

Grupo	Simulação com formas de onda comentadas (2,5 pontos)	Novo VHDL (3,5 pontos)	Programa(s) de teste: .asm e .txt (2,5 pontos)	Documentação e Comentários (1,5 pontos)	Nota
Alan Oliveira e William Rodrigues	A forma de onda para BLTZAL está incompleta, não se visualiza a troca de valores do PC, nem a instrução que está sendo executada, ou ainda o estado da máquina de controle em que as coisas acontecem. Embora a forma de onda cite os sinais i, ck, rst, \$t0 e \$ra nominalmente, os sinais mostrados são rst, ck, \$t0, \$ra e mais 4 sinais não identificados, ruim. A forma de onda da imagem 2 do relatório mostra de forma clara que implementaram incorretamente a instrução LHU, pois ela escreve dados na memória, ao invés de ler dados desta: vejam a ativação dos sinais de controle ce=1, rw=0 (escrita na memória), bem como o sinal de ps=ss indicando o estado de escrita na memória. Como o registrador destino da LHU não está na forma de onda, não é possível ver se ele é escrito também ou não. Outra péssima escolha é representar valores em decimal e não em hexadecimal ou binário no teste da LHU, dificultando muito verificar se extensão de sinal ou de zero ocorre e como. A forma de onda que ilustra a execução da MTHI está incompleta e não parece corresponder ao programa de teste, pois no programa existe apenas 1 instrução MTHI executada, mas na forma de onda o i=MTHI por 8 ciclos de clock. Infelizmente, o valor do sinal ps não está visível, devido ao nível de zoom incorreto usado na imagem. Assim, não há como saber se trata-se de erro ou incoerência de onda/programa de teste. Mas certamente a instrução não executa corretamente, pois HI é escrito em 400ns com o valor 100, durante a instrução addiu! E não durante a instrução MTHI. Pode até ser que MTHI escreva o mesmo valor em HI, mas a imagem mostrada da simulação não permite saber. Outro erro claro é que a instrução ADDIU agora está funcionando de forma incorreta, escrevendo em 2 registradores (um do banco de registradores e no HI, ver o tempo 400ns) e ainda faz acesso à memória (não se sabe se para leitura ou escrita, ver de novo o instante 400ns). (1,0)	Mesmo quando BLTZAL não salta, escreve no registrador \$RA, o que é um erro. A instrução LHU está implementada de maneira incorreta: o 'U' de LHU significa "Unsigned". Logo a extensão deve ser sempre de 16 0s e não extensão de sinal. Esta última ação caracteriza a instrução LH e não a LHU. A decodificação da instrução MTHI está incorreta, assumindo que o registrador fonte é especificado nos bits 15-11 da instrução, mas o correto é que quem especifica este são os bits 25-21. Corrigindo este problema, nota-se que a instrução não executa corretamente, pois não escreve em HI (não habilita o sinal de controle de escrita neste registrador) e ainda escreve no banco de registradores (habilita o sinal wreg). (2,2)	Teste do LHU usou um programa que não executa no MARS, por empregar endereços que o MARS não usa, porquê? Assim, não puderam verificar se o hardware opera como o software simulado. Há um erro mais grave ainda no programa de teste da LHU: 1 e 0x11001 geram valores que possuem o bit 15 em 0 nas suas representações binárias. Assim a extensão de sinal (incorretamente implementada na organização do trabalho) nunca será testada. (1,8).	Mencionam instruções BLZAL e LHI, que não existem. BLTZAL não realiza "desvios condicionais", mas "desvios condicionais PARA SUBROTINA", é importante salientar. LHU escreve 4 bytes em um registrador, não 2. SALTA é um mau nome para uma subrotina. Incorretamente, dizem que LHU estende o sinal dos bytes lidos. O que significa a palavra "butes"? Ela aparece 4 vezes na documentação. É erro ao escrever byte, claro, mas espera-se que pelo menos uma revisão do texto seja feita antes de entregar. (0,6)	5,6
Arthur da Silva, Drielle Teixeira (Cancelou) e Tomás Caldas	Muito difícil de visualizar a forma de onda para BLTZAL, uma vez que usam código binário ao invés de hexadecimal para mostrar o conteúdo de registradores, por quê? Isto se repeta para todas as instruções. É quase impossível identificar os endereços manipulados. Pior, não mostram conteúdo de registradores e o código não parece ser o mesmo do programa de teste, pois mostram uma forma de onda que executa 3 vezes BLTZAL e o código do teste só tem duas execuções da instrução. Desconfio que se trata do programa de teste que fica em loop eterno devido a erros. Nas formas de onda para MTHI e LHU a situação está pior ainda, pois múltiplos sinais começam a apresentar valores uninitialized (U) e conflitos (X) no meio da operação, indicando erros graves no VHDL. Muito problemática esta implementação. (0,3)	A instrução BLTZAL está com uma falha grave, ao saltar não ativa a escrita em \$RA. Ao todo são 3 erros: (1) não ativar wreg para escrever no banco; (2) não escolher \$RA (31) como endereço do registrador a escrever no mux M4; (3) não acertar a saída do mux MZ para passar o valor de NPC_IN para escrita no banco. A instrução MTHI não executa corretamente, pois escreve em HI um valor de 32 bits em 'U' e ainda escreve no banco de registradores (habilita o sinal wreg). A instrução LHU não executa como uma instrução de leitura como deveria, pois nela a máquina de estados passa de ps=Salu para ps=Swbk e não para PS=Slid, como deveria. Ou seja, nada é lido da memória e não se sabe o que se escreve, nem onde se escreve no banco de registradores. (1,7)	O programa de teste da BLTZAL ok, mas apenas testa valores menores e maiores que 0, deveriam ter testado exatamente com 0. O programa de teste da LHU tem falhas mais graves, pois faz quatro testes, e sempre lê valores de uma memória de dados que está totalmente zerada. Bastava gerar uma área de dados qualquer, para garantir que LHU sempre opera com extensão de 0. (2,1)	Documentação sucinta, com algumas informações úteis, mas que não se refletem no código final e não conseguiram capturar as implicações de cada instrução na organização de forma completa. (0,9).	5
Artur Rodrigues e Douglas Carvalho	Listaram 14 alterações realizadas na organização original, de forma sucinta, mas clara. Bons desenhos informam os fluxos de dados e de controle usados em cada uma das três novas instruções. Apenas seria bom ter mostrado na figura do BLTZAL as duas possibilidades de atualização do PC, mostraram apenas quando o salto é executado. (2,4)	Implementação funciona razoavelmente bem. Uma falha ocorre na instrução BLTZAL, que independente de saltar ou não sempre escreve algo no registrador \$RA (só deveria escrever quando saltasse). Outra pequena falha é na decodificação da instrução MTHI, que possui 27 bits com valores constantes e não apenas os 12 testados aqui. (,)	O programa de teste (único para as três instruções) é um tanto estranho. Usa a pilha para fazer algumas manipulações. A parte de teste da BLTZAL está ok, mas apenas testa valores menores e maiores que 0, deveriam ter testado exatamente com 0. LHU não testou com valores que tem o bit 15 em 0 (para confirmar que a extensão de 0 sempre é usada), apenas valores com o bit 15 em 1. (2,3).	Documentação um tanto sucinta, mas razoável. (1,3)	9
Bruno zaffari e Matheus Schneider	Não entregaram formas de onda. (0,0)	Implementação inicialmente sequer compilava, corrigi um erro de sintaxe na linha 551 em um comando de atribuição com um ELSE de ELSE, o que não é válido em VHDL. Depois, corrigi CINCO (5) erros em que o identificador "uins" (abreviatura de microinstrução) havia sido escrito como "unis", por quê? Depois disto, o código compilou, mas ao executar nota-se que o código objeto da instrução BLTZAL é decodificado como sendo uma instrução LHU, e os códigos objeto da LHU e da MTHI não são decodificados, gerando a infomação INVALID_INSTRUCTION. Parei por aí, não há como avançar na correção. (0,5)	Não entregaram nenhum código de programa de teste, seja fonte ou objeto. (0,0)	Não entregaram documentação. (0,0)	0,5