

- # Projeto MIPS Multiciclo

Para o projeto da disciplina devem ser acrescentadas as seguintes instruções:

- **ORi:** `ori rs, rt, imediato breg[rt] = breg[rs] | 0x0000iiii;`

| | | | |
|-----|----|----|----------|
| 0xd | rs | rt | imediato |
|-----|----|----|----------|

- **ANDi:** `andi rs, rt, imediato breg[rt] = breg[rs] & 0x0000iiii;`

| | | | |
|-----|----|----|----------|
| 0xc | rs | rt | imediato |
|-----|----|----|----------|

- **SLL:** deslocamento lógico à esquerda

`sll rd, rt, shamt breg[rd] = breg[rt] << shamt;`

| | | | | | |
|-----|----|----|----|-------|---|
| 0x0 | rs | rt | rd | shamt | 0 |
|-----|----|----|----|-------|---|

- **SRL:** deslocamento lógico à direita

`srl rd, rt, shamt breg[rd] = breg[rt] >> shamt;`

| | | | | | |
|-----|----|----|----|-------|---|
| 0x0 | rs | rt | rd | shamt | 2 |
|-----|----|----|----|-------|---|

- **LB/LBU/LH/LHU:** ler int8_t/uint8_t/int16_t/uint16_t da memória

`lx rt, kte16(rs) breg[rt] = mem[breg[rs]+kte16];`

| | | | | |
|-----|------|----|----|-------|
| LB | 0x20 | rs | rt | kte16 |
| LBU | 0x24 | rs | rt | kte16 |
| LH | 0x21 | rs | rt | kte16 |
| LHU | 0x25 | rs | rt | kte16 |

- **SB/SH:** armazena int8_t/int16_t

`sx rs, kte16(rs) mem[breg[rt]+kte16] = breg[rs];`

| | | | | |
|----|------|----|----|-------|
| SB | 0x28 | rs | rt | kte16 |
| SH | 0x29 | rs | rt | kte16 |

O processador deve ser simulado no ModelSim e implementado em FPGA. Neste caso, para verificação de seu funcionamento, deve-se observar o PC, o RI, o RDM e a saída da ULA através dos mostradores de 7 segmentos.

Para entrada e saída de dados deve-se utilizar:

- botão para acionar o relógio
- mostrador de 7 segmentos para exibir conteúdo de RI e SaidaALU

O PC deve ter 32 bits. A memória tem apenas 256 palavras de 32 bits, de forma que apenas 8 bits do PC devem ser utilizados no seu endereçamento. Para executar um programa gerado pelo MARS, devem ser carregados o código e os dados do programa. Os endereços da área de dados devem ser mapeados para a região de memória que começa no endereço 128 no FPGA. Assim, o endereço de dado deve ser gerado concatenando os bits [8 downto 2] do registrador de saída da ULA com o bit '1' na posição mais significativa: '1' & alu_out(8 downto 2).

Um arquivo comprimido com todos os módulos VHDL do MIPS multiciclo é disponibilizado no Moodle. O código MIPS a ser carregado na memória está contido no arquivo **mem.mif**.

Para exibição dos dados nos mostradores, utilizar os acionadores de display de 7 segmentos feito na primeira aula de laboratório.

A verificação do processador consistirá na execução de programas gerados a partir do MARS.

Entrega: até 11 de dezembro. Apresentar um relatório sucinto de implementação, o código VHDL simulando no ModelSim e executando em FPGA.

Obs: lógica para leitura e escrita de *bytes* e *half-words* na memória:

