

**UFPR – Departamento de Informática – BCC**  
**CI212 - Organização e Arquitetura de Computadores**  
**Primeira Prova**

1) Mostre o código MIPS necessário para implementar os comandos em C abaixo:

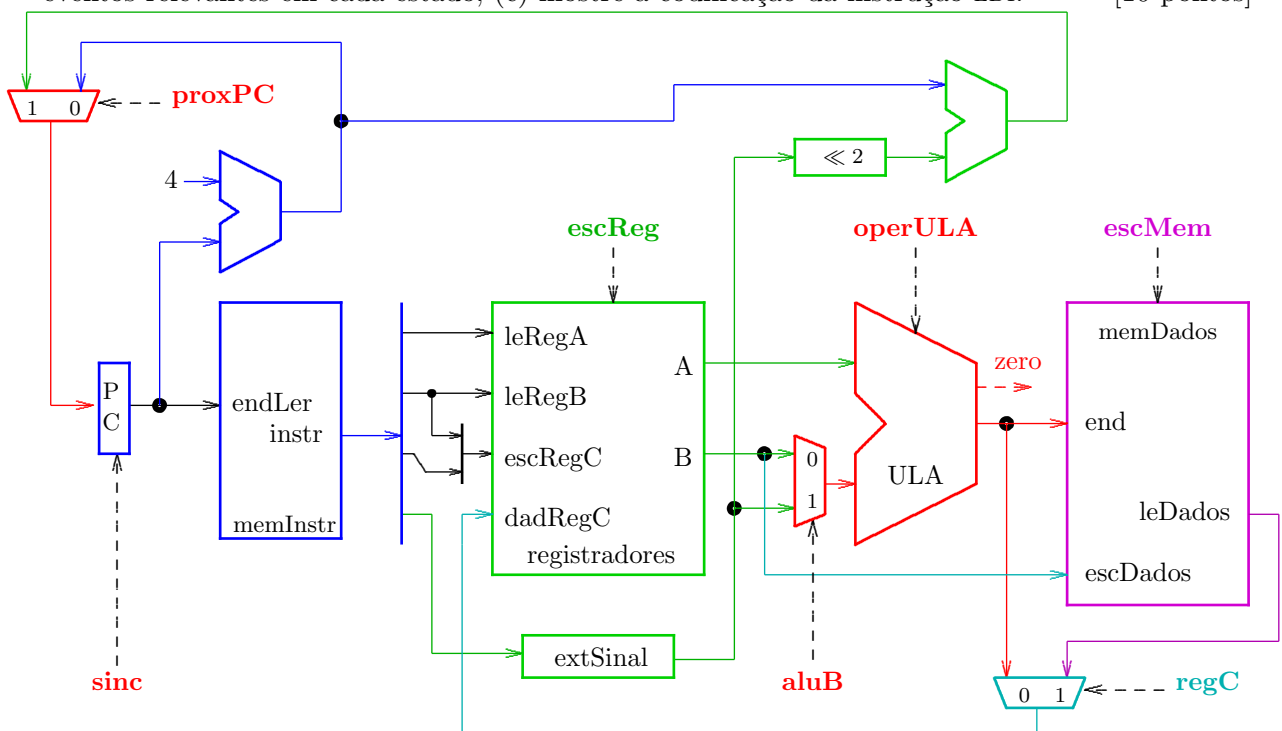
```
int a, b, i, x[NNN], y[MMM];
a = y[65000] / 32;
b = x[ y[i]*16 ];
```

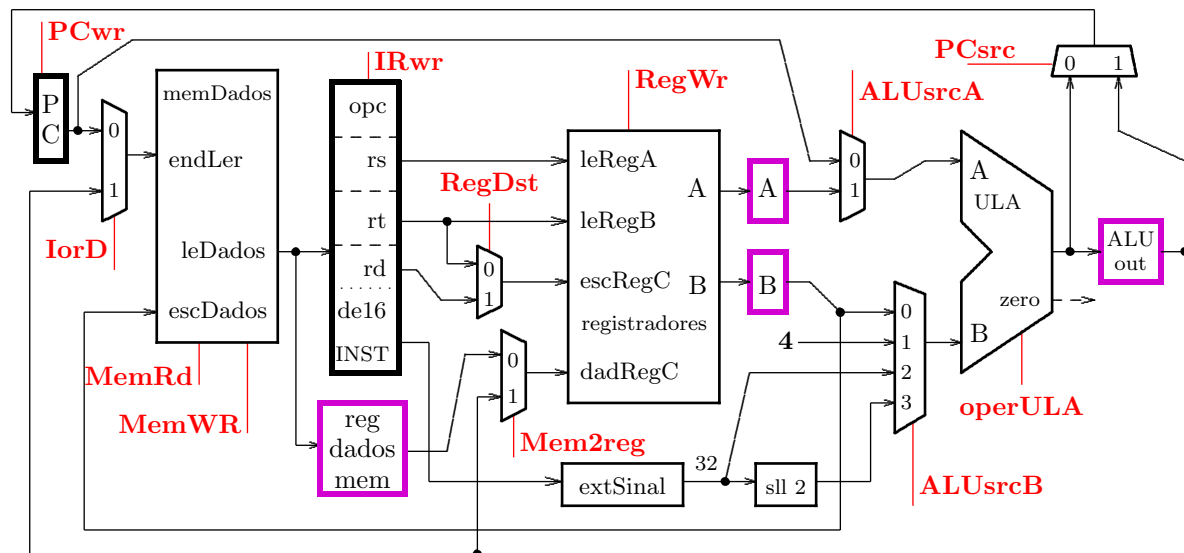
[10 pontos]

As perguntas seguintes devem ser respondidas com o auxílio dos diagramas abaixo.

2) A instrução JAL (*jump and link*) permite desviar para uma função, lembrando do endereço de retorno. A instrução JR (*jump register*) permite carregar o endereço de retorno da função no PC, assim efetuando o retorno de função. (a) Mostre como implementar estas duas instruções num processador com ciclo longo. Acrescente todas as ligações e componentes que julgar necessário; (b) desenhe um diagrama de tempo com a execução desta instrução no processador com suas adições. [10 pontos]

3) A instrução LDI (*load-immediate*) lê um valor imediato de 32 bits do endereço seguinte ao da instrução e o carrega no registrador de destino. (a) Mostre como implementar esta instrução no processador multi-ciclo. Acrescente todas as ligações e componentes que julgar necessário; (b) mostre a seqüência de estados correspondente à execução desta instrução, indicando os eventos relevantes em cada estado; (c) mostre a codificação da instrução LDI. [10 pontos]





### Segunda Prova

1) Considere um processador segmentado em oito estágios: B1, B2, DC, E1, E2, M1, M2, RE. Desvios são resolvidos no estágio E1. Considere ainda a execução de um programa com as seguintes frequências de instruções: 

lw 25%	sw 15%	ULA 30%	desvios 25%	saltos 5%
--------	--------	---------	-------------	-----------

Para esta pergunta não é necessário efetuar a conta, mas sim armá-la e *justificar seus termos*.

- (1a) Considere que a memória tem tempo médio de acesso de 1 ciclo, não há previsão de desvios, que 50% dos desvios são tomados, e que 20% dos *loads* são seguidos por uma instrução de ULA que usa o valor buscado. O adiantamento entre instruções de ULA é perfeito. Qual o CPI? [5 pontos]
- (1b) Considere que a memória é ideal (tempo médio de acesso de 1 ciclo), o circuito de previsão de desvios tem taxa de acerto de 90% mas a recuperação de uma previsão errada custa 5 ciclos, e que 20% dos *loads* são seguidos por uma instrução de ULA que usa o valor buscado. O adiantamento entre instruções de ULA é perfeito. Qual o CPI? [5 pontos]
- (1c) Considere que o circuito de previsão de desvios tem taxa de acerto de 90% mas a recuperação de uma previsão errada custa 5 ciclos, e que o compilador re-escala os *loads* para evitar as paradas. A memória ligada ao circuito de busca é perfeita, mas o circuito de dados é ligado a uma memória cache com taxa de acerto de 90% e penalidade de 10 ciclos. O adiantamento entre instruções de ULA é perfeito. Qual o CPI? [5 pontos]
- (1d) Explique o que ocorre nos dois casos possíveis num desvio tomado previsto corretamente, quando **ou** a instrução de destino existe na cache de instruções (acerto), **ou** a instrução destino está faltando na cache de instruções. [PONTO EXTRA, ATÉ 5 pontos]

2) No trecho de código abaixo, as instruções com sufixo *.s* são instruções de ponto flutuante de precisão simples (operam sobre *float*). As operações de adição e multiplicação de ponto flutuante completam no mesmo tempo que instruções de inteiros. Os registradores *f0..f8* são registradores de ponto flutuante. [10 pontos]

Modifique o trecho de código ao lado para que ele execute da forma mais eficiente possível no processador de oito estágios da questão 1, supondo que o processador não possua adiantamento e nem previsão de desvios, mas que a memória é perfeita. O registrador `r9` contém `r8+1024` no início do laço.

Justifique sua resposta.

```
loop: ld.s f0,0(r4)
      ld.s f2,0(r6)
      mul.s f4,f2,f0
      add.s f6,f4,f8 ; f8 contem escalar
      st.s f4,0(r8)
      st.s f6,8(r8)
      addi r4,r4,8
      addi r6,r6,8
      addi r8,r8,16
      bne r8,r9,loop ; r9 contem limite
```

3) Projete uma cache com 96 Kbytes de capacidade, associatividade ternária, blocos de 32 bytes, e escrita preguiçosa (*write-back*). Indique como um endereço de 32 bits é interpretado pelo controlador da cache. Indique claramente o dimensionamento dos componentes da cache. [5 pontos]

### Terceira Prova

1) Suponha que o tamanho do endereço de um computador seja  $k$  bits (endereço de byte), o tamanho da cache seja  $S$  bytes, o tamanho de um bloco seja  $B$  bytes, e que a cache tem associatividade  $A$ -ária (*A-way set-associative*). Suponha que  $B$  é uma potência de dois ( $B = 2^b$ ). Escreva as quantidades especificadas adiante em termos de  $S, B, A, b$  e  $k$ : [10 pontos]

- (1a) O número de conjuntos na cache (cada conjunto possui  $A$  elementos).
- (1b) O número de bits de índice no endereço.
- (1c) O número de bits necessários para implementar a cache. *Explícite quaisquer escolhas de projeto que sejam necessárias para esta resposta.*

2) Nesta questão pretende-se avaliar o impacto das operações de entrada e saída por programa (*polling* ou *programmed I/O*) para três dispositivos distintos. Suponha que sejam necessários 400 ciclos de relógio para uma operação de consulta (*poll*) —incluindo a transferência para a rotina de consulta, acesso ao dispositivo, e retorno ao programa de usuário— e que o processador executa com relógio de 500 MHz (2ns/ciclo).

Determine a fração do tempo de CPU consumido para os três casos abaixo, supondo que os dispositivos estão potencialmente ocupados todo o tempo. [10 pontos]

- (2a) o *mouse* deve ser consultado 30 vezes por segundo para garantir que nenhum movimento do usuário seja perdido;
- (2b) o controlador de disquete transfere dados ao processador em unidades de 16 bits e tem uma taxa de transferência de 50 KByte/s. Nenhuma transferência pode ser perdida;
- (2c) o disco rígido transfere dados em rajadas de 4 palavras e pode transferir a uma taxa de 4 MByte/s. Nenhuma transferência pode ser perdida.

3) Em sistemas com caches, memória virtual e Controladores de Acesso Direto à Memória (CADM) podem ocorrer problemas na consistência dos dados por conta da atuação em paralelo do processador e do CADM. Explique quais são os problemas que podem ocorrer, e em quais situações eles podem ocorrer. Indique uma possível solução para um dos problemas listados. [10 pontos]

**Exame Final**

1) Descreva TODOS os modos de endereçamento suportados pela arquitetura do MIPS. Desenhe diagramas mostrando os componentes do endereço efetivo de cada um dos modos. [10 pontos]

2) No trecho de código abaixo, as instruções com sufixo `.s` são instruções de ponto flutuante de precisão dupla (operam sobre `double`). As operações de adição e multiplicação de ponto flutuante completam no mesmo tempo que instruções de inteiros. Os registradores `f0..f16` são registradores de ponto flutuante e devem ser usados aos pares (`f0:f1` comportam um `double`). Considere que o circuito de previsão de desvios acerta sempre exceto na última volta do laço, e que a memória é perfeita.

(2a) Re-escreva o código indicando CLARAMENTE as dependências entre as instruções; [10 pontos]	<code>; r9 = 1024</code> <code>loop: ld.s f0,0(r4)</code>
(2b) re-escreva o código para que ele execute da forma mais eficiente possível no processador de cinco estágios do Capítulo 6; [10 pontos]	<code>ld.s f2,0(r6)</code> <code>mul.s f4,f2,f0</code> <code>st.s f4,0(r8)</code>
(2c) indique qual é o ganho de desempenho com relação ao trecho de código original; [10 pontos]	<code>add.s f6,f4,f8 ; f8 contem escalar</code> <code>st.s f6,8(r8)</code>
(2d) desenhe um diagrama do processador e indique quais são os circuitos de adiantamento necessários à execução DESTA programa. [10 pontos].	<code>addi r4,r4,8</code> <code>addi r6,r6,8</code> <code>addi r8,r8,16</code> <code>addi r9,r9,-8</code> <code>bne r0,r9,loop ; r9 contem limite</code>

3) Defina a seqüência de eventos, com TODAS as possibilidades de faltas e acertos nas referências a dados na hierarquia de memória definida abaixo. Para cada evento indique a ação a ser tomada. [40 pontos]

Nível 1: caches separadas para instruções e dados (L1i e L1d), acessadas com endereços virtuais;

Nível 2: TLBs separadas para instruções e dados (Ti e Td);

Nível 3: cache secundária para dados e instruções (L2), acessada com endereços físicos;

Nível 4: memória RAM;

Nível 5: área de swap em disco.

4) Considere um sistema de memória virtual com as seguintes características: (i) endereço virtual de 40 bits (endereço de byte); (ii) páginas com 8 Kbytes; e (iii) endereço físico com 38 bits. A tabela deve conter bits de válido, protegido, executável, sujo e usado.

(4a) Qual é o tamanho total da tabela de páginas nesta máquina? [10 pontos]

(4b) Mostre como implementar a Tabela de Páginas em TRÊS níveis. [10 pontos]