

Primeira Prova

1) Você deve acrescentar ao processador o circuito que suporta uma nova instrução que executa a multiplicação de um registrador por uma constante e soma o produto ao conteúdo do segundo registrador. O resultado é armazenado nos registradores *hi* e *lo*. Esta nova instrução é chamada de *multiply-add*, *madd*, com resultado em 64 bits.

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

for(i=0; i<1024; i++)
    X[i] =
        (i64)( (A[i]*(2<<B[i])+C[i] ) );
```

MULTIPLY-ADD: `madd r1,r2,r3 # hi&lo ← r1*(2<<r2) + r3` (formato R)

Um dos fatores da multiplicação é sempre uma potência de dois, e por limitações da tecnologia de implementação, o circuito que efetua a operação deve ser implementado em dois estágios, com a multiplicação no estágio EXEC e a soma no estágio MEM.

Mostre como acrescentar a instrução *madd* ao conjunto de instruções do processador. Sua resposta consiste de quatro partes:

- (i) um diagrama de blocos claro e limpo mostrando os componentes e as ligações internas da nova unidade funcional *multiply-add*; [5 pontos]
- (ii) um desenho claro e limpo, indicando os circuitos e ligações adicionais para interligar a nova unidade funcional; [7,5 pontos]
- (iii) uma definição clara e precisa das condições de bloqueio para evitar que os resultados de *madd* sejam usados incorretamente – bloqueio não é o mesmo que adiamento; [7,5 pontos]
- (iv) traduza o programa acima para assembly do MIPS. [10 pontos]

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

2) Mostre a codificação em ponto flutuante (*float*) dos dois operandos e do resultado (*x*) da multiplicação: $5 \times \log_2(256) = x$ [5 pontos]

Segunda Prova

1) Descreva o comportamento do sistema de memória virtual durante a execução do trecho de programa ao lado. Considere somente os acessos a dados. A TLB de dados tem 32 elementos, com um mapeamento por elemento, e associatividade quaternária (*4-way set associative*). As páginas são de 4Kbytes. [10 pontos]

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

for(i=0; i<1024; i++)
    X[i] =
        (i64)( (A[i] * 2<<B[i]) + C[i] ) );
```

2) Sua tarefa é projetar uma hierarquia de memória para o processador segmentado de cinco estágios. A hierarquia consiste de caches primárias (L1) separadas, TLBs separadas, cache secundária unificada (L2), e barramento entre cache secundária e memória. Os circuitos da memória principal serão projetados por outro grupo de projetistas. As caches primárias tem capacidade de 16Kbytes, mapeamento direto, 8 palavras por bloco; a cache de dados emprega escrita preguiçosa (*write-back*). As TLBs tem 32 mapeamentos cada, com associatividade quaternária (*4-way set associative*). A cache secundária tem 1Mbytes, associatividade binária (*2-way set associative*) e blocos de 16 palavras. O barramento entre L2 e memória deve ser projetado para operação eficiente.

Sua resposta deve conter: (i) diagramas limpos e claros das 3 caches e das 2 TLBs, com indicações precisas de todas as larguras de todos os campos das matrizes de memória/etiquetas,

bem como de todas as ligações entre as caches; (ii) todas as contas necessárias para projetar as 3 caches e 2 TLBs; (iii) uma justificativa para suas escolhas quanto ao projeto integrado de caches e TLBs. [25 pontos]

Exame Final

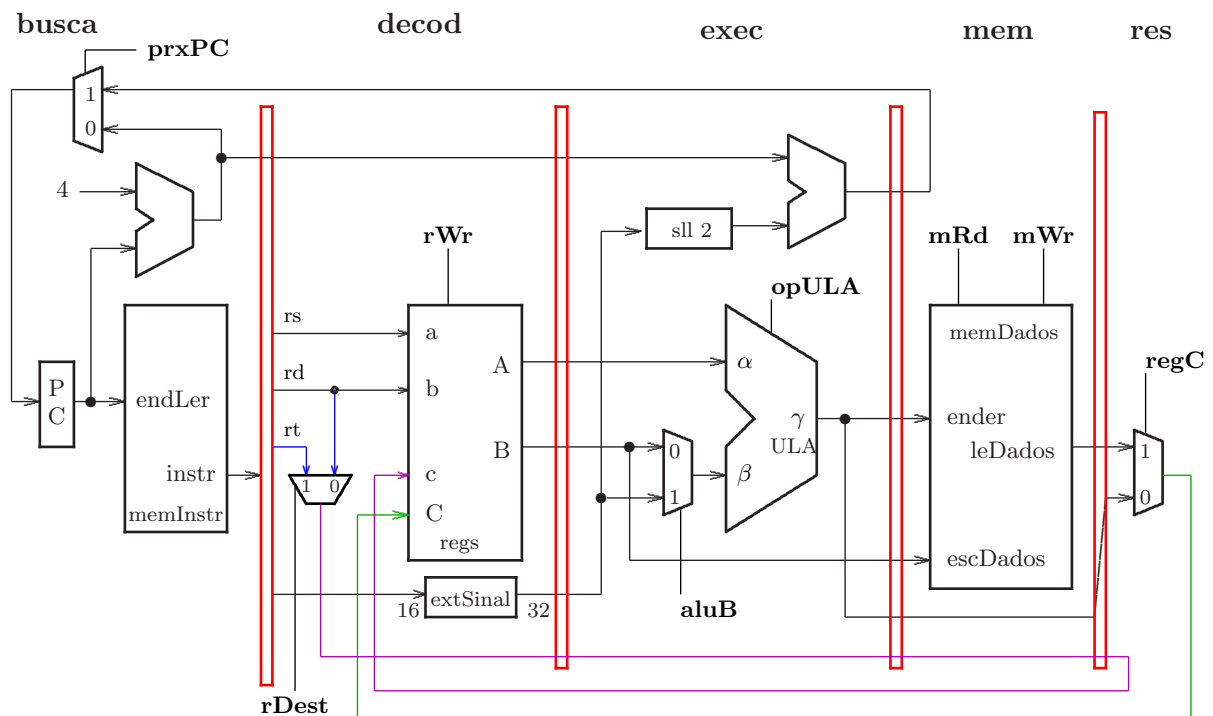
1) Traduza o código ao lado para *assembly* do MIPS, para que seja executado no menor número de ciclos, sob as seguintes restrições: (i) os blocos da cache são de oito palavras e uma falta custa 5 ciclos, a cache é com escrita forçada (*write-through*) e a fila de escrita tem capacidade de 2 palavras; (ii) a cache tem capacidade de 16Kbytes; (iii) o resultado da multiplicação/divisão só pode ser usado por `mf{hi,lo}` após um ciclo de espera. *Para facilitar a correção indique os registradores como ra, rb, etc.* [40 pontos]

```
#define N 1024
int X[N],Y[N];
int A[N],B[N],C[N];

for(i=0; i<N; i++) {
    X[i] = (int)A[i] * B[i];
    Y[i] = (int)X[i] / C[i];
}
```

2) Esta questão tem dois itens: (i) mostre como implementar a instrução BAL (*branch-and-link*) no processador segmentado mostrado na figura, através de uma descrição textual completa das operações que ocorrem em cada estágio; e (ii) indique quaisquer modificações necessárias no circuito original do processador. *Importante:* sua implementação deve provocar o menor número de bolhas a um custo razoável. [30 pontos]

```
bal desl # $31←PC+8 , PC←(PC+4)+ext(desl<<2) (formato I)
```



3) Projete uma TLB com 32 elementos, associatividade quaternária, uma tradução por bloco, endereços virtuais de 32 bits, endereços físicos de 34 bits, páginas de 4 Kbytes, bits de status apropriados, reposição aleatória. Justifique quaisquer parâmetros de projeto adicionais, necessários ao seu projeto, e mostre todas as contas. [30 pontos]