

Pipeline

Hazards (ou conflitos) de pipeline

- Ocorrem quando a próxima instrução não pode ser executada no ciclo de clock seguinte.

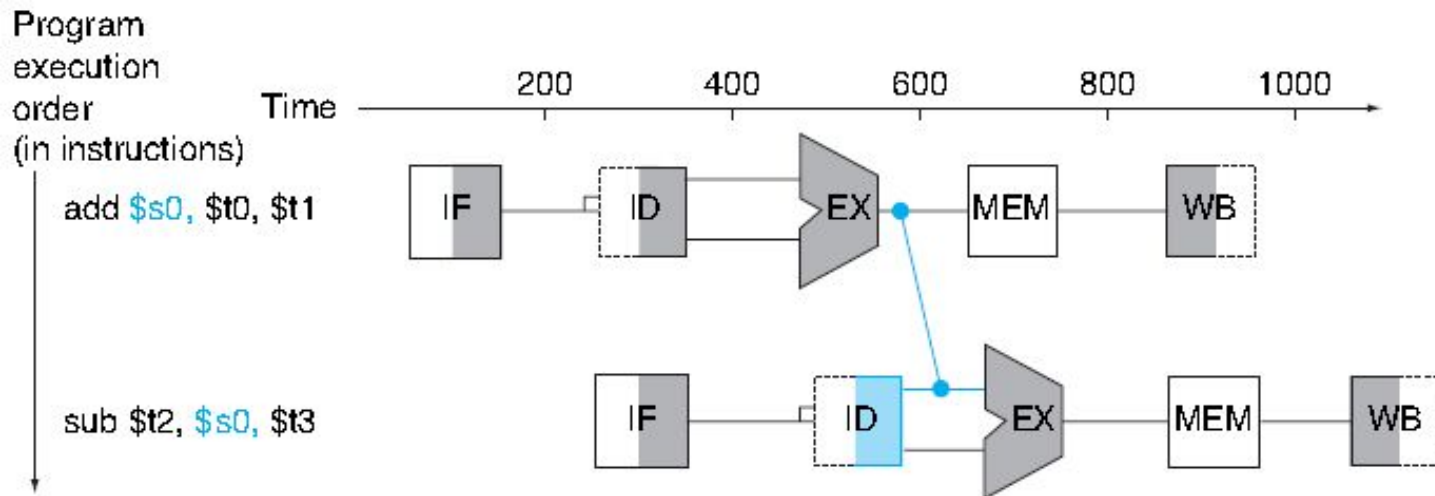
Exemplos?

- Hazards estruturais – conflito de acesso aos **mesmos recursos** (ex. memória – ler instrução e executar lw)
- Hazards de dados – quando ocorre **dependência** de dados. Exemplos?
- Hazards de controle – necessidade de tomar uma **decisão** com base em resultados anteriores.

Hazards de dados

Hazards (ou conflitos) de dados

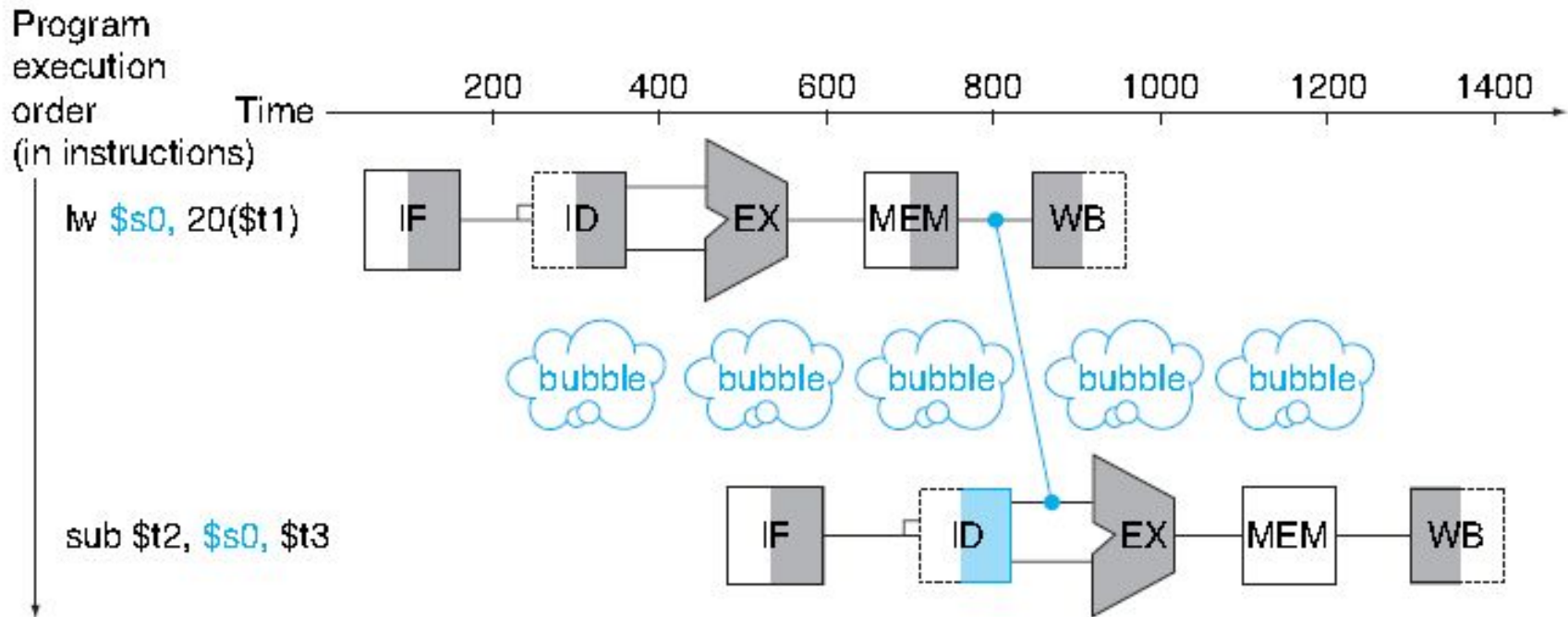
- Como tais dependências acontecem com muita frequência, o ideal é não ter que esperar a instrução anterior ser executada.
- Para isso, utiliza-se técnicas de **forwarding** ou bypassing.
- O **forwarding** é empregado quando o estágio de dependência estiver mais adiante da obtenção do valor requerido.



Hazards de dados

Hazards (ou conflitos) de dados

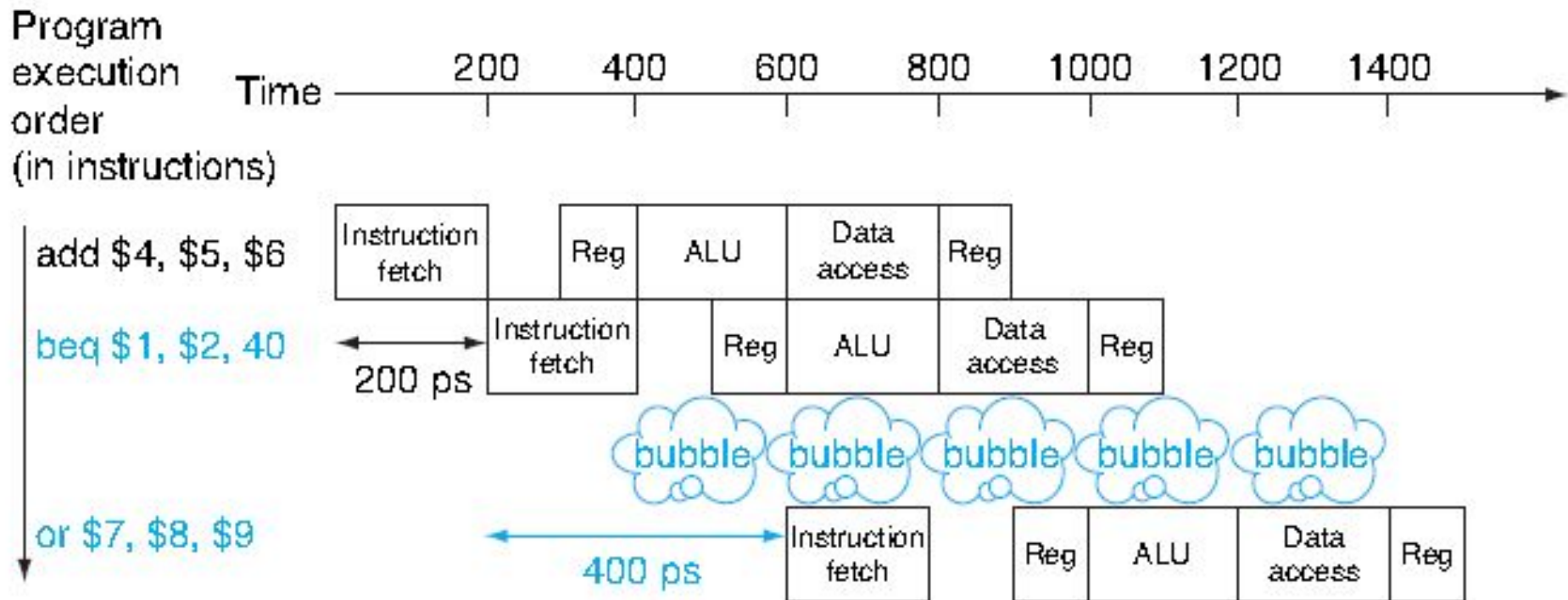
- Se o dado desejado não estiver disponível no estágio a seguir é necessário utilizar um *pipeline stall* (bolha).
- Exemplo: instrução de *load* e após instrução aritmética.



Hazards de controle

Hazards de controle – ocorrem quando as instruções subsequentes dependem de uma decisão anterior.

Exemplo **utilizando HW extra** para testar registradores.

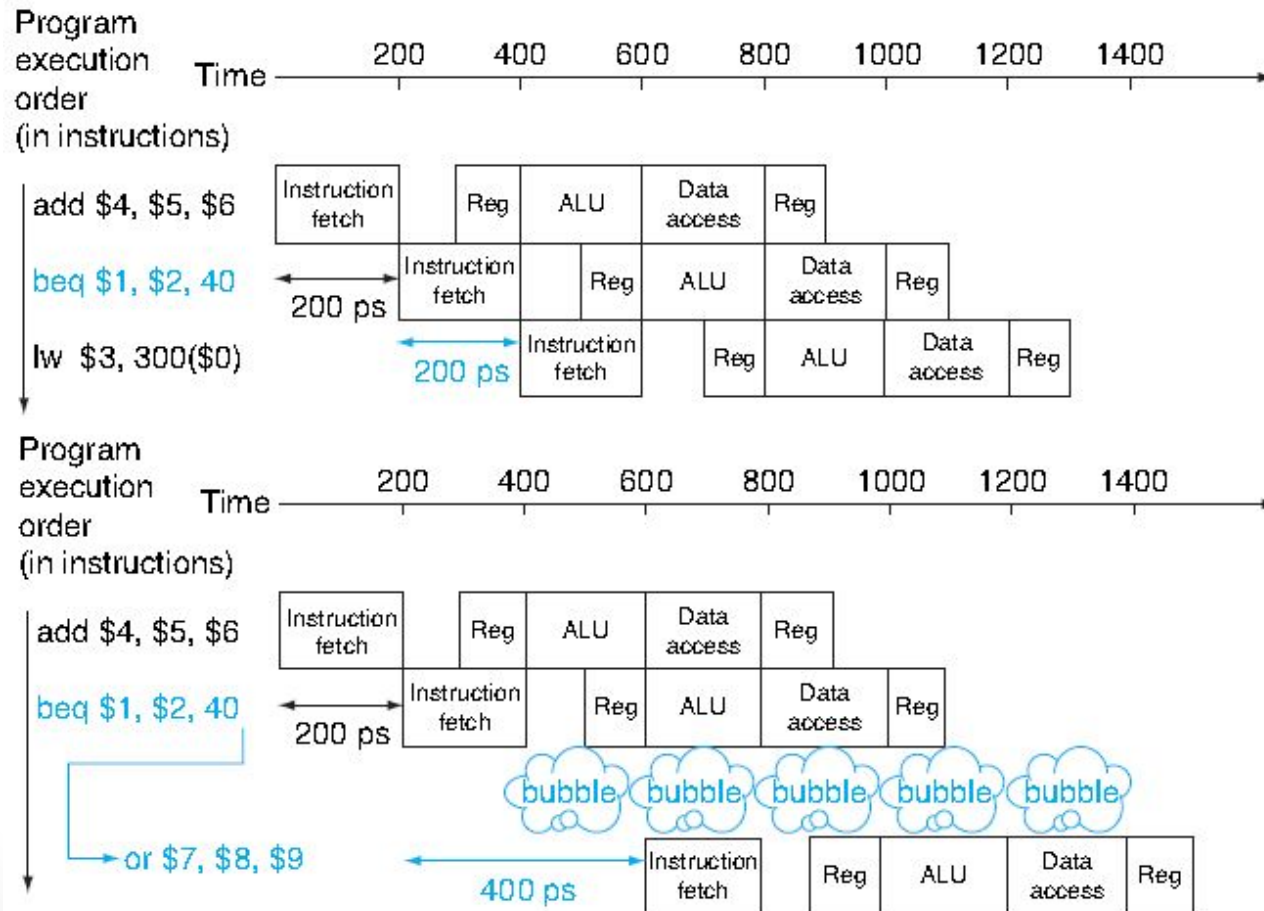


Hazards de controle

Hazards de controle – 2º solução: previsão de desvio.

Se a previsão for correta, não há atrasos no pipeline, porém se a decisão do desvio não ocorrer, então haverá o *stall*.

Uma possível solução é prever que os desvios **não** ocorrerão.



Hazards de controle

Hazards de controle – 2º solução: previsão de desvio.

- Previsores de hardware modernos tomam decisões de que alguns desvios são tomados e outros não.
- **Previsores dinâmicos** fazem as escolhas dependendo do comportamento de cada desvio.
- Uma técnica comum é manter o histórico de cada desvio.
- Previsores de desvios modernos conseguem uma precisão de 90% dos desvios tomados.

Quais são os impactos (vantagens/desvantagens) quando são utilizados pipelines mais longos?

Hazards de controle

Hazards de controle – 3º solução: decisão adiada (*delayed branch*).

- É a decisão utilizada pelo MIPS. O software do MIPS sempre colocará uma instrução imediatamente após a instrução de *delayed branch* que não é afetada pelo desvio.
- A decisão do desvio mudará o endereço da instrução que vem após essa instrução segura. Como é o caso da instrução *add* no exemplo abaixo.

