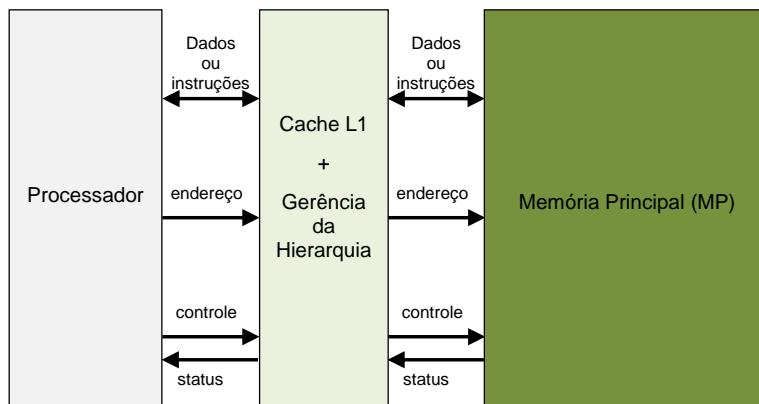


Hierarquia de Memória Cache – Versão 8

Este documento descreve o trabalho da disciplina de Arquitetura e Organização de Computadores, que consiste na implementação de um sistema de gerenciamento da hierarquia de memória. Este gerenciamento deve ser descrito em uma linguagem HDL (VHDL, SystemVerilog, ...), e simulado junto com um processador MIPS.

A seguir está apresentada uma hierarquia de memória com dois níveis: (i) cache de nível 1 (L1), (ii) Memória Principal (MP). O processador se comunica com a hierarquia de memória através das portas de: (i) **endereço** - aponta para o endereço de um dado ou instrução; (ii) **controle** - habilita o acesso à memória e define se este acesso é de leitura ou escrita; (iii) **status** - indica se a informação já foi lida ou escrita na/da memória; e (iv) dados ou instruções - porta bidirecional no caso da memória ser de dados (efetua escrita e leitura), e porta unidirecional no caso da memória ser de instruções (só leitura).



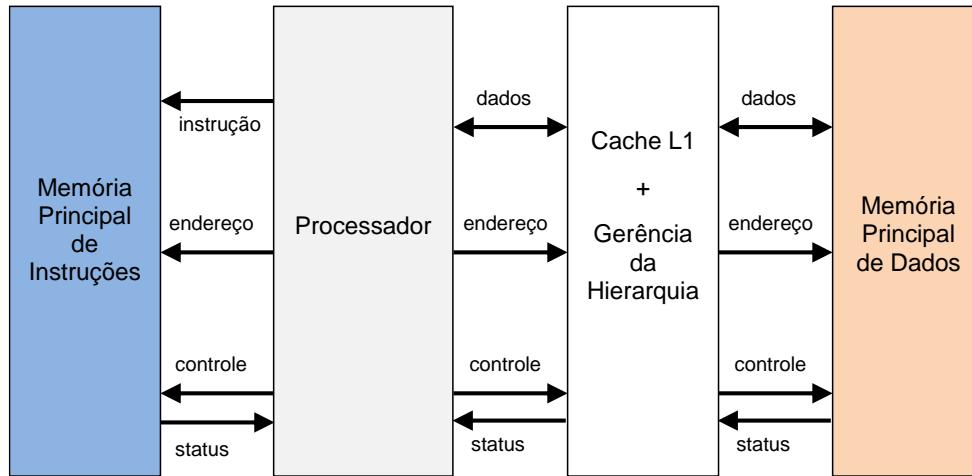
Independente da implementação, pois existem liberdades de escolha, os sinais provenientes do processador devem ir diretamente para o nível logo abaixo da hierarquia de memória, e cada nível será responsável por gerar os endereços para o nível subsequente. Cada nível de cache tem um conjunto de sinais que reportam o resultado da operação de leitura, tais como miss ou hit. Para cada nível, também, devem ser direcionados sinais de controle, bem como endereços.

O objetivo deste trabalho é implementar uma hierarquia de dados com 2 níveis (L1 e MP), sendo a hierarquia de controle implementada apenas com um nível (MP)

A memória de instruções deve ser implementada através de um arquivo contendo um código executável do processador. **Este código deve ser descrito de forma a verificar a funcionalidade da hierarquia da memória que está sendo implementada.** Assim, um programa que testa a hierarquia de memória deve fazer diversos acessos, forçando que ocorram casos de cache miss e cache hit de forma a explorar a localidade espacial e temporal do programa. O arquivo que contém o código executável do processador pode ser obtido com um programa *assembly* do MIPS sendo entrada para o montador do processador (e.g., MARS).

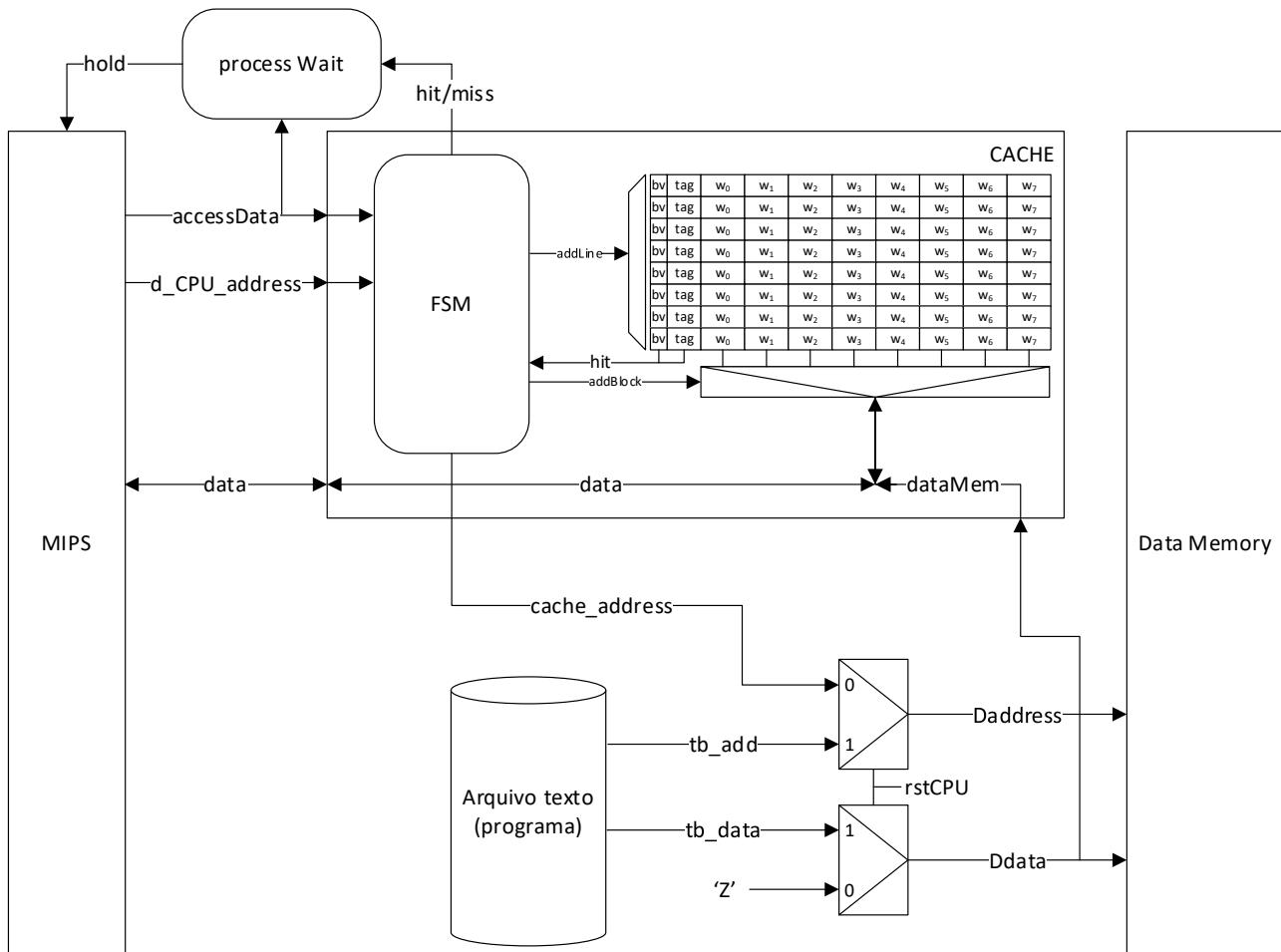
Considere que a cache de nível 1 tem 8 linhas, e cada bloco da linha tem 8 palavras. A cache L1 deve operar na mesma frequência do processador (acesso com um ciclo de relógio e borda invertida), enquanto que a MP tem tempo de acesso de 16 ciclos de relógio. A **Cache L1 deve ser implementada com mapeamento direto e write-through**. Demais características de implementação, não definidas aqui, estão livres para serem escolhidas.

Ao final do trabalho o aluno terá uma organização semelhante a que segue na figura abaixo. Note que o relógio não é apresentado, mas deve ter os valores descritos acima.



Esquema de organização MIPS / Cache / Memória de Dados

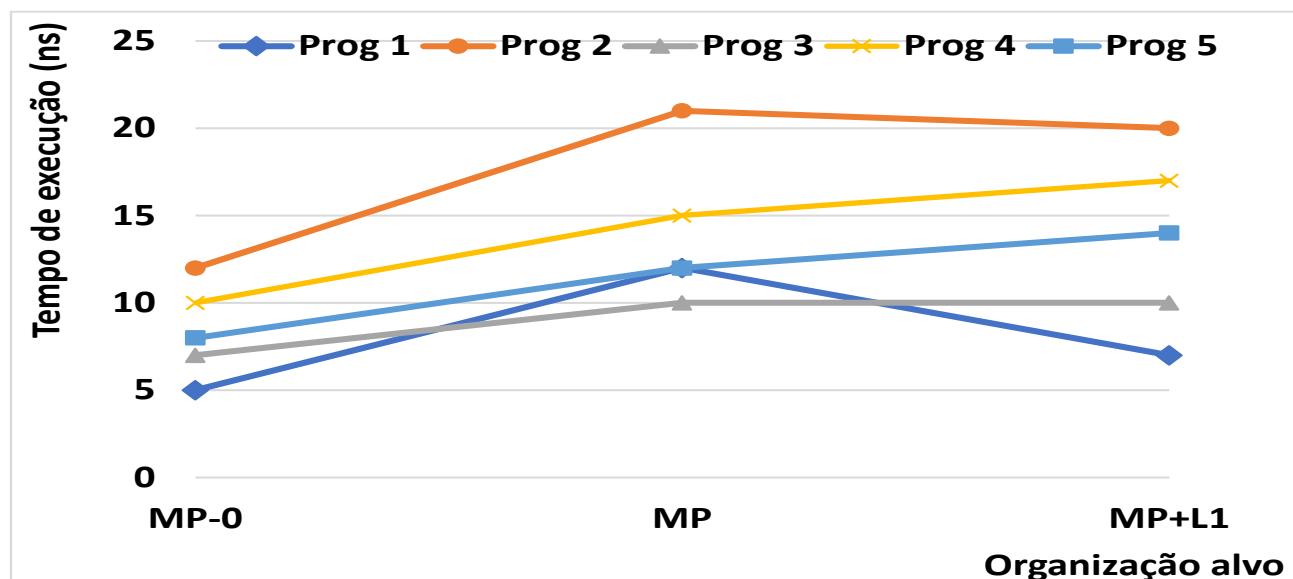
A figura que segue ilustra uma macro organização do trabalho completo, com os principais sinais que permitem multiplexar a informação entre a memória e o MIPS através da cache de dados.



Realização do trabalho

O trabalho deverá ser realizado em **no máximo** grupos de três alunos, e ser entregue até o dia descrito na agenda da disciplina. **Deve ser entregue um relatório descrevendo as atividades feitas, juntamente com os arquivos fonte HDL (VHDL, SystemVerilog, ...) da implementação do trabalho e os arquivos com os experimentos utilizados.** É importante descrever a validação da gerência da hierarquia de memória. Note que validar este tipo de sistema implica experimentos com sucessivos miss e hits.

Os grupos devem apresentar gráficos com as latências de cinco programas (feitos pelos alunos, eventualmente copiados de livros ou Internet) para três organizações alvo, tal como exemplificado no gráfico abaixo. **Procure fazer programas que explorem as localidades espaciais e temporais com laços e saltos de tamanhos diferentes – pense no tamanho da cache.** Os grupos devem validar todos os programas no MARS e obter os arquivos de código e de dados para simular.



As organizações alvo consideradas são:

- MP-0: A organização básica contendo um processador MIPS (e.g., MR2) com a memória implementada sem considerar atraso; ou seja, uma organização usada em Organização e Arquitetura de Processadores ou fornecido no Moodle. Para este caso é necessário apenas simular os cinco programas e coletar os tempos de execução;
- MP: A organização MP-0 inserido atraso na memória principal. A sugestão é inserir um módulo que implemente o atraso definido entre o processador e a memória principal. Neste caso, o HDL que implementa o módulo de memória será igual ao MP-0. Adicionalmente, sempre que o processador acessar a hierarquia de dados, deve aguardar uma resposta de que o dado está disponível. Este procedimento pode ser feito com um sinal de ack (halt), em um modelo semi-síncrono. Note que o processador terá que ser parcialmente modificado para não seguir a sua execução enquanto o ack não for ativado;
- MP+L1: A organização MP com a interposição de uma cache de nível 1 entre processador e memória principal.