

Bacharelado em Engenharia de Software

Disciplina: 15INP – Introdução à Programação de Computadores

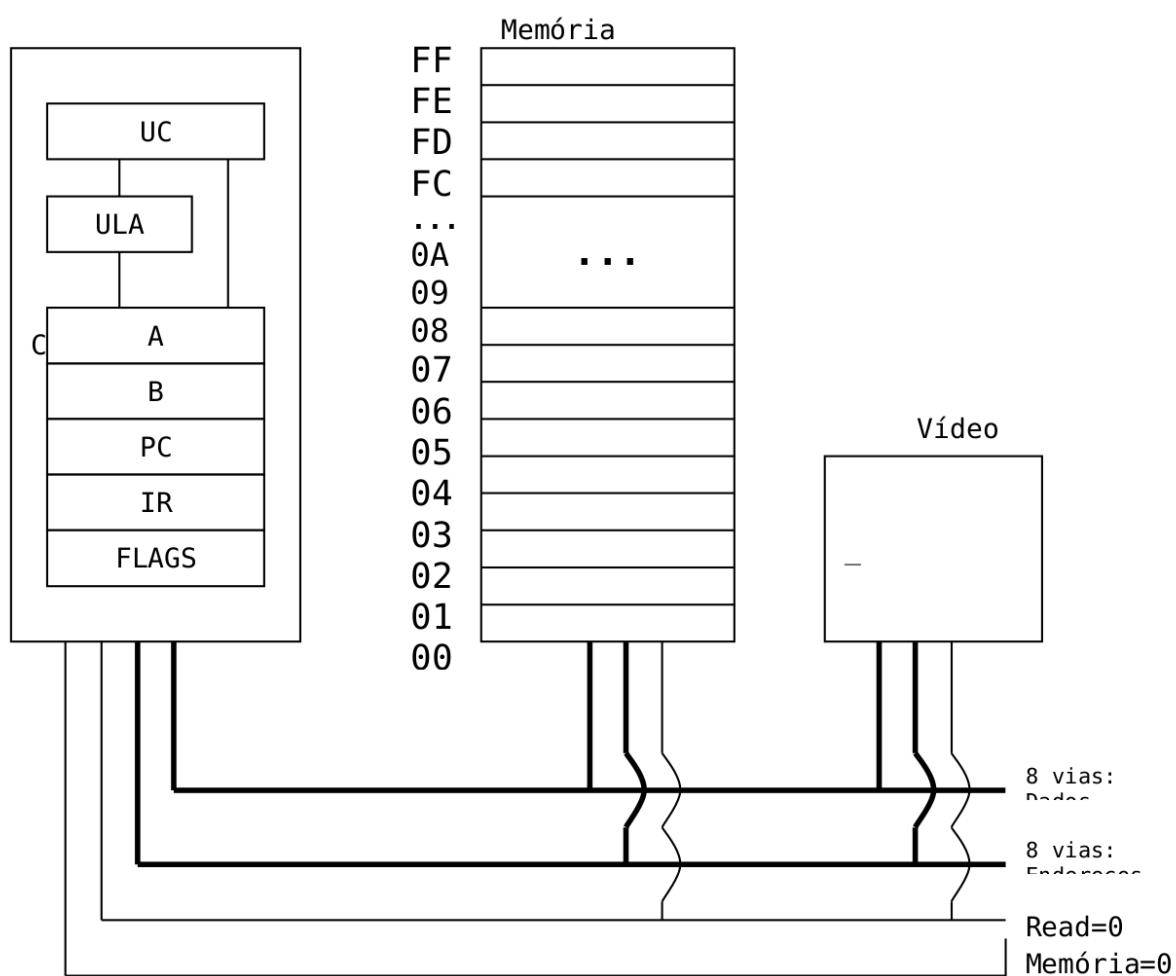
Prof. Paulo Roberto Farah

Trabalho Prático

Observações:

- Penalização por atraso de 10% ao dia, inclusive sábados e domingos
- O trabalho deve ser feito individualmente ou em grupos de no máximo três alunos
- Cópias ou plágios receberão nota zero

Simulador de um Computador Simplificado



O programa deverá apresentar graficamente o estado da CPU, da memória e dos barramentos a cada execução de instrução. O barramento de dados e de endereços deverão ser representados através de uma linha simples, com um indicativo do valor de cada via. O usuário irá controlar o clock via teclado. Supor que cada transferência entre a memória ou periférico e a CPU se completa em 1 ciclo de clock. A representação gráfica de memória deverá apresentar as 10 posições de memória seguintes a partir do valor apontado pelo registrador PC. O valor inicial do PC deve ser 0.

Sequência de busca de Instruções:

1. A UC busca a próxima instrução da memória para o IR;
2. A UC atualiza o contador de programa para que ele aponte para instrução seguinte ($PC = PC + 1$)
3. A UC decodifica a instrução;
4. Se a instrução utiliza dados da memória, determina onde eles estão (ex. instrução 005 faz referência a um endereço de memória). Neste caso, incrementa o contador de programa ($PC = PC + 1$)
5. Executa a instrução

Cada célula de memória possui um Byte e deve ser representada por um vetor de inteiros. A memória total do computador simulado será de 256 bytes . Todos os registradores da CPU possuem 8 bits. Os barramentos de dados e de endereços possuem 8 linhas de sinal e o barramento de endereços possui 8 vias cada. O simulador possui dois sinais de controle: memória/periférico e leitura/escrita, como ilustra a Figura 1. O valor 0 deve ser transmitido por esses barramentos quando a CPU realizar uma operação de leitura com a memória. O computador possui apenas um periférico, que é um controlador de vídeo, localizado no endereço 01. Todas as informações escritas neste periférico terão o seu correspondente em ASCII impressos no vídeo.

Exemplo: Caso seja escrito 01 hexa (binário 00000001) no barramento de endereços e 01 41 hexadecimal (binário 01000001, decimal 65) no barramento de dados , será impressa a letra 'A' no vídeo. Neste caso, o sinal de controle memória/periférico deverá estar setado com 1 para indicar que os dados e endereços devem ser lidos pelo periférico e o sinal de controle leitura/escrita deverá estar com 1 para indicar uma operação de escrita.

2. INSTRUÇÕES

00 - MOV A, VALOR: Transfere o conteúdo da posição de memória seguinte (endereço imediato) para o registrador A.

01 - MOV B, VALOR: Transfere o conteúdo da posição de memória seguinte (endereço imediato) para o registrador B.

02 - MOV A, B: Transfere o conteúdo de B para A.

03 - MOV B, A: Transfere o conteúdo de A para B.

04 - IN ENDEREÇO: Lê do endereço de periférico indicado e coloca o resultado em A (deve escrever 1 no seletor MEMÓRIA/PERIFÉRICO, o ENDEREÇO no barramento de endereços, e escrever 0 no seletor READ/WRITE).

05 - OUT ENDEREÇO: Escreve o conteúdo do registrador A no endereço indicado (deve escrever 1 no seletor MEMÓRIA/PERIFÉRICO, o ENDEREÇO no barramento de endereços, e escrever 1 no seletor READ/WRITE)

2.1. Instruções Aritméticas (implementadas pela ULA):

Obs. Todas as operações da ULA afetam o estado do registrador FLAGS

20 – SUB A, VALOR: Realiza a operação $(A - \text{VALOR})$, alterando o valor do registrador FLAGS. O valor estará na próxima posição de memória.

21 – ADD A, VALOR: Realiza a operação $(A + \text{VALOR})$, alterando o valor do registrador FLAGS. O valor estará na próxima posição de memória.

22 – MUL A, VALOR: Realiza a operação $(A * \text{VALOR})$, alterando o valor do registrador FLAGS. O valor estará na próxima posição de memória.

23 – DIV A, VALOR: Realiza a operação (A / VALOR) , alterando o valor do registrador FLAGS. O valor estará na próxima posição de memória.

24 – SUB A, B: Solicita à UCP para realizar a operação de subtração entre A e B $(A - B)$, com resultado armazenado no registrador A. Esta operação também modifica o conteúdo do registrador FLAGS.

25 – ADD A, B: Solicita à UCP para realizar a operação de adição entre A e B $(A + B)$, com resultado armazenado no registrador A.

26 – MUL A, B: Solicita à UCP para realizar a operação de multiplicação entre A e B $(A * B)$, com resultado armazenado em A

27 – DIV A, B: Solicita à UCP para realizar a operação de divisão entre A e B (A / B) , com resultado armazenado em A.