

Trabalho prático - Arquitetura de Computadores

- Implementação de Simuladores para circuitos programáveis

1- Projetar simuladores de modelos Parte Operativa/Parte de controle para arquiteturas de conjuntos de instruções. As arquiteturas devem ser compatíveis com o conjunto de instruções dos processadores Neander, Cesar e Ramsés. A implementações podem ser em C++ ou Java.

2- Implementar, utilizando os respectivos conjuntos de instruções dos processadores, os algoritmos:

2.1- Somar 2 variáveis de 8 bits;

2.2- Somar 2 variáveis de 16 bits (somente Ramses e Cesar);

2.3- Subtrair 2 variáveis de 8 bits;

2.4- Chamar o programa 2.2 como uma sub-rotina por um programa principal (somente Ramses e Cesar); a sub-rotina deve receber os operandos fonte como parâmetros e retornar o resultado;

2.5- Determinar o maior valor armazenado em um vetor de 32 posições.

3- Os simuladores devem executar cada uma das instruções de cada algoritmo, mostrando na tela ou em arquivo de texto, o valor de cada

componente do processador à cada instrução executada. Os componentes a serem mostrados durante a simulação são (pelo menos):

- contador de programa;
- registrador de instrução;
- flags de controle;
- banco de registradores;
- memória de dados; e
- operação executada na ULA.

Ao final da simulação, o simulador deve escrever na saída (tela e/ou arquivo) o tempo de execução de cada algoritmo em ciclos de relógio, considerando os itens 4.3.1 e 4.3.2 abaixo.

4- Fazer um relatório explicando a implementação de cada componente, bem como de suas conexões. No relatório deve aparecer uma figura da PO e da PC explicando/exemplificando como seus componentes internos executam cada uma das instruções do conjunto de instruções do processador.

Ainda, no relatório deve aparecer uma análise das execuções, em função das características dos processadores. Para tanto se deve:

4.1- justificar o tamanho dos programas (em número de instruções);

4.2- comentar sobre facilidades/dificuldades de programação;

4.3- comparar desempenho em ciclos de relógio, considerando:

4.3.1- 1 ciclo para instruções lógico/aritméticas; e

4.3.2- 3 ciclos para instruções que

acessam a memória: instruções de acesso à memória ou que usam modo de endereçamento de operandos que acessa a memória.

5- Etapas do projeto (para cada processador):

5.1- projetar a PO; diagrama de blocos e implementar uma classe para cada bloco;

5.2- projetar a PC: a PC deve ter funções para ler as instruções da memória, decodificar as instruções e, para cada instrução decodificada, chamar as funções que a executam na PO (isso equivale à geração das microinstruções);

5.3- escrever os algoritmos a serem testados em arquivo de texto;

5.4- ler o arquivo com o código e colocar as instruções na memória de código;

5.5- rodar a simulação do código e escrever as saídas em arquivo/tela; e

5.6- analisar os resultados e comparar os processadores (como explicado nos itens 4.1, 4.2 e 4.3 acima).

6- Entregar na tarefa do SIGAA:

- o relatório;
- um arquivo contendo os códigos fonte; e
- um vídeo apresentando os diagramas de bloco da PO/PC, explicando como funciona o simulador e exemplificando a execução de pelo menos 2

algoritmos.

7- O prazo para a entrega é o dia 14/01/25.

8- O trabalho pode ser feito em grupo de até 3 alunos.