

UNIVERSIDADE FEDERAL DE RORAIMA
PROFESSOR (A) : Herbert Oliveira Rocha .
DISCIPLINA DE ARQUITECTURA DE COMPUTACAO.
ALUNO : Ibukun Chife didier Adjitche

RESOLUÇÃO DA LISTA N°2

Questão1: Quais as vantagens de um processador multiciclo em relação a um uniciclo?

R1: Permitir que instruções usem diferentes quantidade de ciclos de clock e compartilhe unidades funcionais durante a sua execução.

Questão2: Quais as modificações necessárias em um processador multiciclo simples para que se introduza a função de pipeline?

R2: Para introduzir o Pipeline num processador multiciclo que já possuir os Registradores (IR, MDR, A, B, UALSaída) e Multiplexadores (End. Memória, UAL). Temos a possibilidade adicionar novos registradores que terão como função de segurar os bits de Controle propagados Juntos ao dados a cada estações:

- Busca de Instruções e incremento do PC (IF)
- Decodificação da instrução e busca dos registradores (ID)
- Execução e cálculo do Endereço (EX)
- Acesso à Memória
- Gravação do resultado no banco de registradores (WB)

Questão3: Considerando o pipeline do MIPS (simples com MEM compartilhada para instrução e dados) e uma iteração de loop conforme o trecho de programa abaixo, relacione os conflitos que podem ocorrer e seus consequentes stalls. Qual o speedup (por iteração) para o programa em relação à versão sem pipeline?

```
Loop: subi $t2, $t2, 4
      lw $t1, 0($t2)
      add $t3, $t1, $t4
      add $t4, $t3, $t3
      sw $t4, 0($t2)
      beq $t2, $0, loop
```

R3:

Execução com Pipeline: Confere o Quadro 1

Calculo do speedup sem Pipeline:

□ tempo da execução da instrução Sub é : **6ns**

□ tempo da execução da instrução Load é: **8ns**

□ tempo da execução da instrução Add é: **6ns**

□ tempo da execução da instrução Store é: **7ns**

□ tempo da execução da instrução Beq é : **5ns**

$$\begin{aligned} \text{Texec (Sem Pipeline)} &= T_{\text{Sub}} + T_{\text{load}} + 2 * T_{\text{add}} + T_{\text{Store}} + T_{\text{Beq}} \\ &= 6\text{ns} + 8\text{ns} + 12\text{ns} + 7\text{ns} + 5\text{ns} \end{aligned}$$

Texec (Sem Pipeline) = 38ns

Texec (Com Pipeline) = 30ns

Speedup= Texec(Sem Pipeline)/Texec(Com Pipeline) = 38/30= 1.3 vezes

UNIVERSIDADE FEDERAL DE RORAIMA
PROFESSOR (A) : Herbert Oliveira Rocha .
DISCIPLINA DE ARQUITECTURA DE COMPUTACAO.
ALUNO : Ibukun Chife didier Adjitche

Questão4: Na questão anterior, assuma que a memória de instruções e dados podem ser segmentadas e que o processador aplica a técnica de bypassing. Como ficarão os conflitos e seus stalls, e o speedup?

R4: Confere o Quadro2

Com a tecnica de Bypassing o Tempo de execucao ficou em **1900 ps seja 19 ns**

Speedup= Texec(Sem Pipeline)/Texec(Com Pipeline bypassing) = 38/19= 2 vezes

Não houve de Stall durante a execução das instruções.

Questão5: No programa abaixo, relacione as dependências (dados, WAR, WAW e outros) existentes.

div.d F1, F2, F3

sub.d F4, F5, F1

s.d F4, 4(F10)

add.d F5, F6, F7

div.d F4, F5, F6

R5:

- Existem 3 dependências de dados REAIS

- DIV.D e SUB.D
- SUB.D e S.D
- ADD.D e DIV.D

- Existe uma antidependência entre o ADD.D e o SUB.D

- ADD pode escrever em reg. que SUB lê.
- WAR: Uso de F5 por SUB.D

- Existe uma dependência de saída entre o SUB.D e o DIV.D

- SUB.D pode terminar depois do DIV.D
- WAW: Uso de F4

Questão6: Apresente 5 processadores diferentes que trabalham paralelismo. Adicionalmente, descreva o funcionamento e o tipo de paralelismo de cada um dos processadores, bem como suas vantagens e desvantagens.

R6:

AMD64:

Quando a arquitetura AMD64 foi lançada junto com o Athlon 64, ela trouxe um modo de operação de 64 bits para as instruções x86. Este modo é chamado de x86-64 pela AMD e o que ele faz é expandir os registradores de 32 bits existentes em registradores de 64 bits. Todos os processadores AMD64 têm dezesseis registradores de propósito geral de 64 bits quando operando no modo x86-64. Neste modo o barramento de endereço do processador também é expandido de 32 para 40 bits, habilitando o processador para acessar diretamente até 1 TB de memória RAM (2^{40}). Além disso,

UNIVERSIDADE FEDERAL DE RORAIMA
PROFESSOR (A) : Herbert Oliveira Rocha .
DISCIPLINA DE ARQUITECTURA DE COMPUTACAO.
ALUNO : Ibukun Chife didier Adjitche

neste modo de operação o processador pode acessar até 256 TB de memória virtual (2^{48}). A memória virtual é uma técnica que permite ao processador simular mais memória RAM do que realmente existe instalada no micro, criando um arquivo no disco rígido chamado arquivo de troca. A arquitetura AMD64 utiliza um pipeline de 12 estágios para executar instruções inteiras e 17 estágios para executar instruções de ponto flutuante.

Pentium 4 :

O Pentium IV é a sétima geração de microprocessadores com arquitetura x86 fabricados pela Intel, é o primeiro CPU totalmente redesenhado desde o Pentium Pro de 1995. Uma das características da micro arquitetura NetBurst era seu pipeline longo, desenhado com a intenção de permitir frequências elevadas. Também foi introduzido a instrução SSE2 com um integrador SIMD mais rápido, e cálculo de pontos flutuantes em 64-bit. O Pentium 4 original, com o nome de código "Willamette", foi introduzido em novembro de 2000 para o Socket423, sendo lançados em versões 1.3 a 2.0 GHz. Para surpresa da maioria dos observadores da indústria, o Pentium 4 não melhorou em relação ao velho projecto do P3 em qualquer uma das duas medidas chave de desempenho normal: velocidade de processamento de inteiros ou no desempenho de pontos flutuantes: pelo contrário, sacrificou o desempenho por-ciclo a fim de ganhar duas coisas: velocidades de clock muito elevados e desempenho de SSE. Como é tradicional na Intel, o P4 vem também em uma versão Celeron de gama baixa (freqüentemente referida como Celeron 4) e uma versão topo de gama Xeon recomendada para configurações de SMP. O Pentium 4 executa muito menos trabalho por ciclo do que outros microprocessadores (tais como o Athlon ou o velho Pentium III), mas o objetivo do projeto original foi cumprido - sacrificando as instruções por ciclo de pulsos de disparo (clock) a fim de conseguir um número maior de ciclos por segundo (isto é, uma frequência maior ou velocidade de clock).

Questão7: Em relação a memória cache. Um computador tem CPI 1 quando todos os acessos à memória acertam no cache. Loads e Stores totalizam 50% das instruções. Se a penalidade por miss é de 25 ciclos e o miss rate é 2%, qual o desempenho relativo se o computador acertar todos os acessos?

R7:

$$\begin{aligned}\text{CPU execution time} &= (\text{CPU clock cycles} + \text{Memory stall cycles}) * \text{Clock cycle} \\ &= (\text{IC} * \text{CPI} + 0) * \text{Clock cycle} \\ &= \text{IC} * 1 * \text{Clock cycle}\end{aligned}$$

$$\begin{aligned}\text{Memory stall cycles} &= \text{IC} * (\text{Memory Accesses/Instruction}) * \text{Miss rate} * \text{Miss penalty} \\ &= \text{IC} * (1 + 0.5) * 0.02 * 25 \\ &= \text{IC} * 0.75\end{aligned}$$

$$\begin{aligned}\text{CPU execution time (cache)} &= (\text{IC} * 1.0 + \text{IC} * 0.75) * \text{Clock cycle} \\ &= 1.75 * \text{IC} * \text{Clock cycle}\end{aligned}$$

$\frac{\text{CPU execution time (cache)}}{\text{CPU execution time}} = \frac{1.75 * \text{IC} * \text{Clock cycle}}{1 * \text{IC} * \text{Clock cycle}} = 1.75$

UNIVERSIDADE FEDERAL DE RORAIMA
PROFESSOR (A) : Herbert Oliveira Rocha .
DISCIPLINA DE ARQUITECTURA DE COMPUTACAO.
ALUNO : Ibukun Chife didier Adjitche

Questão8: Descreva os seguintes conceitos:

R8:

a) Write through :

Escrita ao mesmo tempo no cache e na memória

b) Write back:

Escrita somente no cache e atualização na memória somente na substituição do bloco.

c) Localidade Temporal:

Se um item é referenciado, ele tende a ser referenciado novamente dentro de um espaço de tempo curto.

d) Localidade Espacial:

Se um item é referenciado, itens cujos endereços sejam próximos dele tendem a ser logo referenciados.

UNIVERSIDADE FEDERAL DE RORAIMA
 PROFESSOR (A) : Herbert Oliveira Rocha .
 DISCIPLINA DE ARQUITECTURA DE COMPUTACAO.
 ALUNO : Ibukun Chife didier Adjitche

Linha de Tempo(ps)	200	400	600	800	1000	1200	1400	1600	1800	2000	2200	2400	2600	2800	3000	3200
subi \$t2, \$t2, 4	IM	REG	ALU	DM	REG											
		✕	✕	✕	✕	✕										
lw \$t1, 0(\$t2)			IM	REG	ALU	DM	REG									
				✕	✕	✕	✕	✕								
					✕	✕	✕	✕	✕							
add \$t3, \$t1, \$t4						IM	REG	ALU	DM	REG						
							✕	✕	✕	✕	✕					
add \$t4, \$t3, \$t3								IM	REG	ALU	DM	REG				
									✕	✕	✕	✕	✕			
sw \$t4, 0(\$t2)										IM	REG	ALU	DM	REG		
											✕	✕	✕	✕	✕	
beq \$t2, \$0, loop												IM	REG	ALU	DM	REG

Quadro1: Quadro explicativa da Resolution da questao 3 [Texec(Pipeline)= 30 ns]

Linha de Tempo (ps)	200	400	600	800	1000	1200	1400	1600	1800	2000	2200	2400	2600	2800	3000
subi \$t2, \$t2, 4	IM	REG	ALU	DM	REG										
lw \$t1, 0(\$t2)		IM	REG	ALU	DM	REG									
add \$t3, \$t1, \$t4			IM	REG	ALU	DM	REG								
add \$t4, \$t3, \$t3				IM	REG	ALU	DM	REG							
sw \$t4, 0(\$t2)					IM	REG	ALU	DM	REG						
beq \$t2, \$0, loop						IM	REG	ALU	DM	REG					

Quadro2: Quadro explicativa da Resolution da questao 4