

Laboratório 3 – Turma A – ULAs e Instanciação de Componentes **Quarta-feira, 04 de Outubro de 2017**

Instruções: Após o visto do professor ou do monitor, submeter via moodle o projeto contendo os arquivos VHD, testbench, reporte de síntese, printscreen de simulação, printscreen da arquitetura RTL, da simulação e do circuito roteado (FPGA Editor) em uma pasta zipada chamada “sobrenome-matricula”.

Exercício 1. Calculadora 12 bits

1.A) Modifique a Máquina de Estados Finitos da ULA de 12 bits em ponto fixo desenvolvida em sala de aula de forma a acrescentar as seguintes operações: soma um (+1), resta um (-1), comparação maior que ($>=$), comparação menor que ($<=$) e comparação de igualdade (=) e negação $\text{not}(A)$. No caso das comparações serem verdadeiras então todos os bits da saída devem ser acertados em ‘1’. Adicionalmente, inclua uma saída de *carry_out* que deve ser mapeada em um led da placa de desenvolvimento. Se acontecer um *overflow* ou *underflow* nas operações de soma e subtração, respectivamente, então a saída *carry_out* deve ser acertada em ‘1’, caso contrário deve ser acertada em ‘0’.

1.B) Simulação: Realize um testbench da ULA e verifique através de simulação o comportamento do circuito. Use pelo menos uma simulação para cada operação. Caracterize o circuito obtendo o tempo de execução de cada operação. Obtenha do reporte de síntese a ocupação do circuito em termos de LUTs, FFs e DSPs.

1.C) Implementação: Mapeie as entradas *reset*, *lopA*, *lopB* e *start* em *pushbuttons*. Mapeie as entradas *oper* e *sel* nos switches. Mapeie as saídas *saida* e *carry_out* nos leds. Obtenha o layout do circuito na ferramenta *FPGA Editor* após o processo de *Place and Route* (PAR). Habilite os *routes* e faça um zoom na área em que o circuito foi roteado.

Vistos: apresentar para o monitor ou para o professor os seguintes vistos:

- Visto 1: verificação no FPGA da calculadora usando os leds

Nota 1: a FSM da ULA de 12 bits em ponto fixo está disponível no moodle da disciplina.

Nota 2: a ULA de 12 bits em ponto fixo opera com sinal. Se o bit mais significativo foi igual a ‘1’ então a palavra é um número negativo representado em complemento a 2

Exercício 2. Expansão por séries de Maclaurin do operador exponencial (e^x) de quarta ordem

2.A) Descreva uma FSM que implemente a expansão por séries de Maclaurin do operador exponencial de quarta ordem mostrada na equação a seguir, onde x deve estar na faixa [-1.0 a 1.0]. Use a ULA de 12 bits em ponto fixo desenvolvida em sala de aula. A FSM deve ter um *start* para indicar o início da operação e uma saída *ready* indicando o fim da operação.

$$f(x) = e^x = \sum_{n=0}^4 \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!}$$

2.B) Simulação: Realize um testbench e verifique através de simulação o comportamento do circuito. Para fazer as simulações use o formato $0.yy$ onde yy são os últimos dois números ou os primeiros dois números da matrícula de cada estudante do grupo. Grupos de 2 pessoas devem realizar no mínimo quatro simulações e grupos de 3 pessoas devem realizar no mínimo 6 simulações. Metade das simulações devem usar argumentos negativos. Caracterize o circuito obtendo o tempo de execução de cada operação. Com base nas simulações estime o erro de precisão da sua solução. Obtenha do reporte de síntese a ocupação do circuito em termos de LUTs, FFs e DSPs.

2.C) Implementação: Mapeie as entradas *reset*, e *start* em *pushbuttons*. Mapeie a entrada x de 12 bits nos switches. Mapeie a saídas $f(x)$ e *ready* nos leds. Obtenha o layout do circuito na ferramenta *FPGA Editor* após o processo de *Place and Route* (PAR). Habilite os *routes* e faça um zoom na área em que o circuito foi roteado.

Nota 1: os valores $1/2!$ e $1/3!$ podem ser previamente calculados e ingressar como parâmetros na arquitetura.

Dica: Veja o exemplo do exercício do polinômio de terceira ordem feito em aula.

Vistos: apresentar para o monitor ou para o professor os seguintes vistos:

- Visto 1: verificação no FPGA

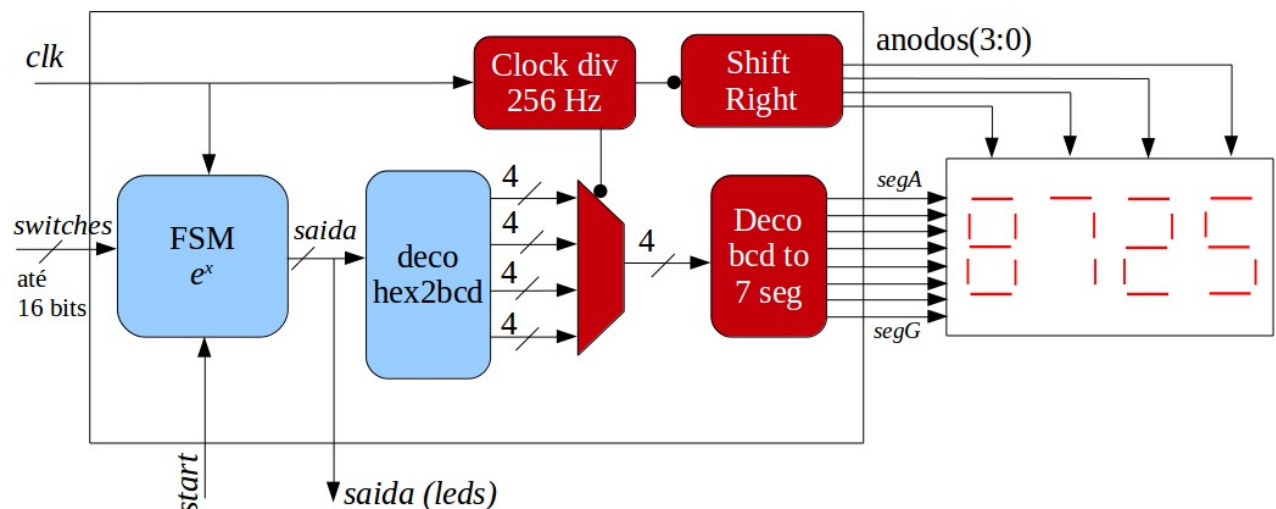
Exercício 3 Opcional. Melhorar precisão da FSM para e^x (Bonus de 2.0 pontos)

3.A) Modifique o tamanho de palavra da Máquina de Estados Finitos da ULA em ponto fixo de forma a melhorar a precisão das contas aritméticas para o cálculo da função exponencial do exercício 2.

3.B) No moodle da disciplina pode ser encontrado o arquivo VHDL de um decodificador **hex2bcd** de 12 bits (hexadecimal para bcd). Modifique ou use o decodificador hex2bcd de forma que a saída binária seja representada em decimal nos displays de sete segmentos. Não é necessário representar a parte inteira e/ou fracionária, apenas represente em decimal toda a palavra binária.

3.C) Implemente um testbench e realize simulações. Calcule o erro de precisão e consumo de recursos. Compare com os resultados obtidos no exercício anterior.

3.D) Implemente uma arquitetura RTL (crie um arquivo chamado *top_module*) que instancie os componentes mostrados na figura abaixo. Deve-se usar um decodificador BCD para 7 segmentos e implementar um divisor de clock de 256 Hz que controle um multiplexador e um registrador de deslocamento no intuito de multiplexar os anodos dos displays. Obtenha o layout do circuito na ferramenta *FPGA Editor* após o processo de *Place and Route* (PAR). Habilite os routes e faça um zoom na área em que o circuito foi roteado.



Vistos: apresentar para o monitor ou para o professor os seguintes vistos:

- Visto 1: Exercício 3 (Bônus) - verificação no FPGA

Bom trabalho!