

Universidade Federal do Rio de Janeiro
DCC/IM

Arquitetura de Computadores II

Predição de Desvio

Gabriel P. Silva

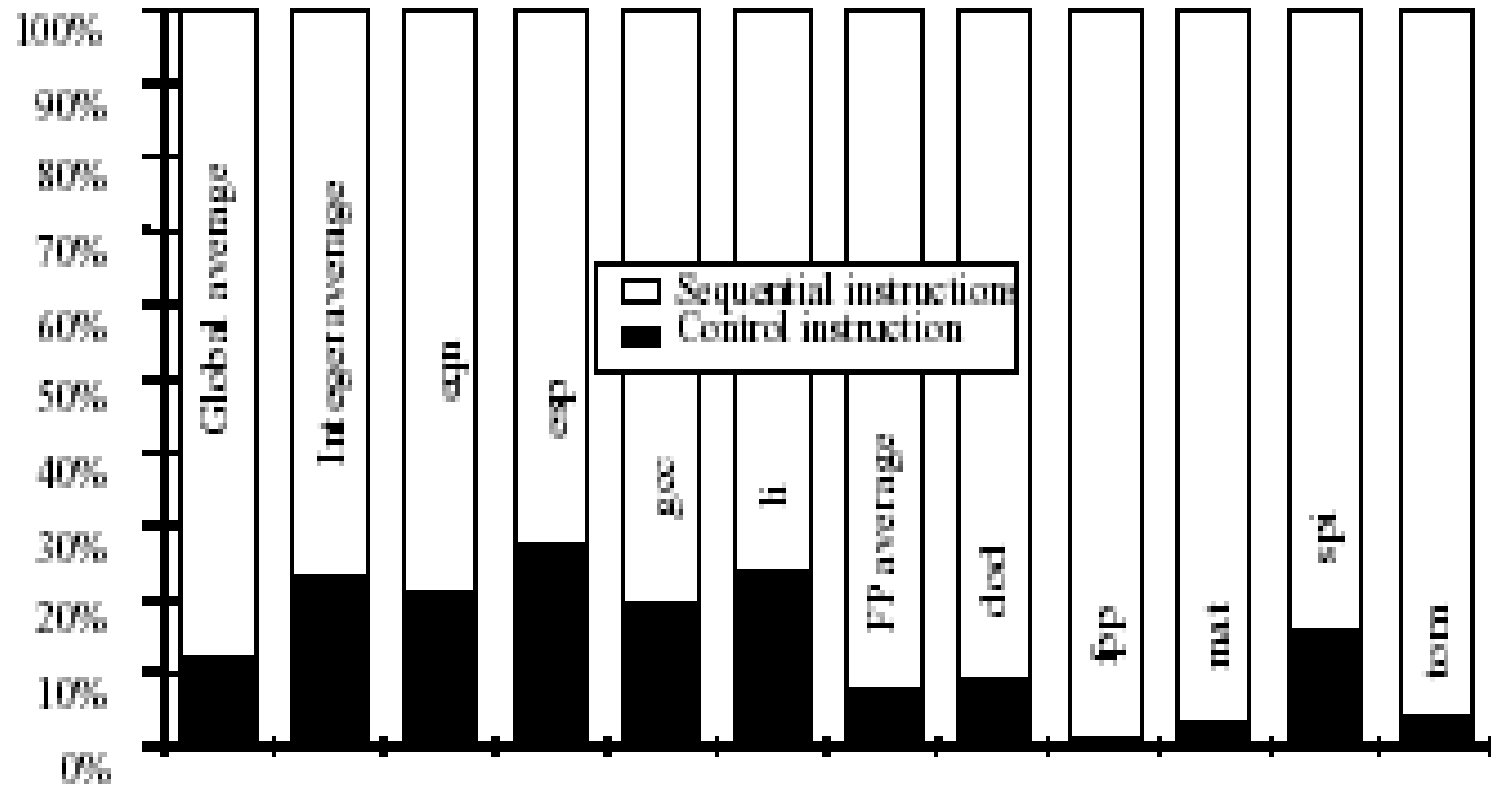
Introdução

◆ Desvios:

- Instruções que podem alterar o fluxo de execução das instruções de um programa

	Condicional	Incondicional
Direto	if - then- else for loops (bez, bnez, etc)	procedure calls (jal) goto (j)
Indireto		return (jr) virtual function lookup function pointers (jalr)

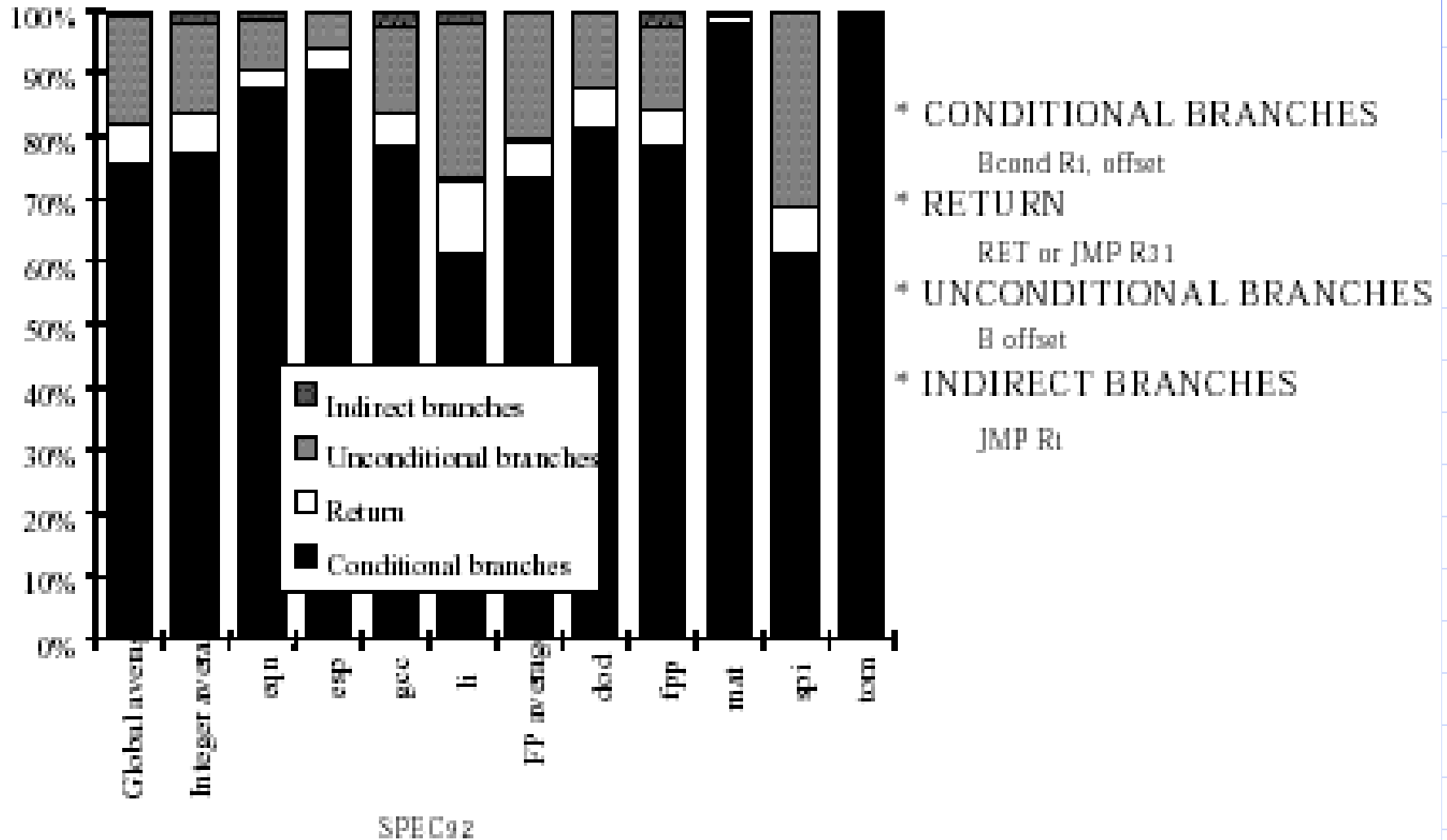
Introdução



SPEC92

Yeh and Patt

Introdução



Introdução

- ◆ Predição de desvio:
 - Aumenta o número de instruções disponíveis para o despacho.
 - Aumenta o paralelismo a nível de instrução.
 - Permite que trabalho útil seja concluído enquanto se espera pela resolução do desvio.

Predição de Desvio

- ◆ Predizer o resultado de um desvio
 - Direção:
 - ◆ Tomado / Não Tomado
 - ◆ Preditores de Direção
 - Endereço Alvo:
 - ◆ Tomado: $PC + \text{offset}$ / Não Tomado: $PC + 4$
 - ◆ Preditores de Endereço Alvo
 - Branch Target Address Cache (BTAC) ou Branch Target Buffer (BTB)

Estratégias de Predição de Desvio

- ◆ As estratégias de predição de desvio podem ser divididas em duas categorias básicas:
 - Estratégia de predição de desvio estática por meio de mecanismos de "software".
 - ◆ Decidida a priori pelo compilador.
 - Estratégia de predição de desvio dinâmica por meio de mecanismos de "hardware".
 - ◆ As decisões de predição podem ser alteradas durante a execução do programa.

Predição Estática do Desvio

- ◆ Três abordagens podem ser adotadas:
 - Assumir que todos os desvios são tomados ("branch taken");
 - Os desvios para trás são assumidos como tomados ("branch taken") e os desvios para frente são assumidos como não tomados ("branch not taken");
 - Fazer a predição com base em resultados coletados de experiências de "profile" realizadas anteriormente .

Predição Dinâmica do Desvio

- ◆ Pequena memória endereçada pela parte baixa do endereço das instruções de desvio;
- ◆ A memória contém 1 bit (bit de predição) que diz se o desvio foi tomado ou não da última vez;
- ◆ Se a predição for errada, o bit correspondente é invertido na memória
- ◆ Problemas:
 - Instruções de desvio diferentes podem mapear para uma mesma posição do buffer
 - O esquema pode falhar quando a decisão do desvio se alterna a cada execução

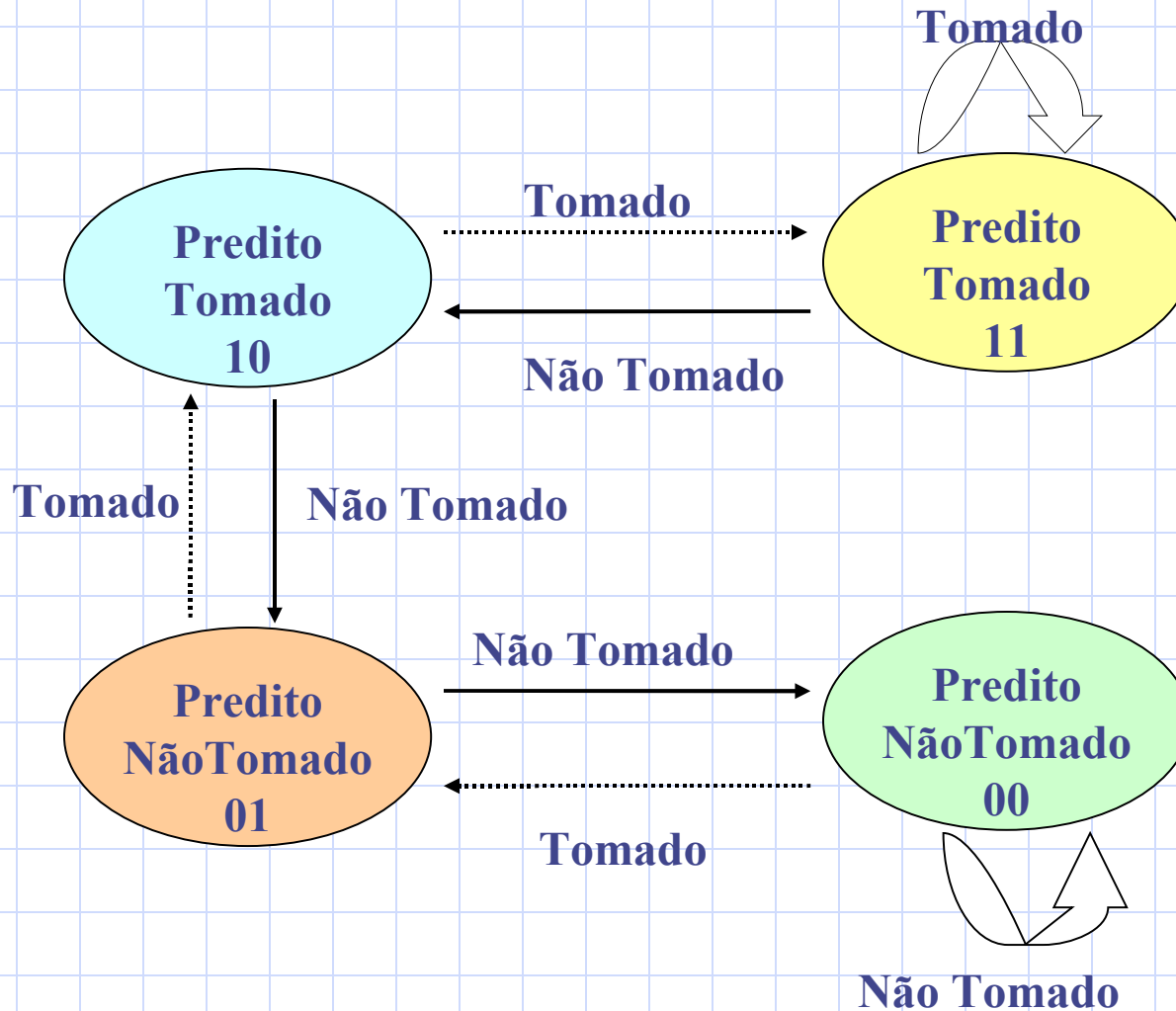
Preditores de 1-bit x 2-bits

- ◆ Um preditor de 1-bit prediz corretamente um desvio ao final de uma iteração de um loop, enquanto o loop não termina.
- ◆ Em loops aninhados, um preditor de 1-bit irá causar duas predições incorretas para o loop interno:
 - Uma vez no final do loop, quando a iteração termina o loop ao invés de ir para o começo do loop, e
 - Uma vez quando a primeira iteração do loop for reiniciada, quando ele prediz o término do loop ao invés do começo do loop.
- ◆ Este erro duplo em loops aninhados é evitado por um esquema de predição de dois bits.
- ◆ **Preditor de 2-bits:** Uma predição deve errar duas vezes antes de ser alterada quando um preditor de 2-bits é utilizado.

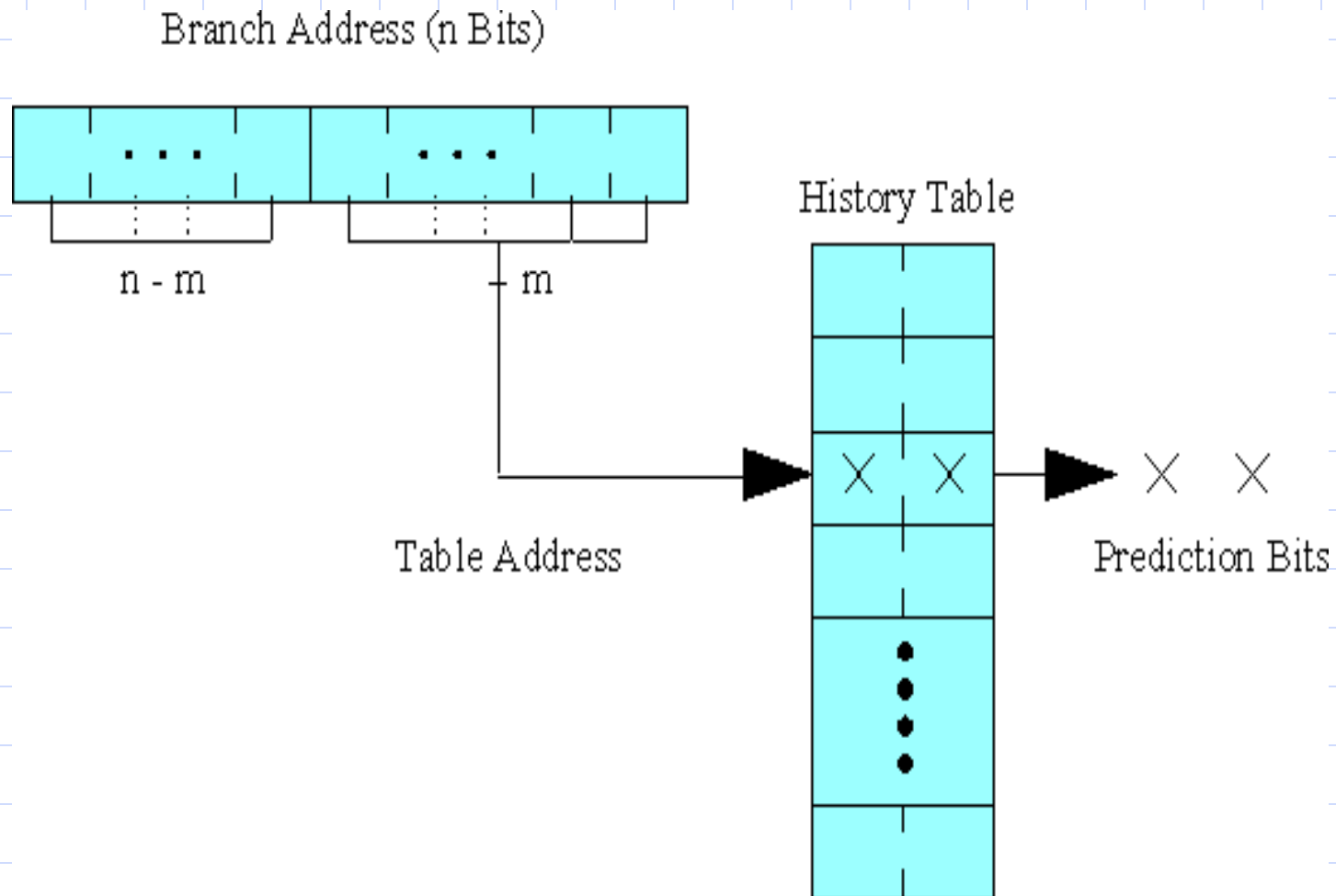
Predição de Desvio Bimodal

- ◆ O preditor de dois bits (bimodal) é essencialmente um contador de dois bits com valores entre 0 e 3.
- ◆ Quando o contador é maior ou igual ao valor 2, o desvio é predito como tomado; em caso contrário é predito como não tomado.
- ◆ O contador é incrementado em um desvio tomado e decrementado em um desvio não tomado.
- ◆ Demonstra-se que um buffer de 2 bits com 4 K entradas tem um desempenho similar que um buffer com um número infinito de entrada e usando n bits para predição nos programa de avaliação do SPEC92.

Diagrama de Estados do Preditor Bimodal



Predição Bimodal



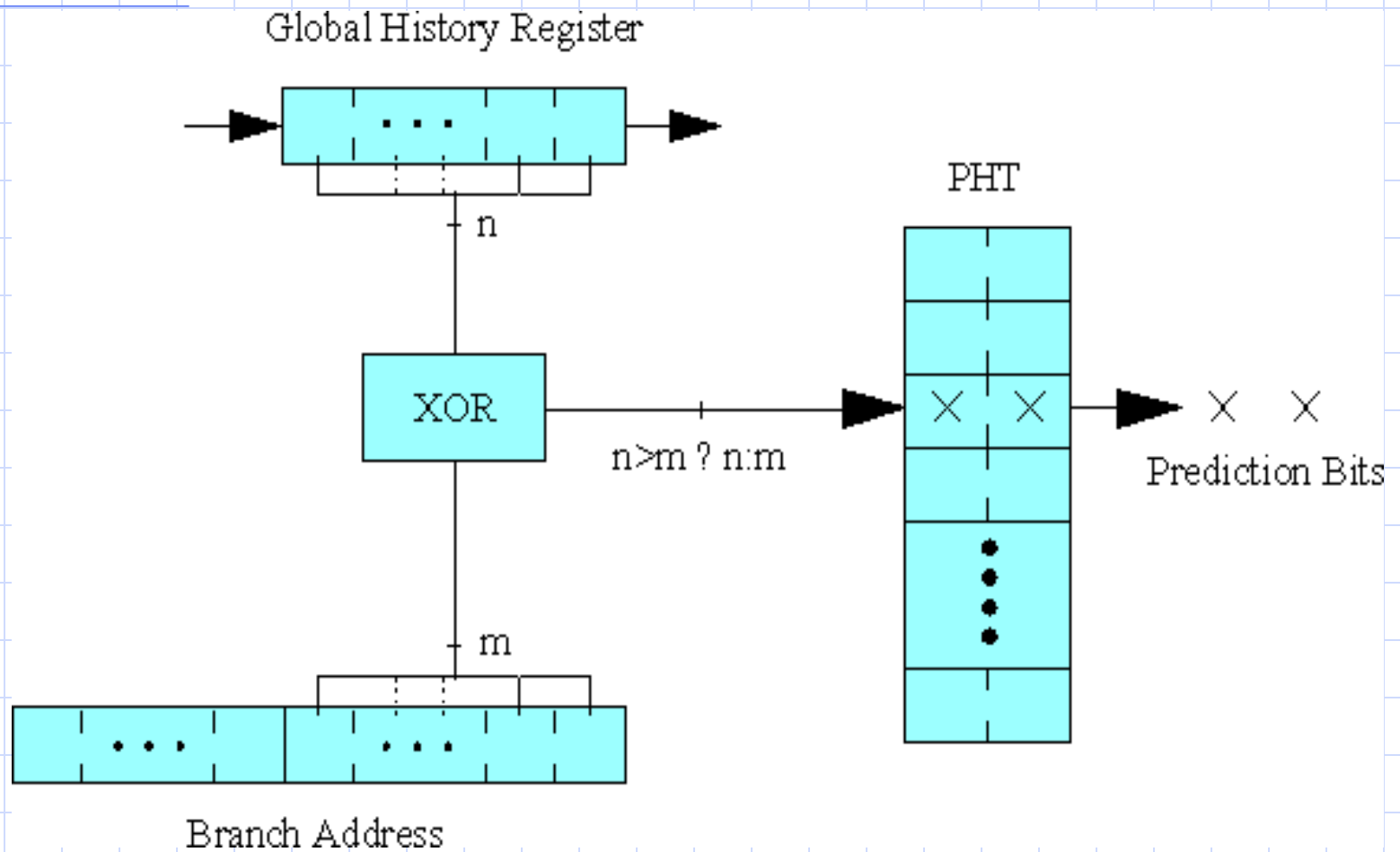
Preditor por Correlação

- ◆ O preditor de 2-bits usa apenas o comportamento mais recente de um único desvio para prever o comportamento futuro daquele desvio.
- ◆ Correlações entre diferentes instruções de desvio não são levadas em conta.
- ◆ Preditores baseados em correlação ou preditores correlacionados usam também o comportamento de outros desvios para fazer uma previsão, ao invés de apenas aquele que estamos tentando prever.
- ◆ Este raciocínio levou ao desenvolvimento dos preditores correlacionados.

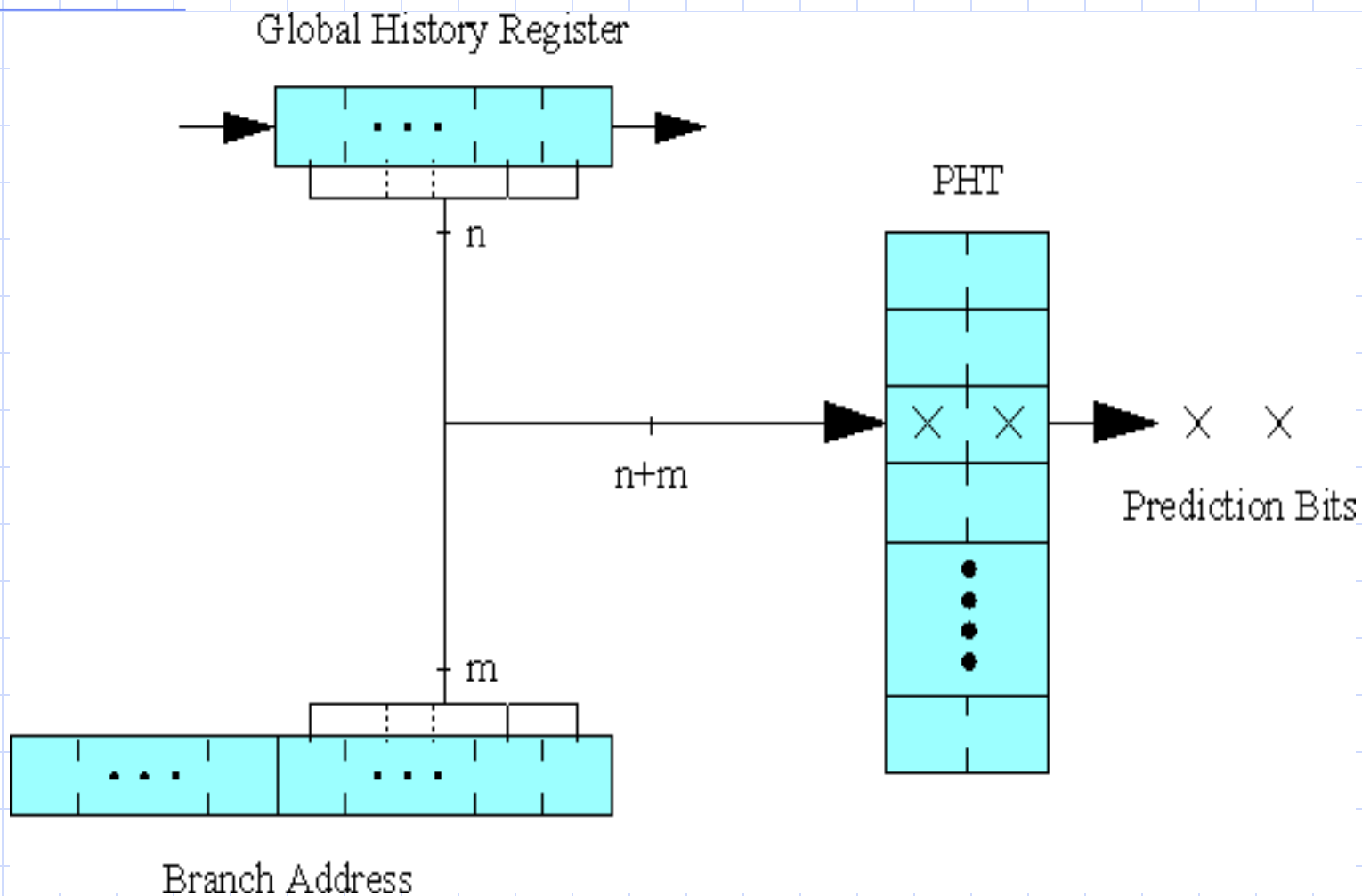
Preditor por Correlação

- ◆ Enquanto os preditores de 2-bits usam apenas o seu próprio histórico, os preditores correlacionados também usam o histórico de desvios vizinhos.
- ◆ Notação: um preditor (n,k) usa o comportamento dos últimos n desvios para escolher entre 2^n preditores de desvios, cada qual é um preditor de k -bits para um único desvio.
- ◆ Registrador de Histórico de Desvio (BHR): O histórico global dos n desvios mais recentes podem ser gravados em um registrador de deslocamento com n -bits, onde cada bit registra e o desvio foi tomado ou não.

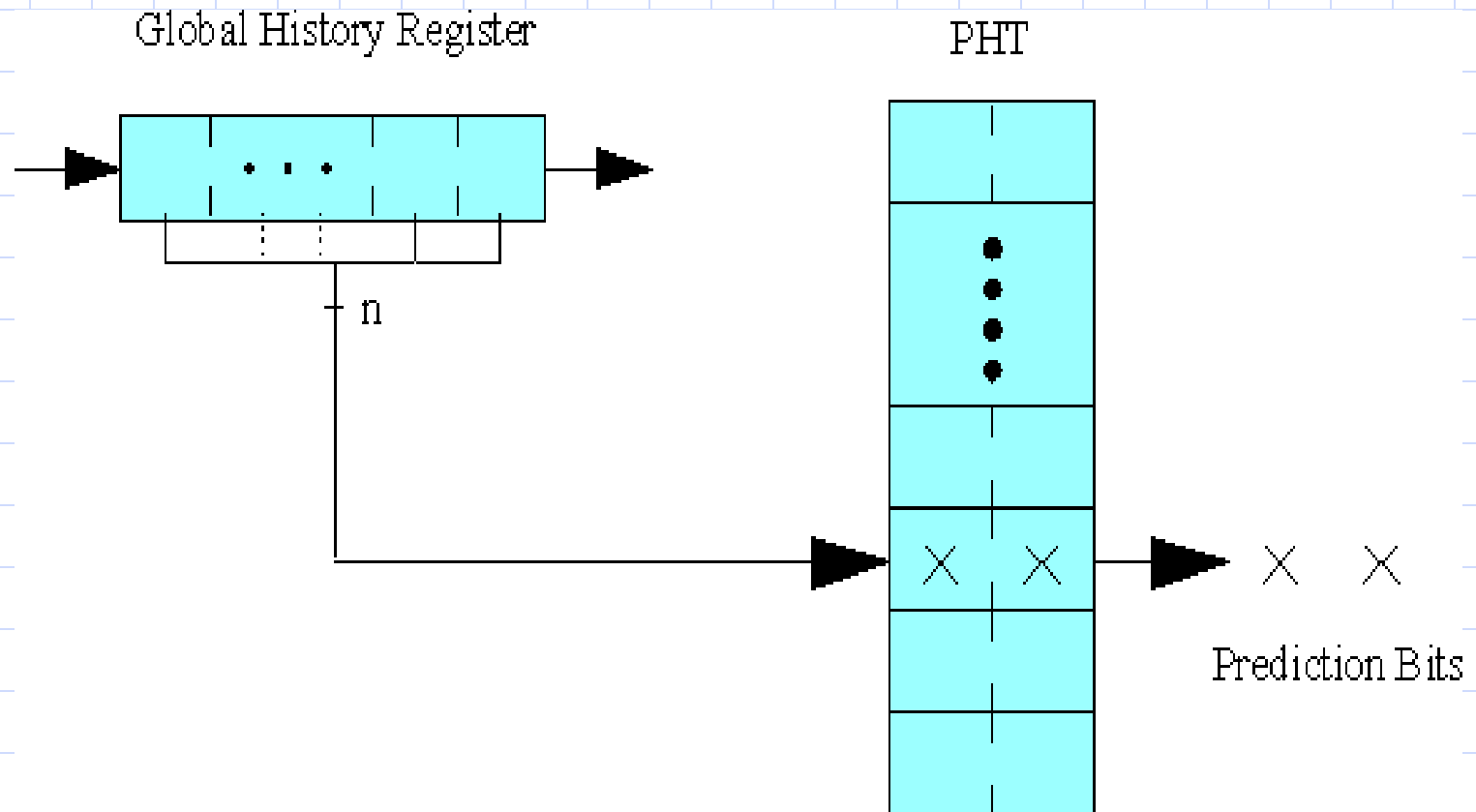
GShare



GSelect



GAg



Preditores gselect e gshare

- ◆ **Preditor gselect:** concatena alguns bits de mais baixa ordem do endereço do desvio e os bits do registrador de histórico global (BHR)
- ◆ **Preditor gshare:** Utiliza o OU exclusivo bit a bit de parte do endereço de desvio com os bits do registrador de histórico como função HASH.
- ◆ **McFarling:** gshare é ligeiramente melhor que o gselect

<u>Branch Address</u>	<u>BHR</u>	<u>gselect4/4</u>	<u>gshare8/8</u>
00000000	00000001	00000001	00000001
00000000	00000000	00000000	00000000
11111111	00000000	11110000	11111111
11111111	10000000	11110000	01111111

Fatores que Influenciam a Predição

◆ Aliasing

- Mais de um desvio pode utilizar a mesma entrada na BHT/PHT

◆ Tempo de treinamento

- É necessário haver desvios suficientes para um padrão ser descoberto
- É necessário algum tempo até se entrar em um estado estável

◆ Histórico “Errado”

- Tipo incorreto de histórico para o desvio

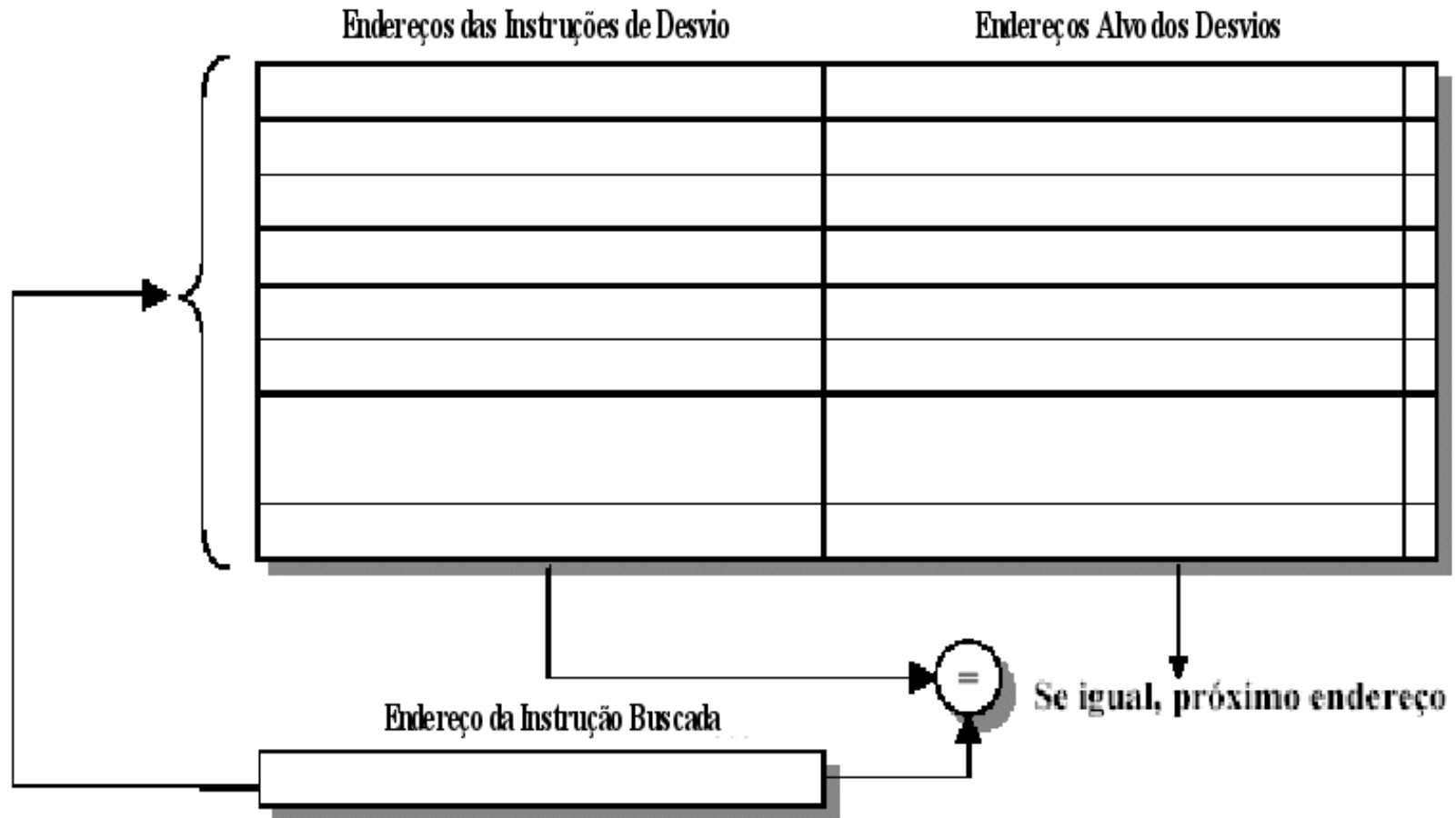
◆ Troca de contexto

- “Aliasing” causado por desvios de diferentes programas

Branch Target Cache

- ◆ Contador de programas (PC) da instrução no estágio de busca é usado para indexar a branch target cache (BTC).
- ◆ Quando ocorre um acerto na BTC, a instrução é considerada como um desvio com predição a ser tomado. Então:
 - A BTC armazena o PC correspondente à instrução alvo do desvio;
 - Esse PC é usado na busca da próxima instrução;
 - Se a predição de desvio tomado se mostrar incorreta, a instrução buscada é anulada, a operação de busca é reiniciada e a entrada correspondente na BTC é invalidada.

Branch Target Cache



Variações na Branch Target Cache

- ◆ Armazenamento da instrução alvo e não do endereço da instrução alvo:
 - Instruções de desvio gastando zero ciclos (“branch folding”) podem ser produzidas uma vez que a instrução obtida do cache pode ser utilizada para substituir a instrução de desvio no pipeline
- ◆ Predição de desvios cujo endereço varia durante a execução do programa
 - Implementação de uma pilha para a predição do endereço de retorno.

Desvios Especiais

◆ Chamadas e Retorno de Procedimento

- Chamadas são sempre tomadas
- O endereço de retorno quase sempre é conhecido

◆ Return Address Stack (RAS)

- Na chamada de um procedimento, colocar o endereço da instrução seguinte na pilha de endereço de retorno
- Utilizado em estruturas de "case" e retorno de rotinas