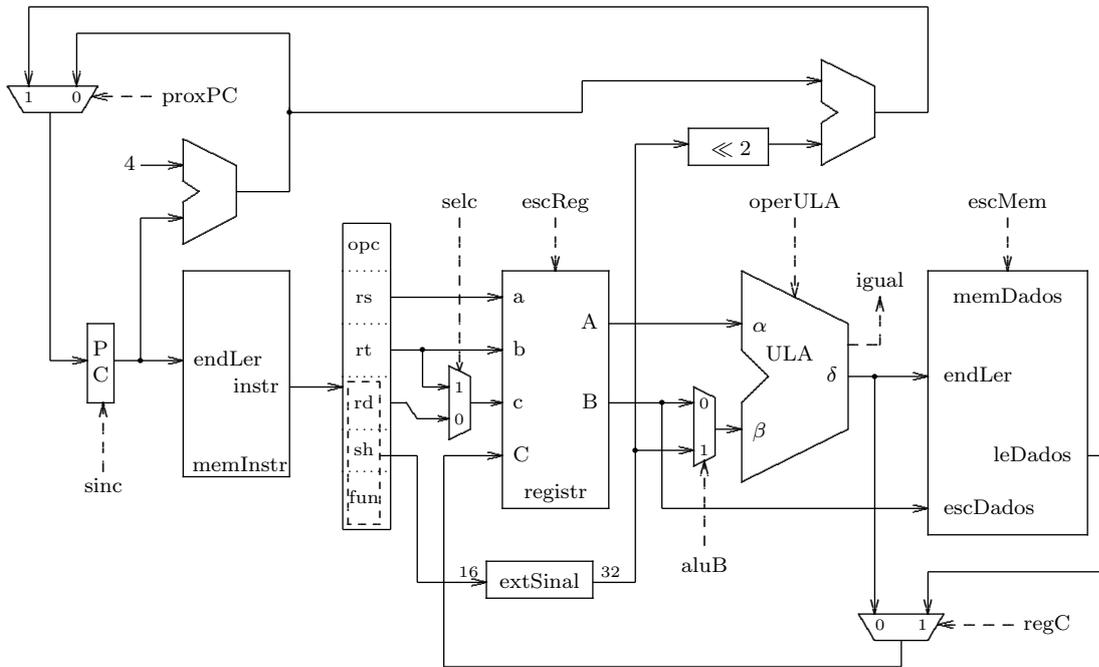


UFPR – Bacharelado em Ciência da Computação
 CI212 – Organização e Arquitetura de Computadores
Primeira Prova — 2010-2

1) Mostre como implementar a instrução listada abaixo no processador com ciclo longo. Sua resposta deve conter: (i) uma indicação clara de quais modificações seriam necessárias no circuito de dados do processador; (ii) um diagrama de tempos completo da execução desta instrução; (iii) explique por que o endereço de retorno é em PC+8? [20 pontos]

BAL BRANCH-AND-LINK `bal desl # $31← PC+8 , PC← (PC+4)+(ext(desl)≪ 2)` (formato I)



2) Esta questão tem quatro itens: (i) traduza o trecho de programa ao lado de C para assembly do MIPS [10 pontos]; (ii) indique as dependências de dados e de controle [5 pontos]; (iii) compute o número de ciclos para a execução correta da versão traduzida num processador segmentado sem nenhum mecanismo para detecção em hardware de dependências de dados ou de controle [2.5 pontos]; (iv) re-organize o código para que ele execute no menor número de ciclos possível. Qual o ganho de desempenho com a versão otimizada? [12.5 pontos]. O processador emprega desvios atrasados (*branch delay slot*).

Para facilitar a correção indique os registradores como *ri, rv, etc.*

```
typedef struct A {
    int a;
    int b;
    int d;
    ind m;
} Atype;

Atype v[1024];
int i;
for (i=0; i < 1024; i+=1) {
    v[i].d = v[i].a / v[i].b;
    v[i].m = v[i].a % v[i].b;
}
```

Segunda Prova — 2010-2

1) Você deve inserir no processador segmentado o circuito de uma nova instrução que executa a multiplicação de dois registradores e soma o produto ao conteúdo de um terceiro registrador. O resultado é armazenado nos registradores *hi* e *lo*. Esta nova instrução é chamada de *multiply-add*, *madd*. Por conta da duração da operação composta, a nova unidade funcional demora dois ciclos (*exec; mem*) para produzir seu resultado, e uma nova instrução *madd* pode ser iniciada a cada ciclo.

```
#define i64 (long long);
i64 X[0124];
int A[1024],B[1024],C[1024];

for(i=0; i<1024; i++)
    X[i] =
        (i64)( (A[i]*B[i])+C[i] );
```

Indique como acrescentar a instrução *madd* ao processador segmentado. Sua resposta deve conter três itens: (i) supondo um processador SEM adiantamento, quais são as condições de bloqueio para evitar riscos estruturais e riscos de dados? (ii) supondo um processador COM adiantamento, indique claramente quais

são os circuitos de adiantamento de/para a nova unidade funcional; (iii) traduza o programa ao lado para assembly do MIPS, para o processador COM adiantamento do item (ii), e indique o número de ciclos necessário para executar a versão assembly. [25 pontos]

Para facilitar a correção indique os registradores como ra, rb, etc.

MULTIPLY-ADD madd r1,r2,r3 # hi,lo ← (r1 x r2) + r3 (formato R)

2) Você deve projetar uma cache de dados para obter o máximo desempenho durante a execução do programa da questão 1.(iii). Seu projeto não pode ser demasiadamente caro. Justifique suas escolhas para os parâmetros de projeto. Sua resposta deve conter um diagrama com as larguras/campos claramente especificados, e deve mostrar quaisquer cálculos necessários para dimensionar os parâmetros de projeto. [15 pontos]

3) Um sistema de memória virtual pode ser encarado como uma “cache”.

(i) qual parte corresponde a um bloco da cache (ou uma linha)?

(ii) qual é sua associatividade?

(iii) qual é sua estratégia de escrita?

(iv) qual circuito é usado para acelerar a tradução de endereços virtuais para físicos?

Responda com, no máximo, duas frases por item.

[5 pontos]

4) Descreva um controlador de acesso direto à memória (CADM) e explique as condições nas quais seu uso é vantajoso. Sua resposta deve conter um diagrama de blocos do CADM. [5 pontos]

Exame Final — 2010-2

1) Considere a execução do trecho de código ao lado no pipeline de 5 estágios, SEM adiantamento. (a) Modifique o código para que ele execute corretamente num processador SEM lógica de bloqueios (*stalls*). (b) Desenrole o laço uma vez e re-ordene as instruções para que ele execute com um mínimo de bolhas, no processador sem adiantamento e sem lógica de bloqueios. [40 pontos]

O sufixo .d indica instrução de ponto flutuante de tipo double; e f0..f8 são registradores de ponto flutuante de 64 bits.

```
loop: ld.d f0,0(r4)
      ld.d f2,0(r6)
      mul.d f4,f2,f0 ; 1 ciclo
      add.d f6,f4,f8 ; f8 escalar
      st.d f4,0(r8)
      st.d f6,8(r8)
      addi r4,r4,8
      addi r6,r6,8
      addi r8,r8,16
      bne r8,r9,loop ; r9 limite
```

2) Faça um diagrama detalhado de uma memória cache com 256Kbytes, associatividade quaternária (4-way set-associative), 8 palavras por bloco, escrita preguiçosa e fila de escrita com capacidade para 4 referências pendentes. O processador emite endereços de 32 bits. Mostre como um endereço é interpretado pelo controlador da cache. Indique o mecanismo de reposição de blocos num mesmo conjunto. [15 pontos]

3) Considere um circuito de memória estática com as seguintes características:

(a) a memória tem 1024 linhas de 64 bytes (palavras de 4 bytes); (b) um ciclo de leitura demora 20ns, um ciclo de escrita 25ns mais 10ns de estabilização antes da próxima operação. Calcule: (i) a latência mínima para uma referência; (ii) a vazão máxima; (iii) a vazão sustentada (ou média) para uma seqüência de referências com 75% de leituras e 25% de escritas; e (iv) explique por que “vazão ≠ 1/latência”. [15 pontos]

4) A figura ao lado mostra somente o circuito de endereços de um sistema com memória virtual com cache primária de dados, cache secundária e memória dinâmica (DRAM). Descreva a operação do primeiro nível da hierarquia de memória, usando como exemplo um load e um store, e justifique **cuidadosamente** este projeto. [30 pontos]

