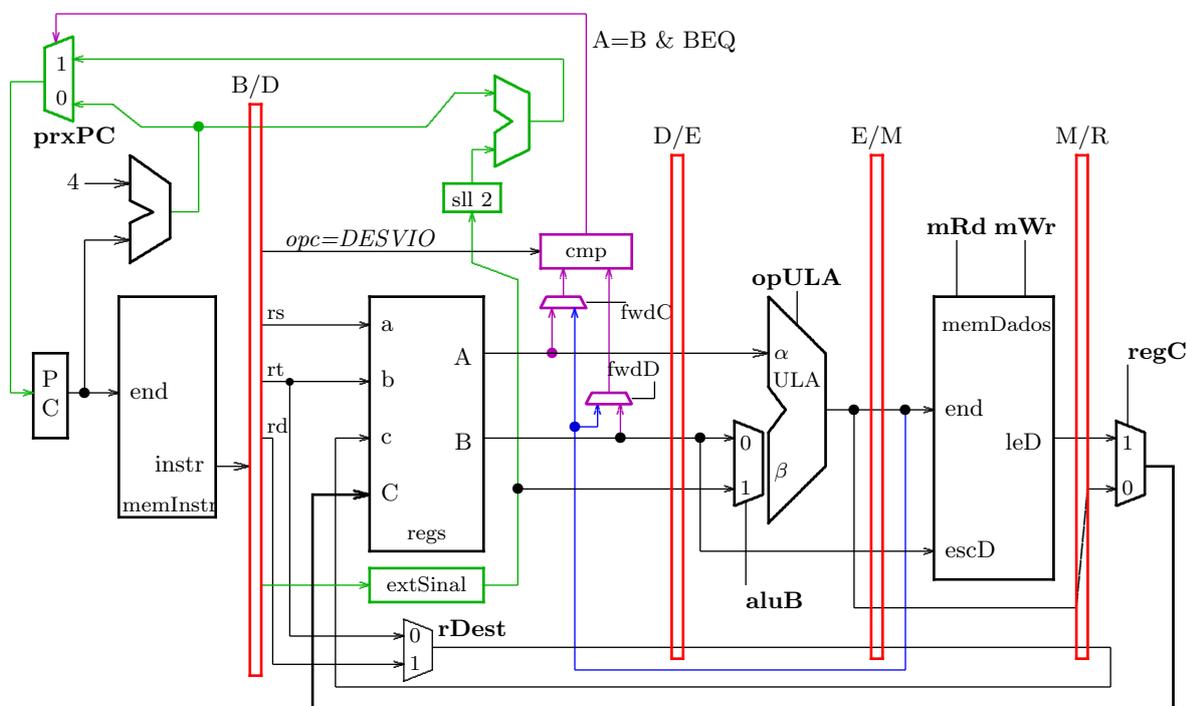


Primeira Prova

1) Considere um sistema embarcado cuja hierarquia de memória contenha somente um nível de cache, e com tempo de acesso à memória DRAM de 60ns. Qual é o limite da frequência do relógio do processador para que o desempenho seja o melhor possível com uma cache com escrita forçada? Justifique sua resposta, apresente os cálculos e explicitamente quaisquer suposições empregadas na sua resposta. [8 pontos]

2) Esta questão tem quatro itens. Mostre como implementar a instrução JALR (*jump-and-link-register*) no processador segmentado. Sua resposta deve conter: (i) uma descrição textual das operações que ocorrem em cada estágio; (ii) indicação de quais modificações seriam necessárias no circuito de dados do processador; (iii) indicação de quais são os caminhos de adiantamento para esta instrução; (iv) um diagrama de tempos completo da execução desta instrução. [10 pontos]

```
jalr rd,rs # R[rd] ← PC+8, PC ← R[rs] formato-R
```



3) Esta questão tem quatro itens. (i) Escreva uma função em C que concatena duas cadeias de caracteres, escreve a cadeia concatenada no endereço apontado pelo terceiro argumento, e retorna o número de caracteres copiados, incluindo o terminador, conforme o protótipo abaixo. (ii) Traduza sua função para o *assembly* do MIPS; (iii) escreva em *assembly* o trecho de código em que `strcat()` é invocada; e (iv) qual é o número de ciclos necessário para concatenar uma cadeia com *N* caracteres com uma de *M* caracteres? Seu código deve executar corretamente num processador com adiantamento e com *branch-delay slot*. [8 pontos]

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

```
int strcat(char *pri, char *seg, char *dst)
```

Segunda Prova

1) Considere um sistema embarcado que efetua uma série de operações sobre pares de vetores de dados com 10.000 elementos, e produz um vetor resultado com 20.000 elementos. Os elementos dos três vetores são de tipo `short`. Projete uma cache de dados para que o sistema execute com o melhor desempenho possível as operações sobre os três vetores.

Justifique sua resposta, apresente todos os cálculos e explicitamente quaisquer suposições empregadas na sua resposta. [5 pontos]

2) 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) descreva o circuito usado para acelerar a tradução de endereços virtuais para físicos.

Responda a cada um dos três primeiros itens com, no máximo, duas sentenças. [5 pontos]

3) Esta questão tem quatro itens.

(i) Escreva uma função em C que copia um vetor em memória para o registrador de saída de um periférico. O registrador do periférico tem 16 bits de largura e o `short` mais significativo deve ser gravado antes do menos significativo. O protótipo da função é dado abaixo.

(ii) Traduza sua função para assembly do MIPS.

(iii) Estime o número mínimo de ciclos para transferir 100 inteiros, considerando que os dados são transferidos da memória DRAM através da cache (blocos de 8 palavras). Um acesso à DRAM custa 100 ciclos e um acesso ao periférico custa 200 ciclos. Considere somente a execução de `cpyVet()`. Apresente todos os cálculos e explicitamente quaisquer suposições empregadas na sua resposta.

(iv) Explique por que sua estimativa é otimista. [15 pontos]

Para facilitar a correção indique os registradores como `rs`, `rd`, etc.

```
void cpyVet(int *src, short *dst, int num)
```

Prova Substitutiva

1) Mostre como implementar a instrução `BRANCH-AND-LINK` definida abaixo. A vírgula indica “execução concorrente”. Sua resposta deve conter: [9 pontos]

(a) uma indicação textual clara das modificações no circuito do processador;

(b) a tabela com os sinais de controle ativos em cada segmento;

(c) uma (ou mais) sequência(s) de instruções com `bal` que necessite(m) de adiantamento; indique claramente qual é o adiantamento necessário.

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

2) Considere um sistema de memória virtual com as seguintes características: (i) endereço virtual de 32 bits (endereço de byte); (ii) páginas com 4 Kbytes; e (iii) endereço físico com 36 bits. A tabela deve conter bits de válido, *read-only*, executável, modificado e usado. Justifique sua resposta, apresente todos os cálculos e explicitamente quaisquer suposições empregadas na sua resposta. [8 pontos]

(a) Mostre como implementar a Tabela de Páginas Hierárquica em DOIS níveis;

(b) suponha que 1/8 dos elementos da tabela do primeiro nível não sejam nulos. Quais os tamanhos máximo e mínimo de espaços de endereçamento que poderiam ser mapeados nesta tabela?

- 3) Esta questão tem três itens. [8 pontos]
- (a) Escreva uma função em C que copia a cadeia de caracteres apontada pelo primeiro argumento para a posição apontada para o segundo argumento, e que retorna o número de caracteres copiados, conforme o protótipo abaixo;
- (b) traduza sua função para o *assembly* do MIPS. Seu código deve atender às convenções do *assembly* do MIPS e executar corretamente num processador sem adiantamento mas com branch-delay slot e load-delay slot.

```
int strcpy(char *fte, char *dst);
```

Exame Final

- 1.a) Enumere e descreva quatro causas de exceções no processador segmentado. Desenhe um diagrama simplificado do processador e mostre o(s) estágio(s) em que as exceções ocorrem, e indique em que estágio(s) são tratadas. [15 pontos]
- 1.b) Liste a sequência de eventos associados ao tratamento de uma das exceções do item anterior, considerando todos os efeitos no processador. [10 pontos]

2) Considere o programa de multiplicação de matrizes abaixo. As matrizes contêm 1024x1024 elementos, cada elemento um double (8 bytes). O programa é executado num único processador, num sistema de memória virtual com páginas de 4 Kbytes. Existe uma cache primária com 64 Kbytes e uma cache secundária com 1 Mbytes. Os blocos de cache primária tem 32 bytes de largura, e os blocos da cache secundária 64 bytes. [35 pontos]

- (i) Descreva o comportamento da hierarquia de caches durante a execução deste programa — ignore memória virtual neste item.
- (ii) Descreva o comportamento do sistema de memória virtual durante a execução deste programa — ignore memória cache neste item.

```
for (i = 0; i < 1024; i++) {
    for (j = 0; j < 1024; j++) {
        sum = 0.0;
        for (k = 0; k < 1024; k++)
            sum += a[i][k] * b[k][j];
        c[i][j] = sum;
    }
}
```

- 3) Traduza para *assembly* do MIPS o trecho de código C. O operador % é o módulo da divisão inteira. Seu código deve atender às convenções do *assembly* do MIPS e executar corretamente num processador com adiantamento, com branch-delay slot e com load-delay slot. [40 pontos]

```
int v; int X[1024], Y[2048];
int reduz(int *, int, int *, int);
...
v = reduz(X, 1024, Y, 2048);
...
int reduz(int *v, int vSz, int *w, int wSz) {
    int i=1; int a=0;
    while (i < vSz) {
        a = a + v[i] + w[ (v[i] % wSz) ];
        i = i*2;
    }
    return a;
}
```