

UNIVERSIDADE FEDERAL DE MINAS GERAIS  
INSTITUTO DE CIÊNCIAS EXATAS  
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Bacharelado em Ciência da Computação  
Disciplina: Introdução aos Sistemas Lógicos  
2o. Semestre de 2017

Professor: Luiz Filipe Menezes Vieira (lfvieira@dcc.ufmg.br)  
Monitor: Vinícius Silva Barros (viniciusbarros@dcc.ufmg.br)

Data da entrega: 20.11.2017

## Laboratório 4: Maior Divisor Comum em Verilog

**Objetivo:** Esta prática tem o objetivo de possibilitar ao aluno uma experiência inicial com programação em Verilog.

Você já aprendeu durante as aulas que as linguagens de descrição de *hardware* são úteis para resolver problemas relacionados à simulação e implementação de circuitos lógicos. Os três principais ganhos destas linguagens são:

1. A possibilidade de executar simulações totalmente controladas de projetos analógicos, digitais e híbridos em vários níveis de abstração;
2. Modularização, que permite reutilizar fragmentos de código já elaborados e propriamente testados, aumentando a eficiência do trabalho;
3. Sintetização do código para circuitos reais, podendo ser carregado em FPGAs ou chips dedicados.

Sua tarefa será implementar um algoritmo que seja capaz de retornar o maior divisor comum entre dois números inteiros de 16 *bits* na linguagem Verilog, apresentada em sala de aula. O maior divisor comum de dois ou mais números é o maior inteiro positivo que divide cada um destes números sem deixar resto. Por exemplo: o maior divisor comum entre 24 e 9 é 3. Fique atento a alguns aspectos matemáticos importantes válidos para este problema:

1.  $\text{gcd}(0, 0) = 0$ ;
2.  $\text{gcd}(0, a) = a$ ;
3.  $\text{gcd}(a, b) = \text{gcd}(b, a)$ ;

Você deverá criar um módulo independente e totalmente funcional, ou seja, seu módulo não pode fazer uso de ferramentas prontas ou de outros módulos. O *test bench* para validação da sua simulação irá percorrer todas as combinações válidas de entradas e verificar, para cada uma, a corretude do seu algoritmo. O módulo criado deverá ter, obrigatoriamente, a seguinte definição, onde **inA** e **inB** são as entradas e **out** é a saída:

```
module gcd([15:0] inA, [15:0] inB, [15:0] out);  
  
    //Your logic goes here.  
    //Do not forget to correctly assign types and sizes  
    //to your variables.  
  
endmodule
```

Além disso seu arquivo deverá ter o obrigatoriamente o nome **gcd.v** para que possa ser executado de forma automática pelo corretor. Para a simulação você pode utilizar o Icarus Verilog que está disponível nas máquinas da graduação ou o programa cver. Seu programa deve ser corretamente compilável com a seguinte linha de comando:

```
iverilog -o compiled.vvp gcd.v
```

## Atividades

Cada grupo deverá submeter as seguintes atividades pelo Moodle, contendo o nome de todos os participantes:

- Documentação resumida contendo uma descrição dos detalhes de implementação e dificuldades encontradas (não mais que 3 páginas).
- O código fonte em Verilog que deverá funcionar no *iverilog*.
- Testes que demonstram o correto funcionamento do programa.

**Importante:** Apenas um membro do grupo deverá submeter o trabalho contendo o nome de todos os integrantes em um arquivo ZIP contendo o código, o relatório e quaisquer outros arquivos relevantes. O relatório deverá ser entregue no formato **pdf**.