1. Imagine o efeito causado por manter em 0 todos os sinais de controle no caminho de dados multiciclo apresentado nas aulas teóricas (tenha como base a Figura 5.33 de Patterson; Hennessy (1998)). Considerando cada sinal separadamente, determine quais instruções não continuariam a funcionar.

2. Faça nesta o mesmo solicitado na questão 3, porém, considerando que todos os sinais de controle permanecem em 1.

3. Considere a seguinte idéia: modificar a arquitetura do conjunto de instruções e retirar a capacidade de especificar um deslocamento para instruções de acesso à memória. Especificamente, todas as instruções *load-store* com deslocamento diferente de zero devem tornar-se pseudo-instruções e devem se implementadas por meio de duas instruções. Por exemplo:

   addi ***$at, $t1, 104*** ***# some o deslocamento a um registrador temporário***
   lw ***$t0, $at*** ***# nova forma de fazer lw $t0, 104($t1)***

   Que mudanças devem ser feitas no caminho de dados multiciclo e no controle para que essa arquitetura simplificada possa funcionar?

4. Adicione a instrução *addi* (soma imediata) ao caminho de dados multiciclo descrito nas aulas teóricas. Acrescente o que for necessário, em termos de elementos e de sinais de controle, ao caminho de dados da Figura 5.33. Mostre as modificações na MEF, em função desta nova instrução. Tente encontrar uma solução que minimize o número de ciclos gastos para executar a nova instrução no nosso caminho de dados modificado. Mostre a nova MEF decorrente da inclusão desta nova instrução.

5. Escreva a sequência de microcódigo para a instrução *addi*. Se você precisar fazer alguma mudança no formato da microinstrução, ou no conteúdo de algum campo, indique claramente o novo formato e os campos que vão colocar valores na saída de controle.

6. Faça nesta questão o mesmo solicitado nas questões 4 e 5, porém, agora para a instrução ***jal (jump and link)***.

7. Faça nesta questão o mesmo solicitado nas questões 4 e 5, porém, agora para a seguinte instrução aritmética:

   add3 $t5, $t6, $t7, $t8 # $t5 = $t6 + $t7 + $t8

   Suponha que a ISA tenha sido modificada com a inclusão de um novo formato de instrução, semelhante às instruções do Formato R, com a diferença de terem os bits[0-4] usados para especificar o registrador adicional (os campos rs, rt e rd ainda são usados da maneira original). A instrução terá um novo *opcode*. Sua solução não deve basear-se em acrescentar novas portas de leitura ao banco de registradores e nem deve propor o uso de uma nova ULA.

8. Implemente as alterações necessárias para que a CPU MIPS Multiciclo vista nas aulas execute também a instrução ***jalr*** (***Jump and Link Register***). Diferente do especificado, considere que o registrador 31 não mais é o default. Para a sua implementação altere o que for necessário diretamente no caminho de dados multiciclo e na MEF, ambos disponibilizados na prova. Caso você conclua que o caminho de dados e/ou a MEF não precisam ser alterados, afirme isto na sua resposta.

9. Implemente as alterações necessárias para que a CPU MIPS Multiciclo vista nas aulas execute também a instrução ***bltzal*** (***Branch if less then and link***). Para a implementação altere o que for necessário no caminho de dados multiciclo e na MEF, ambos disponibilizados na prova. Caso você conclua que o caminho de dados e/ou a MEF não precisam ser alterados, afirme isto na resposta.

10. Implemente as alterações necessárias para que a CPU Multiciclo execute a instrução ***bne***. Para a implementação altere o que for necessário no caminho de dados multiciclo e na MEF. Caso você conclua que o caminho de dados não será alterado, afirme isto e indique as alterações necessárias à MEF.