

Instruções:

1. Esta atividade pode ser realizada em **até TRÊS** alunos (no máximo) para implementações usando HDL ou individualmente para implementações usando esquemático.
2. A defesa da implementação será realizada no dia especificado no sistema, com horário final da postagem até o início da aula. Defesa e/ou entrega em atraso **não serão aceitas**.
3. Os arquivos do modelo desenvolvido devem ser compactados em um único arquivo pela ferramenta do Quartus (**QAR**) e postados no ambiente Material Didático.
4. Se forem identificados trabalhos com grau de similaridade que caracterize cópia ou adaptação (autorizadas ou não pelos seus autores originais), a nota dos grupos será a nota de um trabalho dividida pelo número de grupos que entregou esses trabalhos similares.

PROJETO

HDL:

Implementar o processador MIPS Monociclo com Jump.

Esquemático:

Considerando o modelo do MIPS monociclo para o Quartus II sem Jump, acrescente suporte em hardware para a execução da instrução j, relativa ao desvio incondicional.

VALIDAÇÃO

A validação deve ser feita por simulação. Para a validação será necessário gerar o arquivo rom.vhd ou rom.mif manualmente a partir do código hexadecimal dos programas abaixo, o qual pode ser obtido usando o MARS (*dump file*).

Programa 1:

```
.data
Array_A: .word 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150

.text
main:
    addi $s2, $zero, 1    # Inicializa $s2 em 1
    la   $s3, Array_A     # como o exercício assume que o endereço-base de A[]
                          # está em $s3, foi incluída esta instrução

    lw   $t0, 32($s3)     # $t0 = A[8]
    add  $t0, $s2, $t0     # $t0 = $t0 + h
    sw   $t0, 48($s3)     # A[12] = $t0
```

Programa 2:

```
.text
main:
    addi $s0, $zero, 0
    addi $s1, $zero, 1
    addi $s2, $zero, 2
    addi $s3, $zero, 3
    addi $s4, $zero, 4

    beq $s3, $s4, L1      # if i==j goto L1
    add $s0, $s1, $s2      # f = g + h = 1 + 2 = 3
L1:   sub $s0, $s0, $s3    # f = f - i = 3 - 3 = 0
```

Programa 3:

```
.text
main:
    addi $s0, $zero, 1
loop: addi $s0, $s0, 2
      j    loop
```