



**Faculdade de Computação**  
**Arquitetura e Organização de Computadores 1**  
**1ª Lista de Exercícios Preparatórios**  
*Prof. Cláudio C. Rodrigues*

Esta coleção de exercícios abordam os conceitos relacionados com a Evolução dos Sistemas Computadores, Organização de Sistemas Computadores, Modelo Estrutural e Funcional dos principais componentes de um computador.

**Problemas:**

**Fundamentos de Arquitetura e Organização**

- P1)** A natureza hierárquica de sistemas complexos é essencial para seu projeto e sua descrição. O projetista somente precisa lidar com um nível particular do sistema de cada vez. Em cada nível, o sistema consiste de um conjunto de componentes e seus inter-relacionamentos. O comportamento de cada nível depende somente de uma caracterização abstrata e simplificada do sistema no nível inferior mais próximo. Em cada nível, o projetista está interessado na sua estrutura e função. Defina estrutura e função.
- P2)** Quais são as funções básicas mais importantes de um Sistema Computador?
- P3)** Durante a concepção de um computador, o projetista deve focar nas características funcionais e estruturais de cada nível. Apresente e descreva os elementos funcionais e estruturais essenciais de um computador. Se possível, relacione-os.
- P4)** Qual o objetivo de abstrair o computador em uma estrutura hierárquica com multiníveis? Fundamente sua resposta tanto na parte física (HW) quanto lógica (SW).
- P5)** O termo gargalo de von-Neumann está relacionado a qual fator? Como o “gargalo” é influenciado pela Lei de Moore?
- P6)** O projeto de um Sistema Computador, geralmente, é orientado para desempenho (performance). Você poderia descrever alguma técnica ou estratégia, empregada para alcançarmos um desempenho eficiente e balanceado?
- P7)** A função básica realizada por um Computador é a execução de programas. O programa para ser executado consiste de um conjunto de instruções armazenadas em memória. O componente que realiza esta função é a Unidade Central de Processamento (CPU). Considerando que a CPU trata interrupções, descreva o diagrama de estados para o ciclo de execução desta CPU.
- P8)** O que é um ciclo de instrução? Quais são suas etapas e atividades previstas em cada estágio?
- P9)** Qual é o objetivo do uso de interrupções? O que muda no ciclo de instruções básico para o tratamento de uma interrupção? Descreva as atividades necessárias em um ciclo de interrupção.
- P10)** Como a CPU pode lidar com várias interrupções? Descreva o funcionamento básico de cada alternativa dada.

**Aritmética Computacional**

- P11)** Considere o número hexadecimal **0xCAFE**.
- Quantos bits são necessários para representá-lo?
  - Qual o valor equivalente dele em binário e em decimal?
  - Ele é positivo ou negativo, caso represente um número com sinal?
  - Determine o seu valor e o seu simétrico em decimal e em hexadecimal.
  - O que é “overflow” e qual o valor ele aconteceria neste cenário?
- P12)** Converta os números binários representado em complemento-de-dois de 32 bits para decimal:
- 1111 1111 1111 1111 1111 1111 0000 0110<sub>bin</sub>
  - 1111 1111 1111 1111 1111 1111 1110 1111<sub>bin</sub>
  - 0111 1111 1111 1111 1111 1111 1110 1111<sub>bin</sub>

- P13)** Os bits não possuem significado inerente. Dado o padrão de bits 1010 1101 0001 1111 0000 1111 0000 0111<sub>2</sub>. O que ele representa, supondo que seja?
- Um inteiro representado em complemento-de-dois.
  - Um inteiro sem sinal.
  - Um número em ponto flutuante com precisão simples.
- P14)** Como seria realizada a soma de números hexadecimais de 4 dígitos em notação complemento-de-dois, para que em nenhum momento se produza *overflow*?
- $$7744_{16} + 5499_{16} + 6788_{16} + AB68_{16} + 88BD_{16} + 9879_{16} = 0003_{16}$$
- P15)** Calcule as seguintes operações aritméticas com operandos representados em complemento-de-dois:
- |    |                   |    |                    |    |                   |    |                    |
|----|-------------------|----|--------------------|----|-------------------|----|--------------------|
| a) | 1000 0111         | b) | 0010 1000          | c) | 0000 1111         | d) | 1111 1100          |
|    | <u>+0111 1100</u> |    | <u>- 1111 1010</u> |    | <u>-0101 0000</u> |    | <u>- 1111 0001</u> |
- P16)** Verifique se é válida a seguinte definição alternativa para *overflow* em uma operação aritmética na representação em complemento-de-dois. Se o resultado da operação ou-exclusivo dos bits ‘vai-um’ (*carry-in*) e ‘vem-um’ (*carry-out*) da coluna mais à esquerda for 1, então ocorreu o *overflow*. Caso contrário não ocorreu.
- P17)** O algoritmo de divisão de números inteiros em complemento de dois, descrito na seção 8.3 do livro texto, é conhecido como um método de restauração, porque o valor no registrador A deve ser restaurado após uma subtração malsucedida. Uma abordagem ligeiramente mais complexa, conhecida como não-restauração, evita adições e subtrações desnecessárias. Proponha um algoritmo para essa última abordagem.
- P18)** Quando falamos sobre imprecisão em operações aritméticas em números de ponto flutuante, sempre associamos os erros ao cancelamento de bits que ocorre na subtração de valores muito próximos. Mas, quando X e Y são aproximadamente iguais, a diferença X-Y é obtida precisamente, sem erro. O que realmente querem dizer?