Universidade de Brasília
Departamento de Ciência da Computação

Disciplina: CIC 116394 – Organização e Arquitetura de Computadores – Turma A 2016/1
Prof. Marcus Vinicius Lamar
Equipes de até 5 pessoas.
Entrega do relatório (pdf) e fontes em um arquivo zip pelo Moodle até dia 16/05/2016

Laboratório 2 – PARTE A
- Assembly MIPS Aritmética Fracionária –

Objetivos:

- Treinar o aluno com o Simulador/Montador MARS;
- Desenvolver a capacidade de codificação de algoritmos em linguagem Assembly MIPS;
- Desenvolver a capacidade de análise de desempenho de algoritmos em Assembly;

(3.0) 1) Cálculo das raízes da equação de segundo grau usando operações em ponto flutuante precisão dupla.

Dado o polinômio de segundo grau: $a.x^2 + b.x + c = 0$

a) (1.0) Escreva um procedimento int baskara(double a, double b, double c) que retorne 1 caso as raízes sejam reais e 2 caso as raízes sejam complexas conjugadas, e coloque na pilha os valores das raízes no formato IEEE 754 Precisão Dupla.

b) (1.0) Escreva um procedimento void show(int t) que receba o tipo (t=1 raízes reais, t=2 raízes complexas) e apresente na tela as raízes, conforme os modelos abaixo, que estão na pilha retirando-as da mesma:

Para raízes reais:
R(1)=1234.0000
R(2)=5678.0000

Para raízes complexas:
R(1)=1234.0000 + 5678.0000 i
R(2)=1234.0000 – 5678.0000 i

c) (1.0) Escreva as saídas obtidas para os seguintes polinômios [a, b, c]:
c.1) [1, 0, -9.86960440] c.2) [1, 0, 0] c.3) [1, 99, 2459] c.4) [1, -2468, 33762440] c.5) [0, 10, 100]

(7.0) 2) Cálculo das raízes da equação de segundo grau usando operações em ponto fixo de 32 bits.

Dada a equação de segundo grau: $a.x^2 + b.x + c = 0$

a) (3.0) Escreva um procedimento int baskara(int a, int b, int c) que retorne 1 caso as raízes sejam reais e 2 caso as raízes sejam complexas conjugadas, e coloque na pilha os valores das raízes no formato ponto fixo com um Q adequado.

b) (3.0) Escreva um procedimento void show(int t) que receba o tipo (t=1 raízes reais, t=2 raízes complexas) e apresente na tela as raízes, conforme os modelos abaixo, que estão na pilha retirando-as da mesma:

Para raízes reais:
R(1)=1234.0000
R(2)=5678.0000

Para raízes complexas:
R(1)=1234.0000 + 5678.0000 i
R(2)=1234.0000 – 5678.0000 i

c) (1.0) Escreva as saídas obtidas para os seguintes polinômios [a, b, c]:
c.1) [1, 0, -9.86960440] c.2) [1, 0, 0] c.3) [1, 99, 2459] c.4) [0, 10, 100]

Dicas:
Para o item a) você vai precisar implementar procedimentos para as operações de divisão e raiz quadrada em ponto fixo.
Para o item b) você vai precisar implementar procedimentos que convertam ponto fixo para ponto flutuante.
Para o item c) os polinômios podem ser entrados já no formato adequado em ponto fixo ou em float.

Universidade de Brasília
Departamento de Ciência da Computação

Disciplina: CIC 116394 – Organização e Arquitetura de Computadores – Turma A 2016/1
Prof. Marcus Vinicius Lamar

Laboratório 2 – PARTE B
– ULA e FPULA –

Objetivos:

- Treinar ao aluno com a Linguagem de Descrição de Hardware Verilog;
- Familiarizar o aluno com a plataforma de desenvolvimento FPGA DE2 da Altera e o software QUARTUS-II;
- Desenvolver a capacidade de análise e síntese de sistemas digitais usando HDL;

1) (5.0) Unidade Lógico Aritmética de Inteiros:

a) (0.0) Para a ULA MIPS32 fornecida, descreva suas funções e levante a tabela verdade de seus códigos de operação.

b) (1.0) Verifique cada operação por simulação de forma de onda.

c) (1.5) Sintetize esta ULA na placa DE2-70 e verifique, em tempo real, cada operação implementada usando a ferramenta SignalTap Analyser. Filme o experimento e coloque no YouTube.

d) (1.5) Levante os requisitos físicos necessários, isto é, número de elementos lógicos, tempos envolvidos e máxima frequência de clock utilizável para cada operação da ULA.

e) (1.0) Retire as operações de divisão e módulo. Sintetize novamente e levante os novos requisitos físicos da ULA. Comente.

2) (5.0) Unidade Aritmética de Ponto Flutuante:

a) (0.0) Para a FPULA MIPS fornecida, descreva suas funções (MegaWizard Plug-In Manager Edit) e levante a tabela verdade de seus códigos de operação.

b) (1.0) Verifique cada operação por simulação de forma de onda.

c) (1.5) Sintetize esta FPULA na placa DE2-70 e verifique, em tempo real, cada operação implementada usando a ferramenta SignalTap Analyser. Filme o experimento e coloque no YouTube.

d) (1.5) Levante os requisitos físicos, isto é, número de elementos lógicos necessários, número de ciclos, tempos envolvidos e máxima frequência de clock utilizável para cada operação da FPULA.

e) (1.0) Qual a operação que mais impacta em cada requisito físico?