



## EEC0055 - Projeto de Sistemas Digitais

4º ano - 1º semestre

Exame - 19 de janeiro de 2018

Duração máxima: 2h30m, sem consulta.

[1.5 valores]

- 1 - Considere o ambiente de projeto de sistemas integrados digitais para dispositivos FPGA que foi usado durante as aulas laboratoriais (XILINX ISE). Explique em que consistem os processos de síntese RTL e de *Place & Route* e diga quais são as características relevantes dos modelos do sistema digital em projeto que são usados como origem e destino para cada um desses processos.

[3 valores]

- 2 - Num dos estágios do projeto de um sistema digital síncrono (com um único sinal de relógio) pode ser calculada uma estimativa para a frequência máxima do sinal de relógio usando as ferramentas de projeto apropriadas.

[1.5 valores]

- a) Diga, justificando, qual é a fase do projeto em que é obtida a estimativa mais realista e explique de que forma é determinado esse valor (apresente um exemplo que ilustre a sua resposta).

[1.5 valores]

- b) É possível que o circuito resultante possa funcionar corretamente com um sinal de relógio com uma frequência francamente superior à máxima estimada (por exemplo 10X superior)? Em que condições? Explique como poderia identificar uma situação dessas, referindo-se às ferramentas de projeto para dispositivos FPGA que foram usadas nas aulas laboratoriais.

[3 valores]

- 3 - A linguagem de descrição de *hardware* digital Verilog suporta dois operadores com comportamentos diferentes para representar a atribuição do resultado de uma expressão a um registo ou a um fio: atribuições *blocking* (=) e atribuições *non-blocking* (<=).

[1.5 valores]

- a) Explique por que razão um modelo Verilog pode resultar funcionalmente errado se na codificação dos blocos síncronos com sinal de relógio, representados por processos `always @(posedge clk)`, forem usadas apenas atribuições do tipo *blocking* (=).

[1.5 valores]

- b) Diga, justificando, em que fase de verificação (simulação funcional, simulação pós-síntese ou simulação pós-place&route) esse erro pode ser detetado.

[3 valores]

- 4 - O uso incorreto da declaração `always @*` para codificar o comportamento de um circuito combinacional pode resultar na síntese de elementos de memória do tipo *latch* transparente em vez de circuitos completamente combinacionais.

[1.5 valores]

- a) Diga, justificando, qual dos processos seguintes conduz à síntese desse tipo de elementos de memória, não representando por isso um circuito combinacional.

(i)	(ii)	(iii)	(iv)
<pre>always @* begin   y = output1;   if ( enable )     y = output2; end</pre>	<pre>always @* begin   if ( enable )     y = output1;   else     z = output2; end</pre>	<pre>always @* begin   if ( enable )     y = output1;   else     y = output2; end</pre>	<pre>always @* begin   if ( enable )     y = output1;   else     y = y; end</pre>

[1.5 valores]

- b) Escolha um dos exemplos que tenha referido na alínea a) e admita que se pretendia realmente implementar a funcionalidade (não combinacional) representada pelo modelo Verilog. Explique como deveria corrigir o código apresentado de forma a garantir o comportamento descrito mas usando apenas elementos de memória do tipo *flip-flop* (ativos à transição do sinal de relógio). Considere que os sinais **output1** e **output2** e **enable** são provenientes de funções combinacionais cujas entradas são registos síncronos com um mesmo sinal global de relógio.

---

[3 valores]

5 - *Pipelining* é uma estratégia de projeto que pode ser usada para aumentar o desempenho de um circuito combinacional intercalado entre registos. Considere que o desempenho do circuito é caracterizado pelo número de resultados produzidos por unidade de tempo. Diga, justificando, se são verdadeiras ou falsas as seguintes afirmações:

- É possível conseguir um aumento do desempenho do circuito *pipelined* mesmo que seja mantida a frequência do sinal de relógio.
- O aumento de desempenho que se obtém com a introdução de N registos de *pipeline* (resultando em N+1 partições do circuito combinacional) é sempre inferior a N+1.
- O aumento de desempenho resultante é independente da cadência a que os dados são aplicados ao circuito.

---

[6.5 valores]

6 - Considere o projeto de uma unidade de processamento de dados para efetuar o cálculo da expressão:

$$y = ( a*a + b*b ) * c$$

Os operandos **a**, **b** e **c** e o resultado **y** representam valores inteiros de 32 bits sem sinal e sabe-se que os operandos assumem valores que nunca conduzem a situações de *overflow*. O sistema em que este bloco se integra irá processar sequências de dados longas (formadas por várias dezenas de elementos) que lhe são apresentados à frequência de 50 MHz, devendo produzir resultados a essa mesma cadência. O circuito deve ser capaz de operar com um sinal de relógio de 50 MHz.

Uma primeira versão deste circuito consistiu no módulo Verilog seguinte:

```
module ex1718q6( input clock, input reset,
                input [31:0] a,
                input [31:0] b,
                input [31:0] c,
                output reg [31:0] y
                );

    reg [31:0] ra, rb, rc;

    always @(posedge clock)
    begin
        if ( reset )
            begin
                ra <= 32'd0; rb <= 32'd0; rc <= 32'd0; y <= 32'd0;
            end
        else
            begin
                y <= ( ra * ra + rb * rb ) * rc;
                ra <= a; rb <= b; rc <= c;
            end
    end
end
```

[1 valor]

- a) Esboce um diagrama de blocos que represente o circuito lógico em que será traduzido aquele módulo Verilog. Construa esse diagrama com base em blocos ao nível RTL (registos, multiplexadores, somadores, multiplicadores, etc) identificando devidamente todos os sinais envolvidos (pode omitir a representação explícita dos sinais **reset** e **clock**).

[1.5 valores]

- b) No final de um certo estágio do processo de implementação obteve-se uma estimativa para a frequência máxima de relógio de apenas 39 MHz (período igual a 25.6 ns), longe dos 50 MHz pretendidos (período de 20 ns). Modifique o modelo Verilog dado de forma a conseguir aumentar a frequência máxima do sinal de relógio suportada pelo circuito e justifique o aumento estimado para o relógio com a transformação que propõe.

[1 valor]

- c) Considere que a versão do circuito que propôs em b) consegue operar com o relógio de 50 MHz. Diga quanto tempo é necessário para completar o processamento de sequências de (i) 10 elementos, (ii) 100 elementos e (iii) 1000 elementos.

[1.5 valores]

- d) Admita agora que este circuito deve continuar a operar com um relógio de 50 MHz, mas que só tem de produzir resultados a um ritmo de 10 MHz (i.e. a cada 5 ciclos do relógio de 50 MHz). Modifique o modelo Verilog dado de forma a satisfazer estes novos requisitos temporais, minimizando a sua complexidade lógica (área).

[1.5 valores]

- e) Num cenário de aplicação diferente, pretende-se que aquela função produza resultados a uma cadência de apenas 1 MHz (i.e. a cada 50 ciclos do mesmo relógio de 50 MHz). Diga como poderia implementar uma solução que consiga reduzir ainda mais a área do circuito resultante (nota: não se pretende que construa um modelo Verilog descrevendo o circuito proposto, mas deve justificar a descrição apresentada recorrendo a diagramas de blocos que ilustrem a sua solução).

◀= Fim =▶