

Universidade de São Paulo  
Escola de Artes, Ciências e Humanidades  
Docente: Prof. Dr. Clodoaldo A M Lima.  
Discente: \_\_\_\_\_

No. USP: \_\_\_\_\_

### Atenção

No USP com final 2, 4, 6, 8 – resolver as questões 1ª, 3ª, 5ª, 7ª, 9ª, 11ª, 13ª

No USP com final 0, 1, 3, 5, 7, 9 – resolver as questões 2ª, 4ª, 6ª, 8ª, 10ª, 12ª, 14ª

1ª Questão) (1,0 ponto) Um programa roda em 10 segundos em um computador A, que tem um clock de 400 MHz. Está sendo projetada uma nova máquina B, que deverá rodar este programa em 6 segundos. No entanto, essa redução afetará o projeto do resto da CPU. Por essa razão, a máquina B requer 1,2 vezes mais ciclos de clock que a máquina A para um mesmo programa. Qual a frequência de clock da nova máquina?

$$TA = NINST * NCA * TCA$$

$$TB = NINST * NCB * TCB$$

$$TA/TB = NCA/NCB * TCA/TCB$$

$$10/6 = NCA/1.2 * NCA * TCA/TCB$$

$$TCB = 6/10 * 1.2 * TCA$$

$$TCB = TCA/2$$

$$1/FB = 2/FA$$

$$FB = 2 * 400$$

$$FB = 800 \text{ MHz}$$

2ª Questão) (1,0 ponto) Assuma que os estágios individuais de um fluxo de dados tem as seguintes latências

IF	ID	EX	MEM	WB
250 os	350 os	150 ps	300 ps	200 ps

Assuma que instruções executadas pelo processador são quebradas como segue

formato R	Beq	Lw	Sw
45%	20%	20%	15%

a) (0,3 ponto) Qual é o tempo de ciclo em processador com pipeline e sem pipeline?

Sem pipeline

$$TC = 250ps + 350ps + 150ps + 300ps + 200ps = 1250 \text{ ps}$$

Com pipeline

$$TC = 350ps$$

b) (0,3 ponto) Qual a latência da instrução lw em processador com pipeline e sem pipeline?

Sem pipeline

$$\text{Latência} = 1250 \text{ ps}$$

Com pipeline

$$\text{Latência} = 5 \times 350\text{ps} = 1750\text{ ps}$$

c) (0,4 ponto) Se pudermos dividir um estágio do fluxo de dados no pipeline em dois novos estágios, cada com metade do estágio original, qual estágio você dividiria e qual o novo ciclo de clock do processador?

O estágio com maior duração é o DI (0,2 Ponto). Neste caso, o TC seria de 300ps (0,2 ponto)

3ª Questão) (1,0 ponto) Um processador RISC é implementado em duas versões de organização síncrona: uma monociclo, em que cada instrução executa em exatamente um ciclo de relógio, e uma versão pipeline de 5 estágios. Os estágios da versão pipeline são: (1) busca de instrução, (2) busca de operandos, (3) execução da operação, (4) acesso à memória e (5) atualização do banco de registradores. A frequência máxima de operação das organizações foi calculada em 200 MHz para a versão monociclo e 400 MHz para a versão pipeline. Um programa X que executa 200 instruções é usado para comparar o desempenho das organizações. Das 200 instruções, apenas 40% fazem acesso à memória, enquanto as demais operam apenas sobre registradores internos da organização. Assuma, que o programa não apresenta nenhum conflito de dados ou de controle entre instruções que podem estar simultaneamente dentro do pipeline da segunda organização. Calcule o tempo de execução do programa X nas organizações monociclo e pipeline.

Organização da máquina Versão 1 - Monociclo

Período para execução de uma instrução  $[1/(200 \times 10^6)] s = 5\text{ ns}$

Período para execução de 200 instruções  $200 \times 5\text{ ns} = 1.000\text{ ns}$

Período do pipeline  $[1/(400 \times 10^6)] s = 2,5\text{ ns}$

Período para execução de 200 instruções (Observar o princípio da Figura 1)

40% fazem acesso à memória (80 instruções – utilizam 5 estágios)

60% operam apenas sobre registradores internos da organização (120 instruções – utilizam 4 estágios).

Estágio 1: 200 períodos de clock (200 instruções)

Estágio 2: tem início no segundo período de clock para a instrução 1. Logo, para execução deste estágio em todas as instruções, tem-se mais um período de clock em relação ao estágio anterior; Estágio 3: ....

Importante: 40% e 60% (descritos anteriormente) estarão inseridos no tempo necessários à execução do pior caso (5 estágios).

Logo, o período para execução das 200 instruções é:  $(200 + 1 + 1 + 1 + 1) \times 2,5\text{ ns} = 510\text{ ns}$

4ª Questão) (1,0 Ponto) Assinale verdadeiro (V) ou falso (F).

(Lembre-se: um item assinalado incorretamente anula um item assinalado 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 está no compilador, enquanto na CISC a complexidade está 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 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 mecanismo 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

5ª Questão) (1,5 ponto) Usando o sistema de previsão local de desvio de 2 bits, mostrado na Figura1, um certo loop é executado duas vezes:

- a) considerando-se que o estado inicial seja 10, calcular a porcentagem de acertos e erros de previsão, considerando-se que o loop termina com 10 iterações;

Iteração	Estado	Previsão	Realizado	Acerto	Iteração	Estado	Previsão	Realizado	Acerto
1	10→11	S	S	S	11	10→11	S	S	S
2	11→11	S	S	S	12	11	S	S	S
3	11→11	S	S	S	13	11	S	S	S
4	11→11	S	S	S	14	11	S	S	S
5	11→11	S	S	S	15	11	S	S	S
6	11→11	S	S	S	16	11	S	S	S
7	11→11	S	S	S	17	11	S	S	S
8	11→11	S	S	S	18	11	S	S	S
9	11→11	S	S	S	19	11	S	S	S
10	11→10	S	N	N	20	11	S	N	N

1ª Iteração – 90% acerto

2ª Iteração – 90 % acerto

Total – 2 erros, 90% acerto

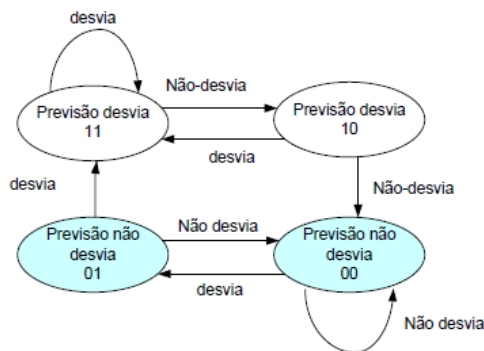
- b) comparar com o caso em que não use esse sistema de previsão, e apenas considere que a **previsão para frente** seja sempre de desvio;

Taxa de acerto 10% - acerta somente na última iteração

- c) comparar com o sistema de previsão de um bit

Considerando estado inicial igual a 0 – 80% acerto

Considerando estado inicial igual a 1 – 85% acerto



6ª Questão) (1,5 ponto) Usando o sistema de previsão local de desvio de 2 bits, mostrado na Figura1, um certo loop é executado duas vezes:

- a) considerando-se que o estado inicial seja 11, calcular a porcentagem de acertos e erros de previsão, considerando-se que o loop termina com 10 iterações;

Iteração	Estado	Previsão	Realizado	Acerto	Iteração	Estado	Previsão	Realizado	Acerto
1	11→11	S	S	S	11	10→11	S	S	S
2	11→11	S	S	S	12	11	S	S	S
3	11→11	S	S	S	13	11	S	S	S
4	11→11	S	S	S	14	11	S	S	S
5	11→11	S	S	S	15	11	S	S	S
6	11→11	S	S	S	16	11	S	S	S
7	11→11	S	S	S	17	11	S	S	S
8	11→11	S	S	S	18	11	S	S	S
9	11→11	S	S	S	19	11	S	S	S
10	11-->10	S	N	N	20	11	S	N	N

1ª Iteração – 90% acerto

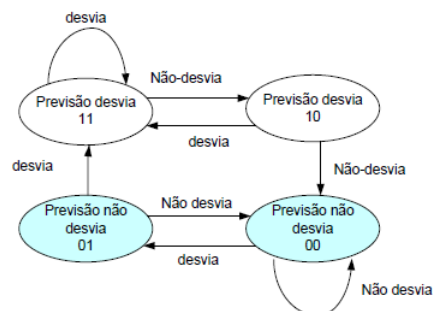
2ª Iteração – 90 % acerto

Total – 2 erros, 90% acerto

- b) comparar com o caso em que não use esse sistema de previsão, e apenas considere que a **previsão para trás** seja sempre de desvio;

Considerando taxa de acerto 90% - erra na saída

- c) comparar com o sistema de previsão de um bit



Considerando estado inicial igual a 0 – 80% acerto

Considerando estado inicial igual a 1 – 90% acerto

7ª Questão) (1,5 ponto) Considere o conjunto de instruções abaixo

					Latência
I1	div	F6	F2	F4	3
I2	sw	F2	45(R3)		2
I3	mult	F0	F6	F4	4
I4	div	F8	F2	F0	4
I5	sub	F10	F0	F6	1
I6	add	F6	F8	F6	2

a) Identifique as situações de dependência (WAW, WAR, RAW) na seguinte sequência de código acima, do MIPS64:

WAW – I1-I6 0,1

WAR – I2-I1, I6 – I5, I6 – I3, I6-I1 0,2

RAW – I3-I1, I4 – I2, I4 – I3, I5- I3, I5 – I1, I6 – I4, I6 – I1 0,7

b) 0,5 Apresente uma sequência de termino em ordem e outra em fora de ordem (que execute no menor tempo)

Termino em ordem

I1	I2		<u>I1</u>	<u>I2</u> , I3				<u>I3</u>	I4	I5			<u>I4</u>	<u>I5</u> I6		<u>I6</u>			
----	----	--	-----------	-------------------	--	--	--	-----------	----	----	--	--	-----------	-----------------	--	-----------	--	--	--

Termino fora de ordem

I1	I2		<u>I1</u> , <u>I2</u>	I3				<u>I3</u>	I4	I5	<u>I5</u>		<u>I4</u>	I6		<u>I6</u>			
----	----	--	--------------------------	----	--	--	--	-----------	----	----	-----------	--	-----------	----	--	-----------	--	--	--

8ª Questão) (1,5 Ponto) Considere o conjunto de instruções abaixo

					Latência
I1	sw	F2	45(R3)		2
I2	div	F6	F6	F4	3
I3	lw	F0	60 (F2)		4
I4	div	F8	F6	F2	5
I5	add	F6	F8	F2	1
I6	sub	F10	F0	F6	2

a) (0,5 ponto) Identifique as situações de dependência (WAW, WAR, RAW) na seguinte sequência de código acima, do

WAW – I2 – I5 0,2

WAR – I5 – I4, I5 – I2, 0,2

RAW – I4 – I2,,I4 – I1, I5 – I4, I6 – I2, I6 – I3, I6 – I5, I3 - I1, I5 – I1 0,6

b) (0,5 ponto ) Apresente uma sequência de termino em ordem e outra em fora de ordem (que execute no menor tempo)

Termino em ordem

I1	I2	<u>I1</u> I3		<u>I2</u>	I4	<u>I3</u>				<u>I4</u>	I5	<u>I5</u>	I6		<u>I6</u>				
----	----	-----------------	--	-----------	----	-----------	--	--	--	-----------	----	-----------	----	--	-----------	--	--	--	--

Termino fora de ordem

I1	I2	<u>I1</u> I3		<u>I2</u>	I4	<u>I3</u>				<u>I4</u>	I5	<u>I5</u>	I6		<u>I6</u>				
----	----	-----------------	--	-----------	----	-----------	--	--	--	-----------	----	-----------	----	--	-----------	--	--	--	--

9ª Questão) (1,5 ponto) 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))

Sequencia Instruções
lw \$11, 40 (\$6)
lw \$12, 20(\$4)
add \$12, \$3, \$11
add \$11, \$6, \$4
or \$11, \$11, \$4

a) (0,25 ponto) Quais dependências são conflitos (hazards) que podem ser resolvidos com adiantamento? Quais dependências que são conflitos e irão provocar a parada (bolhas) na execução?

	1	2	3	4	5	6	7	8	10	11	12	13	14
lw \$11, 40 (\$6)	BI	DI	EX	MEM	W								
lw \$12, 20(\$4)		BI	DI	EX	MEM	W							
add \$12, \$3, \$11			BI	DI	X	EX	MEM	W					
add \$11, \$6, \$4				BI	X	DI	EX	---	W				
or \$11, \$11, \$4						BI	DI	X	X	EX	---	W	

lw \$11, 40 (\$6)

add \$12, \$3, \$11

Este conflito pode ser resolvido com adiantamento. O adiantamento é realizado entre Rd de uma instrução a frente e RS e Rt de uma instrução anterior. (0.15 ponto)

add \$11, \$6, \$4

or \$11, \$11, \$4

Este conflito pode ser resolvido com adiantamento (0,1)

b) (0,25 ponto) Se não há adiantamento ou detecção de conflito, insira nops para assegurar a execução correta e desenhe o diagrama de execução do pipeline para este código

lw \$11, 40 (\$6)

lw \$12, 20(\$4)

NOPS 0,1

add \$12, \$3, \$11

add \$11, \$6, \$4

## NOPS

**NOPS 0,15**

**or \$11, \$11, \$4**

- c) (0,25) Repita o item anterior, mas adicione nops somente quando um conflito não pode ser evitado por mudando ou reorganizando estas instruções. Você pode assumir o registrador R7 para guardar valores temporários em seu código modificado.

**lw \$11, 40 (\$6)**

**add \$R7, \$6, \$4**

**lw \$12, 20(\$4)**

**add \$12, \$3, \$11**

**or \$11, \$R7, \$4**

- d) (0,25) 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 continuar acessando a memória para buscar a instrução.

- e) (0,25) 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 deverão ser fixados em 0 (zero) para os estágios EX, MEM e ER. Estes valores de sinais de controle são gerados no estágio DI e são passados adiante a cada ciclo de relógio, produzindo o efeito desejado (nenhum registrador ou memória é escrito)

- f) (0,25) Apresente o teste de conflito realizado no estágio EX e MEM pela unidade de adiantamento.

Conflitos no Estágio EX

se (EX/MEM.EscReg = 1

e (EX/MEM.RegistradorRd  $\square$  0 )

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

se (EX/MEM.EscReg = 1

e (EX/MEM.RegistradorRd  $\square$  0 )

e (EX/MEM.RegistradorRd = DI/EX.RegistradorRt )) Adianta.B = 10

Conflitos no Estágio MEM

se (EX/MEM.EscReg = 1

e (EX/MEM.RegistradorRd  $\square$  0 )

e (MEM/ER.RegistradorRd = DI/EX.RegistradorRs )) Adianta.A = 01

se (EX/MEM.EscReg = 1

e (EX/MEM.RegistradorRd  $\square$  0 )

e (MEM/ER.RegistradorRd = DI/EX.RegistradorRt )) Adianta.B = 01

10ª Questão) (1,5 ponto) Considere a seguinte sequência 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))

Sequencia Instruções
<b>lw \$11, 40 (\$6)</b>
<b>sw \$12, 20(\$4)</b>
<b>add \$12, \$3, \$11</b>
<b>add \$11, \$6, \$4</b>
<b>and \$11, \$11, \$4</b>

- 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?
- b) Se não há adiantamento ou detecção de conflito, insira nops para assegurar a execução correta e desenhe o diagrama de execução do pipeline para este código
- c) Repita o item anterior, mas adicione nops somente quando um conflito não pode ser evitado por mudando ou reorganizando estas instruções. Você pode assumir o registrador R7 para guardar valores temporários em seu código modificado.
- d) Um conflito estrutural (duas instruções tentando acessar a memória) pode ser resolvido pelo compilador inserindo uma instrução nops?
- 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

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

11ª. Questão) (1,5 ponto) Mostrar o resultado (**décimo ciclo**) do uso do placar (scoreboard) para a sequência de instruções, considerando-se que a instrução LD leva 1 ciclo para execução; MUL, 6 ciclos. ADD e SUB levam 3 ciclos; e DIV, 10 ciclos.

**LD F2, 34(R2)**  
**LD F6, 45(R3)**  
**MUL F0, F2, F4**  
**DIV F10, F0, F6**  
**SUB F0, F6, F2**  
**ADD F6, F8, F2**

Cada linha 0,2



## Estado da instrução

Instrução	j	k	Emite	Lê Oper	EX final	Escreve resultado
LD F2, 34(R2)			1	2	3	4
LD F6, 45(R3)			5	6	7	8
MULD F0, F2, F4			6	7		
DIVD F10, F0, F6			7			
SUBD ,F0, F6, F2						
ADDD F6, F8, F2						

1,0

## Estado das FU

Tempo restante	FU	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU Qj	FU Qk	Fj? Rj	Fk? Rk
Int										
3	Mult1	Y	MULD	F0	F2	F4			Y	Y
	Mult2									
	Ad									
	Div	Y	DIVD	F10	F0	F6	Mult1		N	Y

0,25

## Registro de estado dos resultados

	F0	F2	F4	F6	F8	F10	F12	...	F30
FU	MULD						DIVD		

0,25

12ª. Questão) (1,5 Ponto) Mostrar o resultado (décimo ciclo) do uso do placar (scoreboard) para a sequência de instruções, considerando-se que a instrução LD leva 2 ciclo para execução; MULD, 7 ciclos. ADDD e SUBD levam 3 ciclos; e DIVD, 12 ciclos.

LD F2, 34(R2)  
LD F6, 45(R3)  
MULD F0, F2, F4  
DIVD F10, F0, F6  
SUBD ,F0, F6, F2  
ADDD F6, F8, F2

## Estado da instrução

Instrução	j	k	Emite	Lê Oper	EX final	Escreve resultado
LD F2, 34(R2)			1	2	4	5
LD F6, 45(R3)			6	7	9	10
MULD F0, F2, F4			7	8		
DIVD F10, F0, F6			8			
SUBD ,F0, F6, F2						
ADDD F6, F8, F2						

## Estado das FU

Tempo restante	FU	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU Qj	FU Qk	Fj? Rj	Fk? Rk
	Int									
4	Mult1	Y	MULD	F0	F2	F4			Y	Y
	Mult2	Y	DIVD	F10	F0	F6	Mult1		N	Y
	Ad									
	Div									

## Registro de estado dos resultados

	F0	F2	F4	F6	F8	F10	F12	...	F30	
FU	MULD					DIVD				

13ª Questão) (2,0 pontos) Mostrar o resultado do uso do algoritmo de Tomasulo, com os mesmos números de ciclos para a mesma sequência de instruções da 11ª Questão.

## Estado da instrução

Instrução	j	k	Emite	EX final	Escreve resultado	Busy	Ender (A)
LD F2, 34(R2)			1	3	4		
LD F6, 45(R3)			2	4	5		
MULD F0, F2, F4			3	10			
DIVD F10, F0, F6			4				
SUBD F0, F6, F2			5	9	10		
ADDD F6, F8, F2			6	10			

LD1	N
LD2	N
LD3	N

## Estações de reserva

Tempo restante	RS	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
Add1	Y		ADD	R(F8)	M(A1)		
Add2	N						
Add3	N						
0 Mult1	Y		MULD	M(A1)	R(F4)		
Mult2	Y		DIVD		M(A2)		Mult1

## Registro de estado dos resultados

	F0	F2	F4	F6	F8	F10	F12	...	F30
FU	M(A2)+M(A1)	M(A1)			RF(8)+M(A1)				Mult2

14ª Questão) (2,0 pontos) Mostrar o resultado do uso do algoritmo de Tomasulo, com os mesmos números de ciclos e para a mesma sequência de instruções da 12ª Questão.

## Estado da instrução

Instrução	j	k	Emite	EX final	Escreve resultado	Busy	Ender (A)
LD F2, 34(R2)			1	4	5		
LD F6, 45(R3)			2	5	6		
MULD F0, F2, F4			3				
DIVD F10, F0, F6			4				
SUBD F0, F6, F2			5	9	10		
ADDD F6, F8, F2			6	10			

LD1	N
LD2	N
LD3	N

## Estações de reserva

Tempo restante	RS	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
0 Add1	Y		ADD	R(F8)	M(A1)		
Add2	N						
Add3	N						
1 Mult1	Y		MULD	M(A1)	R(F4)		
Mult2	Y		DIVD		M(A2)		Mult1

## Registro de estado dos resultados

	F0	F2	F4	F6	F8	F10	F12	...	F30
FU	M(A2)+M(A1)	M(A1)			RF(8)+M(A1)				Mult2

