



## **Laboratório 2**

### **- ULA e FPULA -**

#### **Objetivos:**

- Introduzir ao aluno a Linguagem de Descrição de Hardware Verilog;
- Familiarizar o aluno com a plataforma de desenvolvimento FPGA DE1-SoC e o software QUARTUS Prime da Intel;
- Desenvolver a capacidade de análise, síntese e caracterização de sistemas digitais usando HDL;

Dica: Para envio dos arquivos fontes limpe o projeto (Menu project/clean project) antes de gerar o .zip.

#### **1) (0.0) Implementação de um driver para display de 7 segmentos e Multiplexadores**

1.1 (0.0) Faça o download do arquivo lab2.zip do moodle, deszip em um diretório do desktop que NÃO contenha espaço nem caracteres especiais no nome. O arquivo QuartusPrimev1.txt contém instruções sobre como criar um projeto e usar em simulações e como implementar na DE1-SoC.

1.2 (0.0) Para a versão assíncrona do decoder7.v:

- Indique os requerimentos físicos da implementação: i) número de elementos lógicos (ALMs), ii) número de registradores, iii) quantidade de bits de memória e iv) número de blocos DSP usados;
- Crie o arquivo de testbench e realize a simulação funcional usando o ModelSim;
- Crie o arquivo de forma de onda e realize a simulação funcional (University Program);
- Usando o TimeQuest, indique os requerimentos temporais: i) o caminho de maior atraso, ii) maior tempo de atraso tpd e iii) se há algum requerimento não atendido.
- Defina o arquivo TopDE.v como o toplevel, sintetize na DE1-SoC e filme o funcionamento para todos os valores de entrada.

Para a apresentação da verificação dos projetos nesta disciplina, crie um canal para o seu grupo no YouTube e poste os vídeos dos testes com os links no relatório. Passos do vídeo: Apresente o grupo e seus membros, o projeto a ser realizado e os testes solicitados.

1.3 (0.0) Para a versão síncrona do decoder7.v:

- Indique os requerimentos físicos da implementação: i) número de elementos lógicos (ALMs), ii) número de registradores, iii) quantidade de bits de memória e iv) número de blocos DSP usados;
- Com o arquivo de testbench criado, realize a simulação funcional usando o ModelSim;
- Com o arquivo de forma de onda criado, realize a simulação funcional;
- Usando o TimeQuest, para um clock de 50MHz, indique os requerimentos temporais: i) o caminho de maior atraso, ii) tempos th, tco, tsu e slacks, iii) máxima frequência de clock utilizável, e iv) se há algum requerimento não atendido.
- Defina o arquivo TopDE.v como o toplevel, sintetize na DE1-SoC e filme o funcionamento para todos os valores de entrada.

#### **2) (1.0) Multiplexadores:**

Implemente e desenhe uma tabela comparativa de requerimentos físicos (número de ALMs) e temporais (tpd) dos seguintes multiplexadores.

- (0.5) 1 bit: 2×1, 4×1 e 8×1
- (0.5) 12 bits: 2×1, 4×1 e 8×1
- (0.5) 20 bits: 2×1, 4×1 e 8×1
- (0.5) 32 bits: 2×1, 4×1 e 8×1

### 3) (4.5) Unidade Lógico Aritmética de Inteiros:

- a) (0.5) Para a ULA de inteiros assíncrona fornecida, desenhe o diagrama em blocos, descreva suas funções e escreva a tabela verdade de seus códigos para cada operação.
- b) (1.0) Crie um testbench e verifique com o ModelSim cada operação implementada. Dica: Escolha valores de entrada que sejam representativos (comuns) e que possam gerar resultados singulares (overflow, zero, etc).
- c) (1.0) Indique os requisitos físicos da implementação da ULA total e para cada operação separadamente: i) Número de Elementos Lógicos (ALMs), ii) Número de Registradores e iii) Quantidade de bits de memória e iv) Número de blocos DSP usados;  
Dica: Defina manualmente o sinal de controle iControl no arquivo ULA.v, pois o Quartus otimiza o projeto retirando as partes não utilizadas.
- d) (1.0) Usando o TimeQuest, indique os requerimentos temporais para a ULA total e para cada operação separadamente: i) o caminho de maior atraso, ii) maior tempo de atraso tpd e iii) se há algum requerimento não atendido.
- e) (0.5) Defina o arquivo TopDE.v como o toplevel, sintetize na DE1-SoC e filme o funcionamento comprovando seu correto funcionamento.
- f) (0.5) Otimize o circuito e demonstre quais fatores foram melhorados pela sua implementação.

### 4) (4.5) Unidade Aritmética de Ponto Flutuante:

- a) (0.5) Para a FPULA síncrona fornecida, desenhe o diagrama em blocos, descreva suas funções e escreva a tabela verdade de seus códigos para cada operação.
- b) (1.0) Crie um testbench e verifique com o ModelSim cada operação implementada. Dica: Escolha valores de entrada que sejam representativos (comuns) e que possam gerar resultados singulares (divisão por zero, overflow, underflow, NaN, zero).
- c) (1.0) Indique os requisitos físicos da implementação da FPULA total e para cada operação separadamente: i) Número de Elementos Lógicos (ALMs), ii) Número de Registradores e iii) Quantidade de bits de memória e iv) Número de blocos DSP usados;  
Dica: Defina manualmente o iControlSignal pois o Quartus otimiza o projeto retirando os elementos não utilizados.
- d) (1.0) Usando o TimeQuest, defina um clock de 50MHz, indique os requerimentos temporais para a ULA total e para cada operação separadamente: i) número de ciclos necessários, ii) caminho de maior atraso, iii) tempos th, tco, tsu e slacks, iv) máxima frequência de clock utilizável, e v) se há algum requerimento não atendido.
- e) (0.5) Defina o arquivo TopDE.v como o toplevel, sintetize na DE1-SoC e filme o funcionamento comprovando seu correto funcionamento.
- f) (0.5) Otimize o circuito e demonstre quais fatores foram melhorados pela sua implementação.