

Pipeline

Arquitetura de Computadores

Charles Tim Batista Garrocho

Instituto Federal do Paraná – IFPR
Campus Goioerê

charles.garrocho.com/AC2016

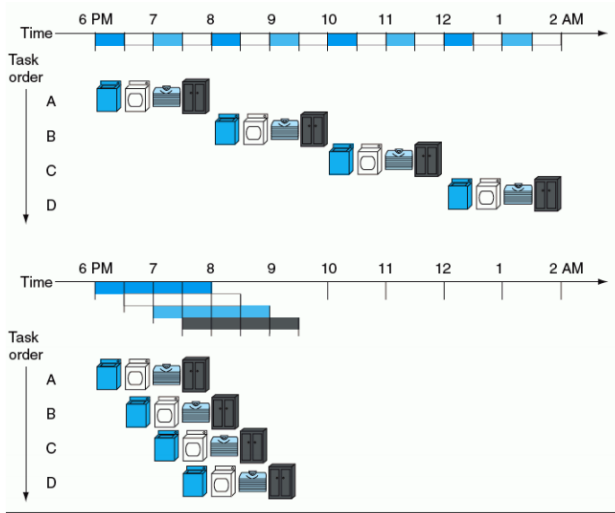
charles.garrocho@ifpr.edu.br

Técnico em Informática



INSTITUTO FEDERAL

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

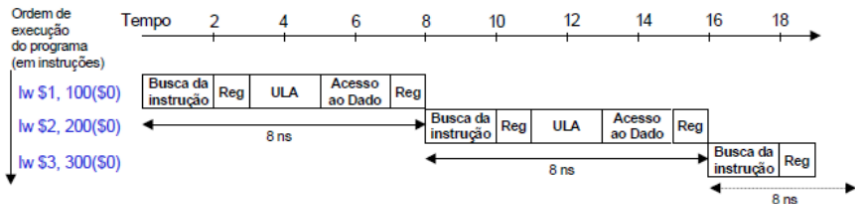
Logo, vamos implementar um caminho de dados multiciclo em pipeline.

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

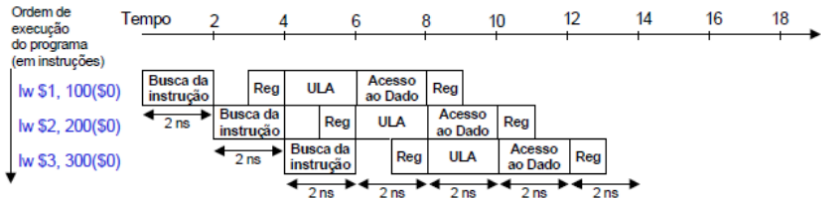


INSTITUTO FEDERAL

Com e Sem Pipeline

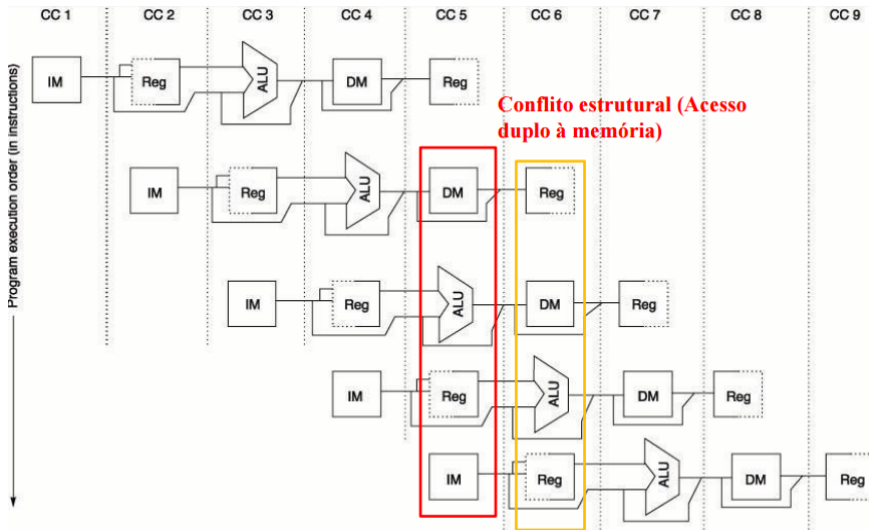


com pipeline



INSTITUTO FEDERAL

Problemas do Pipeline



Conflito estrutural (Leitura e escrita dos registradores)

Conflitos de Dados (Data Hazards)

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

```
add    $s0, $t0, $t1  
sub    $t2, $s0, $t3
```

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 exhibe um conflito.

```
                                # reg $t1 possui o endereço 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 endereço 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
```



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

