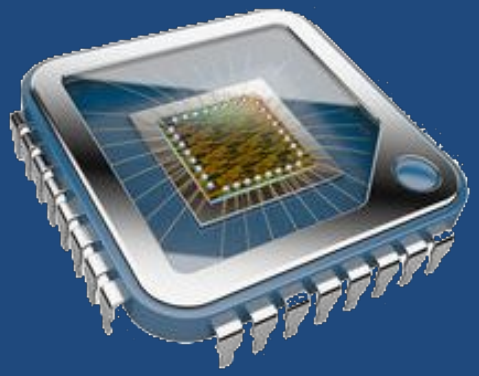


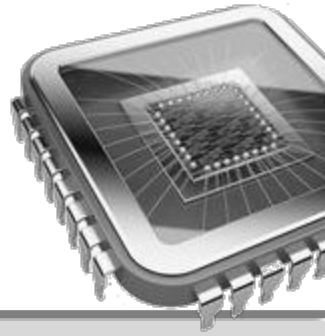
ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES
AULA 21

Prof. Max Santana Rolemberg Farias
max.santana@univasf.edu.br
Colegiado de Engenharia de Computação

GESE
Grupo de Engenharia de Sistemas Embarcados

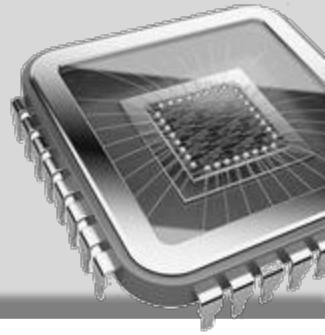
CEComp
Engenharia da Computação

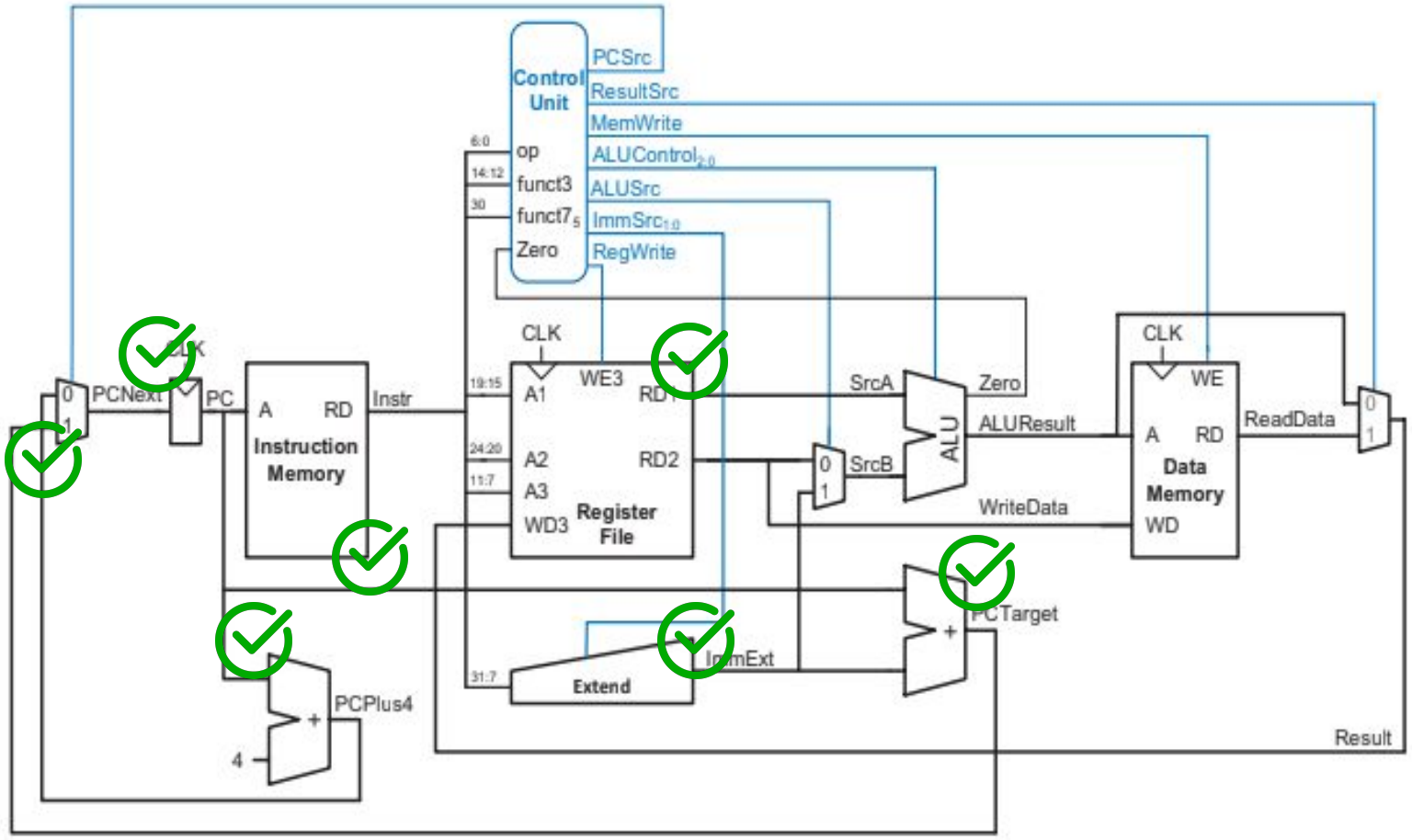
UNIVASF
UNIVERSIDADE FEDERAL DO VALE DO SÃO FRANCISCO



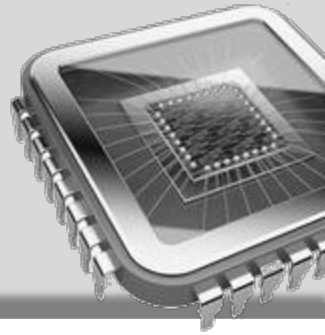
PROCESSADOR DIDÁTICO
myRISCVv1

PROCESSADOR myRISCVv1
DATAPATH



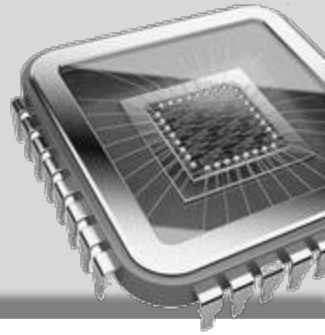


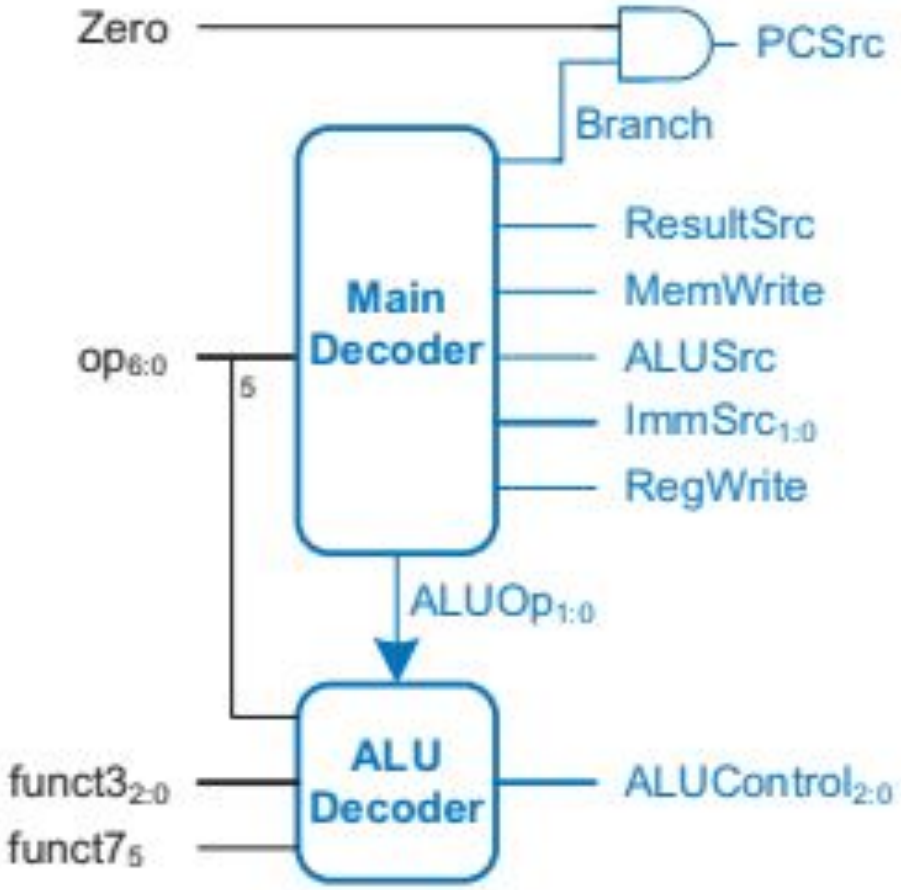
PROCESSADOR myRISCVv1 CONTROLLER



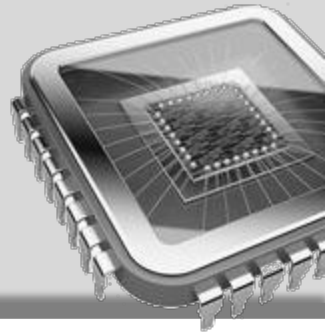
A unidade de controle do myRISCVv1 é capaz de receber o opcode de uma instrução e gerar sinais de controles para os componentes do processador.

PROCESSADOR myRISCVv1
CONTROLLER



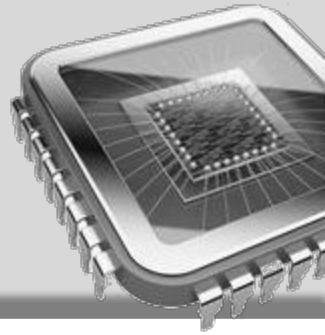


PROCESSADOR myRISCVv1
CONTROLLER (MAIN DECODER)



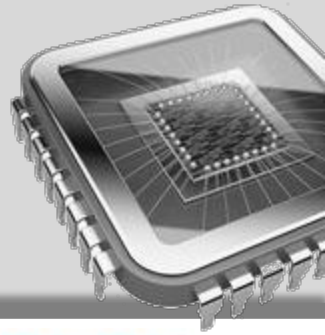
Instruction	Op	RegWrite	ImmSrc	ALUSrc	MemWrite	ResultSrc	Branch	ALUOp
lw	0000011	1	00	1	0	1	0	00
sw	0100011	0	01	1	1	x	0	00
R-type	0110011	1	xx	0	0	0	0	10
beq	1100011	0	10	0	0	x	1	01

PROCESSADOR myRISCVv1
CONTROLLER (ALU DECODER)



ALUOp	funct3	{op_5, $funct7_5$}	ALUControl	Instruction
00	x	x	000 (add)	`lw, sw`
01	x	x	001 (subtract)	`beq`
10	000	00, 01, 10	000 (add)	`add`
	000	11	001 (subtract)	`sub`
	010	x	101 (set less than)	`slt`
	110	x	011 (or)	`or`
	111	x	010 (and)	`and`

PROCESSADOR myRISCVv1
CONTROLLER



```
----------------------------------------------------------------------------------
-- Controller unit                                                              --
-- myRISCVv1                                                                    --
--                                                                              --
-- Prof. Max Santana  (2025)                                                    --
-- CEComp/Univasf                                                               --
----------------------------------------------------------------------------------

library ieee;
use ieee.std_logic_1164.all;

entity controller is
  port(
    op        : in std_logic_vector(6 downto 0);
    funct3    : in std_logic_vector(2 downto 0);
    funct7    : in std_logic_vector(6 downto 0);
    resultSrc : out std_logic;
    memWrite  : out std_logic;
    PCSrc     : out std_logic;
    aluSrc    : out std_logic;
    regWrite  : out std_logic;
    immSrc    : out std_logic_vector(1 downto 0);
    aluControl: out std_logic_vector(2 downto 0)
  );
end;
```