



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

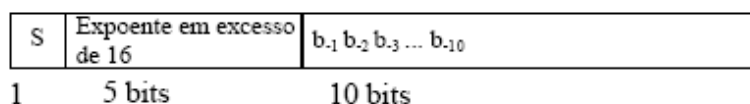
Curso de Tecnologia em Sistemas de Computação

Disciplina: Organização de Computadores

AP2 2º semestre de 2008 - GABARITO

1. (2,0) Considere um computador, cuja representação para ponto fixo e para ponto flutuante utilize 16 bits. Na representação para ponto flutuante, o número a ser representado deve ser expresso na notação científica normalizada ($\pm(1, b_1 b_2 b_3 \dots b_{10}) 2^{\text{expoente}}$). O bit mais à esquerda representa o sinal e deve ser 0 para números positivos e 1 para números negativos.

Como no padrão IEEE 754, quando todos os bits do expoente são iguais a 0 ou iguais a 1, os conjuntos de bits representam casos especiais e não representam números na notação científica normalizada. O expoente está representado em excesso de 16. A representação é mostrada na figura abaixo.



a) (1,0) Suponha que o conteúdo dos 16 bits seja $D080_{16}$. Indique o valor em decimal (pode deixar as contas indicadas) para este conjunto de bits quando considerarmos que ele está representando:

$$(D080)_{16} = (11010000 \ 10000000)_2$$

a.1) (0,2) um inteiro sem sinal

$$2^{15} + 2^{14} + 2^{12} + 2^7 = 53376$$

a.2) (0,3) um inteiro em complemento a 2

$$-2^{15} + (2^{14} + 2^{12} + 2^7) = -12160$$

a.3) (0,5) um número em ponto flutuante utilizando a representação do enunciado

$$\text{sinal} = 1 = \text{negativo}$$

$$\text{expoente} = 10100 = 20 - 16 = +4$$

$$\text{mantissa} = 0010000000$$

$$\text{representando a normalização} = -1,0010000000 \times 2^4$$

$$-10010,0000 = -18,0$$

b) (0,5) Qual o menor e o maior valor positivo de números expressos na notação científica normalizada que podem ser representados utilizando-se a representação em ponto flutuante, descrita no enunciado, para este

computador? Os valores devem ser representados em decimal (pode deixar as contas indicadas). Lembre-se que existem casos especiais.

Maior número positivo

Sinal: 0 (positivo)

Expoente (não considerar casos especiais 00000 ou 11111), portanto = 11110

Mantissa: 111111111

Maior número positivo: $0\ 11110\ 111111111 = 1,111111111 \times 2^{14} = 2^{14} + 2^{13} + 2^{12} + 2^{11} + 2^{10} + 2^9 + 2^8 + 2^7 + 2^6 + 2^5 + 2^4 = 32752$

Menor número positivo

Sinal: 0 (positivo)

Expoente (não considerar casos especiais 00000 ou 11111), portanto = 00001

Mantissa: 000000000

Menor número positivo: $0\ 00001\ 000000000 = 1,000000000 \times 2^{-15} = 2^{-15} \approx 0,00003052$

c) (0,5) Mostre a representação em ponto flutuante do valor em decimal -5,1, na representação do enunciado para ponto flutuante.

$-5,1 = 101,00011001100110011001100...$

$1,0100011001100110011001100... \times 2^2$

Sinal: 1 (negativo)

Expoente $2 + 16 = 10010$

Mantissa: 0100011001

Resposta: 1 10010 0100011001

2 (2,0) Considere uma máquina cujo controlador de scanner possui três registradores para se comunicar com o resto do sistema: o primeiro é utilizado para receber comandos do sistema (comando para indicar que se quer receber um byte do scanner, por exemplo), o segundo para indicar o estado do controlador do scanner (indicação de que o controlador de scanner está pronto para enviar um byte ao sistema, por exemplo), e o terceiro onde o byte a ser enviado ao sistema é armazenado pelo scanner. O primeiro registrador possui o endereço 40, o segundo 41 e o terceiro 42. O processo de recebimento de um byte do scanner consiste no envio do comando 01 para o registrador 40 do controlador de scanner, indicando que o sistema deseja receber um byte do scanner. O controlador de scanner recebe este comando e armazena o valor 01 no registrador de estado 41 quando o controlador de scanner tiver um byte pronto para ser enviado pronto. Este byte deverá ser armazenado no registrador 42 do controlador de scanner. Descreva detalhadamente os três possíveis métodos de comunicação entre o controlador de scanner com a unidade central de processamento e memória principal: por E/S programada, por interrupção e por acesso direto à memória. Você deve descrever estes métodos dentro do contexto do ambiente descrito neste enunciado. Não serão consideradas descrições gerais dos três métodos.

Resposta:

E/S programada:

A UCP envia o comando 01 para o registrador com endereço 40. Depois, fica lendo o conteúdo do registrador com endereço 41 e verificando se o seu conteúdo é igual a 01. Quando o conteúdo for igual a 01, a UCP lê o byte armazenado no registrador com endereço 42. Este procedimento é repetido para receber cada byte gerado durante o processo de digitalização de uma imagem pelo scanner.

E/S por interrupção:

A UCP envia o comando 01 para o registrador com endereço 40. Depois, vai executar outras instruções. Quando o controlador do scanner detecta que existe um byte do processo de digitalização, ela envia um sinal de interrupção para a UCP. A UCP, ao receber o sinal de interrupção, finaliza a instrução que estiver executando,

salva o contexto do programa que estava sendo executado e atende a interrupção, lendo o byte que está armazenado no registrador de endereço 42 do controlador da scanner. Este procedimento é repetido para receber cada byte gerado durante o processo de digitalização de uma imagem pelo scanner.

E/S por acesso direto à memória:

Neste caso, existe um controlador de DMA (Direct Access Memory) que é responsável pela transferência dos bytes. A UCP informa a este controlador que a operação de transferência deverá ser realizada com o controlador do scanner, o endereço inicial da memória onde devem ser armazenados os bytes referentes à imagem digitalizada, o número de bytes a serem recebidos e a indicação de que o controlador de DMA deve transferir bytes do controlador do scanner para a memória. Após receber estas informações, o controlador de DMA realiza a transferência dos bytes entre controlador do scanner e memória sem a intervenção da UCP. Após a transferência de todos os bytes, o controlador de DMA avisa o fim da operação para a UCP através de um sinal de interrupção.

3. (2,0) Considere um processador que possua as seguintes características:

- Um registrador de 8 bits
- Um registrador de 16 bits
- Uma barra de dados de 8 bits
- Uma barra de endereços de 16 bits

Defina instruções que permitam ao processador carregar em um registrador o conteúdo do endereço dado, adicionar a um registrador um valor especificado e carregar no registrador A o conteúdo da posição de memória apontada pelo registrador B. Descreva cada instrução e caracterize o tipo de endereçamento que ela utiliza. Com essas instruções faça um programa que permita carregar no registrador A o elemento de ordem 1E de uma tabela que começa na posição de memória de endereço 013C e que gasta 8 bits de memória para cada elemento.

Instruções:

Seja X registrador qualquer de 8 bits e W registrador qualquer de 16 bits

<i>LD-Mb X,Op => X <- Op</i>	<i>seja Op um valor de 8 bits ⇒ registrador X recebe o conteúdo Op (modo imediato)</i>
<i>LD-MB W,Op => W <- Op</i>	<i>seja Op um valor de 16 bits ⇒ registrador W recebe o conteúdo Op (modo imediato)</i>
<i>LD-D X,(Op) => X <- (Op)</i>	<i>seja Op um valor de 16 bits correspondente a um endereço de memória ⇒ O registrador X recebe o conteúdo que está no endereço de memória Op (modo direto)</i>
<i>ADD-Mb X,Op => X <- X + Op</i>	<i>seja Op um valor de 8 bits ⇒ registrador X recebe o conteúdo Op somado ao conteúdo anterior de X (modo imediato)</i>
<i>ADD-MB W,Op => W <- W + Op</i>	<i>seja Op um valor de 16 bits ⇒ registrador W recebe o conteúdo Op somado ao conteúdo anterior de W (modo imediato)</i>

LD-IR X,W => X <- (W) seja W um registrador com conteúdo correspondente a um endereço de memória
 ⇨ O registrador X recebe o valor contido no endereço armazenado por W
 (modo indireto por registrador)

LD-B X,Op => X <- (W + Op) Seja W um registrador com conteúdo correspondente a um endereço de memória
 Seja Op um valor de 8 bits
 ⇨ O registrador X recebe o valor contido no endereço indicado por W (base) acrescido do deslocamento Op
 (modo base mais deslocamento)

Programa solução 1 (por base mais deslocamento)

Sejam:

A => Um registrador de 8 bits

B => Um registrador de 16 bits

LD-MB B, 013c => B <- 013c Reg.B recebe valor 013C
 LD-B A, 1e => A <- (B + 1e) Reg.A recebe conteúdo de (B +1e)

Programa solução 2 (por modo indireto)

Sejam:

A => Um registrador de 8 bits

B => Um registrador de 16 bits

LD-MB B, 013c => B <- 013c Reg.B recebe valor 013C
 ADD-MB B, 1e => B <- B + 1e Adiciona 1e ao reg.B
 LD-IR A, B => A <- (B) Reg.A recebe conteúdo do endereço contido em B

4) (2,0) Considere um computador que possui uma UCP constituída de um Registrador de Instruções (RI) com 32 bits, Contador de Instruções (CI) e Registrador de Endereços de Memória (REM) de 12 bits, ULA, Unidade de controle e registradores de uso geral. Este computador possui um conjunto de 64 instruções de formato único, como mostrado a seguir, e com modos de endereçamento: direto, indireto e por registrador direto e indireto.

RI = 32bits

CI = REM = 12 bits

Conjunto de 64 instruções

Cód. de Operação	Registrador	Registrador	Registrador	Operando
------------------	-------------	-------------	-------------	----------

a) Quantos registradores de emprego geral podem ser endereçados nesta UCP ?

caso 1) Modo de endereçamento direto e indireto (o operando é endereço de memória)

Tamanho da instrução = RI = 32bits

Instrução = código da operação + 3 registradores + operando

Código de operação = 64 instruções = mínimo de 6 bits para abranger todos os cód.oper.

Operando = endereço de memória = 12 bits

32 bits = 6 bits + 3 registradores + 12 bits

3 registradores = 14 bits,

para termos uma melhor proporção da quantidade de registradores

consideremos o código de operação de 8 bits pois o mínimo é 6 bits, teremos então

32 bits = 8 bits + 3 registradores + 12 bits

3 registradores = 12 bits =>

tamanho para endereçar os registradores = 4 bits =>

permite máximo de 16 registradores

caso 2) Modo de endereçamento por registrador direto e indireto (o operando é endereço de um registrador)

Tamanho da instrução = RI = 32bits

Instrução = código da operação + 3 registradores + operando

Código de operação = 64 instruções = mínimo de 6 bits, considerando 8 bits

Operando = endereço de registrador

32 bits = 8 bits + 4 registradores

4 registradores = 24 bits =>

tamanho em bits para endereçar os registradores = 6 bits =>

permite máximo de 64 registradores

b) Considere uma instrução que acessa a memória principal de modo indireto e uma outra que acessa a memória por modo registrador indireto. Indique qual das duas instruções executa seu ciclo de instrução mais rápido e explique a sua resposta.

A instrução a ser executada mais rapidamente será a de modo registrador indireto em relação ao modo indireto.

Pois em ambos modos, inicialmente deverá ser obtido o endereço onde está o conteúdo desejado. No modo registrador indireto, o endereço está em um registrador, que é muito mais rápido de ser acessado do que na memória onde está o endereço do conteúdo no modo indireto.

5. (2,0) Responda:

5.1 Por que um programa em linguagem Assembly não é diretamente executável pelo processador? Como esse problema é na prática resolvido?

Porque o processador entende apenas a linguagem de máquina que é composta de 0 e 1s, a linguagem assembly representa a linguagem de máquina sob a forma de mnemônicos para que possa ser melhor compreendida ao ser humano.

Para ser executável, um programa elaborado em assembly deverá passar pelo processo de montagem, que consiste em traduzir cada instrução (mnemônicos) para seu respectivo em linguagem binária (máquina)

5.2 Por que um compilador deve ser específico para uma determinada linguagem de programação e para uma determinada UCP?

Mesmo que as linguagens tenham em comum suas principais estruturas de programação (decisões, loops, procedimentos) diferenciam nas formas léxicas, sintáticas e semânticas, daí a necessidade de cada linguagem ter um compilador próprio. Ainda assim, mesmo que a linguagem a ser compilada seja a mesma, o compilador poderá diferenciar devido ao conjunto de instruções em linguagem de máquina do processador a qual foi destinado.

Outro fator relevante como motivador em diferenciar compiladores, mesmo direcionados a uma linguagem de programação comum e também processador, é o sistema operacional. Sistemas operacionais diferentes possuem chamadas também diferentes, chamadas estas que permitem o programa a ter acesso aos recursos de hardware, eventos relacionados aos processos como, criação, encerramento e comunicação entre processos, gerenciamento de memória, entre outros.

5.3 Explique como funcionam os processos de montagem, compilação, interpretação e ligação.

Montagem : *Processo que consiste em traduzir um programa em linguagem de montagem (assembly) para seu equivalente em binário. Processo este realizado pelo montador. Esta tradução consiste em substituir a partir dos programas os códigos de operação simbólicos por valores numéricos, nomes simbólicos de endereços por valores numéricos e converter valores de constantes para valores binários. Tipicamente, em montadores de dois passos, o programa é examinado instrução por instrução duas vezes. No primeiro passo são detectados os erros e montada a tabela de símbolos de endereços. No segundo passo é feita a criação do código objeto.*

Compilação: *Processo que consiste na análise de um programa escrito em linguagem de alto nível (programa fonte) e sua tradução em um programa em linguagem de máquina (programa objeto). Processo este realizado pelo Compilador. A Análise feita pelo compilador consiste em 3 partes:*

- *A análise léxica onde o programa fonte é decomposto em seus elementos individuais (comandos, operadores, variáveis, etc), gerando erros se for encontrada alguma incorreção.*
- *A análise sintática onde são criadas as estruturas de cada comando e verificação de acordo com as regras gramaticais da linguagem.*
- *A análise semântica onde são verificadas as regras semânticas estáticas, podendo produzir mensagens de erros.*

Interpretação: *Processo onde cada comando do código fonte é lido pelo interpretador, convertido em código executável e imediatamente executado antes do próximo comando. Difere da montagem cujo código fonte é em linguagem de montagem, e de forma similar a compilação, os comandos a serem convertidos estão em linguagem de alto nível.*

Ligação: *Processo onde é feita a interpretação à chamada a uma rotina e respectiva conexão entre o código objeto principal e o código da rotina. Este processo é executado pelo ligador. O ligador examina o código-objeto, procura referências externas não resolvidas e suas localizações nas bibliotecas substituindo a linha de chamada pelo código da rotina emitindo mensagem de erro em caso de não encontrar a rotina.*