

Instruções:

1. Esta atividade pode ser realizada individualmente ou em **até TRÊS** alunos (no máximo).
2. O relatório deverá ser entregue impresso em papel, com uma capa identificando a Instituição, o curso, a disciplina, o professor, o nome da atividade, os autores do trabalho e data em que o mesmo for entregue. O corpo do relatório deverá conter as imagens que representam as implementações realizadas (circuitos) e as simulações que provam a validação dessas implementações.
3. A defesa da implementação será realizada dia **07/06/2016**, e a data final da entrega do relatório impresso até o início da aula do dia **14/06/2016**. Defesa e/ou entrega em atraso não serão aceitas.
4. Os arquivos do modelo desenvolvido devem ser compactados em um único arquivo e postados no ambiente Material Didático. Antes de compactar, organize o seu modelo em uma pasta denominada: **MIPS_Nome1_Nome2_Nome3**, onde Nome(1 a 3) refere-se aos nomes dos integrantes do grupo.
5. 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

Considerando o modelo do MIPS monociclo para o Quartus II sem Jump, acrescente suporte em hardware para a execução das instruções j, jal e jr, relativas ao desvio incondicional, à chamada de procedimentos e ao retorno de procedimentos.

Será necessário modificar o caminho de dados e o controle, inserido novas instâncias de alguns componentes já existentes (ex. mux) ou criando novos componentes.

No relatório, devem ser apresentadas imagens que ilustrem as modificações realizadas e os diagramas de formas de onda da validação por simulação (ver abaixo).

IMPORTANTE

A versão original do modelo MIPSmono do assume que o endereço inicial do programa é 0x0040.0020 (ou seja, 4194336), mas em alguns projetos o MARS está posicionando o programa para o endereço 0x0040.0000 (ou seja, 4194304). Nesse caso, é preciso editar parâmetros que definem o endereço inicial do programa nos módulos abaixo:

1. Módulo program_counter: No bloco LPM_DFF, editar o parâmetro LPM_AVALUE, atribuindo-lhe o valor 4194304.
2. Módulo instruction_memory: No bloco LPM_CONSTANT (parte inferior da folha, à esquerda), editar o parâmetro LPM_CVALUE, atribuindo-lhe o valor 4194304.

Para editar esses parâmetros, é preciso abrir o módulo e clicar sobre a área de parâmetros do bloco indicado para que seja apresentada a janela de parâmetros. Nessa janela, deve-se selecionar o parâmetro desejado e editar o seu valor.

VALIDAÇÃO

A validação deve ser feita pela simulação. Para a validação do jump, já existe um programa pronto (rom04). Para a validação do jal e do jr, será necessário gerar o arquivo rom.mif manualmente a partir do código hexadecimal do programa abaixo, o qual pode ser obtido usando o MARS (*dump file*).

```
# Trecho em C para validação do jr e do jal:
# int leaf_example (int g, int h, int i, int j) {
#     int f;
#     f = (g + h) - (i + j);
#     return f;

        .text    # segmento de código (programa)
        j      main
leaf_example:
    add $t0, $a0, $a1    # $t0 = g + h
    add $t1, $a2, $a3    # $t1 = i + j
    sub $v0, $t0, $t1    # f = $t0 - $t1
    jr  $ra              # retorna do procedimento

main:
    addi $a0, $zero, 4   # inicializa 1º parâmetro (g)
    addi $a1, $zero, 3   # inicializa 2º parâmetro (h)
    addi $a2, $zero, 2   # inicializa 3º parâmetro (i)
    addi $a3, $zero, 1   # inicializa 4º parâmetro (j)

    jal leaf_example     # chama o procedimento

    nop                  # não faz nada. $v0 tem o resultado do procedimento
```

IMPORTANTE: Para simplificar a solução do problema, não se deve considerar o uso da pilha. Por isso, não são utilizados registradores \$s no programa de validação.