



INSALUBYTE

Processador 16 bits, RISC, MIPS

Alunos: Victor Hugo, Giovana Oliveira e Ryan Kayky
Disciplina: Arquitetura e Organização de Computadores
Professor: Herbert Oliveira Rocha
Semestre: ERE 2023.2
30 de Novembro de 2023
Boa Vista - RR

CARACTERÍSTICAS DO PROCESSADOR

- 1 Processador 16 bit, 65 536 linhas de código em um programa
- 2 16 bytes de espaço na memória RAM;
- 3 16 registradores disponíveis
- 4 O J realiza um salto de 4095 linhas

COMPONENTES

- 1 ULA or ALU e Branch Helper
- 2 Somador, PC, Divisor de instruções, Extensor de sinal 4x16 bits
- 3 Unidade de controle UC
- 4 Memória RAM
- 5 Banco de registradores
- 6 Multiplexador

TIPO DE INSTRUÇÕES

Tipo R

| | | | |
|--------|--------|--------|--------|
| 4 bits | 4 bits | 4 bits | 4 bits |
| 15-12 | 11-8 | 7-4 | 3-0 |
| Opcode | Reg1 | Reg2 | Reg3 |

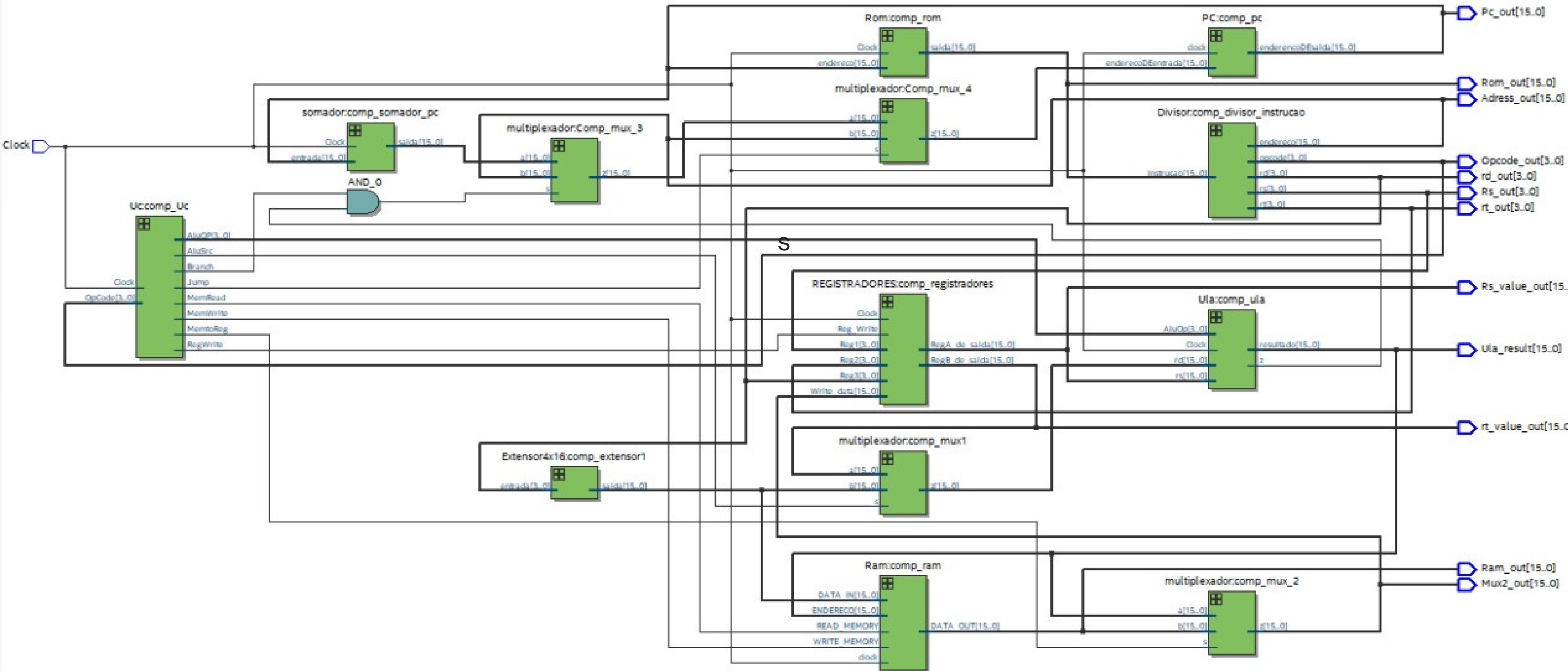
- Este formato aborda instruções baseadas em operações aritméticas, como add, sub e mult.

TIPO DE INSTRUÇÕES

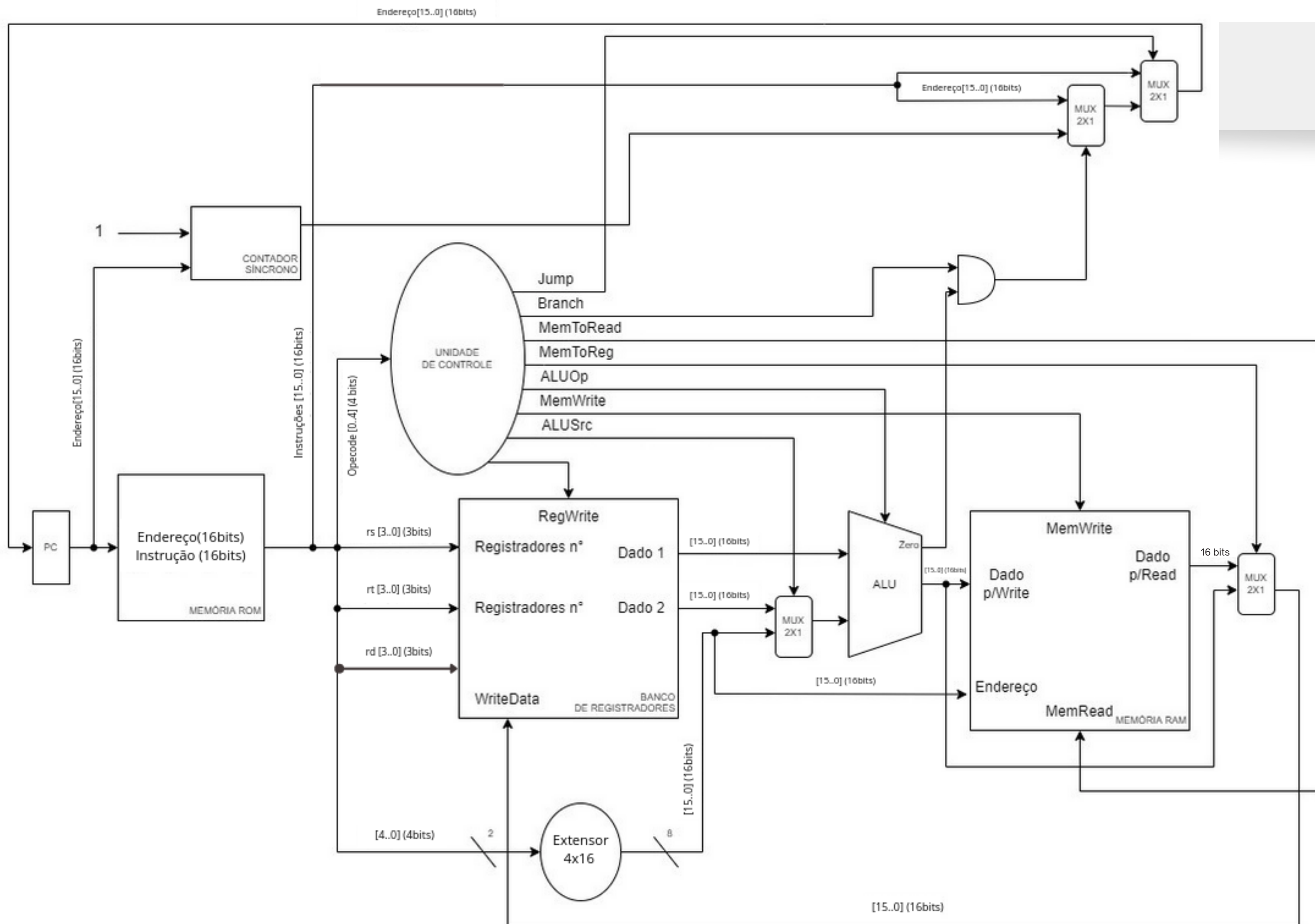
Tabela geral de instruções

| Opcode | Nome | Formato | Breve Descrição | Exemplo |
|--------|-------|---------|------------------------|---|
| 0000 | ADD | R | Soma | add \$S0, \$S1, \$S2, ou seja, \$S0 := \$S1+\$S2 |
| 0001 | ADDi | I | Soma Imediata | addi \$S0, \$S1, X, ou seja, \$S0 := \$S1+X |
| 0010 | SUB | R | Subtração | sub \$S0, \$S1, \$S2, ou seja, \$S0 := \$S1 - \$S2 |
| 0011 | SUBi | I | Subtração Imediata | subi \$S0, \$S1, X, ou seja, \$S0 := \$S1-X |
| 0100 | LW | I | Load Word | lw \$S0, 0(\$0) |
| 0101 | SW | I | Store Word | sw \$S0, 0(\$0) |
| 0110 | LI | I | Load Imediato | li \$S0, 31 |
| 0111 | BEQ | R | Salto Condicional | beq \$S0, \$S1, L1 |
| 1000 | IF | R | Condição | if \$S0, \$S1 |
| 1001 | J | J | Salto | J L1 |
| 1010 | MULT | R | Multiplicação | mult \$S0, \$S1, \$S2, ou seja, \$S0 := \$S1*\$S2 |
| 1011 | MULTi | I | Multiplicação imediata | multi \$S0, \$S1, X, ou seja, \$S0 := \$S1*X |

MAPA GERAL DO INSALUBYTE



DATAPATH



TESTES

EXEMPLO DO CÓDIGO FIBONACCI

| Endereço | Linguagem de Alto Nível | Binário | | | |
|----------|-------------------------|------------------|----------|------|------|
| | | Opcode | Reg1 | Reg2 | Reg3 |
| | | | Endereço | | |
| | | Dado | | | |
| 0 | <u>Addi</u> \$S0 0 | 0001 | 0000 | 0000 | 0000 |
| | | 0001000000000000 | | | |
| 1 | <u>Sw</u> \$S0 | 0101 | 0000 | 0000 | 0000 |
| | | 0101000000000000 | | | |
| 2 | <u>Addi</u> \$S0, 1 | 0001 | 0000 | 0000 | 0001 |
| | | 0001000000000001 | | | |
| 3 | <u>Addi</u> \$S1, 1 | 0001 | 0010 | 0001 | 0001 |
| | | 0001000100010001 | | | |
| 4 | <u>Lw</u> \$S2, 0 | 0100 | 0010 | 0010 | 0000 |
| | | 0100001000100000 | | | |
| 5 | Add \$S2, \$S1 | 0000 | 0010 | 0010 | 0001 |
| | | 0000001000100001 | | | |
| 6 | Add \$S1, \$S0 | 0000 | 0001 | 0001 | 0000 |
| | | 0000000100010000 | | | |
| 7 | <u>Lw</u> \$S0, 0 | 0100 | 0000 | 0000 | 0000 |
| | | 0100000000000000 | | | |
| 8 | Add \$S0, \$S2 | 0000 | 0000 | 0000 | 0010 |
| | | 0000000000000010 | | | |
| 9 | J 0100 | 1001 | 0000 | 0000 | 0100 |
| | | 1001000000000100 | | | |

REFERÊNCIAS

- 1 <https://www.fpga4student.com/2017/09/vhdl-code-for-mips-processor.html>
- 2 <https://allaboutfpga.com/vhdl-code-flipflop-d-t-jk-sr/>
- 3 https://github.com/ed-henrique/8-bit-CPU/blob/main/CPU_EK/SOMADOR_8BITS.vhd
- 4 https://github.com/nataliaalmada/AOC_2GabrielENatalia_UFRR_2022/blob/main/Componentes/MemoriaRAM.vhd