Pipeline

Arquitetura de Computadores

Charles Tim Batista Garrocho

Instituto Federal do Paraná – IFPR Campus Goioerê

charles.garrocho.com/AC2016

charles.garrocho@ifpr.edu.br

Técnico em Informática



Visão Geral

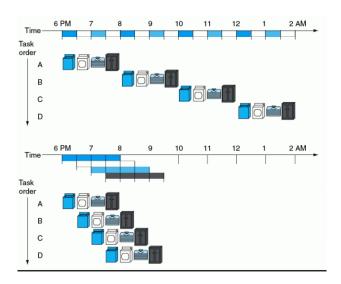
Pipelining é uma técnica de implementação em que várias instruções são sobrepostas na execução.

Etapas para se Lavar roupas:

- Colocar a trouxa suja de roupas na lavadora.
- Quando a lavadora terminar, colocar a trouxa molhada na secadora (se houver)
- Quando a secadora terminar, colocar a trouxa seca na mesa e passar
- Quando terminar de passar, pedir ao seu colega de quarto para guardar as roupas

Será que faz sentido esperar seu colega terminar, para só então começar a lavar a próxima trouxa de roupas?

Sem e Com Pipeline





Etapas do MIPS

As instruções MIPS normalmente exigem 5 etapas:

- Buscar instrução da memória
- Ler registradores enquanto a instrução é decodificada
- Executar a operação ou calcular o endereço
- Acessar um operando na memória de dados
- Escrever o resultado em um registrador

Logo, vamos implementar um caminho de dados multiciclo em pipeline.

Para simplificar a implementação, vamos concentrar a atenção em apenas 8 instruções: lw, sw, add, sub, and, or, slt e beq.



Pensando em termos de instruções MIPS

Considere os tempos de resposta de cada um dos componentes da via de dados e os tempos para execução das instruções conforme a tabela abaixo.

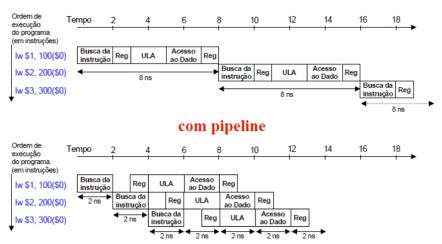
Podemos estimar que o período de clock será igual a 8ns para uma CPU MIPS na sua versão monociclo.

Classe da instrução	Busca da instrução	Leitura de registrador	Operação na ULA	Acesso ao dado	Escrita no registrador	Tempo total
Load word (1d)	2 ns	1 ns	2 ns	2 ns	1 ns	8 ns
Store word (sw)	2 ns	1 ns	2 ns	2 ns		7 ns
Formato R (add, sub, and, or, slt)	2 ns	1 ns	2 ns		1 ns	6 ns
Branch (beq)	2 ns	1 ns	2 ns			5 ns

Pensando na execução de 3 loads (instrução mais lenta)

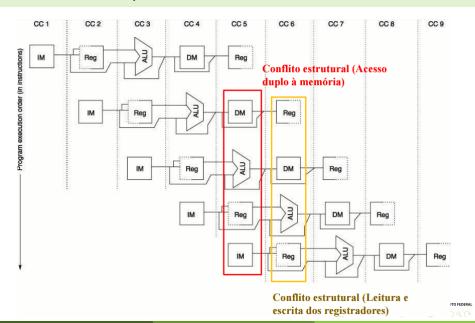


Com e Sem Pipeline





Problemas do Pipeline



Conflitos de Dados (Data Hazards)

Ocorre quando uma instrução depende de dados resultantes da execução de outra que ainda está no pipeline.

Neste caso não há uma solução.



Solução por reordenação de código

O compilador reordena as instruções de forma a evitar os possíveis conflitos de dados. Ex: o código do procedimento swap abaixo exibe um conflito.

```
# reg $t1 possui o endereco de v[k] lw $t0, 0($t1) # reg $t0 (temp) = v[k] lw $t2, 4($t1) # reg $t2 = v[k+1] sw $t2, 0($t1) # v[k] = reg $t2 sw $t0, 4($t1) # v[k+1] = reg $t0 (temp)
```



Solução por reordenação de código

Solução: trocar a 2ª e a 3ª linhas de posição: Código sem conflito

```
# reg $t1 possui o endereco de v[k] lw $t0, 0($t1) # reg $t0 (temp) = v[k] lw $t2, 4($t1) # reg $t2 = v[k+1] sw $t0, 4($t1) # v[k+1] = reg $t0 (temp) sw $t2, 0($t1) # v[k] = reg $t2
```



Exercícios

Resolva as atividades referentes a MIPS e pipelines no site da disciplina.

