

AD2 - Organização de Computadores 2006.2

GABARITO

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:

Resposta:

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

OBS: como os endereços são utilizados mais de uma vez recomenda-se manter os conteúdos destes endereços.

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

Resposta:

```
MOV T1,C    =>    T1 ← C
SUB T1,A    =>    T1 ← T1 - A
MOV T2,E    =>    T1 ← E
DIV T2,B    =>    T2 ← T2 / B
MOV T3,D    =>    T3 ← D
SUB T3,T2   =>    T3 ← T3 - T2
MUL T1,B    =>    T1 ← T1 * B
MUL T3,D    =>    T3 ← T3 * D
ADD T1,T3   =>    T1 ← T1 + T3
ADD T1,A    =>    T1 ← T1 + A
MOV X,T1    =>    X ← T1
```

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

Resposta:

```
MOV T1,B    =>    T1 ← B
SUB T1,F    =>    T1 ← T1 - F
MOV T2,E    =>    T2 ← E
DIV T2,T1   =>    T2 ← T2 / T1
MUL T2,D    =>    T2 ← T2 * D
MOV T3,C    =>    T3 ← C
SUB T3,T2   =>    T3 ← T3 - T2
ADD T3,B    =>    T3 ← T3 + B
MUL T3,B    =>    T3 ← T3 * B
MUL T3,E    =>    T3 ← T3 * E
ADD T3,A    =>    T3 ← T3 + A
MOV X,T3    =>    X ← T3
```

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. 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.

Resposta:

End	Conteúdo	End.	Conteúdo
1AF	E1C0	GET (1C0)	Ler dado para (1C0)
1B0	E1C1	GET (1C1)	Ler dado para (1C1)
1B1	31C1	LDA (1C1)	$ACC \leftarrow (1C1)$
1B2	11C0	ADD (1C0)	$ACC \leftarrow ACC + (1C0)$
1B3	41C1	STA (1C1)	$(1C1) \leftarrow ACC$
1B4	D1BA	JN (1BA)	Se $ACC < 0$, então $CI \leftarrow (1BA)$
1B5	E1C2	GET (1C2)	Ler dado para (1C2)
1B6	31C1	LDA (1C1)	$ACC \leftarrow (1C1)$
1B7	21C2	SUB (1C2)	$ACC \leftarrow ACC - (1C2)$
1B8	41C1	STA (1C1)	$(1C1) \leftarrow ACC$
1B9	A1BE	JMP (1BE)	$CI \leftarrow (1BE)$
1BA	E1C3	GET (1C3)	Ler dado para (1C3)
1BB	31C1	LDA (1C1)	$ACC \leftarrow (1C1)$
1BC	11C3	ADD (1C3)	$ACC \leftarrow ACC + (1C3)$
1BD	41C1	STA (1C1)	$(1C1) \leftarrow ACC$
1BE	F1C1	PRT (1C1)	Imprimir (1C1)
1BF	0000	END	Fim da execução
1C0	31*5		DADOS
1C1	61C4		DADOS
1C2	21C0		DADOS
1C3	11C4		DADOS

- a) (0,5) Qual o valor em hexadecimal de CI, RI e ACC ao final da execução de cada instrução?.

Resposta:

End	Conteúdo	End.	Conteúdo	CI	RI	ACC	
						hexa	decimal
				1AF	20A3	153C	5436
1AF	E1C0	GET (1C0)	Ler dado para (1C0)	1B0	E1C0	153C	5436
1B0	E1C1	GET (1C1)	Ler dado para (1C1)	1B1	E1C1	153C	5436
1B1	31C1	LDA (1C1)	ACC ← (1C1)	1B2	31C1	25	37
1B2	11C0	ADD (1C1)	ACC ← ACC+(1C0)	1B3	11C0	38	56
1B3	41C1	STA (1C1)	(1C1) ← ACC	1B4	41C1	38	56
1B4	D1BA	JN (1BA)	Se ACC<0, então CI← (1BA)	1B5	D1BA	38	56
1B5	E1C2	GET (1C2)	Ler dado para (1C2)	1B6	E1C2	38	56
1B6	31C1	LDA (1C1)	ACC ← (1C1)	1B7	31C1	38	56
1B7	21C2	SUB (1C2)	ACC ← ACC -(1C2)	1B8	21C2	28	43
1B8	41C1	STA (1C1)	(1C1) ←ACC	1B9	41C1	28	43
1B9	A1BE	JMP (1BE)	CI ← (1BE)	1BE	A1BE	28	43
1BE	F1C1	PRT (1C1)	Imprimir (1C1)	1BF	F1C1	28	43
1BF	0000	END	Fim da execução	*	0000	28	43

* Retorno para o endereço que chamou a execução da rotina, endereço este contido na pilha do sistema operacional

- b) (0,5) Quais os valores impressos em decimal?

Resposta:

Será impresso: 43

- c) (0,0) O que aconteceria com o programa se o conteúdo da posição de endereço 1B3 fosse alterado para 81C1?

Resposta:

Não tem esta instrução na relação (8), estou cancelando esta questão c) .

3. (1,0) Quais são as principais características do processador PowerPC? (pesquise nos livros: Introdução à Organização de Computadores, de Mário Monteiro e Arquitetura e Organização de Computadores, de William Stallings)

Resposta:

O PowerPC foi construído a partir de 1991, com um projeto conjunto entre a IBM, Apple e Motorola. Com uso destacado nas estações de trabalho RS/6000, e em alguns modelos especiais para Macintosh, da Apple. Iniciado a partir da implementação do PowerPC 601, o PowerPC620 foi a primeira implementação totalmente 64 bits (barramento e registradores) desta família.

Seguem as principais características dos principais membros da família PowerPC:

601: Processador de 32 bits / de 50 a 120MHz. Composto de uma única pastilha, com cerca de 2,8 milhões de transistores, e quase totalidade de suas instruções programadas diretamente no hardware. É um processador RISC com cerca de 184 instruções de 32 bits. Uma maior quantidade de registradores: 32 para uso geral e 32 para ponto flutuante, ambos de 64bits

O PowerPC 601 é dividido em unidades funcionais independentes e organizadas como pipeline: uma unidade de ponto fixo, uma unidade de ponto flutuante e uma unidade de processamento de desvio, possibilitando a execução de 3 instruções simultaneamente, resultando em um projeto superescalar de grau 3.

A unidade de busca pode obter antecipadamente da cache até oito instruções ao mesmo tempo. A unidade de cache provê uma cache combinada de instruções e dados e é responsável por suprir instruções para outras unidades e por suprir dados para os registradores. A lógica de arbitração da cache envia para a cache o endereço de acesso de maior prioridade

603: Voltado para microcomputadores e portáteis / de 32 bits / menor custo / 100 a 300 MHz / cache L1 de 16Kb para instruções e 16Kb para dados / 1,6 a 2,6 milhões de transistores.

604: Voltado para microcomputadores e servidores de menor desempenho / de 32 bits / 166 a 300 MHz / cache L1 de 16Kb para instruções e 16Kb para dados / de 3,6 a 5,1 milhões de transistores

620: Voltado para máquinas servidoras de alto desempenho / registradores e barramento de 64 bits / 200 a 366 MHz / cache L1 de 32Kb para instruções e 32Kb para dados / cache L2 de 256Kb a 1 Mb / 6,3 milhões de transistores. O PowerPC 620 é a primeira implementação de 64 bits da arquitetura PowerPC, com registradores de uso geral de também 64bits. A partir desta versão são incluídas seis unidades de execução independentes: uma unidade de instrução, três unidades de ponto fixo, uma unidade de carga/armazenamento, uma unidade de ponto flutuante. Desta forma o processador pode despachar até quatro instruções simultaneamente para as três unidade de números inteiros e para a unidade de ponto flutuante.

O Powerpc 620 emprega uma estratégia de previsão de desvio de alto desempenho, que envolve uma lógica de previsão, renomeação de registradores e estações de reserva dentro das unidades de execução. O Powerpc 620 pode executar de forma especulativa até quatro instruções de desvio não resolvidas. A previsão de desvio é baseada no uso de uma tabela de histórico de desvio de 2048 entradas com sucesso de previsão de desvio na ordem de 90%.

740/750 (processador G3): integra dois níveis de memória cache na pastilha do processador principal, o que resulta em uma melhora de desempenho significativa em relação a máquinas com memória cache fora da pastilha

G4 : processador fornece maior paralelismo e velocidade interna da pastilha do processador / 500 MHz / cache L1 de 32Kb para instruções e 32Kb para dados / cache L2 de 256Kb a 1 Mb

4. (1,0) Descreva:

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

Resposta:

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 acesso 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.

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.

5. (1,0) Explique, comparando:

- a) Computadores vetoriais e Computadores matriciais

Resposta:

Muitos problemas da física e engenharia envolvem operações sobre vetores. Frequentemente os mesmos cálculos são executados em muitos conjuntos diferentes de dados ao mesmo tempo. Neste contexto,

existem dois métodos que têm sido usados para executar grandes problemas científicos rapidamente: Os Computadores vetoriais que correspondem a sistemas compostos por processadores vetoriais e freqüentemente são associados à organizações de ULAs com pipeline de operações; e os Computadores matriciais que 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.

6. (1,8) Considere um computador, cuja representação para ponto fixo e para ponto flutuante utilize 32 bits. Na representação para ponto flutuante, utiliza-se o padrão IEEE 754 para precisão simples.

- a) (0,2) Indique como será a representação do inteiro sem sinal 2.156.019.713.

Resposta:

10000000 10000010 01000000 00000001

- b) (0,6) Considerando o padrão de bits obtido no item a, indique o valor deste número em decimal, se considerarmos que o padrão representa um inteiro utilizando-se representação sinal magnitude, um inteiro em complemento a 2 e um real em ponto flutuante.

Resposta:

Sinal magnitude: $-(2^{23}+2^{17}+2^{14}+2^0) = -8536065$

Complemento a 2: $-2^{31}+2^{23}+2^{17}+2^{14}+2^0 = -2138947583$

Ponto flutuante: Sinal = 1, número negativo, representação do expoente em excesso de 127 = 00000001, logo Expoente+127=1, Expoente=-126. Mantissa frac. = 000001001000000000000001.

Número na base 2 = $-(1,000001001000000000000001)_2 \times 2^{-126}$

Número na base 10 = $-(2^{-126}+2^{-132}+2^{-135}+2^{-149}) \cong 1,2 \times 10^{-38}$

- c) (0,4) Qual será a representação em ponto flutuante dos seguintes valores decimais neste computador:

c.1) +13,75

Resposta:

$1101,11 = 1,10111 \times 2^{+3}$

Sinal = 0, Mantissa = 101110000000000000000000, Representação do expoente = $3+127=130=10000010$

Representação: 01000001 01011100 00000000 00000000

c.2) -2,2

Resposta:

$-10,0011001100110011001100 = 1,0001100110011001100 \times 2^{+1}$

Sinal = 1, Mantissa = 00011001100110011001100, Representação do expoente = $1+127=128=10000000$

Representação: 11000000 0001100 11001100 11001100

- d) (0,2) Mostre a representação dos números dos itens da questão acima, caso se utilizasse a representação sinal e magnitude para representar o expoente.

d.1) +13,75

Resposta:

$1101,11 = 1,10111 \times 2^{+3}$

Sinal = 0, Mantissa = 101110000000000000000000, Representação do expoente = 00000011

Representação: 00000001 11011100 00000000 00000000

d.2) -2,2

Resposta:

$-10,0011001100110011001100 = 1,0001100110011001100 \times 2^{+1}$

Sinal = 1, Mantissa = 00011001100110011001100, Representação do expoente = 00000001

Representação: 10000000 10001100 11001100 11001100

- e) (0,4) Supondo que se utilize a representação sinal e magnitude para o expoente ao invés da representação em excesso, indique o menor e o maior valor positivo normalizado na representação em ponto flutuante para este computador. Considere, neste caso, que as representações para o maior e o menor expoente são utilizadas para casos especiais e não para números normalizados.

Resposta:

$$\text{Menor} \quad 01111111 \ 00000000 \ 00000000 \ 00000000 \\ + (1,000000000000000000000000)_2 \times 2^{-126} = 1,2 \times 10^{-38}$$

$$\text{Maior} \quad 00111111 \ 01111111 \ 11111111 \ 11111111 = \\ + (1,111111111111111111111111)_2 \times 2^{+126} = 3,4 \times 10^{+38}$$

7. (0,8) Atualmente, um dispositivo bastante utilizado para armazenar dados é o *pen-drive*. Este dispositivo é manufaturado com memória flash e se conecta ao computador através de uma conexão USB. Faça uma pesquisa sobre memória flash e barramento USB e explique como eles funcionam. Indique as referências que você utilizou. As fontes de pesquisa indicadas são: os livros Introdução à Organização de Computadores, de Mário Monteiro e Arquitetura e Organização de Computadores, de William Stallings e os sites <http://www.infowester.com/cartoesflash.php> e <http://www.apple.com/br/usb/>.

Resposta:

- a) Sobre a memória Flash (Fonte: Arquitetura e Organização de Computadores, de William Stallings e os sites <http://www.infowester.com/cartoesflash.php>)

Os chips de memória Flash possuem características semelhantes à memória EEPROM (Electrically Erasable Programmable Read Only Memory) usada nos computadores, sendo sua principal característica a não volatilidade, isto é, os dados armazenados têm seus valores mantidos mesmo quando não há mais fornecimento de energia.

Este tipo de memória tem vantagens como o baixo consumo de energia, pequeno espaço físico ocupado e é bastante durável. O grande problema da memória Flash é seu preço elevado, mas com o aumento da comercialização os preços vêm se tornando mais acessíveis para uma relativa grande quantidade de memória.

A tecnologia Flash faz uso de semicondutores, ou seja, não tem peças móveis, o que evita problemas de causa mecânica. Juntando isso a recursos de proteção, como ECC, a memória Flash se mostra muito confiável.

Podem ser de 2 tipos: **NOR** que permite acesso às células de memória de maneira aleatória, mas com alta velocidade, tem seu uso comum em BIOS, telefones celulares e placas de redes especiais e a **NAND** que também trabalha em alta velocidade, porém faz acesso sequencial às células de memória e trata-as em conjunto.

Existe uma grande variedade de cartões que utilizam esta tecnologia, sendo que ainda não foi definido um padrão para fabricação destes cartões.

- b) Sobre a tecnologia USB (Fonte: <http://www.howstuffworks.com/usb.htm> e <http://www.microsoft.com/technet/prodtechnol/wce/support/usbce.mspx>).

USB (Universal Serial Bus) é uma arquitetura de barramento utilizada para conectar dispositivos periféricos em um computador hospedeiro. Ela possui um protocolo de comunicação que suporta a transferência de dados em forma serial entre o sistema hospedeiro e os componentes periféricos. Atualmente, vários periféricos possuem uma conexão USB, facilitando a ligação dos dispositivos de entrada e saída ao sistema hospedeiro, pois a utilização de uma única maneira para conexão simplifica o projeto de hardware e de instalação, já que não são necessárias interfaces diferentes para conectar periféricos diferentes. Entre os periféricos que possuem interface USB, podemos citar: impressoras, scanners, mouse, joystick, câmeras digitais, webcams, modems, alto falantes, telefones, vídeo fones, Zip drives, conexões de rede, cartões Flash.

O padrão USB prevê suporte para ligação de até 127 dispositivos que são interligados utilizando-se concentradores de ligações denominados hubs. Ele também prevê fornecimento de energia para dispositivos que não precisem de muita potência (500mA para 5Volts). Dispositivos de maior potência, como impressoras, devem ter um auto suprimento de energia.

O padrão inicial para USB 1.1 definiu uma taxa de transmissão máxima igual a 12 Megabits/s. Em 2000, foi definido o padrão USB 2.0, cuja taxa de transferência máxima é 480 Mbits/s.

Os dispositivos USB são do tipo hot-swappable, ou seja, eles podem ser conectados e desconectados do barramento USB em qualquer momento, sem que seja necessário desligar o sistema de computação. Além disso, existe um suporte Plug-and-Play de tal maneira que o computador hospedeiro consegue identificar o periférico USB quando ele é conectado ao sistema e o driver apropriado ao dispositivo é automaticamente carregado.

O protocolo USB suporta quatro tipos de transferência entre o computador hospedeiro e os periféricos:

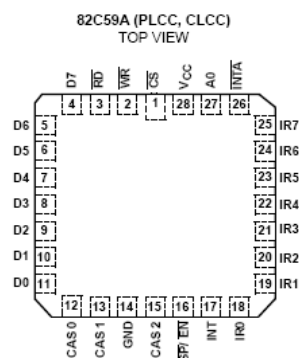
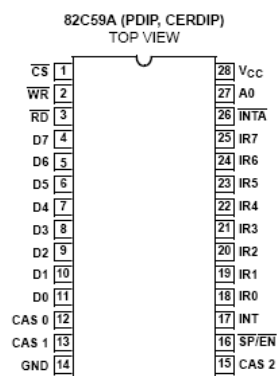
- controle: utilizada para permitir o controle e gerenciamento dos periféricos pelo sistema hospedeiro.
- isócrona: utilizada para realizar transferências que possuem restrições de tempo e tolerância a falhas ou necessitam de taxas constantes de transferência de dados.
- interrupção: utilizada para verificar se o dispositivo possui dado a ser transmitido. A taxa de verificação pode ser variável dependendo da necessidade do dispositivo.
- bloco: utilizada para realizar transferência de grande quantidade de dados que devem ter seu envio e recebimento garantido, mas que não necessitam de requerimentos específicos para os valores de taxa de transmissão e de latência.

8. (0,8) Um componente muito utilizado para controlar interrupções é o controlador de interrupções INTEL 82C59A. Faça uma pesquisa e forneça as seguintes informações sobre ele: descrição dos pinos, modo de funcionamento do controlador e modo de programação do controlador pela UCP. Indique as referências que você utilizou. As fontes de pesquisa indicadas são: o livro Arquitetura e Organização de Computadores, de William Stallings e o site <http://www.intersil.com/data/fn/fn2784.pdf>.

Resposta: (Fonte: livro Arquitetura e Organização de Computadores, de William Stallings e o site <http://www.intersil.com/data/fn/fn2784.pdf>)

O 82C59A é um Controlador de interrupções de alto desempenho (para a época) aliviando o processador da tarefa de gerenciamento de interrupções e permitindo um sistema multinível de prioridades. O 82C59A pode suportar um vetor de 8 prioridades de interrupções podendo ser cascadeáveis até 64. Fontes individuais de interrupção podem ser mascaradas ou priorizadas de forma a permitir uma configuração personalizada.

a) Descrição dos pinos:



- D0 a D7: Barramento de dados (bidirecional), por onde são transferidos os sinais de controle, status, e informações sobre o vetor de interrupções

- RD (Read Input): Quando em nível baixo juntamente com o sinal CS, o controlador coloca o status no barramento de dados para a CPU
- WR (Write Input): Quando em nível baixo juntamente com o sinal CS, o controlador aceita palavras de comando a partir do processador pelo barramento de dados
- A0 (Command Select Address): Atua em conjunto com CS, WR e RD. É utilizado para decifrar vários comandos dados pela CPU e status lidos por essa
- CS (Chip Select) : Quando em nível baixo juntamente com o leitura (RD) e escrita (WR), habilita a comunicação entre a CPU e o Controlador
- CAS 0 - CAS 2 (Cascade Lines): Utilizados para controlar múltiplas estruturas de controladores 82C59A. São saídas para controladores 82C59A mestres e entradas para controladores 82C59A escravos
- SP/EN (Slave Program Input Enable): Possui dupla função. Quando em modo de buffered (armazenamento) é utilizado como modo de controle. Quando não está em modo de armazenamento é utilizado para designar um mestre ou escravo
- INT (Interrupt Output): Quando em nível alto, informa a CPU que uma interrupção foi acionada
- INTA (Interrupt Acknowledge Input): Quando este pino é ativado, o controlador coloca o vetor de dados de interrupção no barramento de dados
- IR0 a IR7 (Interrupt Request Inputs): Linhas de interrupção que indicam um pedido de interrupção
- Vcc e GND: Alimentação, sendo +5V (positivo da fonte) e terra (negativo da fonte) respectivamente

b) Modo de funcionamento do controlador

Resumidamente, o controlador 82C59A tem a responsabilidade de gerenciamento das interrupções procedendo da seguinte forma: recebe requisições de interrupção dos módulos a ele conectados, determina aquela que tem maior prioridade e então envia um sinal para o processador, na linha INT. O processador responde com um sinal de reconhecimento, ativando a linha do INTA. Ao receber o sinal, o 82C59A coloca o vetor de interrupções apropriado no barramento de dados. O processador pode então processar a interrupção, comunicando-se diretamente com o módulo E/S, para ler ou escrever.

c) Modo de programação do controlador pela UCP.

O 82C59A aceita dois tipos de palavras de comando geradas pela UCP:

- Inicialização: São utilizadas para colocar o 82C59A pronto para operar
- Operação: São utilizadas pelo processador para especificar o esquema de prioridades a ser usado, carregando um valor na palavra de controle do 82C59A. Podem ser adotados os seguintes modos de operação: totalmente aninhado, onde as requisições de interrupções são ordenadas de acordo com as prioridades de 0 (para IR0) a 7 (para IR7); circular, onde diversos dispositivos possuem a mesma prioridade de interrupção, e quando a interrupção de um dispositivo acaba de ser atendida, este dispositivo recebe a prioridade mais baixa do grupo; e a máscara especial que possibilita ao processador inibir interrupções de determinados dispositivos.

9. (1,6) Para especificar um sistema de entrada e saída, os projetistas devem ter como objetivo fazer com que a carga imposta ao sistema seja atendida pelos dispositivos de entrada/saída e pelos barramentos que conectam os diversos componentes do sistema. Considere um sistema de computação com as seguintes características:

- A UCP consegue executar 600 milhões de instruções por segundo.
- Cada instrução de entrada/saída requer 240.000 instruções.
- A taxa de transferência de dados no barramento de sistema que conecta UCP, memória e discos é 120 M bytes/s.
- Um controlador de disco pode controlar até 7 discos e a taxa máxima de transferência de bytes entre ele e os discos é 20 M bytes/s.
- Os discos possuem uma taxa de leitura/escrita de 6 M bytes/s e um tempo médio de procura adicionado à latência rotacional de 10 ms.
- Em cada acesso ao disco são transferidos blocos de 64 Kbytes, sendo que os blocos estão localizados seqüencialmente nas trilhas

Para este sistema, deve-se calcular o número de discos e de controladores de tal maneira que a máxima taxa possível de transferência de entrada e saída seja atingida. Neste caso, sempre considere que $1\text{ K}=10^3$, $1\text{ M}=10^6$, $1\text{ G}=10^9$.

Para chegar a estes números, execute os seguintes passos:

- (1) A UCP e o barramento de sistema são os componentes com características já definidas para este sistema. Então, o primeiro passo é calcular a taxa máxima de operações de entrada/saída (número de operações de entrada/saída por segundo) que cada um destes componentes pode sustentar. A menor destas taxas será a maior taxa sustentável do sistema, e o número de discos e de controladores será calculado para atingir esta taxa.

- 1.1 Calcule a taxa máxima suportada pela UCP, sabendo que ela pode executar 600 milhões de instruções por segundo e que cada operação de entrada/saída demanda 240.000 instruções.

Resposta:

$$\begin{aligned} &\text{Taxa máxima suportada pela UCP} \\ &= (\text{quant. instruções executáveis pela UCP} / \text{seg}) / (\text{quant. instruções} / \text{oper. ent.saída}) \\ &= 600.000.000 / 240.000 \\ &= 2500 \text{ operações de entrada e saída} / \text{segundo} \end{aligned}$$

- 1.2 Calcule a taxa máxima suportada pelo barramento do sistema sabendo que em cada operação de acesso ao disco (operação de entrada/saída considerada) são transferidos 64 K bytes e a taxa de transferência no barramento é 120 M bytes/s.

Resposta:

$$\begin{aligned} &\text{Taxa máxima suportada pelo barramento} \\ &= \text{taxa de transferência no barramento} / (\text{taxa de transferência} / \text{oper. ent.saída}) \\ &= (120 \text{ Mbytes} / \text{s}) / (64 \text{ Kbytes} / \text{oper. ent.saída}) \\ &= 1875 \text{ operações de entrada e saída} / \text{segundo} \end{aligned}$$

- (2) A menor das taxas obtidas no item 1 será a taxa considerada para o cálculo do número de discos e controladores. Para calcular o número de discos, execute os seguintes passos:

- 2.1 Calcule o tempo gasto pelo disco para executar uma operação de entrada/saída, lembrando que o tempo de procura mais de latência é 10 ms, a taxa de transferência do disco é 6 M bytes/s e em cada operação são transferidos 64 K bytes.

Resposta:

$$\begin{aligned} &\text{Tempo gasto pelo disco para executar uma operação de entr/saída} \\ &= \text{tempo de procura mais de latência} + ((\text{quant bytes transferidos}) / \text{taxa do disco}) \\ &= 10\text{ms} + ((64 \text{ Kbytes} / \text{operação}) / 6 \text{ Mbytes}) \\ &= 10\text{ms} + 10,67\text{ms} \\ &= 20,67\text{ms} \end{aligned}$$

- 2.2 Calcule a taxa máxima de operações de entrada/saída (o número máximo de operações de entrada/saída em 1 segundo) que cada disco pode executar.

Resposta:

$$\begin{aligned} &\text{Taxa máxima de operações de entrada/saída} \\ &= 1 \text{ segundo} / \text{tempo para ocorrer uma operação} \\ &= 1 \text{ s} / (20,67\text{ms} / \text{operação}) \\ &= 48,38 \text{ operações/seg} \end{aligned}$$

- 2.3 Calcule o número de discos utilizando a taxa máxima suportada pelo sistema e a taxa máxima de cada disco.

Resposta:

Número de discos

$$\begin{aligned} &= \text{taxa máxima suportada pelo sistema (menor valor item 1) / taxa máxima de cada disco} \\ &= (1875 \text{ operações / seg}) / (48,38 \text{ operações/seg}) \\ &= 38,75 \text{ discos (considerar 39 discos)} \end{aligned}$$

- (3) Para calcular o número de controladores de discos, execute os seguintes passos:

- 3.1 Calcule a taxa de transferência (bytes/s) de cada disco, utilizando o tamanho do bloco acessado em cada acesso ao disco e o tempo necessário para transferi-lo de/para o disco.

Resposta:

Taxa de transferência de cada disco

$$\begin{aligned} &= (1 / \text{tempo necessário para transferi-lo}) \times \text{tamanho do bloco} \\ &= (1 / 20,67\text{ms}) \times 64\text{Kbytes} \\ &= 3096 \text{ Kbytes / s} \end{aligned}$$

- 3.2 Calcule o número máximo de discos que cada controlador pode suportar utilizando a taxa de transferência calculada no item anterior e a taxa de transferência suportada por cada controlador (20 M bytes).

Resposta:

Número máximo de discos para cada controladora

$$\begin{aligned} &= \text{taxa de transferência suportada pelo controlador / taxa transferência por disco} \\ &= (20 \text{ Mbytes / s}) / (3096 \text{ Kbytes / s}) \\ &= 6,45 \text{ discos (considerar 6 discos)} \end{aligned}$$

- 3.3 Calcule o número de controladores utilizando o número de discos necessário e o número de discos suportados por cada controlador calculado no item anterior.

Resposta:

Número de controladores

$$\begin{aligned} &= \text{número de discos necessários / (número de discos / controlador)} \\ &= 39 \text{ discos / 6discos/controlador} \\ &= 6,5 \text{ controladores (considerar 7 controladores)} \end{aligned}$$