MELHORA DA PERFORMANCE USANDO PIPELINE

Visão Geral

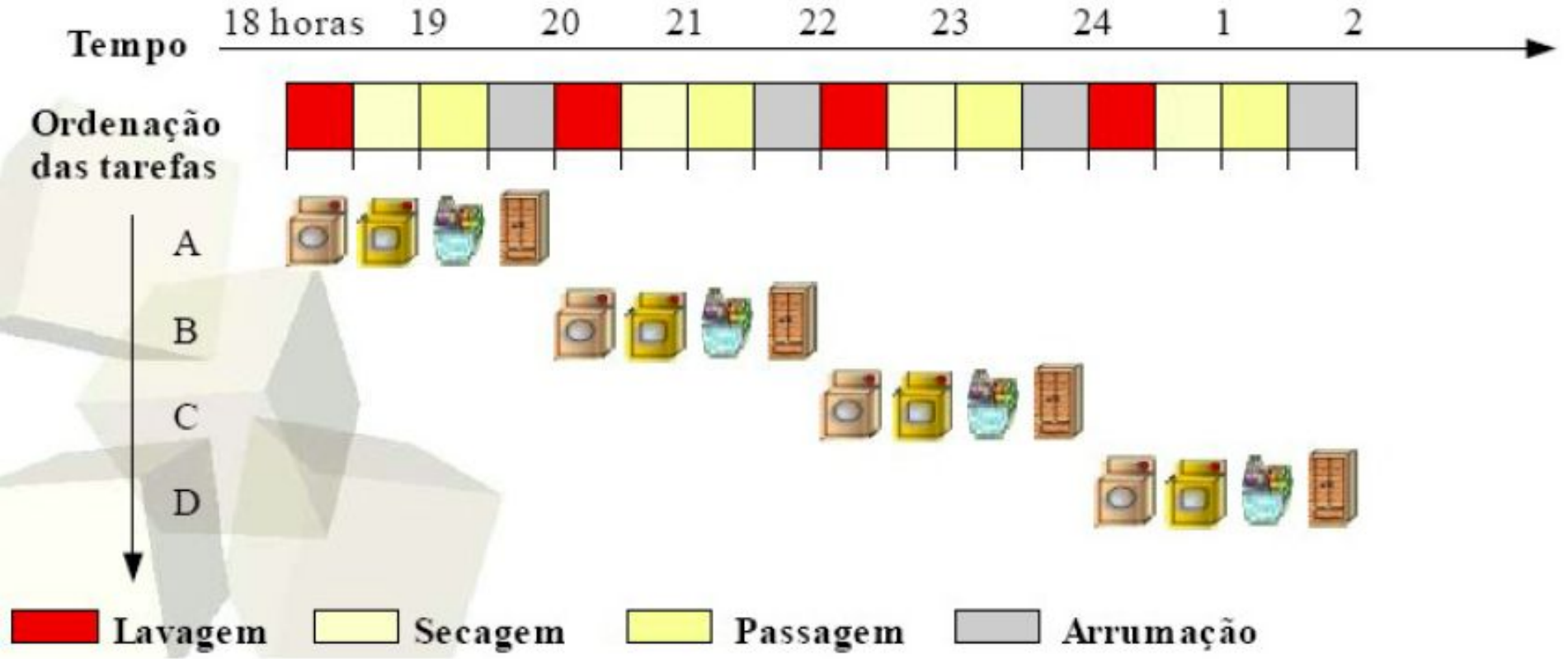
- *É uma técnica de implementação de processadores que permite a sobreposição temporal das diversas fases de execução de instruções.*

- *Atualmente a técnica de pipeline é a base para fazer com que os processadores rodem mais rapidamente.*

- *O princípio do pipeline pode ser entendido com a ajuda de uma analogia com o processo utilizado em uma lavanderia fictícia, onde se deve lavar, secar, passar e guardar as roupas. Sem o uso do pipeline, lava-se roupa da seguinte maneira:*

  - *1 – coloca-se uma carga de roupa na lavadora.*
  - *2 – quando a lavadora terminar, coloca-se as roupas na secadora.*
  - *3 – as roupas secas devem ser passadas.*
  - *4 – guarda-se as roupas nos locais adequados.*

  - *Finalizado o processo, tudo é reiniciado com uma nova carga de roupas sujas.*



- Processo, sem aplicação de pipeline.

- *Se introduzirmos o conceito de pipeline no processo de lavagem de roupa, o tempo gasto será muito menor.*

- *Tão logo a primeira carga de roupa seja transferida da lavadora para a secadora, uma nova carga de roupas sujas é colocada na lavadora.*

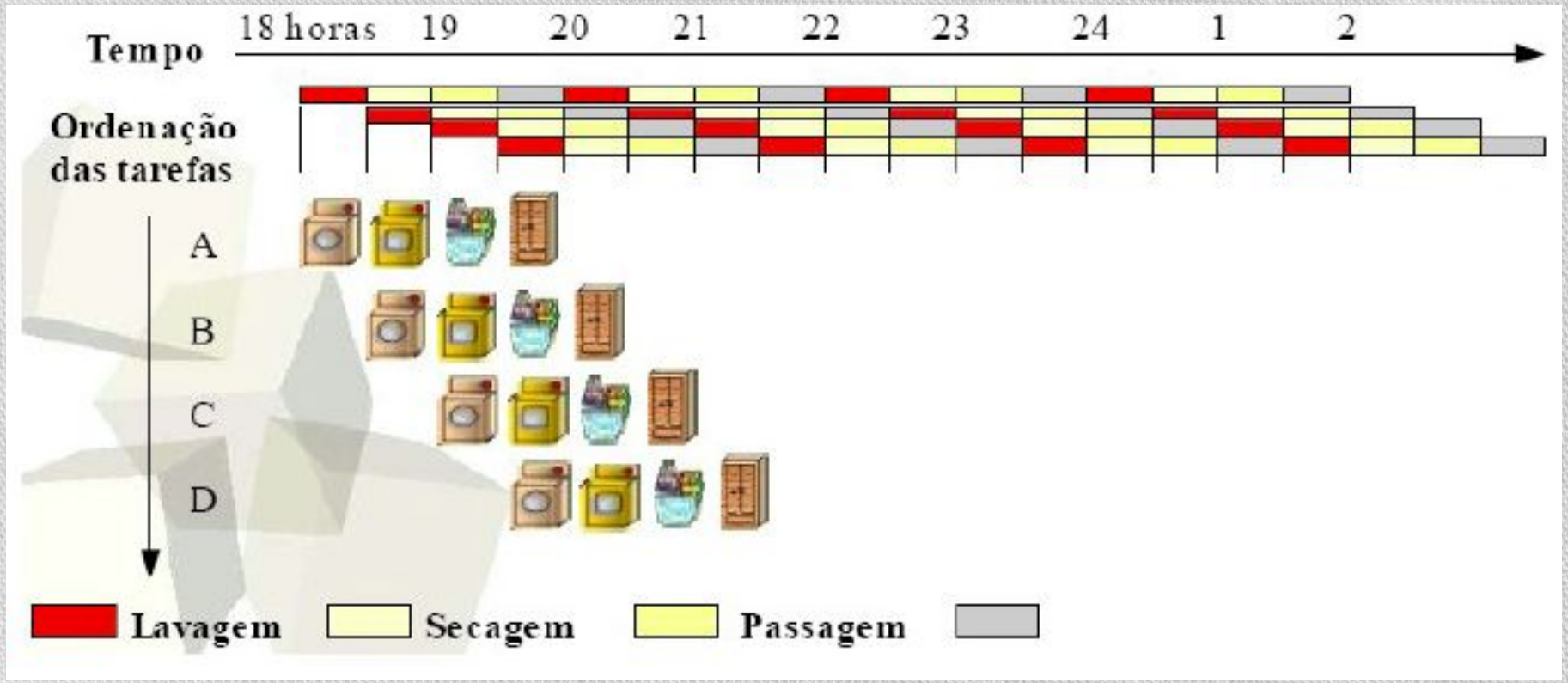
- *Quando a primeira carga estiver seca, coloque-a para passar, enquanto a segunda carga é colocada para secar e uma terceira é colocada para lavar e assim por diante.*

- *Quando a primeira carga estiver sendo guardada, a segunda estará sendo passada, a terceira estará na secadora e uma quarta estará na lavadora.*

- *A essa altura todos os passos do processo de lavagem (estágios de pipeline) estarão operando de maneira concorrente.*

- *Sempre que tivermos recursos separados para executar cada estágio, a tarefa como um todo pode ser executada em pipeline.*

PROCESSO COM APLICAÇÃO DE PIPELINE



- *O paradoxo da técnica de pipeline é: o tempo decorrido entre o momento em que uma camisa entra na lavadora até o momento em que é guardada não diminui com o pipeline.*

- *A explicação para o processo de lavagem de roupas em pipeline ser mais rápido é o fato de podermos colocar todos os recursos envolvidos operando em paralelo.*

- *Se todos os estágios da tarefa gastarem o mesmo tempo em sua execução, a aceleração devida ao pipeline será igual ao número de estágios do pipeline.*

- *O processo de lavagem de roupas no pipeline seria então quatro vezes mais rápido que o processo não pipeline.*

- *Os mesmos princípios podem ser aplicados aos processadores que executam suas tarefas em pipeline. Tradicionalmente, as instruções do MIPS são executadas em até cinco passos:*

  - *1 – busca da instrução na memória.*
  - *2 – leitura dos registradores enquanto a instrução é decodificada.*
  - *3 – execução de uma operação ou cálculo de endereço.*
  - *4 – acesso a um operando na memória.*
  - *5 – escrita do resultado no registrador.*

PERFORMANCE MONOCICLO *VERSUS* PIPELINE

- *O tempo médio de execução para a implementação monociclo, em que uma instrução gasta um ciclo de clock para ser executada, tem o tempo de instrução mínimo limitado pela instrução mais lenta.*

- *Tomando como exemplo um MIPS cujos tempos de operação são apresentados na tabela abaixo, pode-se comparar as duas implementações.*

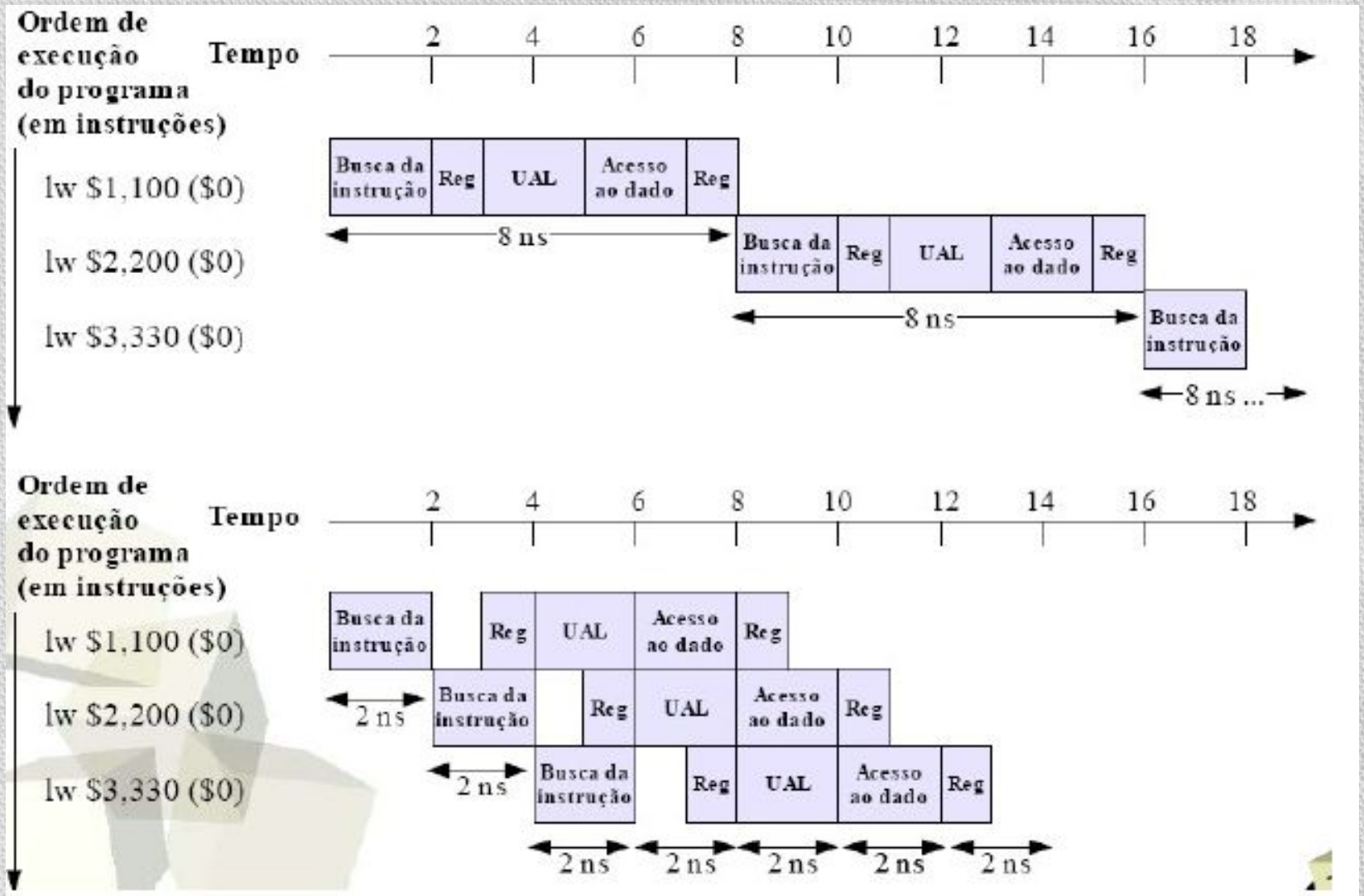
Tipo de Operação	*Tempo (ns)*
Acesso memória	*2*
Operação ULA	*2*
Leitura/escrita de registradores	*1*

- *Abaixo mostra-se a execução das instruções mais comuns:*

Classe Instrução	*Busca Instrução*	*Leitura de registradores*	*Operação ULA*	*Acesso dados*	*Escrita registradores*	*Tempo total*
load word(lw)	*2ns*	*1ns*	*2ns*	*2ns*	*1ns*	*8ns*
store word(Sw)	*2ns*	*1ns*	*2ns*	*2ns*		*7ns*
R-Format (add,sub,and,or)	*2ns*	*1ns*	*2ns*		*1ns*	*6ns*

- *Através da tabela apresentada, pode-se concluir que no projeto monociclo todas as instruções levam 8ns para serem executadas. Isto nos leva a conclusão que a execução de três instruções load word consecutivas gasta 24ns.*

Execução monociclo, não pipeline, em oposição a execução pipeline



- *O diagrama de tempo do slide anterior mostra o comportamento das unidades funcionais durante a execução das três instruções load word.*

- *No projeto pipeline, todos os estágios gastam um único ciclo, sendo necessário que o ciclo seja grande o suficiente para acomodar a operação mais lenta, no caso 2ns.*

- *Como se pode observar, ocorre uma redução de 10ns na execução de três instruções.*

- *Em condições ideais o ganho devido ao pipeline é igual ao número de estágios pipeline.*

- *Tal conceito aplicado ao exemplo mostrado aponta para um ganho cinco. No entanto, o exemplo mostra que a técnica apresenta limitações, principalmente quanto a encher todo o pipeline.*

- *Dessa forma, o tempo de execução de uma instrução em uma máquina pipeline deverá exceder o valor teórico. No caso do exemplo mostrado o ganho será quatro.*

- *Além do mais, mesmo a pretensão de um ganho quatro não se reflete no tempo de execução de três instruções, onde se registra a mudança de 24ns para 14ns.*

- *Em suma, o pipeline melhora a performance por meio do aumento do throughput das instruções, aumentando o número de instruções executadas por unidade de tempo e não por meio da diminuição do tempo de execução da instrução individualmente.*

Projeto Do Conjunto De Instruções Para Execução (Mips)

- *Vamos analisar o conjunto de instruções do MIPS que foi projetado com base na execução pipeline.*

- *Em primeiro lugar, todas as instruções do MIPS têm o mesmo tamanho, o que simplifica a busca das instruções no primeiro estágio pipeline, e a sua decodificação no segundo estágio.*

- *No caso de instruções de tamanho variável (80x86) a implementação do pipeline é um desafio.*

- *Em segundo lugar, o MIPS tem poucos formatos de instrução, sempre com o registrador fonte na mesma posição.*
  - *Esta simetria permite que o segundo estágio possa começar lendo o banco de registradores ao mesmo tempo em que o hardware determina o tipo de instrução que veio da memória.*

- *Em terceiro lugar, na arquitetura MIPS só as instruções load word e store word manipulam aperandos na memória.*
  - *Esta restrição faz com que possamos usar o estágio de execução para efetuar o cálculo do endereço de memória e, no estagio seguinte, acessar a memória.*

Principais Tipos

- *Diversas são as estruturas atualmente encontradas nos processadores e que se encaixam no perfil de pipeline. Dessa forma, algumas das suas principais classificações são apresentadas.*

- *SUPERPIPELINING*

  - *É o tipo de pipeline baseado na idéia de utilizar um grande número de estágios para maximizar a eficiência da técnica (pelo menos 6 estágios).*

  - *Apresenta como maiores vantagens: o maior número de instruções sendo processadas ao mesmo tempo e maior frequência de Clock.*

    - *Por outro lado apresenta alta complexidade, grandes dependências e desvios.*

- *PIPELINE SUPERESCALAR*
  - *Consiste na utilização de pipelines em paralelo. Dessa forma, pode-se ter 2 ou 3 pipelines dentro de um mesmo processador.*
  - *A principal vantagem reside no fato de que o paralelismo real, com 2 ou mais instruções sendo processadas em paralelo, melhora significativa de performance.*
  - *Entretanto, o código precisa ser preparado especialmente para a aplicação, além de grande complexidade de elaboração e problemas de dependências e desvios.*

- *PIPELINE DINÂMICO*
  - *Consiste em várias unidades de processamento em paralelo.*
  - *O processador executa instruções de tipos diferentes em paralelo e em ordem invertida.*
  - *Neste caso o processador mantém um paralelo especializado em instruções de Ponto Fixo (Soma e Subtração), outro paralelo para Ponto Flutuante (senos e cosenos) e, às vezes, um terceiro paralelo para Multimídia (Som, Vídeo).*
  - *Sua principal vantagem é que instruções mais lentas não atrasam o processamento das instruções seguintes. Por outro lado, apresenta como desvantagens a complexidade e dificuldade de testar o processador.*

- *PIPELINE OUT-OF-ORDER*
  - *A Execução Fora-De-Ordem faz o processador reordenar as instruções para melhor aproveitar o pipeline e evitar a inserção de bolhas.*
  - *Ele necessita uma unidade de reordenação de instruções após executá-las. Pode existir em todos os tipos de pipeline mas é obrigatório em um pipeline dinâmico.*
  - *Sua performance é otimizada e apresenta menos bolhas, apesar da alta Complexidade.*
- *Esses são os pipelines mais utilizados e a maioria dos processadores utilizam várias destas técnicas ao mesmo tempo.*

O Que Afeta O Desempenho Do Pipeline

- Existem situações no pipeline em que a instrução seguinte não pode ser executada no próximo ciclo de clock.
  - Tais eventos são conhecidos como conflitos e são classificados como:
    - Conflitos Estruturais: significa que o hardware não pode suportar a combinação de instruções que o pipeline deseja executar no mesmo ciclo de clock.
    - Isto poderia acontecer com o MIPS caso ele não tivesse duas memórias (instrução e dados) e se executasse uma quarta instrução load word no exemplo anteriormente mostrado.
    - A primeira instrução estaria acessando um dado na memória e a quarta estaria sendo buscada na memória, no mesmo ciclo de clock. Sem duas memórias o pipeline estaria sujeito a conflitos estruturais.

- *Conflitos de Controle:*
  - *é originário da necessidade de se tomar uma decisão com base nos resultados de uma instrução, enquanto outras estão sendo executadas.*
  - *Este tipo de conflito abrange instruções de desvio condicional.*
  - *Se o computador tiver que parar quando da instrução de desvio condicional, ele deve interromper a progressão das instruções pelo pipeline. Tal parada gera um atraso na execução do pipeline conhecida como bolha.*

- *Conflitos por Dados: este conflito ocorre quando a execução de uma instrução depende do resultado de outra instrução que ainda está no pipeline. Suponha o caso:*
  - *add $s0, $t0, $t1*
  - *sub $t2, $s0, $t3*
  - *Neste caso, se não houver uma intervenção, o conflito por dados vai fazer com que o progresso das instruções por meio do pipeline seja interrompido.*
  - *A instrução de soma só escreve o resultado no quinto estágio do pipeline, o que significa a necessidade da inserção de três bolhas.*

Conclusões

- *A teoria apresentada mostrou a execução de instruções em pipeline. Juntando todos os pontos vistos, começamos com um caminho de dados monociclo, passamos para caminhos multiciclo e, finalmente, pipeline.*

- *A técnica do pipeline melhora o tempo médio de execução das instruções através do incremento do throughput, embora não mexa no tempo total de execução ou latência.*