

Segunda Lista de Exercícios
Projeto de Sistemas em Chip
Data de entrega: 05 de junho de 2019 às 23:50

Instruções: A lista de exercícios é individual. Enviar no repositório todos os arquivos necessários para replicar o projeto no Vivado e no SDK (arquivos das pastas .src e .sdk, arquivos Matlab/Octave e arquivos .txt com vetores de teste). Adicionalmente deve ser enviada uma folha de dados em PDF, disponível no repositório da disciplina.

Exercício 1 (5 pontos). Co-processador FPadd

a) (1 ponto) Utilize o código do somador em ponto flutuante (FPadd) de 27 bits (disponível no git do curso) para encapsular o IP usando o barramento AXI. Use três (3) registradores auxiliares: dois para envio dos operandos de entrada e um para receber o resultado da soma. Siga como exemplo o código do IP AXI da multiplicação em ponto flutuante realizado em sala de aula.

b) (2 pontos) Implemente um processador Microblaze com configurações típicas, 64KB de memória principal, 8KB de memória cache. Através da conexão AXI, conecte os switches e leds da placa de desenvolvimento, um módulo de comunicação serial UART, um AXI timer e o IPAXI do somador em ponto flutuante (FPadd). Faça um print do *block design* e caracterize o circuito após a implementação (processo PAR). Obtenha os reportes de consumo de recursos, reporte de consumo de energia, reporte de timing e estime a frequência máxima de operação. Apresente os valores na folha de dados.

c) (2 pontos) Desenvolva um aplicativo em C que permita enviar para o co-processador valores aleatórios dos operandos de entrada cada vez que os switches mudarem de valor. Imprima nos leds o resultado da soma, para isso use um switch para selecionar os 16 MSB ou 16 LSB do resultado. Adicionalmente, imprima pelo terminal o resultado obtido usando a comunicação serial. Apresente um *print* do terminal mostrando o resultado e coloque na folha de dados.

Exercício 2 (5 pontos). Co-processador RNA

a) (2 ponto) Modifique o HDL da rede neural artificial 4-4-3 desenvolvida em sala de aula de forma a usar os pesos de conexão obtidos pelo algoritmo de treinamento no Matlab. Use o algoritmo FNNPSOGSA.m e a função float2bin.m disponíveis no git do curso para obter os pesos de conexão.

b) (1 ponto) Encapsule o IP usando o barramento AXI. Para isso siga o exemplo feito em sala de aula no qual foram usados 7 registradores auxiliares: 4 para enviar as entradas da rede neural e 3 para as saídas da rede neural.

c) (1 ponto) Implemente um processador Microblaze com configurações típicas, 64KB de memória principal, 8KB de memória cache. Através da conexão AXI, conecte um módulo de comunicação serial UART, um AXI timer e o IP da rede neural em ponto flutuante. Faça um print do *block design* e caracterize o circuito após a implementação (processo PAR). Obtenha os reportes de consumo de recursos, reporte de consumo de energia, reporte de timing e estime a frequência máxima de operação. Apresente os valores na folha de dados.

d) (1 ponto) Desenvolva um aplicativo em C que permita enviar ao co-processador as entradas da rede neural e receber as três saídas. Imprima pelo terminal usando a comunicação serial as saídas obtidas. Apresente um *print* do terminal mostrando o resultado e coloque na folha de dados.

Bom trabalho!