

SSC0510

Arquitetura de Computadores

9ª Aula – Pipeline

Profa. Sarita Mazzini Bruschi

sarita@icmc.usp.br

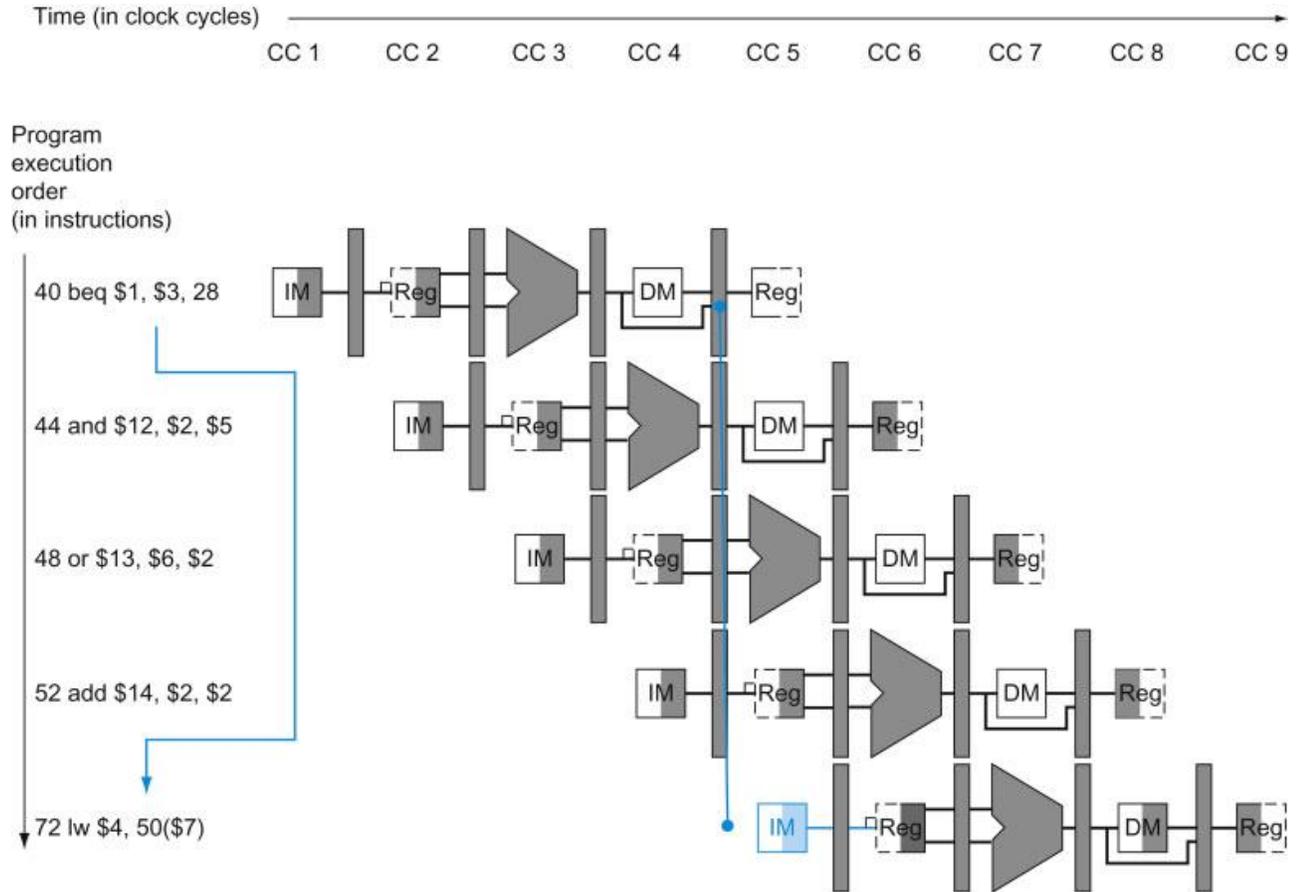
Pipeline

- Dependências ou Conflitos (*Hazards*)
 - Conflitos Estruturais
 - Pode haver acessos simultâneos à memória feitos por 2 ou mais estágios.
 - Dependências de Dados
 - As instruções dependem de resultados de instruções anteriores, ainda não completadas.
 - Dependências de Controle
 - A próxima instrução não está no endereço subsequente ao da instrução anterior.

Pipeline

Conflitos de Controle

- Efeito dos desvios condicionais
 - Se o desvio ocorre, pipeline precisa ser esvaziado
 - Não se sabe se o desvio ocorrerá ou não até o momento de sua execução



Pipeline

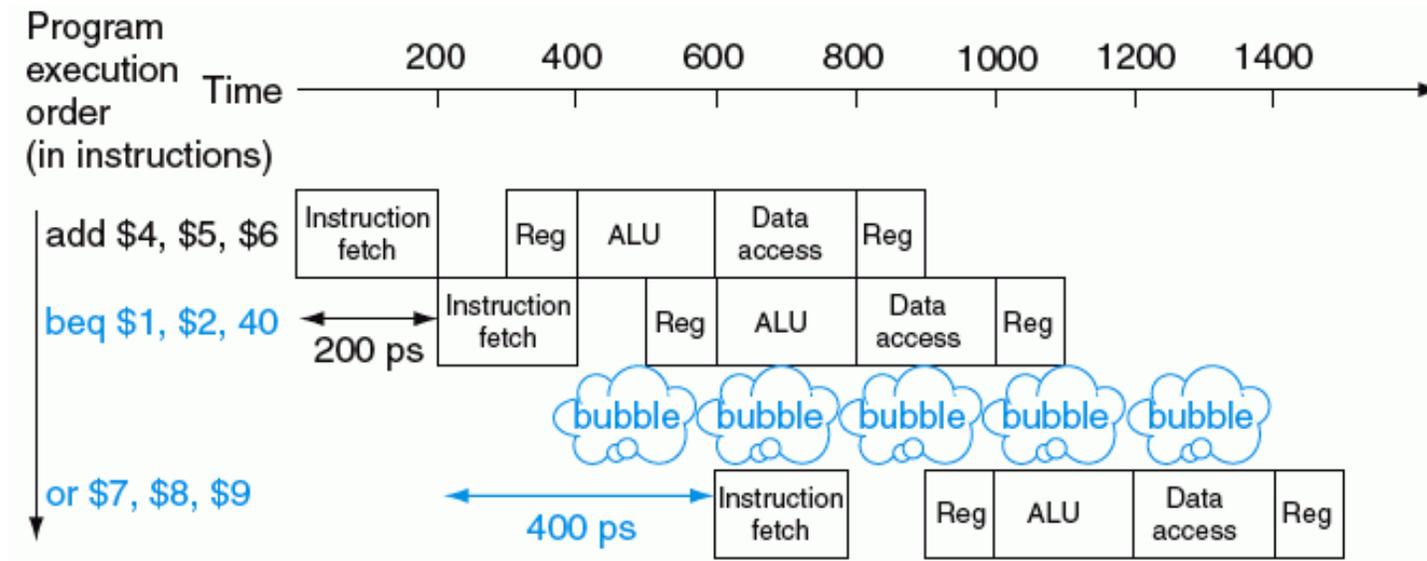
Conflitos de Controle

- Maneiras de se tratar as conseqüências de uma parada no pipeline por causa dos desvios:
 - Previsão Estática de Desvios
 - O compilador faz uma previsão se o desvio vai ser tomado ou não
 - Geração de “bolhas” quando a previsão é errada, baixa taxa de acertos
 - Não permite adaptações com relação ao comportamento do programa
 - Outra opção é um hardware para inserir uma bolha
 - Previsão Dinâmica de Desvios
 - Existem mecanismos em “hardware” que fazem a previsão baseada no comportamento daquele desvio no passado
 - Alta taxa de acertos

Pipeline

Conflitos de Controle

- Previsão estática
 - Congelar o pipeline até que o resultado do desvio seja conhecido
 - Inserir “bolhas” no pipeline
 - Solução ruim quando o pipeline é muito longo



Pipeline

Conflitos de Controle

- Congelar o pipeline pode ser muito lento
- Outra opção é considerar que o desvio não será tomado e portanto, vai continuar no fluxo sequencial
- Se o desvio for tomado, as instruções buscadas e decodificadas precisam ser descartadas
 - Para descartar, basta alterar os valores de controle para 0, como foi feito no tratamento da dependência de dados
 - Como a decisão do desvio é feita somente no estágio MEM, é necessário alterar as instruções nos estágios IF, ID e EX -> **dar flush nas instruções**

Pipeline

Conflitos de Controle

- Previsão Estática (cont.)
 - Assumir que todos os desvios não são executados (*predicted-untaken*)
 - Assumir que todos os desvio são tomados (*predicted-taken*)
 - Assumir que todos os desvios com determinado *operation code* serão tomados
 - Os desvios para trás são assumidos como tomados (*branch taken*) e os desvios para frente são assumidos como não tomados (*branch not taken*) – *Backward-taken, forward not-taken* (BTFNT)

Pipeline

Conflitos de Controle

- Reduzindo o atraso dos desvios
 - Mover a execução do desvio para um estágio anterior a MEM
 - Calcular o endereço de destino
 - Avaliar a decisão de desvio

- Exemplo

36 sub \$10, \$4, \$8

40 beq \$1, \$3, 7

44 and \$12, \$2, \$5

48 or \$13, \$2, \$6

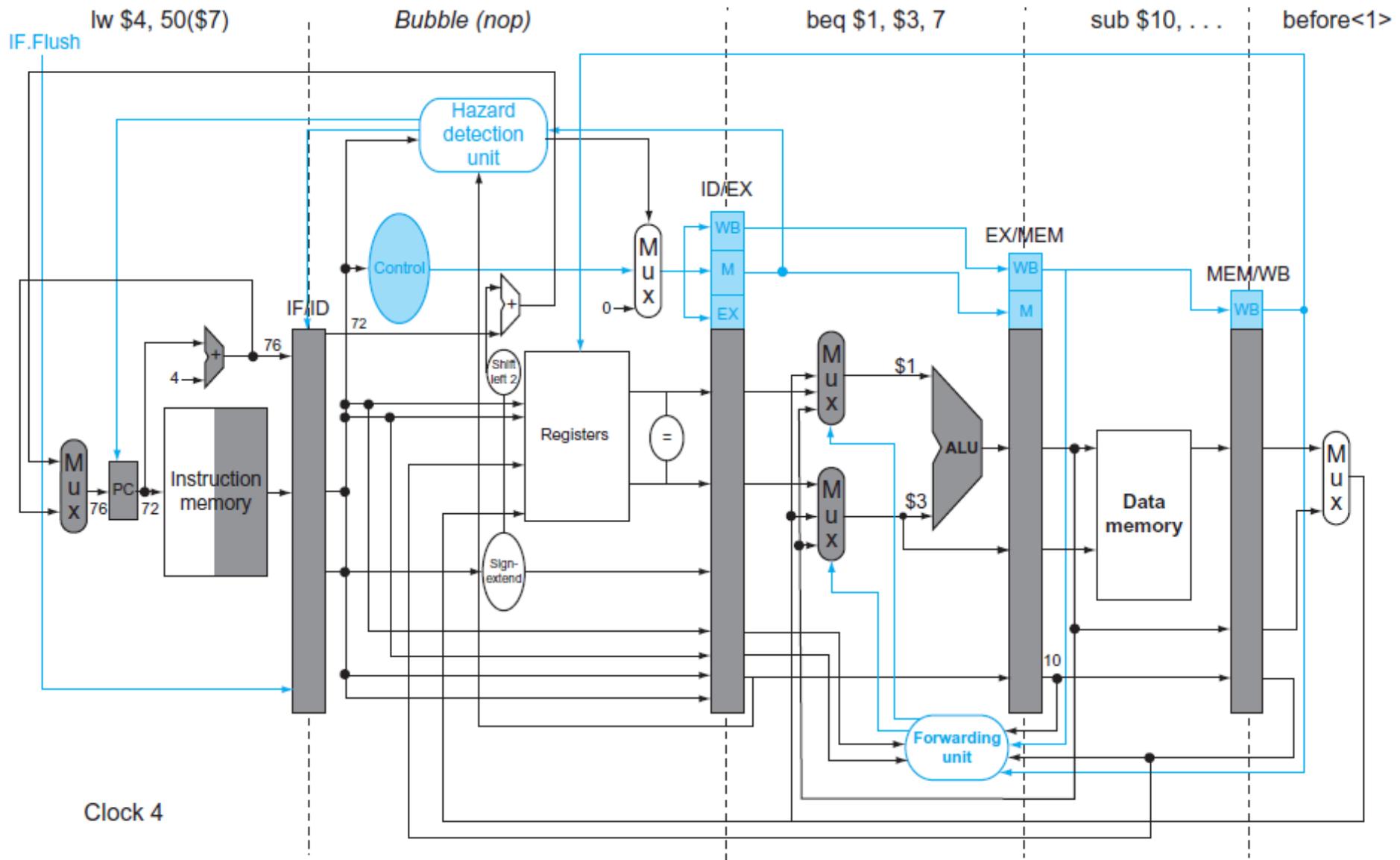
52 add \$14, \$4, \$2

56 slt \$15, \$6, \$7

...

72 lw \$4, 50(\$7)

PC-relative branch to 40 + 4 + 7 * 4 = 72



Pipeline

Conflitos de Controle

- Previsão Estática (cont.)
 - Utilização de *Delayed Branch* (desvio atrasado) para instruções de desvio
 - Instrução após o desvio é sempre executada
 - A próxima instrução é chamada de *delay slot* (posição de atraso)
 - Reordenamento das instruções (desvio atrasado otimizado)

Pipeline

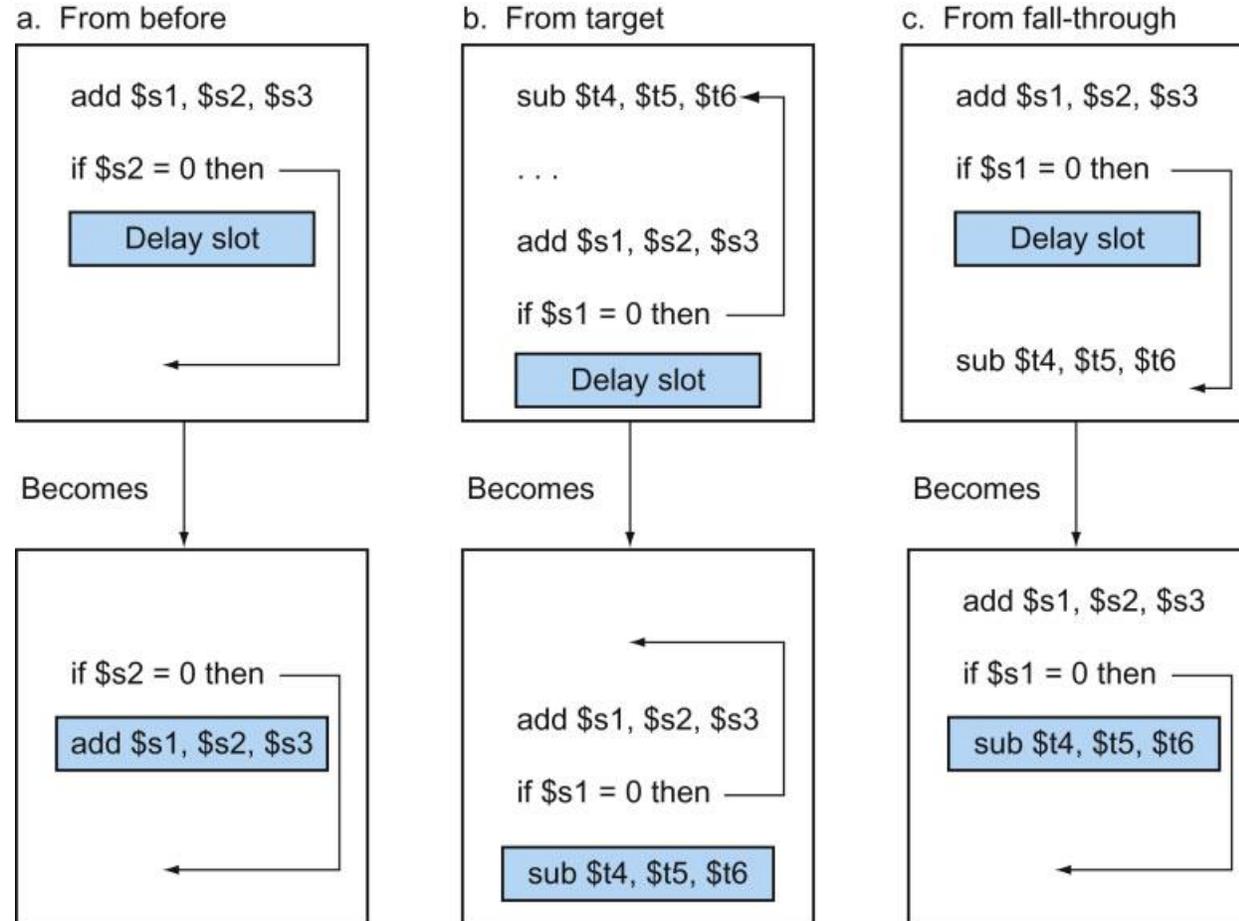
Conflitos de Controle

- Preenchimento do *Delay Slot*
 - Para facilitar o trabalho do compilador no preenchimento do *delay slot* muitas arquiteturas permitem o uso do *delay slot* com a opção de anulação automática dessa instrução se o desvio condicional não for tomado
 - Desse modo, uma instrução do endereço alvo pode ser movida para o *delay slot*, o que é muito útil no caso de loops. Nesse caso, está implícita uma previsão de desvio estática que diz que o desvio será sempre tomado

Pipeline

Conflitos de Controle

- Preenchimento do *Delay Slot* com reordenamento das instruções



Pipeline

Conflitos de Controle

- Previsão dinâmica
 - Previsor de 1 bit
 - Mantém uma pequena memória indexada pela parte menos significativa do endereço da instrução de desvio
 - Buffer de previsão de desvios (*Branch-Prediction Buffer*) ou Tabela de histórico de desvios
 - Contém 1 bit (previsor de 1 bit) que diz se o desvio foi tomado recentemente ou não e que indica se o próximo desvio deve ou não ser considerado
 - Em caso de erros esse bit é alterado
 - Não mantém histórico

Pipeline

Conflitos de Controle

- Previsão Dinâmica

- Previsor de 1 bit

- Um previsor de 1-bit prediz corretamente um desvio ao final de uma iteração de um loop, enquanto o loop não termina.
 - Em loops aninhados, um previsor de 1-bit irá causar duas previsões 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 previsão de dois bits.
 - Previsor de 2-bits: Uma previsão deve errar duas vezes antes de ser alterada quando um previsor de 2-bits é utilizado.

Pipeline

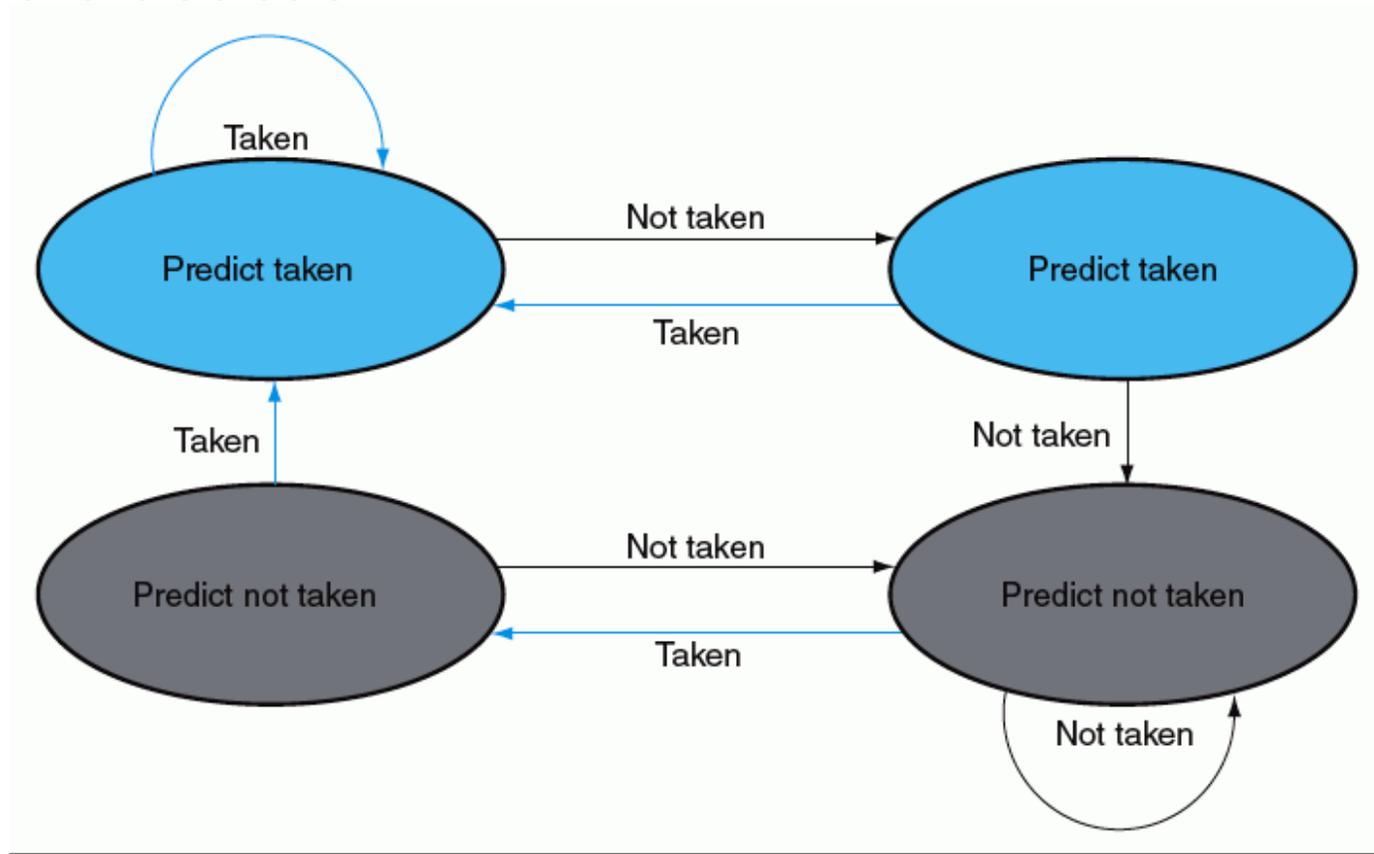
Conflitos de Controle

- Previsão dinâmica
 - Previsor de 2 bits
 - O previsor de 2-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

Pipeline

Conflitos de Controle

- Dois bits para decisão



Pipeline

Conflitos de Controle

- Buffer de Destino de Desvio
 - *Branch Target Buffer* (BTB)
 - A instrução de desvio só é identificada quando for decodificada
 - Nesse ponto já perdeu-se um ciclo
 - Utiliza-se uma cache de instruções de desvios
 - Durante busca de instrução já se sabe se é desvio

Pipeline

Conflitos de Controle

- Buffer de Destino de Desvio

