

UNIVERSIDADE FEDERAL DE LAVRAS

Departamento de Ciência da Computação

Bacharelado em Ciência da Computação

Disciplina: **Arquitetura de Computadores II – GCC123**

2º Semestre de 2019 - Professor: **Luiz Henrique A. Correia**

Trabalho Prático – Valor: 30 pontos

O objetivo deste trabalho é desenvolver o aprendizado sobre preditores de desvio. Serão estudados, implementados e avaliados os seguintes tipos de preditores de desvio: Preditor local de desvios (BHT), Preditor Global (GHT) e Preditor Híbrido. Utilizando-se de linguagens de programação será desenvolvido um programa para **simular** o comportamento dos preditores de desvios.

A descrição deste trabalho é mesma do trabalho solicitado pelo Prof. E. F. Gehringer para o curso de Computer Design & Technology da North Caroline University. Foram realizadas pequenas modificações da descrição original.

Metodologia

Os grupos serão formados por **exatamente 5 alunos**, salvo casos omissos.

O simulador deve ser desenvolvido para apresentação na Web em linguagem e formato da preferência dos alunos. Dessa forma, é importante que o simulador mostre de maneira gráfica todas as **etapas** envolvidas durante a predição.

Especificação do trabalho:

Neste trabalho serão modelados e implementado: Preditor Local de desvios (BHT), Preditor Global de desvios (GHT) e o Preditor Híbrido.

1. Os preditor local BHT pode ser modelado como um preditor de parâmetros (m, n) , onde:

m - é o número de bits do PC de mais baixa ordem (LSB) usados para indexar a tabela de histórico local (branch-history table).

n - número de bits armazenados no registro da tabela de histórico local.

O diagrama do preditor de desvio local é mostrado na Figura 1, tal que:

- A tabela de registros do desvio local é indexada pelos m bits de mais baixa ordem do endereço da instrução de desvio (isto é, o valor do PC quando o desvio é buscado (IF), não o destino do desvio). Esses m bits são os bits menos significativos do endereço, e não incluem os dois bits menos significativos do PC, pois estes são sempre zero (ou seja, use os $m + 1$ a 2 bits do PC).

- Um valor armazenado no registrador em um histórico de desvios é usado para indexar um *array* de contadores. Logo, se n é o número de bits armazenados no registro de histórico de desvios, então existirão 2^n contadores.
- Os registradores do histórico de desvios são todos inicializados em zero no início da simulação.
- Sempre que um desvio é predito, o resultado do branch é deslocado para a posição do bit mais significativo do registrador do histórico de desvio que contém o histórico desse branch (Observe que no hardware real, não poderíamos mudar o resultado do branch até o estágio EX do pipeline. No entanto, isso é apenas uma simulação e lemos o resultado do branch no arquivo de *trace*, para que possamos usá-lo imediatamente.).
- Não há tags e nenhuma verificação de tags (sem sinais de falha para o preditor).
- Cada um dos 2^n contadores contém 2 bits.
 - (a) O contador é incrementado quando o desvio é Tomado e decrementado quando é Não-Tomado.
 - (b) O contador satura em 0 e 3.
 - (c) Se o valor do contador for maior ou igual a dois, o desvio é predito como tomado, caso contrário, é predito como Não-tomado.
 - (d) Todos os contadores devem ser setados para 2 quando a simulação começar.

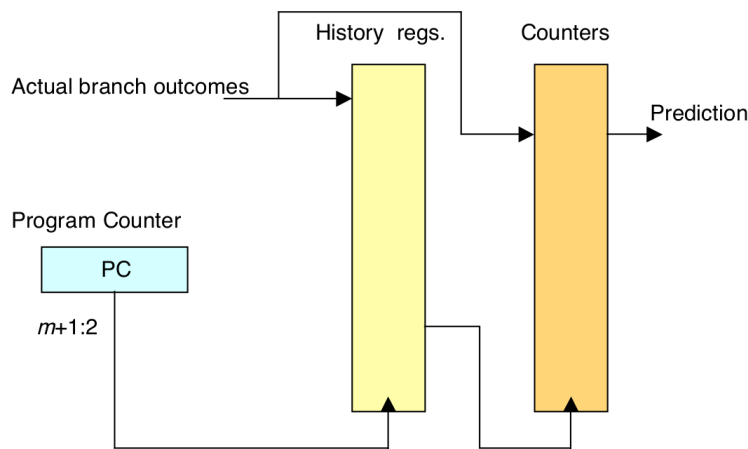


Figure 1: Diagrama do preditor local de desvios.

2. O preditor de desvios global, ou GHT, é modelado com o parâmetro n , onde:
 - n - é o número de bits do registrador de histórico global de desvios.

O diagrama do preditor de desvio Global é mostrado na Figura 2, tal que:

- Sempre que um desvio é predito, o resultado do desvio mais recente é deslocado para a posição do bit mais significativo do registrador global de histórico de desvios para um branch.
- O registrador do histórico global é inicializado com zero no início da simulação.
- Não há tags e nenhuma verificação de tags (sem sinais de falha para o preditor).
- Cada um dos 2^n contadores contém 2 bits:
 - (a) O contador é incrementado quando o desvio é Tomado e decrementado quando é Não-Tomado.
 - (b) O contador satura em 0 e 3 (o contador nunca ficará abaixo de zero ou acima de 3).
 - (c) O contador é atualizado depois que a predição é feita.
 - (d) Se o valor do contador é maior ou igual a dois, o branch é predito como Tomado, caso contrário será predito com Não-tomado.
 - (e) Todos os contadores devem ser setados para 2 quando a simulação começa.

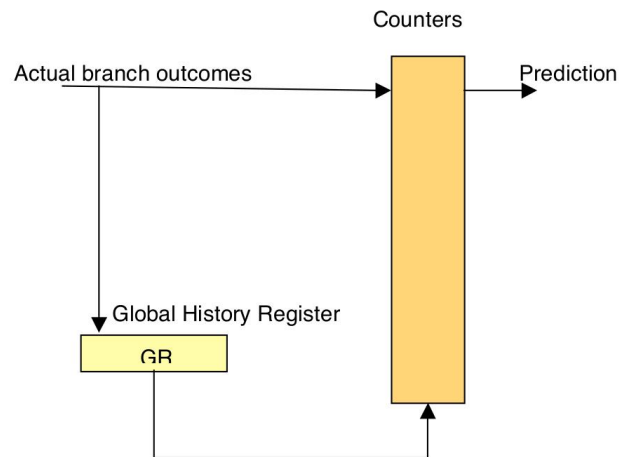


Figure 2: Diagrama do preditor global de desvios.

3. O preditor Híbrido seleciona entre os preditores local e global, usando um preditor de escolha, tal que:
 - O preditor de escolha é uma tabela de 2^k contadores de dois bits.
 - Se a escolha do preditor é 2 ou 3, ele seleciona o preditor global, caso contrário ele seleciona o preditor local.
 - O contador do preditor de escolha é atualizado conforme a regra, mostrada na Tabela 1, tal que:

Resultado dos preditores	Ambos corretos ou ambos incorretos	Local correto, Global incorreto	Global correto, Local incorreto
Contador atualiza política de escolha	Sem mudança	Decrementz	Incrementa

Table 1: Política de atualização do contador de escolha.

- (a) O contador satura em 0 e 3 (o contador nunca ficará abaixo de zero ou acima de 3).
- (b) Todas as escolhas do contador são inicializadas para 1 no início da simulação.
- (c) Somente o preditor que foi selecionado (Local ou Global) é atualizado com o comportamento do branch.
- (d) O registrador do histórico de desvio Global será atualizado mesmo se o preditor local é selecionado.

A seleção dos preditores (Local, Global, ou Híbrida) será baseada nos parâmetros passados para a simulação, como:

- $\{m, n_1\}$ para escolha do preditor Local.
- $\{n_2\}$ para escolha do preditor Global.
- $\{\{m, n_1\}, \{n_2\}, k\}$ para escolha do preditor Híbrido.

Entrada

O simulador lê um arquivo de *trace* contendo o seguinte formato:

$0x < hexbranchPC > t/n$

$0x < hexbranchPC > t/n$

onde:

$< hexbranchPC >$ é a forma hexadecimal do endereço da instrução de branch na memória.

t - indica que o branch foi Tomado.

n - indica que o branch foi Não-Tomado.

Exemplo do *trace*:

b77a8a3a t

b77be7ab n

b77b55a0 t

Saída

A saída da simulação deve conter:

- O número de desvios encontrados no Programa.
- O número de *Miss Prediction* (predição tomada quando o realizado foi não-tomada e vice-versa).
- Taxa de predição errada ($\text{Miss Prediction} / \text{Número de branches}$).

Requisitos da interface do Programa

1. O código deve rodar em ambiente Linux (qualquer distro), e ser desenvolvido em qualquer linguagem que permite a apresentação via Web.
2. O programa deve rodar em um ambiente gráfico, sendo o máximo didático e mostrando os detalhes da execução do código na tela.
3. O programa deve conter um menu de escolha do tipo de preditor, entrada do arquivo de *trace* e entrada dos parâmetros de forma simples: para o preditor local os parâmetros $\{m, n_1\}$, para o Global $\{n_2\}$ e para o Híbrido $\{\{m, n_1\}, \{n_2\}, k\}$.
4. É importante que o programa simule passo a passo as iterações dos preditores.

Experimentos

O arquivo com os *traces* serão fornecidos ou criados pelos grupos de acordo com a necessidade do preditor. O simulador deve ser executado para algumas configurações específicas:

1. Preditor Local de desvios ($5 \leq m \leq 15, 2 \leq n \leq m$), onde n é par:
 - Apresente um gráfico de resultados, onde o eixo y corresponde a taxa de predição errada do desvio (*MissPrediction*), e o eixo x corresponde ao m .
 - Discuta os resultados e os compare com o Preditor Global.
2. Preditor Global de desvios ($5 \leq n \leq 15$):
 - Apresente um gráfico de resultados para cada teste, onde o eixo y corresponde a taxa de predição errada do desvio (*MissPrediction*), e o eixo x corresponde ao n .
 - Discuta os resultados e os compare com o Preditor Local.

3. Pesquise uma maneira de minimizar a taxa de erros de predição errada (*MissPrediction Rate*) e minimizar o custo do preditor em bits, considerando um armazenamento máximo de 16KB e de 32 KB para as suas análises.
4. Para o Preditor Híbrido, encontre a melhor configuração com um armazenamento máximo de 16KB e de 32 KB. Pesquise uma maneira de minimizar a taxa de erros de predição errada (*MissPrediction Rate*) e minimizar o custo do preditor em bits.

Avaliação

O trabalho a ser desenvolvido será avaliado somente se for **inédito**, estiver funcionando e contiver:

- Informações sobre como instalar o simulador;
- Exemplos tendo como entrada um arquivo de *trace* de desvios realizados.
- Descrição detalhada e formal de cada preditor, inserido em alguma aba do simulador.
- Documentação do funcionamento em forma de Help dentro do simulador.
- As notas dos trabalhos serão comparativas, portanto caprichem nos detalhes do simulador.

Cronograma

A entrega do trabalho deve ser realizado em etapas de acordo com o seguinte cronograma:

1. Etapa 1: Entrega do Preditor Local de Desvios - 01/11/2019
2. Etapa 2: Entrega dos Preditores Local e Global - 15/11/2019
3. Etapa 3: Entrega Final, preditores local, global e híbrido - 29/11/2019

Os trabalhos serão **avaliados e pontuados por etapas** e devem ser postados no AVA do curso.

References

- [1] Hennessy, John L. and Patterson, David A.. Computer Architecture, Fourth Edition: A Quantitative Approach. Morgan Kaufmann Publishers Inc., 2006.

- [2] Patterson, David A. and Hennessy, John L.. Computer Organization & Design - The Hardware/Software Interface. 3Th Edition Morgan Kaufmann Publishers, Inc., 2005.
- [3] Gehringer, E. F. Computer Design & Technology. Electrical & Computer Engineering Department of North Caroline University. <https://people.engr.ncsu.edu/efg/521/s06/common/homework/projects/2/p2.pdf>, 2006.