



Universidade de Brasília  
Instituto de Ciências Exatas  
Departamento de Ciência da Computação

Disciplina: CIC 116394 – Organização e Arquitetura de Computadores

Turma D - 2019/2 - Prof. Carla Koike

Equipes de no máximo 5 alunos.

Entrega do relatório (pdf) e fontes em um único arquivo zip pelo Moodle até às 23h55 do dia 18/10/2019. Não serão aceitos relatórios depois do dia e horário definidos. ('Pelamordedeus' não deixe para enviar às 23h54, pois mesmo que o Moodle esteja com problemas ou fora do ar o relatório não poderá mais ser enviado: o melhor é ir enviando versões intermediárias, à medida que for sendo feito).

## **Laboratório 2**

### **- Verilog e Unidade Lógica e Aritmética –**

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 e síntese de sistemas digitais usando uma Linguagem de Descrição de Hardware (LDL).

**Metodologia:**

#### **1) Implementação de um driver para display de 7 segmentos**

1.1 Faça o download do diretório files do Moodle, em um diretório do desktop que *NÃO* contenha espaços nem caracteres especiais no nome. O arquivo Tutorial\_Quartus\_Prime.pdf contém instruções sobre como criar um projeto e usar em simulações e como implementar na DE1-SoC.

1.2 A partir do arquivo decoder7.v, defina-o como o toplevel e compile:

a) 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;

b) Veja o diagrama esquemático do circuito gerado pelo Tools/Netlist Viewers/RTL viewer, e tire printscreens para colocar no relatório;

c) Com o arquivo decoder7\_tb.v realize a simulação funcional usando o ModelSim (Tools/Run simulation tool/RTL simulation), tire printscreens para colocar no relatório;

d) Com o arquivo em forma de onda decoder7.vwf (*University Program*) realize a simulação

funcional (verifique sempre se o *Simulation Settings* estão com os parâmetros *default*), tire printscreens para colocar no relatório;

e) Usando o TimeQuest (Tools/Timing Analyzer), gere a Timing Netlist e em seguida relatório Report Datasheet e indique os requerimentos temporais:

i) o caminho de maior atraso,

ii) maior tempo de atraso tpd.

f) Defina o arquivo TopDE.v como o toplevel, sintetize na DE1-SoC e filme o funcionamento para todos os valores de entrada.

## **2) Unidade Lógico Aritmética de Inteiros**

Abra o Quartus Prime e Project/Restore Archived Project e escolha o arquivo ULA.qar.

a) Para a ULA de inteiros fornecida, descreva suas funções e escreva a tabela de seus códigos para cada operação.

b) Modifique os valores do arquivo de forma de onda ULA.vwf e verifique cada operação implementada. Confira com o RTL Simulation. Dica: Escolha valores de entrada que sejam representativos (comuns) e valores que possam gerar resultados singulares (overflow, zero).

c) Indique os requisitos físicos da implementação da ULA total e para cada operação separadamente:

i) Número de Elementos Lógicos (ALM);

ii) Número de Registradores;

iii) Quantidade de bits de memória;

iv) Número de blocos DSP usados;

v) Somando todos os ALMs do item i) compare com o número de ALMs da ULA total e explique a diferença;

vi) Indique quais são e analise o impacto das funções com maiores circuitos no tamanho da ULA.

Dica: Defina manualmente o sinal de controle iControl no arquivo ULA.v, pois o Quartus otimiza o projeto retirando as partes não utilizadas.

d) 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;

iii) Indique quais são e analise o impacto das funções mais demoradas no tpd.

e) Defina o arquivo TopDE.v como o toplevel, sintetize na DE1-SoC e filme comprovando o correto funcionamento de acordo com o testbench criado no item b).

f) Verifique o quanto seu grupo consegue otimizar esta ULA em termos de requerimentos físicos e/ou temporais, sem retirar nenhuma funcionalidade!

## **3) Unidade Aritmética de Ponto Flutuante**

Abra o Quartus Prime e Project/Restore Archived Project e escolha o arquivo FPALU.qar

a) Para a FPULA fornecida descreva suas funções e escreva a tabela de seus códigos para cada operação.

b) Modifique os valores do arquivo de forma de onda FPULA.vwf e verifique cada operação

implementada. Confira com o RTL Simulation. Dica: Escolha valores de entrada que sejam representativos (comuns) e também valores que gerem resultados singulares (divisão por zero, overflow, underflow, NaN, zero).

c) Indique os requisitos físicos da implementação da FPULA total e para cada operação separadamente:

- i) Número de Elementos Lógicos (ALMs);
- ii) Número de Registradores;
- iii) Quantidade de bits de memória;
- iv) Número de blocos DSP usados;
- v) Somando todos os ALMs do item i) compare com o número de ALMs da ULA total e explique a diferença;
- vi) Indique quais são e analise o impacto das funções com maiores circuitos no tamanho da ULA.

Dica: Defina manualmente o icontrol pois o Quartus otimiza o projeto retirando os elementos não utilizados.

d) 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 à execução de cada operação;
  - ii) tempos  $t_h$ ,  $t_{co}$ ,  $t_{su}$  e slacks;
  - iii) máxima frequência de clock utilizável;
  - iv) se há algum requerimento não atendido. Indique quais são e analise o impacto das funções mais demoradas na máxima frequência utilizável.
- Aumente a frequência para 200MHz e repita os itens i), ii), iii) e iv).  
Analise os resultados obtidos.

e) Defina o arquivo TopDE.v como o toplevel, sintetize na DE1-SoC e filme comprovando o correto funcionamento de acordo com o testbench criado no item b).

f) Verifique o quanto seu grupo consegue otimizar esta FPULA em termos de requerimentos físicos e/ou temporais (sem retirar nenhuma funcionalidade!).