

Questão 1												Questão 2													
pc	inst	1	2	3	4	5	6	7	8	9	0	pc	inst	0	1	2	3	4	5	6	7	8	9	0	
0	Add r1,r2,r2											0	Add r1,r2,r2												
4	Lw r1,0(r1)											4	Lw r1,0(r1)												
8	Beq r1,r2,4											8	Beq r1,r2,4												
12	Add r1,r1,r1											12	Add r1,r1,r1												
16	Sw r1,0(r4)											16	Sw r1,0(r4)												
20	Beq r2,r2,-5											20	Beq r2,r2,-5												
24	Sub r6,r4,r2											24	Sub r6,r4,r2												
28	Or r7,r5,r1											28	Or r7,r5,r1												
32	And r9,r7,r3											32	And r9,r7,r3												

- 1) Suponha um processador com BEQ no quarto estágio. Suponha que o registro Ri tem o valor i, a memória(i) tem o valor i (considerando que está organizada em palavras de 32 bits, como foi considerado em aula, 4 em 4 bytes), Coloque os valores ( caso a linha seja irrelevante não coloque nada) nas linhas de controle e de dados no ciclo 6 na figura do DATAPATH anexo para o código do verso da folha na tabela. Suponha ciclo 1, o ADD está no estágio busca (fetch).
- 2) Refaça o exercício no pipeline simples com inserção de NOPs para o código funcionar corretamente. Refaça considerando apenas unidade de Forwarding.
- 3) Suponha o processador com BEQ no 2 estágio e unidades de forwarding e hazard para o BEQ. Execute novamente o código. Identifique a primeira situação de Forwarding de dados e mostre em qual ciclo ocorrerá. Qual o próximo ciclo que ocorrerá outra situação de forwarding envolvendo outra instrução, identifique também a situação. Identifique agora a primeira situação de hazard de LOAD e em qual ciclo ocorrerá. Quais instruções estarão nos estágios de Fetch, Decode, Exec, Mem e WriteBack ?
- 4) Refaça com BEQ no 2 Estágio porém sem unidades de Forward e Hazard.

4) Considere o código de encontrar o menor elemento de um vetor. Vetor na posição 200 de 4 em 4 bytes, Tamanho na posição 100  
Considerar apenas números positivos. Gravar resposta em R1. R1 – menor = 0, r2 índice, r3 contador, r4 elemento Ai

0: Lw r1, 200(r0); r1 ← Primeiro elemento

4: addi r3, r0, 4 ; r3 = I = 4

8: Lw r2, 100(r0); r2 ← T

12: LOOP Beq r2, r0, FIM if R2=0 FIM

16: Lw r4, 200(r3); r4 ← Ai

20: Addi r3, r3, 4 i++

24: addi r2, r2, -1

28: Slt r5, r1, r4; r5 = 1 if R1 < r4 => R1 menor else 0

32: Bne r5, r0, Cont ; If r5<>0, r1 é menor, pula, else troca

36: Addi r1, r4, 0

40: cont jump Loop

44: Fim: xor r5, r5, r5

48: and r6, t6, r7

Para o pipeline do código Verilog modificado com BEQ e JUMP no **segundo estágio** e **unidade de hazard e forward para BEQ**, Suponha um vetor com 10 elementos, onde o menor é trocado 4 vezes durante a execução do loop. Quantos Stalls/Flush ocorrem no código no total e quais os motivos para cada tipo de stall/flush ? Quantos ciclos irá demorar para o PC ter o valor 48 ?

5) Refaça para os outros modelos de processador: pipeline simples, apenas Forward, Forward/Hazard com Beq no 4 estágio.