Infraestrutura de Hardware - 3. EE - Memória e Multiprocessadores 2017-1  
  
***1. Considere um sistema computacional que possui um processador superescalar que trabalha na frequência de 2,0 GHz. O espaço de endereçamento virtual é de 8192 Gbytes. A quantidade de discos corresponde ao espaço de endereçamento que é organizado em páginas de 16 Kbytes, porém o sistema possui uma memória principal com 1/1024 da capacidade da memória virtual e duas memórias cache, uma para instruções e outra para dados, uma com 1/512 e outra com 1/1024 da capacidade de armazenamento da memória principal. O sistema ainda inclui uma TLB associativa de 2048 slots.***

1. ***[0,5] Quais as vantagens e desvantagens de se ter um sistema hierárquico de memória?***

**Vantagens:**

Tira proveito do princípio da localidade espacial e temporal: dados usados recentemente e dados próximos aos usados recentemente provavelmente serão usados de novo.

Otimiza as diferentes tecnologias que relacionam custo e tempo de acesso: memórias mais rápidas e menores são mais caras, e memórias grandes porém lentas são baratas. O que se deseja atingir com a hierarquia é o custo e tamanho de uma memória grande como o HD, mas ter o tempo de acesso de uma SRAM (memória do tipo mais caro)

Dado a essa otimização, o desempenho da CPU aumenta: as memórias não se desenvolveram no mesmo passo dos processadores, o que faz com que, sem um planejamento do sistema de memória, a capacidade do processador tenha um gargalo, sendo impedido de operar com máxima eficiência.

**Desvantagens:**

Maior complexidade na implementação e custo elevado.

***2. Para o sistema da questão 1, a cache de instruções é associativa por conjunto com grau de associatividade igual a 8 e a política de escrita é write-through. A cache de dados possui metade da associatividade da cache de instruções e a política de escrita é write-back. Para as duas caches a transferência de informação é feita considerando-se blocos de 8 palavras.***

Descrição do sistema especificado na questão 1:

* Processador de 2.0GHz
* 8192 GB de Espaço de Endereçamento Virtual
* Página de 16Kb
* Memória Principal de 8GB ~> 8192 Mb
* Cache de Instruções de 16Mb
* Cache de Dados de 8Mb
* TLB Fully Associative com 2048 Slots.

1. ***[1,0] Qual o layout e tamanho em (bytes) das caches de instrução e dados?***

**Cache de Instruções:**

* 8-way Set Associative
* Write-through
* Block size 8

Isso nos deixa com uma cache de 25 bytes cada bloco. Como a cache é de 16Mb, temos 224 bytes de memória. Dividindo 224/25, temos que nossa cache terá 219 slots.

Como é 8-way associative, temos 219/23 = 216. Para endereçar esses sets, precisamos de **16 bits (set)**

Para achar a word dentro do bloco de tamanho 8, precisamos de **3 bits (block offset)**

E para words de 4 bytes, precisamos de **2 bits (byte offset).**

Os bits restantes são usados para formar a **Tag (15 bits).**

Layout de Endereço da Cache de Instruções:

|  |  |  |  |
| --- | --- | --- | --- |
| Tag (12) | Set (16) | Block Offset (3) | Byte Offset (2) |

Layout da Cache de Instruções:

Cada **set** será da seguinte forma:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 bit | 12 bits | 256 bits | | | | | | | |
| Valid | Tag | Data | Data | Data | Data | Data | Data | Data | Data |
| x 216 Slots \* 8 sets | | | | | | | | | |

**Cache de Dados:**

* 4-way Set Associative
* Write-back
* Block size 8

Isso nos deixa com uma cache de 25 bytes cada bloco. Como a cache é de 8Mb, temos 223 bytes de memória. Dividindo 223/25, temos que nossa cache terá 218 slots.

Como é 4-way associative, temos 218/22 = 216. Para endereçar esses sets, precisamos de **16 bits (set)**

Para achar a word dentro do bloco de tamanho 8, precisamos de **3 bits (block offset)**

E para words de 4 bytes, precisamos de **2 bits (byte offset).**

Os bits restantes são usados para formar a **Tag (12 bits).**

Layout de Endereço da Cache de Dados:

|  |  |  |  |
| --- | --- | --- | --- |
| Tag (12) | Set (16) | Block Offset (3) | Byte Offset (2) |

Layout da Cache de Dados:

Cada **set** será da seguinte forma:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 bit | 1 bit | 12 bits | 256 bits | | | | | | | |
| Dirty | Valid | Tag | Data | Data | Data | Data | Data | Data | Data | Data |
| x 216 Slots \* 4 sets | | | | | | | | | | |

1. ***[1,0] Para uma determinada aplicação o CPI (ideal) é de 1,5. No entanto a taxa de acertos da cache de instruções e da cache de dados é de 95% e 92%, respectivamente, e o tempo para transferir um bloco é de 100ns. Qual o impacto das caches no CPI se as instruções de acesso à memória para a aplicação correspondem a 32% das instruções?***

Como 32% das instruções são de acesso a memória, apenas 32% estão sujeitas ao miss rate de 8% da cache de dados. Como a cache de instruções é sempre acessada primeiro, ela está sujeita a 100% do miss rate de 5% da cache de instruções.

Sabemos que:

Então

1. ***[0,5] Descreva duas técnicas de otimização que reduzem a taxa de faltas. Informe qual o tipo de falta que a técnica impacta diretamente (compulsória, capacidade ou conflito)***
2. ***[0,5] Descreva duas técnicas de otimização de memória cache que pode reduzir a penalidade, informe qual o tipo de penalidade reduzida pela técnica.***

***3. Numa segunda versão do projeto decidiu-se colocar um segundo nível de cache (cache L2) com o dobro do tamanho da cache de instruções, organizada como mapeamento direto e 10 vezes mais rápida que a memória principal. Com esta nova configuração, a taxa de faltas da cache de instruções (com relação à memória principal) passou a ser de 3% e a de dados passou a ser de 6%. Quais as implicações desta mudança no desempenho do sistema? [1,0]***

Em Instruções, da L1 para L2, ainda temos 5% de misses, afinal, a L1 não mudou de tamanho, porém, agora a penalidade é 10ns. Da L2 para P(rincipal), temos 3%, com penalidade de 100ns, pois a P não mudou de tecnologia. O mesmo vale para Dados, porém com taxas de 8%\*32% e 6%\*32% para L1 e L2, respectivamente.

Então

***4. Considerando o sistema da questão 1 responda as questões a seguir.***

1. ***[1,0] Qual o layout e tamanho da tabela usada na tradução do endereço?***

Com uma página de 16KB, 24.210= 214, precisamos de 14 bits para endereçar o byte dentro da página. Ou seja, o page offset será de 14 bits.

Com 8192GB de memória virtual, precisamos de 213.230 = 243 = 43 bits para endereçar a memória virtual. Desses 43 bits, 14 serão de Page Offset e 29 bits para Virtual Page Number.

Com 8GB de memória virtual, precisamos de 33 bits para endereçá-la completamente. Desses 33 bits, 14 serão de Page offset e 19 de Physical Page Number.

Endereço de Memória Virtual:

|  |  |  |  |
| --- | --- | --- | --- |
| Virtual Page Number(29) | | Page Offset(14) | |

Endereço de Memória Física

|  |  |  |  |
| --- | --- | --- | --- |
| Physical Page Number(19) | | Page Offset(14) | |

Layout da Tabela de Tradução:

|  |  |  |  |
| --- | --- | --- | --- |
| Valid(1) | Dirty(1) | Physical Page Number(19) | |

1. ***[1,0] Qual o tamanho e layout da TLB e quais vantagens e desvantagens de se ter este mecanismo?***

Como nossa TLB tem 2048 (211 ) Slots e é fully associative, não necessitamos de um campo Index no nosso layout de endereço, ficando com o seguinte layout:

Layout da TLB:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Valid(1) | Dirty(1) | Tag(29) | Physical Page Number(19) | |

***5. Descreva o mecanismo de RAID, bem como vantagens e desvantagens deste mecanismo num sistema hierárquico de memória. [1,0]***

Esse mecanismo consiste em conjuntos de discos baratos e redundantes onde os dados persistentes são armazenados e distribuídos. Temos vários níveis:

**RAID 0**, onde não existe redundância, não é seguro, mas é barato.

**RAID 1, também chamado de mirroring,** onde a redundância é por dado, isto é, cada dado tem uma cópia, temos um sistema caro e ineficiente.

**RAID 2,** possui um código de correção de erros, mas ninguém usa.

**RAID 3,** onde conta com um disco que armazena bits de paridade, o que lembra o quarto, que armazena paridade em nível de bloco. No quinto, essas paridades são distribuídas.

**RAID 4,** onde

**RAID 5,** onde

**RAID 6,** onde

<https://pt.wikipedia.org/wiki/RAID#RAID_3>

***a vara q enraba***

***6. Processos concorrentes que se comunicam através de variáveis compartilhadas é um mecanismo suportado por várias linguagens paralelas permitindo explorar paralelismo de thread.***

1. ***[0,5] Por que processadores multi-core são mais adequados para se explorar paralelismo de thread em comparação com processadores superescalares?***

Processadores multi core suportam várias threads por núcleo, onde cada núcleo também pode ser um pipeline superescalar. Então, aplicações que exploram de paralelismo de thread terão muito mais suporte e eficiência, uma vez que podem ser distribuídas entre os vários cores do processador. Se a abordagem superscalar fosse mantida, haveria um gargalo nessa eficiência, uma vez que o número de threads suportada pelo processador seria limitada, devido ao próprio paralelismo do superscalar ser limitado ao paralelismo a nível de instrução.

1. ***[1,0] O que é o problema de coerência de memória e porque este problema existe?***

Em um sistema de multi processadores, cada processador possui sua própria cache privada, isso significa que que mais de um processador pode ter o mesmo dado da memória principal em sua cache. Se um dos processadores alterar esse dado compartilhado, a informação salva nas outras caches se tornam inconsistentes, e esse é o problema de coerência de memória.

1. ***[1,0] Descreva a melhor técnica para resolver o problema da coerência se a memória for centralizada. Quais as vantagens e desvantagens desta técnica?***

Como a memória é centralizada podemos utilizar a técnica snooping, que consiste em ligar todas as caches por um barramento e cada controlador de cache “presta atenção” nos dados que passam pelo barramento. Quando um dos controladores vê um dado que foi requisitado que ele já possui na própria cache, ele pode invalidar, atualizar ou fornecer o valor.

A desvantagem dessa técnica é que ela se torna inviável se o barramento for muito complexo ou se a memória não for centralizada.