Exercício

estágio não tenha sua unidade principal acessada, represente por '-'. Bolhas podem ser representadas por 'X' ou 'X $^{\kappa\prime}$ onde K significa o estágio onde instrução está parada. Considere que, caso haja uma Execute o trecho de código abaixo colocando todos os estágios do pipeline (B, D, E, M, W). Caso um dependência, esta deve ser resolvida com bolhas ij

	21						Ciclo	<u>o</u>					
	Instrução	-	7	m	4	1 2 3 4 5 6 7 8 9 10 11 12	9	7	∞	6	10	11	12
<u>:</u>	lui \$s1,0												
addi	addi \$s1, \$s1, 2												
<u>=</u>	lui \$s2,0												
NS.	sw \$s1, (0x11000000) \$s3												
SW	sw \$s2, (0x11000000) \$s3												

Exercício

(Baseado no POSCOMP 2005 - 21) Considere uma CPU usando pipeline com e de instruções separadas, escrita no banco de registradores na borda de subida do relógio e leitura na borda de descida do relógio e o conjunto de instruções a seguir: com memórias de dados 5 estágios (B, D, E, M, W), 2

Quantos ciclos são gastos para a execução deste código? 30 17 16 11 10 \widehat{G} \widehat{G} \widehat{G} \widehat{G} 11: lw \$2, 100(\$5) 12: add \$1, \$2, \$3 13: sub \$3, \$2, \$1 14: sw \$2, 50(\$1) 15: add \$2, \$3, \$3 16: sub \$2, \$2, \$4

	7	7	1 2 3	4	5	5 6 7 8	7	∞	6	10	11	12	13	14	15	9 10 11 12 13 14 15 16 17 18	17	18
lw \$2, 100(\$5)																		
add \$1, \$2, \$3																		
sub \$3, \$2, \$1																		
sw \$2, 50(\$1)																		
add \$2, \$3, \$3																		
sub \$2, \$2, \$4																		

Exercício

- dados seja executada em 5ns. O tempo total para que 5 instruções sejam Suponha que cada uma das etapas do processador (busca, decodificação, execução, leitura ou escrita de dados em memória e escrita em registrador) (POSCOMP 2008 - 54) Um processador tem cinco estágios de pipeline. executadas em pipeline, supondo que não haja dependência de entre as instruções é: ω.
 - a) 15ns
- b) 25ns
- c) 30ns
- d) 45ns
- e) 50ns

Exercício

4. Considere também que vet é um vetor de 4 inteiros iniciados com todas as posições em 0 e endereço Ilustre a execução do trecho de código no MIPS pipeline, colocando os estágios e bolhas e flush necessários para sua execução correta. Considere que o código inicia com \$t1 = 3, \$s0 = &vet[0] e \$t2 = de memória 0x100. Ao final apresente o conteúdo de vet e o valor dos registradores utilizados. OBS. Coloque F em um estágio que está em flush 4

Linstrução Ciclo	ini: addi \$t1, \$t1, 1	bgt \$t1, \$t2, fim	sw \$t1, 0(\$s0)	addi \$s0, \$s0, 4	THE STATE OF THE S	.0, 10	
Instri	ddi \$t	gt \$t	w st	ddi \$s	ini (i \$v	syscall
	ini: a	Ą	Ŋ	์	Ĺ	fim: li \$v0, 10	Ŋ.

\$40

\$s0

\$t2 =

\$t1 =

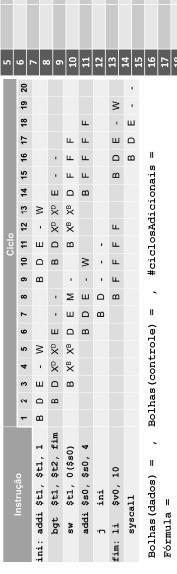
پ اا

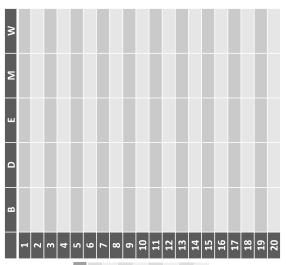
vet[]

Exercício

Um diagrama que relaciona ciclos de instrução com estágios do pipeline, tal como ilustrado na direita, permite compreender melhor a execução em um pipeline 'n.

- Reescreva a execução abaixo no formato proposto 1
- Diga quantas bolhas foram inseridas devido aos dados e devido ao controle - 1
 - Quantos ciclos a mais ocorreram por ter hazards de controle e dados? - 1
- Qual a fórmula básica do pipeline sem considerar hazards? Entenda e comente o efeito das bolhas de dados e de controle na fórmula





Exercício

- [estágio sem operação], B [Busca], D [Decodificação], E [Execução], M [operação com a Dado o trecho de código abaixo em linguagem de montagem do MIPS, preencha o diagrama de execução abaixo. Convenção: X^k [bolha no estágio k], F [Flush], memória de dados), W [Write-back]

ø.

Conteúdos iniciais da memória e dos registradores relevantes: \$t4=0x000 \$t0=0x100, \$t2=0x000, \$t3=0x100, [0x100-0x103] = 0x00000001 [0x200-0x203] = 0x00000010 [0x300-0x303] = 0x00000100 = 0x00000001 0×00000010 [0x400-0x403] [0x500-0x503] Mem Mem Mem Mem Mem

Reg	Instrução	-	2	3	4	2	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20	8	6	10	1	12	13	14	15	16	17	18	19	20
	ori \$t1, \$zero, 0x100																			
	addi \$t0, \$zero, 1																			
	laco: add \$t2, \$t3, \$t1																			
	lw \$t4, 0x100(\$t2)																			
	sw \$t0, 0x200(\$t2)																			
	subi \$t0, \$t0, 1																			
	bne \$t0, \$zero, laco																			
	lw \$t2, 0x400(\$t1)																			
	add \$t4, \$t2, \$t0																			