Aluno: Data

1ª Questão) (1.0 ponto) cada item 0.2 ponto

A CPI de diferentes tipos de instruções é dado na tabela a seguir

	Aritmética	Load/Store	Desvio
a	2	10	3
b	1	10	4

Assuma que a execução de dado programa possa ser quebrada nas seguintes instruções

	Instruções (em milhões)
Aritmética	500
Load/Store	300
Desvio	100

a) Qual é o tempo de execução para o processador ,se a frequência de operação é 5GHz?
 Para a.

$$T_{exec} = \frac{500 * 10^6 * 2 + 300 * 10^6 * 10 + 100 * 10^6 * 3}{5 * 10^9} = 0.86 \text{ segundos}$$

Para b

$$T_{exec} = \frac{500 * 10^6 * 1 + 300 * 10^6 * 10 + 100 * 10^6 * 4}{5 * 10^9} = 0.78 \ segundos$$

Critério

0.1 - realizou a multiplicação do CPI vezes o número de instrução

b) Suponha que um conjunto de instruções aritméticas poderosas sejam adicionadas para o conjunto de instrução. Em média, através destas instruções aritméticas mais poderosas, nós podemos reduzir o número de instruções aritméticas necessárias para executar um programa por 25%, a custo de um aumento no tempo de ciclo de clock em 10%. Esta é uma boa escolha de projeto? Por que?

$$T_{old} = (NINST_{Arit} * CPI_{Arit} + NINST_{Load} * CPI_{Load} + NINST_{Desvio} * CPI_{Desvio}) * Tcc$$

$$T_{new} = (0.75 * NINST_{Arit} * CPI_{Arit} + NINST_{Load} * CPI_{Load} + NINST_{Desv} * CPI_{Desvio}) * 1.1 * Tcc$$

$$T_{new} = (0.825 * NINST_{Arit} * CPI_{Arit} + NINST_{Load} * CPI_{Load} * 1.1 + NINST_{Desv} * CPI_{Desv} * 1.1) * Tcc$$

$$T_{new} - T_{old} = (-0.175 * NINST_{Arit} * CPI_{Arit} + NINST_{Load} * CPI_{Load} * 0.1 + NINST_{Desv} * CPI_{Desv} * 0.1) * Tcc$$

Analisando $T_{new} - T_{old}$, pode-se observar que esta escolha, dependendo da configuração de instruções, pode resultar em um ganho ou não.

Critério

0.1 - dizer que pode não ser uma boa escolha

0.1 - justificativa

0.05 – bom para o problema em questão

c) Supondo que nós encontramos uma forma de dobrar a performance das instruções aritméticas, qual é o speed-up global desta nova maquina?

•	Instruções (em milhões)	a) CPI	a)	b) CPI	b)
Aritmética	500	2	1000	1	500
Load/Store	300	10	3000	10	3000
Desvio	100	3	300	4	400

$$\begin{split} T_{new} &= T_{old} * \left[\left(1 - \frac{1000}{4300} \right) + \frac{\frac{1000}{4300}}{2} \right] = 0.8837 \\ speedup &= \frac{T_{old}}{T_{new}} = 113.16\% \\ T_{new} &= T_{old} * \left[\left(1 - \frac{500}{3900} \right) + \frac{\frac{500}{3900}}{2} \right] = 0.9359 \\ speedup &= \frac{T_{old}}{T_{new}} = 106.85\% \end{split}$$

c) Dada o mix de instruções apresentado na tabela abaixo e a suposição que uma instrução aritmética requer 2 ciclos, uma instrução load/store requer 6 ciclos e a instrução de desvio requer 3 ciclos, encontre o CPI médio.

	Aritmética	Load / store	Desvio
a	60 %	20 %	20 %
b	80 %	15 %	5 %

a)
$$CPI_{medio} = 0.6 * 2 + 0.2 * 6 + 0.2 * 3 = 3.0$$

b) $CPI_{medio} = 0.8 * 2 + 0.15 * 6 + 0.05 * 3 = 2.65$

d) Considerando os dados do item d), para uma melhoria de 25% na performance, quantos ciclos, em média, uma instrução aritmética deve tomar considerando que a instrução load/store e desvio não sofram nenhuma melhoria.

Para a

$$\frac{CPI_{old}-CPI_{new}}{CPI_{new}}=0.25$$

$$\frac{3.0-CPI_{new}}{CPI_{new}}=0.25$$

$$CPI_{new} = 2.4$$

$$0.6 * nciclo + 0.2 * 6 + 0.2 * 3 = 2.4$$

$$nciclo = 1.0$$

Para b

$$\frac{CPI_{old} - CPI_{new}}{CPI_{new}} = 0.25$$

$$\frac{2.65 - CPI_{new}}{CPI_{new}} = 0.25$$

$$CPI_{new} = 2.12$$

$$0.8 * nciclo + 0.15 * 6 + 0.05 * 3 = 2.12$$

$$nciclo = 1.3$$

2ª Questão) (1.0 ponto) Considere o conjunto de instruções abaixo

					Latência
I1	div	F6	F6	F4	4
I2	lw	F2	45(R3)		1
I3	mult	F0	F2	F4	3
I4	div	F8	F6	F2	4
I5	sub	F10	F0	F6	1
I6	add	F6	F8	F2	1

- a) (0.5 ponto) Identifique as situações de dependência (WAW, WAR, RAW) na seguinte sequência de código acima, do MIPS64:
 - (0.1 ponto) WAW I1 e I6;
 - (0.1 ponto) WAR I4 e I6; I5 e I6, I6 e I1
 - (0.3 ponto) RAW I1 e I4; I1 e I5; I2 e I3; I2 e I4; I2 e I6; I3 e I5; I4 e I6
- b) Apresente uma sequência de termino em ordem e outra em fora de ordem (que execute no menor tempo)

			uciiciu											
1	2			<u>1</u>	<u>2</u>	3	4		<u>3</u>	5	<u>4</u>	6	<u>5</u>	<u>6</u>
1	2	2	3	1	4	3	5	5	4	6	6			

3ª Questão) (1.0 ponto) Assinale verdadeiro (V) ou falso (F).

(Lembre-se: um item assinalado incorretamente anula um item assinalo corretamente)

- (V) RISC apresenta poucos formatos de instrução e muitos registradores de uso genérico, enquanto CISC possui instruções de vários comprimentos (no mesmo conjunto)
- (F) **Arquitetura superpipeline** baseia-se no aumento das unidades funcionais de forma que seja possível executar mais de uma instrução em cada ciclo de relógio
- (V) Na RISC a complexidade esta no compilador, enquanto na CISC a complexidade esta no microprograma
- (F) Uma **arquitetura super-escalar** consiste em aumentar o número de estágios da pipeline, conseguindo diminuir Tcc e aumentar a frequência de relógio.
- (V) Arquitetura vetorial especifica uma série de operações a realizar em vários dados, numa só instrução
- (V) Uma arquitetura com grau de grau de super-escalaridade igual a 2 apresenta 2 ciclos de penalização (5 instruções) nos saltos previstos incorretamente
- (F) Programas compilados para arquitetura CISC possuem garantia que serão menores que os compilados para RISC.
- (F) No mecanimo de write back uma escrita modifica o dado na cache e memória juntos
- (V) No caso em que não há escalonamento dinâmico, as instruções são emitidas pela ordem com que são geradas pelo compilador, executadas pela mesma ordem e terminadas ainda em ordem

(V) Tamanhos e posições das instruções são fixos e alinhados de acordo com o tamanho de uma palavra em RISC.

4ª. Questão) **(2.0 ponto**) Assuma que o seguinte código é executado sobre um processador pipeline com 5 estágio, com adiantamento e um preditor de desvio (o qual assume que todo desvio é tomado)

Label1: lw \$1, 40 (\$6)

beg \$2, \$3, Label2 //tomado

add \$1, \$6, \$4

Label2: beq \$1, \$2, Labe11// não tomado

sw \$2, 20 (\$4) and \$1, \$1, \$4

 a) (1.0 ponto) Desenhe o diagrama de execução para este código, assumindo que não há slots de atraso e o que desvio executa no estágio EX

Tomado, significa que o desvio deve ser executado, no entanto, o pípeline somente sabe desta informação quando estíver no estagio EX ou no estágio DI caso seja assumido um preditor. Sem preditor, significa que a instrução será buscada na sequencia e os desvios serão resolvidos no estagio EX

Ciclos	1	2	3	4	5	6	7	8	9	10	11	
lw \$1, 40 (\$6)	BI	DI	EX	MEM	WB							
beq \$2, \$3, Label2		BI	DI	EX	MEM	WB		Desvio	será detectado em EX			
add \$1, \$6, \$4			BI	DI	X	X	X	Instruç	ão buscada incorretamente			
beq \$1, \$2, Labe11				BI	X	X	X	X	Buscad	a Incorre	tamente	
beq \$1, \$2, Labe11					BI	DI	EX	MEM	WB	Devio 1	não realiz	zado
sw \$2, 20 (\$4)						BI	DI	EX	MEM	WB		
and \$1, \$1, \$4							BI	DI	EX	MEM	WB	

Usando preditor no estágio DI, significa que no estágio DI, será analisado se a instrução é um desvio, caso seja, sempre será assumido o desvio. Se o preditor errar, todas as instruções buscada devem ser desconsideradas

Ciclos de Clock	1	2	3	4	5	6	7	8	9	10	11	12		
lw \$1, 40 (\$6)	BI	DI	EX	MEM	WB									
beq \$2, \$3, Label2		BI	DI	EX	MEM	WB		Preditor atua no estagio DI, buscando a instrução						
								correspondente ao Label2 e descartando a						
								instrução buscada add \$1, \$6, \$4						
add \$1, \$6, \$4			BI	X	X	X	X	Instru	ıção de	escartada	após o ní	vel EX. Neste		
								estagi	io, é v	erificado	que o des	vio anterior foi		
								tomac	do corre	etamente				
beq \$1, \$2, Labe11				BI	DI	EX	MEM	WB	Predi	tor atua	no estagio	DI, buscando a		
									instru	ıção cor	respondente	ao Label1 e		
									desca	rtando a i	instrução bu	scada		
									lw \$1	, 40 (\$6)				
lw \$1, 40 (\$6)					BI	DI	X	X	X					
beq \$2, \$3, Label2						BI	X	X	X	X	No está	gio EX da		
											instrução	beq \$1, \$2,		
								Label1, descobre erro de						
								predição						
sw \$2, 20 (\$4)							BI	DI	EX	MEM	WB			
and \$1, \$1, \$4								BI	DI	EX	MEM	WB		

Critério - foi aceito um dos diagramas acima

b) (0.5 ponto) Qual é o speed-up alcançado ao mover a execução de desvio para o estagio ID. Assuma que a comparação no estágio ID não afeta o tempo de ciclo de clock.

Ciclos de Clock	1	2	3	4	5	6	7	8	9	10	
lw \$1, 40 (\$6)	BI	DI	EX	MEM	WB						
beq \$2, \$3, Label2		BI	DI	EX	MEM	WB					
add \$1, \$6, \$4			BI	X	X	X	X				
beq \$1, \$2, Labe11				BI	DI	EX	MEM	WB			
sw \$2, 20 (\$4)					BI	DI	EX	MEM	WB		
and \$1, \$1, \$4						BI	DI	EX	MEM	WB	

c) (0.5 ponto) Um esquema para minimizar os efeitos dos "conflitos de controle" (causados por mudanças no fluxo de controle durante a execução das instruções) é baseado na predição da ocorrência de desvios. Um esquema sofisticado de predição trabalha com a hipótese de que desvios para endereços anteriores sempre se realizam e que desvios para endereços posteriores nunca se realizam. Explique como isso é feito e por que o esquema tem predição tem uma taxa de acerto tão elevada (benchmarks mostraram taxas de acerto da ordem de 90%). Compare o desempenho com um preditor de 1 e 2 bits.

Esta predição é baseada no fato que os laços devem ser realizados, enquanto que os testes condicionais (else) devem sempre falhar. Logo se consideramos que um certo loop é executado duas vezes e que loop termina com 10 iterações. Neste caso, teremos nas duas vezes que forem executado, ocorre um erro somente na última iteração. Logo um acerto de 90%. Se considerarmos um preditor de um bit, teremos um erro na primeira iteração do loop e um erro na última iteração, resultado em 80% de acerto. Já um preditor de dois bts, teremos na primeira vez que o loop for executado uma precisão de 70% e nas próximas execução uma precisão de 90%.

5ª Questão) (3.0 pontos) Considere a seguinte sequencia de instruções, e assuma que estas sejam executadas em um pipeline com 5 estágios (BI(Busca), DI (Decodificação), EX (Execução) MEM (Memória) WB (Write-back))

Sequ	Sequencia Instruções										
lw	\$1, 40 (\$6)										
add	\$2, \$3, \$1										
sw	\$2, 20(\$4)										
add	\$1, \$6, \$4										
and	\$1, \$1, \$4										

a) Quais dependências são conflitos (hazards) que podem ser resolvidos com adiantamento? Quais dependências que são conflitos e irão provocar a parada (bolhas) na execução?

Cada item -0.5 ponto

Sem adiantamento

SCII	Sem adiantamento															
Cicle	os	1	2	3	4	5	6	7	8	10	11	12	13	14	15	16
lw	\$1, 40 (\$6)	BI	DI	EX	MEM	WB										
add	\$2, \$3, \$1		BI	DI	X	X	EX	MEM	WB	Esperan	do \$s1					
sw	\$2, 20(\$4)			BI	X	X	DI	X	X	EX	MEM	WB	Esperan	do \$S2		
add	\$1, \$6, \$4						BI	X	X	DI	EX	MEM	WB			
and	\$1, \$1, \$4									BI	DI	X	X	EX	MEM	WB
										Esperando \$s2			ndo \$s2			

Com adiantamento

COIII	Com administrative											
Ciclo	os	1	2	3	4	5	6	7	8	10	11	
lw	\$1, 40 (\$6)	BI	DI	EX	MEM	WB						
add	\$2, \$3, \$1		BI	DI	X	EX	MEM	WB				
sw	\$2, 20(\$4)			BI	X	DI	EX	MEM	WB			
add	\$1, \$6, \$4					BI	DI	EX <	MEM	WB		
and	\$1, \$1, \$4						BI	DI	EX	MEM	WB	

Conflitos

```
lw $1, 40 ($6) e add $2, $3, $1 - RAW (pode produzir duas bolhas)
lw $1, 40 ($6) e add $1, $6, $4, - WAW (não pode ser resolvido com adiantamento)
lw $1, 40 ($6) e add $1, $1, $4, - WAW, RAW (não gera bolha neste exemplo)
add $2, $3, $1 e sw $2, 20 ($4) - RAW
add $1, $6, $4, and $1, $1, $4 - RAW, WAW
```

Conflitos de dados resolvidos 0.25

```
add $s2, $s3, $1, sw $2, 20($4)
add $1, $6, $4, and $1, $1, $4
Conflitos que irão provocar uma parada 0.25
lw $1, 40 ($6), add $2, $3, $1
```

Se não há adiantamento ou detecção de conflito, insira nops para assegura a execução correta e desenhe o diagrama de execução do pipeline para este código

Sequ	Sequencia Instruções								
lw	\$1,40 (\$6)								
nops									
nops									
add	\$2, \$3, \$1								
nops									
nops									
SW	\$2, 20(\$4)								
add	\$1, \$6, \$4								
nops									
nops									
and	\$1, \$1, \$4								

Ciclos	1	2	3	4	5	6	7	8	10	11						
lw \$1, 40 (\$6)	BI	DI	EX	MEM	WB											
Nops		X	X	X	X	X										
Nops			X	X	X	X	X									
add \$2, \$3, \$1				BI	DI	EX	MEM	WB								
Nops					X	X	X	X	X							
Nops						X	X	X	X	X						
sw \$2, 20(\$4)							BI	DI	EX	MEM	WB					
add \$1, \$6, \$4								BI	DI	EX	MEM	WB				
Nops									X	X	X	X	X			
Nops										X	X	X	X	X		
and \$1, \$1, \$4											BI	DI	EX	MEM	WB	

Cada conjunto de nops – 0.1 diagrama 0.2

c) Repita o item anterior, mas adicione nops somente quando um conflito não pode ser evitado por mudando ou rearranjando estas instruções. Você pode assumir o registrador R7 para guardar valores temporários em seu código modificado.

Renomeando o add \$2, \$3, \$1, para add \$7, \$3, \$1

add \$1, \$6, \$4, and \$1, \$1, \$4, \rightarrow add \$8, \$6, \$4, and \$8, \$8, \$4,

Sequencia Instruções	Sequencia Instruções
lw \$1, 40 (\$6)	lw \$1, 40 (\$6)
add \$2, \$3, \$1	add \$7, \$6, \$4
sw \$2, 20(\$4)	nops
add \$1, \$6, \$4	add \$2, \$3, \$1
and \$1, \$1, \$4	nops
	and \$7, \$7, \$4
	sw \$2, 20(\$4)

d) Um conflito estrutural (duas instruções tentando acessar a memória) pode ser resolvido pelo compilador inserindo uma instrução nops?

Não, pois como o conflito estrutural mencionado é devido a dois acessos a memória. Logo adicionar um nops, não resolve, pois teremos que realizar uma acesso a memória para buscar a instrução.

Não -0.2

Justificativa - 0.3

e) Suponha as instruções abaixo. Qual o procedimento a ser adotado pela unidade de detecção de conflito load \$1,(10) \$2

add \$2, \$1, \$3

Deve se inserir uma bolha (0.15 ponto). Neste caso, todos os sinais de controle em 0 (zero) para os estágios EX, MEM e ER. Estes valores de sinais de controle são passados adiante a cada ciclo de relógio, produzindo o efeito desejado (nenhum registrador ou memória é escrito)

f) Apresente o teste de conflito realizado no estagio EX e MEM pela unidade de adiantamento.

Conflitos no Estágio EX se (EX/MEM.EscReg = 1

```
e (EX/MEM.RegistradorRd 6= 0)
e (EX/MEM.RegistradorRd = DI/EX.RegistradorRs)) Adianta.A = 10

se (EX/MEM.EscReg = 1
e (EX/MEM.RegistradorRd 6= 0)
e (EX/MEM.RegistradorRd = DI/EX.RegistradorRt)) Adianta.B = 10

Conflitos no Estágio MEM
se (EX/MEM.EscReg = 1
e (EX/MEM.RegistradorRd 6= 0)
e (MEM/ER.RegistradorRd = DI/EX.RegistradorRs)) Adianta.A = 01

se (EX/MEM.EscReg = 1
e (EX/MEM.EscReg = 1
e (EX/MEM.EscReg = 1
e (EX/MEM.RegistradorRd 6= 0)
e (MEM/ER.RegistradorRd 6= 0)
e (MEM/ER.RegistradorRd = DI/EX.RegistradorRt)) Adianta.B = 01
```

6ª Questão) (1.0 ponto) Considere o trecho de programa no quadro abaixo e os conteúdos iniciais de registradores e posições de memória relevantes. Convenções: X − bolha, F − flush do pipeline, −− para estágio não usado, −→ adiantamento ou leitura após escrita no mesmo ciclo. Estágios do pipeline: BI(Busca), DI (Decodificação), EX (Execução) MEM (Memória) WB (Writeback)

,	'		
Ī	addi	\$t4, \$zero, 2	Conteúdos iniciais da memória e dos registradores relevantes:
	root : add	\$t1, \$t2, \$t3	\$t1=0x100, \$t2=0x100, \$t3=0x100, \$t4=0x100
	lw	\$t3, 0x100(\$t1)	Mem [0x100-0x103]= 0x002345AB
	sw	\$t3, 0x200(\$t1)	Mem [0x200-0x203]= 0x0000000A
	subi	\$t4, \$t4, 2	Mem [0x300-0x303]= 0x00000000
	beq	\$t4, \$t3, root	Mem [0x400-0x403]= 0x00CD5F00
	addi	\$t3, \$t3, 0x100	

a) Simule a execução completa do programa (considere unidade de adiantamento).

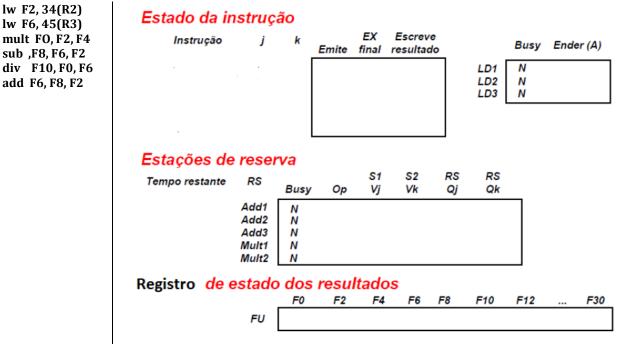
Ciclos	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
addi \$t4, \$zero, 2	BI	DI	EX	M	WB																		
add \$t1, \$t2, \$t3		BI	DI	EX	M	WB																	
lw \$t3, 0x100(\$t1)			BI	DI	EX	M	WB																
sw \$t3, 0x200(\$t1)				BI	DI	X	EX	M	WB														
subi \$t4, \$t4, 2					BI	X	DI	EX 、	M	WB													
beq \$t4, \$t3, root							BI	DI	XX	M	WB				Assur	nindo (decisão	sobre	o des	vio no	estagio	M,	
															poder	ia ser n	o estági	io EX					
addi \$t3, \$t3, 0x100								BI	DI	EX	F	F							etamen	te, será	descart	ada. `	Veja
															que e	ra a últi	ma inst	rução.					
add \$t1, \$t2, \$t3											BI	DI	EX <	M	WB								
lw \$t3, 0x100(\$t1)												BI	DI	EX	M	WB							
sw \$t3, 0x200(\$t1)													BI	DI	X	EX	M	WB					
subi \$t4, \$t4, 2														BI	X	DI	EX	M	WB				
beq \$t4, \$t3, root																BI	DI	EX	M	WB			
addi \$t3, \$t3, 0x100																	BI	DI	EX	M	WB		

b) O que a unidade de adiantamento (forward) está fazendo durante o quinto ciclo de execução? Se algumas comparações estiverem sendo feitas, mencione-as.

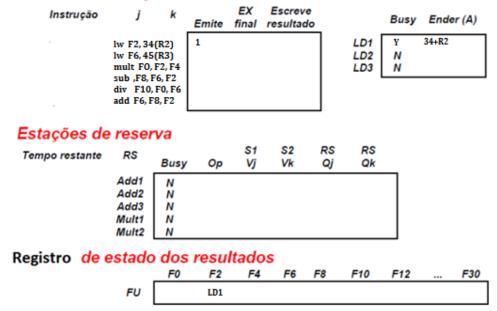
A unidade de adiantamento está realizando duas ações em paralelo:

- 1 Comparando os registradores-fonte da instrução LW \$t3,0x100(\$t1) com os registradores-destino das instruções ADD \$t1, \$t2, \$t3 e ADDI \$t4, \$zero, 2.
- 2 Adiantando o valor correto de \$t1da instrução ADD \$t1, \$t2, \$t3 para a instrução LW \$t3,0x100(\$t1).
- **7ª. Questão**) (**1.0 ponto**) Mostrar o resultado (após décimo quinto ciclo) do uso Tomasulo para a sequência de instruções, considerando-se que a instrução LD leva 1 ciclo para execução; MULD, 6 ciclos. ADDD e SUBD levam 3 ciclos; e DIVD, 20 ciclos.
- 0.8 Estado Principal
- 0.1 Estado de reserva

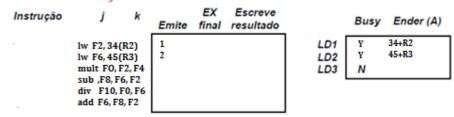
Solução passo a passo. Foi analisado os quadros.



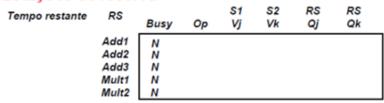
Passo 1



Passo2



Estações de reserva

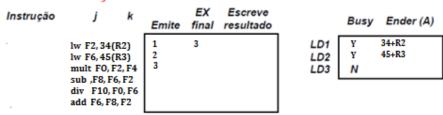


Registro de estado dos resultados

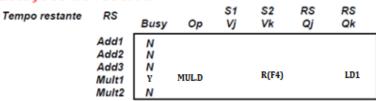
	F0	F2	F4	F6	F8	F10	F12	 F30
FU		LD1		LD2				

3ª Passo

Estado da instrução



Estações de reserva



Registro de estado dos resultados

	F0	F2	F4	F6	F8	F10	F12	 F30
FU	Mult1	LD1		LD2				

⁴ª Passo

Instrução	j k	Emite	EX final	Escreve resultado	_	Busy	Ender (A)
	lw F2, 34(R2) lw F6, 45(R3) mult F0, F2, F4 sub ,F8, F6, F2 div F10, F0, F6 add F6, F8, F2	1 2 3 4	3 4	4	LD1 LD2 LD3	N Y N	45+R3

Estações de reserva

Tempo restante	RS	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
	Add1 Add2	Y N	SUB.D		M(A1)	LD2	
6	Add3 Mult1 Mult2	N Y N	MUL.D	M(A1)	R(F4)		

Registro de estado dos resultados

	F0	F2	F4	F6	F8	F10	F12	 F30
FU	Mult1	M(A1)		LD2	Add1			

5ª Passo

Estado da instrução

Instrução	j k	Emite	EX final	Escreve resultado	_	Busy	Ender (A)
	lw F2, 34(R2) lw F6, 45(R3) mult F0, F2, F4 sub ,F8, F6, F2 div F10, F0, F6 add F6, F8, F2	1 2 3 4 5	3 4	4 5	LD1 LD2 LD3	N N N	

Estações de reserva

Tempo restante	RS	Busy	Ор	S1 Vj	S2 Vk	RS Qj	RS Qk
3	Add1 Add2	Y N	SUB.D	M(A2)	M(A1)		
5	Add3 Mult1 Mult2	N Y Y	MUL.D DIV	M(A1)	R(F4) M(A2)	Mult1	

Registro de estado dos resultados

		F2	 F6	F8	F10	F12	 F30
FU	Mult1	M(A1)	M(A2)	Add1	Mult2		

6ª Passo)

Instrução	j k	Emite	EX final	Escreve resultado		Busy	Ender (A)
	lw F2, 34(R2) lw F6, 45(R3) mult F0, F2, F4 sub ,F8, F6, F2 div F10, F0, F6 add F6, F8, F2	1 2 3 4 5 6	3 4	4 5	LD1 LD2 LD3	N N N	

Estações de reserva

Tempo restante	RS	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
2	Add1 Add2	Y Y	SUB.D ADD.D	M(A2)	M(A1) M(A1)	Add1	
4	Add3 Mult1 Mult2	N Y Y	MUL.D DIV	M(A1)	R(F4) M(A2)	Mult1	

Registro de estado dos resultados

	F0	F2	F4	F6	F8	F10	F12	 F30
FU	Mult1	M(A1)		Add2	Add1	Mult2		

7ª Passo

Estado da instrução

Instrução	j k	Emite	EX final	Escreve resultado		Busy	Ender (A)
	lw F2, 34(R2) lw F6, 45(R3) mult F0, F2, F4 sub ,F8, F6, F2 div F10, F0, F6 add F6, F8, F2	1 2 3 4 5 6	3 4	4 5	LD1 LD2 LD3	N N N	

Estações de reserva

Tempo restante	RS	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
1	Add1 Add2	Y Y	SUB.D ADD.D	M(A2)	M(A1) M(A1)	Add1	
3	Add3 Mult1 Mult2	Y Y	MUL.D DIV	M(A1)	R(F4) M(A2)	Mult1	

Registro de estado dos resultados

	F0	F2	F4	F6	F8	F10	F12	 F30
FU	Mult1	M(A1)		Add2	Add1	Mult2		

8ª Passo

Instrução	j k	Emite	EX final	Escreve resultado		Busy	Ender (A)
	lw F2, 34(R2) lw F6, 45(R3) mult F0, F2, F4 sub ,F8, F6, F2 div F10, F0, F6 add F6, F8, F2	1 2 3 4 5 6	3 4 8	4 5	LD1 LD2 LD3	N N N	

Estações de reserva

Tempo restante	RS	Busy	Ор	S1 Vj	S2 Vk	RS Qj	RS Qk
0	Add1 Add2	Y Y	SUB.D ADD.D	M(A2)	M(A1) M(A1)	Add1	
2	Add3 Mult1 Mult2	N Y Y	MUL.D DIV	M(A1)	R(F4) M(A2)	Mult1	

Registro de estado dos resultados

	F0	F2	F4	F6	F8	F10	F12	 F30
FU	Mult1	M(A1)		Add2	Add1	Mult2		

9ª Passo

Estado da instrução

Instrução	j k	Emite	EX final	Escreve resultado	_	Busy	Ender (A)
	lw F2, 34(R2) lw F6, 45(R3) mult F0, F2, F4 sub ,F8, F6, F2 div F10, F0, F6 add F6, F8, F2	1 2 3 4 5 6	3 4 8	4 5 9	LD1 LD2 LD3	N N N	

Estações de reserva

Tempo restante	RS	Busy	Ор	S1 Vj	S2 Vk	RS Qj	RS Qk
3	Add1 Add2 Add3	N Y	ADD.D	(M-M)	M(A1)		
1	Mult1 Mult2	Y Y	MUL.D DIV	M(A1)	R(F4) M(A2)	Mult1	

Registro de estado dos resultados

	F0	F2	F4	F6	F8	F10	F12	•••	F30
FU	Mult1	M(A1)		Add2	(M-M)	Mult2			

10ª Passo

Instrução	j k	Emite	EX final	Escreve resultado		Busy	Ender (A)
	lw F2, 34(R2) lw F6, 45(R3) mult F0, F2, F4 sub ,F8, F6, F2 div F10, F0, F6 add F6, F8, F2	1 2 3 4 5 6	3 4 10 8	4 5 9	LD1 LD2 LD3	N N N	

Estações de reserva

Tempo restante	RS	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
2	Add1 Add2	N Y	ADD.D	(M-M)	M(A1)		
0	Add3 Mult1 Mult2	N Y Y	MUL.D DIV	M(A1)	R(F4) M(A2)	Mult1	

Registro de estado dos resultados

	F0	F2	F4	F6	F8	F10	F12	•••	F30
FU	Mult1	M(A1)		Add2	(M-M)	Mult2			

11ª Passo

Estado da instrução

Instrução	j k	Emite	EX final	Escreve resultado		Busy	Ender (A)
	lw F2, 34(R2) lw F6, 45(R3) mult F0, F2, F4 sub ,F8, F6, F2 div F10, F0, F6 add F6, F8, F2	1 2 3 4 5 6	3 4 10 8	4 5 11 9	LD1 LD2 LD3	N N N	

Estações de reserva

Tempo restante	RS	Busy	Ор	S1 Vj	S2 Vk	RS Qj	RS Qk
1	Add1 Add2 Add3	N Y N	ADD.D	(M-M)	M(A1)		
20	Mult1 Mult2	N Y	DIV	M*F4	M(A2)		

Registro de estado dos resultados

	F0	F2	F4	F6	F8	F10	F12	•••	F30
FU	(M*F4)	M(A1)		Add2	(M-M)	Mult2			

12ª Passo)

Instrução	j k	Emite	EX final	Escreve resultado		Busy	Ender (A)
	lw F2, 34(R2) lw F6, 45(R3) mult F0, F2, F4 sub ,F8, F6, F2 div F10, F0, F6 add F6, F8, F2	1 2 3 4 5 6	3 4 10 8	4 5 11 9	LD1 LD2 LD3	N N N	

Estações de reserva

Tempo restante	RS	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
0	Add1 Add2 Add3	N Y N	ADD.D	(M-M)	M(A1)		
19	Mult1 Mult2	N Y	DIV	M*F4	M(A2)		

Registro de estado dos resultados

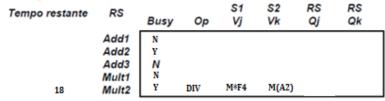
	F0	F2	F4	F6	F8	F10	F12	 F30
FU	(M*F4)	M(A1)		Add2	(M-M)	Mult2		

13ª Passo)

Estado da instrução

Instrução	j k	Emite	EX final	Escreve resultado		Busy	Ender (A)
	lw F2, 34(R2) lw F6, 45(R3) mult F0, F2, F4 sub ,F8, F6, F2 div F10, F0, F6 add F6, F8, F2	1 2 3 4 5 6	3 4 10 8	4 5 11 9	LD1 LD2 LD3	N N N	

Estações de reserva



Registro de estado dos resultados

	F0	F2	F4	F6	F8	F10	F12	 F30
FU	(M*F4)	M(A1)		(M-M+M)	(M-M)	Mult2		

14^a Passo)

Instrução	j k	Emite	EX final	Escreve resultado		Busy	Ender (A)
	lw F2, 34(R2) lw F6, 45(R3) mult F0, F2, F4 sub ,F8, F6, F2 div F10, F0, F6 add F6, F8, F2	1 2 3 4 5 6	3 4 10 8	4 5 11 9	LD1 LD2 LD3	N N N	

Estações de reserva

Tempo restante	RS	Busy	Ор	S1 Vj	S2 Vk	RS Qj	RS Qk	
	Add1 Add2	N Y						٦
	Add3 Mult1	N N						
17	Mult2	Y	DIV	M*F4	M(A2)			

Registro de estado dos resultados

	F0	F2	F4	F6	F8	F10	F12	 F30
FU	(M*F4)	M(A1)		(M-M+M) (M-M)	Mult2		

15^a Passo)

Estado da instrução

Instrução	j k	Emite	EX final	Escreve resultado	_	Busy	Ender (A)
	lw F2, 34(R2) lw F6, 45(R3) mult F0, F2, F4 sub ,F8, F6, F2 div F10, F0, F6 add F6, F8, F2	1 2 3 4 5 6	3 4 10 8	4 5 11 9	LD1 LD2 LD3	N N N	

Estações de reserva

Tempo restante	RS	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
	Add1 Add2 Add3	N Y N					
16	Mult1 Mult2	N Y	DIV	M*F4	M(A2)		

Registro de estado dos resultados

	F0	F2	F4	F6 F	-8	F10	F12	 F30
FU	(M*F4)	M(A1)		(M-M+M) (И-М)	Mult2		