

Gabarito AD2 - Organização de Computadores 2011.1

Data de entrega 14/05/2011

"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) 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:

Instruções com 2 operandos

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

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

```
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
```

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

```
MOV Y, B      => Y <- B
SUB Y, F      => Y <- Y - F
MOV T1, E     => T1 <- E
DIV T1, Y     => T1 <- T1 / Y
MUL T1, D     => T1 <- T1 * D
MOV Y, C      => Y <- C
SUB Y, T1     => Y <- Y - T1
ADD Y, B      => Y <- Y + B
MUL Y, B      => Y <- Y * B
MUL Y, E      => Y <- Y * E
ADD Y, A      => Y <- Y + A
```

2. (1,0) Considere um computador com instruções de um operando e endereçamento por palavras de 16 bits, possuindo o seguinte conjunto de instruções:

Cod. (hexadecimal)	Op.	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?.

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

3. (1,0) Quais são as principais características dos processadores da AMD? (pesquise no livro Introdução à Organização de Computadores, de Mário Monteiro, quinta edição)

No livro do Mário Monteiro podemos obter informações até um K7-Athlon. Deste podemos destacar as seguintes características:

Múltiplos decodificadores de instruções para traduzir instruções CISC em instruções básicas RISC.

Memória cachê L1 com 128KB, esta com a mesma velocidade do clock e memória L2 na mesma pastilha

Velocidade inicial de 500 MHz.

Capacidade de realizar multiprocessamento, capacitando para o uso em servidores

Novo projeto na área de processamento de números com ponto flutuante.

Outros processadores da AMD não incluídos no livro texto (texto retirado de fonte:)

O Athlon FX: versão dos Athlon 64 e Athlon X2 destinada para a realização de overlocks, pelo fato deste modelo vir com o multiplicador de clock destravado. O FX Foi lançado com o nome inicial de Opteron, o qual foi alterado depois de um tempo. Este modelo trabalha nos soquetes 939, 940, AM2 e o Socket F. Todos os seus modelos acompanhavam uma cache de 1MB, o dobro do Athlon 64 comum. É possível descobrir o número de núcleos utilizado através da paridade do número de identificação do modelo. Se o último dígito for par, então é dual-core, se for ímpar, é single-core. Por exemplo, o Athlon 64 FX-62 é dual-core e o Athlon 64 FX-57 single core. Essa família atinge entre 2.2 Ghz e 3.0 Ghz.

Athlon II: lançada em Junho de 2009. Sua principal característica é o uso da tecnologia multi-core, trabalhando inicialmente com dois núcleos (Athlon II X2). Já está em desenvolvimento versões com 3 e 4 cores. Essa CPU trabalha tanto nos soquetes AM2+ e AM3, assim suportando a comunicação com memórias DDR2 e DDR3. Cada um dos núcleos possuem uma cache L2 exclusiva com 1 MB de tamanho.

Sempron : nome utilizado para os processadores de baixo custo desenvolvidos pela AMD, possuindo vários modelos distintos e diversas tecnologias. Mesmo hoje, este processador trabalha somente com um núcleo. Na maioria dos seus lançamentos, esta CPU é uma versão mais simplificada e mais barata do

Athlon 64. Seus modelos foram desenvolvidos para vários sockets diferentes, incluindo o atual AM2, o que implica no fato dos Semprons atuais rodarem sobre a arquitetura AMD64. Desde o seu lançamento, em 2004, este processador trabalhou em frequências que variam desde 1.4 Ghz até 2.2 Ghz. Por ser um modelo de baixo custo, o valor de sua cache sempre foi muito baixo comparado com os demais CPUs da AMD. Por exemplo, somente as últimas versões do Sempron possuem cache L2 de 512 KB, enquanto as demais variam entre 128 e 256 KB.

Phenom : é a mais recente linha de processadores da AMD, lançado no mercado em outubro de 2007, sendo o primeiro modelo Quad-Core desenvolvido por esta companhia. Essa CPU está disponível em duas versões distintas, a Phenom X4 (codinome Agena) e Phenom X3(codinome Toliman) , que trabalham com quatro e três núcleos, respectivamente. O seu valor de clock varia entre 1.8 Ghz até 3.1 Ghz, dependendo da versão de seu lançamento. Cada core possui sua própria cache L2, sendo que existe uma cache L3 que é compartilhada por todos os núcleos ao mesmo tempo. Nesse quesito, o Phenom é o modelo mais potente da AMD, possuindo L2 com 512 KB de capacidade para cada núcleo e L3 de 2 MB. A primeira versão do Phenom (B2) possuía um bug envolvendo a tabela de páginas da memória, o que poderia causar pequenos atrasos. Normalmente, tal problema diminuía em torno de 10% o desempenho da CPU. Tal situação foi resolvida com o lançamento do Phenom B3, pouco tempo depois.

Phenom II : é a linha de processadores de alto desempenho que sucede o Phenom original, lançada em dezembro de 2008. Suporta ambos sockets AM2+ e AM3, assim, trabalhando com tanto memórias DDR2 e DDR3. Seu valor de clock também aumentou, trabalhando entre 2.5 e 3.2 Ghz. Este modelo está disponível em versões com 2, 3 e 4 núcleos.

4. (1,0) Descreva:

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

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.

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.

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 acessos à MP para execução, requer pelo menos 2 acessos à memória principal.

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.

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

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.

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 da 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.

5. (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 às 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 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

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.

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

A= -1;

B=-7;

C=+1;

D=+2

6.1. Considere uma máquina que utiliza 4 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 = 1001

B = 1111

C = 0001

D = 0010

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

E=A+B;

F=A-D;

G=C+D;

H=C-B;

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

$$E = A + B \Rightarrow 1001 + 1111 = 1000 \Rightarrow 0_{10} \text{ (esperado -8)}$$

$$F = A - D \Rightarrow 1001 - 0010 = (\text{obter representação negativa de D})$$

$$= 1001 + 1010 = 1011 \Rightarrow -3_{10} \text{ (esperado -3)}$$

$$G = C + D \Rightarrow 0001 + 0010 = 0011 \Rightarrow +3_{10} \text{ (esperado +3)}$$

$$H = C - B \Rightarrow 0001 - 1111 = (\text{obter representação negativa de B})$$

$$= 0001 + 0111 = 0000 \Rightarrow +0_{10} \text{ (esperado +8)}$$

6.2. Considere uma máquina que utiliza 4 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.

$$A = 1111$$

$$B = 1001$$

$$C = 0001$$

$$D = 0010$$

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

$$E = A + B;$$

$$F = A - D;$$

$$G = C + D;$$

$$H = C - B;$$

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

$$E = A + B \Rightarrow 1111 + 1001 = 1000 \Rightarrow -8_{10} \text{ (esperado -8)}$$

$$F = A - D \Rightarrow 1111 - 0010 = \text{(obter representação negativa de D)}$$

$$= 1111 + 1110 = 1101 \Rightarrow -3_{10} \text{ (esperado -3)}$$

$$G = C + D \Rightarrow 0001 + 0010 = 0011 \Rightarrow +3_{10} \text{ (esperado +3)}$$

$$H = C - B \Rightarrow 0001 - 1001 = \text{(obter representação negativa de B)}$$

$$= 0001 + 0111 = 1000 \Rightarrow -8_{10} \text{ (esperado +8)}$$

7. (1,0) Considere um computador, cuja representação para ponto fixo e para ponto flutuante utilize 20 bits.

7.1. (0,3) Considere o seguinte conjunto de bits representado em hexadecimal CF0B0. Indique o valor deste número **em decimal**, considerando-se que o conjunto representa:

$$(CF0B0)_{16} = (1100\ 1111\ 0000\ 1011\ 0000)_2$$

7.1.1. um inteiro sem sinal

$$2^{19} + 2^{18} + 2^{15} + 2^{14} + 2^{13} + 2^{12} + 2^7 + 2^5 + 2^4 = +848.048$$

7.1.2. um inteiro em sinal magnitude

$$-(2^{18} + 2^{15} + 2^{14} + 2^{13} + 2^{12} + 2^7 + 2^5 + 2^4) = -323.760$$

7.1.3. um inteiro em complemento a 2

$$-2^{19} + (2^{18} + 2^{15} + 2^{14} + 2^{13} + 2^{12} + 2^7 + 2^5 + 2^4) = -200.528$$

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 7 bits representam o expoente e os 12 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 $\pm(1.b_1b_2b_3b_4b_5b_6b_7b_8b_9b_{10}b_{11}b_{12} \times 2^{\text{Expoente}})$, onde o bit mais à esquerda representa o sinal (0 para números positivos e 1 para números negativos), os próximos 7 bits representam o expoente em excesso e os 12 bits seguintes representam os bits b_1 a b_{12} , como mostrado na figura a seguir:

S	Expoente representado em excesso	$b_1\ b_2\ b_3\ b_4\ b_5\ b_6\ b_7\ b_8\ b_9\ b_{10}\ b_{11}\ b_{12}$
1	7	12

- 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 da palavra. Como o expoente tem $n=7$ bits. Então, $2^{n-1} - 1 = 63$.

- 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.

1100 11110000 10110000

Sendo:

Sinal = 1 => negativo

Expoente = 1001111 = 79 - 63 => expoente = 16

Mantissa = 000010110000

*Temos então => -1, 000010110000 x 2¹⁶
- 10000101100000000 = -68352*

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

7.2.3.1. +24,25

Convertendo para binário = 11000,01 => normalizando 1,100001 x 2⁴

Temos então:

Sinal = 0 (positivo)

Expoente = (4 + 63 = 67) = 1000011

Mantissa = , 100001

Resultado: 0 1000011 100001000000

7.2.3.2. -0,05

Convertendo para binário = 0,0000110011001100110011 =>

normalizando 1,10011001100110011 x 2⁻⁵

Temos então:

Sinal = 1 (negativo)

Expoente = (-5 + 63 = 58) = 0111010

Mantissa = , 10011001100110011

Resultado: 1 0111010 100110011001

- 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**.

Maior número positivo: 0 1111110 11111111111 = + 1,84449 x 10⁺¹⁹

Menor número positivo: 0 0000001 00000000000 = + 2,168404 x 10⁻¹⁹

8. (0,5) Explique como é realizada a transferência de dados e a arbitração em um barramento PCI (sugestões de fonte de consulta: livro do Stallings e o site <http://computer.howstuffworks.com>. Na sua resposta indique as suas fontes de consulta).

Textos retirados de:

Livro texto da disciplina do Willian Stallings

Fonte complementar: Notas de aula Disciplina Arquitetura e Organização Interna de Computadores
Universidade do Minho - PT

A interface do PCI é composta por um conjunto de 62 pinos na especificação de 32 bits e por 94 pinos na versão de 64 bits, porém muitos destes sinais não precisam ser tratados por todos os dispositivos, assim uma implementação mínima de um dispositivo PCI é composta de apenas 47 pinos caso o dispositivo nunca se torne um master do barramento ou 49 pinos caso este dispositivo seja projetado para se tornar o mestre do barramento. Esta pinagem reduzida é uma das grandes vantagens do PCI.

Os dispositivos conectados ao barramento podem assumir a posição de mestre, conhecido como iniciador (initiator) ou escravo, conhecido como alvo (target). Abaixo segue uma descrição das vias de um barramento PCI.

AD0-AD31 – Nestas vias do barramento os dados e os endereços são multiplexados, assim no início de uma transferência, este barramento indica o endereço, e na fase seguinte, os dados.

C/BE0-C/BE3 – Estes sinais indicam qual o comando que será executado (leitura, escrita, etc.), essa informação é enviada durante a fase de endereçamento. Já durante a fase de dados cada bit desses quatro sinais indica quais dos quatro bytes do barramento estão ativos, assim é possível acessar os bytes de dados de forma individual.

FRAME – Este sinal é ativado pelo Mestre do Barramento para dar início a um ciclo de transferência.

IRDY - Initiator Ready. Indica que o Mestre está pronto para ler ou enviar dados. Quando este sinal não é ativado, o escravo irá esperar tantos wait states quanto forem necessários.

TRDY - Target Ready. Indica que o Escravo está pronto para receber dados (escrita) ou que o dado lido já está disponível (leitura). Quando este sinal não é ativado, o Mestre irá gerar tantos wait states quantos forem necessários.

DEVSEL - Ativado pelo Escravo quando reconhece o seu endereço. Desta forma o Mestre pode saber se o dispositivo Escravo está ativo ou presente no barramento.

REQ - Requisição enviada ao Bus Arbitrer, para que o dispositivo se torne Mestre do barramento. Cada dispositivo tem seu próprio sinal REQ.

GNT - Grant. Através deste sinal o Bus Arbitrer indica ao dispositivo solicitante que o barramento está liberado, permitindo assim que se torne Bus Master. Cada dispositivo tem seu próprio sinal GNT.

INTA, INTB, INTC, INTD - São linhas de interrupção a serem usadas pelos dispositivos PCI. Cada dispositivo e cada slot é ligado a um desses sinais, que podem ser compartilhados, ou seja, uma mesma linha INT pode ser usada por mais de um slot. O padrão PCI prevê o compartilhamento de interrupções.

AD32-AD63 - Continuação do barramento de dados e endereços nos slots PCI de 64 bits.

C/BE4-C/BE7 - Continuação do barramento de comando e habilitação de bytes nos slots PCI de 64 bits.

REQ64 - Requisição de transferência de 64 bits.

ACK64 - Indica que o Escravo está apto a realizar transferência de 64.

CLK - Clock este é o trem de pulso para todas as transferências do PCI.

RST – Reset é o sinal que instrui o dispositivo a assumir um estado inicial, reiniciando todos os registradores ou qualquer componente de estado para o estado padrão, este sinal normalmente é enviado para o PCI quando o computador inicia.

PERR - Parity Error é utilizado para reportar erros de paridade na transmissão dos dados.

SERR - System Error indica erros do sistema, como erro de paridade no endereço ou qualquer outro tipo de erro fatal.

Transferência de Dados

Uma transferência de dados consiste em uma fase de endereçamento seguido de uma ou mais fases de dados. Uma operação de I/O padrão tem somente uma fase de endereçamento e uma fase de dados. Em operações de transferências com a memória, quando se transfere blocos de dados, a operação consiste em uma fase de endereçamento seguida de múltiplas fases de dados escrevendo/lendo blocos contíguos de uma única vez. Tanto o alvo como o iniciador podem finalizar uma sequência de transferência a qualquer momento, o dispositivo alvo faz isso desabilitando o sinal FRAME após enviar o último dado solicitado, já o iniciador ou master pode ativar o sinal STOP que sinaliza para o alvo o final da transferência.

Todos os eventos são sincronizados pela transição descendente do relógio que ocorre no meio de cada ciclo de relógio.

Texto abaixo retirado da fonte complementar já citada

Os eventos mais significativos

1. Logo que um mestre no barramento assume o controle, pode iniciar a transação ativando a linha *FRAME*. Esta linha mantém a imposição até que o iniciador esteja pronto para completar a última fase de dados. O iniciador coloca também o endereço inicial no barramento de endereços e o comando de leitura nas linhas *C/BE*.
2. No início do segundo ciclo o dispositivo alvo irá reconhecer o seu endereço nas linhas *AD*.
3. O iniciador cessa de alimentar o barramento de endereços. O iniciador muda a informação nas linhas de *C/BE* para designar quais são as linhas de *AD* a serem usadas na transferência para os dados correntemente endereçados (de 1 a 4 octetos). O iniciador, também, ativa a linha *IRDY* para indicar que está pronto para o primeiro item de dados.
4. O dispositivo selecionado ativa a linha *DEVSEL* para indicar que reconheceu o seu endereço e vai responder. Coloca os dados solicitados nas linhas *AD* e ativa a linha *TRDY* para indicar que estão presentes dados válidos no barramento.
5. O iniciador lê o dado no início do ciclo 4 e muda, conforme o necessário, as linhas de habilitação de octetos, em preparação para a leitura seguinte.
6. Neste exemplo, o alvo necessita de algum tempo para preparar o segundo bloco de dados para transmissão. Consequentemente, desativa o sinal *TRDY* para sinalizar o iniciador que não irá haver dados no ciclo seguinte. Em conformidade, o iniciador não lê as linhas de dados no início do quinto ciclo de relógio e não muda a habilitação de octeto durante aquele ciclo. O bloco de dados é lido no início do ciclo 6.
7. Durante o ciclo 6, o alvo coloca o terceiro item de dados no barramento. Contudo, no exemplo, o iniciador não está ainda pronto para ler o item de dados (e.g. há uma condição de *buffer* temporário cheio). Por isso desativa a linha *IRDY*. Isto faz com que o alvo possa manter o terceiro item no barramento durante um ciclo extra de relógio.
8. O iniciador sabe que a terceira transferência de dados é a última. Ativa a linha *IRDY* para sinalizar que está pronto para completar a transferência.
9. O iniciador desativa a linha *IRDY*, regressando o barramento ao estado de espera e o alvo desativa as linhas *TRDY* e *DEVSEL*.

Arbitragem do Barramento:

Texto base retirado de fonte complementar já citada.

O PCI recorre a um esquema de arbitragem centralizada e síncrona no qual cada mestre tem um sinal único para requerimento e para concessão. Estas linhas de sinais estão ligadas a um árbitro central e um simples esquema requerimento-concessão é usado para obter o acesso ao barramento.

A especificação PCI não impõe um algoritmo particular de arbitragem. O árbitro pode usar uma abordagem “primeiro a chegar primeiro a ser servido”, uma abordagem “andar à volta” ou qualquer outra espécie de esquema de prioridade.

Considere 2 dispositivos (A e B) que requerem a posse do barramento.

1. Em algum ponto anterior ao início do ciclo 1, A ativa o sinal *REQ*. O árbitro faz a amostragem do sinal no início do ciclo 1 de relógio.
2. Durante o ciclo 1, B requisita o uso do barramento ativando o seu sinal *REQ*.
3. Ao mesmo tempo, o árbitro ativa o sinal *GNT-A* para garantir a A o acesso ao barramento.

4. O mestre A faz a amostragem da linha GNT-A no início de ciclo 2 e toma conhecimento que lhe foi concedido o acesso ao barramento. Descobre, também, que IRDY e TRDY estão desativados, indicando que o barramento está livre. Em concordância, ativa a linha FRAME e coloca a informação de endereço no barramento de endereços e o comando no barramento C/BE (não visível). Mantém REQ-A, porque tem uma segunda transação a efetuar a seguir a esta.
5. O árbitro de barramento faz a amostragem de todas as linhas GNT no início do ciclo 3 e toma uma decisão de conceder o barramento a B na próxima transação. Ativa o sinal GNT-B e desativa o sinal GNT-A. B não irá poder usar o barramento até que este retorne ao estado de ócio.
6. A linha FRAME desativada indica que a última e (única) transferência de dados está em progresso. Os dados são colocados no barramento e o alvo [e avisado ativando-se a linha IRDY. O alvo lê os dados no início do próximo ciclo de relógio.
7. No início do ciclo 5, B descobre que IRDY e FRAME estão desativados e consequentemente pode tomar o controle do barramento ativando a linha FRAME. Também desativa a linha de REQ porque apenas pretende efetuar uma transação.

9. (0,5) Explique o funcionamento de um monitor de vídeo que utiliza a tecnologia OLED (Organic Light-Emitting Diode) (indique as suas fontes de consulta).

Texto base de pesquisa retirado de <http://electronics.howstuffworks.com> acessado em 02/05/2011

1) Conceitos e composição:

O OLED é um dispositivo semicondutor do estado sólido com cerca de 100 a 500 nanômetros, cerca de 200 vezes menor do que um fio de cabelo humano. Os OLEDs podem ter cerca de duas a três camadas de material orgânico. Em um projeto mais recente, a terceira camada ajuda no transporte dos elétrons do catodo para a camada de emissão.

Um OLED consiste das seguintes partes, ver figura 1

- **Substrate** – que pode ser de plástico transparente, ou vidro – Serve de base de suporte para o OLED.
- **Anode** – material transparente – O anodo remove eletrons quando a corrente elétrica flui através do dispositivo.
- **Organic layers** – São camadas feitas de moléculas orgânicas ou polímeros.
 - **Conducting layer** – Esta camada é feita de molécula plástica orgânica que transporta os “holes” do anodo. O polímero condutor utilizado no OLED é o polyaniline.
 - **Emissive layer** – Esta camada é feita de moléculas orgânicas plásticas. Estas diferem das anteriores por produzir a luz. O polímero usado na camada emissora é o polyfluorene.
- **Cathode** - Este pode ou não ser transparente dependendo do tipo de OLED – O catodo injeta elétrons quando uma corrente atravessa dispositivo

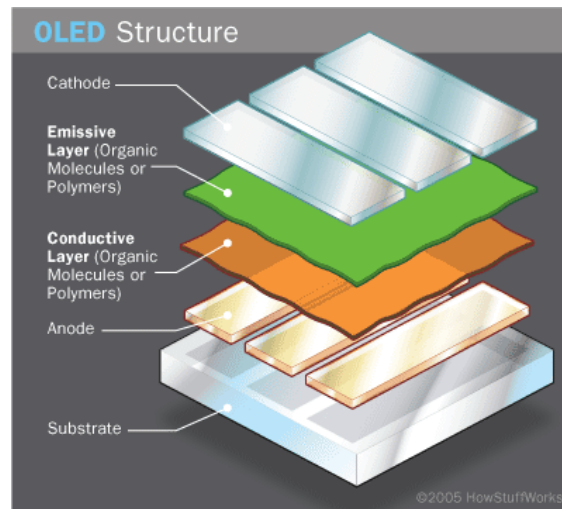


Figura 1: composição do OLED
(fonte: <http://electronics.howstuffworks.com>)

2) Vantagens e desvantagens:

Vantagens do OLED sobre o LED e o LCD.

- O plástico, as camadas orgânicas do OLED são mais finas, mais leves e mais flexíveis do que as camadas cristalinas do LED ou LCD.
- Os substratos do OLED podem ser de plástico ao invés do vidro usado nos LEDs e LCDs. Os OLEDs são mais brilhantes do que os LEDs.
- Os LEDs e os LCDs precisam do vidro como suporte e o vidro absorve alguma luz. Os OLEDs não precisam de vidro. Os OLEDs não necessitam de luz de fundo como os LCDs
- Os LCDs funcionam através do bloqueio seletivo das áreas da luz de fundo para fazer as imagens que você vê, enquanto os OLEDs geram a própria luz. Como os OLEDs não precisam de retroiluminação, eles consomem muito menos energia do que os LCDs (a maior parte da energia do LCD vai para a iluminação). Isto é especialmente importante para dispositivos alimentados por bateria, como telefones celulares.
- Os OLEDs são mais fáceis de produzir e podem ser fabricados em tamanhos maiores, como os OLEDs são essencialmente plásticos, podem ser fabricados em folhas grandes, finas.
- Os OLEDs possuem grandes campos de visualização, aproximadamente 170 graus.
- Os OLEDs produzem sua própria luz, para que tenham um alcance maior de visualização.

Desvantagens do OLED sobre o LED e o LCD.

- Vida útil: Os filmes verdes e vermelhos têm vida útil mais longa (46.000 a 230.000 horas), já os orgânicos azuis apresentam atualmente uma vida útil mais curta (até cerca de 14 mil horas)
- Fabricação - Os processos de fabricação são caros atualmente.
- Água - A água pode facilmente danificar os OLEDs

3) Tipos de OLEDs.

- **OLED de matriz passiva (PMOLED):** têm tiras de catodo, camadas orgânicas e tiras de anodo. As tiras de anodo são arranjadas perpendicularmente às tiras de catodo. As interseções do catodo com o anodo formam os pixels onde a luz é emitida. O circuito elétrico externo aplica uma corrente às tiras selecionadas do anodo e do catodo, determinando quais pixels serão ligados e quais permanecerão desligados pixels. Novamente, o brilho de cada pixel é proporcional à quantidade de corrente aplicada. São mais eficientes para textos e ícones e mais adequados para telas pequenas (2 - 3 polegadas de diagonal), tais como encontrados nos telefones celulares, PDAs e MP3 players
- **OLED de matriz ativa (AMOLED):** têm camadas completas de catodo, moléculas orgânicas e anodo, mas a camada de anodo se sobrepõe a um transistor de película fina (TFT) que forma uma matriz. A própria estrutura TFT é o circuito elétrico que determina quais pixels ficam ligados para formar uma imagem. AMOLEDs são eficientes para grandes displays. AMOLEDs também têm taxas de atualização mais rápidas, adequados para vídeo. Os melhores usos para AMOLEDs são monitores de computador, TVs de tela grande e avisos eletrônicos ou painéis.
- **OLEDs transparentes** têm apenas componentes transparentes (substrato, catodo e anodo) e, quando desligados, são até 85 por cento tão transparentes quanto seu substrato. Quando um display OLED transparente é ligado, ele permite que a luz passe nas duas direções. O display OLED transparente pode ser ativo ou passiva-matriz. Esta tecnologia pode ser usada para displays.
- **OLED de emissão superior** têm um substrato que pode ser opaco ou refletivo. Eles são mais adequados para projetos com matrizes ativas. Fabricantes podem usar os displays OLED de emissão de cartões inteligentes (smart cards)
- **OLED dobrável** têm substratos feitos de lâminas metálicas muito flexíveis ou de plástico. Os OLEDs dobráveis são muito leves e duráveis. Seu uso em dispositivos como telefones celulares e PDAs, pode reduzir as quebras, a maior causa das devoluções ou reparos. Potencialmente, os displays OLED dobráveis podem ser unidos para criar tecidos para roupas "inteligentes", tais como roupas de sobrevivência externa com chip de computador integrado, telefone celular, receptor GPS e display OLED costurado a ela.
- **Os OLEDs brancos** emitem uma luz branca brilhante que é uma energia mais uniforme e mais eficiente do que aquela emitida pelas lâmpadas fluorescentes. Os OLEDs brancos também têm a qualidade das cores reais das lâmpadas incandescentes. Como os OLEDs podem ser feitos em folhas grandes, eles podem substituir as lâmpadas fluorescentes que são usadas atualmente em casas e edifícios. Seu uso poderá reduzir potencialmente os custos de energia para iluminação.
- Monitores OLEDs

procedimento é repetido para receber cada byte referente a uma tecla pressionada pelo usuário.

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 de teclado, o endereço inicial da memória onde devem ser armazenados os bytes referentes às teclas pressionadas, o número de bytes a serem recebidos e a indicação de que o controlador de DMA deve transferir bytes do controlador de teclado para a memória. Após receber estas informações, o controlador de DMA realiza a transferência dos bytes entre controlador de teclado e memória sem a intervenção da UCP. Após a transferência de todos os bytes, o controlador de DMA avisa o fim da operação para a UCP através de um sinal de interrupção.