

GABARITO - AD2 - Organização de Computadores 2012.2

1. (0,6) Considere uma máquina que utiliza n bits para representar dados como inteiros em complemento a 2. Em uma operação de soma de dois valores inteiros representados nesta máquina, indique como analisar os bits na posição $n-1$ dos valores e do resultado de modo a detectar estouro (*overflow*).

Considere os dois valores inteiros A e B , ambos com n bits e C , um número inteiro com tamanho de n bits, resultante da soma de $A + B$

$$A = (a_{n-1}, a_{n-2}, \dots, a_0)$$

$$B = (b_{n-1}, b_{n-2}, \dots, b_0)$$

$$C = (c_{n-1}, c_{n-2}, \dots, c_0)$$

a_{n-1}	b_{n-1}	c_{n-1}	Overflow?
0	0	0	Não, dois números positivos somados tem como resultado outro número positivo
0	0	1	Sim, dois números positivos somados tem como resultado um número negativo
0	1	0	Não, a soma de 2 números de sinais diferentes nunca resulta em overflow
0	1	1	Não, a soma de 2 números de sinais diferentes nunca resulta em overflow
1	0	0	Não, a soma de 2 números de sinais diferentes nunca resulta em overflow
1	0	1	Não, a soma de 2 números de sinais diferentes nunca resulta em overflow
1	1	0	Sim, dois números negativos somados tem como resultado um número positivo
1	1	1	Não, dois números negativos somados tem como resultado um número negativo

2. (1,4) Considere um computador, cuja representação para ponto fixo e para ponto flutuante utilize 16 bits. Na representação para ponto flutuante, o número 0 é representado pelo conjunto de bits $(0000)_{16}$. As outras combinações de bits representam números normalizados do tipo $\pm(1, b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8 b_9 b_{10}) \times 2^{\text{Expoente}}$, onde o bit mais à esquerda representa o sinal (0 para números positivos e 1 para números negativos), os próximos 5 bits representam o expoente em excesso de 15 e os 10 bits seguintes representam os bits b_1 a b_{10} , como mostrado na figura a seguir:

S	Expoente representado em excesso de 15	$b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8 b_9 b_{10}$
---	--	--

1

5

10

- a) (0,4) Considere o seguinte conjunto de 16 bits representado em hexadecimal F000. Indique o valor deste número em decimal, considerando-se que o conjunto representa:

$$(F000)_{16} = (1111000000000000)_{16}$$

(a.1) um inteiro sem sinal

$$2^{15} + 2^{14} + 2^{13} + 2^{12} = +61440$$

(a.2) um inteiro em sinal magnitude

$$-(2^{14} + 2^{13} + 2^{12}) = -28672$$

(a.3) um inteiro em complemento a 2

$$-2^{15} + (2^{14} + 2^{13} + 2^{12}) = -4096$$

(a.4) um número representado em ponto flutuante conforme descrição do enunciado.

11110000 00000000

Sendo:

Sinal = 1 => negativo

Expoente = 28 - 15 => *expoente* = 13

Mantissa = 0000000000

$$\begin{aligned} \text{Temos então } \Rightarrow & -1,0000000000 \times 2^{+13} \\ & -1000000000000 = -8192 \end{aligned}$$

b) **(0,2) Qual será a representação em ponto flutuante dos seguintes valores decimais neste computador:**

b.1) +12,625

Convertendo para binário = 1100,101 => normalizando $1,100101 \times 2^{+3}$

Temos então:

Sinal = 0 (positivo)

Expoente = (3 + 15 = 18) = 10010

Mantissa = 100101

Resultado: 0 10010 1001010000

b.2) -4,2

Convertendo para binário = -100,001100110011001100110011...

=> normalizando $-1,0000110011 \times 2^{+2}$

Temos então:

Sinal = 1 (negativo)

Expoente = (2 + 15 = 17) = 10001

Mantissa = 0000110011

Resultado: 1 10001 0000110011

c) **(0,4) Mostre a representação dos números do item b, caso se utilizasse a representação em complemento a 2 para representar o expoente.**

c.1) +12,625

Convertendo para binário = 1100,101 => normalizando $1,100101 \times 2^{+3}$

Temos então:

Sinal = 0 (positivo)

Expoente = 3 = 00011

Mantissa = 100101

Resultado: 0 00011 1001010000

c.2) -4,2

Convertendo para binário = -100,001100110011001100110011...
=> normalizando $-1,0000110011 \times 2^{+2}$

Temos então:

Sinal = 1 (negativo)

Expoente = 2 = 00010

Mantissa = 0000110011

Resultado: 1 00010 0000110011

- d) (0,4) Indique o menor e o maior valor positivo normalizado na representação em ponto flutuante para este computador. Mostre os valores em decimal.

Menor: $0\ 00001\ 0000000000 = +(1,0000000000)_2 \times 2^{-14} = +0,000061035$

Maior: $0\ 11111\ 1111111111 = +(1,1111111111)_2 \times 2^{+16} = +131008$

3. (0,8) Explique como funciona um monitor de vídeo LED (sugestões de fonte de consulta: sites na Internet. Na sua resposta indique as suas fontes de consulta).

Fonte: <http://eletronicos.hsw.uol.com.br/led-tv1.htm>

LED (light-emitting diode) são diodos que emitem luz quando atravessados por uma corrente elétrica, inicialmente criados pelo russo Oleg Vladimirovich Lozev, os LEDs apresentam muitas vantagens sobre as fontes de luz tradicionais, incluindo baixo consumo de energia, vida útil mais longa, robustez e rapidez no ligar e desligar. Hoje, os LEDs são encontrados em brinquedos, controles remotos, computadores, relógios digitais, forno microondas, elevadores, celulares. Agrupados, os LEDs podem formar imagens, como nas telas jumbo de TV encontradas em estádios ou em painéis publicitários. Enfileirados, eles dão um novo rumo às tevês de LCD.

Os novos monitores LED não passam de monitores LCD que usam luz de LED para iluminar a tela de cristal de líquido. A tecnologia LED captura as cores naturais da vida real com a produção de pretos mais pretos, brancos mais vivos e forte contraste entre cores escuras e vívidas. Um display 100% de LED pode ser visto em estádios, na Times Square, em Nova York, e nas ruas de Tóquio. Eles são realmente grandes. E é justamente por essa razão que um monitor 100% LED não caberia em nossas casas.

Os monitores LED que chegam ao mercado usam LED para substituir as lâmpadas fluorescentes das LCDs. Existem duas maneiras de usar LEDs para iluminar a tela de um monitor LCD: colocando-os em toda a parte traseira da tela como um painel, ou posicionando-os nas bordas da tela. As duas técnicas usam menos energia do que os monitores de plasma e de LCD de tubos fluorescentes. Na iluminação de perímetro (na borda), a maior vantagem está na possibilidade de fabricar aparelhos incrivelmente finos, já que os LEDs estão nas bordas e não atrás de todo o painel de LCD



fonte: <http://eletronicos.hsw.uol.com.br/led-tv1.htm>

A principal vantagem da iluminação direta de LED é que ela pode ser usada para aumentar os níveis de contraste ao desligar os bulbos selecionados, aumentando a quantidade de preto em partes da imagem. Os LEDs são usados em todo o painel, que pode ser dividido em pequenos segmentos controlados independentemente. Como a luz de LED pode ter cores variadas, o segredo no uso eficiente da tecnologia nos monitores está em escolher que tipo de luz de LED usar para iluminar a LCD e obter os melhores resultados em brilho e contraste. Uma das possibilidades é a iluminação com LEDs branco e a outra opção são os LEDs coloridos (RGB dinâmico), potencialmente capazes de uma gama de cores mais ampla. Os RGBs usam três LEDs coloridos - vermelho, azul e verde, proporcionando uma distribuição mais equilibrada do espectro de cores.

4. (0,8) Explique como funciona a tecnologia Wireless USB (WUSB) (sugestões de fonte de consulta: existem vários sites na Internet que explicam esta tecnologia. Na sua resposta indique as suas fontes de consulta)

Fonte: <http://www.clubedohardware.com.br/artigos/Introducao-ao-Wireless-USB-WUSB/1551>

O objetivo do Wireless USB é conectar periféricos tais como impressoras, discos rígidos externos, placas de som, tocadores de mídia e até mesmo monitores de vídeo ao micro sem a utilização de fios. Isto pode ser feito de duas formas. Se o micro e/ou o dispositivo não tiver suporte ao WUSB, você precisará instalar um adaptador para converter uma porta USB convencional em uma porta WUSB. Se o micro e/ou o dispositivo já tiver suporte ao WUSB – ou seja, se eles já vierem com uma antena WUSB – nenhum dispositivo extra será necessário. Até 127 periféricos podem ser conectados usando uma única antena no micro.

A taxa de transferência máxima teórica do WUSB é a mesma do barramento USB 2.0: 480 Mbps (60 MB/s) se o dispositivo estiver em um raio de 3 metros de distância do micro ou 110 Mbps (13,75 MB/s) se o dispositivo estiver em um raio de 10 metros de distância do micro.

A rede sem fio USB funciona na faixa de frequência UWB (Ultra Wide Band ou Banda Ultra-Larga, que vai de 3,1 GHz a 10,6 GHz), enquanto que a tecnologia Bluetooth funciona na frequência de 2,4 GHz, a mesma usada pelas redes sem fio IEEE 802.11 (Wi-Fi).

Exemplo de dispositivos WUSB: o adaptador WUSB (figura 1), o hub WUSB (figura 2) que permite conectar vários periféricos WUSB, o adaptador WUSB para conexão de monitores (figura 3),



Figura 1:
Adaptador WUSB



Figura 2: hub WUSB



Figura 3: adaptador WUSB para conectar monitor de vídeo

5. (0,8) Explique como funciona uma placa de som (sugestões de fonte de consulta: existem vários sites na Internet que explicam o funcionamento como <http://www.hsw.uol.com.br/> e www.clubedohardware.com.br. Na sua resposta indique as suas fontes de consulta)

fonte: texto retirado de <http://informatica.hsw.uol.com.br/placas-de-som.htm>

Os sons e os dados do computador são fundamentalmente diferentes. Os sons são analógicos e os dados do computador são digitais (0 e 1). A placa de som básica é uma placa de circuito impresso que utiliza quatro componentes para traduzir as informações analógicas e digitais

- um conversor **analógico/digital** (ADC)
- um conversor **digital/analógico** (DAC)
- uma **interface ISA ou PCI** para conectar a placa de som à placa-mãe
- conexões de entrada para o microfone e de saída para os alto-falantes

Em vez de conversores A/D e D/A separados, algumas placas de som utilizam um chip codificador/decodificador, também chamado de **CODEC**, que realiza as duas funções.

Um conversor analógico/digital traduz as ondas analógicas de um som em dados digitais que o computador possa entender. Para fazer isso, ele cria uma amostragem ou digitaliza o som medindo as ondas de maneira precisa em intervalos freqüentes. O número de medidas por segundo, chamado de **taxa de amostragem**, é medido em Hz. Quanto mais rápida for a taxa de amostragem de uma placa de som, mais precisa será a reconstrução da onda. Um conversor digital/analógico reproduz uma gravação digital convertendo o sinal digital para a saída do som dos alto-falantes.

O processo físico de transporte do som por cabos também pode causar **distorção**. Os fabricantes utilizam duas medidas para descrever essa redução na qualidade do som:

- **distorção harmônica total** (THD - Total Harmonic Distortion), expressada em percentagem
- **relação sinal/ruído** (SNR - Signal to Noise Ratio), medida em decibéis

Tanto para a THD quanto para a SNR, valores menores significam maior qualidade. Algumas placas possuem uma entrada digital, o que permite o armazenamento de gravações digitais sem convertê-las para o formato analógico.

Além dos componentes básicos necessários para o processamento do som, muitas placas de som possuem hardware ou conexões de entrada e saída adicionais como:

- **Processador Digital de Sinais (DSP):** Ele alivia parte do trabalho da CPU (unidade central de processamento) realizando cálculos para conversões analógicas e digitais. Os DSP podem processar simultaneamente múltiplos sons ou canais.
- **Memória:** uma placa de som pode utilizar sua própria memória para proporcionar um processamento de dados mais rápido.
- **Conexões de entrada e saída:** a maioria das placas de som possui, no mínimo, conexões para microfone e alto-falantes. Algumas delas apresentam tantas entradas e saídas que possuem uma **breakout box**, que normalmente é instalada em uma das baías de drive. Ela inclui as seguintes conexões:
 - conexões múltiplas de alto-falantes para som 3D e som surround;
 - interface digital Sony/Philips (S/PDIF), um protocolo de transferência de arquivos para dados de áudio. Esse protocolo utiliza uma conexão ótica ou coaxial para entrada ou saída da placa de som;
 - interface digital para instrumentos musicais (MIDI - Musical Instrument Digital Interface), utilizada para conectar sintetizadores ou outros instrumentos eletrônicos aos computadores;
 - conexões FireWire e USB, que conectam áudio digital ou gravadores de vídeo à placa de som.

6. (0,6) Explique como funcionam as três técnicas para realizar operações de entrada e saída: E/S programada, E/S dirigida por interrupção e acesso direto à memória. Indique pelo menos uma vantagem e uma desvantagem para cada uma delas.

E/S por programa: O processador tem controle direto sobre a operação de E/S, incluindo a detecção do estado do dispositivo, o envio de comandos de leitura ou escrita e transferência de dados. Para realizar uma transferência de dados, o processador envia um comando para o módulo de E/S e fica monitorando o módulo para identificar o momento em que a transferência pode ser realizada. Após detectar que o módulo está pronto, a transferência de dados é realizada através do envio de comandos de leitura ou escrita pelo processador. Se o processador for mais rápido que o módulo de E/S, essa espera representa um desperdício de tempo de processamento. A vantagem deste método é que o hardware necessário é simples. As desvantagens são: utilização do processador para interrogar as interfaces, o que acarreta perda de ciclos de processador que poderiam ser utilizados na execução de outras instruções, : utilização do processador para realizar a transferência de dados, o que também acarreta perda de ciclos de processador.

E/S por interrupção: Neste caso, o processador envia um comando para o módulo de E/S e continua a executar outras instruções, sendo interrompido pelo módulo quando ele estiver pronto para realizar a transferência de dados, que é executada pelo processador através da obtenção dos dados da memória principal, em uma operação de saída, e por armazenar dados na memória principal, em uma operação de entrada. A vantagem deste método é que não ocorre perda de ciclos de processador para interrogar a interface, já que neste caso, não se precisa mais interrogar a interface, ela avisa quando pronta. As desvantagens são: necessidade de um hardware adicional (controlador de interrupções, por exemplo), gerenciamento de múltiplas interrupções, perda de ciclos de relógio para salvar e recuperar o contexto dos programas que são interrompidos e utilização do processador para realizar a transferência de dados, o que também acarreta perda de ciclos de processador.

E/S por DMA: Nesse caso a transferência de dados entre o módulo de E/S e a memória principal é feita diretamente sem envolver o processador. Existe um outro módulo denominado controlador de DMA que realiza a transferência direta de dados entre a memória e o módulo de E/S. Quando o processador deseja efetuar a transferência de um bloco de dados com um módulo de E/S, ele envia um comando para o controlador de DMA indicando o tipo de operação a ser realizada (leitura ou escrita de dados), endereço do módulo de E/S envolvido, endereço de memória

para início da operação de leitura ou escrita de dados e número de palavras a serem lidas ou escritas. Depois de enviar estas informações ao controlador de DMA, o processador pode continuar executando outras instruções. O controlador de DMA executa a transferência de todo o bloco de dados e ao final envia um sinal de interrupção ao processador, indicando que a transferência foi realizada. As vantagens deste método são: permite transferência rápida entre interface e memória porque existe um controlador dedicado a realizá-la e libera a UCP para executar outras instruções não relacionadas a entrada e saída. A desvantagem é que precisamos de hardware adicional.

7. (1,0) Crie um conjunto de instruções de dois operandos, definidas em Linguagem Assembly, necessárias para a realização de operações aritméticas e elabore programas para o cálculo das seguintes equações:

I_ CONJUNTO DE INSTRUÇÕES PARA 1 OPERANDO:

```
LOAD X      =>  ACC  <-  (X)
STORE X     =>  (X)  <-  ACC
ADD X       =>  ACC  <-  ACC + (X)
SUB X       =>  ACC  <-  ACC - (X)
MUL X       =>  ACC  <-  ACC * (X)
DIV X       =>  ACC  <-  ACC / (X)
```

II_ CONJUNTO DE INSTRUÇÕES PARA 2 OPERANDOS:

```
ADD X,Y     =>  (X) <- (X) + (Y)
SUB X,Y     =>  (X) <- (X) - (Y)
MUL X,Y     =>  (X) <- (X) * (Y)
DIV X,Y     =>  (X) <- (X) / (Y)
MOV X,Y     =>  (X) <- (Y)
```

III_ CONJUNTO DE INSTRUÇÕES PARA 3 OPERANDOS:

```
ADD X , Y, Z  =>  (X) <- (Y) + (Z)
SUB X , Y, Z  =>  (X) <- (Y) - (Z)
MUL X , Y, Z  =>  (X) <- (Y) * (Z)
DIV X , Y, Z  =>  (X) <- (Y) / (Z)
```

- a) $X = A + (B * (C - A) + (D - E / B) * D)$

PARA 1 OPERANDO:

```
LOAD C      =>  ACC  <-  (C)
SUB A       =>  ACC  <-  ACC - (A)
MUL B       =>  ACC  <-  ACC * (B)
STORE T1    =>  (T1) <-  ACC
LOAD E      =>  ACC  <-  (E)
DIV B       =>  ACC  <-  ACC / (B)
STORE T2    =>  (T2) <-  ACC
LOAD D      =>  ACC  <-  (D)
SUB T2      =>  ACC  <-  ACC - (T2)
MUL D       =>  ACC  <-  ACC * (D)
ADD T1      =>  ACC  <-  ACC + (T1)
ADD A       =>  ACC  <-  ACC + (A)
STORE X     =>  (X)  <-  ACC
```

PARA 2 OPERANDOS:

```
MOV X, C    =>  (X)  <-  (C)
SUB X, A    =>  (X)  <-  (X) - (A)
MUL X, B    =>  (X)  <-  (X) * (B)
MOV T1, E   =>  (T1) <-  (E)
DIV T1, B   =>  (T1) <-  (T1) / (B)
MOV T2, D   =>  (T2) <-  (D)
SUB T2, T1  =>  (T2) <-  (T2) - (T1)
MUL T2, D   =>  (T2) <-  (T2) * (D)
ADD X, T2   =>  (X)  <-  (X) + (T2)
```

ADD X, A => (X) <- (X) + (A)

PARA 3 OPERANDOS:

SUB X, C, A => (X) <- (C) - (A)
 MUL X, B, X => (X) <- (B) * (X)
 DIV T1, E, B => (T1) <- (E) / (B)
 SUB T1, D, T1 => (T1) <- (D) - (T1)
 MUL T1, T1, D => (T1) <- (T1) * (D)
 ADD X, X, T1 => (X) <- (X) * (T1)
 ADD X, A, X => (X) <- (A) * (X)

b) $Y = (A + B * (C - D * (E / (B - F)) + B) * E)$

PARA 1 OPERANDO:

LOAD B => ACC ← (B)
 SUB F => ACC ← ACC - (F)
 STORE T1 => (T1) ← ACC
 LOAD E => ACC ← (E)
 DIV T1 => ACC ← ACC / (T1)
 MUL D => ACC ← ACC * (D)
 STORE T1 => (T1) ← ACC
 LOAD C => ACC ← (C)
 SUB T1 => ACC ← ACC - (T1)
 ADD B => ACC ← ACC + (B)
 MUL B => ACC ← ACC * (B)
 MUL E => ACC ← ACC * (E)
 ADD A => ACC ← ACC + (A)
 STORE Y => (Y) ← ACC

PARA 2 OPERANDOS:

MOV T1, B => (T1) ← (B)
 SUB T1, F => (T1) ← (T1) - (F)
 MOV T2, E => (T2) ← (E)
 DIV T2, T1 => (T2) ← (T2) / (T1)
 MUL T2, D => (T2) ← (T2) * (D)
 MOV T1, C => (T1) ← (C)
 SUB T1, T2 => (T1) ← (T1) - (T2)
 ADD T1, B => (T1) ← (T1) + (B)
 MUL T1, B => (T1) ← (T1) * (B)
 MUL T1, E => (T1) ← (T1) * (E)
 ADD T1, A => (T1) ← (T1) + (A)
 MOV Y, T1 => (Y) ← (T1)

PARA 3 OPERANDOS:

SUB Y, B, F => (Y) <- (B) - (F)
 DIV Y, E, Y => (Y) <- (E) / (Y)
 MUL Y, D, Y => (Y) <- (D) * (Y)
 SUB Y, C, Y => (Y) <- (C) - (Y)
 ADD Y, Y, B => (Y) <- (Y) + (B)
 MUL Y, B, Y => (Y) <- (B) * (Y)
 MUL Y, Y, E => (Y) <- (Y) * (E)
 ADD Y, A, Y => (Y) <- (Y) + (A)

8. (1,0) Considere um computador com instruções de um operando e endereçamento por palavras de 16 bits, possuindo o seguinte conjunto de instruções:

Cod. Op. (hexadecimal)	Sigla (assembly)	Descrição
0	END	Fim da execução
1	ADD Op.	ACC←ACC+(Op.)

2	SUB Op.	ACC←ACC-(Op.)
3	LDA Op.	ACC← (Op.)
4	STA Op.	(Op.)← ACC
5	AND Op.	ACC←ACC and (Op.)
6	XOR Op.	ACC←ACC xor (Op.)
A	JMP Op.	CI← (Op.)
B	JP Op.	Se ACC>0, então CI← (Op.)
C	JZ Op.	Se ACC=0, então CI← (Op.)
D	JN Op.	Se ACC<0, então CI← (Op.)
E	GET Op.	Ler dado para (Op.)
F	PRT Op.	Imprimir (Op.)

Considere ainda o código de operação com 4 bits e o campo de operando com 12 bits de endereço. Num dado instante, foi carregado um programa na memória. Os registradores da UCP têm os seguintes valores, em hexadecimal: CI=1AF; RI=20A3;ACC=153C; e a fila de dados de entrada tem os valores decimais: 19, 37, 13 e 52. Considere a instrução contida no RI como já executada.

End.	Conteúdo	End.	Conteúdo
1AF	E1C0	1BA	E1C3
1B0	E1C1	1BB	31C1
1B1	31C1	1BC	11C3
1B2	11C0	1BD	41C1
1B3	41C1	1BE	F1C1
1B4	D1BA	1BF	0000
1B5	E1C2	1C0	31*5
1B6	31C1	1C1	61C4
1B7	21C2	1C2	21C0
1B8	41C1	1C3	11C4
1B9	A1BE		

a) Qual o valor em hexadecimal de CI, RI e ACC ao final da execução de cada instrução?

CI	RI	Descrição	ACC
1AF	20A3		153C
1B0	E1C0	GET (1C0) / (1C0) <- E/S (dispos. Entrada) = 19	153C*
1B1	E1C1	GET (1C1) / (1C1) <- E/S (dispos. Entrada) = 37	153C*
1B2	31C1	LDA (1C1) / ACC <- (1C1)	0025
1B3	11C0	ADD (1C0) / ACC <- ACC + (1C0)	0038
1B4	41C1	STA (1C1) / (1C1) <- ACC	0038
1B5	D1BA	JN 1BA / Se ACC<0, CI <- 1BA	0038
1B6	E1C2	GET (1C2) / (1C2) <- E/S (dispos. Entrada) = 13	0038*
1B7	31C1	LDA (1C1) / ACC <- (1C1)	0038
1B8	21C2	SUB (1C2) / ACC <- ACC - (1C2)	002B
1B9	41C1	STA (1C1) / (1C1) <- ACC	002B
1BE	A1BE	JMP (1BE) / CI <- 1BE	002B
1BF	F1C1	PRT (1C1) / exibe valor de (1C1) (Saída)	002B*
**	0000	END	002B

* Considerando por exemplo o uso do DMA (gravação direto da memória, sem uso dos registradores do processador)

** Endereço de retorno para a instrução seguinte a que fez a chamada desta rotina.

b) Quais os valores impressos em decimal?

Será exibido apenas um valor: 43 em decimal (2B em hexadecimal)

c) **O que aconteceria com o programa se o conteúdo da posição de endereço 1B3 fosse alterado para 81C1?**

Na ausência do código de instrução definido pelo fabricante, o processador poderá interromper a execução informando a ocorrência de um erro, ou entraria em estado de halt, ou reinicializaria iniciando a partir do endereço 0 (reboot), pois na decodificação não seria possível saber a quantidade de operandos e nem o tamanho da instrução, em uma arquitetura CISC. Certamente, com a continuidade da execução, o processador se perderia na execução das instruções seguintes.

Imaginemos que a máquina tivesse um tamanho padrão de instruções, como na arquitetura RISC, e a possibilidade da implementação por parte do fabricante do processador poder ignorar a instrução não decodificada, passando para a seguinte. Poderíamos ter uma outra solução, mas bem provável que aquela instrução, cujo código errado possa ter sido obtido de um erro de bit vindo da memória, faça falta na execução do programa

9. (1,0) Quais são as principais características dos processadores da Intel XEON?

<http://www.computerworld.com.pt/2012/03/09/intel-recomenda-xeon-e5-para-as-pme>

O termo “Xeon” (pronuncia-se “zión”) foi incluído pela Intel aos seus processadores que são voltados para o mercado de servidores e estações de trabalho. Esses processadores reconhecem mais memória RAM, permitem trabalhar em ambiente multiprocessado (isto é, com placas-mãe com vários processadores instalados sobre ela) e possui um desempenho maior que os processadores voltados para o mercado doméstico. Devido a grande quantidade de arquiteturas Xeon desenvolvidas e já obsoletas, destacaremos nesta resposta apenas a mais recente.

A Intel colocou recentemente no mercado a nova linha de processadores de arquitetura x86, que diz suportar uma melhor flexibilidade de gestão e maior eficiência energética são as séries Xeon E3, E5 e E7. Características presentes também nos outros processadores da linha, citaremos as principais do E5.

O Xeon E5 suporta até oito núcleos por processador, mas a tecnologia de HyperThreading dobra essa capacidade para 16 (por socket). Os novos processadores também suportam mais memória – até 768GB em 24 slots – fornecendo mais DRAM para aplicações de software e máquinas virtuais.

A tecnologia Advanced Vector Extension (AVX) deverá aumentar o desempenho dos processadores até duas vezes em aplicações de computação intensiva, como a análise financeira e a computação de alto desempenho. Os Xeon E5 vêm também equipados com uma nova versão do Turbo Boost 2.0 o mecanismo de overclocking.

Intel afirma que os Xeon E5 podem melhorar a eficiência energética em mais de 50%, reduzindo o custo total de propriedade. Além disso, suportam ferramentas para monitorizar e controlar o consumo de energia elétrica, como o Intel Node Manager e Data Intel Gestor do Centro.

Em termos de entrada e saída de dados (I/O), os processadores Intel E5 beneficiam de tecnologia Data Direct. Esta permite que os routers e adaptadores Ethernet para encaminhem o tráfego directamente para a cache do processador, reduzindo o consumo de energia e a latência, segundo o fabricante. Os E5 são também os primeiros processadores a incluir o controlador de I/O usando o PCI Express 3.0 directamente no seu corpo. Isso permite triplicar o movimento de dados e reduzir a latência em até 30%, diz a Intel.

A linha Xeon E5 usa ainda a tecnologia Advanced Encryption Standard de Instrução Novo (AES-NI) para rapidamente encriptar e descriptar dados usados numa série de aplicações e transações. Desta forma, o fabricante diz ser possível usufruir dessa segurança sem afetar o desempenho do processador. Juntamente com a Tecnologia Intel Trusted Execution, o processador acabará por ajudar as organizações a protegerem seus centros de dados contra ataques, defende o

fabricante.

10. (1,0) Descreva:

a) Os modos de endereçamento, explicitando suas aplicações, vantagens e desvantagens.

Imediato: Seu campo operando contém o dado, não requer acessos a memória principal sendo mais rápido que o modo direto. Possui como vantagem a rapidez na execução da instrução e como desvantagem a limitação do tamanho do dado, e é inadequado para o uso com dados de valor variável.

Direto: Seu campo operando contém o endereço do dado, requer apenas um acesso a memória principal, sendo mais rápido que o modo indireto. Possui como vantagem a flexibilidade no acesso a variáveis de valor diferente em cada execução do programa e como desvantagem a perda de tempo, se o dado for uma constante.

Indireto: O campo operando corresponde ao endereço que contém a posição onde está o conteúdo desejado, necessita de 2 acessos a memória principal, portanto mais lento que os 2 modos anteriores. Tem como vantagem o manuseio de vetores e utilização como ponteiro, e desvantagem como muitos acessos a memória principal.

Base mais deslocamento: o endereço é obtido da soma do campo de deslocamento com o conteúdo do registrador base. Este modo de endereçamento tem como principal objetivo permitir a modificação de endereço de programas ou módulos destes, bastando para isso alterar o registrador base.

Indexado: o registrador é fixo e o deslocamento é variável. Este modo é freqüentemente utilizado no manuseio de estruturas de dados que são armazenadas em endereços contíguos de memória tais como vetores.

b) Os modos compilação e interpretação, indicando em que circunstâncias um modo é mais vantajoso do que o outro.

A compilação 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).

Na interpretação cada comando do código fonte é lido pelo interpretador, convertido em código executável e imediatamente executado antes do próximo comando.

A interpretação tem como vantagem sobre a compilação a capacidade de identificação e indicação de um erro no programa-fonte (incluindo erro da lógica do algoritmo) durante o processo de conversão do fonte para o executável.

A interpretação tem como desvantagem o consumo de memória devido ao fato de o interpretador permanecer na memória durante todo o processo de execução do programa. Na compilação o compilador somente é mantido na memória no processo de compilação e não utilizado durante a execução. Outra desvantagem da interpretação está na necessidade de tradução de partes que sejam executadas diversas vezes, como os loops que são traduzidos em cada passagem. No processo de compilação isto só ocorre uma única vez. Da mesma forma pode ocorrer para o programa inteiro, em caso de diversas execuções, ou seja, a cada execução uma nova interpretação.

11. (1,0) Explique, comparando:

a) Computadores vetoriais e Computadores matriciais

O termo computadores vetoriais que correspondem a sistemas compostos por processadores vetoriais que freqüentemente são associados à organizações de ULAs com pipeline de operações.

E o termo computadores matriciais correspondem a sistemas compostos por processadores matriciais cuja organização é formada de ULAs paralelas.

b) Sistemas SMP e Sistemas NUMA

Sistemas SMP (ou UMA) têm como característica o acesso a todas as partes da memória principal com tempo de acesso uniforme. Em sistemas NUMA, todos os processadores possuem também acesso a todas as partes da memória principal podendo diferir o tempo de acesso em relação às posições da memória e processador.

Nos sistemas SMP o aumento no número de processadores tem como consequência problemas de tráfego no barramento comum degradando o desempenho. Uma solução para isto é a utilização de clusters, que tem, usualmente, como consequência alterações significativas na aplicação (software). Nos sistemas NUMA podemos ter vários nós multiprocessadores, cada qual com seu próprio barramento, resultando em pequenas alterações na aplicação (software).

c) Arquiteturas RISC e Arquiteturas CISC

RISC: Reduced Instruction Set Computer – Computador com um conjunto reduzido de instruções

CISC - Complex Instruction Set Computer: Computador com um conjunto complexo de instruções

CISC: Principais características:

Possui microprogramação para aumento da quantidade de instruções incluindo novos modos de endereçamento, de forma a diminuir a complexidade dos compiladores e em consequência permitir linguagens de alto nível com comandos poderosos para facilitar a vida dos programadores. Em contrapartida, muitas instruções significam muitos bits em cada código de operação, instrução com maior comprimento e maior tempo de interpretação.

RISC: Principais características:

Menor quantidade de instruções e tamanho fixo. Não há microprogramação. Permite uma execução otimizada, mesmo considerando que uma menor quantidade de instruções vá conduzir a programas mais longos. Uma maior quantidade de registradores e suas utilizações para passagem de parâmetros e recuperação dos dados, permitindo uma execução mais otimizada de chamada de funções. Menor quantidade de modos de endereçamento com o objetivo de reduzir de ciclos de relógio para execução das instruções. Instruções de formatos simples e únicos tiram maior proveito de execução com pipeline cujos estágios consomem o mesmo tempo.