

Gabarito AD2 - Organização de Computadores 2011.2

Data de entrega 15/10/2011

"Atenção: 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. (1,0) Descreva detalhadamente e explique, exemplificando, todos os tipos de endereçamento de dados visto em aula.

Imediato: O campo operando contém o dado, desta forma o dado é transferido da memória juntamente com a instrução.

Vantagem: Rapidez na execução da instrução, pois não requer acesso à memória principal, apenas na busca da própria instrução.

Desvantagem. Limitação do tamanho do campo operando das instruções reduzindo o valor máximo do dado a ser manipulado. Trabalho excessivo para alteração de valores quando o programa é executado repetidamente e o conteúdo das variáveis serem diferentes em cada execução.

Exemplo:

$LDI\ Op \Rightarrow ACC <--- Op$ = carrega o valor (hexadecimal) do operando no acumulador.

$LDI\ 80\ (load\ immediate) \Rightarrow ACC <---- 80$ (carrega o valor hexadecimal 50 no acumulador)

Direto: O campo operando da instrução contém o endereço onde se localiza o dado.

Vantagem. Flexibilidade no acesso a variáveis de valor diferente em cada execução do programa

Desvantagem. Limitação de memória a ser usada conforme o tamanho do operando.

Exemplo:

$LDA\ Op\ ("load\ accumulator")$, sendo Op um endereço na memória $\Rightarrow ACC <--- (Op)$

(carrega o conteúdo da posição de memória indicada pelo operando no acumulador).

$LDA\ 7B5C \Rightarrow ACC <--- (7B5C)$ = carrega no acumulador o conteúdo da posição de memória 7B5C.

Indireto: O campo de operando contém o endereço de uma célula, sendo o valor contido nesta célula o endereço do dado desejado.

Vantagem: Usar como "ponteiro". Elimina o problema do modo direto de limitação do valor do endereço do dado. Manuseio de vetores (quando o modo indexado não está disponível).

Desvantagem: Muitos acesso à MP para execução, requer pelo menos 2 acessos à memória principal.

Exemplo:

$ADD\ M\ (add\ memory) \Rightarrow (ACC) <--- (ACC) + ((M))$ -> soma o conteúdo da posição de memória indicada por M ao conteúdo do acumulador

Por registrador: característica semelhante aos modos direto e indireto, exceto que a célula (ou palavra) de memória referenciada na instrução é substituída por um dos registradores da UCP. O endereço mencionado na instrução passa a ser o de um dos registradores.

Vantagens: Menor quantidade de bits para endereçar os registradores, por consequência, redução da instrução. E o dado pode ser armazenado em um meio mais rápido (registrador).

Desvantagens: Devido ao número reduzido de registradores existentes na UCP causa uma dificuldade em se definir quais dados serão armazenados nos registradores e quais permanecerão na UCP.

Exemplo:

ADD r (add register) ==> (ACC) <--- (ACC) + (r) -> soma o conteúdo do registrador r ao conteúdo do acumulador (endereçamento por registrador, direto)

No modo indexado: *consiste em que o endereço do dado é a soma do valor do campo operando (que é fixo para todos os elementos de um dado vetor) e de um valor armazenado em um dos registradores da UCP (normalmente denominado registrador índice).*

Vantagem: Rapidez de execução das instruções de acesso aos dados, visto que a alteração do endereço dos elementos é realizada na própria UCP

Exemplo:

LDX Ri, Op ==> ACC <--- ((Op) + (Ri))

ADX Ri, Op ==> ACC <--- ACC + ((Op) + (Ri))

No modo de endereçamento 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.*

Vantagem: Reduz o tamanho das instruções e facilita o processo de relocação de programas.

Exemplo:

LDB Op ==> (ACC) <--- ((Rb) + (Op)) (modo de endereçamento Base + Deslocamento)

2. (1,0) Explique a diferença entre compiladores e interpretadores, especificando vantagens e desvantagens de cada tipo. Pesquise exemplos atuais de linguagens de programação compiladas e interpretadas e seus usos mais comuns.

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.

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.

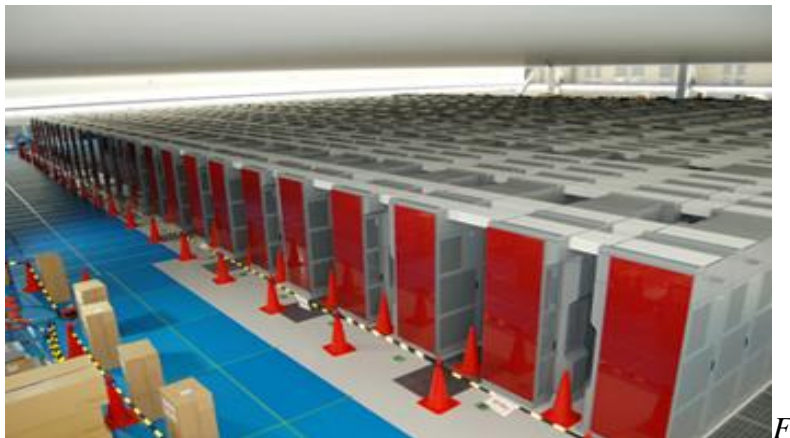
Exemplos de linguagem interpretadas: ASP, BASIC, Java, PHP, Python, Lisp entre outras.

Exemplos de linguagem compilada: C, Pascal, Delphi, Visual Basic, entre outras.

3. (1,0) Pesquise no site www.top500.org os cinco sistemas computacionais de maior desempenho e os descreva. Ache no mesmo site o sistema computacional brasileiro com maior desempenho e o descreva.

1o. lugar: K computer

O "K" supercomputador japonês, um projeto conjunto entre Fujitsu e do centro RIKEN, atualmente na fase de configuração, características 68.544 Sparc64 CPUs de 2.0GHz, cada uma com oito núcleos, para um total de 548.352 núcleos, quase o dobro do que qualquer outro sistema no TOP500. O Computador K também é mais poderoso do que os próximos cinco sistemas na lista combinada. O sistema está implantado em 672 racks de computador e tem um consumo de 9,89 megawatts. O nome do computador chama a K sobre a palavra japonesa "Kei" por 10^{16} (dez quatrilhões), representando o objetivo do sistema de desempenho de 10 petaflops. Usa como sistema operacional o Linux.



2o. lugar: Tianhe-1A - National Supercomputing Center in Tianjin

O sistema Tianhe-1A chinês no Centro Nacional de Supercomputação em Tianjin, alcança um nível de desempenho de 2,6 petaflops / s (quatrilhões de cálculos de ponto flutuante por segundo). Tianhe-1A é um dos 19 sistemas no Top 500 que usam GPUs NVIDIA (unidades de processamento gráfico) para acelerar a computação. Sistema operacional utilizado é o Linux. Arquitetura híbrida com processadores heterogêneos. Possui cerca de 6144 Intel Xeon CPU e 5120 AMD GPUs. Memória total de cerca de 98TB. Utilizado na área de exploração de petróleo, recursos de medicina biológica, simulação de projeto de grandes aeronaves, entre outros.



Fonte: <http://www.nscj-tj.gov.cn/en/>

3o. lugar: Jaguar - DOE/SC/Oak Ridge National Laboratory

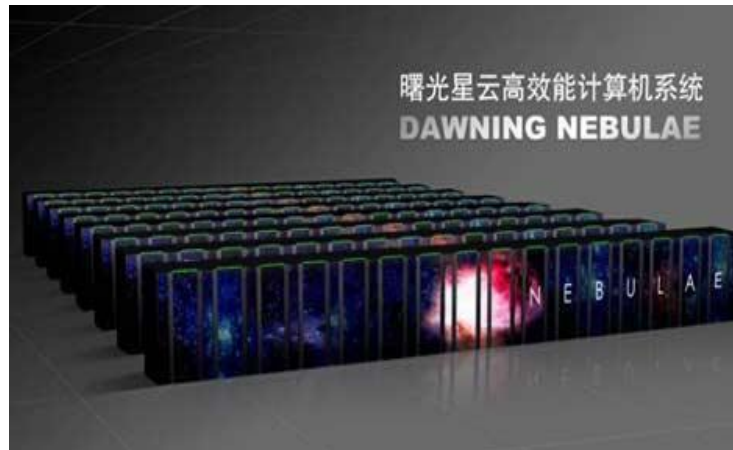
Jaguar foi o melhor desempenho em Junho de 2010 da pesquisa, mas caiu para a segunda em novembro de 2010 e agora escorrega para terceiro da geral com uma velocidade de 1,75 petaflops de desempenho rodando o benchmark Linpack. Jaguar é um Cray XT5 sistema localizado no Departamento de Energia do Oak Ridge do Leadership Computing Facility, no Tennessee. Enquanto Jaguar caíram alguns flops curta este ano, mantém uma posição de liderança no estilo de supercomputação com seu motivo Jaguar impressionante. Utiliza-se de processadores AMD x86_64 Opteron Six Core 2600 MHz



Fonte: <http://www.datacenterknowledge.com/the-top-10-supercomputers-illustrated-june-2011/>

4o. lugar: Nebulae - [National Supercomputing Centre in Shenzhen \(NSCS\)](#)

Está localizado no Centro Nacional de Supercomputação, em Shenzhen, China. Atingiu 1,271 PFlop / s para classificar em quarto lugar, uma posição abaixo do levantamento de novembro. Nebulas foi construído a partir de um sistema Dawning TC3600 com processadores Intel X5650 processadores e GPUs NVIDIA Tesla C2050. O sistema operacional utilizado neste sistema é o Linux.



Fonte: <http://www.datacenterknowledge.com/the-top-10-supercomputers-illustrated-june-2011/>

5o. Colocado: Tsubame 2.0 - [GSIC Center, Tokyo Institute of Technology](#)

Tsubame 2.0 é um sistema de sucessor que se baseia no projeto de um sistema já previamente classificado. Tsubame 2.0 foi desenvolvido pelo Instituto de Tecnologia de Tóquio, em colaboração com a NEC e HP, e é alimentado por mais de 1.400 nós usando ambos os servidores HP Proliant e NVIDIA Tesla GPUs. Planos estão sendo desenvolvidos para Tsubame 3.0. Possui em sua arquitetura processadores Intel EM64T Xeon X56XX de 2930MHz.



Fonte: <http://www.datacenterknowledge.com/the-top-10-supercomputers-illustrated-june-2011/>

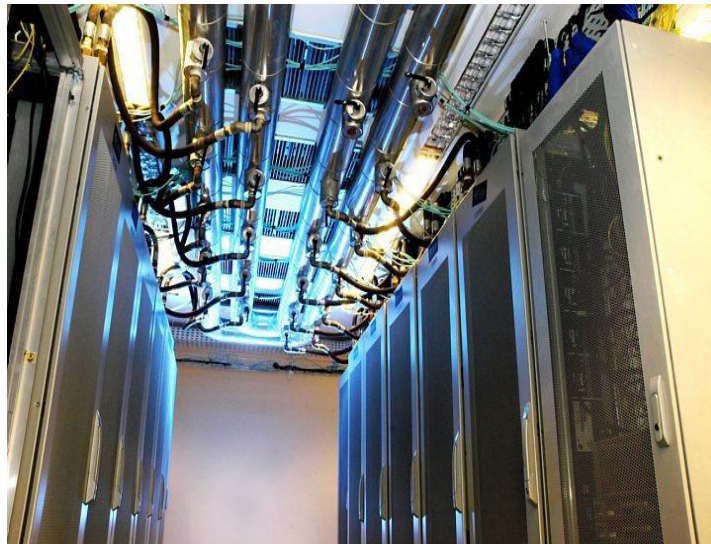
Sistema nacional mais bem colocado está na 34a. Posição, consiste no Tupã, instalado no Instituto Nacional de Pesquisas Espaciais (INPE), e alcançou a 29a posição na avaliação de nov/2010.

O Brasil passou a ter o supercomputador mais rápido do hemisfério sul e, pela primeira vez na história, atingiu a 29ª colocação da lista das máquinas mais poderosas em atividade em todo mundo. De acordo com o TOP500, órgão que avalia os supercomputadores, o Tupã, instalado no Instituto Nacional de Pesquisas Espaciais (Inpe) de Cachoeira Paulista (a 202 km de São Paulo) desde outubro, é o terceiro mais poderoso entre os equipamentos dedicados à previsão de tempo e de clima sazonal. A máquina, um XT6 da Cray, é capaz de executar 258 TFlops, equivalente a 258 trilhões de cálculos de ponto-flutuante por segundo. O equipamento

comprado pelo Ministério da Ciência e Tecnologia e pela Fapesp por US\$ 23 milhões e é cerca de 2.580 vezes mais veloz que as máquinas domésticas.



Na lista da top 500, além do Tupã temos na 167a. Posição o Galileu – [NACAD/COPPE/UFRJ](#), da Oracle/Sun, pertencente ao Núcleo de Atendimento em Computação de Alto Desempenho (NACAD) da COPPE/UFRJ. O Galileu tem 7.200 processadores e capacidade computacional total de 64,63 teraflops (operações de ponto flutuante) por segundo. A memória total do sistema é de 21 terabytes e o armazenamento em disco, de 50 terabytes. O Galileu foi financiado pelo Fundo de Participação Especial da Petrobras/ANP (um investimento de R\$ 9,5 milhões) e, não por acaso, seu principal uso se relaciona ao pré-sal - em simulações de prospecção em águas profundas - e a modelos meteorológicos.



4. (1,0) Crie instruções com zero, um e dois operandos. Para cada um dos 3 casos, faça um programa que resolva a seguinte equação: $A = B \times C + D - E$. Pressuponha que os dados estão na memória principal, inicialmente, e que existem 10 registradores de propósito geral. As instruções aritméticas usam endereçamento por registrador.

CASO 1: CONJUNTO DE INSTRUÇÕES PARA 1 OPERANDO:

consideremos os seguintes registradores de uso geral R1,R2,R3,R4,R5,R6,R7,R8,R9 e o R0 que aqui também chamaremos de ACC. Seja Rx qualquer um dos R1 até R9. E seja Z endereço de memória

ADD Rx	=>	(ACC) <- (ACC) + (Rx)	ou	(R0) <- (R0) + (Rx)
SUB Rx	=>	(ACC) <- (ACC) - (Rx)	ou	(R0) <- (R0) - (Rx)
MUL Rx	=>	(ACC) <- (ACC) * (Rx)	ou	(R0) <- (R0) * (Rx)
DIV Rx	=>	(ACC) <- (ACC) / (Rx)	ou	(R0) <- (R0) / (Rx)
MVx Rx	=>	(Rx) <- (ACC)	ou	(Rx) <- (R0)
MV0 Rx	=>	(ACC) <- (Rx)	ou	(R0) <- (Rx)
LOAD Z	=>	(ACC) <- (Z)	ou	(R0) <- (Z)
STORE Z	=>	(Z) <- (ACC)	ou	(Z) <- (R0)

CASO 2: CONJUNTO DE INSTRUÇÕES PARA 2 OPERANDOS:

consideremos os seguintes registradores de uso geral R1,R2,R3,R4,R5,R6,R7,R8,R9 e o R0 que aqui poderemos chamar de ACC. Seja Rx qualquer um dos R1 até R9. E seja Z endereço de memória

ADD Rx, Ry	=>	(Rx) <- (Rx) + (Ry)
SUB Rx, Ry	=>	(Rx) <- (Rx) - (Ry)
MUL Rx, Ry	=>	(Rx) <- (Rx) * (Ry)
DIV Rx, Ry	=>	(Rx) <- (Rx) / (Ry)
MOV Rx, Ry	=>	(Rx) <- (Ry)
LOAD Z	=>	(R0) <- (Z)
STORE Z	=>	(Z) <- (R0)

CASO 3: CONJUNTO DE INSTRUÇÕES PARA 0 OPERANDOS:

Para este modo utilizaremos um organização de memória especial, denominada pilha (livro texto Willian Stallings, 5a. Edição pag. 346 e apêndice 9A na página 385).

Operações com pilhas:

PUSH => Coloca um novo elemento no topo da pilha

POP => Retira o elemento do topo da pilha

Operação unária => Efetua a operação sobre o elemento do topo da pilha. Substitui o elemento do topo do resultado.

Operação binária => Efetua a operação sobre os dois elementos no topo da pilha. Retira os dois elementos do topo da pilha. Coloca o resultado da operação no topo da pilha.

PUSH Z	=>	Coloca o conteúdo de Z no topo da pilha
POP Z	=>	Retira o elemento do topo da pilha e coloca em Z
ADD	=>	Soma os 2 ultimos elementos da pilha armazenando o resultado. Equivale em algoritmo a:
		POP X
		POP Y
		$Z \leftarrow X + Y$
		PUSH Z

SUB	=>	Subtrai os 2 ultimos elementos da pilha. Equivale a:
		POP X
		POP Y
		$Z \leftarrow X - Y$
		PUSH Z

MUL => Multiplica os 2 ultimos elementos da pilha. Equivale a:

POP X
POP Y
 $Z \leftarrow X * Y$
PUSH Z

DIV => Divide os 2 ultimos elementos da pilha. Equivale a:

POP X
POP Y
 $Z \leftarrow X / Y$
PUSH Z

Expressão: $A = B \times C + D - E$

SOLUÇÃO PARA O CASO 1:

LOAD B => (R0) <- (B)
MVX R1 => (R1) <- (R0)
LOAD C => (R0) <- (C)
MVX R2 => (R2) <- (R0)
LOAD D => (R0) <- (D)
MVX R3 => (R3) <- (R0)
LOAD E => (R0) <- (E)
MVX R4 => (R4) <- (R0)
MOV R1 => (R0) <- (R1)
MUL R2 => (R0) <- (R0) * (R2)
ADD R3 => (R0) <- (R0) + (R3)
SUB R4 => (R0) <- (R0) - (R4)
STORE A => (A) <- (R0)

SOLUÇÃO PARA O CASO 2:

LOAD B => (R0) <- (B)
MOV R1, R0 => (R1) <- (R0)
LOAD C => (R0) <- (C)
MUL R1, R0 => (R1) <- (R1) * (R0)
LOAD D => (R0) <- (D)
ADD R1, R0 => (R1) <- (R1) + (R0)
LOAD E => (R0) <- (E)
SUB R1, R0 => (R1) <- (R1) - (R0)
MOV R0, R1 => (R0) <- (R1)
STORE A => (A) <- (R0)

SOLUÇÃO PARA O CASO 3:

Inicio fila = / (vazia)

PUSH B => enfileira conteúdo de B

fila = /(B)/

PUSH C => enfileira conteúdo de C

fila = /(B)/(C)/

MUL => multiplica 2 últimos números e enfileira resultado

fila = /((B)*(C)) /

PUSH D => enfileira conteúdo de D

fila = /((B)*(C)) / (D) /

ADD => Soma 2 últimos números e enfileira resultado

fila = /((B)*(C)+(D)) /

POP T => retira valor fila e guarda em endereço temporário

$$fila = / \quad (vazia)$$
PUSH E => enfileira conteúdo de E

$$fila = /(E) /$$
PUSH T => enfileira valor armazenado no endereço temporário

$$fila = / (E) / ((B)*(C)+(D)) /$$
SUB \Rightarrow *Subtrai os 2 últimos números e enfileira resultado*

$$fila = / ((B)*(C)+(D) - (E)) /$$
POP A \Rightarrow *retira valor fila e guarda no endereço A*

$$fila = / \quad (vazia)$$

5. (1,0) Explique a diferença entre o endereçamento por registrador base mais deslocamento, e endereçamento por registrador índice. Explique em detalhes suas aplicabilidades, usando também um exemplo.

O base mais deslocamento tem como seu principal objetivo permitir a modificação de endereço de programas ou módulos destes (que é a relocação de programa), bastando para isso uma única alteração no registrador base.

O base mais deslocamento tem como característica o endereço ser obtido da soma do deslocamento com o registrador base, diferindo do modo indexado onde o do registrador base é fixo e variar no deslocamento, ao contrário deste onde o deslocamento é fixo e com a alteração do registrador base permite-se a mudança do endereço.

Exemplos de instruções modo indexado:

LDX Ri, Op ==> ACC <--- ((Op) + (Ri))

ADX Ri, Op ==> ACC <--- ACC + ((Op) + (Ri))

Exemplo: instrução base mais deslocamento:

LDB Rb, Op ==> (ACC) <--- ((Op) + (Rb))

ADB Rb, Op ==> ACC <--- ACC + ((Op) + (Rb))

6. (2,0) Considere as seguintes variáveis com as atribuições de valores decimais:

A= -15;

B=-1;

C=+2;

D=+14

- 6.1. Considere uma máquina que utiliza 5 bits para representar inteiros com sinal em sinal e magnitude e não detecta quando ocorre estouro nos valores atribuídos às variáveis.

- 6.1.1. (0,4) Indique a representação dos valores atribuídos nesta máquina às variáveis A, B, C e D.

A = 11111

B = 10001

C = 00010

D = 01110

- 6.1.2. (0,6) Considere que as seguintes atribuições foram executadas nesta máquina:

E=A+B;

F=B-D;

G=C+D;

H=A-C;

Indique o valor em **decimal** atribuído às variáveis E, F, G e H.

Em uma máquina que utiliza inteiros representados em sinal e magnitude, o algoritmo de soma consiste nos seguintes passos:

1. Verificar sinais das parcelas.
2. Caso sinais das parcelas sejam iguais, sinal do resultado será o sinal das parcelas e o módulo do resultado será dado pela soma dos módulos expressos em bits.
3. Caso contrário, será verificada a parcela que possui maior módulo e o sinal do resultado será o sinal desta parcela.
4. O módulo do resultado será dado pela subtração da parcela de menor módulo da parcela de maior módulo.

$E = A + B \Rightarrow 1111 + 10001 =$ parcelas com mesmo sinal 1, resultado terá sinal 1 e somam-se os módulos 1111 e 00001 para se obter o módulo resultante, resultado será 10000 $\Rightarrow 0_{10}$ (esperado -16) (houve estouro ou overflow, ou seja, o resultado da soma é um valor que não pode ser representado com 5 bits em sinal e magnitude)

$F = B - D = B + (-D) \Rightarrow$ parcelas com mesmo sinal 1, resultado terá sinal 1 e somam-se os módulos 0001 com 1110 para se obter o módulo resultante, resultado será 1111 $\Rightarrow -15_{10}$ (esperado -15)

$G = C + D \Rightarrow$ parcelas com mesmo sinal 0, resultado terá sinal 0 e somam-se os módulos 0010 e 1110 para se obter o módulo resultante, resultado será 00000 $\Rightarrow +0_{10}$ (esperado +16)(houve estouro ou overflow, ou seja, o resultado da soma é um valor que não pode ser representado com 5 bits em sinal e magnitude)

$H = A - C = A + (-C) \Rightarrow$ parcelas com mesmo sinal 1, resultado terá sinal 1 e somam-se os módulos 1111 e 0010 para se obter o módulo resultante, resultado será 10001 $\Rightarrow -1_{10}$ (esperado -17)(houve estouro ou overflow, ou seja, o resultado da soma é um valor que não pode ser representado com 5 bits em sinal e magnitude)

- 6.2. Considere uma máquina que utiliza 5 bits para representar inteiros com sinal em complemento a 2 e não detecta quando ocorre estouro nos valores atribuídos às variáveis.

6.2.1. (0,4) Indique a representação dos valores atribuídos nesta máquina às variáveis A, B, C e D.

$A = 10001$
 $B = 11111$
 $C = 00010$
 $D = 01110$

6.2.2. (0,6) Considere que as seguintes atribuições foram executadas nesta máquina:

$E = A + B;$

$F = B - D;$

$G = C + D;$

$H = A - C;$

Indique o valor em **decimal** atribuído às variáveis E, F, G e H.

Em uma máquina que utiliza inteiros representados em complemento a 2, o algoritmo de soma consiste em somar as duas parcelas bit a bit.

$E = A + B \Rightarrow 10001 + 11111 = 10000 \Rightarrow -16_{10}$ (esperado -16)

$F = B - D \Rightarrow 11111 + (-01110) =$ (deve-se obter e representação negativa da segunda parcela)
 $= 11111 + 10010 = 10001 \Rightarrow -15_{10}$ (esperado -15)

$G = C + D \Rightarrow 00010 + 01110 = 10000 \Rightarrow -16_{10}$ (esperado +16)(houve estouro ou overflow, ou seja, o resultado da soma é um valor que não pode ser representado com 5 bits em complemento a 2. Observe que neste caso, somamos dois números positivos e o resultado incorreto é negativo)

$H = A - C \Rightarrow 10001 + (-00010) =$ (deve-se obter e representação negativa da segunda parcela)
 $= 10001 + 11110 = 01111 \Rightarrow +15_{10}$ (esperado -17) (houve estouro ou overflow, ou seja, o resultado da soma é um valor que não pode ser representado com 5 bits em complemento a 2. Observe que neste caso, somamos dois números negativos e o resultado incorreto é positivo)

7. (1,0) Considere um computador, cuja representação para ponto fixo e para ponto flutuante utilize 24 bits.

7.1. (0,3) Considere o seguinte conjunto de bits representado em hexadecimal FB0800. Indique o valor deste número **em decimal**, considerando-se que o conjunto representa:

$$(FB0800)_{16} = (11111011\ 00001000\ 00000000)_2$$

7.1.1. um inteiro sem sinal

$$2^{23} + 2^{22} + 2^{21} + 2^{20} + 2^{19} + 2^{17} + 2^{16} + 2^{11} = +16.451.584$$

7.1.2. um inteiro em sinal magnitude

$$-(2^{22} + 2^{21} + 2^{20} + 2^{19} + 2^{17} + 2^{16} + 2^{11}) = -8.062.976$$

7.1.3. um inteiro em complemento a 2

$$-2^{23} + (2^{22} + 2^{21} + 2^{20} + 2^{19} + 2^{17} + 2^{16} + 2^{11}) = -325.632$$

7.2. (0,7) Na representação em ponto flutuante, como na representação IEEE 754, utiliza-se o bit mais à esquerda para representar o sinal, os próximos 8 bits representam o expoente e os 15 bits seguintes representam os bits depois da vírgula. Quando todos os bits que representam o expoente são iguais a 0 ou iguais a 1 temos os casos especiais. Caso contrário, as combinações possíveis de bits representam números normalizados no formato $\pm (1, b_1b_2b_3b_4b_5b_6b_7b_8b_9b_{10}b_{11}b_{12}b_{13}b_{14}b_{15} \times 2^{\text{expoente}})$, onde o bit mais à esquerda representa o sinal (0 para números positivos e 1 para números negativos), os próximos 8 bits representam o expoente em excesso e os 15 bits seguintes representam os bits b_1 a b_{15} , como mostrado na figura a seguir:

S	Expoente representado em excesso	$b_1\ b_2\ b_3\ b_4\ b_5\ b_6\ b_7\ b_8\ b_9\ b_{10}\ b_{11}\ b_{12}\ b_{13}\ b_{14}\ b_{15}$
1	8	15

7.2.1. (0,2) Determine o valor do excesso utilizado, sabendo que os projetistas desta máquina utilizaram o mesmo critério utilizado pelo padrão IEEE754 para definir o valor do excesso.

O excesso será determinado pela fórmula: $2^{n-1} - 1$, sendo n o número de bits utilizado para representar o número.

Como são utilizados 8 bits para representar o expoente, o valor do excesso deve ser $2^{n-1} - 1 = 127$.

7.2.2. (0,2) Indique o valor do conjunto de bits do item anterior considerando que este conjunto está representando um número normalizado em ponto flutuante com a representação acima.

1 11110110 000100000000000

Sendo:

Sinal = 1 \Rightarrow negativo

Expoente = 11110110 = $246 - 127 \Rightarrow$ expoente = 119

Mantissa = 0001000000000000

$$\begin{aligned} \text{Temos então} \Rightarrow & -1,0001 \times 2^{119} \\ & -10001 \times 2^{115} = 7,06 \times 10^{+35} \end{aligned}$$

7.2.3. (0,2) Qual será a representação em ponto flutuante dos seguintes valores decimais neste computador:

7.2.3.1. +10,75

Convertendo para binário = 1010,11 \Rightarrow *normalizando* $1,01011 \times 2^3$

Temos então:

Sinal = 0 (positivo)

Expoente = (3 + 127 = 130) = 10000010

Mantissa = ,01011

Resultado: 0 10000010 0101100000000000

7.2.3.2. -0,01

Convertendo para binário = 0,0000001010001111010111

normalizando $1,010001111010111 \times 2^{-7}$

Temos então:

Sinal = 1 (negativo)

Expoente = (-7 + 127 = 120) = 01111000

Mantissa = ,010001111010111

Resultado: 1 01111000 010001111010111

7.2.4. (0,1) Indique o menor e o maior valor positivo **normalizado** na representação em ponto flutuante para este computador. Mostre os valores **em decimal**.

$$\begin{aligned} \text{Maior número positivo: } & 0 \ 11111110 \ 111111111111 = + (1,111111111111)_2 \times 2^{127} \approx \\ & + 3,4 \times 10^{+38} \end{aligned}$$

$$\text{Menor número positivo: } 0 \ 00000001 \ 000000000000 = + (1,0)_2 \times 2^{-126} \approx + 1,18 \times 10^{-38}$$

8. (0,5) Explique como funciona o sistema Universal Serial Bus (USB) utilizado para conectar dispositivos de Entrada/Saída em um computador (sugestão de fonte de consulta: o site <http://informatica.hsw.uol.com.br/>. Na sua resposta indique as suas fontes de consulta).

Texto retirado do site <http://informatica.hsw.uol.com.br/>

As portas USB permitem que se conectem dispositivos como mouses e impressoras a um computador. São suportáveis pelos sistemas operacionais atuais e sua utilização é bem mais simples quando comparada a outras formas de conexão de dispositivos (incluindo-se portas paralelas, portas seriais e placas especiais instaladas dentro do gabinete da máquina).

Em caso de um dispositivo novo, recém-conectado, o sistema operacional o auto-detecta e solicita o disco do driver do dispositivo. Quando o driver for corretamente instalado, o computador ativará o dispositivo e dará início à comunicação com o mesmo. Os dispositivos USB poderão ser conectados e desconectados a qualquer momento.

*Quando o host (computador) é iniciado, ele interroga todos os dispositivos conectados ao barramento USB e designa um endereço para cada um. Esse processo é chamado de **enumeração**.*

Os dispositivos são também enumerados ao se conectarem ao barramento. O host também obtém de cada dispositivo a informação sobre o tipo de transferência de dados que o mesmo deseja realizar:

- **Interrupção** - usado em dispositivos como o mouse ou teclado, que enviarão pouquíssimos dados.
- **Bulk** - um dispositivo como uma impressora, que recebe dados em grandes pacotes, utiliza esse modo de transferência. Um bloco de dados é enviado à impressora (em fragmentos de 64 bytes) e existem procedimentos para verificar se ele foi transferido de forma correta.
- **Isócrono** - um dispositivo de transmissão contínua (como alto-falantes) utiliza esse modo. Os dados fluem entre o dispositivo e o host em tempo real e não há correção de erros.

O host também poderá enviar comandos ou verificar parâmetros através do envio e recebimento de pacotes de controle.

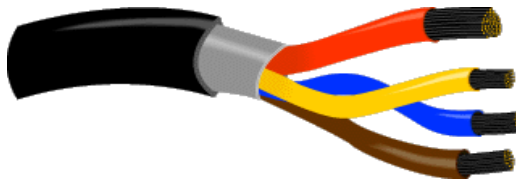
Conforme os dispositivos são enumerados, o host manterá um registro da largura de banda total que todos os dispositivos isócronos e os dispositivos de interrupção exigem. Eles podem consumir até 90% dos 480 Mbps da largura de banda disponível. Após 90% de uso, o host nega acesso a qualquer outro dispositivo isócrono ou de interrupção. As transferências de grandes pacotes de dados e de controle utilizam qualquer largura de banda restante (pelo menos 10%).

O USB divide a largura de banda disponível em **frames** e o host controla os frames. Os frames contêm 1.500 bytes e um novo frame tem início a cada milissegundo. Durante um frame, dispositivos isócronos e de interrupção obterão um slot, assim as larguras de banda necessárias serão garantidas. As transferências de pacotes de dados e de controle utilizam qualquer espaço que sobrar.

O padrão USB 2.0 foi lançado em abril de 2000, como uma versão atualizada do USB 1.1. O USB 2.0 (**USB de alta velocidade**) fornece uma largura de banda adicional para as aplicações de multimídia e de armazenamento. O USB 2.0 suporta três velocidades (1.5, 12 e 480 megabits por segundo), e suporta tanto dispositivos que necessitam baixa largura de banda, tais como teclados e mouses, bem como os dispositivos que necessitam alta largura de banda, como: Webcams, scanners, impressoras e sistemas de armazenamento de alta capacidade.

O Universal Serial Bus possui as seguintes características:

- O computador atua como um **host**.
- Podem ser conectados ao host até **127 dispositivos**, diretamente ou através de hubs USB.
- Cabos individuais USB podem ter até 5 metros; com hubs, os dispositivos podem ficar até 30 metros de distância do host (o equivalente a seis cabos).
- Com o USB 2.0, o barramento possui uma taxa máxima de transferência de dados de **480 megabits por segundo**.
- Um cabo USB possui dois fios para energia (+5 volts e o fio terra) e um par trançado para a condução dos dados.



Interior de um cabo USB: existem dois fios para a energia, +5 volts (vermelho) e o fio terra (marrom), e um par trançado (amarelo e azul) de fios para a condução dos dados. O cabo é blindado.

- Nos cabos de energia, o computador poderá fornecer até 500 milliampères de energia a 5 volts.
- Os dispositivos de baixa-potência (como o mouse) poderão ser alimentados diretamente do barramento. Os dispositivos de alta-potência (como impressoras) possuem fonte própria de

alimentação e exigem pouca energia do barramento. Os hubs podem ter suas próprias fontes de energia para fornecer energia aos dispositivos conectados a ele.

- *Os dispositivos USB são **hot-swappable** (conectáveis "a quente"), ou seja, podem ser conectados e desconectados a qualquer momento.*

9. (0,5) Explique o funcionamento de impressoras a jato de tinta (indique as suas fontes de consulta).

Texto retirado de <http://www.hardware.com.br/termos/impressora-jato-de-tinta>

As impressoras jato de tinta trabalham espirrando gotículas de tinta sobre o papel, conseguindo uma boa qualidade de impressão, próxima à de impressoras a laser. Outra vantagem destas impressoras é seu baixo custo, o que as torna perfeitas para o uso doméstico.

As impressoras jato de tinta podem usar basicamente três tecnologias de impressão: a Buble Jet, ou jato de bolhas, a Piezoelétrica e a de troca de estado.

A tecnologia Buble Jet foi criada pela Canon, que detém a patente do nome até hoje. Esta tecnologia consiste em aquecer a tinta através de uma pequena resistência, formando pequenas bolhas de ar, que fazem a tinta espirrar com violência sobre o papel. Esta tecnologia é usada em várias marcas de impressoras, como as da própria Canon. No caso das impressoras HP, a tecnologia recebe o nome de "Ink Jet", apesar do princípio de funcionamento ser o mesmo.

Uma desvantagem desta tecnologia é que, devido ao aquecimento, a cabeça de impressão costuma se desgastar depois de pouco tempo, perdendo a precisão. Por outro lado, por serem extremamente simples, as cabeças são baratas, e por isso são embutidas nos próprios cartuchos de impressão.

As impressoras Epson por sua vez, utilizam uma cabeça de impressão Piezoelétrica, que funciona mais ou menos como uma bomba microscópica, borrifando a tinta sobre o papel. A cabeça de impressão consiste em uma pequena canalização, com um cristal piezoelétrico próximo da ponta. Quanto recebe eletricidade, este cristal vibra, fazendo com que gotículas de tinta sejam expelidas para fora do cartucho.

Esta tecnologia traz como vantagem uma maior precisão, que permite uma impressão com uma resolução muito maior. A Epson Stylus 600, por exemplo, é capaz de imprimir com até 1440 DPI (pontos por polegada) enquanto uma HP 692C, que está na mesma faixa de preço, imprime no máximo a 600 DPI.

Como as cabeças de impressão Piezoelétricas possuem maior durabilidade e são muito mais complexas e caras do que as buble-jet, elas não são trocadas junto com os cartuchos, fazendo parte da impressora. Por um lado isso é bom, pois permite baratear um pouco os cartuchos, mas por outro lado, torna a impressora mais susceptível a problemas, como o entupimento das cabeças de impressão.

Existem também impressoras de troca de estado, que utilizam tinta sólida, um tipo de cera, geralmente em forma de fitas. Nestas impressoras, a tinta é derretida e espirrada sobre o papel. A vantagem é que, como a tinta é um tipo de cera, a impressão assume um aspecto brilhante, com qualidade semelhante à de uma foto, mesmo usando papel comum.

Um exemplo de impressora de troca de estado é a Citizen Printiva, que possui uma qualidade de impressão surpreendente, mas que demora cerca de 6 minutos para imprimir uma página colorida e possui um alto custo de impressão.

10. (1,0) Considere uma máquina cujo controlador de impressora possui três registradores para se comunicar com o resto do sistema: o registrador de comandos utilizado para receber comandos do sistema (pedido de envio de byte para a impressora, por exemplo), o registrador de estado utilizado para indicar se a impressora está pronta para receber dado, e o registrador de dados utilizado para armazenar o byte a ser enviado para a impressora. O primeiro registrador possui o endereço 40, o segundo 41 e o terceiro 42. Para que um byte seja enviado para o controlador de impressora, a

UCP deve enviar o comando 01 para o registrador de comandos do controlador de impressora. O byte só pode ser enviado para o controlador de impressora, quando o controlador de impressora armazenar o valor 01 no registrador de estado indicando que a impressora está pronta para receber o byte. Quando isto ocorrer, o byte deve ser enviado para o registrador de dados. Descreva detalhadamente os três possíveis métodos de comunicação entre o controlador de impressora com a unidade central de processamento e memória principal: por E/S programada, por interrupção e por acesso direto à memória.

E/S programada:

A UCP envia o comando 01 para o registrador de comando do controlador da impressora cujo endereço é 40. Depois, fica lendo o conteúdo do registrador de estado do mesmo controlador cujo endereço é 41, e verifica se o seu conteúdo é igual a 01. Quando o conteúdo for igual a 01, o controlador de impressora informa que está pronto para receber o dado. A UCP envia o byte a ser impresso para o registrador de dados do controlador de endereço igual a 42. Este procedimento é repetido para receber cada byte a ser impresso pela impressora

E/S por interrupção:

A UCP envia o comando 01 para o registrador de comando do controlador cujo endereço é 40. Depois, vai executar outras instruções. Quando o controlador da impressora está pronto para receber o dado, ele 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 está sendo executado e atende a interrupção, enviando o byte a ser impresso para o registrador de dados cujo endereço é 42. Este procedimento é repetido para receber cada byte a ser impresso pela impressora

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 da impressora, o endereço inicial da memória onde estão armazenados os bytes a serem enviados para a impressora, o número de bytes e a indicação de que o controlador de DMA deve transferir bytes da memória para o controlador da impressora. Após receber estas informações, o controlador de DMA realiza a transferência dos bytes entre a memória e o controlador de impressora sem a intervenção da UCP. Após a transferência de todos os bytes, o controlador de DMA avisa o fim da operação através de um sinal de interrupção para a UCP.