

UFPR – PPGInf – BCC
CI702 – Arquitetura de Computadores
CI312 – Arquiteturas Avançadas de Computadores
2012-1

Primeira Prova

1) **Responder individualmente esta questão.** Estudamos várias organizações empregadas para aproveitar o paralelismo no nível de instrução disponível em programas sequenciais. Liste, em ordem crescente de complexidade, as organizações estudadas e indique as principais alterações, com relação à(s) organização(ões) menos complexa(s), de cada “geração” e os ganhos de desempenho obtidos com cada uma das alterações introduzidas naquela geração. Seja conciso na sua resposta. [10 pontos]

As questões que se seguem podem ser respondidas em duplas. As respostas deverão ser enviadas por e-mail para `roberto@inf.ufpr.br` até as 21:00 do dia 11 de maio, sexta-feira. As respostas devem ser entregues em arquivo PDF, em letra de 11 pontos, e com formatação simples.

Responda as questões abaixo supondo que o programa é executado num processador MIPS-8000, que emprega o conjunto de instruções MIPS-64, e é descrito em <http://www.inf.ufpr.br/roberto/ci312/mips8k.pdf>. Suas respostas devem explicitar quaisquer suposições a respeito do comportamento do processador MIPS-8000, e se outra fonte de informações for consultada, ela deve ser citada. Aspectos relacionados a processamento paralelo devem ser ignorados.

Se você usar uma planilha para resolver as questões, entregue-a como parte de sua resposta (prefiro gnumeric).

2) Supondo que não ocorram faltas nas caches nem nas TLBs, qual o CPI ao executar o trecho de código ao lado? Indique claramente o custo (em ciclos) de cada instrução, os riscos entre elas e os atrasos decorrentes de cada risco. Ignore faltas nas caches de instruções. [10 pontos]

```
0.      la      r1, 0x04008000 ; r1 <- &X[0]
1.      la      r3, 0x04008000+24*8192
2. lasso: ld.d   f2, 0(r1)      ; f2 <- X[i]
3.      ld.d   f4, 8(r1)      ; f4 <- X[i+1]
4.      mult.d f6, f2, f12    ; X[i]*A
5.      mult.d f8, f4, f14    ; X[i+1]*B
6.      add.d  f0, f6, f8
7.      st.d   f0, 16(r1)    ; X[i+2] <- f0
8.      addi   r1, r1, 24    ; pointer++
9.      beq    r1, r3, lasso ; X[8192] ?
```

3) Repita a questão anterior, agora considerando faltas nas caches L1 e L2, mas acertos nas TLBs. Suponha que o vetor X[] está carregado em RAM. Compute a taxa de acertos nas duas caches. Ignore faltas nas caches de instruções. [5 pontos]

4) Repita a questão anterior, agora considerando faltas nas caches L1 e L2 e nas TLBs. Suponha que o vetor X[] está carregado em RAM. Compute a taxa de acertos nas duas TLBs. Ignore faltas nas caches e TLBs de instruções. [5 pontos]

5) Reescreva o código para minimizar as bolhas e recompute seus resultados para a questão (2). O ganho de desempenho obtido afeta sua resposta à questão (3)? Como? [5 pontos]

Segunda Prova

O que se segue é um trecho da especificação do trabalho prático desta disciplina.

Sua tarefa é projetar, implementar em VHDL e testar um circuito que efetue uma aproximação

inteira de um “filtro de média” de duas dimensões. Dada uma matriz X de inteiros positivos representados em 8 bits, seu circuito deve computar a matriz Y , segundo:

$$Y(i, j) = 1/2X(i, j) + 1/2\{1/4[X(i-1, j) + X(i+1, j) + X(i, j-1) + X(i, j+1)]\}$$

Seu circuito deve percorrer a matriz de entrada X com $L \times C$ pontos e produzir a matriz de saída Y com os pontos computados como indicado acima. O tamanho máximo dos lados da matriz de entrada é 1024. Os valores dos pontos nas bordas da matriz de entradas são fixos.

No trabalho, as memórias de leitura e de escrita tem largura de 8 bits, e portanto somente um elemento da matriz pode ser lido e/ou escrito a cada acesso à memória. Supondo que o tamanho da matriz de entrada é fixo, indique como tirar proveito do paralelismo nos dados, que é inerente a este algoritmo, para interfaces de memória com 64 bits de largura (8 bytes) – a cada leitura (ou escrita), 8 bytes em endereços contíguos são lidos da (ou escritos na) memória.

1) Descreva um algoritmo para tirar proveito da memória larga e do paralelismo nos dados. Empregue texto e pseudo-C (ou pseudo-VHDL) para formalizar seu algoritmo. Não esqueça de descrever as estruturas de dados usadas pelo seu algoritmo. *Não é necessário detalhar o comportamento do algoritmo nas bordas das matrizes.* [15 pontos]

2) Com base na sua resposta anterior, projete um circuito de dados (*datapath*) que tire máximo proveito do paralelismo nos dados. Indique claramente o que cada unidade funcional faz, bem como as larguras dos sinais que as interligam. *Não é necessário detalhar o comportamento do circuito nas bordas das matrizes.* [10 pontos]

3) Escreva a equação do tempo de acesso para leitura de um setor num disco e (i) explique sucintamente a contribuição de cada termo para o desempenho global do sistema de E/S, e (ii) indique um mecanismo e/ou política para melhorar o desempenho do sistema de E/S. [5 pontos]

4) Considere um disco com 2000 trilhas, numeradas de 0 a 1999. O braço se encontra sobre a trilha 143 e a última requisição foi para a trilha 125. A fila de requisições pendentes, na ordem de chegada das requisições é 86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130. Iniciando da trilha atual, indique a ordem de atendimento das requisições para os algoritmos de escalonamento listados abaixo: [5 pontos]

- (a) FCFS (*First-Come First-Served*);
- (b) SSTF (*Shortest Scan-Time First*);
- (c) SCAN (algoritmo do elevador); e
- (d) C-SCAN (algoritmo do elevador unidirecional).

Exame Final

1) Mostre como o trecho de código abaixo seria escalonado dinamicamente para execução no processador da figura que usa o algoritmo de Tomasulo, com especulação. O processador é capaz de emitir duas instruções por ciclo. As latências são de um ciclo para operações com inteiros (**add**, **sub**, **bra**), dois ciclos para adição em ponto flutuante (**add.d**), e quatro ciclos para a multiplicação em ponto flutuante (**mult.d**). Uma referência a um operando em memória custa três ciclos (**ld.d**, **st.d**). Há duas estações de reserva para as unidades de ponto flutuante e três estações para a ALU de inteiros e três para a unidade de memória. Quantos ciclos são necessários para completar (aposentar) a ÚLTIMA INSTRUÇÃO DA SEGUNDA VOLTA do laço? A resposta deve ser entregue na folha em anexo. [40 pontos]

