

Universidade do Vale do Itajaí - Projeto de Sistemas Digitais
Professor: Douglas Rossi de Melo
Avaliação 5 – Projeto RTL

Instruções:

1. Esta avaliação pode ser realizada em grupos de até três alunos.
2. Esta avaliação tem por objetivo consolidar o aprendizado sobre o projeto RTL usando VHDL.
3. Os circuitos devem ser descritos em VHDL, verificados com o uso de testbenches utilizando as ferramentas Quartus e ModelSim, e prototipados em FPGA.
4. Deve-se apresentar os diagramas RTL dos circuitos e identificar os custos de implementação dos sistemas expressos pelo número de LUTs e FFs, além da frequência máxima de operação.
5. Os nomes dos sinais de entrada e de saída dos circuitos apresentados são voltados a facilitar a prototipação em dispositivo FPGA.
6. Aplique o seguinte guia de estilo na codificação VHDL:
 - a. Tabulação com 02 espaços (substituindo *tabs* por espaços);
 - b. Palavras reservadas da linguagem em caixa baixa;
 - c. Nomes de sinais em caixa alta (exceto os prefixos);
 - d. Utilize os seguintes prefixos para identificar os nomes:
 - i_NOME : pino de entrada
 - o_NOME : pino de saída
 - w_NOME : fio interno
 - u_NOME : instância de componente
 - r_NOME : registrador
 - t_NOME : novo tipo
 - s_NOME : estado
7. Deve ser postado um relatório básico em formato **PDF**, no ambiente **Material Didático**, que contenha:
 - a. Identificação dos autores e do trabalho
 - b. Enunciado de cada circuito
 - c. Códigos VHDL dos circuitos e testbenches
 - d. Diagrama RTL gerado pelo Quartus
 - e. Diagrama de máquina de estados gerado pelo Quartus
 - f. Diagrama de formas de onda da simulação do Modelsim
 - g. Foto do sistema prototipado em FPGA
 - h. Tabela de custos do circuito (número de LUTs e FFs)
 - i. Frequência máxima de operação
 - j. Discussão dos resultados

PROJETO

Dado o algoritmo do máximo divisor comum (MDC) descrito a seguir:

```
1:  int x, y;                                // variáveis internas
2:  while (1) {
3:      o_rdy = 1;                            // o_rdy: saída de 1 bit
4:      while (!i_go);                       // i_go: entrada de 1 bit
5:      x = i_x;                             // i_x : entrada de 8 bits
6:      y = i_y;                             // i_y : entrada de 8 bits
7:      o_rdy = 0;
8:      while (x != y) {
9:          if (x < y)
10:             y = y - x;
11:          else
12:             x = x - y;
13:      }
14:      o_d = x;                             // o_d: saída de 8 bits
15: }
```

REQUISITOS PARA O PROJETO

1. O sistema deve ser composto de um controle e um caminho de dados
2. O projeto deve seguir uma abordagem hierárquica, com a seguinte estrutura sugerida:
 - a. mdc_control.vhd (máquina de estados);
 - b. mdc_datapath.vhd (caminho de dados);
 - c. mdc_top.vhd (integração da máquina de estados e do caminho de dados);
3. Considere que as entradas de dado, a saída de dado e as variáveis internas são **números inteiros de 8 bits** (se identificar a necessidade de tratar número negativos, use o pacote IEEE correspondente);
4. Utilize o método de projeto RTL visto em sala de aula, o qual parte de um algoritmo C e a partir dele obtém uma máquina de estados de alto nível e desta extrai os blocos de controle e de processamento (caminho de dados);
5. Ao realizar o projeto, identifique claramente os nomes das entradas, saídas e sinais internos nos diagramas esquemáticos de modo a facilitar a documentação e a descrição do projeto em VHDL;
6. Para sinais internos, utilize como referência a porta de saída e o nome do bloco ao qual o sinal está conectado. Exemplo: para um sinal de 8 bits conectado à saída de um somador, utilize o seguinte identificador: w_ADDER_OUT. Caso o sistema contenha mais de uma instância de um mesmo bloco, diferencie-os com algum sufixo (ex. w_ADDER0_OUT, w_ADDER1_OUT,...). Nota: os exemplos utilizados são apenas a título de ilustração, pois o algoritmo não utiliza operação de soma;
7. Inclua um sinal de entrada para indicar quando o processamento deve ser iniciado (ex. i_GO ou i_START);

8. Inclua um sinal de saída para indicar quando o processamento for concluído (ex. o_RDY). Este sinal deve ser igual a 0 enquanto o processamento estiver em andamento; e
9. Para o projeto, procure utilizar alguma ferramenta de apoio (sugestão draw.io) para documentar e facilitar a implementação posterior.