Relatório Sistemas Digitais Avançados

Mathaus C. Huber

[1]Universidade Federal de Pelotas (UFPEL) – Discente do Curso Superior de Ciência da Computação
R. Gomes Carneiro, 1 - Centro – 96075-630 – Pelotas – RS – Brazil

mchuber@inf.ufpel.edu.br

Abstract. *This article describes the report of the third practical task of the discipline of advanced digital systems, optional discipline, given to the ninth semester of the computer science course, at the Federal University of Pelotas, which refers to the creation of two arithmetic logical units (ALU) with the Quartus II tool, with the tests with the ModelSim tests or the ModelSim Testench.*

Resumo. *Este artigo descreve o relatório da terceira tarefa prática da disciplina de Sistemas Digitais Avançados, disciplina optativa, conferida ao nono semestre do curso de Ciência da computação, da Universidade Federal de Pelotas, no qual se refere a criação de duas unidades lógicas aritméticas (ULA) com a ferramenta Quartus II, juntamente com os testes desenvolvidos com o TesteBench do ModelSim.*

1. Informação Geral

Para a tarefa 2 da disciplina de Sistemas Digitais Avançados, era necessário fazer a criação de duas ULAS + banco de registradores. A minha ideia inicial era criar uma ULA contendo todas as operações necessárias entre as duas ulas junto com um banco de registradores para a mesma. Porém, com a especificação do trabalho, a criação de duas ULA era explícita. Com isso, decidi fazer duas ULAS, mudando apenas as operações de cada uma delas, e fazendo ”port maps”em um arquivo principal ”TOP-LEVEL”, na qual chamei de ULAControl. Para cada ULA criei um banco de registradores, não sei se seria a forma mais correta de ser feito, porém achei o mais simples de fazer, cada ULA também tem seus sinais específicos, como zero, overflow...

Como dá pra ver, pelo desenho do circuito, cada ULA é ligada ao seu banco de registradores, onde um valor é carregado através do sinal valorLoad e pode ser definido pelas ULAS em sequencia.

2. ULA 1

O código de cada operação tem 2 bits, utilizei um ”WHEN- ”SELECT”, para definir a operação da ULA. A ULA um tem as seguintes operações:

- A
- A or B
- A + B
- A AND (NOT B)

3. ULA 2

O código de cada operação tem 2 bits, utilizei um ”WHEN- ”SELECT”, para definir a operação da ULA. A ULA dois tem as seguintes operações:

- D
- NOT (C)
- D ¡¡ 2
- C XOR D

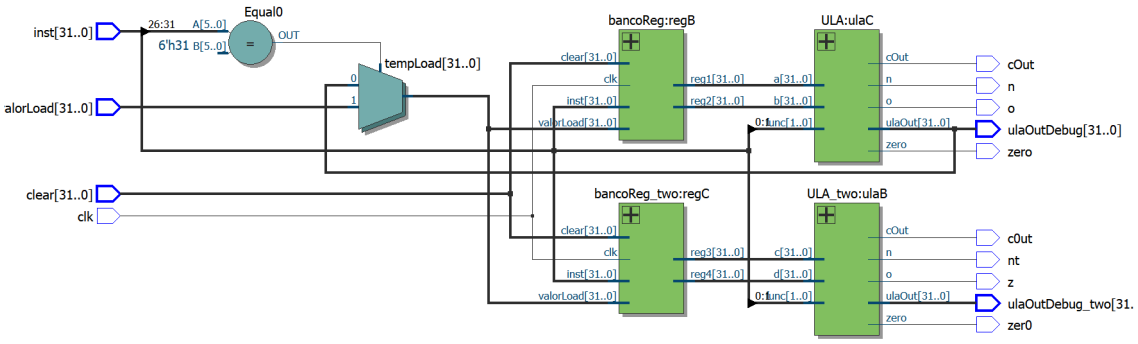

Figura 1. Desenho do circuito

4. TesteBenchs

De acordo com o assunto visto nas video-aulas, para realizar o TestBench era necessário criar um arquivo ”Top-Level”com o component da ULAControl, com o intuito de termos uma declaração de Componente para a Unidade em Teste (UUT). Para fazer os testes utilizei a ferramenta do ModelSim, onde criei um script, chamado script.do, como foi ensinado em aula, para rodar todos os arquivos .vhd, dessa vez utilizando o arquivo ULAControl_TB.vhd como ”TOP-LEVEL”. Este arquivo compila todos os arquivo .vhd do projeto das ULAS + banco de registradores, junto com o formato de onda salvo e rodando a 20 us.

Eu estava tendo alguns problemas para salvar o arquivo wave.do, eu clicava em salvar na pasta e, posteriormente, após fechar a simulação do arquivo o arquivo wave.do desaparecia da pasta. Tive que reiniciar o ModelSim várias vezes, e mesmo assim quando vou rodar o comando ”do script.do”no terminal do ModelSim, ainda tenho alguns erros em abrir o arquivo de imagem wave.do.

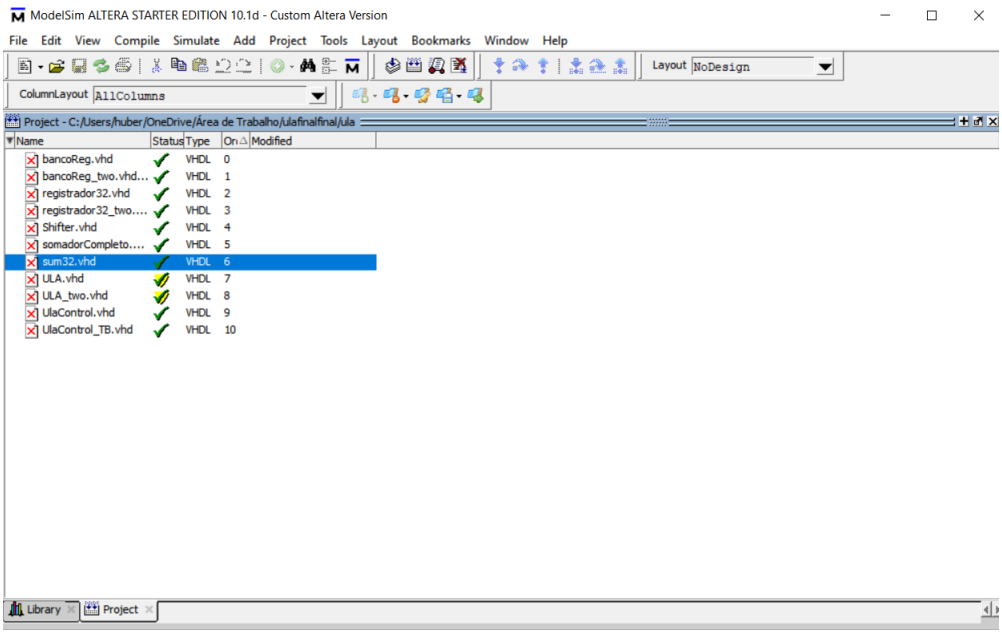

Figura 2. ModelSim

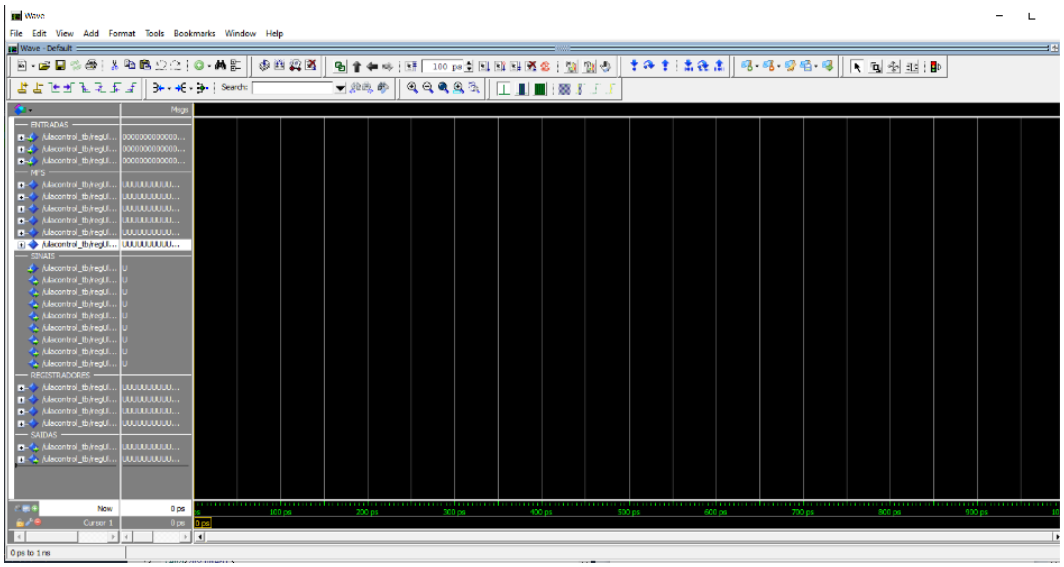

Figura 3. Wave