

UFPR - Depto de Informática
CI086 - Tópicos em Arquitetura de Computadores

Primeira Prova — 2006-1

1) Considere um circuito combinacional que não é segmentado e têm latência T ; para este circuito, a taxa de computação é $1/T$. O ganho de desempenho de uma implementação segmentada pode ser modelada pela Equação 1, na qual T é a latência do circuito original sem segmentação, S é o atraso adicional introduzido pelos registradores e k é o número de estágios, supondo que a computação possa ser dividida em k estágios iguais.

$$G = \frac{1}{(T/k + S)} \quad (1)$$

Assim, $(T/k + S)$ é o tempo mínimo necessário para a computação em cada estágio do novo sistema, e $1/(T/k + S)$ é sua taxa de computação (*throughput*).

A Equação 1 é similar à Lei de Amdahl. (a) Descreva a relação entre estas duas equações; (b) dê uma explicação intuitiva para a semelhança entre as duas equações; (c) identifique e discuta duas razões pelas quais a Equação 1 somente é útil para aproximações simplistas para o ganho potencial com segmentação; e (d) altere **um** termo da Equação 1 para evidenciar de maneira mais precisa um dos custos adicionais incorridos pela segmentação. Justifique a alteração. [10 pontos]

2) O programa abaixo é executado numa cache infinita, com associatividade total, indexada com endereços virtuais, e supondo que todas as variáveis exceto os vetores $A[]$, $B[]$ e $C[]$ são mantidos em registradores, e que os três vetores são armazenados em posições contíguas em memória. (a) Compute a taxa de faltas para dados; e (b) repita para uma cache com mapeamento direto, capacidade de 8Kbytes e blocos de 64 palavras. [10 pontos]

```
double A[1024], B[1024], C[1024];
for (int i=0; i<1000; i+=2)
    A[i] = 35.0 * B[i] + C[i+1];
```

3) Escreva, em pseudocódigo, uma função com o protótipo abaixo que percorre uma tabela de páginas de dois níveis e retorna **1** se a página está em memória, ou **0** numa falta. O endereço físico é atribuído à `*enderfis` num acerto. Explícite quaisquer suposições que forem necessárias. [10 pontos]

```
int buscatp( void *basetp, void* endervirt, void** enderfis);
```

Segunda Prova — 2006-1

1) **Multicomputadores versus Multiprocessadores** (i) Indique como implementar o modelo de computação paralela por *troca de mensagens* num multiprocessador com *memória fisicamente compartilhada*; (ii) indique como implementar o modelo de computação paralela com *memória compartilhada* num multicomputador construído para suportar o modelo de *troca de mensagens*; (iii) discuta a eficiência que se pode esperar das implementações dos dois itens anteriores. [10 pontos]

2) **E/S por acesso direto à memória** Explique porque, em sistemas com módulos de acesso direto à memória (ADM), os acessos do controlador de ADM à memória principal tem maior prioridade do que os acessos da CPU à memória. [5 pontos]

3) **Sincronização em multiprocessadores** Um *semáforo* é um tipo abstrato de dados composto de um inteiro não-negativo s , e das operações: [15 pontos]

$P(s)$: *if $s > 0$ decrementa s de 1; senão espera*

$V(s)$: *incrementa s de 1 e acorda um dos processos que esperam*

Mostre como implementar eficientemente as operações $P(s)$ e $V(s)$ para execução num multiprocessador com memória fisicamente compartilhada, usando as primitivas definidas abaixo. As operações entre as chaves $\{\dots\}$ são executadas atomicamente.

```
TEST_and_SET(lock) { tmp←mem[lock]; mem[lock]←1; return tmp; }
```

```
RESET(lock) { mem[lock]←0; }
```

```
FETCH_and_ADD(x,a) { tmp←mem[x]; mem[x]←tmp+a; return tmp; }
```

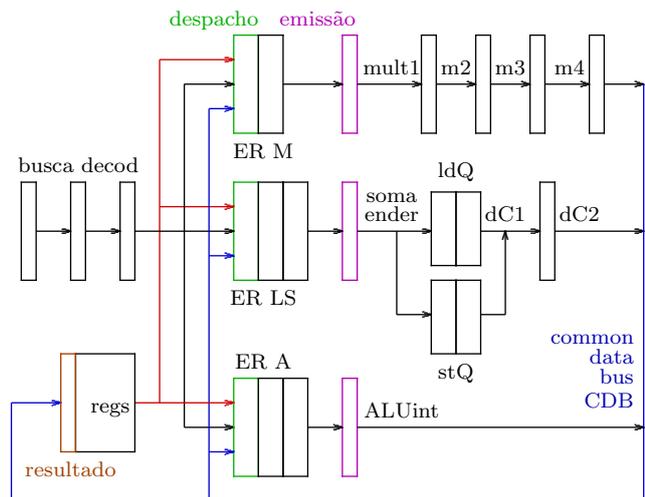
Exame Final — 2006-1

1) **Segmentação** O que são *excessões precisas*? Apresente duas razões que justifiquem por que elas são desejáveis. [20 pontos]

2) **Escalonamento dinâmico com buffer de reordenação** Defina e justifique um conjunto de restrições necessárias à correta execução de programas num processador que efetua escalonamento dinâmico através de um buffer de reordenação, como aquele da lista de exercícios, com suas estruturas de dados são repetidas abaixo. [40 pontos]

| hd | | BdReord + BOrdMem | | | | | | | | rStat | |
|----|---|-------------------|---|---|-------|----|----|----|-----|-------|--|
| tl | # | instr | R | V | ender | EM | EX | CM | reg | T+ | |
| | 1 | ldf f0,X(r1) | | | | | | | f0 | | |
| | 2 | mulf f4,f0,f2 | | | | | | | f2 | | |
| | 3 | stf f4,Z(r1) | | | | | | | f4 | | |
| | 4 | add r1,r1,8 | | | | | | | r1 | | |
| | 5 | ldf f0,X(r1) | | | | | | | CDB | | |
| | 6 | mulf f4,f0,f2 | | | | | | | T | V | |
| | 7 | stf f4,Z(r1) | | | | | | | | | |

| Est Resv | | | | | | | |
|----------|-------|-----|---|----|----|----|----|
| # | UF | bsy | T | Vj | Vk | Qj | Qk |
| 1 | alu | | | | | | |
| 2 | load | | | | | | |
| 3 | store | | | | | | |
| 4 | fp1 | | | | | | |
| 5 | fp2 | | | | | | |



3) **Memória Virtual** (a) Projete e desenhe um diagrama DETALHADO de uma cache de mapeamento de endereços (TLB) com 32 blocos e associatividade quaternária. Cada bloco contém um mapeamento, o processador emite endereços de 32 bits, o endereço físico possui 34 bits, e as páginas virtuais tem 4Kbytes. (b) Indique como um endereço é interpretado pelo controlador da TLB. (c) Qual o tamanho da Tabela de Páginas em memória? (d) Mostre como organizar a Tabela de Páginas em dois níveis. [40 pontos]