

# ARQUITETURA DE COMPUTADORES

## Prática 6

Pretende-se analisar a hierarquia de memória do processador ilustrado na Figura 1. O processador funciona a uma frequência de 2GHz e é implementado por uma arquitetura *pipelined* que suporta o subconjunto do conjunto de instruções RV64G para operações com inteiros e virgula flutuante, utilizando **32 bits para instruções** e **64 bits para dados**. A unidade de armazenamento do processador é composta por um banco de 32 registos de inteiros de 64 bits.

A hierarquia de memória é composta por uma cache de instruções e uma cache de dados, ambas ligadas a uma memória principal partilhada. A configuração de todos os componentes da hierarquia de memória é apresentada na Tabela 1.

Resolva os exercícios abaixo de acordo com informação fornecida no enunciado.

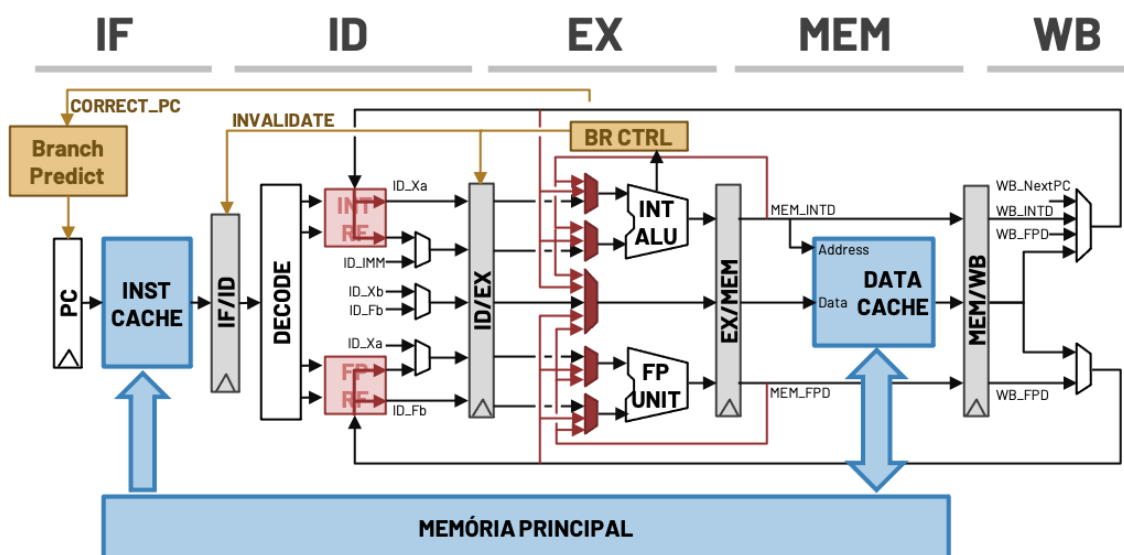


Figura 1 - Arquitetura do Processador

Tabela 1 – Configuração da Hierarquia de Memória.

CACHE INSTRUÇÕES	CACHE DADOS	MEMÓRIA PRINCIPAL
Mapeamento Direto Capacidade (dados) 64B Linhas de dados 16B	2 Vias de Associatividade Capacidade (dados) 128B Linhas de dados 16B Política de substituição: - <i>First-In First-Out (FIFO)</i> Política de escrita: - <i>Write Back, Write Allocate</i>	Capacidade 256MB
Latência $T_I=0.9ns$	Latência $T_D=1.2ns$	Latência $T_M= 20ns$

## Exercício 1

De acordo com as configurações apresentadas na Tabela 1, para cada uma das caches indique a decomposição da palavra de endereço (*Tag*, *Index* e *Offset*) e determine o número de ciclos de relógio necessários para realizar o acesso às caches.

## Exercício 2

Considere o código *Assembly* apresentado abaixo. Simule a execução do código na arquitetura da Figura 1 e apresente: **(1)** o preenchimento das caches de instruções e dados após o final da execução; e **(2)** o número total de *Hits* e *Misses* em cada cache.

**NOTA:** Para representar valores de dados utilize a representação da indexação ao vetor correspondente (ou seja,  $A[0]$ ,  $A[1]$ , ... e  $B[0]$ ,  $B[1]$ , ....).

```

000h|    ori    x5, x0, N        ; N=16
004h|    addi   x5, x5, -1       ; N-1 (15)
008h|    sll    x6, x5, 3        ; 64-bit data
00Ch|    addi   x10, x6, 80h     ; vector A
010h|    addi   x11, x6, 100h    ; vector B
014h|    addi   x12, x0, 200h    ; result
018h|    fcvt.d.w f12, x0       ; f12=0.0
01Ch| L:    jal    x1, R
020h|    addi   x10, x10, -8
024h|    addi   x11, x11, -8
028h|    addi   x5, x5, -1
02Ch|    bge    x5, x0, L
030h|    fsd    f12, 0(x12)

...
424h| R:    fld    f10, 0(x10)
428h|    fld    f11, 0(x11)
42Ch|    fmul.d f11, f11, f10
430h|    fadd.d f12, f12, f11
434h|    jalr   x0, x1, 0

```

CACHE INSTRUÇÕES		
V	TAG	DATA
00		
01		
10		
11		

CACHE DADOS		
V	TAG	DATA
00		
01		
10		
11		

## Exercício 3

Calcule a taxa de sucesso no acesso (*Hit Rate*) de cada cache ( $HR_I$  e  $HR_D$ ) com base nos resultados obtidos no Exercício 2. Calcule os tempos médios de acesso (em nanossegundos) às instruções ( $T_I$ ) e aos dados ( $T_D$ ) da hierarquia de memória.

## Exercício 4

Considere um processador que opera a uma frequência  $f_{clk} = 1$  GHz. Este processador contém duas caches de dados (L1 e L2). A configuração da hierarquia de memória é apresentada na Tabela 1.

