

## AD1 - Organização de Computadores 2008.2

### Data de entrega 13/09/2008 - GABARITO

1. (1,0) Considere uma máquina com arquitetura semelhante àquela apresentada em aula. Suponha que uma nova instrução denominada LDADDSTR Op seja inserida no conjunto de instruções desta máquina. Esta instrução soma o conteúdo do acumulador com o conteúdo da célula de memória cujo endereço é Op e armazena o resultado na memória no endereço Op. Descreva detalhadamente como será a execução desta instrução.

**Como será realizada a execução desta instrução:**

- ⇒ *Passo 1: A CPU coloca no REM o valor do operando ( REM <- Op ),*  
     *Passo 1.1: O conteúdo do REM é disponibilizado no barramento de endereço*
- ⇒ *Passo 2: A CPU aciona pelo barramento de controle o sinal de leitura de memória*
- ⇒ *Passo 3: A memória coloca o valor no barramento de dados, e por consequência no RDM da CPU ( RDM <- MP(Op) )*
- ⇒ *Passo 4: A CPU executa a soma do valor recebido com o acumulador enviando o resultado para o registrador de dados para memória; RDM <- ACC + RDM*
- ⇒ *Passo 5: A CPU coloca no REM o valor do operando (Op),*  
     *Passo 3.1: O conteúdo do REM é disponibilizado no barramento de endereço*  
     *Passo 3.2: A CPU disponibiliza no barramento de dados o conteúdo do RDM*
- ⇒ *Passo 6: A CPU disponibiliza o sinal de escrita de memória no barramento de controle*
- ⇒ *Passo 7: A memória armazena o conteúdo do barramento de dados na posição que está contida no barramento de endereços MP(Op) <- RDM*

2. (1,0) Considere uma máquina com arquitetura semelhante àquela apresentada em aula. Pode-se endereçar no máximo 1M células de memória onde cada célula armazena uma palavra e cada instrução tem o tamanho de uma palavra. Esta máquina possui uma capacidade de armazenamento de 3M bytes. Todas as instruções desta máquina possuem o mesmo formato: um código de operação e um operando que indica um endereço de célula de memória.

$$N = 1M \text{ células}$$

$$T = \text{Capacidade máxima} = 3M\text{bytes (ou 24Mbits)} \Rightarrow$$

$$M = 3M\text{bytes} / 1M\text{células} \Rightarrow M = 3 \text{ bytes / célula. (24bits/célula)}$$

- a) Qual o tamanho mínimo do REM?

$$REM = E = \text{tamanho em bits necessários para acessar toda a memória (N)}$$

$$N = 2^E = 1M \text{ células} = 2^{20} \text{ células} \Rightarrow E = 20 \Rightarrow REM = 20 \text{ bits}$$

- b) Qual o tamanho mínimo do CI?

*O CI deverá ter o tamanho necessário para acessar toda a memória:*

$$CI = REM = 20 \text{ bits}$$

- c) Qual o tamanho do barramento de endereços?

$$\text{Barramento de endereços} = REM = 20 \text{ bits}$$

- d) Qual o tamanho mínimo do RI?

$$\text{Tamanho mínimo de RI} = \text{tamanho da instrução} = \text{tamanho da palavra} = \text{tamanho da célula}$$

$$\text{Tamanho mínimo de RI} = 3 \text{ bytes (24 bits)}$$

e) Qual o número máximo de códigos de operação?

*Tamanho da instrução = 3 bytes (24 bits)*

*Tamanho da instrução = código de operação + operando (endereço de memória)*

*Código de operação = Tamanho da instrução – operando = 24bits – 20 bits = 4bits*

*Quantidade máxima de instruções =  $2^4 = 16$  instruções*

**3. (1,0) Considere uma máquina cujo relógio possui uma frequência de 1 GHz e um programa no qual são executadas 1000 instruções desta máquina.**

**a) Calcule o tempo de UCP utilizado para executar este programa, considerando que cada instrução é executada em um ciclo de relógio e a execução de uma instrução só se inicia quando a execução da instrução anterior é finalizada.**

*1GHz = 1.000.000.000 Hz*

*Tempo de um ciclo de relógio =  $1/1.000.000.000 = 0,000\ 000\ 001$  seg. ou 1ns (nanosegundos)*

*Tempo de execução de 1 instrução = tempo de um ciclo de relógio = 1ns*

*1000 instruções executadas sequencialmente =  $1000 \times 1ns = 1000ns$  ou  $1\mu s$  (para executar 1000 instruções)*

**b) Considere que essa máquina utilize um pipeline de 5 estágios, todos de igual duração. Calcule o tempo máximo que o estágio deve durar para que o tempo de execução do programa seja menor do que o tempo calculado no item anterior.**

*Tempo total para execução das 1000 instruções deverá neste caso < que 1000ns que foi o tempo do*

*1º.caso*

*Consideremos  $t$  como sendo o tempo para execução de um estágio*

*Tempo para execução de uma instrução = 5 estágios  $\times t = 5t$*

*Tempo para execução das demais em pipeline =  $999 \times t = 999t$*

*Tempo para execução das 1000 instruções =  $5t$  (primeira instrução) +  $999t$  (para as demais) =  $1004t$*

*O tempo total para execução do 2º. Caso < tempo total execução do 1º. Caso  $\Rightarrow$*

*$1004t < 1000ns \Rightarrow t < 1000/1004 \Rightarrow t < 0,996ns$  ou  $t_{max} = 0,99599...ns$  (tempo para um estágio)*

**4. (1,0) Algumas placas mãe de computadores atuais utilizam os chipsets. Explique o que são e como funcionam os chipsets (fontes de consulta: Guia do Hardware (<http://www.guiadohardware.net/>) e Clube do Hardware (<http://www.clubedohardware.com.br/>)).**

**(texto e fotos retirados do site [www.clubedohardware.com.br](http://www.clubedohardware.com.br/))**

*Chipset é o nome dado ao conjunto de chips (set significa “conjunto”, daí o seu nome) usado na placa-mãe.*

*Nos primeiros PCs, a placa-mãe usava circuitos integrados discretos. Com isso, vários chips eram necessários para criar todos os circuitos necessários para fazer um computador funcionar.*

*Após algum tempo os fabricantes de chips começaram a integrar vários chips dentro de chips maiores. Como isso, em vez de usar uma dúzia de pequenos chips, uma placa-mãe poderia ser construída usando apenas meia dúzia de chips maiores.*

*O processo de integração continuou e em meados dos anos 90 as placas-mãe eram construídas usando apenas dois ou até mesmo um único chip grande.*

*Com o lançamento do barramento PCI, um novo conceito, que ainda hoje em dia é utilizado, pôde ser empregado pela primeira vez: a utilização de pontes. Geralmente as placas-mãe possuem dois chips grandes: um chamado ponte norte e outro chamado ponte sul. Às vezes, alguns fabricantes de chip podiam integrar a ponte norte e a ponte sul em um único chip; neste caso a placa-mãe terá apenas um circuito integrado grande. Com o uso da arquitetura em pontes, os chipsets puderam ser padronizados.*

*Os fabricantes de placas-mãe compram dos fabricantes de chipsets os chipsets para serem integrados em suas placas. Na verdade, existe um aspecto muito interessante nessa relação. Para construir uma placa-mãe, o fabricante da placa pode seguir o projeto padrão do fabricante do chipset, também conhecido como “modelo de referência”, ou pode criar seu próprio projeto, fazendo modificações no circuito para oferecer maior desempenho e mais funcionalidades.*

## Ponte Norte

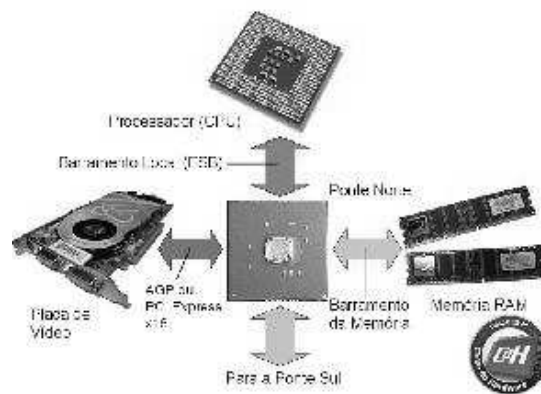
O chip ponte norte, também chamado de MCH (Memory Controller Hub, Hub Controlador de Memória) é conectado diretamente ao processador e possui basicamente as seguintes funções:

- Controlador de Memória (\*)
- Controlador do barramento AGP (se disponível)
- Controlador do barramento PCI Express x16 (se disponível)
- Interface para transferência de dados com a ponte sul

(\*) Exceto para processadores soquete 754, soquete 939 e soquete 940 (processadores da AMD, como é o caso do Athlon 64), já que nesses processadores o controlador de memória está localizado no próprio processador, e não na ponte norte.

Alguns chips ponte norte também controlam o barramento PCI Express x1. Em alguns outros é a ponte sul que controla o barramento PCI Express x1. Em nossas explicações assumiremos que a ponte sul é o responsável por controlar as pistas PCI Express x1, mas tenha em mente que isso pode variar de acordo com o modelo do chipset.

Na Figura 1 você pode ver um diagrama que mostra a função da ponte norte no computador.



**Figura 1:** Ponte norte.

Como você pode ver, o processador não acessa diretamente a memória RAM ou a placa de vídeo. É a ponte norte que funciona como intermediário no acesso do processador a estes dispositivos. Por causa disso, a ponte norte tem influência direta no desempenho do micro. Se um chip de ponte norte tem um controlador de memória melhor do que outro, o desempenho geral do micro será melhor. Isto explica o motivo pelo qual você pode ter duas placas-mãe voltadas para a mesma classe de processadores e que obtêm desempenhos diferentes.

Como comentamos anteriormente, nos processadores Athlon 64 o controlador de memória está integrado no próprio processador e é por isso que praticamente não existe diferença de desempenho entre placas-mãe para esta plataforma.

Como o controlador de memória está na ponte norte, é este chip que limita o tipo e a quantidade máxima de memória que você pode instalar no micro (no caso do Athlon 64, quem é o responsável por tais limites é o próprio processador, já que o controlador de memória está embutido nele).

A conexão entre a ponte norte e a ponte sul é feita através de um barramento. No início, o barramento utilizado para conectar a ponte norte à ponte sul era o barramento PCI. Atualmente, o barramento PCI não é mais usado para esse tipo de conexão e foi substituído por um barramento dedicado. Falaremos mais sobre isso adiante, já que o tipo de barramento utilizado nesta conexão pode afetar o desempenho do micro.

## Ponte Sul

O chip ponte sul, também chamado ICH (I/O Controller Hub, Hub Controlador de Entrada e Saída) é conectado à ponte norte e sua função é basicamente controlar os dispositivos on-board e de entrada e saída tais como:

- Discos Rígidos (Paralelo e Serial ATA)

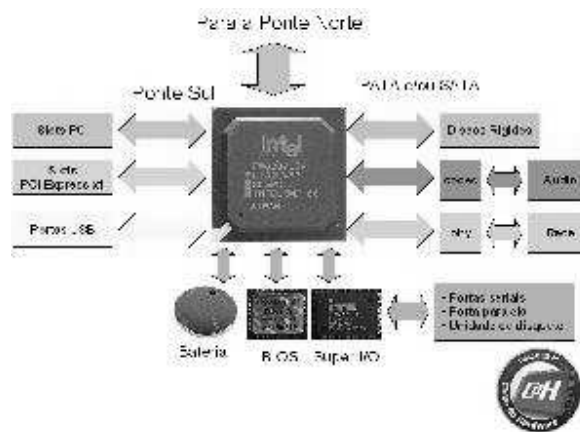
- Portas USB
- Som on-board (\*)
- Rede on-board (\*\*)
- Barramento PCI
- Barramento PCI Express (se disponível)
- Barramento ISA (se disponível)
- Relógio de Tempo Real (RTC)
- Memória de configuração (CMOS)
- Dispositivos antigos, como controladores de interrupção e de DMA

(\*) Se a ponte sul tiver controlador de som on-board, será necessária a utilização de um chip externo chamado de codec (abreviação de codificador/decodificador) para funcionar.

(\*\*) Se a ponte sul tiver controlador de rede on-board, será necessária a utilização de um chip chamado phy (pronuncia-se “fái”, abreviação de physical, camada física, em português) para funcionar.

A ponte sul é também conectada a dois outros chips disponíveis na placa-mãe: o chip de memória ROM, mais conhecido como BIOS, e o chip Super I/O, que é o responsável por controlar dispositivos antigos como portas seriais, porta paralela e unidade de disquete.

Na Figura 2 você pode ver um diagrama que mostra a função da ponte sul no computador.



**Figura 2:** Ponte sul.

Como você pode ver, enquanto que a ponte sul pode ter alguma influência no desempenho do disco rígido, este componente não é tão crucial no que se refere ao desempenho geral do micro quanto à ponte norte. Na verdade, a ponte sul tem mais a ver com as funcionalidades da sua placa-mãe do que com o desempenho. É a ponte sul que determina a quantidade (e velocidade) das portas USB e a quantidade e tipo (ATA ou Serial ATA) das portas do disco rígido que sua placa-mãe possui, por exemplo.

##### **5. (1,0) Descreva como funciona uma arquitetura microprogramada e indique a diferença entre microinstruções verticais e horizontais.**

Em uma arquitetura microprogramada, a unidade de controle é especificada por um microprograma que consiste de uma seqüência de instruções de uma linguagem de microprogramação. Estas instruções são muito simples e especificam microoperações. Uma unidade de controle microprogramada é implementada com circuitos lógicos e é capaz de seguir uma seqüência de microinstruções gerando sinais de controle para que cada uma delas seja executada. Os sinais de controle gerados por uma microinstrução são usados para causar transferências de dados entre registradores e memória e execução de operações pela ULA.

As microinstruções horizontais têm como característica ter funções distintas para cada bit que a compõe, como por exemplo, controlar uma linha de controle interna da UCP, controlar uma linha de barramento externo de controle, definir condição de desvio e endereço de desvio entre outras. Tem a vantagem de ser simples, podendo controlar várias microoperações em paralelo, além de uma eficiente utilização do hardware. E possui a desvantagem de maior ocupação de espaço de memória de controle em relação à microinstrução vertical.

As microinstruções verticais se caracterizam por possuir um decodificador extra para identificar quais as linhas que serão efetivamente ativadas. Sua principal vantagem é reduzir o custo da Unidade de controle em função do menor tamanho da instrução, o que poderá ser necessário para uma maior quantidade de instruções. Tem como principal desvantagem o aumento do tempo de execução, devido à necessidade da decodificação dos campos de cada microinstrução.

**6. (1,0) Descreva passo a passo as operações de leitura da memória e de escrita na memória, indicando como os registradores RDM e REM são utilizados e como a unidade de controle gera os sinais necessários.**

*Passos de uma operação de escrita*

- 1) (REM) <- (outro registrador)
  - 1.1) O endereço é colocado no barramento de endereços
- 2) (RDM) <- (outro registrador)
  - 2.1) O dado é colocado no barramento de dados
- 3) Sinal de escrita é colocado no barramento de controle
- 4) (MP(REM)) <- (RDM)

*Passos de uma operação de leitura*

- 1) (REM) <- (outro registrador da UCP)
  - 1.1) O endereço é colocado no barramento de endereços
- 2) Sinal de leitura é colocado no barramento de controle
  - 2.1) Decodificação do endereço e localização da célula na memória
- 3) (RDM) <- (MP(REM)) pelo barramento de dados
- 4) (outro registrador da UCP) <- (RDM)

**7. (1,0) Um computador possui uma capacidade máxima de memória principal com 64K células, cada uma capaz de armazenar uma palavra de 8 bits.**

a. Qual é o maior endereço em decimal desta memória ?

$$N = 64K \text{ células} = 2^{16}$$

$$\text{Último endereço} = N - 1 = 67108864 - 1 = 67108863$$

b. Qual é o tamanho do barramento de endereços deste sistema ?

$$\text{Barramento de endereços} = E$$

$$N = 2^E = 2^{16}, \text{ portanto } E = 16, \text{ Barramento de endereços} = 16 \text{ bits}$$

c. Quantos bits podem ser armazenados no RDM e no REM ?

$$\text{O REM terá que ter o tamanho do barramento de endereços} = 16 \text{ bits}$$

$$\text{CI terá o tamanho necessário para endereçar toda a memória} = 16 \text{ bits}$$

$$\text{RDM} = \text{tamanho da palavra} = 8 \text{ bits}$$

d. Qual é o número máximo de bits que pode existir na memória ?

$$T = N \times M = 64K \text{ células} \times 8 \text{ bits/célula} = 512 \text{ Kbits}$$

**8. (1,0) Considere uma máquina que possa endereçar 512 Mbytes de memória física, utilizando endereço referenciando byte, e que tenha a sua memória organizada em blocos de 16 bytes. Ela possui uma memória cache que pode armazenar 8K blocos, sendo um bloco por linha. Mostre o formato da memória cache, indicando os campos necessários (válido, tag, bloco) e o número de bits para cada campo, e o formato de um endereço da memória principal, indicando os bits que referenciam os campos da cache, para os seguintes mapeamentos:**

**a) Mapeamento direto.**

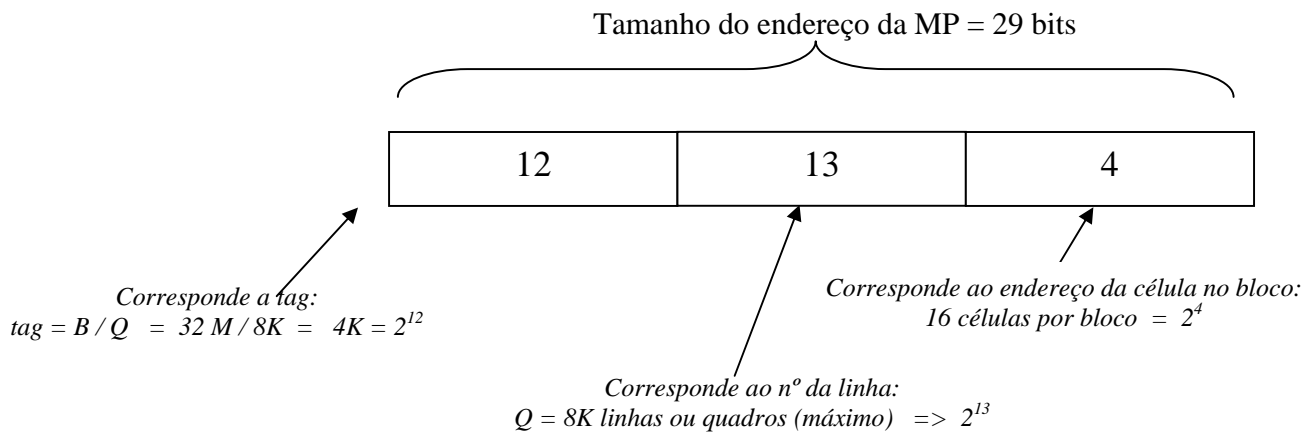
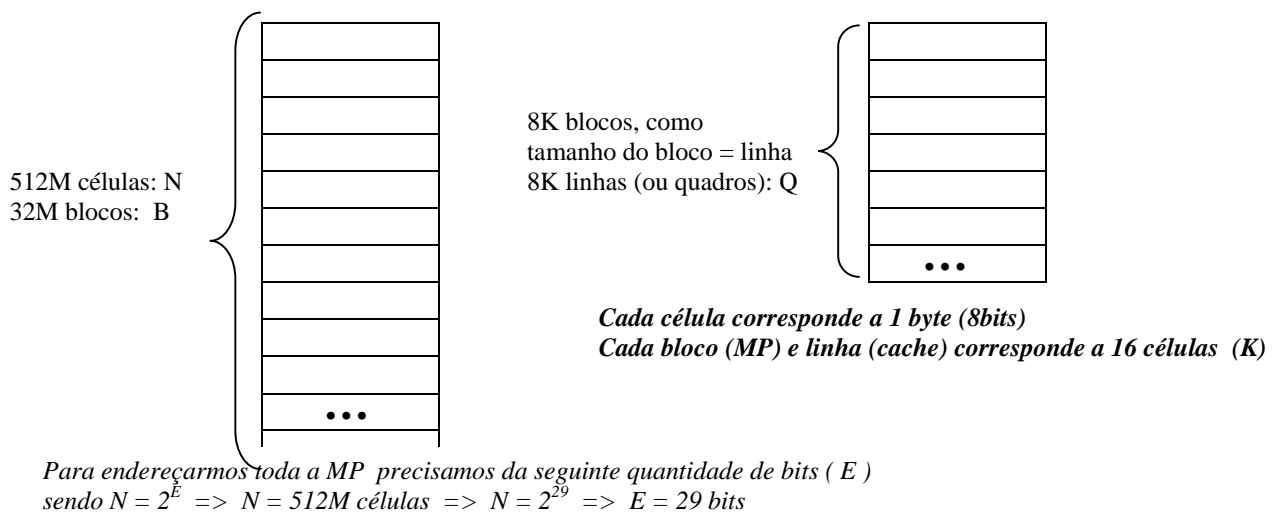
**Memória Principal**

- ⇒ Tamanho da memória (em bytes) = 512Mbytes, como 1 célula referencia a 1 byte, temos  $N = 512M$  células
- ⇒ Será organizada em blocos de 16 bytes, como 1 célula = 1 byte, temos cada bloco = 16 células,  $K = 16$
- ⇒ Sendo  $N$  o tamanho endereçável da memória e  $K$  que é a quantidade de células por blocos temos:  
 $N = 512M$  células e  $K = 16$  células / blocos o total de blocos da MP (  $B$  ) será:  
 Total de blocos:  $B = N / K \Rightarrow B = 512M \text{ células} / 16 \text{ células/bloco} \Rightarrow B = 32 M \text{ blocos}$

**Memória Cache**

OBS: O  $K$  (quantidade de células/bloco) tem de ser igual a MP.

- ⇒ Tamanho da memória cache (em blocos ou linhas)  $\Rightarrow Q = 8K$  blocos
- ⇒ Tamanho da memória cachê em células =  $Q \times K = 8K \text{ blocos} \times 16 \text{ células/blocos} = 128K \text{ células}$



**b) Mapeamento totalmente associativo.**

**Memória Principal**

=>  $N = 512M$  células

=>  $K = 16$  bytes por bloco

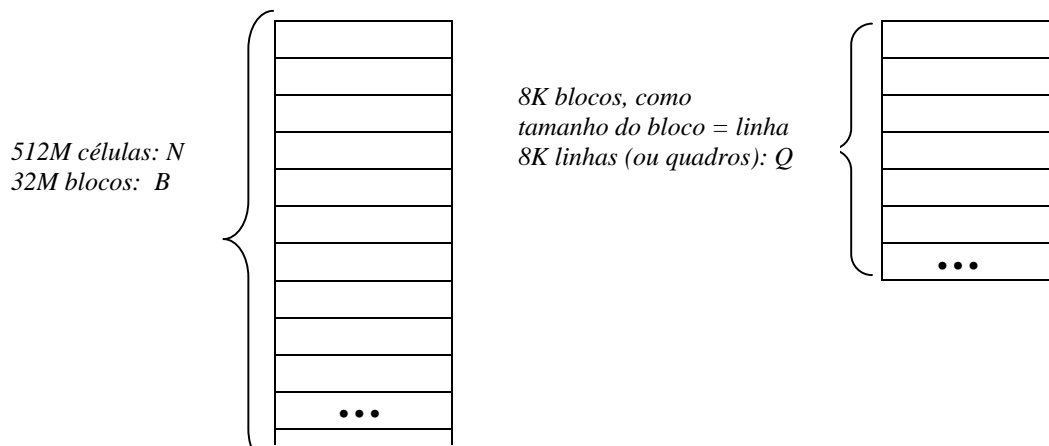
=>  $B = 32M$  blocos

**Memória Cache**

OBS: O  $K$  (quantidade de células/bloco) tem de ser igual a MP.

=>  $Q = 8K$  blocos

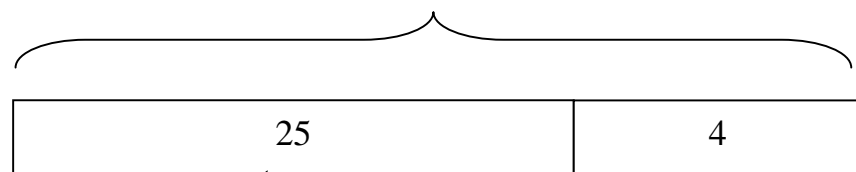
=> Tamanho da memória cache =  $128K$  células



Para endereçarmos toda a MP precisamos da seguinte quantidade de bits:  $E = 29$  bits

Como o bloco pode ser alocado em qualquer posição da memória cache a tag indicará qual dos blocos da MP está alocado naquela posição da memória cache

Tamanho do endereço da MP = 29 bits



Corresponde ao bloco da MP:  
 $tag = B = 32M = 2^{25}$

Corresponde ao endereço da palavra:  
 $16$  células por bloco =  $2^4$

c) **Mapeamento associativo por conjunto, onde cada conjunto possui quatro linhas, cada uma de um bloco.**

#### Memória Principal

=>  $N = 512M$  células

=>  $K = 16$

=>  $B = 32 M$  blocos

#### Memória Cache

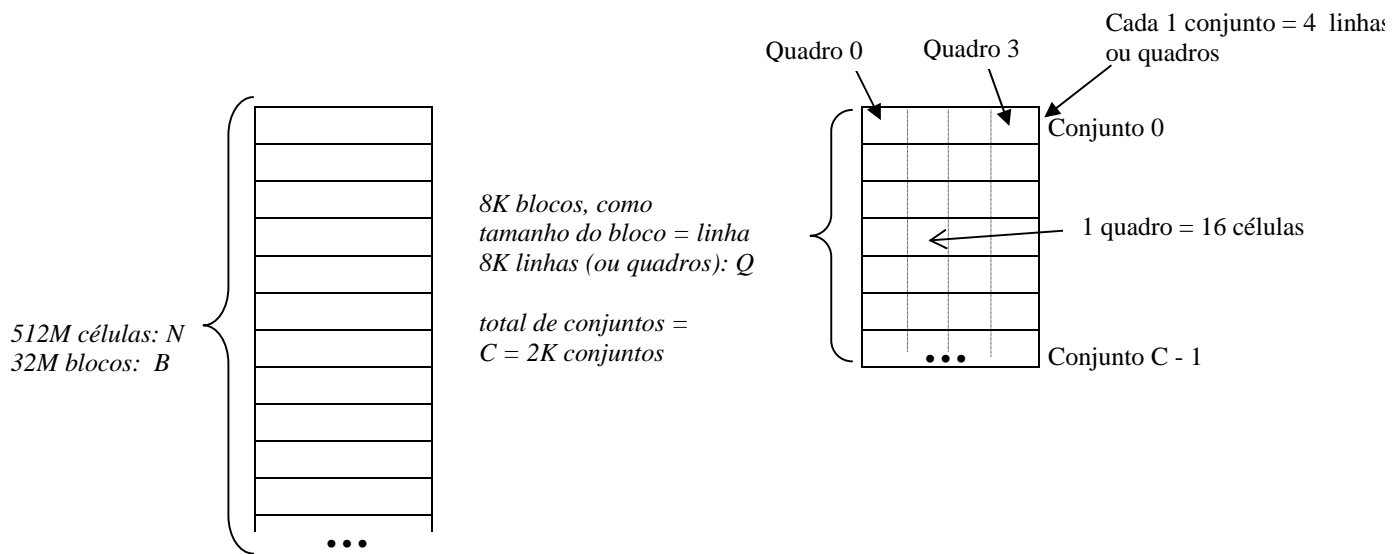
OBS:  $O \cdot K$  (quantidade de células/bloco) tem de ser igual a MP.

=>  $Q = 8K$  blocos

=> Tamanho da memória cache =  $128K$  células

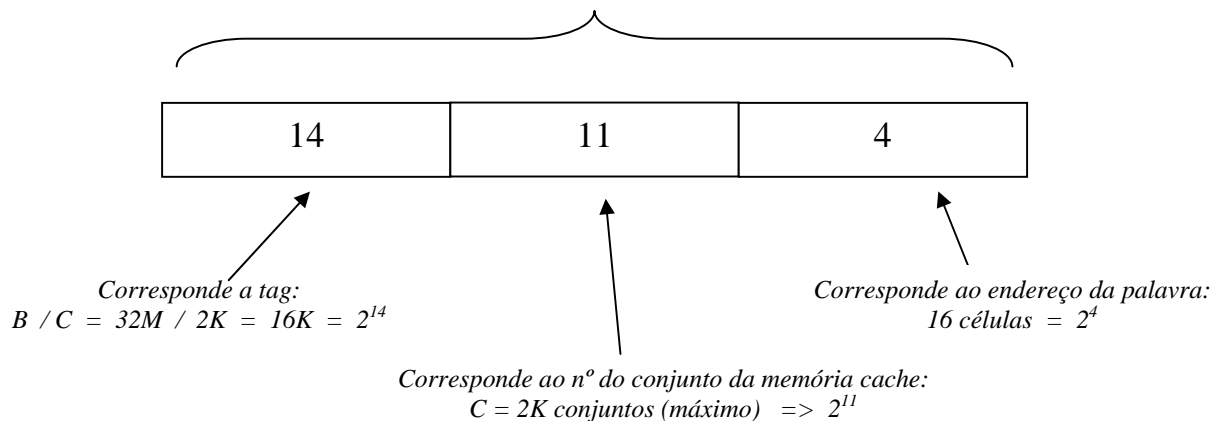
=> 1 conjunto = 4 linhas (ou quadros) =>

Total de conjuntos =>  $C = 8K \text{ blocos} / 4 \Rightarrow C = 2K \text{ conjuntos}$



Para endereçarmos toda a MP precisamos da seguinte quantidade de bits:  $E = 29$  bits

Tamanho do endereço da MP = 29 bits





**9. (1,0) Explique em detalhes a organização hierárquica do subsistema de memória nos computadores atuais.**

*O subsistema de memória é interligado de forma bem estruturada e organizado hierarquicamente em uma pirâmide com os níveis descritos a seguir.*

*No topo da pirâmide teríamos os registradores, que são pequenas unidades de memória que armazenam dados na UCP. São dispositivos de maior velocidade com tempo de acesso em torno de 1 ciclo de memória, menor capacidade de armazenamento além de armazenar as informações por muito pouco tempo.*

*Em um nível abaixo teríamos a memória cache, cuja função é acelerar a velocidade de transferência das informações entre UCP e MP e, com isso, aumentar o desempenho do sistema. A UCP procura informações primeiro na Cache. Caso não as encontre, as mesmas são transferidas da MP para a Cache. A cache possui tempo de acesso menor que a da Memória principal, porém com capacidade inferior a esta, mas superior ao dos registradores e o suficiente para armazenar uma apreciável quantidade de informações, sendo o tempo de permanência do dado menor do que o tempo de duração do programa a que pertence.*

*Abaixo da memória cache teríamos a memória básica de um sistema de computação, que é a memória principal. Dispositivo onde o programa (e seus dados) que vai ser executado é armazenado para que a UCP busque instrução por instrução para executá-las. A MP são mais lentas que a cache e mais rápidas que a memória secundária, possui capacidade bem superior ao da cache e os dados ou instruções permanecem na MP enquanto durar a execução do programa.*

*Finalmente, na base da pirâmide teríamos a memória secundária, memória auxiliar ou memória de massa, que fornece garantia de armazenamento mais permanente aos dados e programas do usuário. Alguns dispositivos são diretamente ligados: disco rígido, outros são conectados quando necessário: disquetes, fitas de armazenamento, CD-ROM. São os mais lentos em comparação com os outros níveis de memória, mas possuem a maior capacidade de armazenamento e armazenam os dados de forma permanente.*

**10. (1,0) Faça uma pesquisa e explique a organização de memórias nos processadores Multicore do Intel Xeon.**

*Fonte de consulta: [www.clubedohardware.com.br](http://www.clubedohardware.com.br)*

*O termo “Xeon” (pronuncia-se “zión”) foi incluído pela Intel aos seus processadores que são voltados para o mercado de servidores e estações de trabalho. Esses processadores reconhecem mais memória RAM, permitem trabalhar em ambiente multiprocessado (isto é, com placas-mãe com vários processadores instalados sobre ela) e possui um desempenho maior que os processadores voltados para o mercado doméstico.*

**ARQUITETURAS MULTICORE XEON COMERCIAIS:**

*=> Xeon de 2 núcleos*

*A tecnologia de dois núcleos traz dois processadores inteiros dentro de um mesmo invólucro. Como os processadores Xeon de núcleo duplo modelos 50xx, 70xx e 71xx têm a tecnologia HyperThreading – que simula a existência de dois processadores em cada núcleo – o sistema operacional reconhece cada processador Xeon de núcleo duplo como sendo quatro processadores. Assim, em um servidor com dois processadores Xeon de núcleo duplo, o sistema operacional reconhecerá oito processadores (quatro núcleos, dois por processador, e dois processadores lógicos por núcleo).*

*Os processadores Xeon de dois núcleos das séries 50xx, 70xx e 71xx são baseados na microarquitetura do Pentium 4 (NetBurst) e por isso possuem a tecnologia HyperThreading, que não está presente na microarquitetura Core.*

*Os processadores Xeon 31xx e 52xx, assim como os modelos 30xx, 51xx e 72xx, são baseados na microarquitetura Core. A principal diferença entre esses modelos é a tecnologia de fabricação. Enquanto os modelos 30xx, 51xx e 72xx usam o processo de fabricação de 65 nm, as séries 31xx e 52xx usam o novo processo de 45 nm.*

*Principais características dos Xeon das séries 50xx, 70xx e 71xx:*

- *Tecnologia de dois núcleos*
- *Mesma arquitetura interna no Pentium 4 (NetBurst)*

- **Cache L1 de dados de 16 KB e cache de execução de 150 KB.**
- **Cache L2 de 2 ou 4 MB compartilhado**
- **Cache L3 interna de 4 MB, 8 MB ou 16 MB (somente nos modelos 71xx).**
- **Suporte a multiprocessamento simétrico com até dois processadores por placa-mãe.**

*Principais características dos Xeon 30xx, 51xx e 72xx Microarquitetura Core*

- *Tecnologia de dois núcleos*
- *Microarquitetura Core*
- *Tecnologia de fabricação de 65 nm*
- **Cache L1 dividido, sendo 32 KB para dados e 32 KB para instruções por núcleo**
- **Cache L2 de 4 MB compartilhado entre os núcleos**

*Principais características dos Xeon 31xx e 52xx*

- *Tecnologia de dois núcleos*
- *Microarquitetura Core*
- *Tecnologia de fabricação de 45 nm*
- **Cache L1 dividido, sendo 32 KB para dados e 32 KB para instruções por núcleo**
- **Cache L2 de 6 MB compartilhado**

**=> Xeon de 4 núcleos**

Os processadores Xeon Séries 33xx e 54xx, assim como os modelos 32xx, 53xx e 73xx, são baseados na microarquitetura Core, a mesma usada pelos processadores Core 2 Duo Enquanto os modelos 32xx, 53xx e 73xx usam o processo de fabricação de 65 nm, as séries 33xx e 54xx usam o novo processo de 45 nm..

Os quatro núcleos dos processadores Xeon 32xx, 53xx e 73xx bem como as séries 33xx e 54xx são obtidos a partir de duas pastilhas de dois núcleos cada, assim como ocorre com os modelos descritos na página anterior. Com isso, o cache L2 desses processadores não é compartilhado entre todos os seus núcleos: os núcleos 1 e 2 compartilham um mesmo cache L2, enquanto que os núcleos 3 e 4 compartilham um outro cache L2. O valor divulgado é o valor total (soma dos dois caches). Leia o nosso artigo *Visão Geral dos Futuros Processadores de Quatro Núcleos da Intel* para uma explicação mais detalhada sobre a arquitetura usada por estes processadores.

*Principais características dos processadores Xeon das séries 32xx, 53xx e 73xx:*

- *Microarquitetura Core*
- *Tecnologia de quatro núcleos*
- **Tecnologia de fabricação de 65 nm**
- **Cache L1 dividido, sendo 32 KB para dados e 32 KB para instruções por núcleo**
- **Cache L2 de 4 MB, 6 MB ou 8 MB, dependendo do modelo, dividido em dois**

*Principais características dos processadores Xeon 33xx e 54xx:*

- *Microarquitetura Core*
- *Tecnologia de quatro núcleos*
- **Tecnologia de fabricação de 45 nm**
- **Cache L1 dividido, sendo 32 KB para dados e 32 KB para instruções por núcleo.**
- **Cache L2 de 6 MB ou 12 MB dividido em dois**