UFPR – Bacharelado em Ciência da Computação
CI212 – Organização e Arquitetura de Computadores — 2016-1

Primeira Prova

1. Esta questão tem dois itens. Para a sequência de instruções mostrada abaixo: (a) indique quais os problemas que podem ocorrer durante a execução das instruções (quais as dependências); e (b) projete o mecanismo necessário para resolver o(s) problema(s). Por *'resolver'* entenda-se que programas com estas sequências serão executados num processador no qual os eventuais problemas são completamente resolvidos em *hardware*. [10 pontos]

```
add r5, r6, r7
lw  r8, 0(r5)
beq r8, r0, dest
```

2. Traduza a função ao lado para *assembly* do MIPS, e também o trecho de código em que ela é invocada. Seu código deve executar corretamente num processador com adiantamento e com *branch-delay slot*. [10 pontos] *Sua resposta DEVE atender às convenções para a implementação de funções.*

```
int Y[NN];
...
x = fun(a,b,Y);
...
int fun(int p, int q, int* v){
   int i; int s = 0;
   for(i = p; i < q; i+=4)
      s += v[i];
   return s;
}
```

3. Esta questão tem três itens. (a) Projete e desenhe um diagrama de uma memória cache com capacidade para 32 palavras, com associatividade binária e blocos de quatro palavras. Supondo que a cache está inicialmente vazia, indique seus conteúdos após a seguinte sequência de referências (a palavras): 3, 11, 61, 29, 17, 12, 21, 5, 14, 23, 62 . Indique qual a estratégia para reposições que você empregou. Para esta sequência, calcule: (b) taxa de faltas; e (c) número de faltas compulsórias. [10 pontos]

Segunda Prova

4. Você foi encarregado de projetar o subsistema de comunicação para uma aplicação embarcada que efetua cálculos complexos e que deve produzir resultados com um mínimo de atraso (*soft real-time*). A frequência do relógio de recepção é 10.000 vezes menor do que a frequência do relógio do processador, as mensagens recebidas tem sempre 512 bytes, e são recebidas byte a byte por uma interface de caractere (similar a uma UART). Sua tarefa é garantir que as mensagens sejam depositadas num armazenador em memória e que a recepção de mensagens interfira o mínimo possível com a computação efetuada pelo processador.

Sua resposta deve conter um diagrama de blocos do seu projeto, uma breve descrição textual do seu funcionamento e um trecho de pseudo-C indicando as interações do processador com a interface de comunicação. [10 pontos]

5. Considere um multiprocessador simétrico com 4 processadores interligados por um barramento. As caches são mantidas coerentes por um protocolo de invalidação, que é mostrado na tabela. Os blocos das caches são de 8 palavras de 32 bits. O protocolo usa escrita forçada na primeira escrita em um bloco; após a primeira escrita, o protocolo usa escrita preguiçosa. Quando um bloco MODificado é substituído na cache, todo o bloco deve ser copiado para a linha em memória. Este protocolo usa os estados EXCLusivo e MODificado para reduzir o tráfego causado por escritas.

A tabela contém uma representação do diagrama de estados e inclui as ações dos controladores das caches. O topo da tabela indica os eventos e a tabela mostra o estado final e a ação do(s) controladore(s) da(s) cache(s) envolvidas. *Snoop*s são operações iniciadas por caches remotas.

estado	RD hit	RD miss	WR hit	WR miss	RD snoop	WR snoop
mod	mod	sha/wrBack	mod	excl/wrBack	sha/wrBack	inv/wrBack
excl	excl	sha	mod	excl	sha	inv
sha	sha	sha	excl/purge	excl/fill	sha	inv
inv	X	sha/fill	X	excl/fill	X	X

wrBack — única cópia de bloco no estado MOD deve ser gravada em memória. Durante atualização, outras caches podem copiar novo valor do barramento;

fill — bloco deve ser preenchido com valor fornecido pelo barramento, que pode ser da memória ou de outra cache;

purge — todas as cópias devem ser invalidadas pelos outros controladores de caches. A cache que emitiu comando mantem cópia exclusiva;

SHAred — cópia válida, só de leitura, compartilhada com outras caches. Antes de atualização, outras cópias devem ser invalidadas;

EXCLusiva — única cópia, memória mantém cópia atualizada. Nova escrita torna MODificada;

MODificada — cópia exclusiva, diferente da memória. Quando for reposta por outro bloco, memória deve ser atualizada (wrBack).

Qual o estado final das caches, supondo o estado inicial mostrado abaixo, e a execução dos seguintes comandos pelos três processadores. Os comandos estão mostrados na ordem absoluta de tempo; os índices dos blocos nas caches são irrelevantes, e as caches são infinitas.

P0: MOD | v[0]..v[7] **P1**: SHA | u[0]..u[7] **P2**: SHA | u[0]..u[7]

P0

```
1  ...
2  ...
3  ...
4  ...
5  ...
6  a=0;
7  for(j=0; j<7; j++)
8    a += t[j] + v[j+4];
9  ...
10 ...
11 ...
12 ...
13 ...
14 ...
15 for(j=0; j<15; j++)
16   u[j+8] = t[j+4]*v[j];
17 ...
18 ...
```

P1

```
1  ...
2  ...
3  ...
4  v[5] = 0;
5  ...
6  ...
7  ...
8  ...
9  ...
10 v[6] = 0;
11 ...
12 ...
13 ...
14 ...
15 ...
16 ...
17 ...
18 v[7] = 0;
```

P2

```
1  for(i=0; i<7; i++)
2    t[i] = u[i]*v[i];
3  ...
4  ...
5  ...
6  ...
7  ...
8  ...
9  ...
10 ...
11 ...
12 for(i=0; i<16; i++)
13   t[i] = u[i]*v[i];
14 ...
15 ...
16 ...
17 ...
18 ...
```

6. Considere um sistema de memória virtual com as seguintes características: (i) endereço virtual de 32 bits (endereço de byte); (ii) páginas com 4 Kbytes; e (iii) endereço físico com 38 bits. A tabela de páginas deve conter bits de `válido`, `modificado` e `usado`.

(a) Qual é o tamanho de uma tabela de páginas linear nesta máquina? [2 pontos]

(b) mostre como implementar a Tabela de Páginas em dois níveis; [5 pontos]

(c) suponha que na sua implementação do item (b), um quarto dos elementos da tabela de primeiro nível sejam inválidos. Quais os tamanhos máximo e mínimo do espaço de endereçamento utilizado pelo programa? [3 pontos]

Exame Final

7. Projete uma TLB com 48 elementos, associatividade ternária, duas traduções em cada bloco, endereços virtuais de 32 bits, endereços físicos de 34 bits, páginas de 4 Kbytes, bits de status apropriados, reposição aleatória. Justifique quaisquer parâmetros de projeto adicionais que sejam necessários ao seu projeto, mostre todas as contas, e desenhe um diagrama detalhado do seu projeto. [30 pontos]

8. Considere o programa ao lado, que é executado no processador segmentado. Explicite quaisquer suposições necessárias para responder à pergunta. [40 pontos]

(i) Considere a CPU segmentada mais simples, sem nenhuma forma de adiantamento nem bloqueios. Acrescente ao código o que for necessário para garantir a execução correta deste programa. Qual o número total de ciclos necessários para armazenar a redução contida em `r5`? *Existe adiantamento através do bloco de registradores.*

(ii) Otimize o código para reduzir o número de ciclos. Qual o novo número total de ciclos? Qual o ganho?

(iii) Repita (ii) considerando adiantamento.

(iv) Qual o ganho considerando as respostas dos itens (ii) e (iii)?

```
        .data
        .org   0x0040.0000
A:      .space 0x1000
B:      .space 0x1000
        .text
        la r20, A
        li r21, 400
        move r5, r0
red:    lw  r10, 0(r20)
        lw  r11, 0x1000(r20)
        add r10, r10, r11
        add r5, r5, r10
        addiu r20, r20, 4
        addiu r21, r21, -4
        bne r21, r0, red
        nop
        sw  r5, 0x2004(r20)
```