

## SSC611 – Arquitetura de Computadores

### Lista de Exercícios sobre Pipeline

1. Os 5 estágios de um processador têm as seguintes latências:

	Fetch	Decode	Execute	Memory	Write back
<b>a</b>	300 ps	400 ps	250 ps	500 ps	100 ps
<b>b</b>	200 ps	150 ps	120 ps	190 ps	140 ps

Assuma que, na versão pipeline, cada estágio consome outros 20 ps com os registradores entre os estágios.

- Na versão não pipeline, qual o tempo de ciclo? Qual a latência de uma instrução? Qual o *throughput*?
  - Na versão pipeline, qual o tempo de ciclo? Qual a latência de uma instrução? Qual o *throughput*?
  - Se um dos ciclos pudesse ser dividido ao meio, qual você escolheria? Qual o novo tempo de ciclo? Qual a nova latência? Qual o novo *throughput*?
2. Considere o seguinte trecho de código

```

I0. lw $s2, 0($s1)
I1. lw $s1, 40($s6)
I2. sub $s6, $s1, $s2
I3. add $s6, $s2, $s2
I4. or $s3, $s6, $zero
I5. sw $s6, 50($s1)

```

- Quais são as dependências de dados? E os *hazards*?<sup>1</sup>
- Assuma um pipeline MIPS de 5 estágios sem *forwarding*, e que cada estágio demora 1 ciclo. Ao invés de inserir operações *nops*, você deixar o processador parar quando se tem *hazards*. Quantos ciclos o processador para? Qual o tamanho de cada parada, em ciclos? Qual o tempo de execução (em ciclos) do programa inteiro?
- Assuma um pipeline MIPS de 5 estágio com *forwarding* total. Escreva o programa com *nops* para eliminar os *hazards*.

<sup>1</sup> Dependência é quando o dado de uma instrução depende de outra instrução.

Hazard é uma situação na qual a execução do pipeline é impedida de continuar, causando uma parada.

3. Considere o seguinte trecho de código em linguagem assembly MIPS:

```

I0: add $t4, $t0, $t1
I1: sub $t7, $t3, $t4
I2: add $t4, $t5, $t6
I3: lw $t2, 100($t3)
I4: lw $t2, 0($t2)
I5: sw $t4, 100($t2)
I6: and $t2, $t2, $t1
I7: beq $$t7, $t1, target
I8: and $t7, $t7, $t1
    
```

Considere um pipeline com *forwarding*, detecção de dependências, e 1 *delay slot* para desvios. O pipeline tem os 5 estágios estudados em aula: IF, ID, EX, MEM e WB. Para o código acima, complete o diagrama de pipeline abaixo (instruções e ciclos). Preencha com os símbolos IF, ID, EX, MEM e WB para cada instrução da linha. Assuma que os dois níveis de *bypass* são implementados (MEM->EX e WB->EX), que a segunda metade do estágio de decodificação lê os registradores de origem e a primeira metade do estágio de escrita no banco de registradores escreve no banco de registradores.

Indique as paradas por dependência de dados (desenhe um X na linha/coluna correspondente). Indique todas as dependências detectadas pela unidade de *forwarding* através de setas que saem do estágio de origem do dado e chegam no estágio de destino do dado.

Qual o número de clocks utilizados para executar esse código?

	CC0	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9	CC10	CC11	CC12	CC13	CC14	CC15	CC16
I1																	
I2																	
I3																	
I4																	
I5																	
I6																	
I7																	
I8																	

4. Dependências estruturais, de dados e de controle normalmente fazem com o que o pipeline pare. Abaixo está especificada uma lista de técnicas de otimização implementadas por compiladores ou em no hardware do processador pipeline de modo a reduzir ou eliminar as paradas devido às dependências. Para cada uma das técnicas de otimização, indique qual dependência é tratada e como isso é feito. Algumas otimizações podem atender mais do que uma dependência, portanto, fique atento e explique todas as dependências tratadas por cada uma das otimizações.

- a) Previsão de desvios
- b) Escalonamento de instruções
- c) Delay Slots
- d) Aumentar a disponibilidade das unidades funcionais (ULAs, somadores, etc.)
- e) *Forwarding*

5. Considere a seguinte sequência de saídas para um desvio único. T significa que o desvio foi tomado e N significa que o desvio não foi tomado. Assuma que esse é o único desvio do programa.

TTTNTNTTTNTNTTTNTN

- a) Assuma que tentamos prever a sequência utilizando uma tabela de história dos desvios com 1 bit. Os contadores na tabela são iniciados com o valor N. Quais desvios da sequência serão previstos de maneira errada? Utilize a seguinte tabela para ajudar

Estado do previsor antes da previsão	Decisão da instrução de desvio	Previsto errado?
T	T	
	T	
	T	
	N	
	T	
	N	
	T	
	T	
	N	
	T	
	N	
	T	
	T	
	T	
	N	
	T	
	N	

- b) Agora assumo um previsor de desvio com dois bits. Quais desvios da sequência serão previstos de maneira errada? Utilize o diagrama de estado estudado em sala de aula. O estado inicial

Estado do previsor antes da previsão	Decisão da instrução de desvio	Previsto errado?
T (1º)	T	
	T	
	T	
	N	
	T	
	N	
	T	
	T	
	T	
	N	
	T	
	N	
	T	
	T	
	T	
	N	
	T	
	N	

6. Considere um processador MIPS com um pipeline de 5 estágios e tempo de ciclo de 10ns. Assuma que você está executando um programa com uma fração,  $f$ , onde todas as instruções imediatamente após um load dependem do resultado da instrução load.
- Com o *forwarding* habilitado, qual o tempo total de execução de  $N$  instruções, em termos de  $f$ ?
  - Considere agora um cenário onde o estágio MEM e os registradores do pipeline do estágio precisam de 12ns. Há duas possibilidades: dividir o estágio em dois (MEM1 e MEM2) ou aumentar o tempo de ciclo para 12ns, definindo um novo tempo de ciclo mas mantendo a mesma quantidade de estágio. Para um programa que possui as características acima, quando a primeira opção é melhor do que a segunda? Sua resposta deve se basear no valor de  $f$ .