

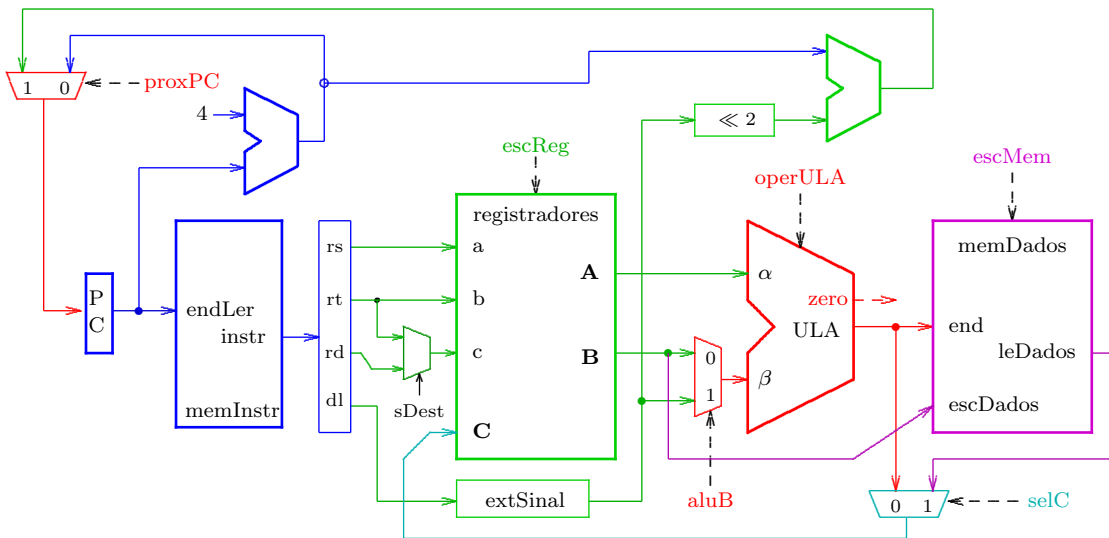
Primeira Prova — 2007-2

1) Esta questão tem três itens. (i) 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, incluindo o terminador, conforme o protótipo abaixo. (ii) Traduza sua função para o assembly do MIPS; e (iii) escreva em assembly o trecho de código em que `strcpy()` é invocada.

```
int strcpy(char *fte, char *dst) [12 pontos]
```

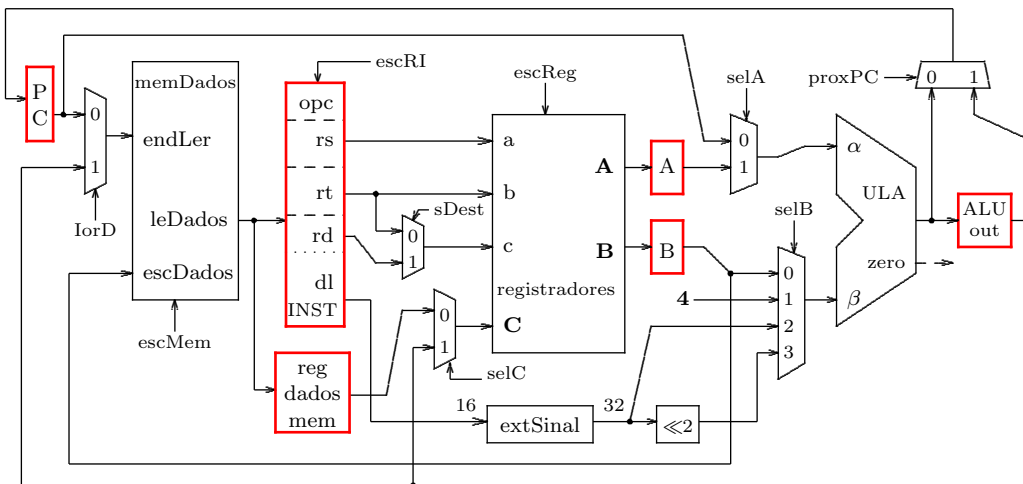
2) Esta questão tem dois itens. O conjunto de instruções do MIPS-64 contém uma nova versão da instrução LOAD, chamada *load indexed*, descrita abaixo. (i) Mostre como implementá-la no processador de ciclo longo, e (ii) desenhe um diagrama de tempos completo da execução desta instrução.

```
ldx r1,r2,r3 # r1 ←mem[r2 + r3] [10 pontos]
```



3) Esta questão tem três itens. Mostre como implementar a instrução JALR (*jump-and-link-register*) no processador com ciclo longo. Sua resposta deve conter (i) uma descrição textual completa das ações da instrução (indicação do que ocorre em cada estado), (ii) indicação de quais modificações seriam necessárias no circuito de dados do processador, (iii) um diagrama de tempos completo da execução desta instrução.

```
jalr r1,r2 # r1←PC+8 , PC←r2 [10 pontos]
```



Segunda Prova — 2007-2

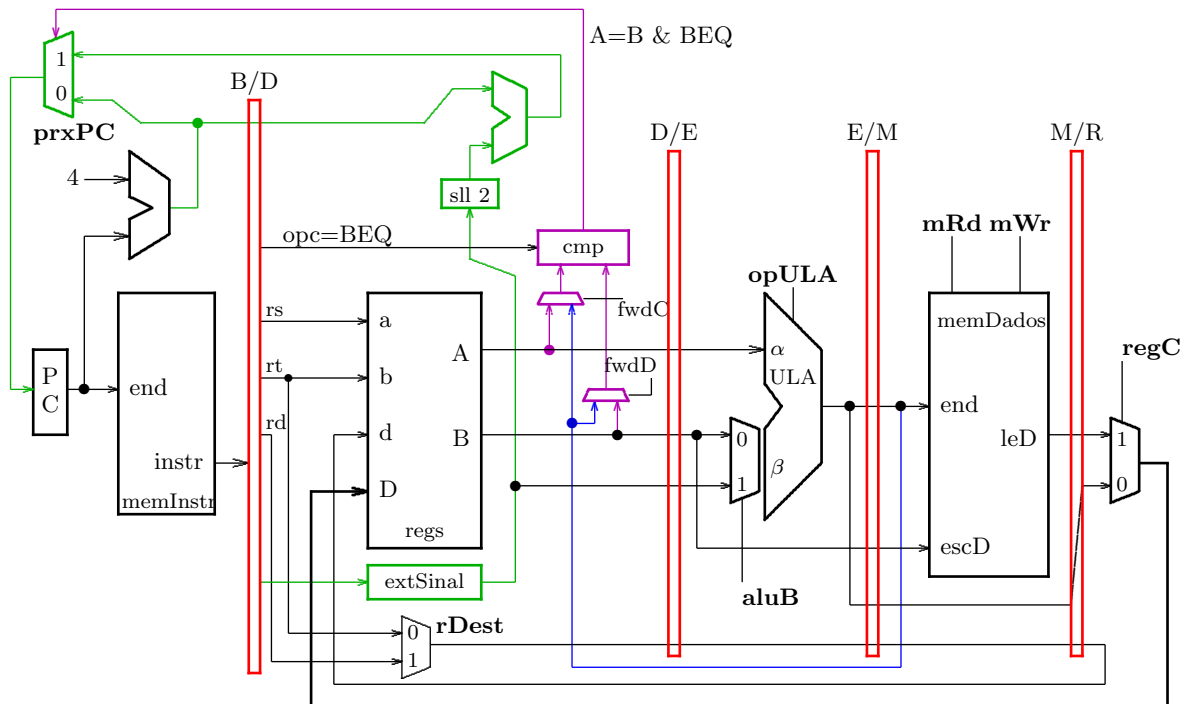
1) Para cada um dos itens abaixo, explique CLARA e SUCINTAMENTE o que são e porque produzem ganho de desempenho. (a) Segmentação em cinco estágios; (b) eliminação de riscos estruturais; (c) eliminação de riscos causados por dependências de dados; (d) eliminação de riscos causados por dependências de controle; e (e) super-escalaridade. [10 pontos]

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.

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

(formato I)

[10 pontos]



3) Esta questão tem três itens: (i) projete uma cache de instruções com 32 Kbytes, com 8 instruções por bloco e mapeamento direto; (ii) mostre como integrar um predictor de desvios ao seu projeto de cache; (iii) descreva precisamente o que acontece com sua cache+predictor quando um desvio é previsto corretamente, e quando a previsão é errada, e as informações no predictor devem ser atualizadas —para este item considere um processador como o do diagrama acima. [20 pontos]

Terceira Prova — 2007-2

1) Você foi encarregado de projetar um sistema de memória de alto desempenho. Os circuitos de memória tem tempo de acesso de 33ns, e os *drivers* ou *buffers* das linhas de dados, de endereço e de controle necessitam de 5ns para garantir que os sinais sejam propagados através de todo o comprimento do barramento. Seu projeto deve garantir a máxima vazão possível, sob as seguintes condições: (i) largura máxima de 32 bits, (ii) não há limite no número de transações simultâneas, exceto pelo custo das etiquetas, (iii) arbitragem custa um ciclo, (iv) é necessário um ciclo de intervalo entre uma transação de escrita e uma de leitura para inverter a direção dos *drivers* nas linhas de dados, e (v) o barramento deve ser síncrono.

Seu projeto deve conter (a) diagrama/s de tempos indicando o seqüenciamento dos sinais de controle, endereços e dados, (b) uma descrição textual do protocolo de sinalização —que pode ser uma descrição dos diagramas. [15 pontos]

2) Considerando que o código abaixo descreve as estruturas de dados necessárias, escreva, em C, uma função para o tratador de faltas na TLB de dados. Suponha que os registradores da interface da TLB estão mapeados no endereço físico 0xFFFF.0000, e que a estrutura TP é alocada no endereço 0xF000.0000.

Se as estruturas, como definidas, não são suficientes, acrescente os componentes necessários e justifique os acréscimos. Explique claramente suas suposições quanto a quaisquer estruturas em hardware ou em software de que seu código depende. [10 pontos]

```
typedef unsigned int uint;
typedef struct {
    uint:1 val;
    uint:1 usd;
    uint:1 mod;
    uint:1 wr;
    uint:1 ex;
    uint:20 pfn;
    uint:7 pad;
} elmtTP;

typedef struct {
    elemTP map[32];
    uint:5 rand;
    uint:1 pgFlt;
    uint:26 pad;
} TLB;

elemTP TP[2**20];
TLB dtlb;
TLB *p;
```

3) Esta questão tem três itens: (i) enuncie a Lei de Amdahl; (ii) explique como esta lei influencia no projeto de sistemas de Entrada e Saída; e (iii) dê exemplos. [5 pontos]

Exame Final — 2007-2

1) Considerando que o código abaixo descreve as estruturas de dados necessárias, escreva uma função (em C) para o *tratador de faltas de página*. Suponha que os registradores da interface da TLB estão mapeados no endereço físico `0xFFFF.0000`, e que a estrutura TP é alocada no endereço `0xF000.0000`. Se as estruturas, como definidas, não são suficientes, acrescente os componentes necessários e justifique os acréscimos. Explique claramente suas suposições quanto a quaisquer estruturas em hardware ou em software de que seu código depende. A função `encontraPaginaLivre()` retorna o número de uma página física livre (quadro disponível), e a função `lePaginaDoDisco()` retorna TRUE se a página solicitada foi lida com sucesso da área de *swap* e copiada do disco para a memória, no local indicado. Estas são duas funções de biblioteca. [40 pontos]

```
typedef unsigned int uint;
typedef struct {
    uint:1 val;
    uint:1 usd;
    uint:1 mod;
    uint:1 wr;
    uint:1 ex;
    uint:20 npf;
    uint:7 pad;
} elmtTP;

uint encontraPaginaLivre(void);
typedef struct {
    elemTP map[32];
    uint:5 rand;
    uint:1 pgFlt;
    uint:26 pad;
    uint epc;
    uint etq[32];
} TLB;

elemTP TP[2**20];
TLB dtlb;
```

2) Esta questão tem dois itens: (i) mostre como implementar a instrução `bgtzal` (*branch on greater-than or equal-to zero and link*) no processador da figura com uma descrição textual completa das operações que ocorrem em cada um dos 5 estágios; e (ii) indique quaisquer modificações necessárias. [20 pontos]

```
bgtzal rs,desl # if(rs ≥ 0) { r31 ← PC+8 , PC ← (PC+4)+ext(desl << 2) } (formato I)
```

3) Esta questão tem dois itens: (i) mostre como implementar uma memória cache de instruções com capacidade para 128 Kbytes, associatividade quaternária, e 8 palavras de 32 bits em cada bloco; (ii) indique como o controlador da cache interpreta um endereço emitido pela CPU. [20 pontos]

4) Considere um sistema de uso geral com um processador super-escalar e dois níveis de memória cache. Para Entrada e Saída (E/S) neste sistema são usados E/S por programa (*polling, programmed I/O*), E/S por interrupções e E/S por acesso direto à memória (ADM) com interrupções. Esta questão tem dois itens: (i) explique porque é necessário particionar o espaço de endereçamento de um processo em áreas distintas (segmentos); (ii) indique exemplo(s) que justifique(m) as escolhas do item (i). [20 pontos]