o Git

Linha de comando:

```
Elementos-de-Sistemas-2016 — -bash — 80x24
[[lpsoares:~] $ git clone https://github.com/lpsoares/Elementos-de-Sistemas-2016
Cloning into 'Elementos-de-Sistemas-2016'...
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 3
Unpacking objects: 100% (3/3), done.
Checking connectivity... done.
[[lpsoares:~] 2s $ cd Elementos-de-Sistemas-2016/
[[lpsoares:~/Elementos-de-Sistemas-2016] master ± ls
README.md
[[lpsoares:~/Elementos-de-Sistemas-2016] master ±
```

**Não usem o Github Desktop**

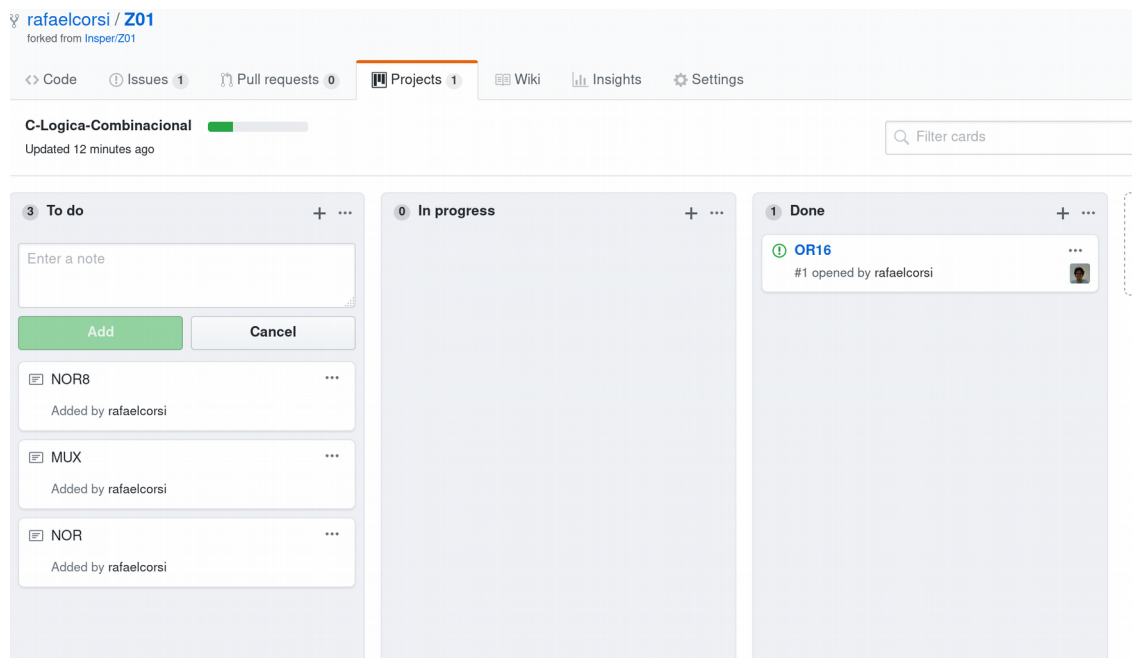


# FORK

- Cada grupo deve possuir apenas um FORK !
- O fork deve ser atualizado com o upstream a cada novo projeto !
- Todos do grupo devem possuir acesso de escrita ao FORK.

# Ajustem o Kanban board seu grupo

Criem as tarefas no Kanban (project-githubb) para se organizarem. Cada dupla, move a tarefa Todo para Doing, scram master deve mover a tarefa para Done só quando aceito o pull-request.



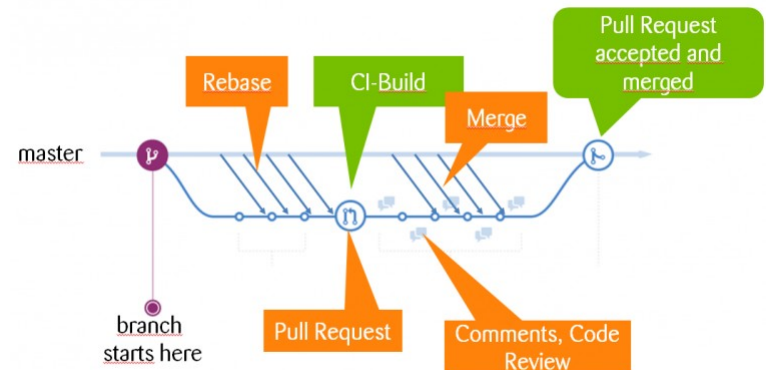
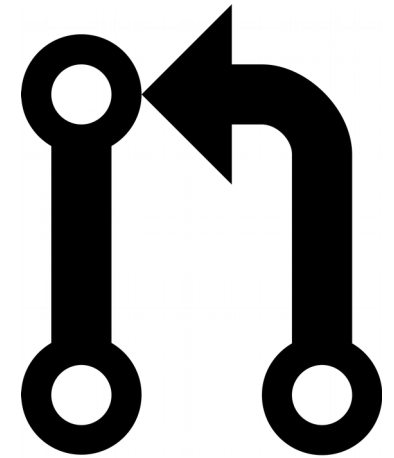


# Lembretes

- Desenvolvedores :  
Testem + testem + testem
  - Antes de realizarem o pull-request !
  - Implementa nos branches
- Scrum Master

Teste + Teste + Teste

- Antes de aceitar o pull-request !
- Gerencia o master



# Lembretes

- Desenvolver no diretório do GIT.
- Todos os arquivos estão no formato de texto, o que simplifica seu uso com o git.
- Cuidado com os arquivos de projeto do Quartus:
  - .qpf: arquivo que armazena o projeto;
  - .qsf:
    - armazena as informações dos arquivos do projeto
    - FPGA em uso, no caso da DE0-CV
    - Da conexão de pinos
    - Etc

# Tutorial disponível !

- Fiz um tutorial que mostra passo a passo como realizar essas etapas, mas a literatura é ampla !s

<https://www.youtube.com/watch?v=oFYyTZwMyAg>

<https://www.youtube.com/watch?v=FQsBmnZvBdc>

<https://help.github.com/articles/creating-a-pull-request/>

# Próxima Aula

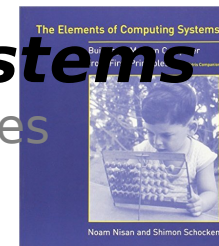
- Ver estudo para aula 6 sobre **Dados Digitais**
- Estudar Lista de Exercícios Aula 5 (opcional):
- Ler (opcional)

## Capítulo X

### ***The Elements of Computing Systems***

Building a Modern Computer from First Principles

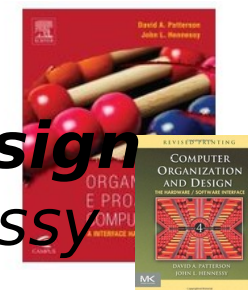
*Noam Nisan e Shimon Schocken*



## Capítulo X

### ***Computer Organization and Design***

*David A. Patterson e John L. Hennessy*



# Insper

[www.insper.edu.br](http://www.insper.edu.br)