

PCS3115 - Sistemas Digitais I - Projeto 1 - Resto de Divisão em VHDL

por Glauber De Bona

17/06/2020

Introdução

A divisão entre dois números inteiros positivos pode ser implementada em um sistema digital usando somas/subtrações de diferentes maneiras. Os métodos mais rápidos são baseados no algoritmo desloca-e-subtrai, similar ao que utilizamos no papel. O método mais simples, no entanto, consiste em contar quantas vezes podemos subtrair o divisor do dividendo e é conhecido como método das subtrações sucessivas, ilustrado abaixo. Se estamos interessados no resto da divisão, e não no quociente, não precisamos contar o número de subtrações.

Listagem 1: Cálculo do quociente por subtrações sucessivas

```
int divide (int dividendo , int divisor){
    int quociente = 0;
    while(dividendo >= divisor){
        dividendo = dividendo - divisor;
        quociente = quociente +1;
    }
    return quociente;
}
```

Listagem 2: Cálculo do resto por subtrações sucessivas

```
int resto (int dividendo , int divisor){
    while(dividendo >= divisor)
        dividendo = dividendo - divisor;
    return dividendo;
}
```

Por exemplo, para calcular o resto da divisão de 14 (dividendo) por 4 (divisor), subtraímos 4 de 14 até que obtenhamos um número menor do que 4, que será o resto, que neste caso é igual a 2:

dividendo	divisor	quociente
14	4	0
10	4	1
6	4	2
2	4	3

Tabela 1: Exemplo de execução do método de subtrações sucessivas para divisão.

Atividade

Este projeto consiste em projetar, em VHDL, uma sistema digital capaz de calcular o resto da divisão pelo método das subtrações sucessivas. Os operandos e o resto serão representados na notação binária sem sinal, com 16 bits cada.

Além das entradas de dados dividendo e divisor, o circuito deve possuir uma entrada de controle inicio, ativa em ALTO, que determina quando o cálculo deve se iniciar. Além da saída de dados resto, deve haver uma saída de controle fim, ativa em ALTO, que avise quando o cálculo terminou. O projeto deve ser implementado via fluxo de dados (FD) e unidade de controle (UC), e a entrada reset, ativa em ALTO, deve levar a máquina de estados finita da UC ao seu estado inicial.

Seu projeto deve seguir **estritamente** a entidade abaixo:

```
entity resto is
  port (
    clock, reset: in bit;
    inicio: in bit;
    fim: out bit;
    dividendo, divisor: in bit_vector(15 downto 0);
    resto: out bit_vector(15 downto 0)
  );
end entity;
```

Listagem 3: Entidade a ser seguida

Note que o resto da divisão não é definido quando o divisor é igual a zero, e os algoritmos ilustrados acima entrariam em loop infinito neste caso. Para evitar esta situação, seu sistema deve fazer com que o resto seja igual ao próprio dividendo quando o divisor é igual a zero.

Como restrição de tempo, cada operação de cálculo de resto deve ser realizada em no máximo 200 mil ciclos de clock. Este limite será calculado entre a ativação da entrada inicio e da saída fim.

Como sistema digital, entenda um circuito sequencial síncrono, onde as mudanças no estado de elementos de memória são sincronizadas por um sinal de clock.

Após o término de uma operação de resto, o sistema também deve voltar ao estado inicial, pronto para realizar outra operação.

Para satisfazer esta restrição, o laço principal da máquina de estados, que faz as subtrações, deve ter no máximo dois estados.

Instruções para Entrega

Há um *link* específico no e-Disciplinas para submissão deste projeto. Acesse-o somente quando estiver confortável para enviar sua solução. Em cada atividade, você pode enviar apenas um único arquivo codificado em UTF-8. O nome do arquivo não importa, mas sim a descrição VHDL que está dentro. A entidade da implementação está na Listagem 3 e deve ser idêntica na sua solução ou o juiz não irá processar seu arquivo.

Quando acessar o *link* no e-Disciplinas, o navegador abrirá uma janela para envio do arquivo. Selecione-o e envie para o juiz. Jamais recarregue a página de submissão pois seu navegador pode enviar o arquivo novamente, o que vai ser considerado pelo juiz como um novo envio e pode prejudicar sua nota final. Caso desista do envio, simplesmente feche a janela.

Depois do envio, a página carregará automaticamente o resultado do juiz, quando você poderá fechar a janela. A nota dada pelo juiz é somente para a submissão que acabou de fazer. Esta atividade permite **até 5 submissões** e a nota será a **maior nota** dentre todas as suas submissões. Sua nota na atividade poderá ser vista no e-Disciplinas e pode diferir da nota dada pelo juiz.

Não está permitido usar a biblioteca `std_logic_1164`, a `textio` e qualquer outra biblioteca não padronizada. Também certifique-se que não imprime nada na saída da simulação.

Atenção: não atualize a página de envio e não envie a partir de conexões instáveis (e.g. móveis) para evitar que seu arquivo chegue corrompido no juiz.

Qualquer editor de código moderno suporta UTF-8 (e.g. Atom, Sublime, Notepad++, etc).

Pode demorar alguns segundos até o juiz processar seu arquivo.

Está permitido o uso da biblioteca `numeric_bit`.