

Lista de Exercícios 2 Jean-Org-De-PC

Nome: Luiz Fernando Tagliaferro Brito, TIA 3186180-6

1)

a) É uma memória intermediária entre registrador e memória principal(ram), utilizada pela CPU para evitar procurar blocos de instruções ou dados na ram, possui 3 níveis se falarmos de uma arquitetura multicore, primeiro nível subdividido em 2 níveis, um para dados e outro para instruções. O bloco dois é exclusivo de cada core e o bloco 3 é compartilhado com todos os cores.

b) O gerenciador de cache faz com que os blocos mais usados fiquem armazenados temporariamente no mesmo, evitando uma frequente consulta à mem. principal, pois o tempo de busca de instruções na ram é maior e conseqüentemente afeta o desempenho da máquina. Assim com o cache, o tempo de busca de instruções fica menor, melhorando seu desempenho

c) O cache está dividido em 3 níveis:

1 Nível

1a- dados / 1.b instruções

2 Nível

3 Nível

O L1 é subdividido em bloco de instruções e bloco de dados. Nível 2 exclusivo de cada core, Nível 3 é compartilhado entre todos os núcleos.

d) No mapeamento direto, cada linha da memória principal é alocada em uma linha do cache, sendo que o primeiro a entrar é o último a sair (no caso das instruções)

Já no mapeamento associativo, um bloco todo de mem. principal é alocado no cache.

E por conjunto ele é dividido por conjuntos e cada um pode ter x números de linhas.

2)

16GB de DDR4(ram) 50,00 | 16 x 50

2TB HD 0,30 | 2480 x 0,30 ==> TOTAL PC DA ANA R\$ 1798,40

256GB SSD 1,50 | 256GB x 1,50

3)

As memorias DRAM, precisam de um refresh(dependem de corrente constante) para atualizar os dados , assim gastando mais energia porém seu custo é mais barato

Já as SRAM usam flip-flops, armazenando seus dados estaticamente , nao dependendo de uma corrente constante, porém são mais caras

As DDR4 sao classificadas como DRAM

4) Pois qunato maior o tamanho do endereço, maior o numero de palavras armazenadas por endereço

2^{32} suporta ate 4GB de ram

2^{64} suporta 16PB de ram

5) $2^{16} = 64536$

$2^{32} = 4294967296$

$2^{64} = 1,844674407 \times 10^{19}$

6)

$2^{16} = 64536 \rightarrow 64536 / 8\text{bits} = 8067$ palavras

$2^{32} = 4294967296 / 8\text{bits} = 536870912$ palavras

$2^{64} = 1,844674407 \times 10^{19} / 8\text{bits} = 2,305843009 \times 10^{18}$ Palavras

**caso dobre

$2^{16} = 64536 \rightarrow 64536 / 16\text{bits} = 4033,5$ palavras

$2^{32} = 4294967296 / 16\text{bits} = 26843545,5625$ palavras

$2^{64} = 1,844674407 \times 10^{19} / 16\text{bits} = 1,152921504 \times 10^{18}$

7)

**ALEATORIO--> Tempo de acesso constante independente da localização do dado

Exemplo: Memória RAM

***SEQUENCIAL --> acesso feito segundo uma sequência linear específica
tempo de acesso depende da localização dos dados
Exemplo: Fita cassete!

***ASSOCIATIVO:

Um dado é localizado baseado em parte de seu conteúdo e não por seu endereço
O tempo de acesso é constante e independente da localização ou do acesso anterior
parte da informação é utilizada para gerar uma tag, posição do dado está associada a parte do seu conteúdo
Ex. cache

***DIRETO --> Blocos individuais ou registros tem endereço único(referência), não necessário ler todos os arquivos até o desejado
Exemplo: HD
eof --> indica o fim do arquivo(fim da leitura)
end of file

8)

--volátil- os dados continuam armazenados em quanto houver corrente no sistema, assim após o dispositivo for desligado,
todos os dados contidos nessa memória serão perdidos.
Exemplos: RAM, Registrador e cache.

--estática- os dados continuam armazenados mesmo após a interrupção da corrente elétrica, seu HD não perde os dados quando
seu computador for desligado.
Exemplos: HD, SSD. Pendrive

9)

$8 \times 2^{10} / 1 = 8192$ palavras

10) Serve para transferência de dados e indicação de endereços de memórias dos dados. Interliga as memórias como cache, ram etc...

11) 64bits --> 8bytes e sua frequência é de 1800Mhz, então: $8B \times 1800\text{mhz} = >14.400\text{MB/s}$

12) As modificações seriam aumentar o tamanho do barramento e sua frequência máxima, assim, estaremos melhorando a quantidade de dados transmitidos por ciclo e juntamente aumentando ainda mais os ciclos.

13) Palavras 1byte e endereços de 32bits

a) $2^{32} \rightarrow 4\text{GB}$ de memoria

b) $8\text{KBytes} = 8 \times 2^{10} \rightarrow 2^3 \times 2^{10} \rightarrow 2^{13}$

$4\text{GB} / 8\text{KB} = 0,5 \times (2^{30} / 2^{10}) \rightarrow 0,5 \times 2^{20} \rightarrow 2^{-1} \times 2^{20} \rightarrow 2^{19}$

então: $2^{32} / 2^{13} = 2^{19}$ blocos!

c) $8 \times 2^{10} = 8 \times 1024 \rightarrow 8192$ Palavras

14) Palavras 1byte e endereços de 64bits

a) $2^{64} \rightarrow 16\text{PB}$

b) $4\text{KBytes} = 4 \times 2^{10} \rightarrow 2^2 \times 2^{10} \rightarrow 2^{12}$:

$16\text{PB} / 4\text{KB} \rightarrow (16 / 4) \times (2^{60} - 2^{10}) \rightarrow 2^2 \times 2^{50} \rightarrow 2^{52}$

então: $2^{64} / 2^{12} = 2^{52}$ blocos!

c) $4 \times 2^{10} = 4 \times 1024 \rightarrow 4096$ Palavras

15) x-> tamanho

$2^x = 16\text{GB} / 2\text{Bytes} \rightarrow 2^x = 8\text{GB} \rightarrow 2^x = 2^3 \times 2^{30} \rightarrow x = 33$

OU

$2^{64} \rightarrow$ supondo que queira em um endereçamento que exista

16)

a) 0110

b)

1 1 1 1 0 1 1 0 1 1 1 0

17)

PALAVRA 10001101

a) 1110

b)

0	1
1	0
1	1
1	1

18)

a) $8\text{GB}/2\text{KB} = 2^{30}/2^{10} = 2^{20}$ BLOCOS

b) $15002\text{Mhz} \times 2(\text{DDR}) = 3.000.000.000$ operações por seg

c) $T = 1/32^{30} \approx 0,332^{30}\text{s}$

d) $1500 \times 10^6 \times 8\text{bytes} \times 2 = 24000 \times 10^6 = 24\text{GB/s}$

e) $1500 \times 10^6 \times 8\text{bytes} \times 2 \times 2 = 48000 \times 10^6 = 48\text{GB/s}$

f) item d = 24 em 1seg
1 em x

X = 0,041s

Item f = 0,020s

19)

a)-78 em binário --> 01001110

em 8bits --> 10110010

em 16bits --> 1111111110110010

b)-101 em binário --> 01100101

em 8bits --> 10011011

em 16bits --> 1111111110011011

c) 83 em binário --> 01010011

em 8bits --> 10101101

em 16bits --> 000000001010011

d)-43 em binário --> 00101011

em 8bits --> 11010011

em 16bits --> 1111111111010011

20)

a) -78,50 Simples = 1 10000101 001110100000000000000000

Dupla = 1 10000000101
001110100

001110100

b) -101,125

Simples = 1 10000101 1001010010000000000000

Dupla = 1 10000000101
10010100100

10010100100

c) -97,750 Simples = 1 10000101 100001110000000000000000

Dupla = 1 10000000101
10000111000

10000111000
