Manual do Usuário & Documentação

Simulador de Memória Cache de Multinível

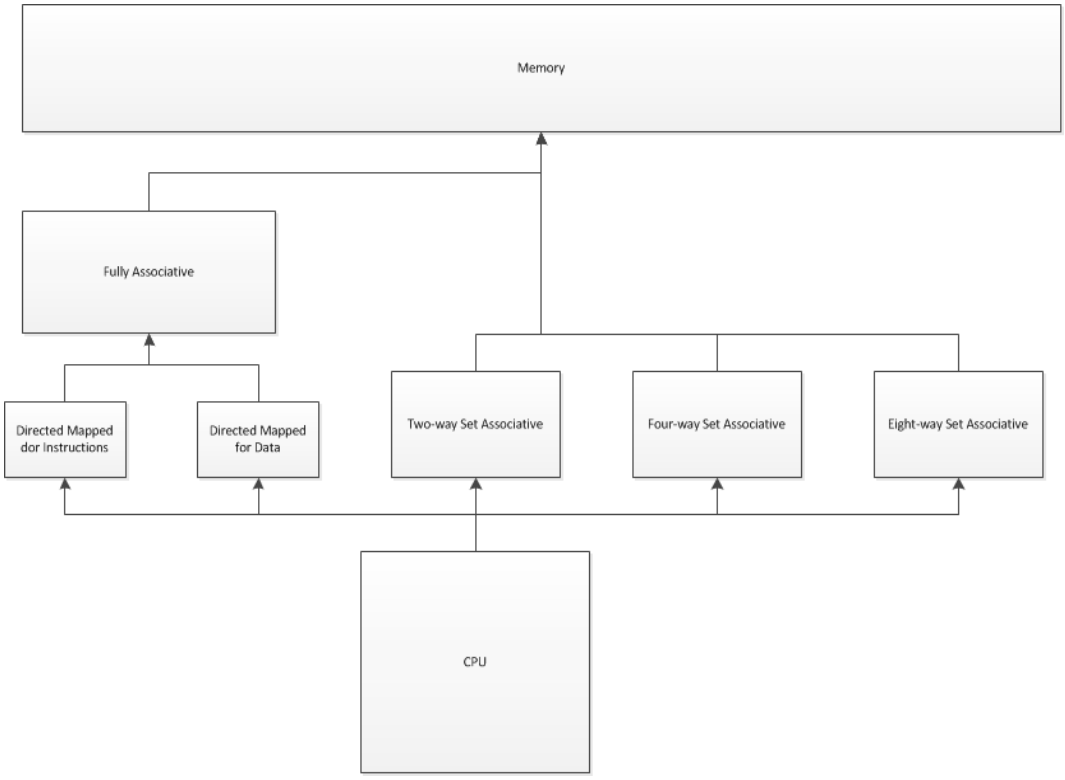
Desenvolvido por Aramis Hornung Moraes

Aramis

Introdução:

O simulador de memória cache a seguir é projetado de acordo com uma arquitetura definida, com uma cache multinível de duas *Directed Mapped* para instrução e para dados em primeiro nível e uma *Fully associative* em segundo nível, em paralelo está mais três *Set Associative* de duas, quatro e oito níveis.

A seguir está a ilustração que representa esta arquitetura.



De acordo com o determinado durante o desenvolvimento do projeto, as caches possuem tamanhos diferentes, mas possuem mesmo tamanho de linha de cache. A cache *FA* possui tamanho duas vezes o tamanho informado pelo usuário, as *DMs* possuem um oitavo o tamanho informado da cache.

Note que o tamanho de cache informado está na escala de KB – kilobytes – e está sujeito a arredondamento de valores. Vale a pena ressaltar que este valor de tamanho de cache não pode ser menor que 1 KB e não pode ser maior que 2 GB. Além disso, foi definido durante o planejamento que a arquitetura deste simulador seria de 32 bits.

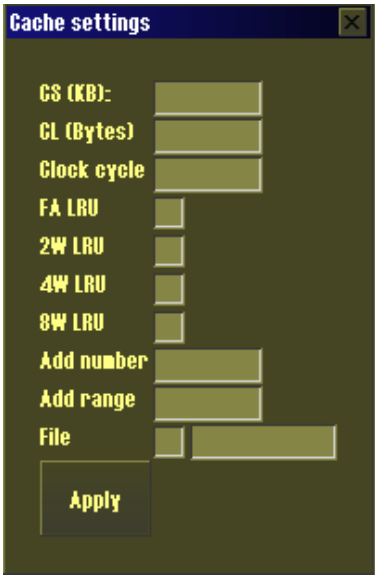
Interface:

A seguir será mostrado os módulos de janelas de que o simulador é composto, e o que cada uma faz, assim como um exemplo de simulação.

A imagem a seguir mostra a janela principal do simulador. Existem 3 menus: ***Cache Settings***, ***Console***, ***Simulation***.

Cache Settings

Permite que o usuário defina o tamanho da cache, a linha de cache, o tamanho da janela (em clocks) para o algoritmo de substituição LRU, *checkboxes* que definem quais caches vão usar o LRU durante a simulação, e na parte de endereços o usuário terá a opção de gerar endereços para uma simulação, ou simular através de um arquivo *txt*.



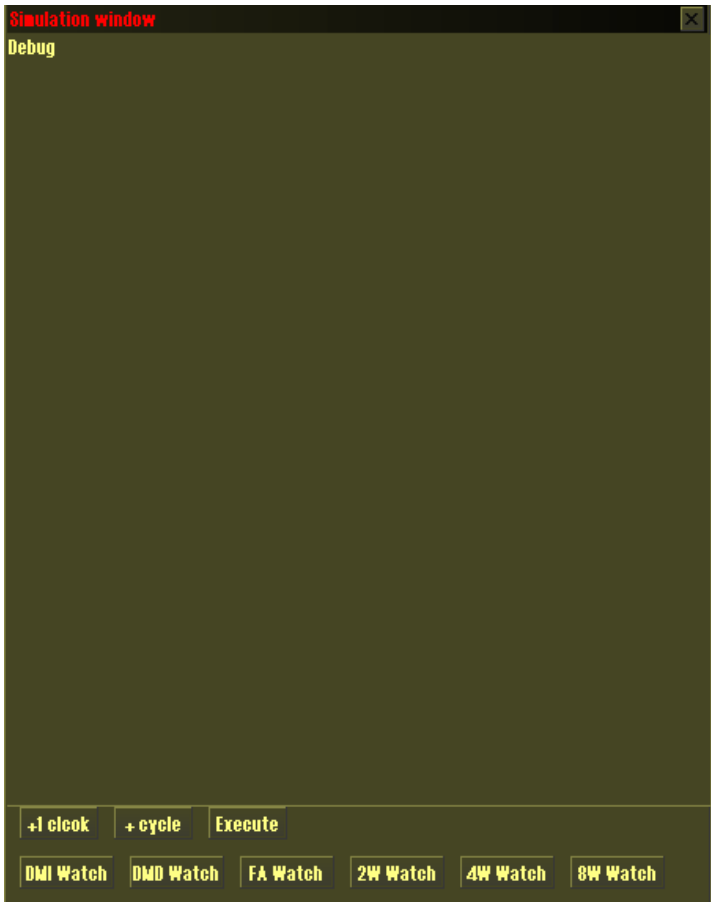
Depois de definidas esses parâmetros é possível fazer uma simulação.

Console

O console permite a visualização de cada passo que ocorre na simulação, nele serão mostrados as mensagens de erro, ocorrências de cada cache a cada informação de *hit* e *miss* em detalhe.

É altamente recomendado fechar o console na hora de executar simulações. Simulações com grande número de endereços com console aberto pode deixar a execução da simulação lenta, e fazer com que demore mais do que seria com console fechado para o resultado ser gerado, devido ao fato de a *engine* gráfica usada consumir um tempo para a inserção do evento no histórico e imprimir o texto no console, por isso é recomendado o fechamento do console a partir de simulações que usem mais de 1000 endereços.

Simulation

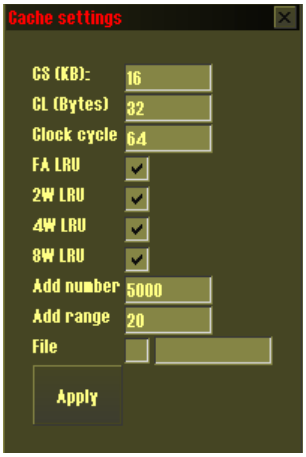


A janela de simulação mostra as informações gerais da simulação e controla a mesma. Existem três botões de navegação da simulação, um para simular de *clock* à *clock*, uma para ir de janela de clock à *janela* de *clock* e finalmente um para executar todo a simulação, que na em resumo faz a simulação até acabarem os endereços existentes no vetor de endereços.

Também existem seis botões que mostram as *views* de cada memória cache da arquitetura. A *view* é uma tabela que ilustra os bits de controle e o endereço que está armazenado em qual linha e ou via

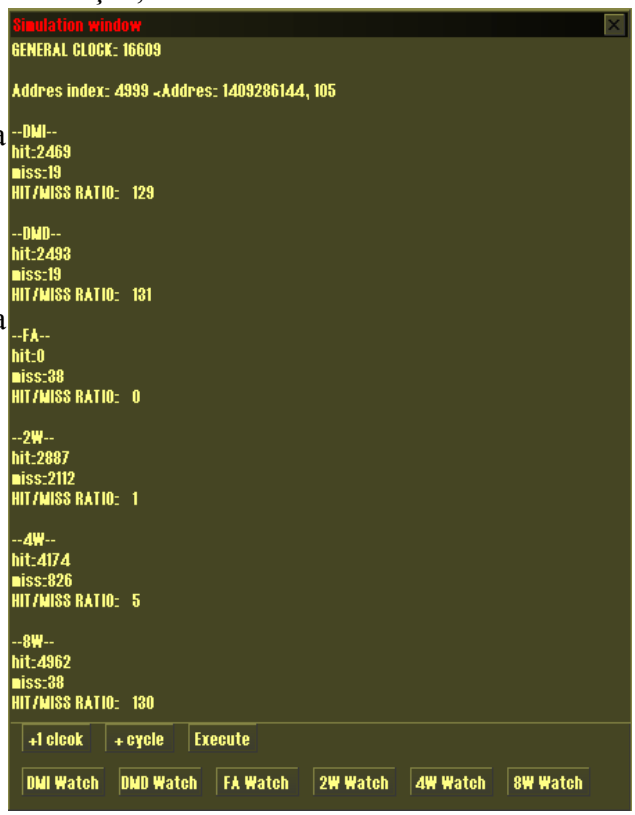
Exemplo de simulação

A seguir será mostrado o resultado de uma simulaçao proposta pelo autor. O tamanho de cache será de 16KB, linha de cache de 32b, janela de atualização de 64 ciclos, todos os algoritmos de LRU aplicado a caches que suportam. Serão gerados cinco mil endereços de 40 tipos (considerando que cada endereço pode ter instrução e dado)

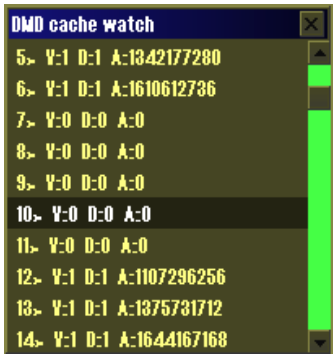


A simulação pode ser executada passo a passo, mas como estamos usando cinco mil endereços, é **altamente recomendado fechar o console**. Apos gerarmos os endereços podemos abrir as *views* e ver passo a passo do que está ocorrendo durante a simulação, ao invés disso nos simplesmente, novamente, fechamos o console, e clicamos em execute (executar). O resultado geral é impresso na própria janela de simulação, e as *views* também são atualizadas.

Aqui você pode ver o resultado gerado na janela de simulação, onde foram usados 16609 ciclos de *clock* para simular os 5000 endereços (4999 porque internamente dentro do programa o índice começa contar a partir do zero). Também durante uma simulação é possível checar qual endereço esta sendo solicitado, e a sua natureza em código ASCII.



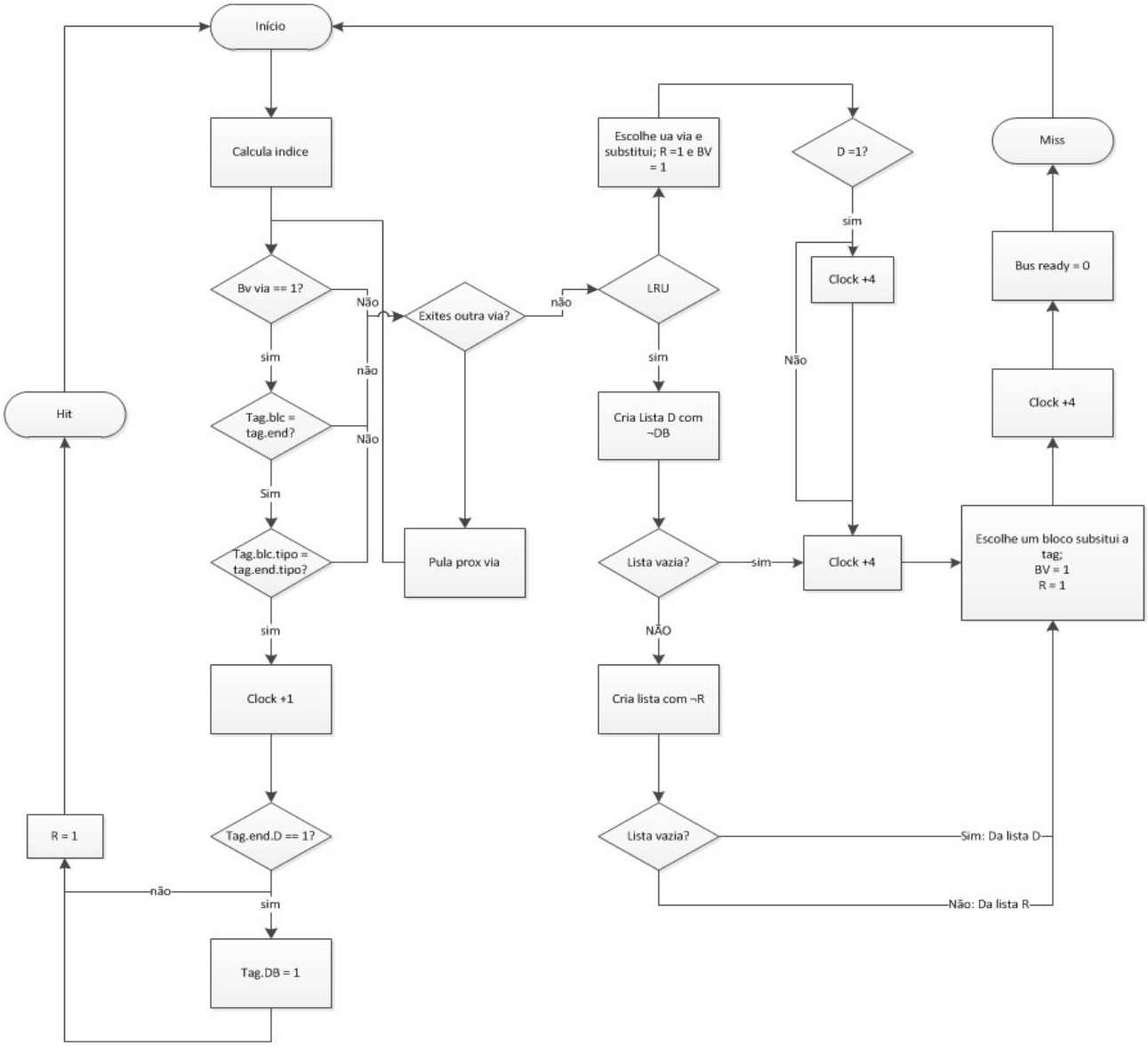
Também é feito um resumo de cada uma das seis caches da arquitetura, mostrando para cada uma, a taxa de acerto, erro e a media de acertos por erro. Abaixo a *view* da *DMD* para mostrar como ela aparece após uma simulação.



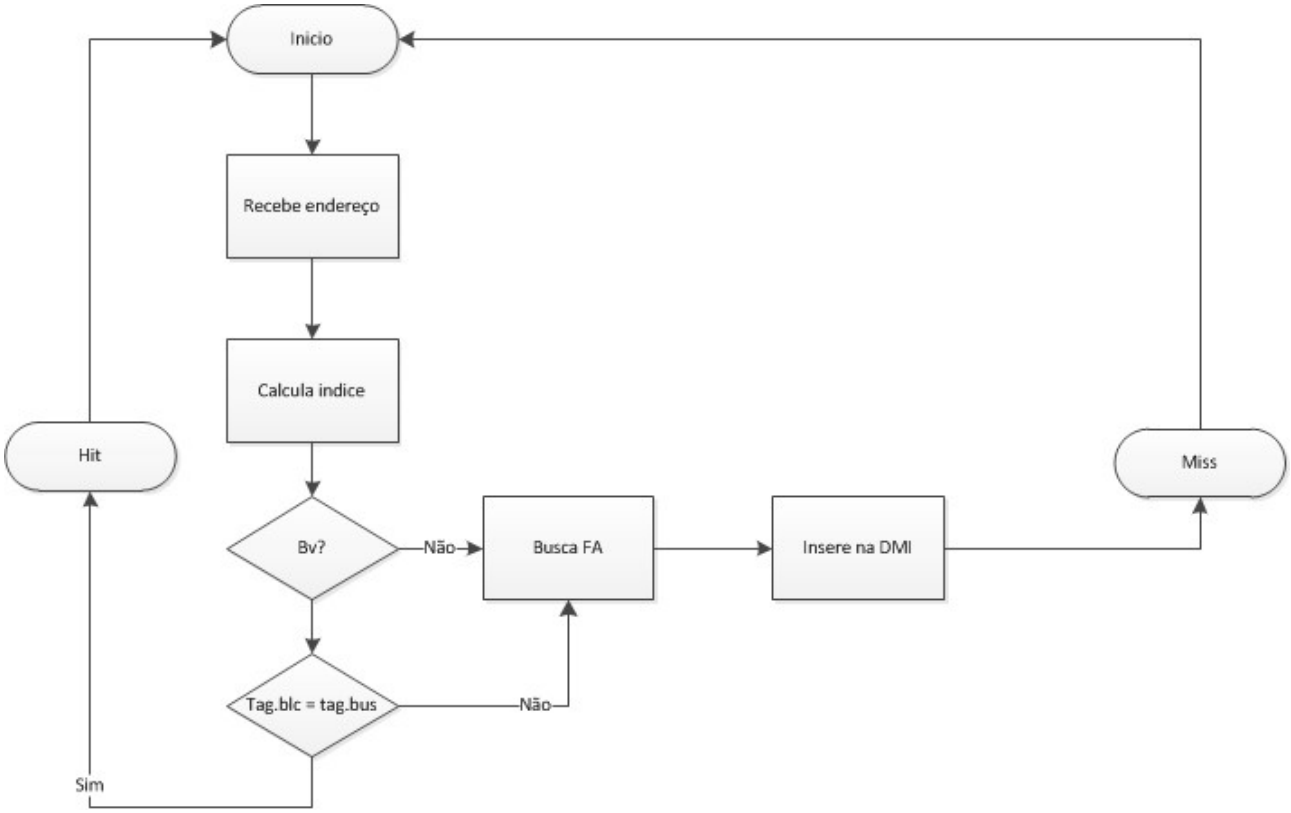
Diagramas do Simulador

A seguir todos os diagramas de funcionamento para cada tipo de cache.

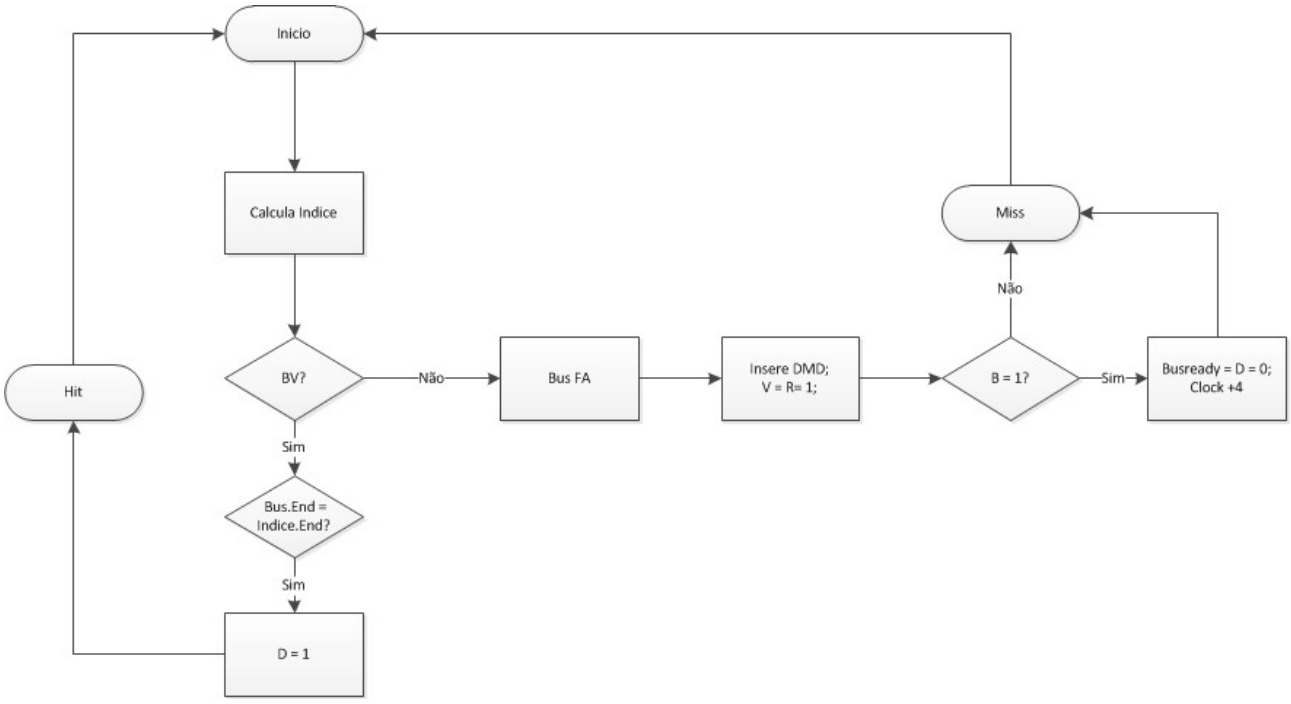
Set-Associative N-Way



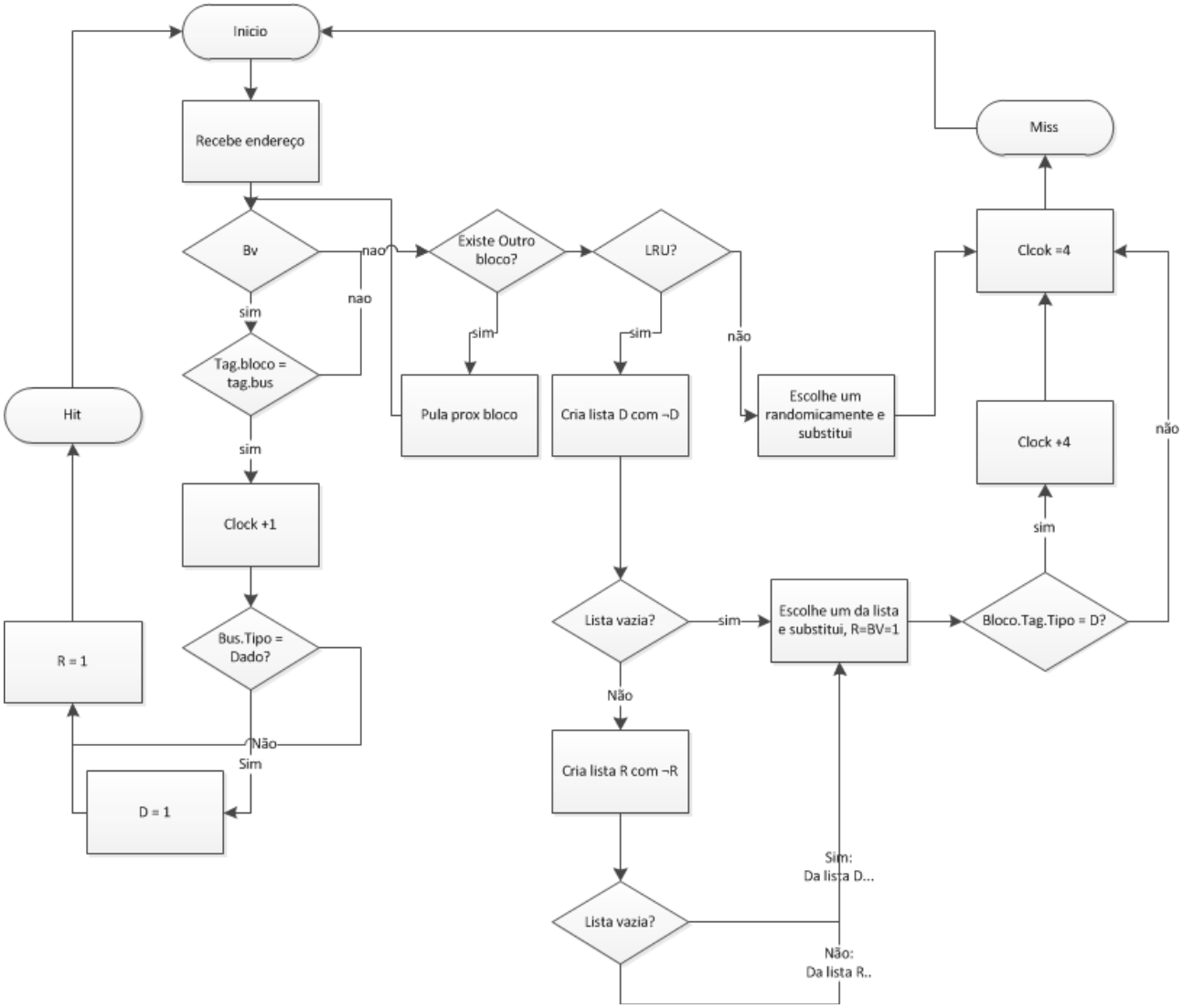
Direct-Mapped Instruction



Direct-Mapped Data



Fully-Associative



Manual escrito por Aramis Hornung Moraes