



EEC0055 - Projeto de Sistemas Digitais

4º ano - 1º semestre

Recurso - 2 de fevereiro de 2018

Duração máxima: 2h30m, sem consulta.

[3 valores]

1 - Considere o ambiente de projeto de sistemas integrados digitais para dispositivos FPGA que foi usado durante as aulas laboratoriais (XILINX ISE e QuestaSim).

[1.5 valores]

a) Explique por que razão é fundamental executar o processo de simulação funcional com a análise de cobertura de código (*code coverage*) ativada.

[1.5 valores]

b) Essa necessidade mantém-se para o processo de verificação por simulação que deve ser realizado após a síntese do código HDL (simulação *post-translate*)? Explique porquê.

[3 valores]

2 - Considere que um sistema digital síncrono A operando com um único sinal de relógio com frequência F_{CA} tem uma entrada X que provém de um outro sistema B, igualmente síncrono, mas que opera com um sinal de relógio diferente com frequência F_{CB} . Sabe-se que F_{CB} tem um valor nominal que é aproximadamente 10X menor que F_{CA} .

[1.5 valores]

a) Considere que a entrada X é formada por apenas um bit e que o seu valor pode ser mudado pelo circuito B em qualquer ciclo do seu sinal de relógio. Explique por que razão o excerto de código Verilog seguinte não deve ser implementado pelo sistema A, onde o sinal X é essa entrada, (*não funcionaria porque...*) e apresente um exemplo que ilustre a sua resposta.

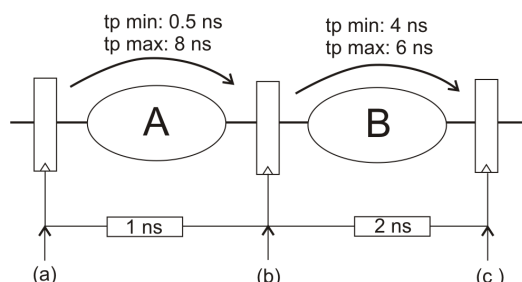
```
always @(posedge clock_A)
if ( X )
y <= a*b + c*d;
else
y <= a*c + b*d;
```

[1.5 valores]

b) Mostre como resolver o problema identificado na alínea anterior.

[2.5 valores]

3 - Considere o circuito mostrado na figura, onde os blocos combinacionais A e B apresentam os tempos máximos e mínimos de propagação mostrados e os 2 ramos que interligam o sinal de relógio entre os 3 registos têm os tempos de propagação indicados (admitindo qualquer sentido de propagação). Nesta análise pode considerar que são nulos os tempos de propagação, tempos de *setup* e tempos de *hold* dos *flip-flops*.



Devido aos atrasos de propagação introduzidos pelas ligações do sinal de relógio, o relógio não está em fase em todos os seus registos. Diga, justificando, para que pontos (a), (b) ou (c) onde o sinal de relógio pode ser aplicado o circuito funcionaria corretamente e para quais não funcionaria. Admita que o sinal de relógio tem uma frequência de 50 MHz (período de 20ns).

[2.5 valores]

4 - Os dois modelos Verilog não representam o mesmo circuito lógico, embora ambos possam ser usados para sintetizar circuitos que realizam o cálculo da expressão $w * (x+y) * z$:

(a)	(b)
<pre>always @(posedge clock) begin a <= x + y; b <= a * z; c <= b * w; end always @* out = c;</pre>	<pre>always @* begin a = x + y; b = a * z; c = b * w; end always @(posedge clock) out <= c;</pre>

[1.5 valores]

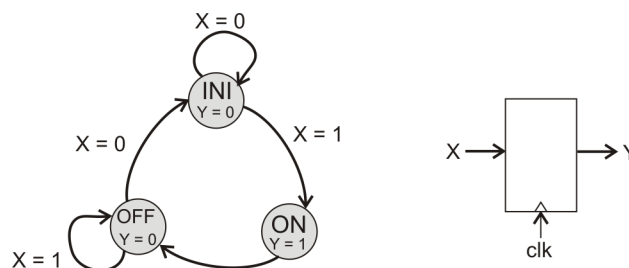
a) Explique as diferenças os circuitos representados pelos 2 excertos de código Verilog (e gerados por uma ferramenta de síntese RTL) e diga em que condições podem ser usados para realizar o cálculo referido.

[1.5 valores]

b) Apresente cenários em que cada uma das implementações tenha vantagens face à outra.

[3 valores]

5 - Construa um modelo Verilog sintetizável que implemente a máquina de estados (modelo de Moore) cujo diagrama de transição de estados é o mostrado na figura. A entrada X provém de outro circuito síncrono com o mesmo sinal de relógio. Considere que estão definidos os parâmetros INI, ON e OFF com códigos de 2 bits que representam os estados com os nomes correspondentes e construa o seu modelo de forma a minimizar a complexidade lógica do circuito.



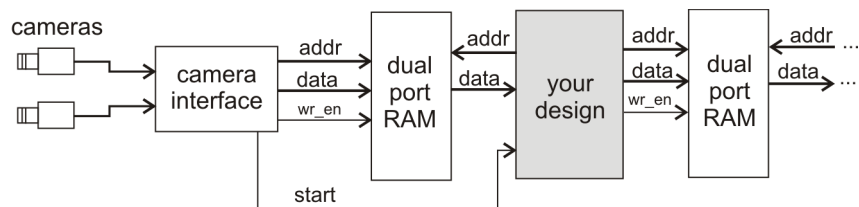
[6 valores]

6 - No projeto de um sistema de processamento de vídeo em tempo real pretende-se implementar um módulo para realizar a média geométrica de duas imagens monocromáticas com 960 x 640 pixel, geradas a um ritmo de 200 imagens por segundo (fps). Funcionalmente, a operação pode ser representada pelo pseudo-código seguinte, onde N e M são o número de *pixels* segundo cada dimensão da imagem:

```
For i=0 to N-1
  For j=0 to M-1
    Image_result( i, j ) = sqrt( Image_A( i, j ) * Image_B( i, j ) )
```

O interface entre o sistema de aquisição de imagem e o módulo a desenvolver é realizado através de memória partilhada, que é preenchida com as duas imagens origem ao ritmo de aquisição das câmaras (200 fps). Esse interface produzirá um sinal **start** para indicar quando na memória foram escritas duas novas imagens, e que deve ser usado para iniciar o processo de cálculo. A imagem resultado deve ser produzida ao mesmo ritmo das imagens origem (200 Hz), o que significa que aquele processo de cálculo tem que ser completado em menos de 5 ms.

A versão inicial do sistema usa uma memória *dual-port*, em que um porto de escrita é usado pelo interface com as câmaras para carregar as imagens e o outro porto (de leitura e escrita) é usado para ler as imagens origem, sendo a imagem destino escrita numa outra memória (ver figura). Cada porto de acesso a essas memórias lê/escreve um *pixel* por ciclo de relógio (palavra de 10 bits). Ambas as memórias têm um acesso síncrono com o relógio global do sistema. Após o sinal **start** ser ativado, o conteúdo da memória quer contém as imagens origem é preservado durante o tempo de *frame* (5 ms). As imagens são armazenadas na memória linha a linha, com um *pixel* por palavra de memória e em posições contíguas.



[1.5 valores]

- a) Considere que as duas imagens origem são lidas da mesma memória e que a imagem destino é escrita numa memória independente. Diga justificando qual deverá ser a frequência mínima do sinal de relógio para que sejam cumpridos os requisitos temporais referidos acima, considerando apenas as restrições impostas pelo sistema de memória (admita que os operadores aritméticos raiz quadrada e produto não restringem a duração do período mínimo do relógio).

[1.5 valores]

- b) Para reduzir o consumo energético pretende-se que a frequência de relógio deste módulo não ultrapasse 100 MHz. Uma solução consiste em incrementar o paralelismo do processamento, reorganizando o sistema de memória descrito acima de forma a aumentar o ritmo de dados lidos (e escritos) da(s) memória(s). Explique como poderia ser implementada essa solução, admitindo que pode usar várias memórias *dual-port* com organizações variadas e que a sua solução poderá implicar o redesenho do bloco que realiza o interface com as câmaras.

[1.5 valores]

- c) Uma forma de implementar o operador raiz quadrada consiste em aproximar a função por N segmentos de reta para diferentes intervalos do domínio de interesse. Considere que cada segmento de reta aproximante para o intervalo $[x_{s_i}, x_{f_i}]$ é caracterizado pelo declive m_i e ordenada na origem b_i (equação da reta na forma $y = m_i x + b_i$). Apresente um diagrama de blocos devidamente comentado que mostre como implementar esse circuito como um bloco combinacional.

[1.5 valores]

- d) O operador raiz quadrada pode também ser implementado usando o algoritmo CORDIC, sabendo-se que para obter um resultado com N bits são necessárias (aproximadamente) $N+3$ iterações. Diga justificando qual das alternativas para a implementação desse algoritmo seria apropriada nesta aplicação: um circuito iterativo no tempo usando um só bloco que realiza uma iteração do CORDIC por ciclo de relógio, um circuito combinacional construído através da ligação em cascata de vários blocos iteradores CORDIC ou uma versão *pipelined* desse circuito combinacional. Considere que um bloco iterador CORDIC pode operar com uma frequência de relógio até 500 MHz.

-◀ Fim =>-