



# **Aula 14 – Implementando um processador didático**



## Tópicos da aula

- Caminho de Controle
- Caminho de Dados
- Estrutura de um processador didático



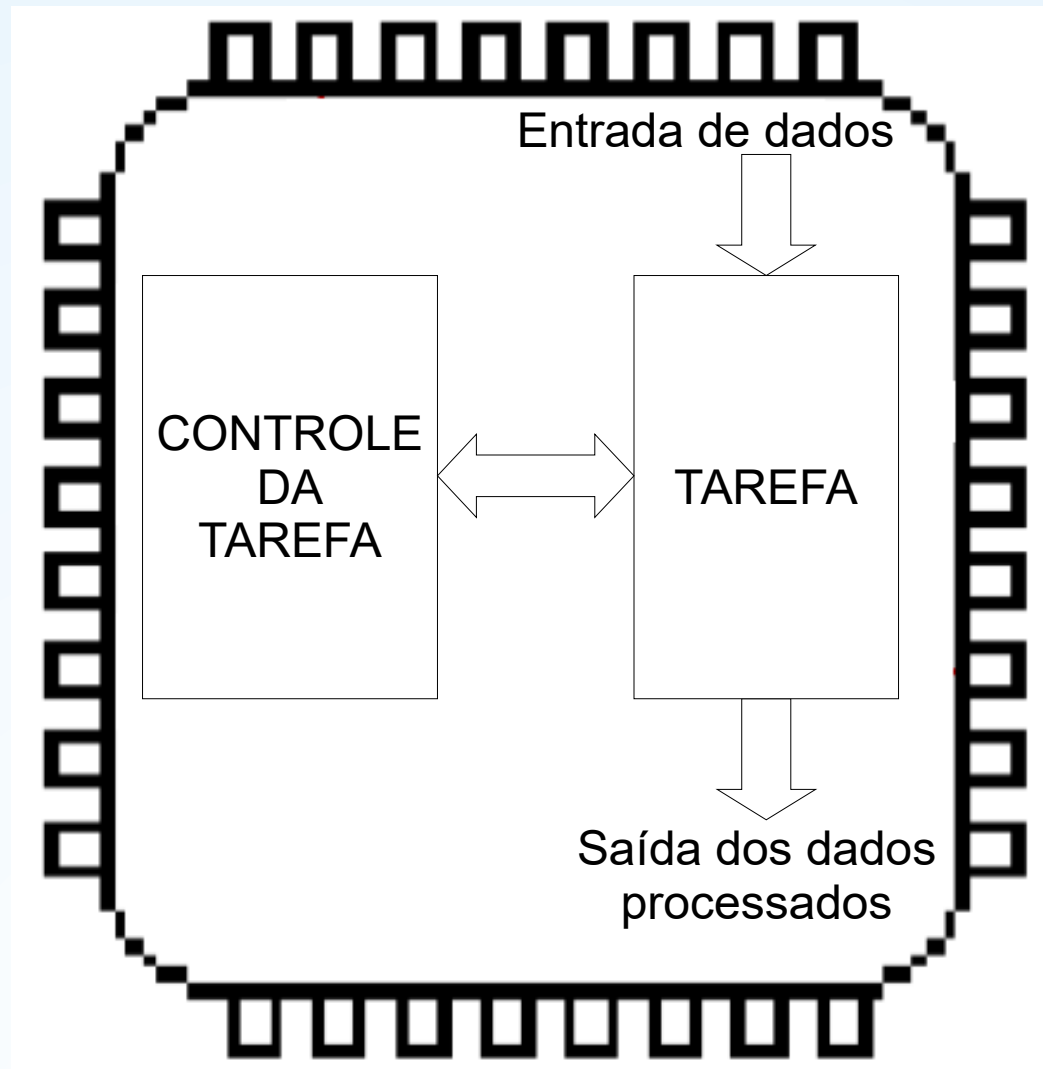
## Introdução

- Dispositivos lógicos programáveis como o FPGA são muito usados para realizar **tarefas intensas**
  - Co-processamento
  - Processamento de sinais
- Tipicamente o projeto de um sistema precisa levar em conta o **fluxo dos dados** ao serem processados
- Este “**fluxo de dados**” precisa ser **controlado** de alguma forma



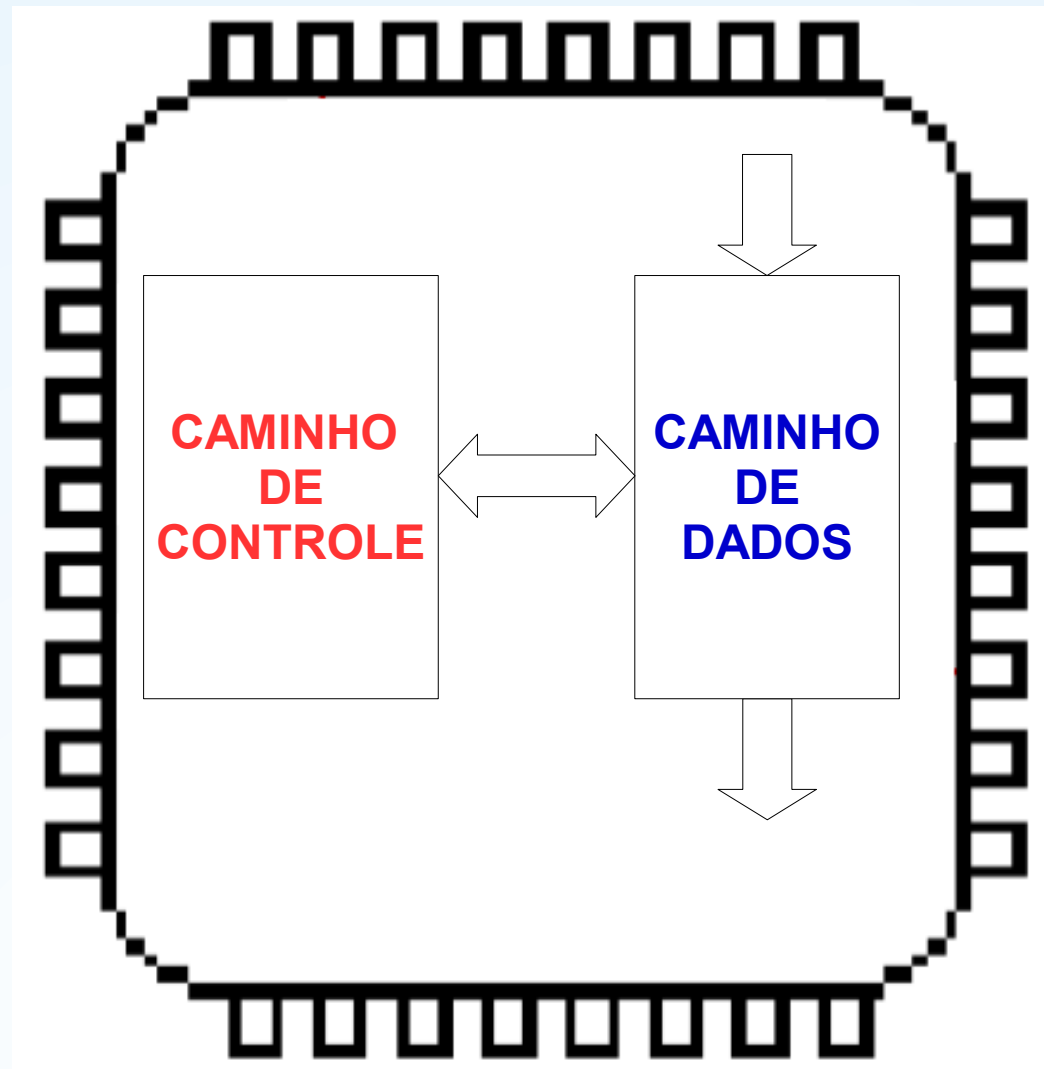
Universidade Federal  
de Santa Catarina

## Tarefas executadas em um FPGA





## Caminhos de Dados e Controle





## Caminhos de Dados e Controle

- O projeto de um sistema Digital complexo deve **iniciar** pelo **Caminho de Dados**
- O **Caminho de Controle** é a **última** tarefa a ser realizada



## Construindo um processador

- A construção de um **processador** também segue esta filosofia de ter um **Caminho de Dados** e outro de **Controle**
- A primeira medida a ser tomada é a definição da **Arquitetura**
- Uma vez definida a Arquitetura do Processador, define-se a sua **Organização**



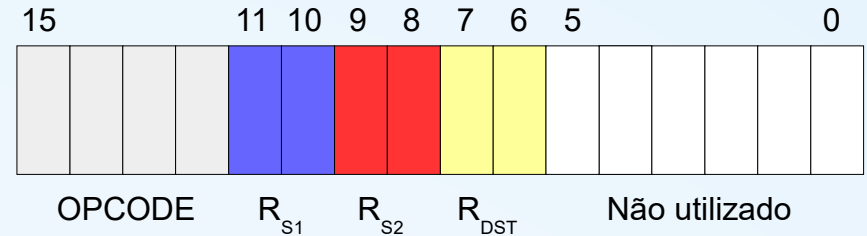
## Definindo uma Arquitetura

- A primeira medida a ser tomada é a definição da **Arquitetura**. Vamos definir um processador do seguinte modo:
  - Processador terá **4 registradores** de uso geral
  - Os registradores serão de **16 bits**
  - Inicialmente o processador irá executar **Soma** e **Subtração**
  - Comandos Assembly: Carregar valor imediato (LDI); Somar (ADD); Subtrair (SUB)

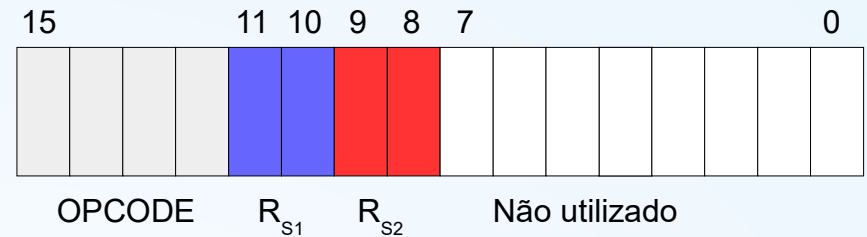




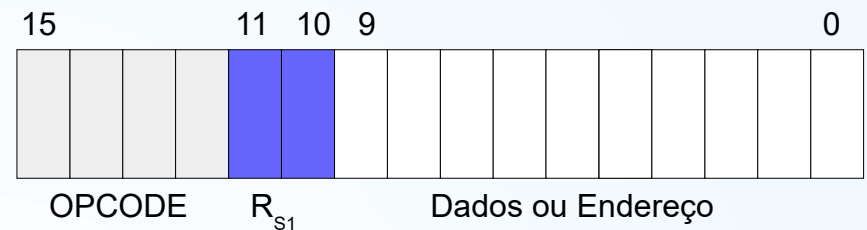
## Formato Registradores



(Tipo III)



(Tipo II)



(Tipo I)



Universidade Federal  
de Santa Catarina

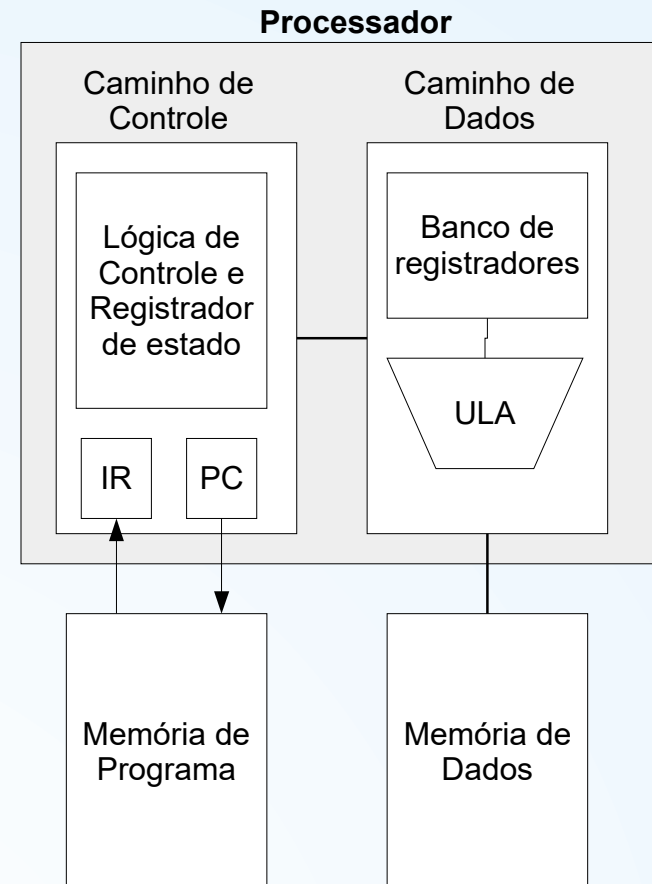
## Definindo uma Arquitetura

Mnemônico	Opcode (binário)	Tipo	Descrição
<b>STOP</b>	0000	na	"No operation". Processador gasta um ciclo de relógio se fazer nenhuma operação.
<b>LDI</b>	0001	I	Carrega valor imediato contido na instrução (bits de 0 a 9) em um determinado Registrador ( $R_{S1}$ )
<b>ADD</b>	0010	II	Soma o conteúdo de dois registradores e armazena o resultado em um terceiro registrador.
<b>SUB</b>	0011	III	Subtrai o conteúdo de dois registradores e armazena o resultado em um terceiro registrador.
<b>OUT</b>	0100	I	Escreve o valor contido em um registrador ( $R_0$ , $R_1$ , $R_2$ ou $R_3$ ) nos pinos de I/O.



## Definindo uma Organização

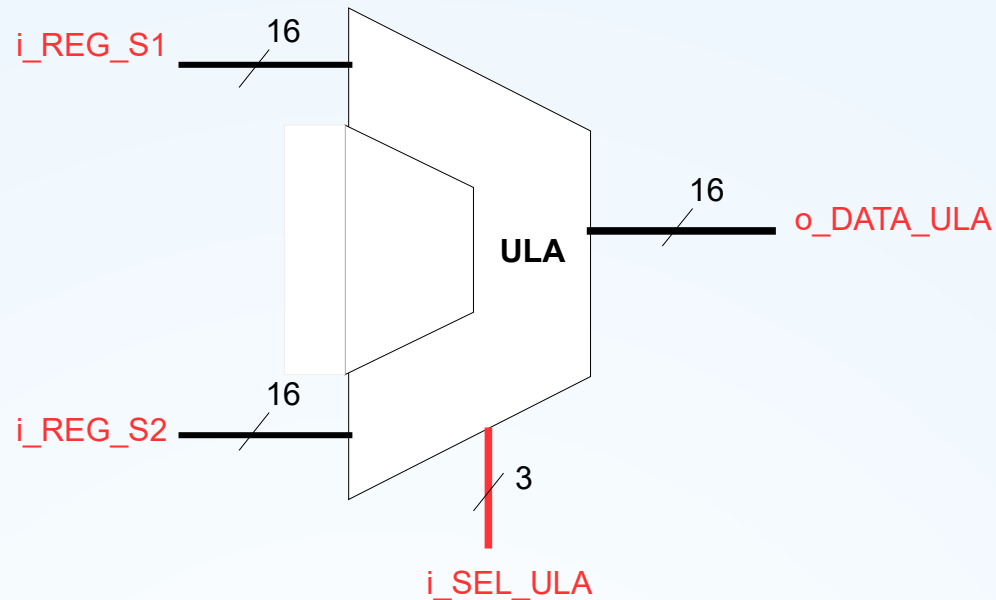
- Agora que temos a **Arquitetura** definida, é hora de definir a **Organização** do **processador**, começando pelo **Caminho de Dados**
- O **Caminho de Dados** é basicamente constituído por um **Banco de Registradores** e uma **Unidade Lógica e Aritmética (ULA)**
- Demais registradores (IR e PC) serão implementados mais tarde





Universidade Federal  
de Santa Catarina

## Unidade Lógica e Aritmética (ULA)





## Programa exemplo ( $R_0 = 2 + 3$ )

LDI	$R_1, 2$	0001 0100 0000 0010	1402h
LDI	$R_2, 3$	0001 1000 0000 0011	1803h
ADD	$R_0, R_1, R_2$	0010 0001 1000 0000	2180h
OUT	$R_0$	0100 0000 0000 0000	4000h
STOP		0000 0000 0000 0000	0000h



Universidade Federal  
de Santa Catarina

## **FIM AULA XIV**