UNIVERSIDADE FEDERAL DA BAHIA
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

DISCIPLINA: MATA48 – Arquitetura de Computadores
PROFESSOR: Marcos Ennes Barreto
SEMESTRE: 2019.2

TRABALHO PRÁTICO III (PROVA 2)

Especificação

1. Objetivo geral: resolução de questões sobre paralelismo em nível de instruções (ILP), arquiteturas superescalares, paralelismo em nível de threads (TLP) e de dados (DLP).
2. Cada **aluno ou dupla** deve responder às questões de cada tópico e enviar suas respostas em arquivo específico, conforme especificado no item Formato de Submissão.

Data máxima de submissão: 08/12/2019, 23h59m.

Formato de submissão:

1. arquivo com respostas (.doc ou .pdf ou .rtf).
2. o arquivo deve ter o seguinte nome: MATA48_TP3_ALUNO1_ALUNO2.ext.

Local de submissão:

1. Link disponibilizado na página da disciplina no Moodle.

Critérios de correção:

1. As respostas serão corrigidas considerando-se i) a correta explicação dos conceitos questionados, ii) a cobertura (ou "completude") da resposta, considerando-se o que foi questionado e o quão completa é a resposta, e iii) uso de exemplos ou aplicações práticas (quando cabível).
2. Cada erro de sintaxe será descontado em 0,3 pontos.
3. Respostas idênticas serão zeradas, por caracterizarem plágio.
4. A nota do trabalho corresponderá também à nota da Prova 2, para fins de composição de média.

Divulgação de resultados:

1. Será feita até a data de **13/12/2019**, 23h59m, via Moodle.

Resolução de dúvidas referentes ao trabalho:

1. Acessar o fórum específico para o Trabalho III no Moodle e inserir lá toda e qualquer pergunta pertinente ao trabalho.
2. OBS.: preferencialmente, usar o fórum do Moodle em vez de e-mail para que toda a turma possa interagir e socializar dúvidas e dicas.

TÓPICO 1: paralelismo em nível de instruções (ILP)

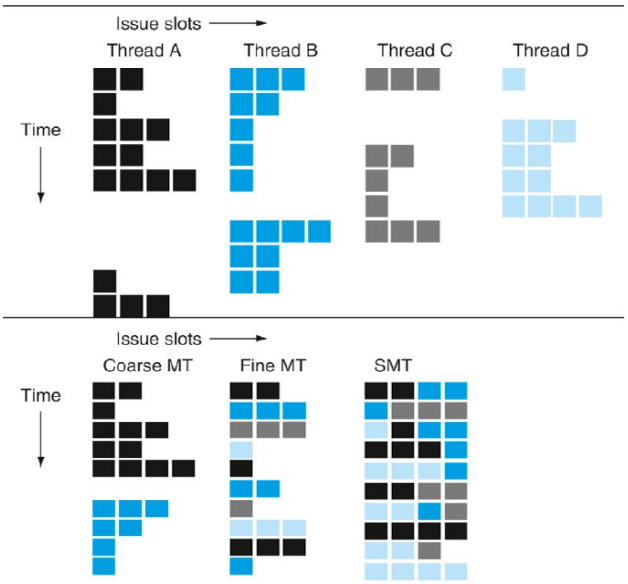
A) O paralelismo em nível de instrução é limitado por diferentes tipos de dependências existentes tanto no hardware quanto no código do programa a ser executado. Explique e dê um exemplo de i) dependência de dados verdadeira (RAW), ii) antidependência (WAR) e iii) dependência de saída (WAW).

B) Explique as diferenças entre multiprocessadores com memória privada e com memória compartilhada. Explique qual é o principal problema a ser tratado em cada classe destes multiprocessadores. Indique como cada tipo de arquitetura pode ser programada (uso de threads, troca de mensagens, acesso global à memória etc). Para multiprocessadores com memória compartilhada, explique as diferenças entre arquitetura UMA (*uniform memory access*) e arquitetura NUMA (*non-uniform memory access*).

C) Explique as características de cada uma das categorias de hardware propostas na Taxonomia de Flynn (SISD, SIMD, MISD e MIMD). Para cada categoria, quando possível, mostre um exemplo de programa ou aplicação (algoritmo ou código de exemplo).

		Data Streams	
		Single	**Multiple**
Instruction Streams	Single	SISD: Intel Pentium 4	SIMD: SSE instructions of x86
	Multiple	MISD: No examples today	MIMD: Intel Core i7

D) Considerando 4 threads (A até D, conforme figura abaixo), explique a diferença entre os modelos de execução: i) paralelismo de grão grosso (*coarse grain*), ii) paralelismo de grão fino (*fine grain*) e iii) *simultaneous multithreading* (SMT).



E) Considere os threads A e B descritas na tabela abaixo, compostos por 4 instruções cada. Considere também 2 organizações de processadores:
- CPU_MT: processador com paralelismo de grão fino que pode executar instruções de 2 threads concorrentemente (tem 2 unidades funcionais), embora somente instruções de um único thread possam ser despachadas em um determinado ciclo.

- **CPU_SMT**: processador SMT que pode executar instruções de 2 threads concorrentemente (tem 2 unidades funcionais) e que despacha instruções de um ou ambos os threads em um determinado ciclo.

Para cada organização (CPU_MT e CPU_SMT), indique i) quantos ciclos demora para executar estes 2 threads; e ii) quantos slots de unidades funcionais ficam ociosos devido às dependências entre instruções?

Thread A	Thread B
A1 – demora 3 ciclos para executar	B1 – demora 2 ciclos para executar.
A2 – demora 1 ciclo. Sem dependências.	B2 – demora 1 ciclo. Depende de B1.
A3 – demora 1 ciclo. Depende de A1.	B3 – demora 1 ciclo. Depende de B2.
A4 – demora 1 ciclo. Depende de A3.	B4 – demora 2 ciclos. Sem dependências.

TÓPICO 2: arquiteturas superescalares

F) Explique as características e diferenças entre arquiteturas superescalares e arquiteturas superpipeline. Dê um exemplo de processador ou arquitetura de cada um destas categorias de hardware.

G) Arquiteturas superescalares empregam três diferentes políticas de emissão e conclusão de instruções. Explique as diferenças entre estas políticas. Usando o código abaixo e considerando i) as dependências verdadeiras entre instruções e ii) um pipeline composto por 2 unidades de decodificação (ED), 3 unidades funcionais de execução (UF) e 2 unidades de escrita/conclusão (EE), mostre como este código pode ser executado a partir das três diferentes políticas de emissão e conclusão.

```
I1. R1 = R1 + R5
I2. R2 = 1
I3. R1 = R5 + 1
I4. R4 = R1 * R2
I5. R3 = R2 – 2
I6. R5 = 3
I7. R6 = R4 + R2
```

TÓPICO 3: paralelismo em nível de threads (TLP)

H) No estudo sobre paralelismo em nível de threads, discutimos sobre threads físicas e threads virtuais, tendo por base a quantidade de threads que um processador é capaz de criar fisicamente e a quantidade de threads que o programador pode especificar no seu código (até a quantidade máxima que a biblioteca de programação paralela suporta). Comente sobre vantagens e desvantagens no uso de threads física e de threads virtuais. Cite exemplos de aplicações mais indicadas para um ou outro tipo de organização de threads.

I) A biblioteca OpenMP implementa cinco diferentes estratégias de escalonamento de laços FOR para um bloco de threads (diretiva omp parallel). Explique as diferenças entre cada um dos cinco tipos de escalonamento (*static, dynamic, guided, auto, runtime*). Considerando o material anexo (OpenMP_Dynamic_Scheduling.pdf), que compara as estratégias de escalonamento *static, dynamic* e *guided*, monte uma tabela com os tempos de execução para cada um dos programas descritos neste material para fins de comparação. Qual configuração (tipo de escalonamento, quantidade de threads e tamanho de *chunk*) resultou em menor tempo de execução?

TÓPICO 4: paralelismo em nível de dados (DLP)

J) Arquiteturas vetoriais foram os primeiros modelos de arquiteturas capazes de explorar o paralelismo em nível de dados, principalmente sobre dados em formato matricial. Comente sobre os principais elementos componentes desta arquitetura (registradores vetoriais, unidades funcionais e unidades de acesso à memória).

K) O tempo de execução de aplicações sobre arquiteturas vetoriais depende, além do tamanho do vetor de dados, das dependências de dados e estruturais. O hardware implementa diferentes estratégias para maximizar o desempenho, dentre elas o conceito de comboio (*convoy*) e o escalonamento por encadeamento (*chaining*). Explique estes dois conceitos e exemplifique com trechos de código.