

AD2 - Organização de Computadores 2007.1- GABARITO

Data de entrega – 26 de maio

1. (1,0) Crie 3 conjuntos de instruções de um, dois, e três operandos, definidas em Linguagem Assembly, necessárias para a realização de operações aritméticas e elabore programas para o cálculo das seguintes equações (3 programas para cada uma):

Resposta:

I_ CONJUNTO DE INSTRUÇÕES PARA 1 OPERANDO:

```
LOAD M    =>    ACC    <-  M
STORE M    =>    M      <-  ACC
ADD M      =>    ACC    <-  ACC + M
SUB M      =>    ACC    <-  ACC - M
MUL M      =>    ACC    <-  ACC * M
DIV M      =>    ACC    <-  ACC / M
```

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

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

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

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

II_ CONJUNTO DE INSTRUÇÕES PARA 2 OPERANDOS:

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

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

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

```
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    =>    T1 <- 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
```

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

```
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
```

III_ CONJUNTO DE INSTRUÇÕES PARA 3 OPERANDOS:

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

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

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

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

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

```
MUL T1, E, T1    =>    T1 = T1 * E
ADD A, T1, X      =>    X = A + T1
```

2. (1,0) Considere o programa assembly abaixo (baseado no assembly visto no capítulo 9 do livro texto) e o respectivo código de máquina:

```

ORG
      LDA Z      11F
      ADD Y      320
      SUB T      421
      STR X      222
      JZ FIM     51E
      PRT X      B22
FIM   HTL        000

```

Considere que as variáveis Y, Z, T e X estão armazenadas na memória nos seguintes endereços e com os conteúdos apresentados abaixo:

Variável	Endereço (hexadecimal)	Valor (hexadecimal)
Y	1F	051
Z	20	03E
T	21	003
X	22	01A

Considere que o programa está armazenado na MP a partir do endereço 18 (hexadecimal), e o contador de instruções tem o valor 18. Para a execução de cada instrução, mostre os valores do CI, RI, acumulador e das variáveis Y, Z, T e X

Resposta:

OBS: Devido a apresentar uma troca entre endereços e instruções nas duas primeiras instruções, será aceito qualquer uma das soluções abaixo. Não haverá diferença no resultado final, apenas na 2ª. E 3ª. Linha da tabela.

1ª. Solução:

```

LDA Y  11F
ADD Z  320
SUB T  421
STR X  222
JZ FIM 51E
PRT X  B22

```

End	Cont.	End .	Conteúdo	Y (1F)	Z (20)	T (21)	X (22)	CI	RI	ACC
				051	03E	003	01A	18		
18	11F	LDA 1F	ACC ← (1F)	051	03E	003	01A	19	11F	051
19	320	ADD 20	ACC ← ACC + (20)	051	03E	003	01A	1A	320	08F
1ª	421	SUB 21	ACC ← (21)	051	03E	003	01A	1B	421	08C
1B	222	STR 22	(22) ← ACC	051	03E	003	08C	1C	222	08C
1C	51E	JZ 1E	Se ACC=0, então CI ← (1E)	051	03E	003	08C	1D	51E	08C
1D	B22	PRT 22	Exibe conteúdo de (22)	051	03E	003	08C	1E	B22	08C
1E	000	HLT	HALT	051	03E	003	08C	*	000	08C

2ª. Solução:

LDA Z 120
ADD Y 31F
SUB T 421
STR X 222
JZ FIM 51E
PRT X B22

End	Cont.	End	Conteúdo	Y (1F)	Z (20)	T (21)	X (22)	CI	RI	ACC
				051	03E	003	01A	18		
18	120	LDA 20	ACC <- (20)	051	03E	003	01A	19	120	03E
19	31F	ADD 1F	ACC <- ACC + (1F)	051	03E	003	01A	1A	31F	08F
1ª	421	SUB 21	ACC <- (21)	051	03E	003	01A	1B	421	08C
1B	222	STR 22	(22) <- ACC	051	03E	003	08C	1C	222	08C
1C	51E	JZ 1E	Se ACC=0, então CI <- (1E)	051	03E	003	08C	1D	51E	08C
1D	B22	PRT 22	Exibe conteúdo de (22)	051	03E	003	08C	1E	B22	08C
1E	000	HLT	HALT	051	03E	003	08C	*	000	08C

3. (1,0) Faça uma busca na lista dos 500 sistemas de computadores com melhor desempenho do mundo em <http://www.top500.org> e descreva os três primeiros colocados (pesquise neste mesmo site e na internet).

Resposta:

Primeiro Colocado é o BlueGene/L - eServer Blue Gene Solution IBM, com cerca de 131.072 processadores. Entrou em operação em 2005. Com poder máximo de processamento de cerca de 280.600 (Gflops) e de processamento de pico de cerca de 367.000 (Gflops).

Pertence ao Laboratório Nacional de Lawrence Livermore (LLNL) ligado à Administração Nacional da Segurança Nuclear (NNSA), do departamento de energia (DOE) do governo de Estados Unidos.

O BlueGene/L possui 131.072 processadores, e cerca de 32 Tbytes de Memória. Foi instalado em 2005 e implantado pela IBM.

BlueGene/L possui uma arquitetura escalável, em que o poder computacional da máquina pode ser expandido adicionando-se mais blocos, sem introdução de gargalos.



Segundo Colocado : Red Storm - Sandia/ Cray Red Storm, Opteron 2.4 GHz dual core com cerca de 26544, com poder máximo de processamento de cerca de 101.400 (Gflops) e de processamento de pico de cerca de 127.411 (Gflops).

O Red Storm é um computador de processamento maciçamente paralelo (MPP) projetado de forma a ser escalável, variando de um único armário e de relativamente poucos processadores até centenas dos armários e milhares de processadores. O sistema usa processadores de elevado volume da produção combinados em malhas de interconexão 3-D de alto desempenho para produzir uma elevada eficiência paralela em uma larga gama de aplicações científicas e da engenharia. O Red Storm é da família da CRAY XT3 e fabricado pela Cray Inc.

Foi instalado nos Laboratórios Nacionais de Sandia em Albuquerque, no inverno de 2004 (fim do ano de 2004).

Características iniciais:

- 124.42 teraOPS (performance teórica de pico)
- 135 armários
- 12,960 processadores
- Processadores AMD Opteron™ 2.4 GHz dual core
- 40 terabytes de memória DDR
- 340 terabytes de armazenamento de disco
- Sistema Operacional Linux/Catamount
- Consumo de cerca de 2.5 megawatts de energia e refrigeração



Terceiro Colocado : BGW - eServer Blue Gene Solution IBM com cerca de 40960 processadores, com poder máximo de processamento de cerca de 91.290 (Gflops) e de processamento de pico de cerca de 114.688 (Gflops). Instalado em 2005. Pertence a família do Blue Gene Solution da IBM, com organização similar ao do 1º. colocado.

4. (1,0) Responda:

a) Dados os valores de memória abaixo e uma máquina de 1 endereço com um acumulador:

palavra 20 contém 40
palavra 30 contém 50
palavra 40 contém 60
palavra 50 contém 70

Quais valores as seguintes instruções carregam no acumulador?

- LOAD IMEDIATO 20

Resposta: Nesta instrução o valor a ser colocado no acumulador corresponde ao valor fornecido como operador, portanto:

ACC <- 20 (o valor a ser colocado no acumulador é 20)

- LOAD DIRETO 20

Resposta: Nesta instrução o valor a ser colocado no acumulador corresponde ao valor contido no endereço de memória fornecido como operador, portanto:

ACC <- (20) (o valor a ser colocado no acumulador é 40)

- LOAD INDIRETO 20

Resposta: Nesta instrução o valor a ser colocado no acumulador corresponde ao valor contido no endereço que consta como valor no endereço de memória fornecido como operador, portanto:

$ACC \leftarrow ((20))$ (o valor a ser colocado no acumulador é 60)

b) Analise os modos de endereçamento direto, indireto e imediato, estabelecendo diferenças de desempenho, vantagens e desvantagens de cada um.

Resposta:

MODO DE ENDEREÇAMENTO	DEFINIÇÃO	VANTAGENS	DESvantagens	DESEMPENHO
Imediato	O campo operando contém o dado	Rapidez na execução da instrução	Limitação do tamanho do dado. Inadequado para o uso com dados de valor variável	Não requer acesso a memória principal. Mais rápido que o modo direto
Direto	O campo operando contém o endereço do dado	Flexibilidade no acesso a variáveis de valor diferente em cada execução do programa	Perda de tempo, se o dado é uma constante	Requer apenas um acesso a memória principal. Mais rápido que o modo indireto
Indireto	O campo operando contém o endereço da posição onde está o dado	Manuseio de vetores (quando o modo indexado não está disponível). Usar como "ponteiro"	Muitos acessos à MP para execução	Requer 2 acessos a memória principal

c) Qual é o objetivo do emprego do modo de endereçamento base mais deslocamento? Qual é a diferença de implementação entre esse modo e o modo indexado?

Resposta:

O base mais deslocamento tem como seu principal objetivo permitir a modificação de endereço de programas ou módulos destes (que é a realocação 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. Difere do modo indexado onde o do registrador base é fixo e ocorre variação no deslocamento, pois ao contrário, em endereçamento base mais deslocamento o deslocamento é fixo e há alteração do registrador base.

5. (1,0) Explique, comparando:

a) Compilação e Interpretação

Resposta: A compilação consiste na análise de um programa escrito em linguagem de alto nível (programa fonte) e sua tradução em um programa em linguagem de máquina (programa objeto).

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

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

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 podemos 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 de reduzir de ciclos de relógio para execução das instruções. Instruções de formatos simples e únicos tiram maior proveito de execução com pipeline cujos estágios consomem o mesmo tempo.

6. (1,0) Verifique se é válida a seguinte definição para overflow (estouro) em uma operação aritmética na representação complemento a dois utilizando-se n bits: Caso os bits “vai-um” nas posições $n-1$ e n sejam diferentes ocorreu overflow. Caso contrário não ocorreu overflow.

Resposta: Esta definição é válida, como mostrado a seguir.

Considere dois números A e B representados utilizando-se n bits:

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

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

Os bits c_n e c_{n-1} correspondem aos bits “vai-um” nas posições n e $n-1$ respectivamente

$$\begin{array}{r} c_n \quad c_{n-1} \\ a_{n-1} \dots a_0 \\ + \\ b_{n-1} \dots b_0 \end{array}$$

Verificando cada caso possível dos valores dos bits a_{n-1} , b_{n-1} , e c_{n-1} :

a_{n-1}	b_{n-1}	c_{n-1}	Overflow?	$c_n \neq c_{n-1}$?
0	0	0	Não. Dois números positivos somados tem como resultado um número positivo	Não
0	0	1	Sim. Dois números positivos somados tem como resultado um número negativo	Sim
0	1	0	Não. Nunca ocorre overflow quando um número positivo é somado com um número negativo	Não
0	1	1	Não. Nunca ocorre overflow quando um número positivo é somado com um número negativo	Não

1	0	0	Não. Nunca ocorre overflow quando um número positivo é somado com um número negativo	Não
1	0	1	Não. Nunca ocorre overflow quando um número positivo é somado com um número negativo	Não
1	1	0	Sim. Dois números negativos somados tem como resultado um número positivo	Sim
1	1	1	Não. Dois números negativos somados tem como resultado um número negativo	Não

7. (1,0) 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) (0,4) Considere o seguinte conjunto de bits representado em hexadecimal D1200100. Indique o valor deste número em decimal, considerando-se que o conjunto representa:

Resposta:

O número $(D1200100)_{16} = (11010001\ 00100000\ 00000001\ 00000000)_2$

(a.1) um inteiro sem sinal

Temos: $2^{31} + 2^{30} + 2^{28} + 2^{24} + 2^{21} + 2^8 = 3.508.535.552$

(a.2) um inteiro em sinal magnitude

Temos: $-(2^{30} + 2^{28} + 2^{24} + 2^{21} + 2^8) = -1.361.051.904$

(a.3) um inteiro em complemento a 2

Temos: $-2^{31} + (2^{30} + 2^{28} + 2^{24} + 2^{21} + 2^8) = -786.431.744$

(a.4) um real em ponto flutuante.

1 10100010 01000000000000100000000

Sinal: 1 (negativo)

Mantissa: 1,0100000 00000001

Expoente: $10100010 = 162 - 127 = 35$

Normalizado: $-1,0100000\ 00000001 \times 2^{35} \Rightarrow$
 $-(10100000\ 000000010000000000000000000000)_2 = (-42.950.721.536)_{10}$

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

b.1) +122,25

Convertido para base 2: 1111010,01

Normalizado: $1,11101001 \times 2^6$

Sinal: 0 (positivo)

Expoente: $6 + 127 = 133 = 1000101$

Mantissa: 11101001

Resposta: 0 1000101 111010010000000000000000

Arrumado em bytes: 01000010 11110100 10000000 00000000

b.2) -33,1

Convertido para base 2: 100001,0001100110011001100...

Normalizado: $1,000010001100110011001100... \times 2^5$

Sinal: 1 (negativo)

Expoente: $5 + 127 = 132 = 10000100$

Mantissa: 000010001100110011001100...

Resposta: 1 10000100 00001000110011001100110

Arrumado em bytes: 11000010 00000100 01100110 01100110

c) (0,2) Mostre a representação dos números dos itens da questão acima, caso se utilizasse a representação em complemento a 2 para representar o expoente.

c.1) +122,25

Convertido: 1111010,01

Normalizado: $1,11101001 \times 2^6$

Sinal: 0 (positivo)

Expoente: +6 = 00000110

Mantissa: 11101001

Resposta: 0 00000110 111010010000000000000000

Arrumado em bytes: 00000011 01110100 10000000 00000000

c.2) -33,1

Convertido: 100001,0001100110011001100...

Normalizado: $1,000010001100110011001100... \times 2^5$

Sinal: 1 (negativo)

Expoente: +5 = 00000101

Mantissa: 000010001100110011001100...

Resposta: 1 00000101 00001000110011001100110

Arrumado em bytes: 10000010 10000100 01100110 01100110

d) (0,2) Supondo que se utilize a representação complemento a 2 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.

Menor 01000000 10000000 00000000 00000000
 $+(1,000000000000000000000000)_2 \times 2^{-127} \approx 5,9 \times 10^{-39}$

Maior 00111111 01111111 11111111 11111111 =
 $+(1,111111111111111111111111)_2 \times 2^{+126} \approx 1,7 \times 10^{-39}$

8. (1,0) Explique como funciona um barramento SCSI (sugestões de fonte de consulta: livro do Stallings e a página <http://informatica.hsw.com.br/scsi.htm>)

Resposta:

O barramento SCSI (Small Computer System Interface) é um barramento de comunicação rápida que conecta vários dispositivos a um computador ao mesmo tempo, incluindo discos rígidos, scanners, CD-ROM/RW e impressoras. O SCSI tem vários benefícios: é muito rápido - chega a 320 Mb/s - e está no mercado há mais de 20 anos, sendo

incansavelmente testado - portanto, tem a reputação de ser confiável. Da mesma maneira que o Serial-ATA eo Fire-Wire, ele aceita a ligação de vários dispositivos diferentes em um barramento.

No entanto, o SCSI também tem alguns problemas. Tem suporte limitado a BIOS do sistema e precisa ser configurado para cada computador. Também não existe uma interface de software comum para SCSI. Por fim, todos os tipos diferentes de SCSI têm velocidades, largura de barramento e conectores diferentes, o que pode ser confuso. Mas, uma vez conhecido o significado de "Fast", "Ultra" e "Wide", fica fácil entender.

O SCSI tem três especificações básicas:

- SCSI-1: especificação original desenvolvida em 1986, está obsoleta. Tinha largura de barramento de 8 bits e velocidade de clock de 5 MHz.
- SCSI-2: adotada em 1994, esta especificação incorporou o Common Command Set - CCS (Conjunto de Comandos Comuns) - 18 comandos considerados imprescindíveis para o suporte a qualquer dispositivo SCSI. Também tinha a possibilidade de dobrar a velocidade de clock para 10 MHz (Fast), dobrar a largura de barramento para 16 bits e aumentar o número de dispositivos para 15 (Wide), ou ambos (Fast/Wide). O SCSI-2 também incorporou o enfileiramento de comandos, permitindo que os dispositivos armazenassem e priorizassem comandos do computador onde estavam instalados.
- SCSI-3: especificação de 1995 que trouxe uma série de padrões menores dentro de sua abrangência geral. Um conjunto de padrões envolvendo a Interface Paralela SCSI (SPI), que é a forma como os dispositivos SCSI se comunicam, continuou a evoluir dentro do SCSI-3. A maioria das especificações SCSI-3 começam com o termo Ultra: Ultra para variações SPI, Ultra2 para variações SPI-2 e Ultra3 para variações SPI-3. As designações Fast (rápido) e Wide (largo) funcionam como no SCSI-2. SCSI-3 é o padrão atualmente em uso.

Combinações diferentes de velocidade de barramento dobrada e de clock dobrada, e especificações SCSI-3 resultaram em muitas variações SCSI. Além da velocidade de barramento expandida, o Ultra320 SCSI usa transferência de dados em pacotes, aumentando sua eficiência. O Ultra2 também foi o último modelo a ter largura de banda "estreita", de 8 bits.

Os tipos citados até então de SCSI são paralelos: bits de dados movem-se simultaneamente pelo barramento, ao invés de um por vez. O tipo mais novo de SCSI, chamado de Serial Attached SCSI (SAS), usa comandos SCSI, mas transmite dados de forma serial. O SAS usa uma conexão ponto-a-ponto para mover dados a 3 Gb/s, e cada porta SAS suporta até 128 dispositivos ou expansores.

Os tipos diferentes de SCSI usam controladoras e cabos para se comunicar com os dispositivos.



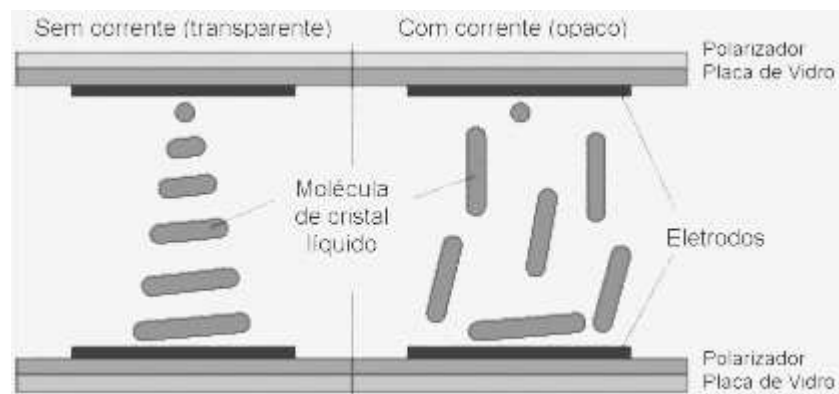
Controladora SCSI

9. (1,0) Explique como funcionam os monitores de vídeo de cristal líquido (LCD) (sugestões de fonte de consulta: livro do Mário Monteiro e as páginas: <http://www.infowester.com/monlcd.php> e <http://www.guiadohardware.net/tutoriais/042/>)

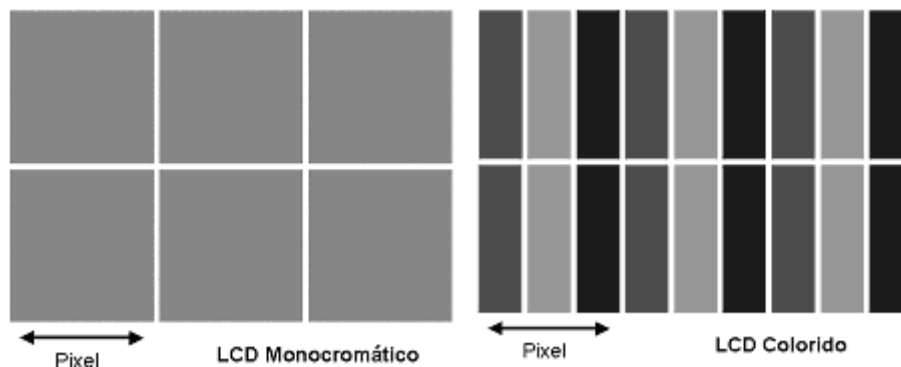
A imagem no LCD utiliza uma tecnologia baseada no uso de cristais líquidos para formar a imagem. Estes cristais líquidos são substâncias que tem sua estrutura molecular alterada quando recebem corrente elétrica. Em seu estado normal, estas substâncias são transparentes, mas ao receberem uma carga elétrica tornam-se opacas, impedindo a passagem da luz. Nos visores de cristal líquido mais primitivos, como os dos relógios de pulso, temos apenas estes dois estados, transparente e opaco, ou seja, ou o ponto está aceso ou está apagado. Nos visores mais sofisticados, como os usados em notebooks, temos também estados intermediários, que formam as tonalidades de cinza ou as cores. Estes tons intermediários são obtidos usando-se tensões diferentes.

Para formar a tela de um monitor, uma fina camada de cristal líquido é colocada entre duas camadas de vidro. Estas finas placas possuem pequenos sulcos, isolados entre si, cada um com um eletrodo ligado a um transistor. Cada um destes sulcos representa um dos pontos da imagem. Este sanduíche por sua vez é colocado entre duas camadas de um

elemento polarizador. Atrás desta tela é instalada uma fonte de luz, geralmente composta de lâmpadas fluorescentes (usadas por gerarem pouco calor) ou então Leds, responsáveis pela iluminação da tela.



No caso de LCDs mono-cromáticos, cada ponto da tela corresponde a um dos pontos da imagem. Já no caso dos monitores coloridos, cada pixel da imagem é formado por um grupo de 3 pontos, um verde, um vermelho e outro azul. Como nos monitores CRT as cores são obtidas através de diferentes combinações de tonalidades dos três pontos.



Existem atualmente duas tecnologias de fabricação de telas de LCD, conhecidas como matriz passiva (DSTN) e matriz ativa (TFT). As telas de matriz passiva apresentam um ângulo de visão mais restrito, e um tempo maior é necessário para a imagem ser atualizada. Enquanto num monitor CRT, um ponto demora cerca de 15 a 20 milissegundos para mudar de cor, em um monitor LCD de matriz passiva são necessários entre 150 e 250 milissegundos. Por isso que pode ser difícil enxergar o cursor do mouse na tela de um notebook, ou mesmo rodar programas ou jogos que demandem mudanças rápidas de imagem de uma forma aceitável. A própria imagem nestes monitores apresenta uma qualidade inferior, devido ao baixo contraste. Os monitores de matriz passiva são encontrados apenas em equipamentos antigos, não sendo mais fabricados atualmente.

Os LCDs de matriz ativa, usados atualmente, já apresentam uma qualidade muito superior, com um tempo de atualização de imagem mais próximo do dos monitores CRT, entre 40 e 50 milissegundos. Isto significa entre 20 e 25 quadros por segundo, o que já é suficiente para assistir a um filme em DVD por exemplo, apesar de ainda atrapalhar um pouco nos jogos de ação, onde a imagem é alterada muito rapidamente. Os monitores de matriz ativa também um maior ângulo de visão e contraste maiores, além de serem mais finos e leves.

Ao contrário dos monitores CRT atuais, todos os monitores de cristal líquido são digitais. Como todas as placas de vídeo atuais enviam sinais analógicos para o monitor, é usado um novo circuito que converte os sinais analógicos enviados pela placa de vídeo novamente para o formato digital que é entendido pelo monitor.

A mudança digital-analógico-digital neste caso é totalmente desnecessária, e serve apenas para degradar a qualidade da imagem e aumentar a quantidade de circuitos usados no monitor, encarecendo-o. Segundo os fabricantes, o custo dos monitores LCD pode cair em mais de 100 dólares com o uso de placas de vídeo que emitam sinais digitais.

Os monitores LCD trazem várias vantagens sobre os monitores CRT como:

- São muito mais finos que os tradicionais, o que explica seu uso em computadores portáteis.
- Possuem uma tela realmente plana, o que elimina as distorções de imagem causadas pelas telas curvas dos monitores CRT, e aumenta a área útil do monitor, já que não temos espaços desperdiçados nos cantos da imagem.
- Gastam menos eletricidade.
- Emitem uma quantidade muito menor de radiação nociva (praticamente nenhuma em alguns modelos) o que os torna especialmente atraentes para quem fica muito tempo em frente ao monitor diariamente.
- Não existe flicker, pois ao invés da imagem ser formada pela ação do feixe de elétrons, como nos monitores CRT, cada ponto da tela atua como uma pequena lâmpada, que muda sua tonalidade para formar a imagem.

E trazem como desvantagens.:

- A área de visão (ângulo) é mais limitada,
- Contraste é mais baixo,
- As resoluções permitidas são bem mais limitadas.
- Custo maior em relação aos de tubo de raios catódicos

10. (1,0) Explique como funcionam as três técnicas para realizar operações de entrada e saída: E/S programada, E/S dirigida por interrupção e acesso direto à memória. Para cada uma das técnicas, indique pelo menos um dispositivo que a utiliza em sistemas comerciais.

Resposta:

a) E/S por programa: O processador tem controle direto sobre a operação de E/S, incluindo a detecção do estado do dispositivo, o envio de comandos de leitura ou escrita e transferência de dados. Para realizar uma transferência de dados, o processador envia um comando para o módulo de E/S e fica monitorando o módulo para identificar o momento em que a transferência pode ser realizada. Após detectar que o módulo está pronto, a transferência de dados é realizada através do envio de comandos de leitura ou escrita pelo processador. Se o processador for mais rápido que o módulo de E/S, essa espera representa um desperdício de tempo de processamento. Esta técnica é utilizada por alguns sistemas para se comunicar com alguns tipos de mouse e teclado. Nestes sistemas, uma rotina que verifica se houve alguma mudança na posição do mouse ou se o usuário digitou alguma nova tecla é acionada com uma determinada frequência.

b) E/S por interrupção: Neste caso, o processador envia um comando para o módulo de E/S e continua a executar outras instruções, sendo interrompido pelo módulo quando ele estiver pronto para realizar a transferência de dados, que é executada pelo processador através da obtenção dos dados da memória principal, em uma operação de saída, e por armazenar dados na memória principal, em uma operação de entrada. Geralmente impressoras podem trabalhar em modo E/S por programa ou por interrupção.

c) E/S por DMA: Nesse caso a transferência de dados entre o módulo de E/S e a memória principal é feita diretamente sem envolver o processador. Existe um outro módulo denominado controlador de DMA que realiza a transferência direta de dados entre a memória e o módulo de E/S. Quando o processador deseja efetuar a transferência de um bloco de dados com um módulo de E/S, ele envia um comando para o controlador de DMA indicando o tipo de operação a ser realizada (leitura ou escrita de dados), endereço do módulo de E/S envolvido, endereço de memória para início da operação de leitura ou escrita de dados e número de palavras a serem lidas ou escritas. Depois de enviar estas informações ao controlador de DMA, o processador pode continuar executando outras instruções. O controlador de DMA executa a transferência de todo o bloco de dados e ao final envia um sinal de interrupção ao processador, indicando que a transferência foi realizada. Alguns dispositivos que podem trabalhar com DMA: gravador de CD/DVD, disco rígido e placas de som.