

---

# Exceções

---

---

# Sumário

- Introdução;
- Como tratar Exceções;
- Como o controle verifica Exceções;
- Exceções no Pipeline;
- Bibliografia.

# Introdução

- Uma das partes mais difíceis do controle é implementar exceções e interrupções – eventos diferentes de desvios ou jumps que mudam o fluxo normal de execução das instruções;
- Uma exceção é um evento inesperado de dentro do processador:
  - overflow aritmético;
- Uma interrupção é um evento que também causa mudança inesperada no fluxo de controle mas vem de fora do processador:
  - Dispositivos de E/S

# Introdução

- Segundo a convenção MIPS, o termo exceção é usado para se referir a qualquer mudança inesperada no fluxo de controle;
- A detecção de condições excepcionais e a tomada da ação apropriada frequentemente estão no caminho crítico da sincronização da máquina;
  - Sem atenção apropriada às exceções podemos reduzir significativamente o desempenho;

# Como tratar Exceções

- A ação básica que a máquina precisa realizar quando ocorre uma exceção é:
  - salvar o endereço da instrução problemático no contador de programa (*Exception Programa Counter - EPC*) para exceções e, depois,
  - transferir o controle para o S.O. em alguns endereços especificados;
- O Sistema Operacional, então, pode tomar a ação apropriada, que pode:
  - fornecer algum serviço ao programa do usuário;
  - Tomar alguma ação predefinida em resposta a um overflow ou
  - Interromper a execução do programa e retornar um erro

# Como tratar Exceções

- Depois de realizar qualquer ação necessária devido à exceção, o S.O. pode terminar o programa ou continuar sua exceção usando o EPC para determinar onde reiniciar a execução do programa;
- Para que o Sistema Operacional trate a exceção, ele precisa conhecer o motivo de exceção, além da instrução que a causou;
- Existem dois métodos principais usados para comunicar o motivo de uma exceção:
  - Registrador de status; (MIPS)
  - Interrupções vetorizadas;

# Como tratar Exceções

- Em uma interrupção vetorizada, o endereço para o qual o controle é transferido é determinado pela causa da exceções;

Tipo de exceção	End. do vetor de exceções
Instrução indefinida	C000 0000
Overflow aritmético	C000 0020

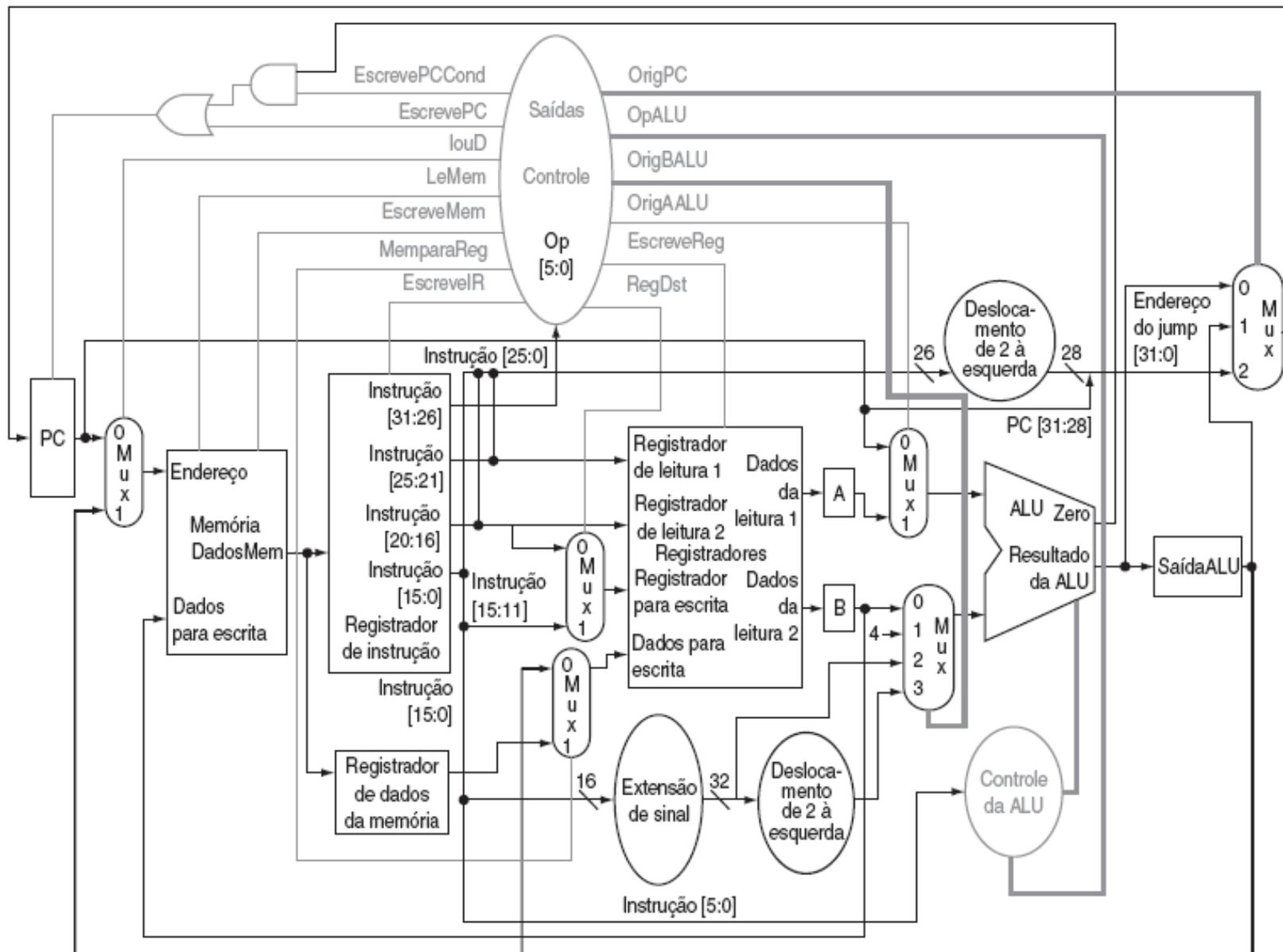
- O método usado na arquitetura MIPS é incluir dois registradores
  - de status (chamado *Registrador Cause*), que contém um campo indicado o motivo da exceção;
  - EPC: um registrador de 32 bits usado para conter o endereço da instrução afetada; (Esse registrador é necessário mesmo quando as exceções são vetorizadas)

# Como tratar Exceções

- Precisamos incluir dois sinais de controle para fazer com que os registradores EPC e Cause sejam escritos:
  - Escreve EPC e EscreveCause;
- Além disso, precisamos de um sinal de controle (*Causalnt*) de 1 bit para definir corretamente o bit menos significativo do registrador Cause;
- Finalmente, precisaremos ser capazes de escrever no PC o endereço de exceção
  - OrigPC pode ser colocado em 11 para selecionar esse valor para ser escrito no PC;

# Como tratar Exceções

- Como o PC é incrementado durante o primeiro ciclo de cada instrução, não podemos simplesmente escrever o valor PC no EPC;
  - Solução: usar a ULA para subtrair 4 do PC e escrever a saída no EPC;
- Usando o caminho de dados da Figura 5.39 a ação a ser tomada para cada tipo diferente de exceção pode ser tratada em um estado para cada uma;
  - Entretanto, antes de incluir os esses estados, precisamos determinar como verificar exceções.

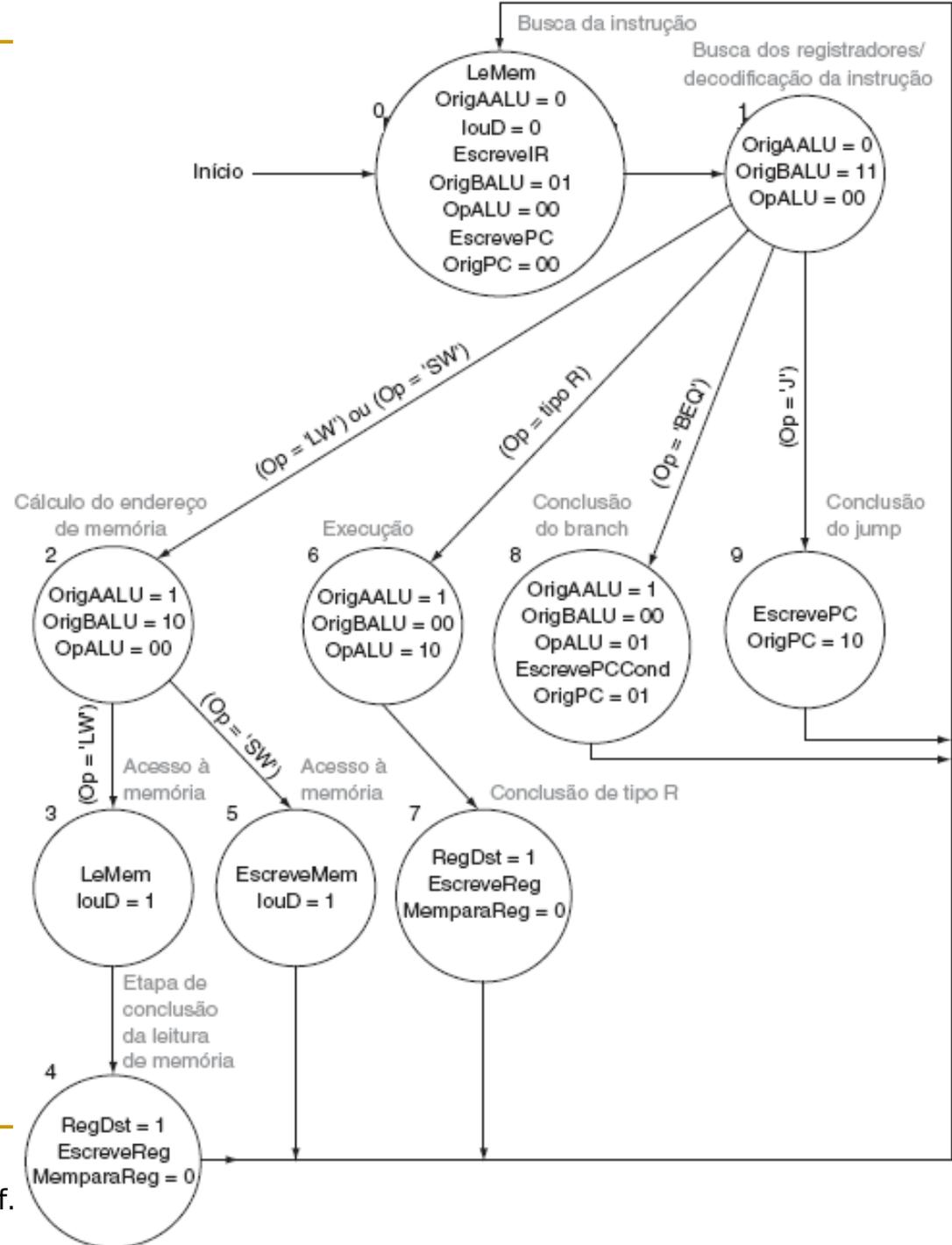


# Como o controle verifica Exceções

- A Figura 5.40 mostra os dois novos estados para detectar exceções e transferir o controle para o estado apropriado;
- Cada uma das duas exceções possíveis é detectada diferentemente:
  - Instrução indefinida: detectado quando nenhum próximo estado é definido do estado 1 para o valor de Op.
  - Overflow aritmético: a ULA inclui lógica para detectar overflow e um sinal chamado *Overflow*;

# Etapas de Execução

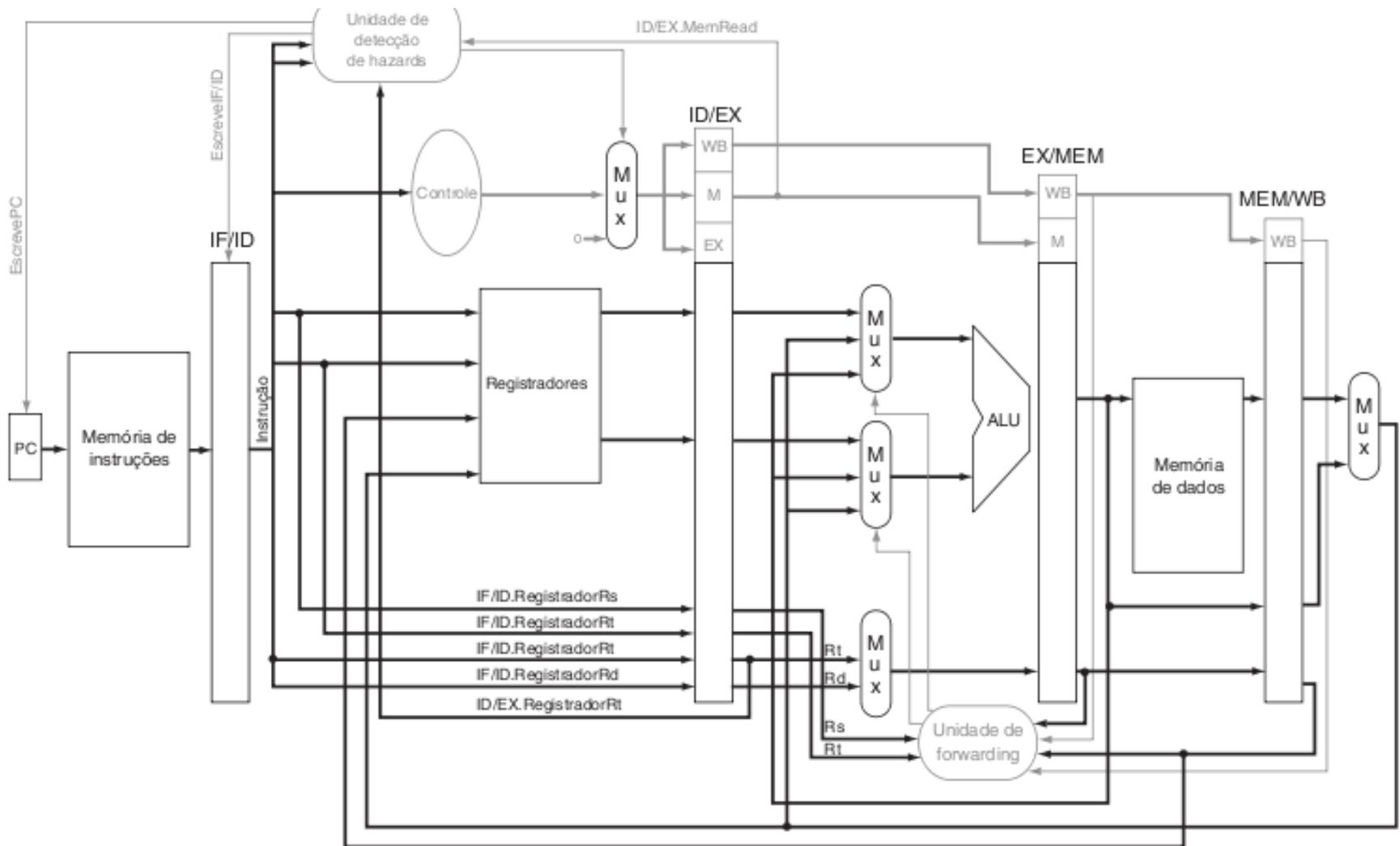
- Especificação gráfica:



# Exceções no Pipeline

- As exceções no pipeline são vistas como outra forma de hazard de controle;
- Para tanto vejamos o caminho de dados e de controle com as unidades par lidar com forwarding e hazards

# Exceções no Pipeline



---

# Bibliografia

- PATERSON, D. A. & HENNESSY, J. L. *Organização e Projeto de Computadores: a interface hardware/software*, Editora Campus. 3ª ed. RJ: 2005.