



Universidade de São Paulo - São Carlos, SP

Instituto de Ciências Matemáticas e de Computação

USP – ICMC – SSC

SSC0610 - Organização de Computadores

Professor responsável: *Fernando Santos Osório*

Semestre: 2010/2

Horário: Seg. 10h / Qui. 16h

E-mail: fosorio@icmc.usp.br

fosorio@gmail.com

Web: <http://www.icmc.usp.br/~fosorio/>

LISTA DE EXERCÍCIOS - Nro. 01

[Arquiteturas Clássicas]

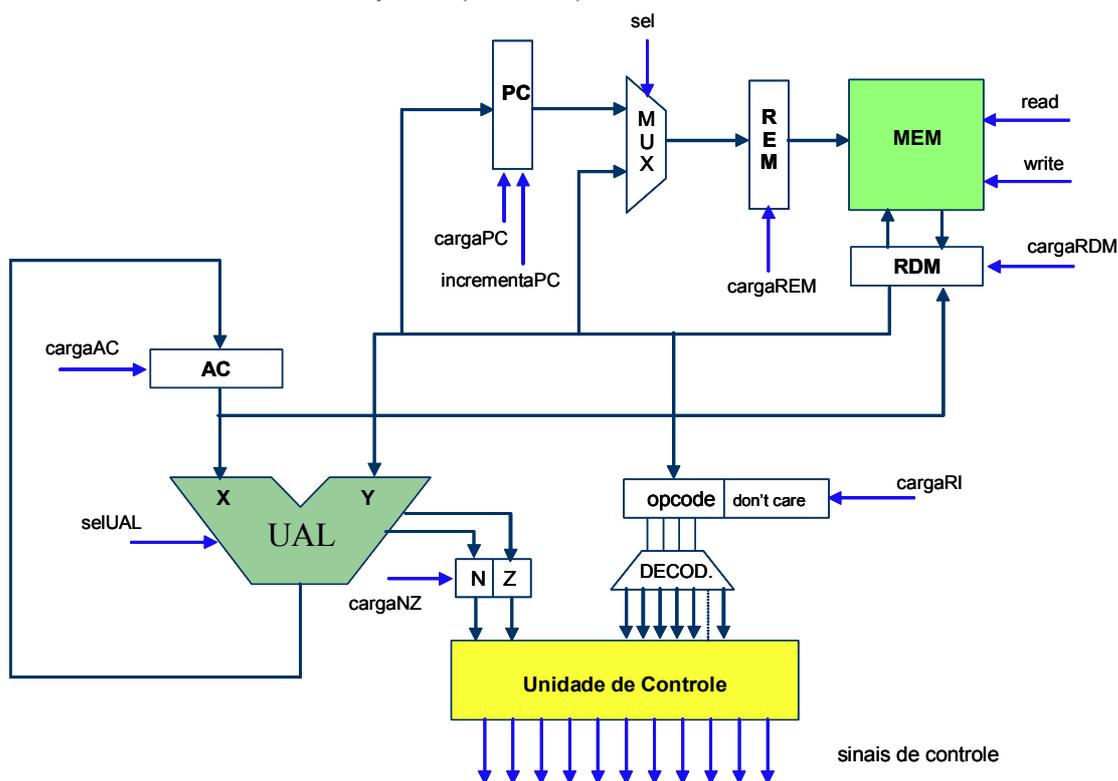
1. Descreva as principais características e compare as **Arquiteturas de Von Neumann e de Harvard**. Quais as vantagens e desvantagens de cada uma delas? Em relação aos equipamentos disponíveis no mercado atualmente (PCs, Notebooks, Video-Games, etc), indique um exemplo de equipamento e classifique em que tipo de arquitetura (Von Neumann ou Harvard) ele se enquadra.
2. Quais os principais componentes de uma arquitetura de computador do tipo Von Neumann? Dentre estes componentes, indique quais os elementos e propriedades (capacidade) de cada um que é usada para definir uma família/geração de microprocessadores (Exemplo: Intel 4004, Intel 8080, Intel 8086, Mostech 6502, Motorola 68000). Em outras palavras, “o que tem dentro de cada um destes processadores (componentes de uma Arquitetura de Von Neumann) que os caracteriza como pertencentes a uma família de processadores”?
3. Complete a tabela abaixo com os dados referentes a cada processador:

Característica/Processador	Neander*	Ramses*	4004	6502	Z80**	8086**
Fabricante						
Dados (nro. de bits)						
Endereços (nro. de bits)						
Clock						
Nro. de Instruções						
Flags disponíveis						
Registradores						
Modos diferentes de Endereçamento						
Tipos de E/S						

* **Neander** e **Ramses**: Computadores hipotéticos definidos por Raul Weber no Livro “Fundamentos de Arquitetura de Computadores” (2001)

** No caso dos microprocessadores não estudados em aula, procure em outras fontes (na Internet) estas informações.

- Quais os modos de endereçamento disponíveis no microprocessador Mostech 6502? Liste os modos de endereçamento do 6502, descrevendo as principais características de cada um deles, e dando um exemplo de instrução para cada um destes modos de endereçamento.
- Em relação a ULA: como é obtido o Flag de Carry em operações da ULA de um processador? Quais as instruções típicas que afetam o Flag de Carry? Descreva um exemplo de aplicação em um programa onde o uso do Carry é necessário.
- Em relação ao processador Neander: observe a figura abaixo que representa a arquitetura deste processador, e baseado nesta figura **descreva a execução passo-a-passo** da seguinte instrução, que se encontra armazenada no endereço \$06 (PC = \$06) da memória: LDA \$50 => \$06: \$20 \$50



- Conforme o exercício acima (nro. 6), e baseado na mesma figura, **descreva a execução passo-a-passo** da seguinte instrução, que se encontra armazenada no endereço \$10 (PC = \$10) da memória de uma máquina baseada no processador do Neander: JZ \$40 => \$10: \$AO \$40
- Considerando o processador Neander, é possível realizar uma adição de um valor de 16 bits? Se você considera que não é possível explique porque. Se você considera que é possível, explique como você faria esta operação.
- Considerando a arquitetura do processador Neander, como você implementaria as seguintes novas instruções indicadas a seguir. Descreva em termos de operações de transferência de registradores (RTL) e da seqüência de sinais de controle que teriam que ser ativados para executar tais instruções:
 - Load A com modo de endereçamento indireto: LDA (\$XX)
 - Store A com modo de endereçamento indireto: STA (\$XX)
 - Add A com modo de endereçamento imediato: ADD #\$XX
 - Store PC com modo de endereçamento direto: SPC \$XX [No caso desta instrução, quais seriam as modificações necessárias na arquitetura do processador Neander?]

10. Considerando a arquitetura do processador Neander e de seu conjunto de instruções, implemente os seguintes programas em linguagem de montagem:
- Procurar por um valor (armazenado no endereço \$A0) em um vetor de comprimento de 32 bytes que está armazenado a partir do endereço \$80.
 - Ler 2 valores armazenados nos endereços \$80 e \$81. Se ambos forem positivos, realizar a soma destes dois valores. Se ambos forem negativos (complemento de 2), retirar o sinal deles considerando-os como valores positivos e realizar a soma destes dois valores. Se um for positivo e outro negativo, o resultado a ser obtido é o do maior valor (positivo). Armazenar o resultado no endereço \$82.
 - Ler 3 valores armazenados nos endereços \$80, \$81 e \$82. Salvar estes 3 valores ordenados na memória, nos endereços \$90 (menor valor), \$91 (valor intermediário) e \$91 (maior valor).
11. Considerando um processador hipotético Ozorium (fictício), definido como um processador cujo registrador PC (Program Counter) possui 12 bits e cuja memória seja organizada em bancos de N posições de 12 bits (palavras de 12 bits). Qual seria o tamanho máximo da memória de programa que poderia se endereçar com este processador? Ou seja, quantas palavras de 12 bits poderiam ser armazenadas nesta memória?
12. Considerando novamente o processador hipotético Ozorium, e sabendo que suas instruções são também de 12 bits (RI – Registrador de Instrução de 12 bits), qual o tamanho máximo do Conjunto de Instruções (Instruction Set) deste processador?
13. Defina e descreva as seguintes siglas e termos relacionados com a disciplina de Arquitetura de Computadores, relacionando em qual(is) componente(s) da Arquitetura de Von Neumann são usados:
- 1) Half-adder; 2) SR Latch, D Latch; 3) IR; 4) SP; 5) IRQ, INTR, NMI; 6) WR/RD Enable;
 - 7) OPCode; 8) NOP; 9) BusReq; 10) DMA; 11) Address Bus, Data Bus, Control Bus; 12) DRam, SRam; 14) EDVAC, IAS; 15) CLK.
14. O que é o *pipeline* de execução de uma instrução (passos da execução de uma instrução)? Liste e descreva as principais etapas do ciclo de execução de uma instrução em uma CPU.
15. Como funciona o modo (métodos) de E/S do tipo IO mapeado em memória (usados no 6502)? Descreva este modo, indicando suas principais características, apresentando um exemplo.
16. Os processadores mais modernos estudados na disciplina, como o 6502, Z80, 8086 e 68000, são caracterizados por transições em termos de sua capacidade de armazenamento de dados e instruções. Em relação ao barramentos de dados e endereços e registradores internos, qual era a capacidade (nro. de bits) de cada um destes processadores? [Pesquisar e preencher a tabela abaixo]

Nro. de Bits	6502	Z80	8088	8086	68000
Barramento de Dados					
Barramento de End.					
Registrador PC					
Registrador Acumulador					

[Questões de Programação do 6502]

17. Faça um programa em linguagem de montagem para o 6502 para adicionar 2 valores de 16 bits. O programa deve iniciar sua execução no endereço \$1000 da memória, e os dados devem estar armazenados nos endereços \$2000 (MSB) e \$2001 (LSB) para o Valor1 e \$2002 (MSB) e \$2003 (LSB) para o Valor2. O resultado deve ser armazenado no endereço \$3000 (MSB) e \$3001 (LSB).
MSB = *Most Significant Byte* / LSB = *Least Significant Byte*

18. Faça um programa em linguagem de montagem para o 6502 para subtrair 2 valores de 16 bits. O programa deve iniciar sua execução no endereço \$0500 da memória, e os dados devem estar armazenados na pilha (stack). O Valor2 (16 bits) deve ser subtraído do Valor1 (16 bits). O Valor1 está armazenado em baixo na pilha, com o MSB mais acima na pilha e o LSB mais abaixo na pilha. O Valor2 está armazenado no topo da pilha, acima do Valor1, com o MSB mais acima na pilha e o LSB mais abaixo na pilha. O resultado deve ser armazenado no endereço \$1000 (MSB) e \$1001 (LSB). MSB = *Most Significant Byte* / LSB = *Least Significant Byte*
19. Faça um programa em linguagem de montagem para o 6502. Considere que existe uma função de I/O mapeado em memória para escrita na tela do *display* do computador. O display é capaz de exibir até 32 caracteres, onde basta escrever na memória de I/O o código ASCII, escrevendo assim o caracter nos endereços correspondentes a cada posição do display. O endereço de base do display é \$E000 (primeiro caracter), sendo que o último caracter do display está mapeado na posição \$E01F. Faça um programa que escreva a mensagem “Hello world!” na tela do display.
20. Faça um programa em linguagem de montagem para o 6502, baseado no programa anterior. Faça uma sub-rotina que exiba a mensagem “Hello world!” e uma outra subrotina que limpe o visor do display, ou seja, escreva espaços em branco em todas as 32 posições do display. Faça um programa exiba a mensagem “hello” de modo que esta fique piscando na tela de display. O programa deve alternar entre exibir a mensagem “hello”, pausar (delay) um certo tempo, limpar o display, pausar um certo tempo, e repetir o ciclo novamente (mensagem, pausa, limpa, pausa) de modo indefinido. Observações:
- O programa deve ser composto de pelo menos 3 sub-rotinas (exibe mensagem , limpa tela, pausa);
 - A rotina de pausa pode ser implementada com um grande “loop” de instruções que não fazem nada (faça um contador que repita a execução do laço de instruções “gasta tempo” como o NOP).
21. Faça um programa em linguagem de montagem para o 6502. Este programa deve soma um vetor de bytes armazenado a partir do endereço \$8000, sendo que o tamanho deste vetor é indicado por um byte (Max. 255 elementos) armazenado no endereço \$8100. A soma deve ser realizada considerando um vetor de dados de 8 bits, mas no entanto o resultado final da soma deve considerar um valor de 16 bits (acumular os valores de 8 bits em uma somador de 16 bits). Armazenar o resultado final da soma no endereço \$8101 e \$8102.
22. Faça um programa em linguagem de montagem para o 6502. Este programa ler 3 valores de 16 bits, ordená-los e armazená-los em ordem crescente. Os 3 valores estão armazenados a partir do endereço \$8000 da memória, sendo que os valores ordenados devem ser armazenados a partir do endereço \$8010 da memória.
23. Resolva os exercícios discutidos em aula de programação do 6502.