

## Lista 1 pag 1 ex 1

A arquitetura de Von Neumann é composta por 3 componentes, sendo eles: CPU, memória e dispositivos de entrada e saída. A memória principal é o local onde são armazenados tanto os dados como as instruções de um programa. Os dispositivos de entrada e saída são usados para que o usuário envie o que deve ser processado pela CPU e, é por eles que a CPU exhibe ao usuário o resultado do processamento. Já a CPU é o módulo responsável por executar as operações de computação de dados, ele é composto por diversos tipos de registradores com objetivos e funções variadas como armazenar instruções, dados entre outras, as operações lógicas e aritméticas são realizadas pela ALU que usa as instruções e dados armazenados nos registradores p/ realizar as operações, e, para orquestrar todos os elementos existe um relógio que determina quando a nova computação deve ser realizada.

## ex 2

tempo acesso memória principal = 100 ms  
tempo de pesquisa na cache = 50 ms  
taxa de acerto é igual a 90%.

$$\begin{aligned} & (9 \cdot 50 \text{ ms} + 100 \text{ ms} + 50 \text{ ms}) / 10 \\ & (450 \text{ ms} + 100 \text{ ms} + 50 \text{ ms}) / 10 \\ & 600 \text{ ms} / 10 \text{ s} \end{aligned}$$

60 ms é o tempo médio

## ex 3

64 K células

256 operações } instrução  
2 operadores de endereço

a)  $2^{16} = 64 \text{ K células}$ , onde cada célula tem 1 instrução  
instrução =  $8 + 16 + 16 = 40 \text{ bits}$

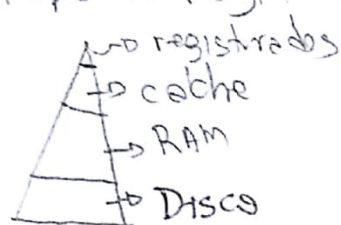
$$\text{Total bits} = 2^{16} \cdot 40 = 2621440$$

b) 16 bits, p/ indexar 64 K células

c) Seriam necessários 2 ciclos de busca

Ex 4

Devido as diferentes características das memórias existentes, em um computador existe uma hierarquia de memórias. As principais diferenças entre os tipos de memória são: volatilidade, tempo de acesso, preço e consequentemente disponibilidade. Comumente organizamos esta hierarquia usando uma pirâmide, onde, na base estão os discos e no topo os registradores.



Os elementos de memória mais rápidos e também os mais caros são os registradores e ficam no núcleo do CPU.

Abaixo deles estão as memórias cache que podem ser divididas em L1 e L2, também tendo seus tamanhos bem limitados e tendo um custo bem elevado.

Em seguida temos as memórias de acesso aleatório RAM. Atualmente na casa de Giga de disponibilidade em sistemas domésticos.

Por fim temos as memórias não voláteis, ou seja, que mesmo sem energia ainda mantêm os dados armazenados.

As mídias não voláteis como HD's tendem a ter um tempo de acesso muito maior do que as mídias voláteis.

Ex 5

512 Mbytes de memória física  
32 bytes blocos de memória  
8K blocos cache  
16 M blocos

end. palavra = 5 bits = 32 bytes  
nº blocos = 13 bits = 8K blocos  
tam memória = 29 bits 512 Mbytes

$$2^{29} = 512 \text{ Mb}$$

tamanho da cache = 256 Kbytes

a) Mapeamento direto

$$\text{TAG} = 11$$

$$\text{BLOCO} = 13$$

$$\text{Palavra} = 5$$

b) mapeamento associativo

$$\text{TAG} = 24$$

$$\text{Palavra} = 5$$

c) cada conj possui 4 linhas = 2 bits

$$\text{TAG} = 13 \quad \text{CONJUNTO} = 11 \quad \text{Palavra} = 5$$

$$\text{Palavra} = 5$$



Pag 2 ex 6

A memória RAM é finita, sendo assim ocorrerão situações onde o nº de processos precisando da memória, excede a capacidade da mesma de atender essa requisição, além disso, um processo não poderia acessar a memória usada por um outro processo. Para atender a essas duas características, o SO usa uma estratégia chamada de memória virtual. Nela, cada processo ganha um conj de endereçamento virtual que será posteriormente mapeado para um endereço de memória física. Ou seja endereço físico é usado pelo SO no passo que endereços virtuais são usados pelos processos.

Ex 7

CI = 16 bits

RI = 40 bits

cada instrução tem 2 operandos do mesmo tamanho

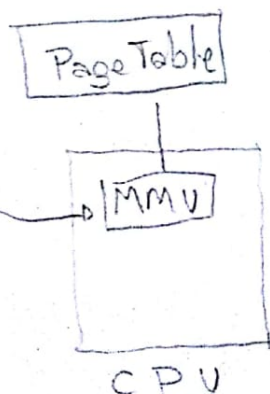
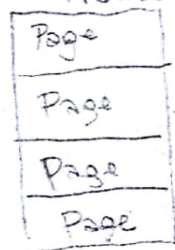
a) Se o RI = 40 bits então a instrução tem 40 bits

b) 8 bits

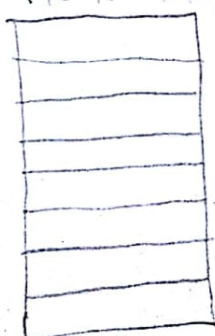
Ex 8

A memória dada pelo SO para um programa é virtual. Quando um processo acessa um endereço virtual, que é o único que ele conhece, o SO precisa converter esse endereço virtual em um endereço físico. Quem realiza a tradução entre endereçamento físico e virtual é a MMU, ela tem acesso a uma tabela de paginação que indica a posição inicial e final de memória física de uma página virtual. Quando a posição virtual que está sendo acessada não está na memória RAM, o SO copia os dados do disco para a memória RAM.

memória virtual do processo



memória física



FRAME Mapeado

10-

Uma falta de página ocorre quando um processo solicita dados de uma posição de memória cujos dados não estão na memória RAM, mas sim no disco. Uma falta de página pode ser causada por diversas razões como por exemplo: a memória ainda não foi inserida na RAM, os dados da RAM não são mais válidos, em alguma troca de página realizada anteriormente a posição solicitada estava em uma página que foi enviada ao disco. O SO recebe uma notificação da CPU sobre a falta de página, então, ele deve ir até o disco, carregar os dados para a memória, nesse momento pode ser necessário remover da memória RAM outra página para dar espaço para a nova página.

11- RAM

1	2	3	4		1	2	3	4	FIFO	LRU
7	4	1	0	}	7	4	1	0	0 miss	miss
4	1	0	2		7	4	1	0	1 hit	hit
1	0	2	3		7	4	0	1	7 hit	hit
0	2	3	7		4	0	1	7	2 miss	miss
2	3	7	1		0	1	7	2	3 miss	miss
3	7	1	0		1	7	2	3	2 hit	hit
					1	7	3	2	7 miss	hit
					1	3	2	7	1 miss	hit
					3	2	7	1	0 miss	miss
					2	7	1	0	3 hit	miss
									6	5

12-

$$a = 20 - 0 = 20 + 8 \cdot 1024 = \boxed{8212}$$

$$b = 4100 - 4096 = 4 + 4 \cdot 1024 = \boxed{4100}$$

$$c = 8300 - 8192 = 108 \text{ (deslocamento)} + 24 \cdot 1024 = \boxed{24684}$$

9- Ambos os tamanhos são potências de 2 pois assim a conversão do endereço lógico p/ o físico é feito de forma mais rápida usando deslocamento de bit.