

# Organização e Arquitetura de Computadores II

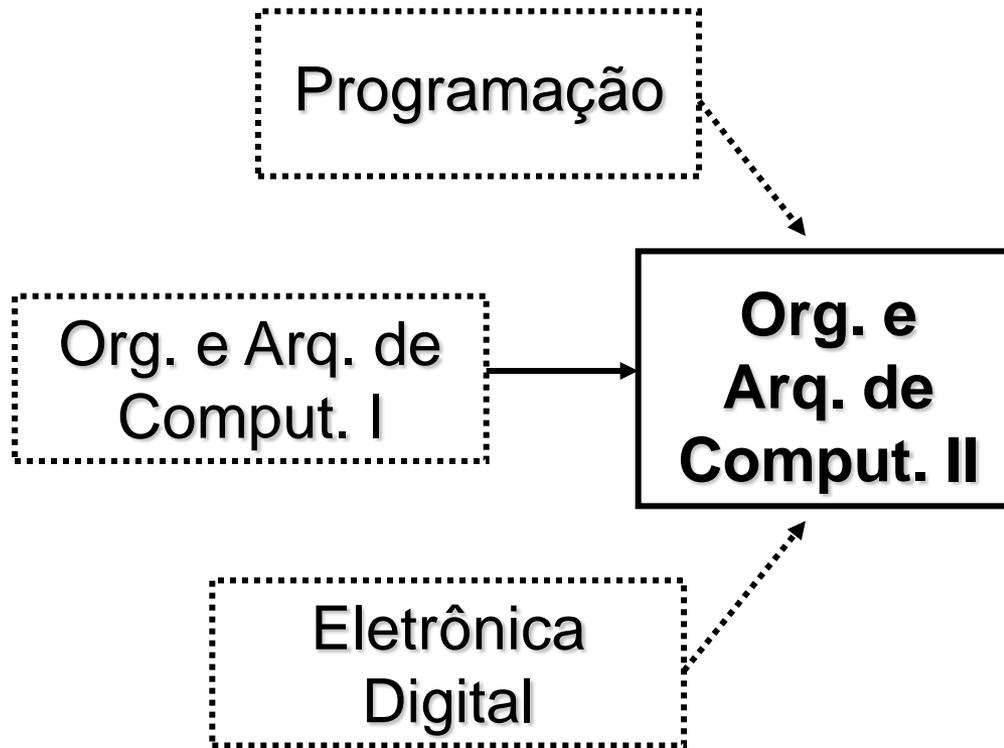
Rômulo Calado Pantaleão Camara

[Romulo.camara@univasf.edu.br](mailto:Romulo.camara@univasf.edu.br)

[www.univasf.edu.br/~romulo.camara](http://www.univasf.edu.br/~romulo.camara)

60h/02h

# Disciplinas Correlacionadas



# Ementa

- ✓ Linguagem de descrição de *hardware*.
- ✓ Máquinas paralelas:
  - Processadores virtuais;
  - Sistemas multiprocessados;
  - Multicomputadores.
- ✓ Sistemas multiprogramados.
- ✓ Programação, comunicação e sincronização em máquinas paralelas e sistemas multiprogramados.
- ✓ Superpipeline.
- ✓ Arquiteturas superescalares.
- ✓ Arquiteturas VLIW.

# Objetivos

## ✓ Geral:

- Fornecer ao aluno conhecimentos adicionais sobre a organização e arquitetura de computadores, em especial, os conceitos relacionados ao processamento paralelo e à supercomputação.

# Objetivos

## ✓ Específicos:

- Identificar os princípios avançados da arquitetura e organização de computadores;
- Reconhecer a importância da evolução e tecnologias da arquitetura de computadores;
- Avaliar e comparar diferentes arquiteturas paralelas;
- Identificar paradigmas de processamento não convencionais.

# Metodologia

- ✓ Aulas expositivas.
- ✓ Resolução de problemas.
- ✓ Elaboração de trabalho extra-sala.
- ✓ Atividades em laboratório de Hardware.
- ✓ Realização de provas escritas.

# Avaliação

- ✓ Três unidades
  - Cada unidade terá uma avaliação
    - Possibilidade de trabalhos e seminários.
  - É permitido faltar apenas uma prova.
    - Justificar falta com atestado.
    - O assunto da prova de reposição será o assunto da unidade.
- ✓ Projeto

# Calendário 2014.1

14/04	Aula 1
16/04	Aula 2
23/04	Aula 3
05/05	Aula 4
07/05	Aula 5
12/05	Aula 6
14/05	Aula 7
19/05	Aula 8
21/05	Aula 9
26/05	Aula 10
<b>28/05</b>	<b>P1</b>

02/06	<b>Aula 12</b>
04/06	<b>Aula 13</b>
09/06	<b>Aula 14</b>
16/06	<b>Aula 15</b>
25/06	<b>Aula 16</b>
07/07	Aula 17
09/07	Aula 18
14/07	Aula 19
16/07	Aula 20
21/07	Aula 21
<b>23/07</b>	<b>P2</b>

28/07	Aula 22
30/07	Aula 23
04/08	Aula 24
06/08	Aula 25
11/08	Aula 26
13/08	Aula 27
<b>18/08</b>	<b>P3</b>
<b>20/08</b>	<b>Repos.</b>
<b>25/08</b>	<b>Final</b>

# Bibliografia

- ✓ **Patterson, D. A., Hennesy, J. L. Arquitetura de computadores: uma abordagem quantitativa. 4. Ed. Campus/Elsevier, 2008.**
- ✓ **Stallings, W. Arquitetura e organização de computadores. 5. Ed. Pearson, 2002.**
- ✓ **Tanenbaum, A. S. Organização estruturada de computadores. 5. Ed. Pearson/Prentice Hall, 2007.**
- ✓ **Navaux, P. O. A., De Rose, C. A. F. Arquiteturas paralelas. Bookman/UFRGS, 2008.**
- ✓ **Patterson, D. A., Hennesy, J. L. Organização e projeto de computadores. 4. Ed. Campus/Elsevier, 2005.**

# Projeto

- ✓ Desenvolver uma Arquitetura de Computador em SystemVerilog
  - Grupo de ...
  - Incremental
  - Apresentações parciais

# Projeto de Circuitos Digitais para Microprocessadores Usando SystemVerilog

# Introdução

- ✓ **Objetivo** → Entender os princípios básicos do projeto de circuitos digitais e como aplicá-los na construção de circuitos complexos de microprocessadores.
- ✓ Linguagens de Descrição de *Hardware* (HDL).
  - VHDL (Very high speed integrated circuit Hardware Description Language).
  - Verilog 1995, 2002.
  - **SystemVerilog**

# SystemVerilog X VHDL

```
✓ module up_counter ( output reg [7:0]
  out , input wire enable, clk, reset);
  always_ff @(posedge clk)
    if (reset) begin
      out <= 8'b0 ;
    end else if (enable) begin
      out ++;
    end
  end
endmodule
```

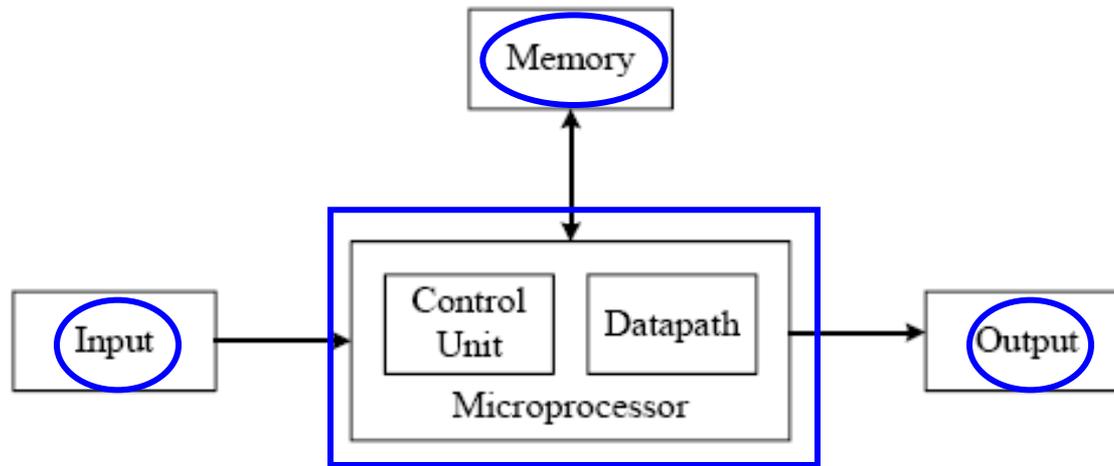
```
✓ library ieee;
  use ieee.std_logic_1164.all;
  use ieee.std_logic_unsigned.all;
  entity up_counter is
    port ( cout :out std_logic_vector (7 downto 0);
          enable :in std_logic;
          clk :in std_logic;
          reset :in std_logic );
  end entity;
  architecture rtl of up_counter is
    signal count :std_logic_vector (7 downto 0);
  begin
    process (clk, reset) begin
      if (reset = '1') then
        count <= (others=>'0');
      elsif (rising_edge(clk)) then
        if (enable = '1') then
          count <= count + 1;
        end if;
      end if;
    end process;
    cout <= count;
  end architecture;
```

# Microprocessadores

- ✓ Existem dois tipos de microprocessadores:
  - **Microprocessadores de propósito geral.**
    - São projetados para executarem diversas tarefas.
    - Utilizados nos computadores pessoais.
  - **Microprocessadores dedicados.**
    - São projetados para executarem apenas uma tarefa específica.
    - Utilizados em celulares, brinquedos eletrônicos, TVs, aparelho microondas, sistemas *anti-lock brakes* em carros entre outros.

# Microprocessadores

- ✓ Visão geral de um microprocessador - Modelo Von Neumann.



- ✓ Nosso interesse → Projeto dos circuitos digitais do microprocessador, memória e outros circuitos.

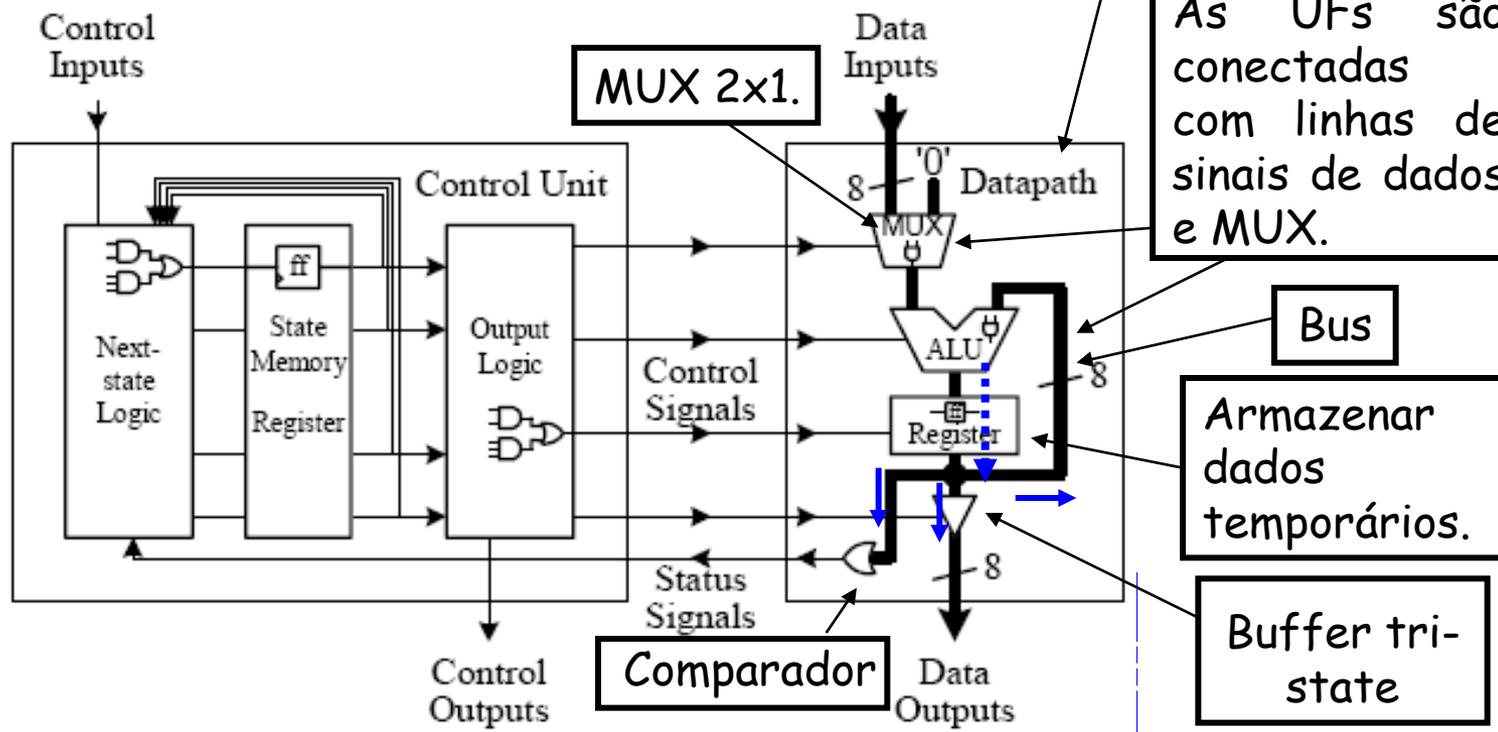
# Microprocessadores

## ✓ Circuito lógico do microprocessador.

- É dividido em duas partes:
  - Datapath;
  - Unidade de controle.

Responsável pela execução atual de todas as operações de dados. Ex: adição de dois números na ULA.

As UFs são conectadas com linhas de sinais de dados e MUX.



Bus

Armazenar dados temporários.

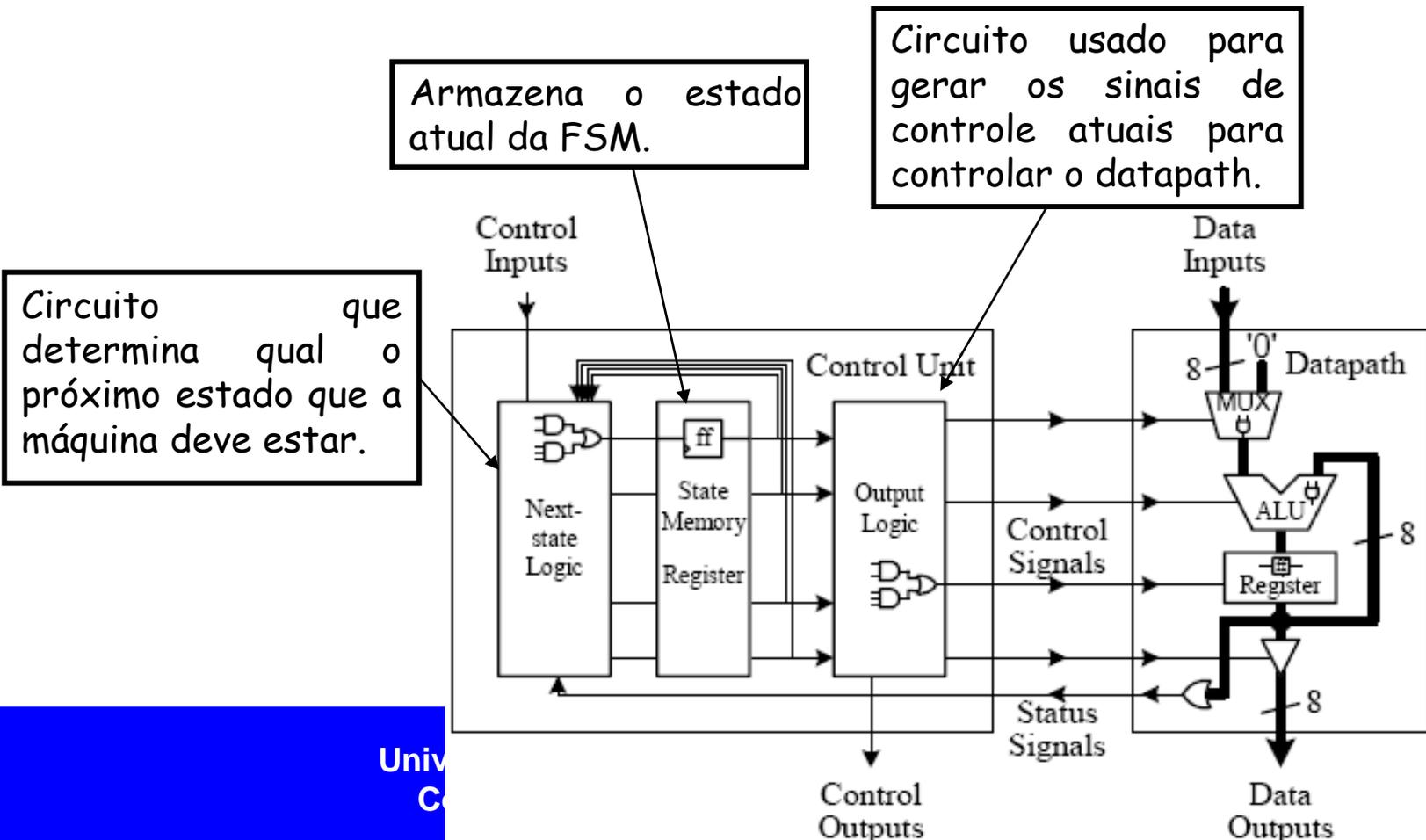
Buffer tri-state

# Microprocessadores

- ✓ **Circuito lógico do microprocessador.**
  - **Unidade de controle.**
    - É uma máquina de estados finito (FSM) que controla todas as operações do *datapath*.
      - **Máquina de estado finito** → Máquina que passa de um estado para outro de acordo com a ocorrência de eventos. O número de possíveis estados é finito.
    - A unidade de controle é formada por três partes:
      - Lógica do próximo estado;
      - Memória de estado;
      - Lógica de saída.

# Microprocessadores

- ✓ Circuito lógico do microprocessador.
  - Unidade de controle.



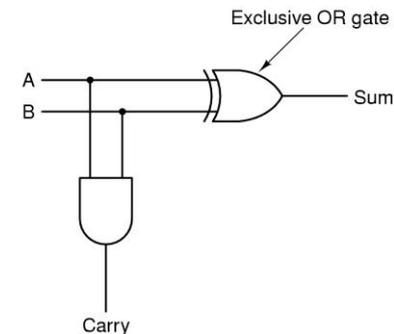
# Microprocessadores

## ✓ Circuito lógico do microprocessador.

- Os circuitos lógicos digitais do microprocessador são classificados como:
  - Circuitos combinacionais.
  - Circuitos seqüenciais.
- **Circuitos combinacionais** → Circuitos onde as suas saídas dependem apenas

- Ex: circuito somador.

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



# Microprocessadores

- ✓ **Circuito lógico do microprocessador.**
  - **Circuitos sequenciais** → Dependem não apenas das suas entradas atuais, mas também das suas entradas passadas.
    - Devem conter elementos de memória para lembrar das entradas passadas.
    - Ex.: partes do circuito de um controle de TV - Botão *up-channel*.

# Microprocessadores

- ✓ **Circuito lógico do microprocessador.**
  - Exemplos destes tipos de circuitos dentro do microprocessador.
    - **Combinacionais** → Circuitos da lógica do próximo estado e lógica de saída na unidade de controle, e ULA, MUX, buffers *tri-states* e comparadores no *datapath*.
    - **Sequenciais** → Registrador para a memória de estado na unidade de controle, e os registradores no *datapath*.

# Microprocessadores

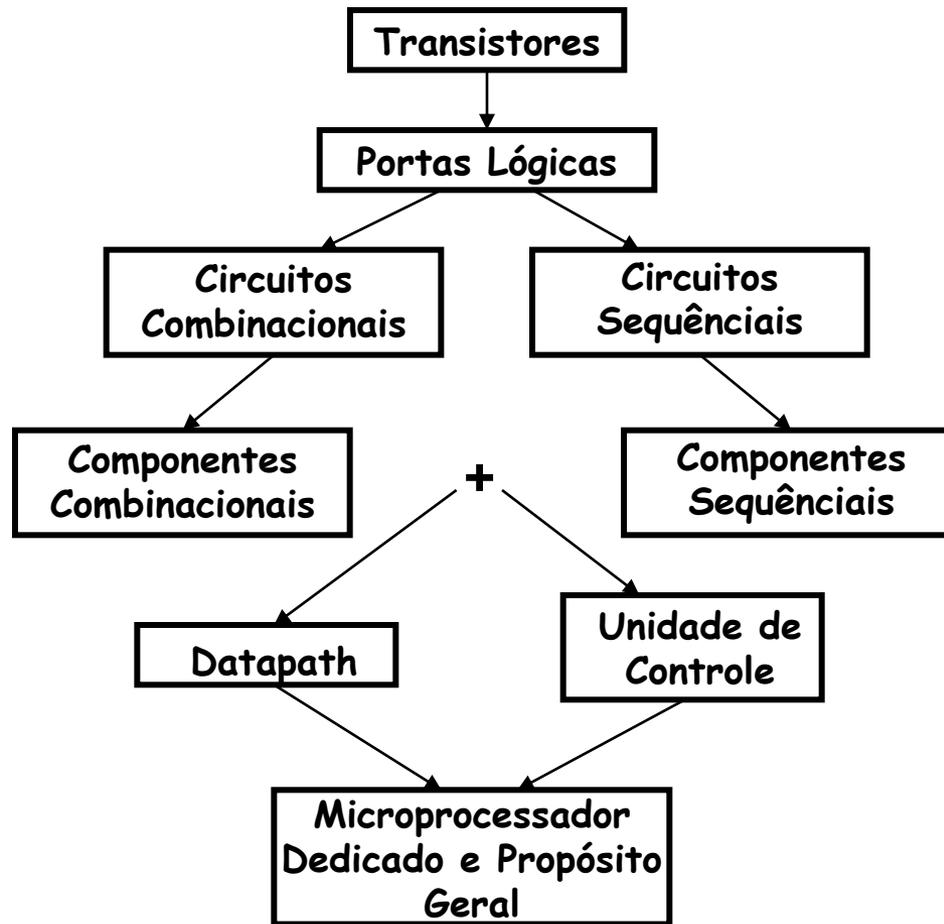
- ✓ **Circuito lógico do microprocessador.**
  - Independente da classificação do circuito, eles são constituídos pelas três portas lógicas básicas: **and**, **or** e **not**.
  - O PC mais poderoso que existe é feito a partir destas três portas.

# Microprocessadores

- ✓ **Circuito lógico do microprocessador.**
  - As portas básicas são construídas usando transistores.
  - Transistores são "chaves" binárias eletrônicas que podem ser ligadas ou desligadas. Os estados ligado e desligado de um transistor são usados para representar os dois valores binários: 0 e 1.

# Microprocessadores

- ✓ Ligação entre os componentes e partes que formam um microprocessador.



# Microprocessadores

- ✓ **Níveis de abstração do projeto.**
  - Circuitos digitais podem ser projetados em um dos vários níveis de abstração:
    - **Nível de transistor;**
    - **Nível de porta;**
    - **Nível registrador-transferência;**
    - **Nível comportamental.**

# Microprocessadores

## ✓ Níveis de abstração do projeto.

- **Nível de transistor** → É o nível mais baixo. Os transistores são conectados entre si para formar os circuitos.
- **Nível de porta** → Utilizamos as portas lógicas para construir os circuitos.
  - O circuito é especificado usando tabelas verdade ou uma equação booleana.
  - Componentes combinacionais e sequenciais poderão ser desenvolvidos para construir circuitos maiores.

# Microprocessadores

- ✓ Níveis de abstração do projeto.
  - **Nível registrador-transferência** → Utilizamos os componentes combinacionais e seqüenciais para construir o *datapath* e a unidade de controle do microprocessador.
    - Neste nível, o foco está em como os dados são transferidos entre os vários registradores e as unidades funcionais.

# Microprocessadores

- ✓ Níveis de abstração do projeto.
  - **Nível comportamental** → Nível mais alto.
    - Os circuitos são construídos por meio da descrição do seu comportamento ou operação usando uma **Linguagem de Descrição de Hardware (HDL)**.
    - É semelhante a escrever um programa de computador usando uma linguagem de programação.

# Quartus II e Modelsim

- ✓ Quartus II 13.1 Web Edition.
  - <http://www.altera.com/products/software/sfw-index.jsp>
  - É necessário ter um cadastro no *site* Altera para baixar o Quartus II.