



EEC0055 - Projeto de Sistemas Digitais

4º ano - 1º semestre

Recurso - 5 de fevereiro de 2020

Duração máxima: 2h30m, sem consulta.

[3 valores]

1 - O projeto de sistemas digitais ao nível RTL, usando linguagens de descrição de hardware digital como Verilog ou VHDL, permitem que o projetista se abstraia, pelo menos parcialmente, dos detalhes da tecnologia alvo em que se pretende implementar o sistema em projeto. Considere duas possíveis tecnologias integradas para a implementação de sistemas digitais: uma tecnologia FPGA semelhante à que utilizou nos trabalhos laboratoriais e uma tecnologia de *standard-cell* destinada à fabricação de um circuito integrado de aplicação específica (ASIC).

[1.5 valores]

a) Existem cenários em que um projetista pode ignorar completamente a tecnologia alvo em que irá implementar o seu circuito digital e outros em que o desenho do circuito e o desenvolvimento do código RTL correspondente é condicionado pela tecnologia alvo. Comente esta afirmação apresentando e justificando devidamente dois exemplos que ilustrem os cenários referidos.

[1.5 valores]

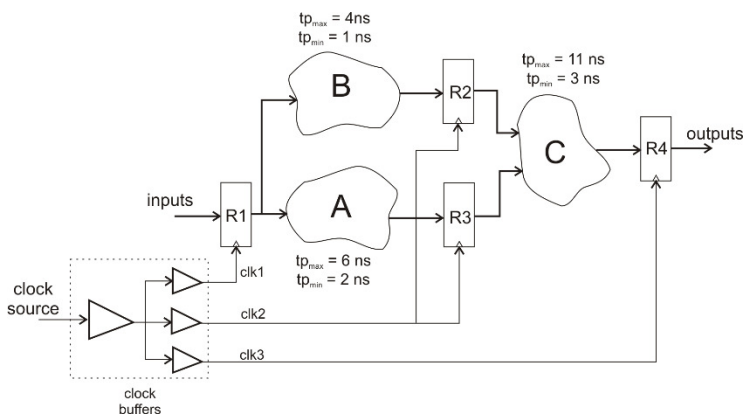
b) Explique se o processo de verificação funcional do código RTL, realizado por simulação lógica, é ou não igualmente condicionado pela tecnologia alvo.

[4 valores]

2 - Num circuito digital síncrono operando com um único sinal global de relógio, o sinal de relógio tem que ser encaminhado desde a sua origem até todos os *flip-flops* do circuito através de uma rede de *buffers* e de ligações dedicadas, de forma a garantir que os sinais que chegam às entradas dos *flip-flops* apresentem uma diferença de fase *suficientemente pequena* para que não afete o comportamento síncrono do circuito (idealmente deveria ser zero).

[1 valor]

a) Considere um circuito síncrono com sinal de relógio, representado pelo seguinte diagrama de blocos. Conhecendo os tempos máximos e mínimos de propagação dos blocos combinacionais A, B e C, indicados na figura, determine a máxima diferença de fase (expressa em percentagem do período ou em ns) que pode ter o sinal de relógio que chega aos registos R2, R3 e R4, em relação ao sinal de relógio que alimenta o registo R1. Considere nesta análise que os tempos de *setup* e de *hold* dos *flip-flops* são iguais a zero e que a frequência do sinal de relógio é igual a 50 MHz (período 20ns).



[1.5 valores]

b) Como é sabido, o maior tempo de propagação dos caminhos combinacionais entre registos determina, aproximadamente, o período mínimo (ou a frequência máxima) do sinal de relógio com que o circuito pode trabalhar. E de que forma pode ser estimada a frequência mínima com que o circuito pode funcionar? Exemplifique para o exemplo dado na alínea anterior.

[1.5 valores]

c) Mostre que ajustando devidamente a fase do sinal *clk3*, em relação à fase do sinal de relógio *clk2* é possível que o circuito opere sincronamente com um sinal de relógio de 100 MHz (período 10 ns, inferior aos 11 ns de tempo máximo de propagação do bloco C).

[3 valores]

3 - A modelização de um circuito combinacional em Verilog usando a declaração `always @*` deve seguir um conjunto de regras de codificação, universalmente consideradas pelas ferramentas de síntese RTL, para que o circuito sintetizado seja realmente um circuito combinacional (i.e. cuja função pode ser descrita por uma tabela de verdade). Considere os 3 excertos de código Verilog seguintes:

i)	ii)	iii)
<pre>always @* begin if (count != maxcount) count = count + 1; else count = 0; end</pre>	<pre>always @* begin Rout = 32'd0 if (Xsel) Rout = Ain; if (Ysel) Rout = Bin; end</pre>	<pre>always @* begin if (maxA & minB) Yout = AAin; else Zout = BBin; end</pre>

[1.5 valores]

a) Apresente um diagrama de blocos que represente um circuito lógico com a mesma funcionalidade descrita por cada um dos excertos de código Verilog. Desenhe o diagrama com base em blocos lógicos padrão, como operadores aritméticos, contadores, codificadores, multiplexadores, registos, etc.

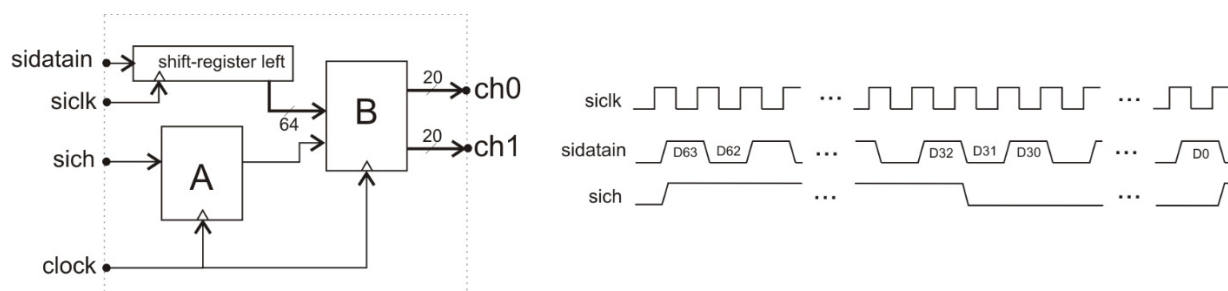
[1.5 valores]

b) Diga justificando se para cada um dos modelos apresentados é sintetizado um circuito realmente combinacional. Se algum dos modelos não resultar num circuito combinacional, explique se, apesar disso, o circuito lógico resultante poderia ser utilizável no contexto do projeto de um circuito digital síncrono com sinal de relógio e mediante que eventuais restrições.

[5.5 valores]

4 - O módulo Verilog listado abaixo representa um circuito síncrono que realiza a interface com um conversor analógico-digital de 2 canais, com frequência de amostragem de 192 kHz e amostras codificadas em 20 bits. O ADC tem uma interface série síncrona com um sinal de relógio proveniente do ADC com frequência igual a 12.288 MHz (este sinal de relógio é derivado de um oscilador de 98.304 MHz que alimenta apenas o ADC). O diagrama temporal mostrado na figura ilustra a relação entre os sinais provenientes da interface série do ADC. Este módulo irá operar com o sinal global de relógio de 200 MHz que alimenta o restante circuito digital em que ele se integra.

A figura mostra um diagrama de blocos simplificado deste circuito e um diagrama temporal que ilustra a operação da interface série do ADC: quando `sich` passa de 0 a 1 inicia-se a transmissão de 64 bits que são apresentados pelo ADC sincronamente com a transição positiva de `siclk` e que devem capturados na transição negativa de `siclk`. Os 20 bits menos significativos do primeiro bloco de 32 bits constituem a amostra do canal 0 (bits D51 a D32 no diagrama temporal) e os 20 bits menos significativos do segundo bloco de 32 bits (bits D19 a D0) formam a amostra capturada pelo canal 1. Sabe-se que o atraso entre a subida de `siclk` e a mudança de `sidatain` e `sich` é inferior a 6 ns.



```
1 : module ADCInterface( input  clock, // Master clock, 200 MHz
2 :                     input  reset, // Master reset, synchronous high
3 :                     input  siclk,  // Serial interface input clock, 12.288 MHz
4 :                     input  sidatain, // Serial interface input serial data
5 :                     input  sich,    // Serial interface input channel select
6 :                     output reg [19:0] ch0, // Output channel 0, synch with clock
7 :                     output reg [19:0] ch1, // Output channel 0, synch with clock
8 :                     );
9 :
10: // input shift-register:
11: reg [63:0] srdatain;
12:
13: // shift left input shift register, synch with negedge of the ADC serial clock:
14: always @(negedge siclk)
15: begin
16:   srdatain <= { srdatain[62:0], sidatain }; // Shift left serial data in
17: end
```

```

18:
19: // ---- Block A ----
20: always @(posedge clock)
21: begin
22:     sich0 <= sich;
23:     sich1 <= sich0;
24:     sich2 <= sich1;
25: end
26: assign S1 = ~sich2 & sich1;
27: // ---- Block A ----
28:
29: // ---- Block B ----
30: always @(posedge clock)
31: if ( S1 )
32: begin
33:     ch0 <= srdatain[51:32];
34:     ch1 <= srdatain[19: 0];
35: end
36: // ---- Block B ----
37:
38: endmodule

```

[1 valor]

- a) Desenhe um diagrama de blocos que represente o circuito cuja descrição RTL corresponde ao código delimitado pelos comentários “// ---- Block A ----”, que implementa o bloco A da figura.

[1.5 valores]

- b) O bloco A tem um papel essencial para o correto funcionamento deste circuito. Explique qual é essa função e diga qual seria a consequência se a declaração na linha 31 fosse implementada como “if (sich)”. Note que se é dito que este bloco tem um papel essencial, isso quer dizer que sem esse bloco o circuito não funcionaria corretamente. Assim, a resposta não deverá ser apenas “*não funcionaria*” mas sim “*não funcionaria porque...*”.

[1.5 valores]

- c) Apesar de ter sido declarado o sinal global de *reset*, este não é usado em nenhum dos processos síncronos (always @(posedge...)) mostrados na descrição RTL. Considere que o sinal *reset* que entra neste módulo está devidamente sincronizado com o sinal global de relógio de 200 MHz.
- i) Explique se essa falta é ou não fundamental para a correta operação do circuito.
- ii) Caso considere que o sinal *reset* deveria ser realmente utilizado, explique como e em que processos.

[1.5 valores]

- d) Admita agora que se pretende reduzir a frequência do sinal global de relógio, mas mantendo-a submúltipla inteira dos 200 MHz atuais (i.e., $F_{\text{clock}} = 200/N$ MHz, considerando os valores possíveis para N entre 2 e 20). Diga e justifique qual a frequência mais baixa do sinal de relógio que pudesse assegurar o correto funcionamento do circuito, tal como está implementado.

[4.5 valores]

- 5 - Pretende-se desenvolver um bloco digital que implemente um temporizador capaz de contar minutos e segundos, semelhante aos temporizadores existentes em equipamentos de cozinha. As saídas do temporizador alimentam um painel LCD com 4 dígitos de 7 segmentos mostrando minutos e segundos. O circuito é constituído por 2 contadores *up/down*, um único descodificador BCD-7 segmentos (multiplexado pelos 4 mostradores) e uma unidade de controlo que recebe 4 entradas provenientes de 4 botões de pressão, para acerto do temporizador e arranque/paragem. O desenvolvimento do projeto deve ser conduzido no sentido de otimizar a área (minimizar a complexidade lógica), não sendo relevante o consumo de energia já que este circuito será usado em equipamentos domésticos permanentemente ligados à rede elétrica.

No sentido de reduzir a complexidade lógica do circuito optou-se por implementar os contadores como contadores BCD (*Binary-Coded Decimal*) formados por 2 contadores binários de 4 bits em que cada um representa um dígito decimal (note que os contadores de minutos e de segundos são semelhantes entre si). Assim, o contador que implementa o dígito das unidades conta módulo 10 (de 0 a 9) e o contador que implementa o dígito das dezenas conta de 0 a 5 quando o contador das unidades atingir o valor máximo (se a contar para cima, em modo de acerto) ou quando atingir o valor mínimo (em modo de decrementar tempo). A figura na página seguinte mostra um diagrama de blocos simplificado deste circuito.

[1.5 valores]

- a) Construa um módulo Verilog sintetizável que implemente o contador BCD de segundos cujo interface é dado abaixo. O sinal de relógio global (*clock*) tem frequência 200 Hz e o sinal de *clock enable* (*cken*) é gerado pela unidade de controlo e deve ser globalmente usado pelos processos síncronos implementados pelo circuito. Esse sinal terá a frequência de 1 Hz quando no modo de contagem regressiva de tempo e terá uma frequência superior quando em modo acerto, determinada pelo estado da unidade de controlo.

```

module countsecBCD (
    input clock,                // 200 Hz clock
    input reset,                // synchronous reset, active high
    input cken,                 // global clock enable, active high
    input updw,                 // up / down control: 1 = up, 0 = down
    output [3:0] sec1,          // counter output, units of second
    output [3:0] sec10         // counter output, tens of second
);

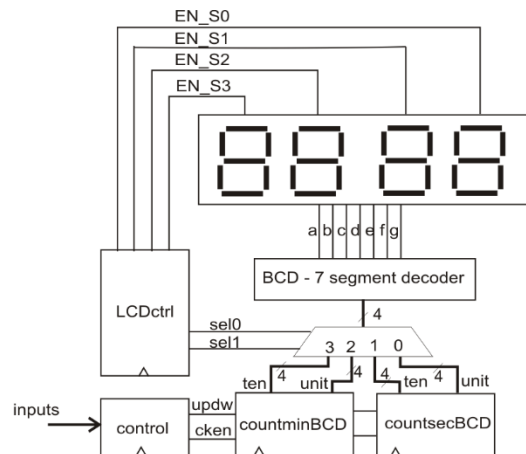
```

[1.5 valores]

- b) A opção de implementar este sistema com contadores decimais permite que as 2 saídas de 4 bits que representam as unidades e as dezenas de cada contador alimentem diretamente o decodificador BCD-7 segmentos que traduz o código binário de 4 bits no código apropriado de 7 bits para o mostrador de 7 segmentos. Mostre que a alternativa de usar um único contador binário de 6 bits para contar de 0 a 59 resultaria num circuito globalmente mais complexo do que a solução proposta. Note que não se pretende que projete esse circuito mas sim que justifique devidamente a razão pela qual o circuito resultante seria mais complexo.

[1.5 valores]

- c) Os valores dos 4 contadores são apresentados rotativamente nas entrada do único decodificador BCD - 7 segmentos através do multiplexador 4÷1, sincronamente com a ativação do dígito respetivo (sinais EN_S0 a EN_S3 , ligam o dígito respetivo quando são iguais a 1), sendo cada dígito do mostrador ativado durante 20 ms. Esta operação é implementada pelo bloco **LCDctrl**, mostrado no esquema da figura. Desenvolva um módulo Verilog que implemente este controlador, procurando minimizar a complexidade lógica do circuito.



◀ Fim ▶