Caminho de Dados com Pipeline

Yuri Kaszubowski Lopes

UDESC

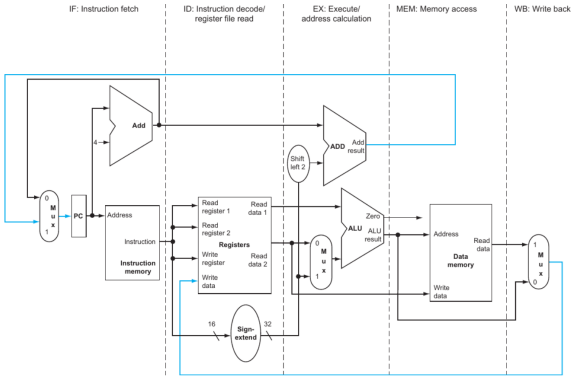
Anotações

Revisão: Pipeline

- Dividimos nosso processador MIPS em 5 estágios:
  1. `IF`: Instruction fetch (busca de instrução)
  2. `ID`: Instruction decode and register read (decodificação e leitura de registradores)
  3. `EX`: Execution or address calculation (execução)
  4. `MEM`: Data memory access(acesso a memória)
  5. `WB`: Write back (escrita dos resultados)
- Nunca voltamos no tempo
- No geral, nossos estágios vão da esquerda para a direita, exceto `WB`, onde o circuito retorna o resultado para os registradores
  - não viola os princípios do nosso pipeline
  - Basta visualizar que apesar de alguns componentes desses estágios estarem antes no pipeline, eles são utilizados em estágios posteriores

Anotações

Revisão: Pipeline

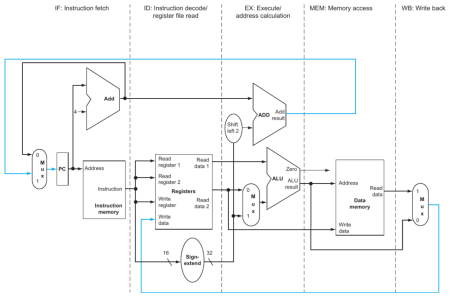


Anotações

Problemas na fronteira

- Considere o seguinte programa:

```
1 lw $2, 100($0)
2 lw $3, 200($0)
3 lw $4, 300($0)
```



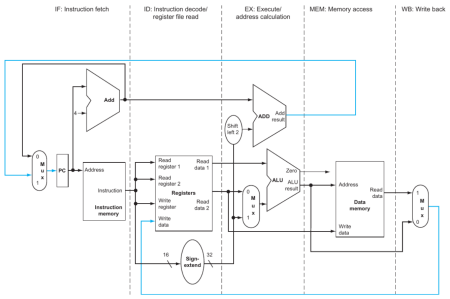
Anotações

Problemas na fronteira

- Considere o seguinte programa:

```
1 lw $2, 100($0)   # No estágio IF
2 lw $3, 200($0)
3 lw $4, 300($0)
```

- `PC` é enviado no primeiro estágio, de onde vai sair a primeira instrução



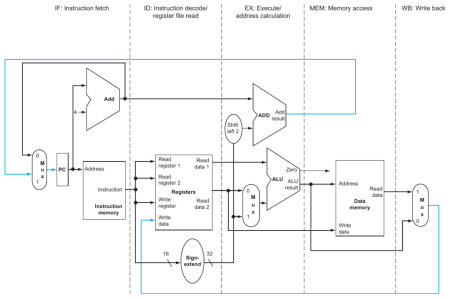
Anotações

Problemas na fronteira

- Considere o seguinte programa:

```
1 lw $2, 100($0)   # No estágio ID
2 lw $3, 200($0)   # No estágio IF
3 lw $4, 300($0)
```

- `PC` (atualizado) é enviado no primeiro estágio, de onde vai sair a segunda instrução
- Primeira instrução é enviado para o estágio `ID`, para ler os registradores



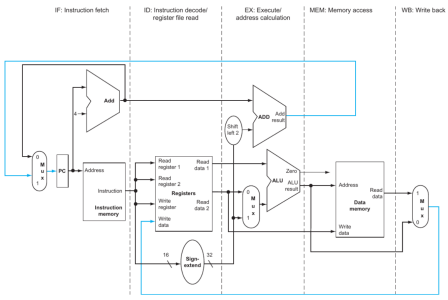
Anotações

Problemas na fronteira

- Considere o seguinte programa:

```
1 lw $2, 100($8)   # No estágio ID
2 lw $3, 200($8)   # No estágio IF
3 lw $4, 300($8)
```

- Problema: no estágio `ID`, desejamos a instrução que havia sido carregada no estágio `IF` no ciclo de clock anterior (`lw $2, 100($8)`), mas agora só temos o sinal da instrução atual em `IF`



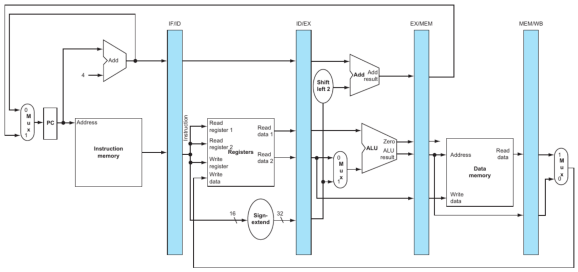
Anotações

Problemas na fronteira

- Problema: no estágio `ID`, desejamos a instrução que havia sido carregada no estágio `IF` no ciclo de clock anterior (`lw $2, 100($8)`), mas agora só temos o sinal da instrução atual em `IF`
- O problema se repete nos demais estágios, conforme as instruções "caminham" em nosso fluxo
- Como resolver?
  - No próximo ciclo de clock, o próximo estágio espera continuar o trabalho do estágio anterior
  - O trabalho feito no ciclo de clock anterior necessita ser salvo
  - Utilizamos **registradores de pipeline**
    - Salvamos toda a informação que é pertinente para o próximo estágio do pipeline no próximo ciclo de clock
    - Continuando com a **analogia** da lavanderia, teríamos cestos de roupas para armazenar a roupa antes de passar para o próximo estágio

Anotações

Registradores de Pipeline (Em azul)



- Os registradores que separam o estágio `i` do estágio `j`, são chamados registradores `i/j`, e.g., `IF/ID`
- Os registradores IF/ID precisam armazenar pelo menos 64 bits (32 do PC+4, e 32 da instrução)
- Considerando o estado atual do circuito, quantos bits possuem os demais registradores de pipeline?
  - ID/EX: 128 bits
  - EX/MEM: 97 bits
  - MEM/WB: 64 bits

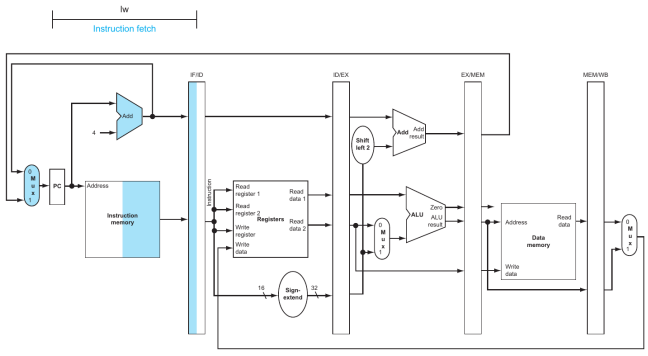
Anotações

Exemplo `lw`

- Fluxo de uma instrução `lw` no pipeline
- Considere que:
  - Quando a área sombreada dos registradores é a esquerda, os registradores estão sendo escritos
  - Quando a área sombreada dos registradores é a direita, os registradores estão sendo lidos

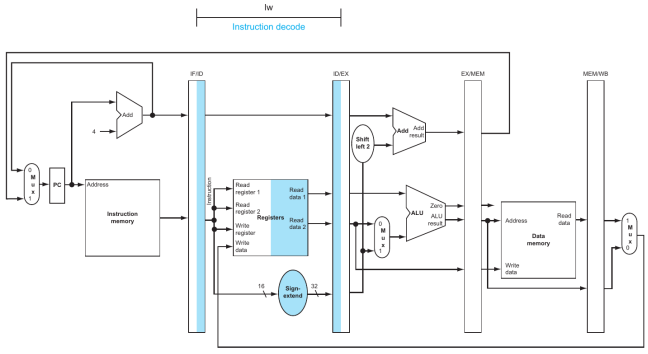
Anotações

`lw` no pipeline



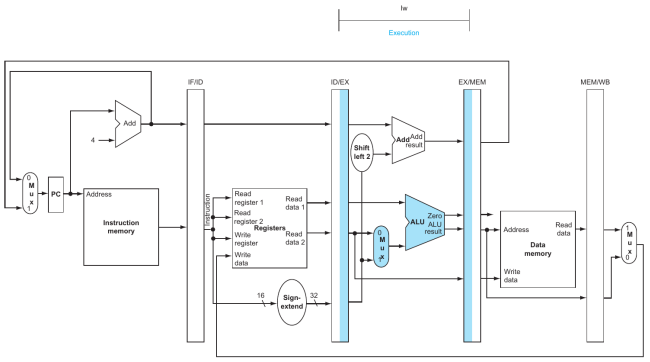
Anotações

`lw` no pipeline



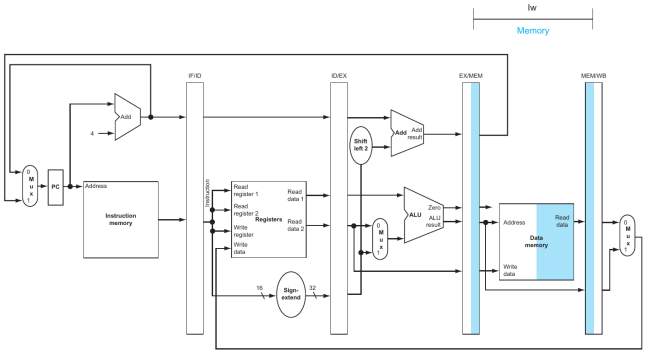
Anotações

lw no pipeline



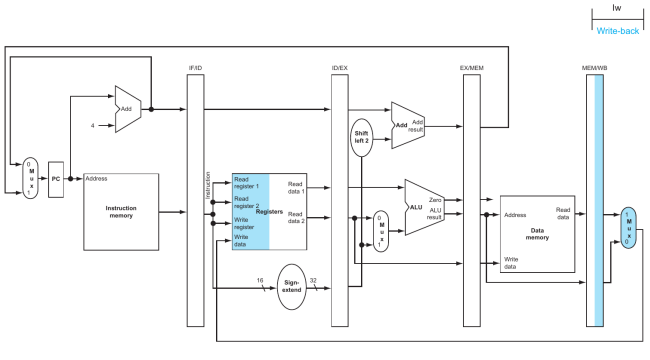
Anotações

lw no pipeline



Anotações

lw no pipeline



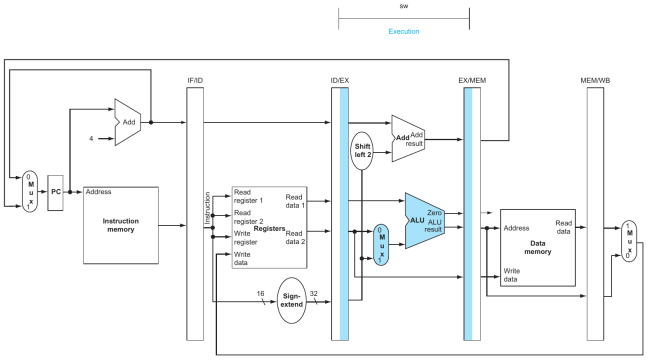
Anotações

Exemplo `sw`

- Fluxo de uma instrução `sw` no pipeline
- Os primeiros estágios do sw são os mesmos do lw

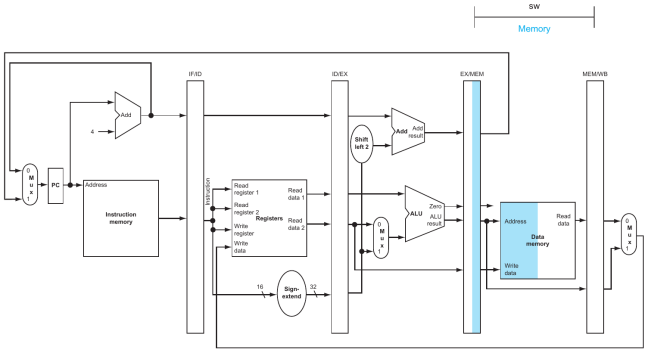
Anotações

`sw` no pipeline



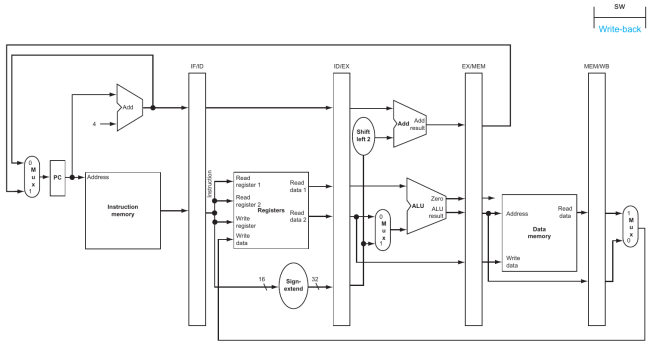
Anotações

`sw` no pipeline



Anotações

`sw` no pipeline



Anotações

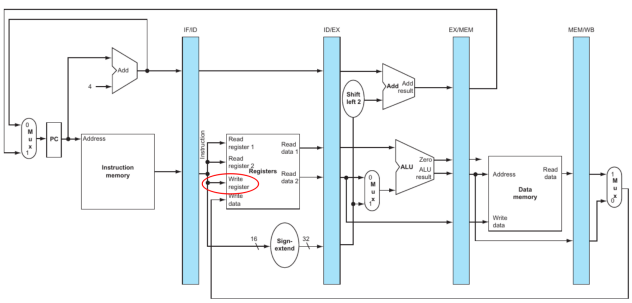
`sw` no pipeline

- No estágio `WB`, a instrução `sw` não realiza trabalho algum
  - Ainda assim não podemos "pular estágios"
  - "Adiantar" a execução da instrução que vem logo após o `sw` **não** pode ser feito
    - O estágio anterior pode ainda não ter terminado o seu trabalho
    - O componente poderá não estar livre

Anotações

BUG!

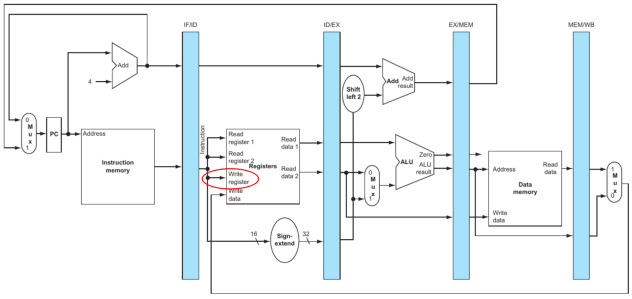
- O que é feito no estágio WB?
- Qual o problema com o registrador a ser escrito?
  - Dados de um estágio (uma instrução anterior), endereço de outro estágio (instrução mais "à frente").



Anotações

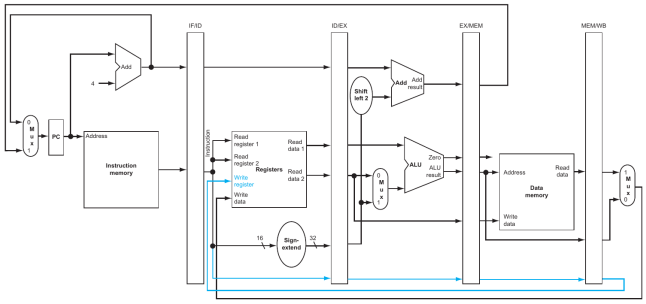
BUG!

- Precisamos salvar pelo menos o endereço do registrador de escrita até o estágio WB
- Por enquanto esse valor está se perdendo em nosso pipeline, e estamos escrevendo no registrador endereçado pela instrução que se encontra no estágio ID, e não pela instrução do estágio WB



Anotações

Correção do BUG



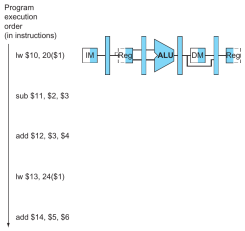
Anotações

Exercício

- Considere as instruções

```
1 lw $10, 20($1)
2 sub $11, $2, $3
3 add $12, $3, $4
4 lw $13, 24($1)
5 add $14, $5, $6
```

Faça um diagrama de múltiplos ciclos de clock para essas instruções (veja o exemplo para o lw). Em cada componente do diagrama, pinte-o de acordo com o exemplo para indicar que a unidade está sendo utilizada naquele estágio



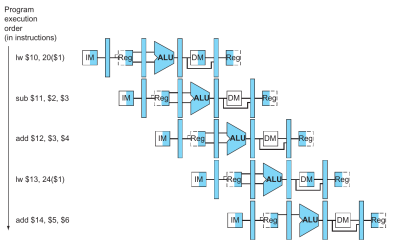
Anotações

Exercício

1. Considere as instruções

```
1 lw $10, 20($1)
2 sub $11, $2, $3
3 add $12, $3, $4
4 lw $13, 24($1)
5 add $14, $5, $6
```

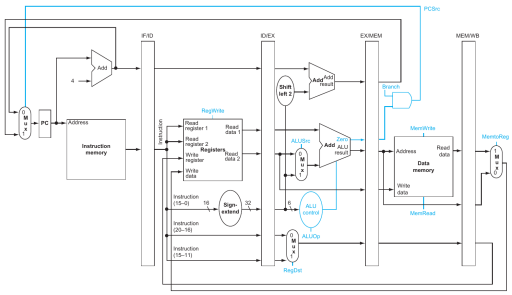
Faça um diagrama de múltiplos ciclos de clock para essas instruções (veja o exemplo para o lw). Em cada componente do diagrama, pinte-o de acordo com o exemplo para indicar que a unidade está sendo utilizada naquele estágio



Anotações

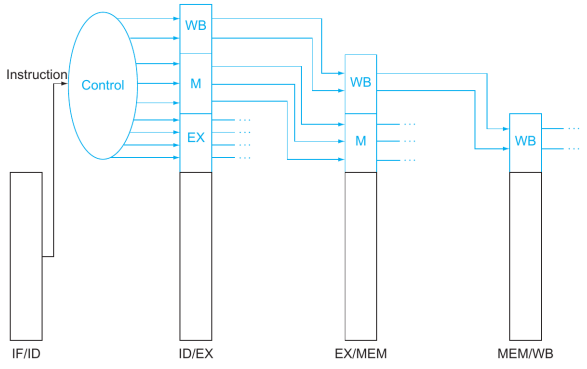
Adicionando sinais de controle

- Os sinais de controle são (por enquanto) os mesmos que na máquina de ciclo único
  - O sinal pode ser definido já no estágio ID
  - Podemos simplesmente ligá-los em nossa CPU? Problemas?
    - Diferentes sinais são utilizados em diferentes estágios do nosso pipeline
    - Assim como os dados, devemos salvar os sinais de controle
    - ... também nos registradores de Pipeline



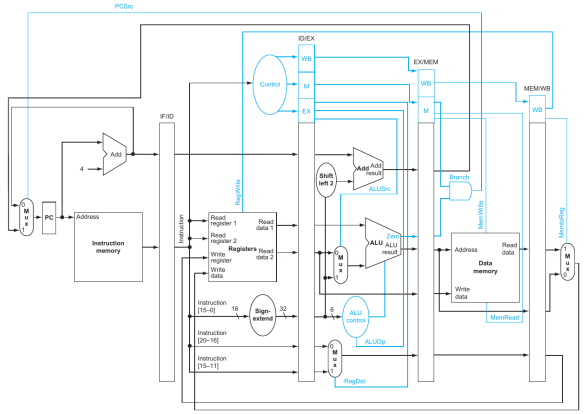
Anotações

Adicionando sinais de controle



Anotações

Adicionando sinais de controle



Anotações

Exercícios

2. Considere os registradores de pipeline do exercício anterior, agora com os sinais de controle. Qual o tamanho de cada um dos registradores de pipeline (`IF/ID`, `ID/EX`, ...)?

Anotações

Referências

- D. Patterson; J. Henessy. **Organização e Projeto de Computadores: Interface Hardware/Software**. 5a Edição. Elsevier Brasil, 2017.
- Andrew S. Tanenbaum. **Organização estruturada de computadores**. 5. ed. São Paulo: Pearson, 2007.
- Harris, D. and Harris, S. **Digital Design and Computer Architecture**. 2a ed. 2012.
- courses.missouristate.edu/KenVollmar/mars/

Anotações