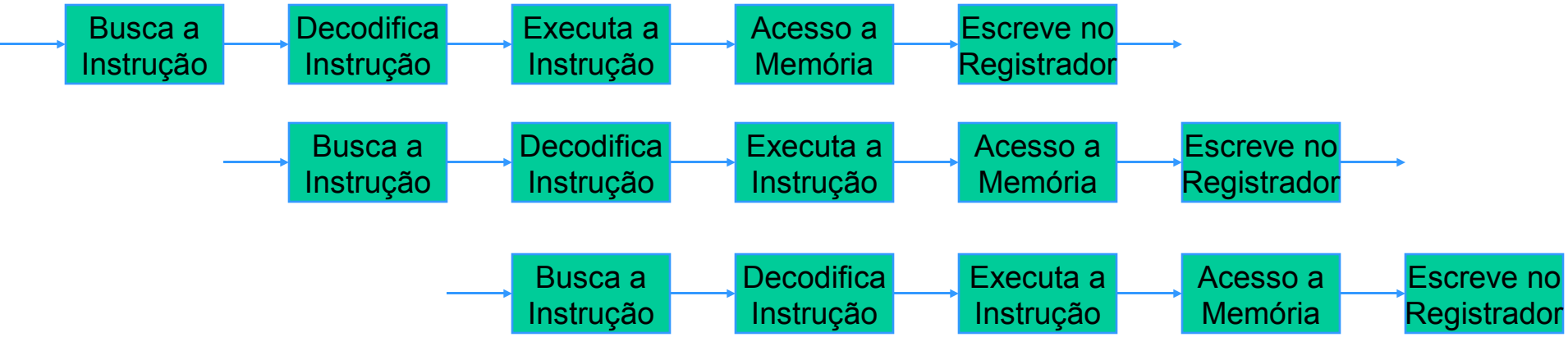
Previsão de Desvios

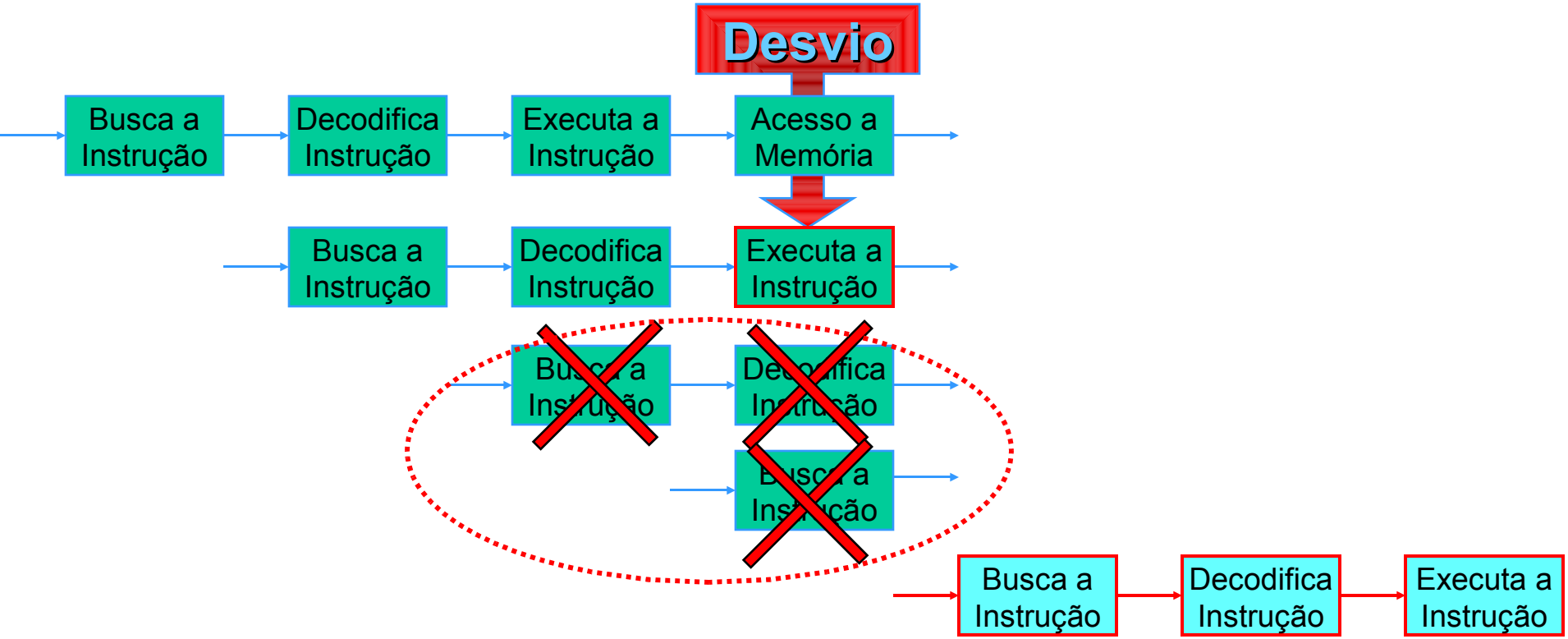
Regina Helena Carlucci Santana

Pipeline



- **Utiliza melhor o processador**

Pipeline



Problemas:

Importância

- Os processadores utilizam-se cada vez mais de pipelines:
- O pipeline melhora a performance dos processadores;
- O pipeline vem tornando-se mais longos;
- Pipelines Longos ->Bolhas Maiores -> Queda na Performance;

Importância

- A latencia para resolver um desvio não diminui;
- As arquiteturas superescalares são mais afetadas que as escalares (desvios chegam mais frequentemente).

Previsão de Desvios é Fundamental !

Efeitos da previsão de desvios

- Precisão
  - Impacta significativamente o desempenho
  - Se não for preciso, desempenho pode ser pior que não ter previsão
- Dois aspectos a serem analisados:
  - Desvio pode ou não ter sido considerado
  - Desvio pode ser correto ou incorreto
- Quatro tipos de latência

Efeitos da previsão de desvios

- **Desvio não considerado:**
  - Desvio não previsto;
  - Causado pela demora para identificar uma instrução como um desvio e para calcular o endereço destino.

- **Erro na Previsão:**
  - O processador assume que a desvio pode ser previsto;
  - Erro na previsão do destino ou direção do desvio;
  - Instruções buscadas em um curso de instruções incorretos devem ser descartadas;

Dinâmicos X Estáticos

- Previsão de desvios:
  - Estáticas – todas as decisões em tempo de compilação
    - Não permite adaptações com relação ao comportamento do programa
  - Dinâmicas - Decisões em tempo de execução

Estratégias Estáticas

Fazem sempre a mesma previsão toda vez que uma instrução de desvio é encontrada

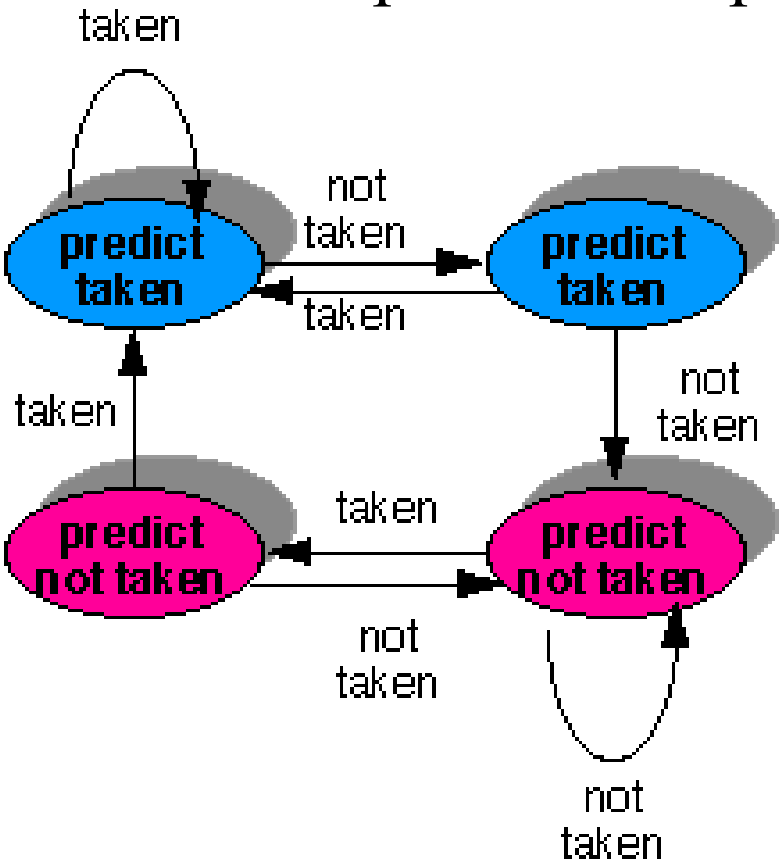
- Prevê que todos os desvios serão executados.
- Prevê que todos os desvios com determinado ‘operation code’ serão executados.
- Backward-taken, forward not-taken (BTFNT).

Estratégias Dinâmicas

- Mantém um buffer indexado com parte do endereço da instrução de desvio (**Branch-Prediction Buffer)**
- O conteúdo do buffer indica se o desvio foi ou não considerado
- Bit que indica se o próximo desvio deve ou não ser considerado
- Em caso de erros esse bit é alterado
- Não mantém histórico
- Quando o caminho seguido é sempre o mesmo com exceção da última execução (loop), ocorrerão dois erros.

Branch-Prediction Buffer

- Dois bits para decisão
  - Melhora a precisão da previsão



90% de acerto

Branch-Prediction Buffer

- N-bits para decisão
  - Incrementa a cada acerto e decrementa em caso de erro
  - Se maior que valor estipulado, considera o desvio
  - N >> 2 não apresenta melhoras significativas

Localização do buffer de previsão

- Cache Especial
  - O cache é acessado durante a Busca de Instrução e os bits de previsão usados durante a Decodificação da instrução (se a instrução for uma decisão)
- Cache de Instrução
  - Os bits de decisão são armazenados junto com as instruções que estão no cache

Precisão da Previsão

- Utilizando um cache de 4kbytes
- Dois bits de decisão
- Executando o SPEC89
  - Obtém-se de 1 a 18% de erro
- Previsão estática – 30% de erro

Precisão da Previsão

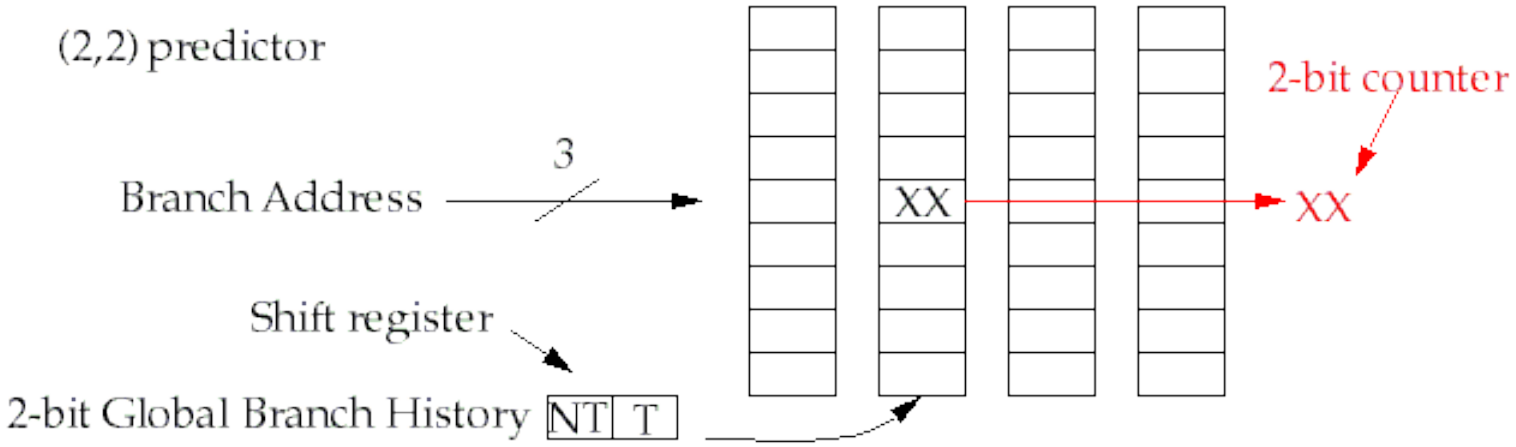
- Como melhorar:
  - Aumento do Cache – limitado!
  - Aumento do número de bits – limitado!
  - Verificar correlação com desvios vizinhos

Previsores de Desvio Com Relacionamento

- Considera a decisão do desvio anterior
- Previsor (1,1)
  - Possui 2 bits:
  - Desvio Anterior - Desviou ou não
  - Se desviou, considero bit 1 senão considero bit 2

Previsores de Desvio Com Relacionamento

- Previsor (2,2)



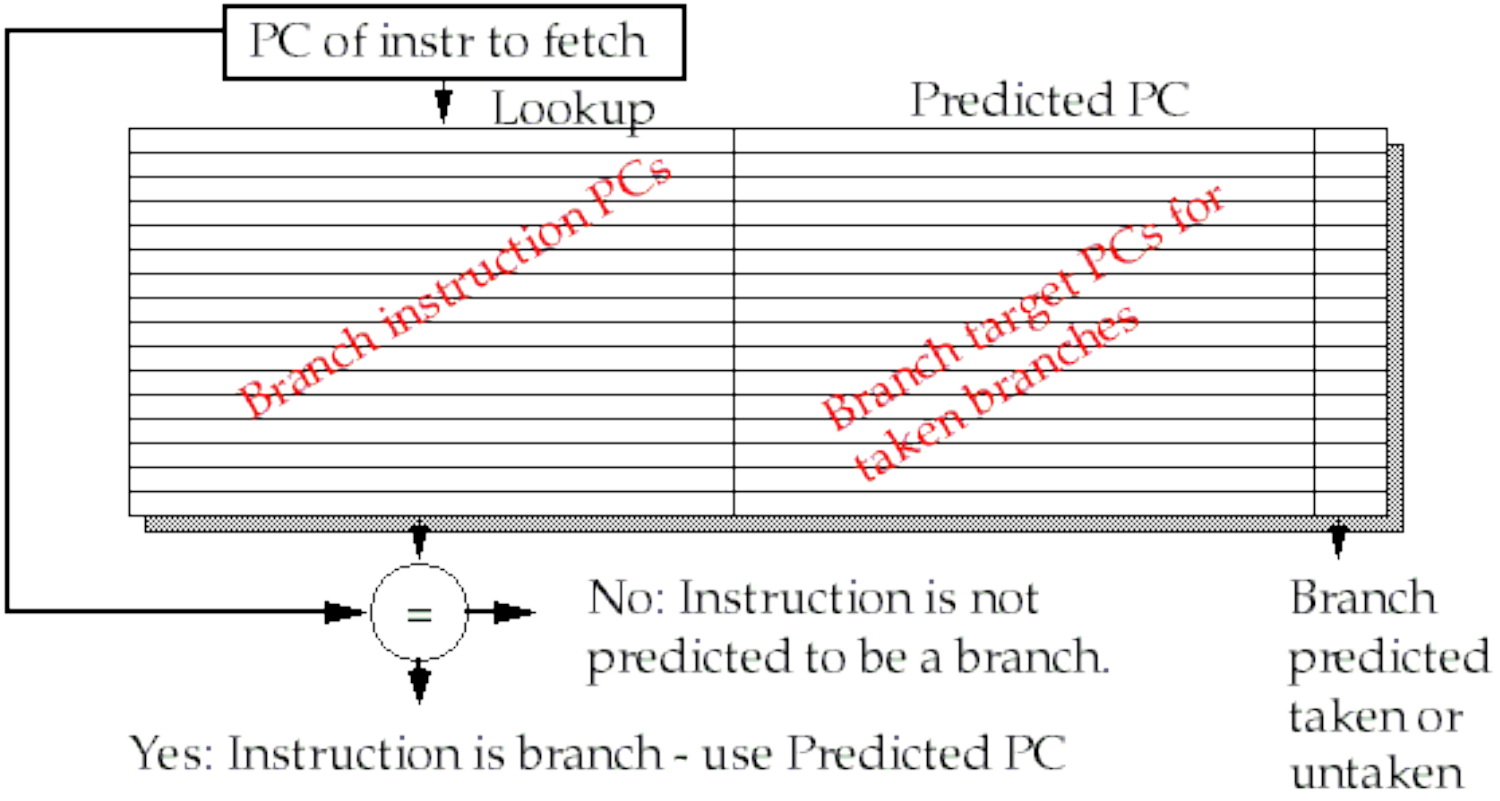
Previsores de Desvio Com Relacionamento

- Previsor (n,m)
  - Considera os n desvios anteriores
  - Para cada combinação tem-se m bits de previsão
  - Possui 2^n bits

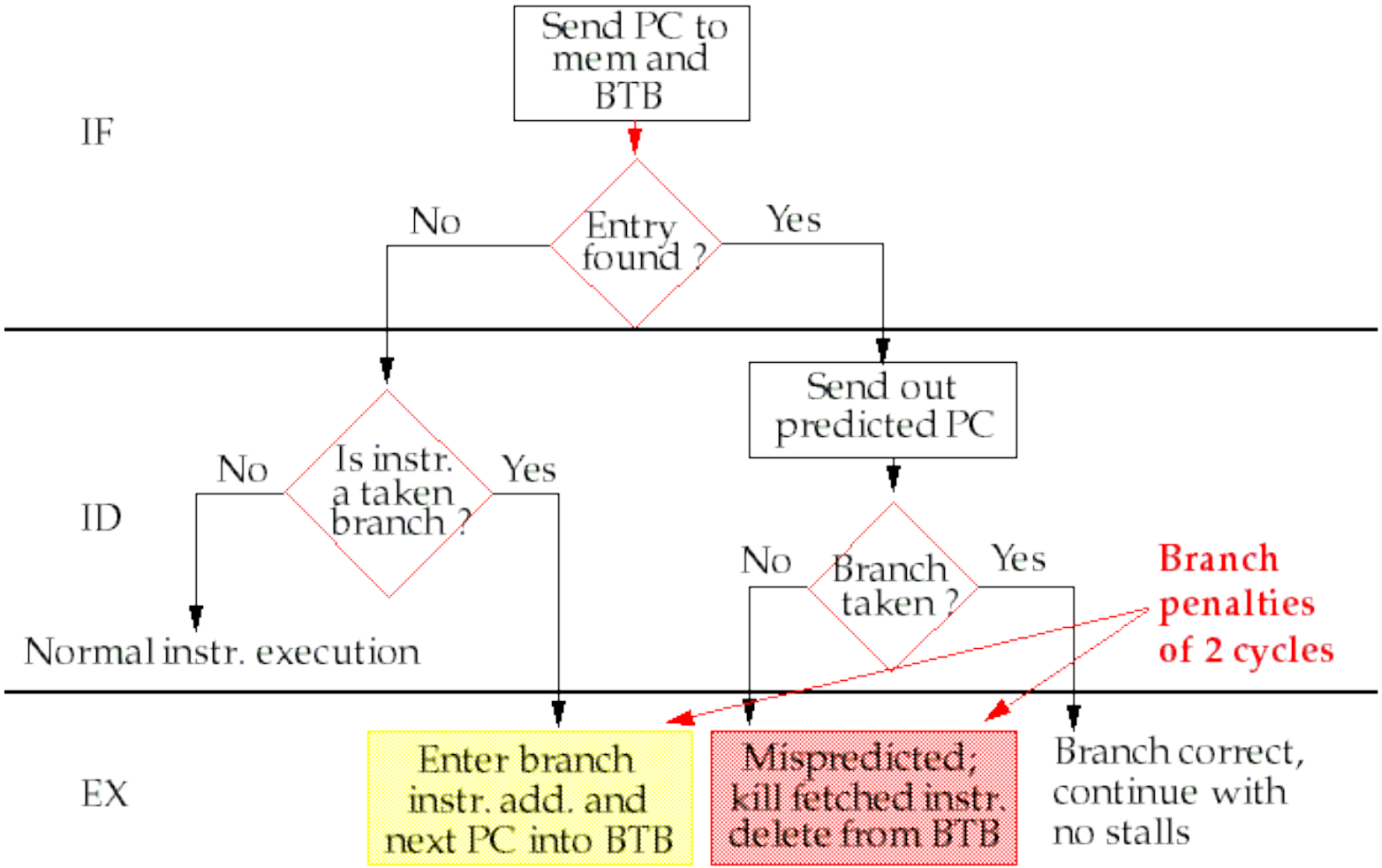
Buffer de Destino de Desvio

- A instrução de desvio só é identificada quando for decodificada
- Nesse ponto já perdeu-se um ciclo
- Utiliza-se um cache de instruções de desvios
- Durante busca de instrução já se sabe se é desvio

Buffer de Destino de Desvio



Buffer de Destino de Desvio

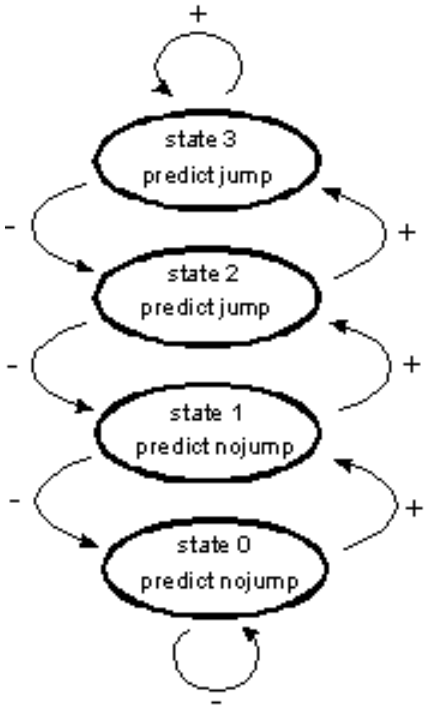


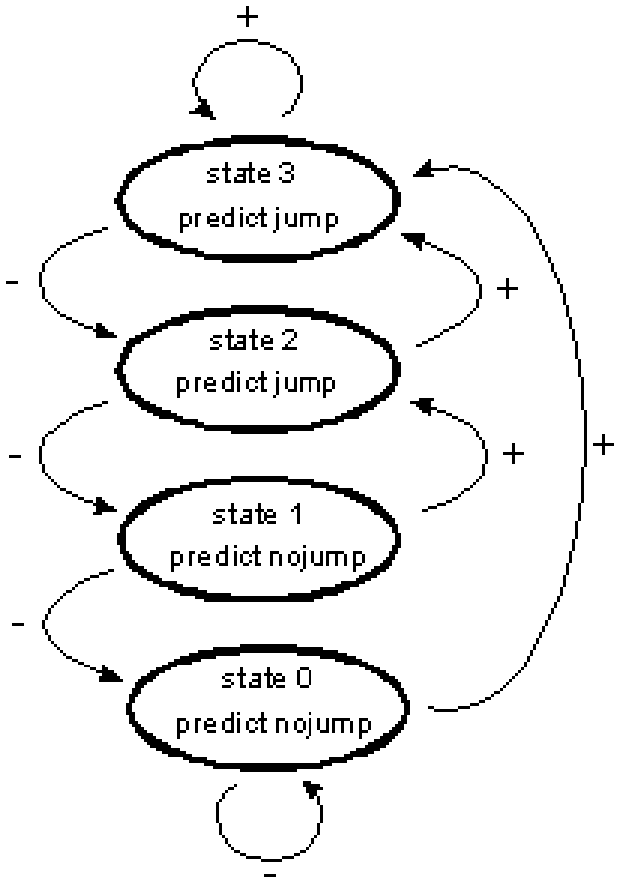
Previsão de Endereços de Retorno

- Buffer de endereços de retorno operando como uma pilha
- Nos programas atuais, maioria de desvios incondicionais são retornos de subrotinas
- Desempenho de previsão nesses casos é baixo
- Rotinas são chamadas de diversos pontos do programa

Exemplos - Pentium

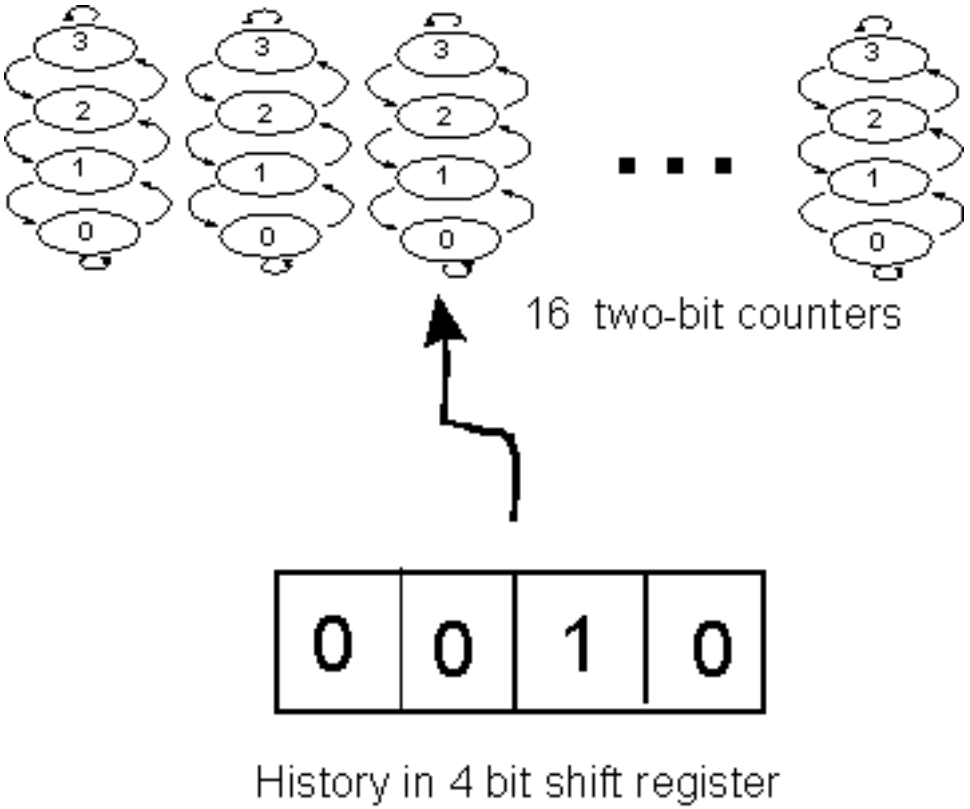
- Primeiros Pentiuns





Exemplos - Pentium

- Pentium MMX, Pentium PRO, Pentium II e III



Exemplos - Pentium

Pentium 4

- Previsão Estática e Dinâmica
- Buffer com 4K entradas - BHT (Branch History Table)
- Se instrução de desvio não está no buffer usa previsão estática
- Se está usa dinâmica - BTB (Branch Target Buffer)
- Taxa de acerto de 91%

Exemplos - AMD - K6

- Desvios Incondicionais são manipulados diretamente

- Desvios Condicionais usam Tabela de Histórico dos Desvios
  - Não guarda os endereços fontes previstos.
  - Endereços são calculados com ULAs no estagio de decodificação

Exemplos - PowerPC

- PowerPC 601 e 603
  - Desvios incondicionais simples -> processa e remove o desvio antes da instrução terminar (*branch folding)*.
  - Desvios Condicionais -> BTFNT (Backward Taken, Forward Not Taken).

Exemplos - PowerPC

- PowerPC 604
  - Rastreia dinamicamente os desvios sob suposição que um desvio que foi recentemente verdadeiro será verdadeiro na próxima vez que for encontrado.
  - Primeiro processador PowerPC que pode de forma especulativa executar dois desvios de uma vez.

Henessy e Paterson – Arquitetura de Computadores – Uma Abordagem Quantitativa

http://www.x86.org/articles/branch/branchprediction.htm#fib1a

http://www.csee.umbc.edu/~plusquel/611/slides/chap4_5.html