Universidade de São Paulo
Escola de Artes, Ciências e Humanidades
Docente: Prof. Dr. Clodoaldo A M Lima.
Discente: _____ No. USP:_____

1ª Questão) (1,0 ponto) Usando o sistema de previsão local de desvio de 2 bits, mostrado na Figura1, um certo loop é executado duas vezes:
a) (0,5 ponto) Considerando que o estado inicial seja 00, calcule a porcentagem de acertos e erros de previsão, considerando-se que o loop termina com 10 iterações;
b) (0,5 ponto) Compare com o caso em que não use esse sistema de previsão, e apenas considere que a previsão seja sempre de desvio;

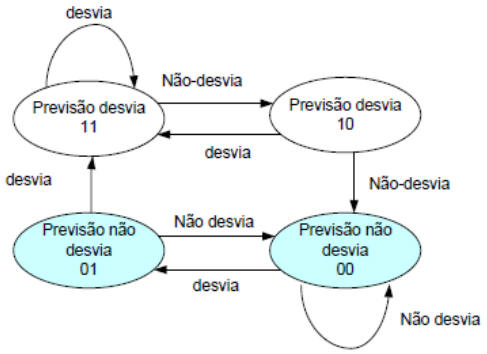

Figura 1 – Preditor de dois bits

2ª Questão) (1,0 ponto) Usando o sistema de previsão local de desvio de 1 bits, um certo loop é executado duas vezes:
a) (0,5 ponto) Considerando-se que o estado inicial seja 0 (previsão não desvia), calcular a porcentagem de acertos e erros de previsão, considerando-se que o loop termina com 10 iterações;
b) (0,5 ponto) Comparar com o caso em que não use esse sistema de previsão, e apenas considere que a previsão seja sempre de desvio;

3ª Questão) (1,0 ponto) Supondo que melhoramos uma máquina fazendo as instruções de adição serem executadas 2 vezes mais rápidas, e as instruções de multiplicação 4 vezes mais rápidas. Se o tempo de execução de certo benchmark antes do melhoramento é de 100s, sendo 20s em adição e 40s em multiplicação.
a) (0,5 ponto) Calcule o speedup
b) (0,5 ponto) Calcule o fator de melhoramento total

4ª Questão) (1,0 ponto) Suponhamos que melhoramos uma máquina fazendo todas as instruções de ponto-flutuante serem executadas 5 vezes mais rápido.
a) (0,5 ponto) Se o tempo de execução de certo benchmark antes do melhoramento é de 10s, qual seria o speedup se metade dos 10s é despendida em instruções de ponto-flutuante?
b) (0,5 ponto)Estamos procurando um benchmark para testar a nova unidade de ponto-flutuante acima, e queremos que o benchmark todo mostre um speedup de 3. Um benchmark é executado em 100s, com o antigo hardware de ponto-flutuante. Quanto do tempo de execução as instruções de ponto-flutuante devem corresponder nesse programa para que possamos produzir o speedup desejado nesse benchmark?

5ª Questão) (2,0 pontos) Considere a seguinte ideia: modificar a arquitetura do conjunto de instruções e retirar a capacidade de especificar um deslocamento para instruções de acesso à memória. Especificamente, todas as instruções *load-store* com deslocamento diferente de zero devem se tornar pseudo-instruções e devem ser implementadas por meio de duas instruções. Por exemplo:
addi $at, $t1, 104 # some o deslocamento a um registrador temporário
lw $t0, $at # nova forma de fazer lw $t0, 104($t1)

Que mudanças devem ser feitas no caminho de dados multiciclo e no controle para que essa arquitetura simplificada possa funcionar?

6ª Questão) (2,0 pontos) Considere a seguinte ideia: modificar a arquitetura do conjunto de instruções e retirar a capacidade de especificar um deslocamento para instruções de acesso à memória. Especificamente, todas as instruções *load-store* com deslocamento diferente de zero devem se tornar pseudo-instruções e devem ser implementadas por meio de duas instruções. Por exemplo:

addi $at, $t1, 104 # some o deslocamento a um registrador temporário
lw $t0, $at # nova forma de fazer lw $t0, 104($t1)

Que mudanças devem ser feitas no caminho de dados multiciclo e no controle para que essa arquitetura simplificada possa funcionar?

7ª Questão) (2,0 pontos) Assuma que o seguinte código é executado sobre um processador com pipeline de 5 estágio, com adiantamento e um preditor de desvio de 1 bit (com estado inicial em 0)

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)

```
Label1: sw $2,20($4)
        beq $2, $3, Label2 //tomado
        add $1,$6,$4
        slti $2,$s1, 100
Label2: beq $1,$2,Label1//não tomado
        and $1,$1,$4
        lw $2, 0($s1)
```

a) (1,0 ponto) Desenhe o diagrama de execução com e sem preditor para este código, assumindo que não há slots de atraso e o que desvio executa no estágio MEM

b) (1,0 ponto) Qual é o speed-up alcançado ao mover a execução de desvio para o estagio DI. Assuma que a comparação no estágio DI não afeta o tempo de ciclo de clock.

8ª Questão) (2,0 pontos) Assuma que o seguinte código é executado sobre um processador com pipeline de 5 estágio, com adiantamento e um preditor de desvio de 2 bit (com estado inicial em 01)

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)

```
Label1: sw $2,20($4)
        beq $2, $3, Label2 //tomado
        add $1,$6,$4
        slti $2,$s1, 100
Label2: beq $1,$2,Label1//não tomado
        and $1,$1,$4
        lw $2, 0($s1)
```

a) Desenhe o diagrama de execução com e sem preditor para este código, assumindo que não há slots de atraso e o que desvio executa no estágio EX

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

9ª Questão) (2,0 pontos) Considere o conjunto de instruções abaixo

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

a) (0,4 ponto) Identifique as situações de dependência (WAW, WAR, RAW) na seguinte sequência de código acima, do MIPS64:

b) (0,4 ponto) Apresente uma sequência de termino em ordem e outra em fora de ordem (que execute no menor tempo)

c) (0,4 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?

d) (0,4 ponto) 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

e) (0,4 ponto) 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.

10ª Questão) (2,0 pontos) Considere o conjunto de instruções abaixo

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

a) (0,4 ponto) Identifique as situações de dependência (WAW, WAR, RAW) na seguinte sequência de código acima, do MIPS64:

b) (0,4 ponto) Apresente uma sequência de termino em ordem e outra em fora de ordem (que execute no menor tempo)

c) (0,4 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?

d) (0,4 ponto) 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

e) (0,4 ponto) 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.

11ª. Questão) (2,0 pontos) Mostrar o resultado (**para décimo ciclo**) do uso do placar (scoreboard) 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.

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

Estado da instrução

Instrução	j	k	Emite	Lê Oper	EX final	Escreve resultado

Estado das FU

Tempo restante	FU	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU Qj	FU Qk	Fj? Rj	Fk? Rk
	Int									
	Mult1									
	Mult2									
	Ad									
	Div									

Registro de estado dos resultados

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

12ª 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 questão anterior.

Estado da instrução

Instrução	j	k	Emite	EX final	Escreve resultado

	Busy	Ender (A)
LD1	N	
LD2	N	
LD3	N	

Estações de reserva

Tempo restante	RS	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
	Add1	N					
	Add2	N					
	Add3	N					
	Mult1	N					
	Mult2	N					

Registro de estado dos resultados

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