1. Considere um sistema computacional que executa aplicações em um processador RISC, nas quais 35% das instruções são de acesso à memória, 50% das instruções são aritméticas e 15% são instruções de desvio. O espaço de endereçamento virtual é de 1024 G bytes. O número de discos corresponde ao espaço de endereçamento que é organizado como páginas de 16K bytes, porém o sistema possui uma memória principal com 1/256 da capacidade da memória virtual e duas memórias cache, uma para instruções e outra para dados, uma com 1/64 e outra com 1/128 da capacidade de armazenamento da memória. O sistema ainda inclui uma TLB associativa de 512 entradas.
    a. Descreva DUAS vantagens e UMA desvantagem de se ter duas caches em vez de uma cache apenas para dados e instruções? (0.5)
    b. Descreva como acontece um acesso a memória cache desde que um endereço é colocado no barramento pelo processador (1.5).

2. Para o sistema da questão anterior, a cache de instruções é associativa por conjunto com grau de associatividade igual a 8 e a política de escrita é write-back. A cache de dados possui metade da associatividade da cache de instruções e a política de escrita é write-through. Para as duas caches a transferência de informação é feita considerando-se blocos de 8 palavras.
    a. Para cada cache do sistema, qual layout do endereço, e quantos bytes possuem cada uma das caches? (1.0).
    b. Para este sistema sabe-se que a taxa de hit da cache de instruções e da cache de dados é de 95 e 93%, respectivamente. Se o hit time para a cache de instruções e de dados é de 1 ciclo e a penalidade é de 32 ciclos, qual o tempo médio de acesso de cada cache? (0.5).

3. Para o sistema da questão anterior decidiu-se duplicar o grau de associatividade da cache de dados e o tamanho do bloco. Quais as implicações destas mudanças no custo do sistema de memória (tamanho da cache) e no tempo médio de acesso de cada cache? Justifique quantitativamente e/ou qualitativamente. (1.5).

4. Para o sistema da questão 1 responda as questões a seguir:
    a. qual o percentual da memória será ocupado pela tabela de páginas? (1.0)?
    b. qual o tamanho da TLB e quais as vantagens e desvantagens de se ter este mecanismo? (1.0)?

5. No caso de uma falta de página, a mesma é carregada na memória pelo DMA, como funciona esta técnica e quais suas vantagens e desvantagens? (1.5)

6. Descreva algumas características de barramentos na realização de operações de Entrada/Saída considerando os aspectos abaixo:

    a. quem sincroniza a transferência de E/S, (0.5)
    b. como os dispositivos conseguem acesso ao barramento, (0.5)
    c. como melhorar a taxa de transferência do barramento. (0.5)

Q2)
End. virtual é 1024GB = 2^1 * 2^3 , tamanho da memória será 2^2 * 2^3 (4GB).

Para a ICache: Teremos 8 palavras de 4 bytes, totalizando 32 bytes, logo serão 3 e 2 bits para referenciar o block offset e byte offset, respectivamente.
Para os conjuntos são necessários 3 bits.
Vale lembrar que o tamanho desta cache é de 64MB, **implicando em 26 bits para** endereçamento, mas desses bits, **3 são desnecessários por conta da associatividade** e (3 + 2 =) 5 são para os offsets, nos levando a 18 bits para representar cada linha. Como são necessários 32 bits para endereçar a memória principal, precisaremos de 18 bits para a tag.

Tag(9)	Set(18)	Block Offset(3)	Byte Offset(2)
32 bits			

Layout da cache de instruções(write back):

Validate(1)	Dirty(1)	Tag(9)	Dados(32*8)

seu tamanho será calculado multiplicando a quantidade de bits de uma “linha” pelo número de linhas:

Para o DCache: também serão 8 palavras de 4 bytes.
Porém, o tamanho desta cache será a metade da cache acima, então serão necessários 25 bits para endereçar cada byte. O grau de associatividade será 4.

Tag(9)	Set(18)	Block Offset(3)	Byte Offset(2)
32 bits			

Layout da cache de instruções(write through):

Validate(1)	Tag(9)	Dados(32*8)

seu tamanho será calculado multiplicando a quantidade de bits de uma “linha” pelo número de linhas:

b)

Q3)

Quantitativamente:

Tag(9)	Set(17)	Block Offset(4)	Byte Offset(2)
32 bits			

Qualitativamente:

Como visto na questão anterior, o número de bits para o Set do endereço é igual ao número de slots/associatividade, isso significa que, quanto maior a associatividade, menos bits serão necessários para o Set e mais bits poderão ser usados na Tag. Isso implica no aumento da cache com o aumento da associatividade, já que é necessário salvar a Tag junto dos dados. No entanto, com o aumento da associatividade, reduzimos a quantidade de misses por conflitos de endereçamento, ao preço de um aumento no custo, já que todas as tags devem ser comparadas simultaneamente em todos os sets, aumentando a complexidade e o Hardware necessário.

Mas o aumento do bloco causa o efeito contrário do aumento da associatividade na tag, isso é, precisamos aumentar o bit de block offset em 1, logo temos que diminuir a tag em 1 também.

No final das contas, dobrar a associatividade e o número de blocos não causa mudanças no tamanho da cache. Ao custo de mais comparadores e um hardware mais complexo, pode haver uma redução na taxa de miss. Com o aumento de blocos, haverá maior aproveitamento da localidade espacial, porém ao preço de um aumento na penalidade caso um miss ocorra, já que mais dados terão que ser transferidos da memória principal.

Q4)

Layout de Endereço Físico

PPN(18)	Page Offset(14)

Layout de Endereço Virtual

VPN(26)	Page Offset(14)

Layout da Tabela de Tradução

Valid(1)	Dirty(1)	PPN(18)

Layout da TLB

Valid(1)	Dirty(1)	Tag(26)	PPN(18)

a) O page offset necessitará de 14 bits para endereçá-lo.
b) Para endereçar a mem. virtual, serão necessários 40 bits, resultando num VPN de 26 bits.
c) Para a memória física, serão necessários 32 bits, resultando num PPN de 18 bits.

Vantagens da TLB:

Aproveitamento da localidade temporal, reduz a penalidade de consultar a tabela de páginas na memória toda vez que houver uma troca de página.

Desvantagens da TLB:

Uma TLB completamente associativa é pequena e cara de ser implementada. O uso de uma TLB aumenta significativamente a complexidade do sistema para lidar com misses e penalidades, já que o lookup deve ser feito como um estado intermediário antes do acesso a memória principal.

Q5)

A técnica DMA serve para transportar dados pelo barramento sem que o processador precise ficar verificando se a operação já foi finalizada. No caso de falta de página, é necessário carregar a página do disco para a memória. Com o DMA, a CPU dá sinal que necessita desses dados mas não executa a transferência, normalmente deixando o DMA como controlador do barramento para realizá-la. Enquanto o DMA está ocupado transferindo os dados, o processador está livre para realizar outras tarefas, sendo essa a principal vantagem dessa técnica. Quando a transferência é finalizada, o DMA envia uma interrupção para sinalizar que o dado está pronto para uso.

As desvantagens de se implementar um DMA são a necessidade de um hardware específico (a unidade de DMA), que aumenta o custo total do sistema e a complexidade, além de contribuir para os problemas de coerência de cache.

Q6)

a)

Sobre sincronia, podemos ter dois tipos de barramentos:

1) Síncrono: o barramento síncrono é o mais simples de ser implementado, porque todo os dispositivos estarão atuando pelo mesmo clock.
2) Assíncrono: este barramento possibilita a individualidade dos dispositivos, isso é, cada dispositivo atua na sua própria velocidade. Para isso ele utiliza de bits de checagem, que sinalizam se o dispositivo quer utilizar o barramento e se ele finalizou de usar o barramento.

b)

No quesito acesso, normalmente no barramento é necessário uma técnica de arbitragem, mais popularmente duas são discutidas:

1) Centralizado: Nesse tipo de arbitragem, é necessário um dispositivo “árbitro” que decide quem deve usar o barramento ou não. Uma das implementações é dar prioridade aos dispositivos de acordo com sua proximidade, quanto mais perto do árbitro, maior sua prioridade.
2) Descentralizado: Já nesse tipo, não é necessário um novo componente que decide tudo, todos os dispositivos são ligados por barramento e conseguem saber se alguém está usando ou não e decidem entre si quem será o próximo a utilizá-lo.

c)

Para melhorar a taxa de transferência, podemos melhorar o barramento das seguintes formas:

1) Podemos dividir o barramento em linhas de dados e de endereços.
2) Transferência por blocos.
3) Aumentar o barramento de dados.