

## **AD2 - Organização de Computadores 2010.1**

**Data de entrega: 22/05/2010**

### **Atenção:**

- 1. ADS enviadas pelo correio devem ser postadas cinco dias antes da data final de entrega estabelecida no calendário de entrega de ADs.**
- 2. Como a avaliação a distância é individual, caso seja constatado que provas de alunos distintos sejam cópias umas das outras, independentemente de qualquer motivo, a todas será atribuída a nota ZERO. As soluções para as questões podem ser buscadas por grupos de alunos, mas a redação final de cada prova tem que ser individual.**

- 1) (1,5) Faça uma pesquisa sobre os processadores da Intel da linha Nehalem, descrevendo suas características básicas, incluindo hierarquia de memória, núcleos de processamento (cores), conjunto de operações, registradores, etc...

#### **Resposta:**

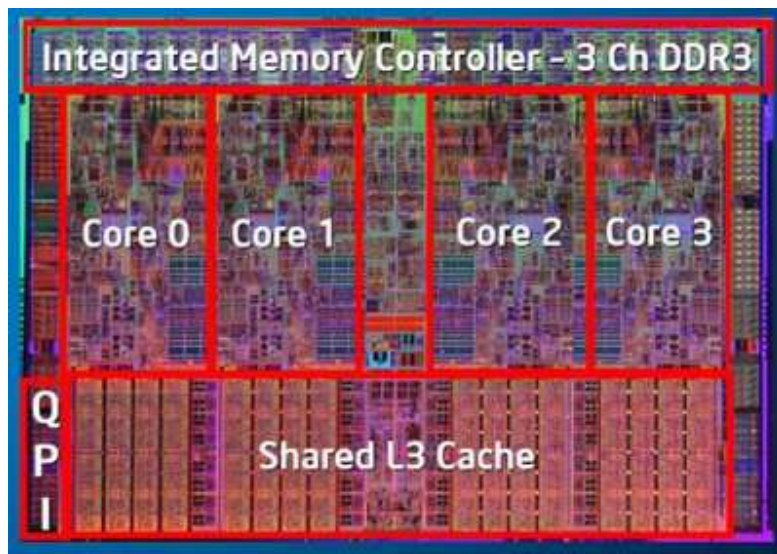
*Características básicas e melhorias com relação aos seus predecessores.*

- *Nova organização das caches (L1 com 32Kb e L2 com 256KB, ambas individuais por core e L3 com 8MB compartilhada por todos cores)*
- *Controlador de memória integrado, permitindo o uso de 3 canais independentes. Linhas independentes para comunicação com a memória, com o chipset (E/S) e para coordenação com outros processadores (SMP), barramento Quickpath*
- *Retorno do Hyper Threading*
- *Projetos de 2 a 8 núcleos*
- *Tecnologia de 45nm e de 32nm.*
- *Aprimoramento no detector de laços*
- *Aprimoramentos no modo de gerenciamento de energia*
- *Aprimoramento no pipeline, com o aumento da capacidade de fusão de instruções, e adição de novos buffers: TLB para tradução de endereços e BTB como buffer de desvios)*

*Estas características estão descritas nos textos a seguir. Textos estes retirados dos sites referenciados:*

*Texto retirado do site [www.guiadohardware.com](http://www.guiadohardware.com) (acessado em 02/05/2010)*

*“O Nehalem (pronuncia-se "nerreilem") representa uma atual arquitetura Intel, produzida usando a técnica de 45 nanômetros e mais atualmente com 32nm, mas com diversas mudanças arquiteturais em relação ao Penryn (arquitetura predecessora). As mudanças são tantas que podemos dizer que o Nehalem está para o Penryn assim como o Core 2 Duo está para o Pentium D; ou seja, trata-se realmente de uma nova arquitetura e não apenas de um Penryn com algumas melhorias.*



*Uma das principais mudanças reside no sub-sistema de memória cache. O Nehalem mantém os mesmos 32 KB de cache L1 por núcleo usado no Penryn e em outros processadores anteriores, mas as similaridades param por aí. Em vez de um grande cache L2 compartilhado, a Intel optou por utilizar uma arquitetura similar à utilizada pela AMD no Phenom, com um pequeno cache L2 (de 256 KB) para cada núcleo e 8 MB de cache L3, compartilhado entre todos os núcleos.*

*A grande diferença reside na forma como os dados são armazenados nos caches. Nos processadores AMD é usado um cache "exclusivo", onde o cache L2 armazena dados diferentes do cache L1 e o L3 armazena dados diferentes dos do L2, maximizando o espaço de armazenamento. A Intel, por outro lado, utiliza um sistema "inclusivo" onde os cache L1 e L2 armazenam cópias de dados também armazenados no cache L3.*

*Embora reduza o volume total de dados que pode ser armazenado nos caches, o sistema da Intel permite um acesso mais rápido aos dados armazenados. A latência do cache L1 subiu de 3 para 4 ciclos em relação ao Penryn, mas em compensação a latência do cache L2 caiu consideravelmente, de 15 para 11 ciclos, de forma que o saldo final é positivo. O cache L3 trabalha com uma latência de 39 ciclos, o que pode parecer bastante se comparado com a latência dos caches L1 e L2, mas é um pouco mais rápido do que o cache L3 usado no Phenom, que trabalha com uma latência de 43 ciclos.*

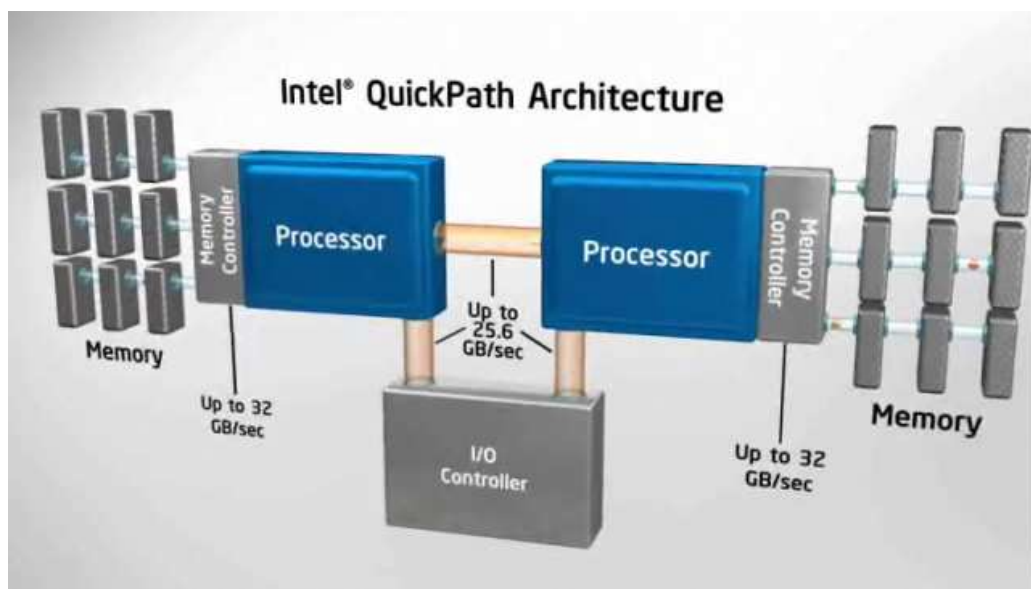
*Outra mudança dramática é a inclusão de um controlador de memória integrado, assim como temos nos processadores AMD. O controlador de memória integrado reduz substancialmente o tempo de latência da memória, resultando em um ganho de desempenho considerável.*

*Em vez de utilizar um controlador single-channel, ou dual-channel, a Intel optou por utilizar um controlador triple-channel, com suporte a memórias DDR3, operando a até 1.33 GT/s. Isso significa uma banda total de até 32 GB/s (ao utilizar 3 módulos). Para ter uma idéia, isso é 40 vezes mais do que tínhamos há 10 anos, quando utilizávamos módulos de memória SDR PC-100 em conjunto com o Pentium III.*

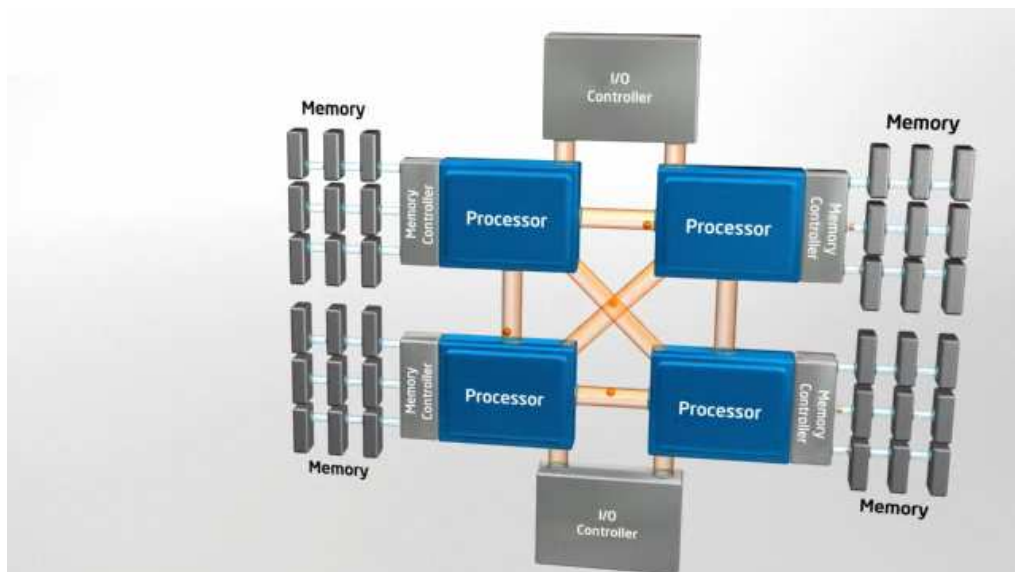
*Os três canais operam de forma independente, de forma que o processador por iniciar uma nova leitura em um dos módulos enquanto ainda espera os dados referentes a uma leitura anterior, realizada em outro módulo. Isso contribui para reduzir o tempo de latência do acesso à memória, que é, proporcionalmente, muito mais alto nos módulos DDR3.*

*O problema com o controlador integrado é que ele aumenta substancialmente o número de contatos do processador, o que quebra completamente a compatibilidade com as placas soquete 775 atuais. A versão triple-channel do Nehalem utilizará um soquete LGA com nada menos do que 1366 contatos. O formato do processador também mudou, passando a ser retangular, assim como no antigo Pentium Pro:*

*Até o Penryn a Intel remediou o problema na base da força-bruta, simplesmente adicionando mais cache L2 aos processadores. Com o QuickPath, resolveram atacar a raiz do problema, substituindo o FSB por um barramento modernizado, composto por links independentes que operam a 6.4 GT/s (a sigla "GT/s" indica o volume de transações por segundo, diferente de "GHz", que indica o clock), com a transmissão de 16 bits de dados em cada direção por ciclo, resultando em um barramento de 12.8 GB/s em cada direção (25.6 GB/s no total) por linha de dados. Como a memória é agora acessada diretamente pelo controlador de memória, este link fica inteiramente disponível para o tráfego de I/O. Ao utilizar dois processadores em SMP, cada processador passa a se comunicar com o chipset através de uma linha independente e uma terceira linha de dados é implantada para coordenar a comunicação entre os dois:*



*Ao usar 4 processadores (possibilidade que deverá ser bem explorada no caso dos servidores de alto desempenho) são incluídos barramentos adicionais, que fazem com que cada processador tenha acesso direto a todos os demais:*



*O Nehalem marca também a volta do Hyper Threading, o que faz com que o processador se apresente ao sistema operacional como tendo 8 núcleos em vez de 4. Naturalmente, o Hyper Threading não dobra o desempenho do processador, servindo apenas como um recurso extra que permite que ele aproveite melhor os recursos de processamento, processando dois threads simultaneamente sempre que possível.*

*Graças à arquitetura modular usada no Nehalem, a Intel tem uma boa flexibilidade para desenvolver versões do processador com mais ou menos núcleos de acordo com a demanda. Além da versão com 4 núcleos, estão planejadas também versões com 2 e 8 núcleos*

*.Texto complementar retirado do site [www.clubedohardware.com.br](http://www.clubedohardware.com.br) (acessado em 02/05/2010)*

*A microarquitetura Core, usada nos processadores Core 2, introduziu o conceito de fusão de instruções (“macro-fusion”), que é a capacidade de combinar duas instruções x86 dentro de apenas uma microinstrução. Isto aumenta o desempenho e diminui o consumo do processador, já que ele executará apenas uma microinstrução em vez de duas. Este esquema, no entanto, é limitado a instruções de desvio condicional e de comparação (ou seja, instruções CMP, TEST e Jcc).*

*A microarquitetura Nehalem aumenta a capacidade da fusão de instruções de duas formas. Primeiro adicionando suporte a várias instruções condicionais que não poderiam ser fundidas nos processadores Core 2. Segundo, nos processadores baseados na microarquitetura Nehalem a fusão de instruções é usada nos modos de 32 bits e de 64 bits, enquanto que nos processadores Core 2 a fusão de instruções funciona apenas quando o processador está trabalhando no modo de 32 bits.*

*A microarquitetura Core adicionou um detector de laços (“Loop Stream Detector”), basicamente um pequeno cache de 18 instruções entre as unidades de busca e de decodificação do processador. Quando o processador está executando um laço de repetição (mais conhecido como “loop”, parte de um programa que se repete várias vezes) o processador não precisa buscar as instruções requeridas novamente do cache L1 de instruções: elas já estão próximas da unidade de decodificação. Além disso, o processador desliga as unidades de busca e de previsão de desvio quando detecta a execução de um laço, fazendo com que o processador consuma menos energia.*

*A microarquitetura Nehalem adicionou também dois buffers extras: um segundo Buffer de Tradução de Endereços (TLB, Translation Look-aside Buffer) com 512 entradas e um segundo Buffer de Desvios (BTB, Branch Target Buffer). A adição desses dois buffers aumenta o desempenho do processador.*

*O TLB é uma tabela usada pelo circuito de memória virtual para conversões entre endereços físicos e endereços virtuais. A memória virtual é uma técnica onde o processador simula mais memória RAM em um arquivo no disco rígido (chamado arquivo de troca) para permitir que o micro continue operando mesmo quando não há memória RAM suficiente disponível (o processador pega o que está na memória RAM, armazena neste arquivo de troca e então libera espaço na memória para uso).*

*A previsão de desvios é um circuito que tenta “adivinhar” os próximos passos de um programa, carregando para dentro do processador as instruções que ele acha que o processador tentará carregar. Se ele estiver certo, o processador não perderá tempo carregando essas instruções da memória, já que elas já estarão dentro do processador. Aumentando o tamanho desse buffer (ou adicionando um segundo buffer, no caso dos processadores baseados na microarquitetura Nehalem) permite que este circuito carregue antecipadamente ainda mais instruções, aumentando o desempenho do processador.*

- 2) (1,5) 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.	$ACC \leftarrow ACC + (Op.)$
2	SUB Op.	$ACC \leftarrow ACC - (Op.)$
3	LDA Op.	$ACC \leftarrow (Op.)$
4	STA Op.	$(Op.) \leftarrow ACC$
5	AND Op.	$ACC \leftarrow ACC \text{ and } (Op.)$
6	XOR Op.	$ACC \leftarrow ACC \text{ xor } (Op.)$
A	JMP Op.	$CI \leftarrow (Op.)$
B	JP Op.	Se $ACC > 0$ , então $CI \leftarrow (Op.)$
C	JZ Op.	Se $ACC = 0$ , então $CI \leftarrow (Op.)$
D	JN Op.	Se $ACC < 0$ , então $CI \leftarrow (Op.)$
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?

**Resposta:**

CI	RI	Descrição	ACC
1AF	20A3		153C
1B0	E1C0	GET (1C0) / (1C0) <- E/S (dispos. Entrada)	153C*
1B1	E1C1	GET (1C1) / (1C1) <- E/S (dispos. Entrada)	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)	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) / exhibe valor de (1C1) (Saída)	002B
**	0000	HALT	002B

\* Transferência por 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?

**Resposta:**

Será exibido apenas um valor: 2B

- c) *Questão anulada, apenas como exercício complementar.* O que aconteceria com o programa se o conteúdo da posição de endereço 1B3 fosse alterado para B1BA?

**Resposta:**

CI	RI	Descrição	ACC
1AF	20A3		153C
1B0	E1C0	GET (1C0) / (1C0) <- E/S (dispos. Entrada)	153C*
1B1	E1C1	GET (1C1) / (1C1) <- E/S (dispos. Entrada)	153C*
1B2	31C1	LDA (1C1) / ACC <- (1C1)	0025
1B3	11C0	ADD (1C0) / ACC <- ACC + (1C0)	0038
1BA	B1BA	JP 1BA / Se ACC>0, CI <- 1BA	0038
1BB	E1C3	GET (1C3) / (1C3) <- E/S (dispos. Entrada)	0038*
1BC	31C1	LDA (1C1) / ACC <- (1C1)	0025
1BD	11C3	ADD (1C3) / ACC <- ACC + (1C3)	0032
1BE	41C1	STA (1C1) / (1C1) <- ACC	0032
1BF	F1C1	PRT (1C1) / exhibe valor de (1C1) (Saída)	0032
*	0000	HALT	0032

\* Transferência por 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.

Em geral, este endereço fica na pilha no topo da memória ou no espaço de endereçamento do processo.

- 3) (2,0) Explique, comparando:

- a) Computadores vetoriais e Computadores matriciais

**Resposta:**

O termo computadores vetoriais que correspondem a sistemas compostos por processadores vetoriais que frequentemente 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

**Resposta:**

*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 podem-se ter vários nós multiprocessadores, cada qual com seu próprio barramento, resultando em pequenas alterações na aplicação (software).*

c) Arquiteturas RISC e Arquiteturas CISC

**Resposta:**

*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 da redução 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.*

d) Compilação e Interpretação

**Resposta:**

*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.*

4) (1,8) Um sistema de computação possui somente 6 bits para representar dados. Considere o seguinte conjunto de bits:

A=111111

B=100001

C=011111

D=000011

a) (0,6) Indique o valor decimal para A, B, C e D considerando que eles representam:

- i) inteiros sem sinal
- ii) inteiros em sinal e magnitude
- iii) inteiros em complemento a 2

As seguintes operações foram realizadas neste sistema:

E=A+B

F=B-C

G=C+D

H=A-D

A=111111

B=100001

C=000001

D=011111

**Resposta:**

i). inteiros sem sinal

$$A=111111 = 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = +63$$

$$B=100001 = 2^5 + 2^0 = +33$$

$$C=000001 = 2^0 = +1$$

$$D=011111 = 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = +31$$

ii) inteiros em sinal e magnitude

$$A=111111 = - (2^4 + 2^3 + 2^2 + 2^1 + 2^0) = -31$$

$$B=100001 = - (2^0) = -1$$

$$C=000001 = + (2^0) = +1$$

$$D=011111 = + (2^4 + 2^3 + 2^2 + 2^1 + 2^0) = +31$$

iii) inteiros em complemento a 2

$$A=111111 = -2^5 + (2^4 + 2^3 + 2^2 + 2^1 + 2^0) = -1$$

$$B=100001 = -2^5 + (2^0) = -31$$

$$C=000001 = + (2^0) = +1$$

$$D=011111 = 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = +31$$

b) (1,2) Indique o valor decimal para E, F, G e H considerando que A, B, C, D, E, F, G e H representam:

- i) inteiros sem sinal
- ii) inteiros em sinal e magnitude
- iii) inteiros em complemento a 2

Explique, para cada uma das 4 operações acima, se a operação irá causar estouro (overflow) ou não.

**Resposta:**

i) inteiros sem sinal

$$E=A+B = 111111 + 100001 = 1) 100000 = ( +32 )_{10} \Rightarrow \text{HOUE OVERFLOW, PRESENÇA DE BIT DE ESTOURO}$$

$$F=B-C = 100001 - 000001 = 100000 = ( +32 )_{10} \Rightarrow \text{NÃO HOUE OVERFLOW}$$

$$G=C+D = 000001 + 011111 = 100000 = ( +32 )_{10} \Rightarrow \text{NÃO HOUE OVERFLOW}$$

$$H=A-D = 111111 - 011111 = 100000 = ( +32 )_{10} \Rightarrow \text{NÃO HOUE OVERFLOW}$$

*Ocorrerá overflow para E, porque o valor correto da soma é 64 que não é possível de ser representado com 6 bits utilizando-se representação inteiro sem sinal.*



Não ocorrerá overflow para F, G e H porque seu valores podem ser representados com 6 bits utilizando-se representação inteiro sem sinal.

ii) inteiros em sinal e magnitude

$$\begin{aligned} E=A+B &= 111111 + 100001 = 100000 = (0)_{10} \Rightarrow \text{HOUE OVERFLOW} \\ F=B-C &= 100001 - 000001 \Rightarrow \text{TRANSFORMAR O 2ºTERMO PARA NEGATIVO} \\ &= 100001 + 100001 \Rightarrow 100010 = (-2)_{10} \Rightarrow \text{NÃO HOUE OVERFLOW} \\ G=C+D &= 000001 + 011111 \Rightarrow 000000 = (0)_{10} \Rightarrow \text{HOUE OVERFLOW} \\ H=A-D &= 111111 - 011111 \Rightarrow \text{TRANSFORMAR O 2ºTERMO PARA NEGATIVO} \\ &= 111111 + 111111 \Rightarrow 111110 = (-30)_{10} \Rightarrow \text{HOUE OVERFLOW} \end{aligned}$$

Ocorrerá overflow para E cujo valor correto é -32, para G cujo valor é +32 e para H cujo valor é -62 que não é possível de ser representado com 6 bits utilizando-se a representação sinal e magnitude. Não ocorrerá overflow para F pois seu valor pode ser representados com 6 bits utilizando-se a representação sinal e magnitude.

iii) inteiros em complemento a 2

$$\begin{aligned} E=A+B &= 111111 + 100001 = 1) 100000 = (-32)_{10} \Rightarrow \text{NÃO HOUE OVERFLOW (o resultado da soma de dois números negativos é um número negativo)} \\ F=B-C &= 100001 - 000001 = \Rightarrow \text{TRANSFORMAR O 2ºTERMO PARA NEGATIVO} \\ &= 100001 + 111111 = 1) 100000 = (-32)_{10} \Rightarrow \text{NÃO HOUE OVERFLOW (o resultado da soma de dois números negativos é um número negativo)} \\ G=C+D &= 000001 + 011111 = 100000 = (-32)_{10} \Rightarrow \text{HOUE OVERFLOW (SOMA DE 2 NÚMEROS POSITIVOS RESULTANDO EM UM NEGATIVO)} \\ H=A-D &= 111111 - 011111 = \Rightarrow \text{TRANSFORMAR O 2ºTERMO PARA NEGATIVO} \\ &= 111111 + 100001 = 1) 100000 = (-32)_{10} \Rightarrow \text{NÃO HOUE OVERFLOW (o resultado da soma de dois números negativos é um número negativo)} \end{aligned}$$

Ocorrerá overflow para G cujo valor correto é +32 e que não é possível de ser representado com 6 bits utilizando-se a representação por complemento a 2. Não ocorrerá overflow para E, F e H pois seus valores podem ser representados com 6 bits utilizando-se a representação complemento a 2.

- 5) (1,2) Considere um computador, cuja representação para ponto fixo e para ponto flutuante utilize 16 bits. Na representação em ponto flutuante, números normalizados estão no formato  $\pm(1, b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8 b_9 b_{10} \times 2^{\text{Expoente}})$ , 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

**Resposta:**

- a) (0,4) Para o conjunto de bits  $(BE80)_{16}$ , indique o valor que está sendo representado em decimal neste computador, considerando-se que o conjunto representa:

$$(BE80)_{16} = (1011111010000000)_{16}$$

- i) um inteiro sem sinal

$$2^{15} + 2^{13} + 2^{12} + 2^{11} + 2^{10} + 2^9 + 2^7 = +48768$$

ii) um inteiro em sinal magnitude

$$-(2^{13} + 2^{12} + 2^{11} + 2^{10} + 2^9 + 2^7) = -16000$$

iii) um inteiro em complemento a 2

$$-2^{15} + (2^{13} + 2^{12} + 2^{11} + 2^{10} + 2^9 + 2^7) = -16768$$

iv) um número em ponto flutuante utilizando a representação do enunciado

1 01111 1010000000

Sendo:

Sinal = 1 => negativo

Expoente = 01111 = 15 - 15 => expoente = 0

Mantissa = 1010000000

$$\text{Temos então } \Rightarrow -(1, 1010000000)_2 \times 2^0 = -1,625$$

b) (0,4) Indique a representação dos seguintes valores decimais neste computador utilizando a representação em ponto flutuante do enunciado:

i) +25,5

Convertendo para binário = 11001,1 => normalizando  $1,10011 \times 2^4$

Temos então:

Sinal = 0 (positivo)

Expoente = (4 + 15 = 19) = 10011

Mantissa = 10011

Resultado: 0 10011 1001100000

ii) -0,1

Convertendo para binário = 0,000110011001100110011 => normalizando  $1,10011001100110011 \times 2^{-4}$

Temos então:

Sinal = 1 (negativo)

Expoente = (-4 + 15 = 11) = 01011

Mantissa = 1001100110

Resultado: 1 01011 1001100110

c) (0,4) Na representação em ponto flutuante do enunciado, os conjuntos de bits que representam o menor e maior valor para o expoente não são utilizados para representar números normalizados. Indique o menor e o maior valor positivo e o menor e maior valor negativo de números normalizados que podem ser representados na representação em ponto flutuante para este computador. Mostre os valores em decimal.

Maior número positivo: 0 11110 1111111111 =  $+(1,1111111111)_2 \times 2^{+15} = +65504$

Menor número positivo: 0 00001 0000000000 =  $+(1,0000000000)_2 \times 2^{-14} = +0,00006103515625$

Maior número negativo: 1 00001 0000000000 =  $-(1,0000000000)_2 \times 2^{-14} = -0,00006103515625$

Menor número negativo: 1 11110 1111111111 =  $-(1,1111111111)_2 \times 2^{+15} = -65504$

6) (1,0) Explique detalhadamente como funcionam os três mecanismos utilizados para transferir dados entre um dispositivo de E/S e a memória de um sistema de computação: por programa (polling), por interrupção e por acesso direto à memória. Mostre um exemplo de cada mecanismo que seja utilizado em alguma máquina comercial.

**Resposta:**

a) *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.*

*Exemplo: Discos rígidos conectados através da interface ATA*

b) *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.*

*Exemplo: Placa de rede Ethernet, um modem conectado através de uma porta USART*

c) *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.*

*Exemplo: Disco rígido conectado através da interface SCSI*

7) (1,0) Explique o mecanismo de funcionamento de um disco DVD (indique a referência bibliográfica que você usou (sugestão: site HowStuffWorks)).

Texto e imagens retirados do site <http://eletronicos.hsw.uol.com.br> (acessado em 01/05/2010)

## Introdução ao DVD

Os DVDs possuem o mesmo diâmetro e espessura que os CDs e são feitos usando os mesmos materiais e métodos de fabricação. Assim como no CD, os dados em um DVD são codificados na forma de pequenas cavidades e saliências na trilha do disco.

Um DVD é composto de várias camadas de plástico que totalizam uma espessura de cerca de 1,2 milímetros. Cada camada é criada por injeção de plástico policarbonato moldado. Esse processo forma um disco que possui saliências microscópicas dispostas na forma de uma única, contínua e extremamente longa trilha espiral de dados.

Assim que as peças transparentes de policarbonato são formadas, uma fina camada refletora de alumínio é micropulverizada sobre o disco, cobrindo as saliências. O alumínio é usado por trás das camadas internas, mas uma camada de ouro semi-refletora é usada para as camadas externas, permitindo que o laser focalize através das camadas externas e atinja as camadas internas. Depois que todas as camadas são feitas, cada uma é revestida com uma laca, prensadas juntas e curadas sob luz infravermelha. Nos discos de face única, a etiqueta é aplicada por serigrafia sobre o lado não lido. Os discos de face dupla contêm impressão somente na área não lida próxima ao furo central. As seções transversais dos diversos tipos de DVDs finalizados (fora de escala) se parecem com isto:

**Face única, camada única (4.7GB)**



**Face única, camada dupla (8.5GB)**



**Face dupla, camada dupla (17GB)**

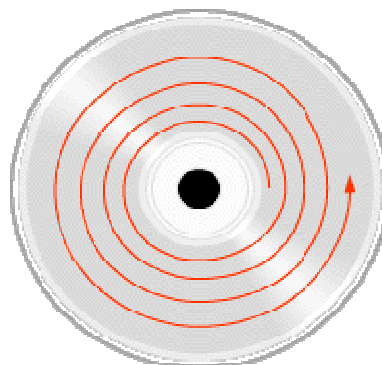


©2000 How Stuff Works

*Formatos de DVD*

*Fig 1. Seções transversais dos DVDs*

*Cada camada gravável de um DVD possui uma trilha espiral de dados. Nos DVDs de camada única, a trilha sempre circula do lado interno para o externo do disco. O fato de a trilha espiral se iniciar no centro significa que um DVD de camada única poderia ser menor do que 12 centímetros, se desejado.*



©2000 How Stuff Works

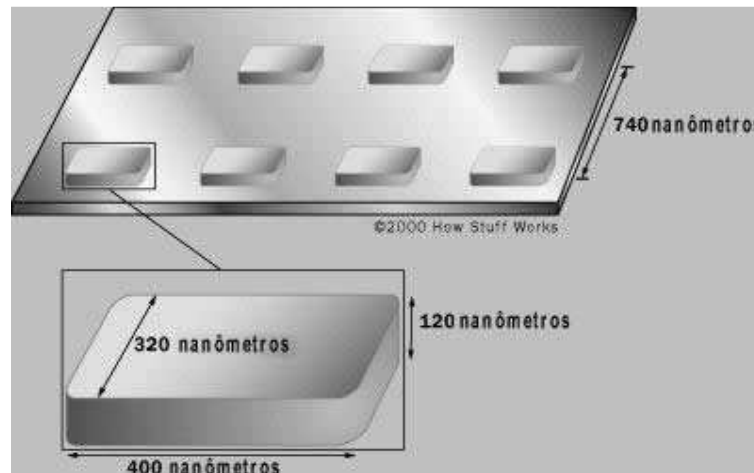
*Fig2 . Formato espiral de gravação dos  
Dados em um DVD*

*Cerca de 740 nanômetros separam uma trilha da próxima (um nanômetro é um bilionésimo de metro). E cada uma das saliências alongadas que compõem a trilha possui 320 nanômetros de largura, um comprimento mínimo de 400 nanômetros e uma altura de 120 nanômetros.*

*A denominação usual é o de "cavidades" ("pits") em vez de saliências ("bumps") em um DVD. Elas aparecem como cavidades no lado do alumínio, mas no lado de leitura do laser, elas são saliências. As dimensões microscópicas das saliências tornam a trilha espiral de um DVD extremamente longa. Se fosse possível arrancar a trilha de dados de um DVD e esticá-la em uma linha reta, ela teria quase 12*

quilômetros de comprimento! Isso significa que um DVD de lado duplo e camada dupla teria 48 quilômetros de dados!

Para ler saliências assim tão pequenas necessita-se, então, de um mecanismo de leitura de disco incrivelmente preciso.



### O aparelho de DVD e mecanismo de leitura e gravação

Um aparelho de DVD é muito parecido com um aparelho de CD. Ele possui um mecanismo que projeta um raio laser sobre a superfície do disco para ler o padrão de saliências. O aparelho de DVD decodifica o filme MPEG-2, transformando-o em um sinal de vídeo composto padrão. O aparelho também decodifica o fluxo de dados de áudio ("stream") e o envia para um decodificador, onde ele é amplificado e enviado para os alto-falantes.

O aparelho de DVD tem o trabalho de localizar e ler os dados armazenados como saliências no DVD. Considerando o tamanho extremamente pequeno dessas saliências, o aparelho de DVD é um equipamento de precisão excepcional. O dispositivo consiste de três componentes fundamentais:

- um motor de acionamento para girar o disco - o motor de acionamento é controlado com precisão para girar entre 200 e 500 rpm, dependendo de qual trilha é lida;
- um sistema de laser e lente para focalizar e ler as saliências - a luz proveniente deste laser possui um comprimento de onda menor (640 nanômetros) do que a luz do laser de um aparelho de CD (780 nanômetros), o que permite que o laser do DVD focalize as cavidades menores do DVD;
- um mecanismo de rastreamento que pode mover o conjunto do laser de modo que seu raio acompanhe a trilha espiral - o sistema de rastreamento deve ser capaz de mover o laser com resoluções da ordem de milésimo de milímetro.

O trabalho fundamental do aparelho de DVD é focalizar o laser sobre a trilha de saliências. O laser pode focalizar o material refletor semitransparente atrás da camada mais próxima ou, no caso de um disco de camada dupla, através dessa camada e até o material refletor atrás da camada interna. O raio laser passa através da camada de policarbonato, se reflete na camada refletora atrás dela e atinge o

*dispositivo optoeletrônico que detecta mudanças na luz. As saliências refletem a luz diferentemente das áreas planas do disco e o sensor optoeletrônico detecta essa mudança na refletividade. A eletrônica no drive interpreta as mudanças na refletividade para ler os bits que compõem os bytes.*

*A parte mais difícil da leitura do DVD é manter o raio laser centralizado sobre a trilha de dados. Essa centralização é o trabalho do sistema de rastreamento. Conforme o DVD é reproduzido, o sistema de rastreamento precisa mover o laser continuamente para fora. À medida que o laser se move para fora em relação ao centro do disco, as saliências passam pelo laser com maior velocidade. Isso acontece por causa da velocidade linear ou tangencial das saliências, que é igual ao raio do disco multiplicado pela velocidade na qual o disco está girando. Assim, conforme o laser se move para fora, o motor do fuso precisa diminuir a velocidade de giro do DVD, para que os ressaltos passem pelo laser em velocidade constante e os dados sejam lidos do disco também a uma taxa constante.*

*Algo interessante de observar é que, se um DVD possui uma segunda camada, a trilha pode ser iniciada no lado externo do disco, e não no interno. Isso permite que o aparelho faça uma transição rápida de uma camada para a próxima, sem retardo na saída de dados, pois não é preciso mover o laser de volta para o centro do disco para ler a camada seguinte.*

## AD2 - Organização de Computadores 2010.1

Data de entrega: 22/05/2010

### ERRATA: Questão 4

4. (1,8) Um sistema de computação possui somente 6 bits para representar dados. Considere o seguinte conjunto de bits:

A=111111

B=100001

C=011111

D=000011

- a) (0,6) Indique o valor decimal para A, B, C e D considerando que eles representam:

- i) inteiros sem sinal
- ii) inteiros em sinal e magnitude
- iii) inteiros em complemento a 2

As seguintes operações foram realizadas neste sistema:

E=A+B

F=B-C

G=C+D

H=A-D

**i). inteiros sem sinal**

$$A=111111 = 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = +63$$

$$B=100001 = 2^5 + 2^0 = +33$$

$$C=011111 = 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = +31$$

$$D=000011 = 2^1 + 2^0 = +3$$

**ii) inteiros em sinal e magnitude**

$$A=111111 = - (2^4 + 2^3 + 2^2 + 2^1 + 2^0) = -31$$

$$B=100001 = - (2^0) = -1$$

$$C=011111 = + (2^4 + 2^3 + 2^2 + 2^1 + 2^0) = +31$$

$$D=000011 = + 2^1 + 2^0 = +3$$

**iii) inteiros em complemento a 2**

$$A=111111 = -2^5 + (2^4 + 2^3 + 2^2 + 2^1 + 2^0) = -1$$

$$B=100001 = -2^5 + (2^0) = -31$$

$$C=011111 = +2^4 + 2^3 + 2^2 + 2^1 + 2^0 = +31$$

$$D=000011 = +2^1 + 2^0 = +3$$

- b) (1,2) Indique o valor decimal para E, F, G e H considerando que A, B, C, D, E, F, G e H representam:

- i) inteiros sem sinal
- ii) inteiros em sinal e magnitude
- iii) inteiros em complemento a 2

Explique, para cada uma das 4 operações acima, se a operação irá causar estouro (overflow) ou não.

### i) inteiros sem sinal

$$E=A+B = 111111 + 100001 = 1) 100000 = ( +32 )_{10} \Rightarrow \text{HOUE OVERFLOW, PRESENÇA DE BIT DE ESTOURO}$$

$$F=B-C = 100001 - 011111 = 000010 = ( +02 )_{10} \Rightarrow \text{NÃO HOUE OVERFLOW}$$

$$G=C+D = 011111 + 000011 = 100010 = ( +34 )_{10} \Rightarrow \text{NÃO HOUE OVERFLOW}$$

$$H=A-D = 111111 - 000011 = 111100 = ( +60 )_{10} \Rightarrow \text{NÃO HOUE OVERFLOW}$$

*Ocorrerá overflow para E, porque o valor correto da soma é 64 que não é possível de ser representado com 6 bits utilizando-se representação inteiro sem sinal.*

*Não ocorrerá overflow para F, G e H porque seu valores podem ser representados com 6 bits utilizando-se representação inteiro sem sinal.*

### ii) inteiros em sinal e magnitude

$$E=A+B = 111111 + 100001 = 100000 = ( 0 )_{10} \Rightarrow \text{HOUE OVERFLOW}$$

$$\begin{aligned} F=B-C &= 100001 - 011111 \Rightarrow \text{TRANSFORMAR O 2ºTERMO PARA NEGATIVO} \\ &= 100001 + 111111 \Rightarrow 100000 = ( 0 )_{10} \Rightarrow \text{HOUE OVERFLOW} \end{aligned}$$

$$G=C+D = 011111 + 000011 \Rightarrow 000010 = ( 2 )_{10} \Rightarrow \text{HOUE OVERFLOW}$$

$$\begin{aligned} H=A-D &= 111111 - 000011 \Rightarrow \text{TRANSFORMAR O 2ºTERMO PARA NEGATIVO} \\ &= 111111 + 100011 \Rightarrow 100010 = ( -2 )_{10} \Rightarrow \text{HOUE OVERFLOW} \end{aligned}$$

*Ocorrerá overflow para E e F, ambos com valor correto igual a -32. Ocorrerá também para G cujo valor é +34 e para H cujo valor é -34 que não é possível de ser representado com 6 bits utilizando-se a representação sinal e magnitude.*

### iii) inteiros em complemento a 2

$$E=A+B = 111111 + 100001 = 1) 100000 = ( -32 )_{10} \Rightarrow \text{NÃO HOUE OVERFLOW (soma de dois números negativos teve como resultado um número negativo)}$$

$$\begin{aligned} F=B-C &= 100001 - 011111 = \Rightarrow \text{TRANSFORMAR O 2ºTERMO PARA NEGATIVO} \\ &= 100001 + 100001 = 1) 000010 = ( +2 )_{10} \Rightarrow \text{HOUE OVERFLOW (soma de dois números negativos teve como resultado um número positivo)} \end{aligned}$$

$$G=C+D = 011111 + 000011 = 100010 = ( -30 )_{10} \Rightarrow \text{HOUE OVERFLOW (soma de dois números positivos teve como resultado um número negativo)}$$

$$\begin{aligned} H=A-D &= 111111 - 000011 = \Rightarrow \text{TRANSFORMAR O 2ºTERMO PARA NEGATIVO} \\ &= 111111 + 111101 = 1) 111100 = ( -4 )_{10} \Rightarrow \text{NÃO HOUE OVERFLOW (soma de dois números negativos teve como resultado um número negativo)} \end{aligned}$$

*Ocorrerá overflow para G cujo valor correto é +34 e que não é possível de ser representado com 6 bits utilizando-se a representação por complemento a 2*

*Ocorrerá overflow para F cujo valor correto é -62 e que não é possível de ser representado com 6 bits utilizando-se a representação por complemento a 2*

*Não ocorrerá overflow para E e H pois seus valores podem ser representados com 6 bits utilizando-se a representação complemento a 2.*