

Sistemas Microprogramados

Pós-Aulas 01 a 05 - Prof. Clayton A. Valdo, Me

Pós - aula 01

1. Qual o valor decimal de 011011012? Qual a representação binária de 654?

R. 011011012 => 109;

R. 654 => 1010001110;

2. Converter para binário os seguintes números decimais:

a) 39 => 100111;

b) 0,4475 => 01110;

c) 256,75 => 100000000,11

d) 129,5625 => 10000001,1001

3. Converter para decimal os seguintes números binários:

a) 01101 => 13

b) 0,001101 => 0,203125

c) 010110011 => 179

d) 0111011,1011 => 59,6875

4. Quantos números diferentes podem ser representados em uma palavra binária de 6 bits?

R. Podemos representar 63 números diferentes em uma palavra binária de 6 bits.

5. Quantos números de base 4 podemos representar com 7 bits com e sem sinal?

R. Podemos representar 8191 números diferentes com base 4 com 7 bits, sendo positivos ou negativos.

6. Escrever os 12 primeiros números no sistema de numeração de base 5.

R. 05 – 15 – 25 – 35 – 45 – 105 – 115 – 125 – 135 – 145 – 205 – 215

7. Converta os números 17 e 15 para binário usando 6 bits e efetue a operação de soma entre eles (17 + 15), usando as seguintes representações e verifique se houve estouro:

a) 15dec=1111bin

17dec=10001bin

O sinal magnitude de ambos é 0, pois são positivos

b) Complemento de um;

01111
+10001
100000

$2^*B^{(n-1)}-1=$

$2^*2^{(5-1)}-1= 31$

C). Analise os resultados obtidos quanto ao estouro.

R: Houve estouro de uma casa.

8. Repita o exercício 7 para os números -17 e -15 (realizando a soma -17 + (-15)).

a) Sinal magnitude

R: -15dec=111110bin

-17dec=110001bin

O sinal magnitude de ambos é 1, pois são negativos

b) Complemento de um;

```
  111110
+110001
-----
 1101111
```

$$2^*B^{(n-1)}-1=$$

$$2^*2^{(6-1)}-1= 63$$

c) Analise os resultados obtidos quanto ao estouro.

R: Houve estouro de uma casa.

9. Mostre como somar em complemento de um, para n=6 bits, as seguintes parcelas decimais:

a) 27 e -8

$$27 = 011011$$

$$-8 = 101000$$

```
  0011011
+1101000
-----
 10000011
```

b) 17 e -18

$$17 = 010001$$

$$-18 = 110010$$

```
  0010001
+1110010
-----
 10000011
```

c) 1 e -5

$$1 = 000001$$

$$-5 = 100101$$

```
  0000001
+1100101
-----
 1100110
```

Pós - aula 02

1. Quais as consequências da Lei de Moore? Explique.

R. As consequências da lei de Moore foram:

- * O custo de implementação da lógica computacional caiu drasticamente, fazendo com que o custo de uma pastilha de silício permanecesse praticamente inalterado;
- * A velocidade de operação aumentou, devido as portas lógicas e as células de memória estarem cada vez mais próximas, tornando o caminho elétrico menor;
- * Miniaturização dos equipamentos eletrônicos;
- * Redução no consumo de energia;
- * O número de conexões necessárias entre pastilhas é muito menor devido ao aumento de circuitos em cada pastilha. A interconexão de um CI é mais confiável que as conexões soldadas.

2. Quais os problemas da arquitetura de Von Neumann?

R. O principal problema da arquitetura é conhecido por “Gargalo de Von Neumann”, que em resumo, só pode ser trafegado uma instrução ou um dado pelo barramento do sistema a cada ciclo. Não é possível explorar o paralelismo de operações para acelerar o processamento. O segundo problema refere – se ao tempo que a CPU passa ociosa, pois o tempo de execução das instruções que envolve apenas operações internas à CPU é muito menor do que o tempo para transferência entre memória e CPU.

3. Uma linguagem revolucionária foi escrita em um novo patamar L6, nível este capaz de ler os pensamentos e traduzi-los em comandos de código. Descreva os níveis desta nova linguagem.

R: Visto que, quanto mais rápida, maior o custo e menor a capacidade e quanto mais lenta, mais barata e maior a capacidade, a memória L6 ficaria no topo da pirâmide, pois exigiria alta flexibilidade, por um custo e capacidade de armazenamento não tão baixos.

A alta flexibilidade, seria pela rapidez que os pensamentos fluem. A capacidade de armazenamento, baixa, uma vez que os pensamentos são voláteis, inconstantes e instáveis. E por fim, o custo, seria ofertado de acordo com a demanda.

Pós - aula 03

1. Qual o princípio na hierarquia de memórias que as caches utilizam? Explique.

R: A memória cache fica entre o processador e a memória RAM. E consiste numa pequena quantidade de memória SRAM, incluída no chip do processador.

Quando este precisa ler dados na memória RAM, um circuito especial, chamado de controlador de cache, transfere os dados mais requisitados da RAM para a memória cache.

2. Analise as dissertações a seguir, considerando um mesmo processador em 2 arquiteturas distintas: RISC e CISC.

A. O comando ADD na arquitetura CISC utiliza 5 ciclos de execução e 1 instrução. Na arquitetura RISC utiliza 1 ciclo de execução e 3 instruções. Qual destas arquiteturas tem melhor performance? Explique.

R: RISC, pois essa arquitetura tem um conjunto de instruções reduzido, ou seja, é reduzido para executar instruções mais simples e comuns como uma instrução de soma (ADD), podendo executar 3 instruções em apenas um ciclo de execução.

B. O comando MOV na arquitetura CISC utiliza 3 ciclos de execução e 1 instrução. Na arquitetura RISC utiliza 1 ciclo de execução e 3 instruções. Qual destas arquiteturas tem melhor performance? Explique.

R: RISC, pois em apenas um ciclo de execução foi capaz de executar 3 instruções.

C. Um comando XXX na arquitetura RISC utiliza 1 ciclo de execução e 4 instruções. Como deve ser a implementação deste mesmo comando em uma arquitetura CISC para ser equivalente em termos de processamento? Explique.

Pós - aula 04

1. Pesquise os dispositivos de memória não-volátil a seguir:

R:

Hard Disk Drive (HDD): O disco rígido é considerado o principal meio de armazenamento de dados em massa. Nos sistemas recentes ele é também utilizado para expandir a memória RAM, através da memória virtual.

Solid State Drive (SSD): Dispositivo sem partes móveis para armazenamento não volátil de dados digitais.

2. Descreva suas características e funcionalidades técnicas.

R:

Hard Disk Drive (HDD): Os discos magnéticos de um disco rígido são recobertos por uma camada magnética extremamente fina; laminada (plated media), mídia mais densa, de qualidade superior. A cabeça de r/w de um disco rígido funciona como um eletroímã composta de uma bobina de fios que envolve um núcleo de ferro; dispositivo este extremamente pequeno e preciso, a ponto de ser capaz de gravar trilhas medindo menos de um centésimo de milímetro de largura.

Solid State Drive (SSD): Construídos em torno de um circuito integrado semicondutor que é responsável pelo armazenamento, de maneira diferente dos sistemas magnéticos ou óticos. Alguns dispositivos usam memória RAM e outros memória flash, mais comum e mais barato. Não têm partes móveis e, portanto, procura e outros atrasos inerentes de discos eletro-mecânicos convencionais são insignificantes. Não produzem calor como os HDDs tradicionais e também são mais resistentes a impactos.

3. Monte uma tabela comparativa com suas principais diferenças.

Característica	SSD	HDD
Tempo de acesso randômico	0.1 a 0.3ms	5 a 10ms
Latência de leitura	Baixa pois a leitura é direta de qualquer local do disco	Alta pois requer o tempo de posicionamento do leitor no local correto.
Desfragmentação	Não requer pois a leitura de qualquer local do disco é rápida	Requer desfragmentação contínua para ter melhor rendimento
Ruído	Não produz ruído durante o funcionamento	As partes que se movimentam durante o funcionamento produzem ruído
Fatores Externos	Não é sensível a choque, altitude, vibração, magnetismo	Sensível à choque, altitude, vibração e magnetismo (o último pode danificar arquivos)
Custos	O preço por GB de espaço é alto, já o consumo de energia é bastante baixo	Preço por GB de espaço é baixo, consumo de energia alto
Capacidade	A grande maioria dos SSDs comercializados atualmente (2015) é de 64GB à 480GB	Capacidade alta é comum, exemplares com 2TB são comercializados a preços acessíveis
Longevidade	Possuem limitação de ciclos de escrita (em geral de 1 à 5 milhões de ciclos dependendo da tecnologia)	Não possuem limites de escrita, isso só ocorre em caso de defeito na trilha ou setor

Pós - aula 05

1. Dada as memórias Cache e RAM a seguir:

Endereço Cache	Bit	Flag	Valor
000			
001			
010			
011			
100			
101			
110			
111			

Endereço RAM	Bit	Flag	Valor
0000	1	00	B
0001	1	01	C
0010	1	10	D
0011	1	11	E
0100	1	10	A
0101	1	11	A
0110	1	01	A
0111	1	10	E
1000	1	10	B
1001	1	11	A
1010	1	10	C
1011	1	11	E
1100	1	01	D
1101	1	10	C
1110	1	11	B
1111	1	10	B

Mostre os resultados das ações a seguir (HIT / FAULT) e as memórias finais resultantes, das ações dos comandos READ(<<ADDRESS>>) e WRITE(<<ADDRESS>>,<<CHAR>).

```

READ(0000)
READ(1010)
READ(1111)
WRITE(0010, 'A')
WRITE(1101, 'B')
READ(0100)
READ(0110)
WRITE(1110, 'C')
```

```

READ(1011)
READ(1110)
WRITE(1110, 'D')
WRITE(0001, 'E')
READ(1100)
READ(0111)
WRITE(0110, 'A')
READ(1111)
```

R:

	END	BIT	FLAG	VL
A. R 0 000	0000	1	00	B
B. R 1 010	0001	1	01	C
C. R 1 111	0010	1	10	D
D. W 0 010.A	0011	1	11	E
E. W 1 101.E	0100	1	10	A
F. R 0 100	0101	1	11	A
G. R 0 110	0110	1	01	A
H. W 1 110.C	0111	1	10	E
I. R 1 011	1000	1	10	B
J. R 1 110	1001	1	11	A
K. W 1 110.D	1010	1	10	C
L. W 0 001.E	1011	1	11	E
M. R 1 100	1100	1	01	D
N. R 0 111	1101	1	10	C
O. W 0 110.A	1110	1	11	D
	1111	1	10	B

2. Dada a linguagem PUPILO++ a seguir:

Comando	Parâmetros		Descrição
BEG			inicia a execução do programa.
ADD	X	Y	soma o valor do endereço Y ao endereço X e armazena-o no endereço X.
SUB	X	Y	subtrai o valor do endereço Y ao endereço X e armazena-o no endereço X.
INC	X		incrementa em 1 o valor do endereço X e armazena-o no endereço X.
DEC	X		decrementa em 1 o valor do endereço X e armazena-o no endereço X.
SET	X	Y	atribui o valor do endereço Y ao endereço X.
END			finaliza a execução do programa.

A memória CACHE L1 a seguir:

Endereço Cache	Bit	Flag	Valor
0000	1	10	21
0001	0	10	16
0010	1	11	49
0011	0	01	42
0100	1	00	21
0101	1	11	28
0110	0	10	35
0111	0	11	3
1000	1	01	29
1001	1	00	7
1010	0	10	17
1011	1	10	2
1100	1	11	20
1101	1	01	22
1110	0	00	1
1111	0	11	47

E a memória RAM a seguir:

Endereço RAM	Bit	Flag	Valor
000000	1	00	47
000001	1	01	45
000010	1	10	41
000011	1	11	11
000100	1	10	9
000101	1	11	17
000110	1	01	34
000111	1	10	48
001000	1	11	8
001001	1	00	45
001010	1	01	1
001011	1	10	36
001100	1	11	39
001101	1	10	17
001110	1	11	31
001111	1	01	35
010000	1	10	14
010001	1	11	40
010010	1	10	42
010011	1	11	24
010100	1	01	31
010101	1	10	43
010110	1	11	7
010111	1	00	39
011000	1	01	7
011001	1	10	8
011010	1	11	34
011011	1	10	19
011100	1	11	45
011101	1	00	43
011110	1	01	7
011111	1	10	48

Endereço RAM	Bit	Flag	Valor
100000	1	11	44
100001	1	10	10
100010	1	11	18
100011	1	01	27
100100	1	10	17
100101	1	11	47
100110	1	00	17
100111	1	01	22
101000	1	10	21
101001	1	11	36
101010	1	10	39
101011	1	11	37
101100	1	01	14
101101	1	10	8
101110	1	11	48
101111	1	10	14
110000	1	11	48
110001	1	01	8
110010	1	10	29
110011	1	11	36
110100	1	00	22
110101	1	01	27
110110	1	10	43
110111	1	11	12
111000	1	10	30
111001	1	11	50
111010	1	00	9
111011	1	00	0
111100	1	00	37
111101	1	00	44
111110	1	10	1
111111	1	11	37

a) R:
 BEG SET [010111], [101010]
 ADD [010111], [010011]
 END [111111], [1110101]

b) R:
 BEG SET [001010], [011101]
 ADD [000010], [001001]
 SET [000010], [000000]
 END [000000], [000000]

c) R:

BEG

SET [111111], [100001]

SET [010111], [100001]

INC [010111]

DEC [111111]

SUB [010111], [111111]

SET [100000], [010111]

END (010111), (100000)