

1. Paralelismo

- (a) Why is exploiting *parallelism* important or relevant?
Por que explorar *paralelismo* é importante ou relevante?
- (b) Parallelism is often used in two contexts: the *available parallelism* in programs and the *utilised parallelism* when executing the program on an architecture. In each context, parallelism has various levels and types, what are they and how do the two contexts relate? Paralelismo é frequentemente usado em dois contextos: paralelismo disponível em programas e paralelismo utilizado quando executando o programa em uma arquitetura. Em cada contexto, paralelismo pode ser identificado em vários níveis e tipos. Quais são esses níveis e como os dois contextos mencionados se relacionam?

2. Explorando paralelismo a nível de instrução

- (a) Describe the characteristics and differences between *Scalar ILP processors*, *VLIW processors* and *Superscalar processors*.
Descreva as características e diferenças entre processadores escalares de ILP, processadores de VLIW e processadores super escalares.
- (b) What are *dependencies* and what effect do they have on the three types of processors mentioned in question 2a?
O que são dependências e quais são os efeitos causados nos três tipos de processadores mencionados na questão 2a?
- (c) Apply Bernsteins Conditions to detect the maximum parallelism embedded in the following segment of code. Given that the array size is always a factor of four, maximise the amount of instruction-level parallelism that can be exploited.
Aplique Bernsteins Conditions para detectar o paralelismo máximo embutido no seguinte segmento de código. Dado que o tamanho do array sempre é um fator de quatro, maximize a quantidade de paralelismo a nível de instrução que pode ser explorado.

```
loop: LD    F0, 0(R1)  ; F0=array element
      ADDD  F4, F0, F2 ; add scalar in F2
      SD    0(R1), F4  ; store result
      SUBI  R1, R1, 8   ; decrement pointer
                        ; 8 bytes per word
      BNEZ  R1, loop   ; branch if R1 != 0
```

F_n , where n is a number, represents a 64 bit floating point register.

F_n representa um registrador de ponto flutuante de 64 bits, onde n é um número.

3. Arquiteturas super escalares

- (a) Why is *Shelving* a useful technique for issuing instructions in superscalar processors?
Por que *Shelving* é uma técnica útil para enviar instruções em processadores super escalares?
- (b) How does the processor know when a particular instruction in a shelving buffer can be dispatched to a given execute unit?
Como o processador sabe quando uma instrução particular em *shelving buffer* pode ser despachado para uma determinada unidade de execução?
- (c) *Register renaming* involves six operations or tasks, describe them.
Register renaming envolve seis operações ou tarefas. Descreva-as.

- (d) What is the significance of each of the following three states once an instruction calculates its result?
Qual é o significado de cada um dos três estados, uma vez uma instrução calcula seu respectivo resultado?
- “finished”* (*“terminado”*);
 - “completed”* (*“completado”*);
 - “retired”* (*“aposentado”*).
- (e) How does the *reorder buffer* support sequential consistency, precise interrupts and speculative execution of instructions?
Como o *reorder buffer* suporta consistência seqüencial, interrupções precisas e execução especulativa de instruções?
- (f) In fact, there exists two types and two degrees of sequential consistency.
De fato, existe dois tipos e dois graus de consistência seqüencial.
- What are they?
Quais são eles?
 - What is the current trend in processors and why?
O que é a tendência atual em processadores e por que?

4. Palavras-chaves

- (a) Quais são o nomes em português dos seguintes termos:
- true, output and anti-dependencies;
 - superscalar processors;
 - register renaming;
 - instruction scheduling;
 - floating point unit.

5. Processing Control Transfer Instructions and Code Scheduling for Instruction-level Parallel Processors

- (a) How does processing control transfer instructions and code scheduling affect the performance of superscalar processors?
Como é que o processamento de instruções de transferência de controle e o escalonamento de código influenciam o desempenho de processadores superescalares?

6. As Apresentações

- (a) Summarise one of the presentations that you assisted.
Faça um resumo de uma das apresentações que você assistiu.