.

Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina: Organização de Computadores
AP1 1° semestre de 2015.

Nome –

Assinatura –

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
3. Você pode usar lápis para responder as questões.
4. Ao final da prova devolva as folhas de questões e as de respostas.
5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
6. Respostas sem desenvolvimento ou justificativas não serão consideradas

---

1. (2,3) Suponha que você deve projetar uma máquina com as seguintes especificações:

   - Capaz de endereçar 64 M células de memória principal, sendo que cada célula armazena 2 bytes.
   - Deve possuir os registradores RDM (utilizado para enviar e receber dados para/de o barramento de dados), REM (utilizado para enviar endereços no barramento de endereços), CI (utilizado para indicar o endereço da instrução a ser lida da memória) e RI (utilizado para armazenar uma instrução).
   - Cada instrução deve conter um código de operação, dois operandos e um registrador como mostrado abaixo:

Cód. Oper	Operando 1	Operando 2	Reg.

   onde o Operando 1 e o Operando 2 são endereços da memória principal e Reg. é o identificador de um Registrador, sendo que a máquina possui 16 registradores.
   - Deve poder ter um máximo de 256 códigos de operação diferentes.

   a) (0,2) Indique qual deve ser o tamanho mínimo em bits do REM

   Memória com 64Mcélulas => N = 64M células
   tamanho mínimo do REM será o tamanho do barramento de endereços necessário para endereçar toda a memória.
   Barramento de endereços (BE) = $\log_2 N = \log_2 64M = 26$ bits
   REM = tamanho do BE = 26 bits

b) (0,3) Indique qual deve ser o tamanho mínimo em bits do do barramento de endereços.

tamanho do BE = 26 bits

c) (0,6) Calcule o número de células que uma instrução necessita para ser armazenada.

Cada instrução = código de operação + 3 operandos
1o. e 2o. operando = endereço de uma célula = 26 bits
3o. operando = endereço de um registrador = 4 bits (total de 2^4 = 16 registradores)
cod.operação = tamanho necessário para 256 códigos diferentes = 8 bits
tamanho da instrução = 8 + 26 + 26 + 4 = 64 bits
Cada célula tem 2 bytes, logo uma instrução necessita de 4 células.

d) (0,6) Indique **o tamanho do RDM e do barramento de dados** de modo que a Unidade Central de Processamento obtenha uma instrução da memória principal realizando somente um acesso à memória principal.

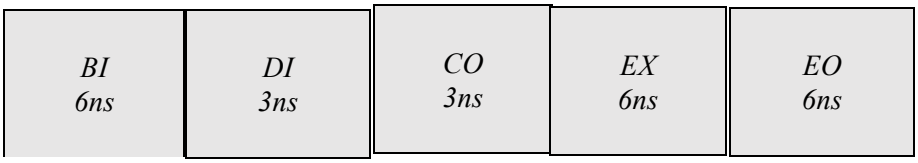
RDM = barramento de dados = tamanho necessário para transferir uma instrução
RDM = barramento de dados = 64 bits (4 células).

e) (0,6) Calcule a capacidade de armazenamento em bits dos registradores RI e CI, utilizando-se os valores calculados nos itens anteriores.

CI = tamanho necessário para endereçar toda a memória = 26 bits
RI = tamanho necessário para uma instrução = 64 bits

2. (2,5) Considere uma máquina que pode ter seu ciclo de busca e execução de uma instrução dividido em 5 estágios totalmente independentes: Busca de Instrução (BI), Decodificação (DI), Cálculo de Endereços de Operandos (CO), Execução (EX) e Escrita de Operandos(EO). Cada um dos estágios BI, EX e EO possui a duração de 6 ns e cada estágio DI e CO tem duração de 3 ns. Cada instrução desta máquina precisa executar os 5 estágios que serão sempre executados na seqüência BI, DI, CO, EX e EO.

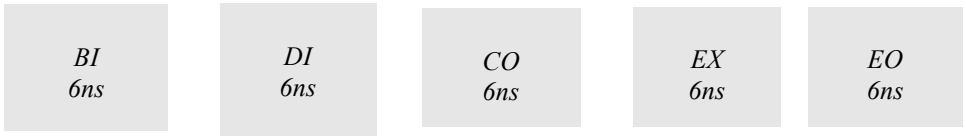
a) (0,2) Uma implementação desta máquina foi realizada de modo que cada instrução deve ser completamente realizada em um único ciclo de relógio. Calcule a duração do ciclo de relógio que esta implementação deve possuir. Lembre-se que todas as instruções necessitam dos 5 estágios.



24 ns

b) (0,5) Como cada estágio é independente um do outro, deseja-se implementar uma nova arquitetura utilizando-se um pipeline de 5 estágios. Nesta nova implementação cada estágio do pipeline deve ser executado em um ciclo de relógio. Calcule a duração do ciclo de relógio que esta implementação pipeline deve possuir.

1º. Estágio	2º. Estágio	3º. Estágio	4º. Estágio	5º. Estágio
6ns	3ns	3ns	6ns	6ns



Ciclo de relógio será igual ao tempo para execução do estágio com maior tempo de execução = 6 ***ns.***

c) (0,5) Considere um programa que necessita executar 100 instruções. Calcule o tempo de execução deste programa na máquina do item a e na máquina do item b.

Seja Tex = tempo de execução de uma instrução = número de estágios x ciclo de relógio (determinado nos itens anteriores)

Para o item a (sem pipeline) :
Tex = 1 estágio x 24ns = 24ns
Ttotal = 100 instruções x Tex = ***2400 ns***

Para o item b (pipeline: 5 estágios) :
Tex = 5 estágios x 6ns = 30ns
Ttotal = Tex + 99 x tempo de 1 estágio
Ttotal = 30ns + 99 x 6ns = ***624 ns***

d) (0,5) Considere um programa que necessita executar 10000 instruções. Calcule o tempo de execução deste programa na máquina do item a e na máquina do item b.

Seja Tex = tempo de execução de uma instrução = número de estágios x ciclo de relógio (determinado nos itens anteriores)

Para o item a (sem pipeline) :
Tex = 1 estágio x 24ns = 24ns
Ttotal = 10.000 instruções x Tex = ***240.000 ns***

Para o item b (pipeline: 5 estágios) :
Tex = 5 estágios x 6ns = 30ns
Ttotal = Tex + 9999 x tempo de 1 estágio
Ttotal = 30ns + 9999 x 6ns = ***60.024 ns***

e) (0,8) Compare as diferenças dos tempos de execução obtidos pela máquina do item a e a do item b para os programas dos itens c e d.

O uso do pipeline permite que mais de uma instrução seja executada ao mesmo tempo reduzindo, assim, o tempo total para a execução dos programas. Esta afirmação pode observada nos tempos de execução calculados nos itens c e d para a máquina do item b (com pipeline) que é bem inferior quando comparada com a máquina do item a (sem pipeline).

3. (1,2) Explique a função de cada um dos seguintes barramentos vistos em aula: barramento de dados, barramento de endereços e barramento de controle.

- *Barramento de dados tem o objetivo de interligar a CPU à Memória principal, por ele trafegam os dados ou instruções da memória para a CPU (processo de leitura em MP) ou da CPU para a memória (processo de gravação em MP).*

- *Barramento de endereços tem o objetivo de interligar a CPU à Memória principal, por ele trafegam os endereços originados da CPU para a memória principal, a transferência é unidirecional.*

- *Barramento de Controle interliga a CPU com diversos componentes do sistema computacional como a MP e o chipset. Trafegam ali sinais de controle como: leitura, escrita, wait, clock, IRQ, entre outros.*

4. (2,0) Explique a hierarquia de memória de um computador, descrevendo cada nível.

Podemos ilustrar essa hierarquia de memória na forma de uma pirâmide dividida em 4 níveis. 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.

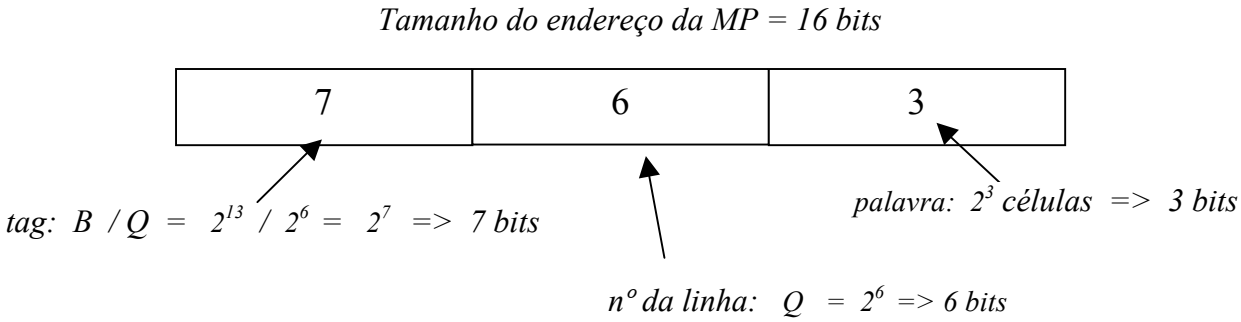
Finalmente, na base da pirâmide teríamos a memória secundária, memória auxiliar ou memória de massa, que fornece garantia de armazenamento mais permanente aos dados e programas do usuário. Alguns dispositivos são diretamente ligados: disco rígido, outros são conectados quando necessário: disquetes, fitas 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.

5. (2,0) Explique através de exemplos os seguintes tipos de mapeamento em memória cache:

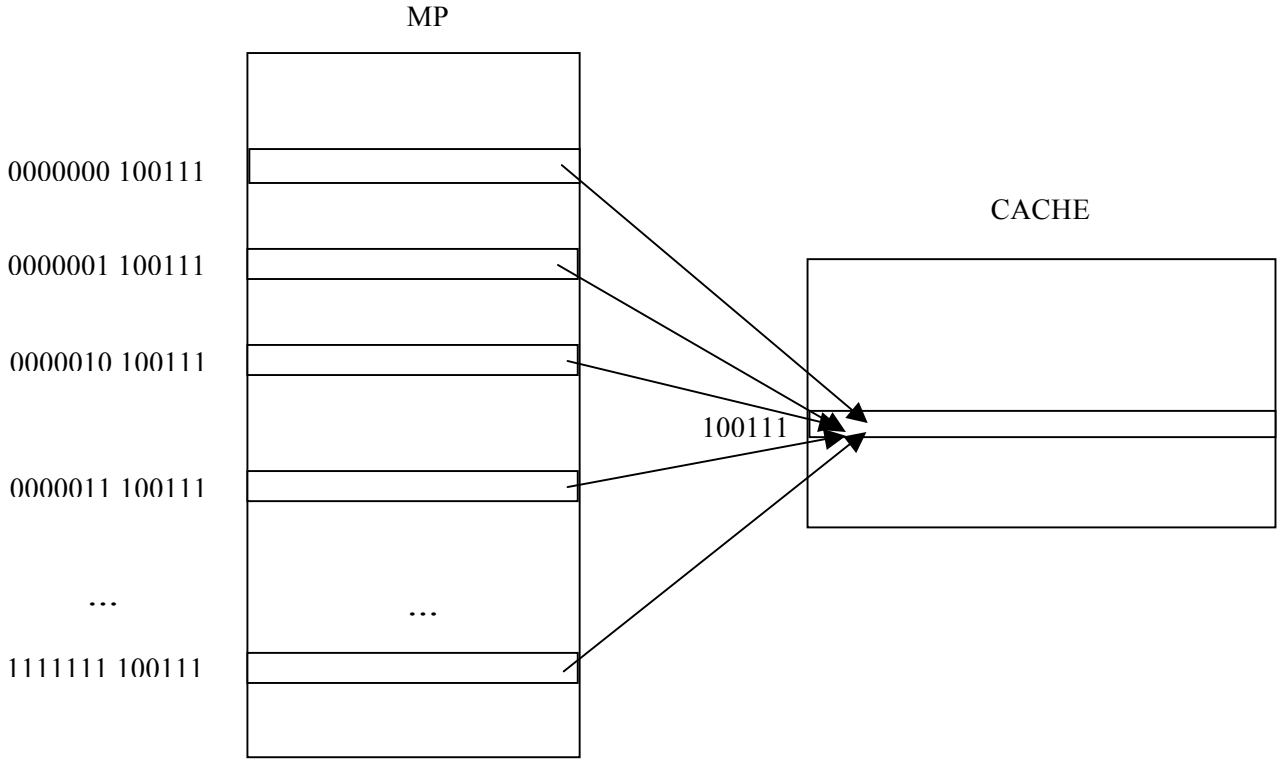
a) Mapeamento direto.

A técnica de mapeamento direto é simples e pouco dispendiosa, pois neste tipo de mapeamento há uma linha fixa na cache para cada bloco da MP. Esta característica também é a sua desvantagem, pois se um programa referenciar palavras repetidamente de dois blocos diferentes, mapeados para a mesma linha, a taxa de acerto será baixa como também será baixo o desempenho na execução de programas..

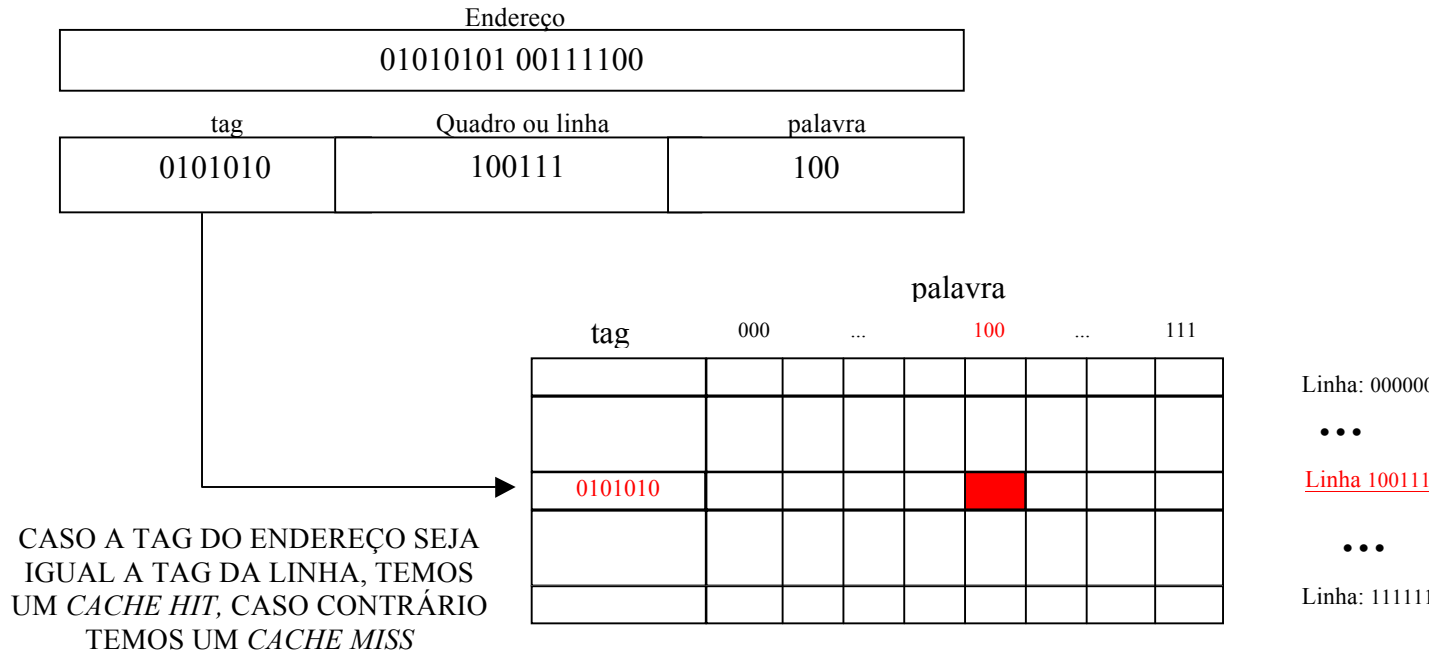
Tomemos como exemplo um sistema computacional com uma MP com $N = 2^{16}$ células, assim, cada endereço possuirá 16 bits, e uma cache com 2^6 linhas, cada linha com 2^3 palavras (assumindo palavra = célula de memória). Como cada bloco possui 2^3 palavras, a MP possui, então, 2^{13} blocos.



Cada Bloco da MP tem sua linha definida na cache que poderá ocupar. Por exemplo, a linha 100111 da cache só poderá ser ocupada, não simultaneamente, por qualquer um dos blocos da MP: 000000100111, 0000001100111, 0000010100111, . ., 1111111100111.



Tendo a CPU fornecido um endereço de MP, inicialmente será procurada na cache a palavra solicita pela CPU. O campo tag da linha será comparada com a tag do endereço da MP, caso não sejam iguais, teremos um cache miss, ou seja, a palavra não está na memória cache havendo a necessidade de transferir o bloco da MP para a cache. Abaixo segue um exemplo com o endereço de MP = 01010101 00111100.

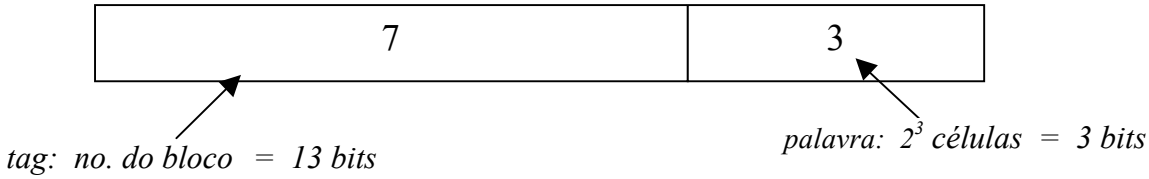


b) No mapeamento totalmente associativo

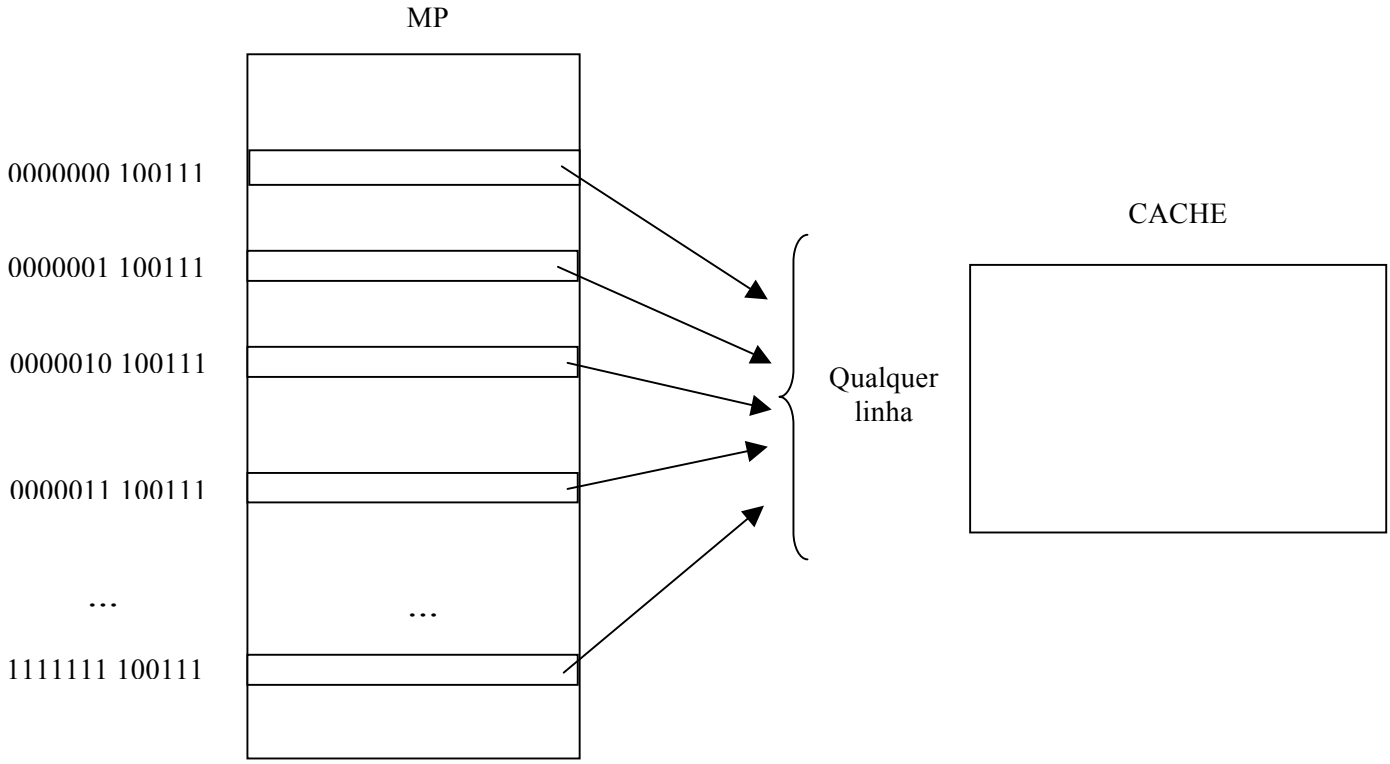
A técnica de mapeamento totalmente associativo possui a máxima flexibilidade no posicionamento de qualquer bloco da MP em qualquer linha da cache. Este mapeamento possui a desvantagem da implementação de um hardware para execução da comparação da tag do endereço solicitado pela CPU, com a tag de cada linha da cache.

Tomemos como exemplo o mesmo sistema computacional do item anterior. Uma MP com N = 2^{16} células, assim, cada endereço possui 16 bits, e uma cache com 2^6 linhas, cada linha com 2^3 palavras (assumindo palavra = célula de memória). Como cada bloco possui 2^3 palavras, a MP possui, então, 2^{13} blocos.

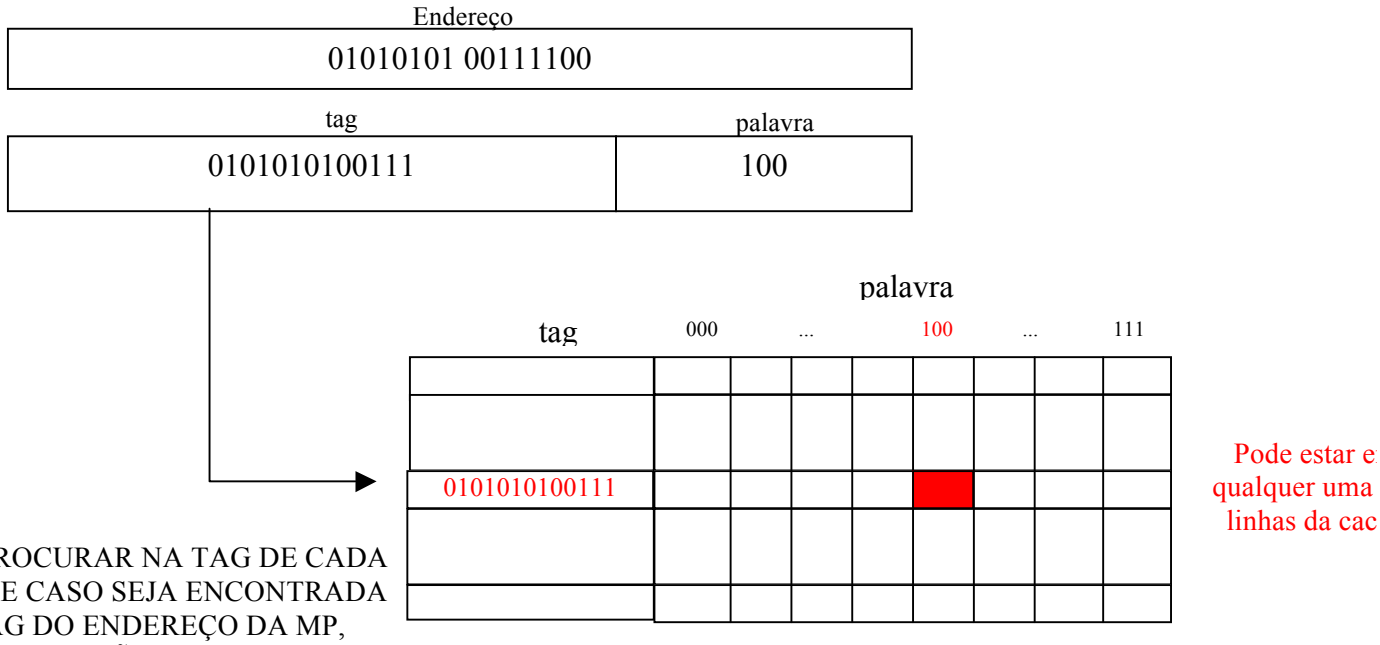
Tamanho do endereço da MP = 16 bits



Cada Bloco da MP poderá ocupar qualquer linha da cache. Quando um bloco da MP for alocado em uma linha da cache, o campo tag da linha assumirá o conteúdo do campo tag do bloco recebido



Tendo a CPU fornecido um endereço de MP, inicialmente será procurada na cache a palavra solicita pela CPU. O conteúdo do campo tag do endereço será procurado linha a linha da cache, caso não seja encontrado teremos um cache miss, ou seja, a palavra não está na memória cache sendo necessário a transferência do bloco da MP para a CPU.



APÓS PROCURAR NA TAG DE CADA LINHA, E CASO SEJA ENCONTRADA A TAG DO ENDEREÇO DA MP, TEREMOS, ENTÃO, UM CACHE HIT