

# Arquitetura de Computadores

Paralelismo em Nível de Instrução  
Previsão de Desvios Avançadas

# Previsão de Desvios

- As previsões de desvios tentam evitar atrasos no pipeline, existem previsões estáticas e dinâmicas implementadas em hardware ou em software ou ainda uma combinação.

# Previsão de Desvios Estáticas

- As ações para um desvio são estáticas — elas são fixas para cada desvio durante toda a execução.
- O software pode tentar minimizar a penalidade de desvio usando o conhecimento do esquema de hardware e do comportamento do desvio.

# Previsão de Desvios Estáticas

- O esquema mais simples para lidar com desvios é congelar ou liberar (***freeze or flush***) o pipeline, retendo ou excluindo quaisquer instruções após o desvio até que o destino do desvio seja conhecido.
- A atratividade dessa solução reside principalmente em sua simplicidade, tanto para hardware quanto para software.
- Nesse caso, a penalidade por desvio é fixa e não pode ser reduzida por software.

# Previsão de Desvio *Freeze or Flush*

- Exemplo:

- A Figura mostra que o método mais simples de lidar com desvios é refazer a busca da instrução(IF) seguinte a um desvio, uma vez que detectamos o desvio durante a ID.
- Note que, se o desvio não for realizado, a repetição do estágio IF é desnecessária, pois a instrução correta foi de fato buscada.

|                       | 1         | 2         | 3          | 4          | 5          | 6          | 7          | 8  | 9 |
|-----------------------|-----------|-----------|------------|------------|------------|------------|------------|----|---|
| Instrução DESVIO      | <u>IF</u> | ID        | <u>EXE</u> | <u>MEM</u> | WB         |            |            |    |   |
| Instrução Sequente    |           | <u>IF</u> | ID         | <u>EXE</u> | <u>MEM</u> | WB         |            |    |   |
| Instrução Sequente +1 |           |           | <u>IF</u>  | ID         | <u>EXE</u> | <u>MEM</u> | WB         |    |   |
| Instrução Sequente +2 |           |           |            | <u>IF</u>  | ID         | <u>EXE</u> | <u>MEM</u> | WB |   |

# Previsão de Desvios Estáticas

- Um esquema de maior desempenho, e apenas um pouco mais complexo, consiste em tratar cada desvio como não realizado (***not taken***), simplesmente permitindo que o hardware continue como se o desvio não tivesse sido executado.
- Nesse caso, é preciso ter cuidado para não alterar o estado do processador até que o resultado do desvio seja definitivamente conhecido.
- A complexidade desse esquema advém da necessidade de saber quando o estado pode ser alterado por uma instrução e como "reverter" essa alteração.

# Previsão de Desvios Estáticas

- Um esquema alternativo é tratar cada desvio como realizado (***taken***). Assim que o desvio é decodificado e o endereço de destino é computado, assumimos que o desvio foi realizado e iniciamos a busca e a execução no destino.
- Isso nos proporciona uma melhoria de um ciclo quando o desvio é efetivamente realizado, pois conhecemos o endereço de destino no final do ID, um ciclo antes de sabermos se a condição de desvio foi satisfeita na etapa da ALU.
- Em um esquema de previsão ***taken*** ou ***not taken***, o compilador pode melhorar o desempenho organizando o código de forma que o caminho mais frequente corresponda à escolha do hardware.

# Previsão de Desvio *taken* – *not taken*

- Exemplo:

|                                   | 1         | 2         | 3          | 4          | 5          | 6          | 7          | 8          | 9  |
|-----------------------------------|-----------|-----------|------------|------------|------------|------------|------------|------------|----|
| Instrução DESVIO <u>not taken</u> | <u>IF</u> | ID        | <u>EXE</u> | <u>MEM</u> | WB         |            |            |            |    |
| Instrução I+1                     |           | <u>IF</u> | ID         | <u>EXE</u> | <u>MEM</u> | WB         |            |            |    |
| Instrução I+2                     |           |           | <u>IF</u>  | ID         | <u>EXE</u> | <u>MEM</u> | WB         |            |    |
| Instrução I+3                     |           |           |            | <u>IF</u>  | ID         | <u>EXE</u> | <u>MEM</u> | WB         |    |
| Instrução I+4                     |           |           |            |            | <u>IF</u>  | ID         | <u>EXE</u> | <u>MEM</u> | WB |
| <br>                              |           |           |            |            |            |            |            |            |    |
| Instrução DESVIO <u>taken</u>     | <u>IF</u> | ID        | <u>EXE</u> | <u>MEM</u> | WB         |            |            |            |    |
| Instrução I+1                     |           | <u>IF</u> | bolha      | bolha      | bolha      | bolha      |            |            |    |
| Instrução ALVO                    |           |           | <u>IF</u>  | ID         | <u>EXE</u> | <u>MEM</u> | WB         |            |    |
| Instrução ALVO+1                  |           |           |            | <u>IF</u>  | ID         | <u>EXE</u> | <u>MEM</u> | WB         |    |
| Instrução ALVO+2                  |           |           |            |            | <u>IF</u>  | ID         | <u>EXE</u> | <u>MEM</u> | WB |

# Previsão de Desvios Estáticas

- Um quarto esquema, muito utilizado nos primeiros processadores RISC, é chamado de desvio atrasado (***delayed branch***). Em um desvio atrasado, o ciclo de execução com um atraso de desvio igual a um é:
  - instrução de desvio
  - Instrução sucessora sequencial
  - Instrução alvo do desvio (se realizado)
- O instrução sucessora sequencial está no *slot* de atraso de desvio. Esta instrução é executada independentemente de o desvio ser realizado ou não.

# Previsão *Branch Delayed*

- Exemplo:

|                                   | 1         | 2         | 3          | 4          | 5          | 6          | 7          | 8          | 9  |
|-----------------------------------|-----------|-----------|------------|------------|------------|------------|------------|------------|----|
| Instrução DESVIO <b>not taken</b> | <u>IF</u> | ID        | <u>EXE</u> | <u>MEM</u> | WB         |            |            |            |    |
| Instrução no slot (I+1)           |           | <u>IF</u> | ID         | <u>EXE</u> | <u>MEM</u> | WB         |            |            |    |
| Instrução I+2                     |           |           | <u>IF</u>  | ID         | <u>EXE</u> | <u>MEM</u> | WB         |            |    |
| Instrução I+3                     |           |           |            | <u>IF</u>  | ID         | <u>EXE</u> | <u>MEM</u> | WB         |    |
| Instrução I+4                     |           |           |            |            | <u>IF</u>  | ID         | <u>EXE</u> | <u>MEM</u> | WB |
| <br>                              |           |           |            |            |            |            |            |            |    |
| Instrução DESVIO <b>taken</b>     | <u>IF</u> | ID        | <u>EXE</u> | <u>MEM</u> | WB         |            |            |            |    |
| Instrução no slot (I+1)           |           | <u>IF</u> | ID         | <u>EXE</u> | <u>MEM</u> | WB         |            |            |    |
| Instrução ALVO                    |           |           | <u>IF</u>  | ID         | <u>EXE</u> | <u>MEM</u> | WB         |            |    |
| Instrução ALVO+1                  |           |           |            | <u>IF</u>  | ID         | <u>EXE</u> | <u>MEM</u> | WB         |    |
| Instrução ALVO+2                  |           |           |            |            | <u>IF</u>  | ID         | <u>EXE</u> | <u>MEM</u> | WB |

# Previsão de Desvios Estática

- Desempenho dos Esquemas de Previsão Desvios
  - Qual é o desempenho efetivo de cada um desses esquemas?

A aceleração efetiva do pipeline com penalidades de desvios, assumindo um CPI ideal de 1, é:

$$\text{aceleração do pipeline} = \frac{\text{profundidade pipeline}}{1+n^{\circ} \text{ ciclos stall dos desvios}} \quad \text{onde:}$$

$n^{\circ}$  ciclos stall dos desvios = frequência de desvios X penalidade do desvio.

A frequência de desvios e a penalidade de desvios podem ter componentes de desvios incondicionais e condicionais (mais frequente).

# Desempenho dos Esquemas de Desvios

- Para um pipeline mais profundo, como o de um processador MIPS R4000 e RISC posterior, são necessários pelo menos três estágios de pipeline antes que o endereço de destino do desvio seja conhecido e; um ciclo adicional antes que a condição de desvio seja avaliada, assumindo que não haja stalls nos registradores na comparação condicional.

Um atraso de três estágios leva às penalidades de desvio para os três esquemas de previsão mais simples listados abaixo:

- Esquema Flush → incodicional=2 not taken=3 taken = 3
- Esquema Taken → incodicional=2 not taken=3 taken = 2
- Esquema Not Taken → incodicional=2 not taken=0 taken = 3

- Encontre a efetiva adição ao CPI decorrente de desvios para este pipeline, assumindo as seguintes frequências:

- Desvios Incodicionais 4%
- Desvios Condicionais Realizados 6%
- Desvios Condicionais Não Realizados 10%

# Desempenho dos Esquemas de Desvios

- Calculamos os CPIs multiplicando a frequência relativa de desvios incondicionais, condicionais não realizados e condicionais realizados pelas respectivas penalidades. Logo:

|           | Incondicionais | Cond.Realizado  | Não realizado   | Total |
|-----------|----------------|-----------------|-----------------|-------|
| Flush     | $0,04*2=0,08$  | $0,06 * 3=0,18$ | $0,01*3 = 0,30$ | 0,56  |
| Taken     | $0,04*2=0,08$  | $0,06 * 3=0,18$ | $0,01*2 = 0,20$ | 0,46  |
| Not Taken | $0,04*2=0,08$  | $0,06 * 0=0,00$ | $0,01*3 = 0,30$ | 0,36  |

- As diferenças entre os esquemas aumentam substancialmente com esse atraso maior. Se o CPI base fosse 1 e os desvios fossem a única fonte de paradas, o pipeline ideal seria 1,56 vezes mais rápido do que um pipeline que utilizasse o esquema de stalls. O esquema desvios not taken seria 1,13 vezes melhor do que o esquema de flush sob as mesmas premissas.

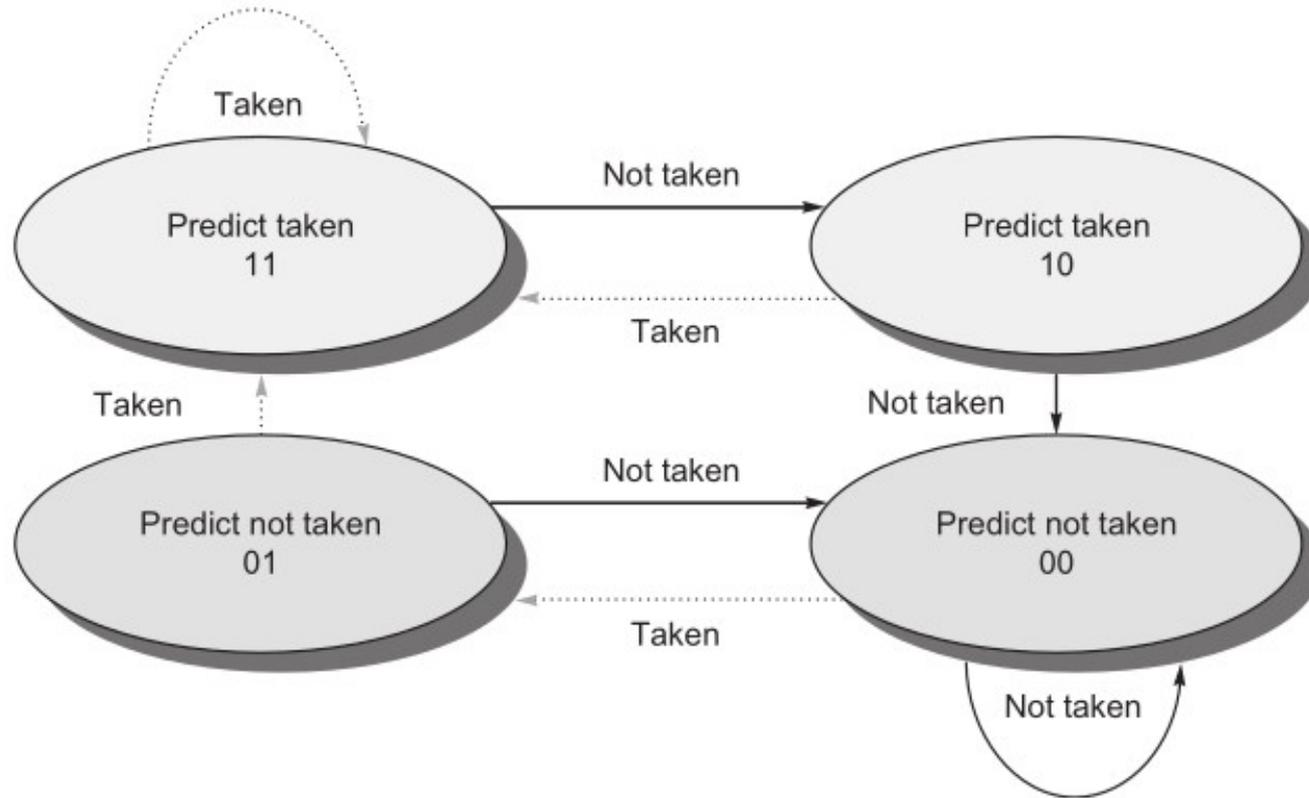
# Previsão de Desvios Dinâmica

- O esquema mais simples de previsão dinâmica de desvios é um **buffer de previsão de desvios** ou **uma tabela de histórico de desvios**.
  - Um buffer de previsão de desvios é uma pequena memória indexada pela parte inferior do endereço da instrução de desvio. A memória contém um bit que indica se o desvio foi realizado recentemente ou não.
  - Este esquema é o tipo mais simples de buffer; não possui tags
  - Útil apenas para reduzir o atraso do desvio quando este for maior que o tempo necessário para calcular os possíveis Pcs-alvo.
  - É uma dica que se presume estar correta, se a dica estiver errada, o bit de previsão é invertido e armazenado de volta.
  - Este buffer é efetivamente um cache onde cada acesso é um acerto e, o desempenho depende tanto da frequência da previsão para o desvio de interesse quanto da precisão da previsão quando ela corresponde.

# Previsão de Desvios Dinâmica

- Este esquema simples de previsão de 1 bit tem uma deficiência de desempenho: mesmo que um desvio seja quase sempre tomado, provavelmente faremos previsões incorretas duas vezes, em vez de uma, quando ele não for tomado, porque a previsão incorreta faz com que o bit de previsão seja invertido.
- Para remediar essa fraqueza, esquemas de previsão de 2 bits são frequentemente utilizados. Em um esquema de 2 bits, uma previsão precisa falhar duas vezes antes de ser alterada. A Figura a seguir mostra o processador de estados finitos para um esquema de previsão de 2 bits.

# Previsão de Desvios 2-bits

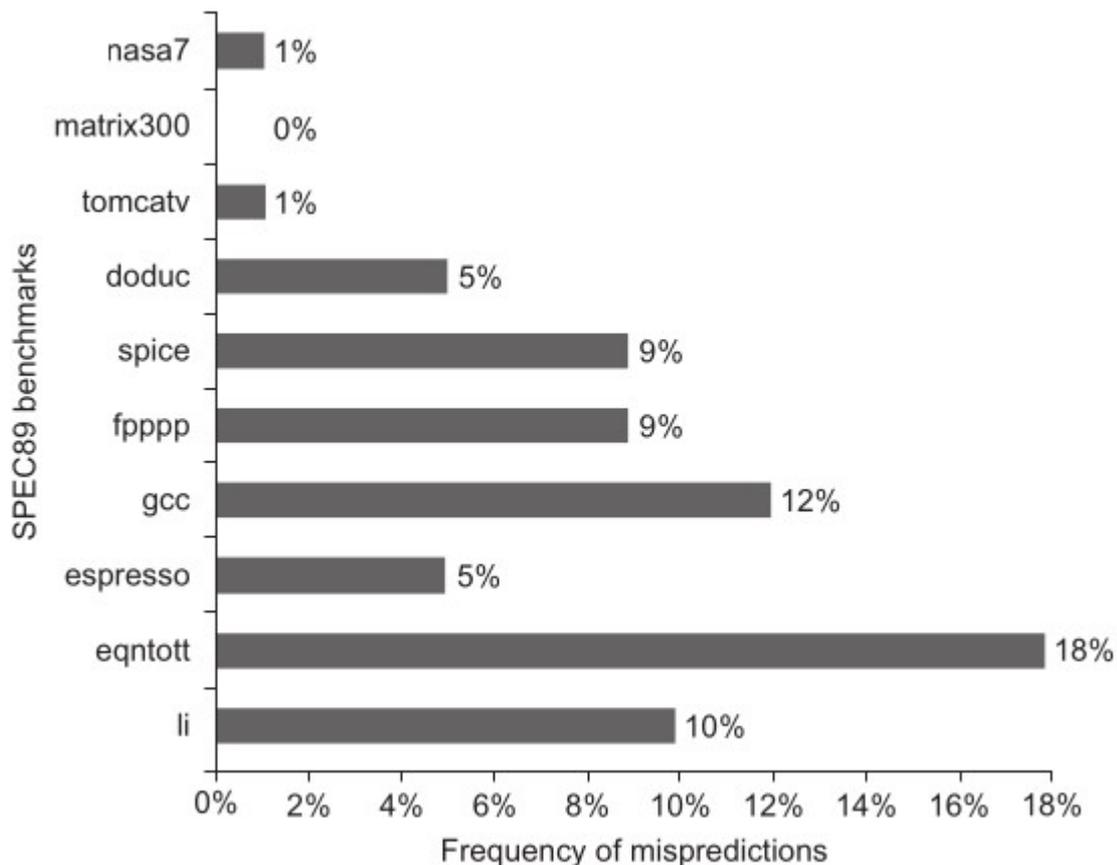


# Previsão de Desvios 2-bits

- Um buffer de previsão de desvio pode ser implementado como um pequeno "cache" especial acessado com o endereço da instrução durante o estágio de pipe IF, ou como um par de bits anexado a cada bloco no cache de instruções e buscado com a instrução. Se a instrução for decodificada como um desvio e se o desvio for previsto como realizado, a busca começa no alvo assim que o PC for conhecido. Caso contrário, a busca e a execução sequenciais continuam. Como mostra a Figura anterior, se a previsão estiver errada, os bits de previsão são alterados.

# Previsão de Desvios 2-bits

- Que tipo de precisão pode ser esperada de um buffer de previsão de desvios usando 2 bits por entrada em aplicações reais?
- O gráfico mostra que, para os benchmarks SPEC89, um buffer de previsão de desvios com 4.096 entradas resulta em uma precisão de previsão



# Previsão de Desvios Avançadas

- Preditores de desvios Correlacionados
  - Os esquemas de predição de 2 bits usam apenas o comportamento recente de um único desvio para prever o comportamento futuro desse desvio.
  - Pode ser possível melhorar a precisão da previsão se também observarmos o comportamento recente de outros desvios, em vez de apenas o desvio que estamos tentando prever.
  - Considere um pequeno fragmento de código do benchmark eqntott, um membro das primeiras suítes de benchmarks SPEC que apresentou um comportamento de previsão de desvio particularmente ruim:

# Previsão de Desvios Avançadas

- Seja o código:

```
if (aa==2)
    aa=0;
if (bb==2)
    bb=0;
if (aa!=bb) {
```

- Traduzindo temos:

```
    add  x3, x1, -2 //x1 é aa
    bnez x3, L1
    add  x1, x0, x0 // aa=0
L1: add  x3, x2, -2 //x2 é bb
    bnez x3, L2
    add  x2, x0, x0 //bb=0
L2: sub  x3, x1, x2 //x3=aa-bb
    beqz x3, L3
```

# Previsão de Desvios Avançadas

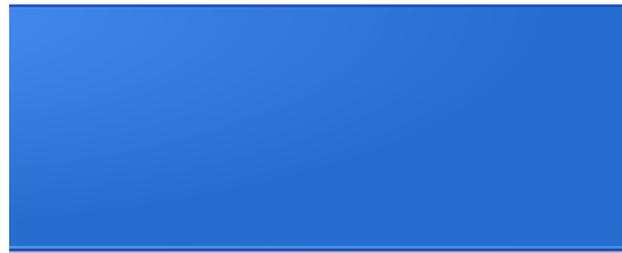
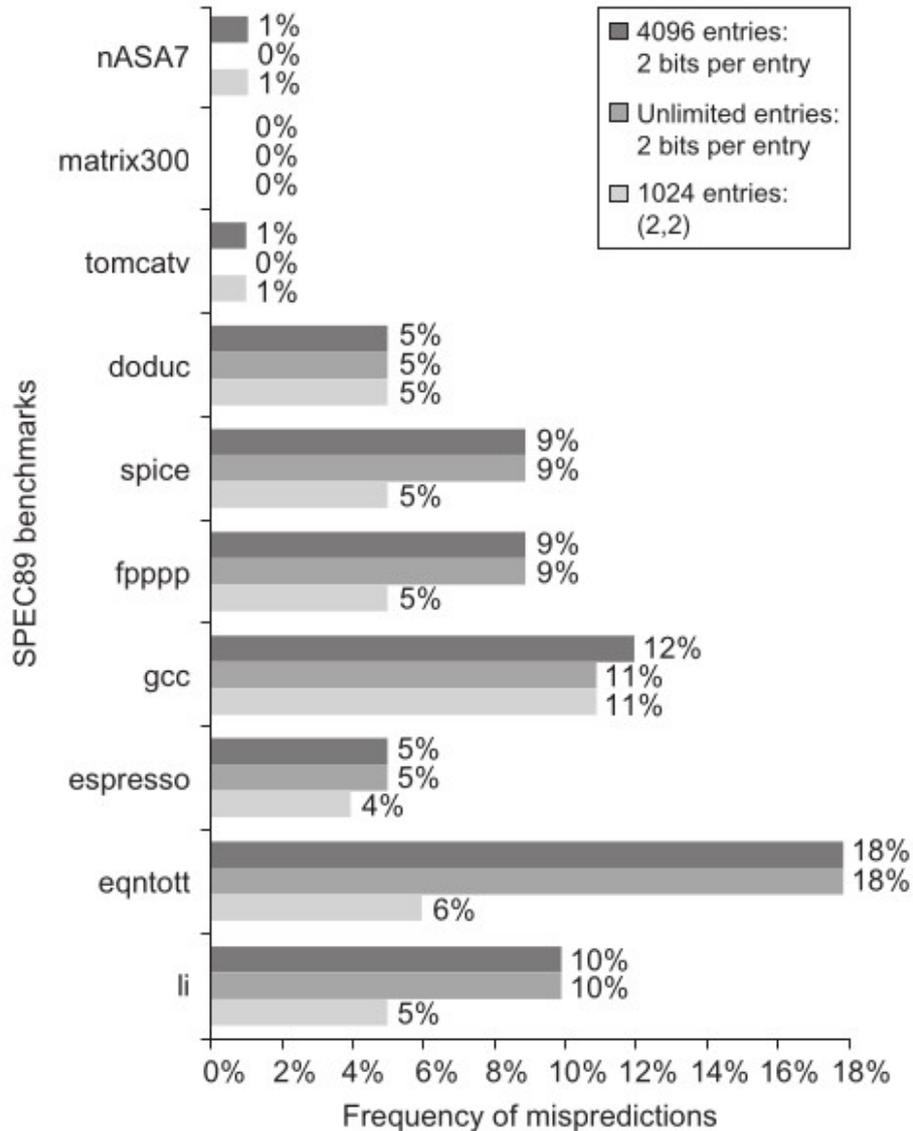
- Vamos chamar esses desvios de  $b_1$ ,  $b_2$  e  $b_3$ .
  - A observação principal é que o comportamento do desvio  $b_3$  está correlacionado com o comportamento dos desvios  $b_1$  e  $b_2$ .
  - Claramente, se nenhum dos desvios  $b_1$  nem  $b_2$  for realizado então  $b_3$  será realizado, porque  $a_a$  e  $b_b$  são claramente iguais (0).
  - Um preditor que utiliza o comportamento de apenas um desvio para prever o resultado desse desvio nunca poderá capturar esse comportamento.

# Preditores de Dois Níveis

- Os preditores de desvios que usam o comportamento de outros desvios para fazer uma previsão são chamados de preditores correlacionais ou preditores de dois níveis.
- Os preditores de correlação existentes adicionam informações sobre o comportamento dos desvios mais recentes para decidir como prever um determinado desvio.

# Preditor(m,n)

- Preditor (1,2)
  - Usa o comportamento do último desvio para escolher entre um par de preditores de desvio de 2 bits para prever um desvio específico.
  - No caso geral, um preditor (m,n) usa o comportamento dos últimos m desvios para escolher entre  $2^m$  preditores de desvios, cada um dos quais é um preditor de n bits para um único desvio.
  - A vantagem desse tipo de preditor de desvio correlacionado é que ele pode gerar taxas de predição mais altas do que o esquema de 2 bits e requer apenas uma quantidade trivial de hardware adicional.



A Figura compara as taxas de erro de previsão do preditor anterior (0,2) com 4 mil entradas e de um preditor (2,2) com 1 mil entradas.

Como você pode ver, o preditor correlacional não só supera um preditor simples de 2 bits com o mesmo número total de bits de estado, como também frequentemente supera um preditor de 2 bits com um número ilimitado de entradas.

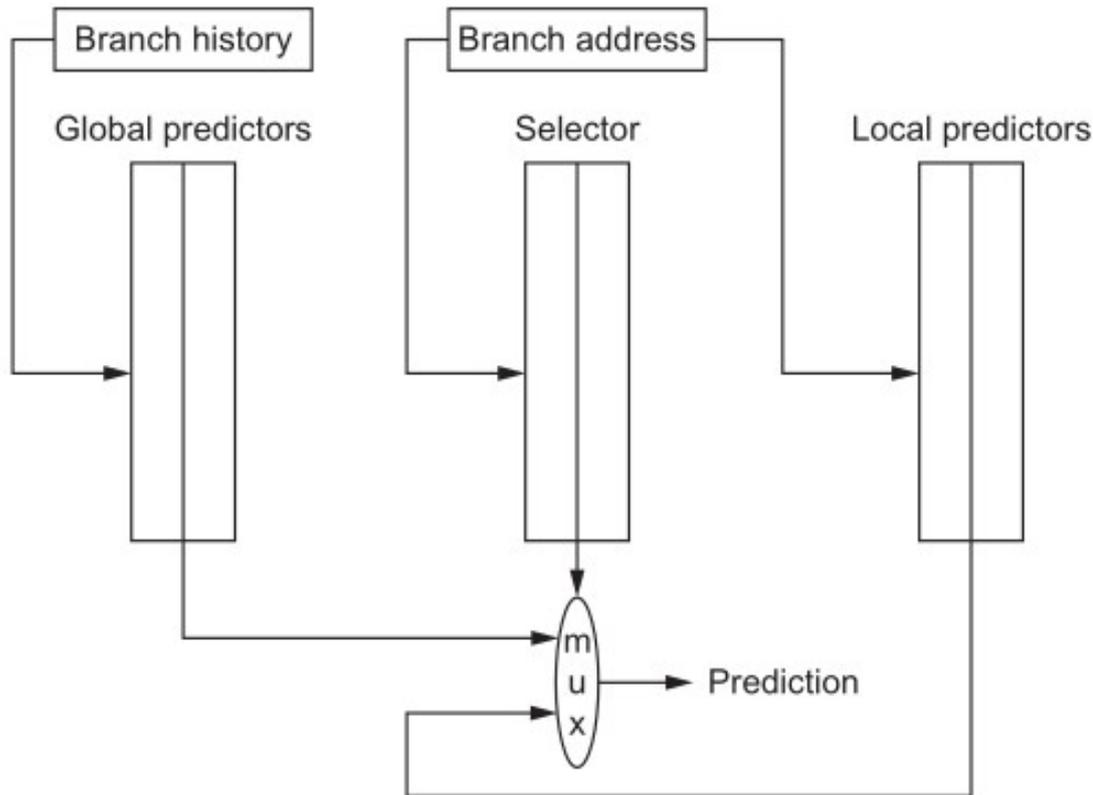
# Preditores de Torneios

- Combinam desvios locais e desvios globais.
- Os preditores de torneio podem alcançar maior precisão em tamanhos médios (8K–32K bits) e também utilizar com eficiência um grande número de bits de previsão.
- A vantagem de um preditor de torneio é sua capacidade de selecionar o preditor correto para um desvio específico.

# Preditores de Torneios

- Um preditor global usa o histórico de desvios mais recente para indexá-lo, enquanto um preditor local usa o endereço do desvio como índice. Os preditores de torneio são outra forma de preditores híbridos ou combinados.

# Preditores de Torneios



**Um preditor de torneio que usa o endereço do desvio para indexar um conjunto de contadores de seleção de 2 bits, que escolhem entre um preditor local e um global.**

-Índice da tabela do seletor é o endereço do desvio atual.

-As duas tabelas são preditores de 2 bits indexados pelo histórico global e pelo endereço do desvio.

-O seletor é um preditor de 2 bits, alterando o preditor preferencial para um endereço de desvio quando duas previsões incorretas ocorrem em sequência.

-A predição incorreta implica em alterar tanto a tabela de seletores quanto os preditores global ou local.

# Preditores de Torneios

- Um preditor de torneio típico selecionará o preditor global quase 40% das vezes para os benchmarks inteiros do SPEC e menos de 15% das vezes para os benchmarks de FP do SPEC.
- Além dos processadores Alpha, pioneiros nos preditores de torneio, vários processadores AMD utilizaram preditores no estilo de torneio.

# Preditores de Torneios

