



EEC0055 - Projeto de Sistemas Digitais

4º ano - 1º semestre

Exame - 23 de janeiro de 2019

Duração máxima: 2h30m, sem consulta.

[4 valores]

1 - Ao longo do projeto de um sistema digital para tecnologias digitais integradas, semelhantes à que foi usada nos trabalhos laboratoriais, são realizadas duas transformações fundamentais (recorrendo a ferramentas de projeto) em que o modelo do sistema é traduzido do código fonte construído pelo projetista (p.ex. Verilog) até uma descrição que representa a sua organização ao nível físico. Explique:

[2 valores]

a) Qual é a primeira dessas transformações, que tem por origem o código fonte Verilog, referindo-se ao resultado que é produzido e às razões pelas quais a execução desta tarefa pode obrigar ao re-desenho do código fonte RTL.

[2 valores]

b) Qual é a segunda dessas tarefas que produz a representação do circuito ao nível físico, referindo-se também aos motivos que podem obrigar ao re-desenho do código fonte RTL, ou à re-execução da tarefa anterior, depois de realizar esta transformação.

[4 valores]

2 - Considere as ferramentas de projeto e de simulação que foram usadas nos trabalhos laboratoriais e um cenário semelhante ao do 3º projeto laboratorial, em que teve de desenhar um bloco que foi mais tarde integrado num outro projeto.

[2 valores]

a) Imagine que o processo de verificação funcional do código Verilog que implementa o seu bloco foi completado com sucesso (antes deste ser integrado no sistema). No entanto, depois da sua integração no projeto final e após completar a implementação verificou-se o sistema não cumpre a função esperada e que a falha se deve ao seu bloco e/ou à forma como foi integrado no resto do sistema. Apresente exemplos de motivos que podem explicar essa situação (pelo menos 2) e diga como poderiam ser identificados os problemas que refere.

[2 valores]

b) Uma das especificações que lhe foram definidas para o bloco que projetou foi que o sistema deveria ser síncrono com um único sinal de relógio de 150 MHz. Diga, justificando, em que fase do processo de projeto é possível verificar se esse requisito é ou não satisfeito.

[4 valores]

- 3 - A construção de modelos Verilog para alimentar processos de síntese RTL deve seguir um conjunto de regras de codificação de circuitos combinacionais e sequenciais síncronos, que são universalmente aceites pelas ferramentas de síntese. Os dois excertos de código Verilog seguintes, que pretendem representar um circuito sequencial e outro combinacional, violam regras elementares de codificação (admita que os sinais referidos estão devidamente declarados):

(i)	(ii)
<pre>always @* begin if (clken) begin if (~muxsel) zsum = ra + rb[12:0]; else zdiff = rb - ra[12:0]; end else begin zsum = 0; zdiff = 0; end end end</pre>	<pre>always @(posedge clock or posedge rst) begin if (rst) zout = output1; ztemp = Ra * Ra - Rb; zout = ztemp * 1.34; end</pre>

[2 valores]

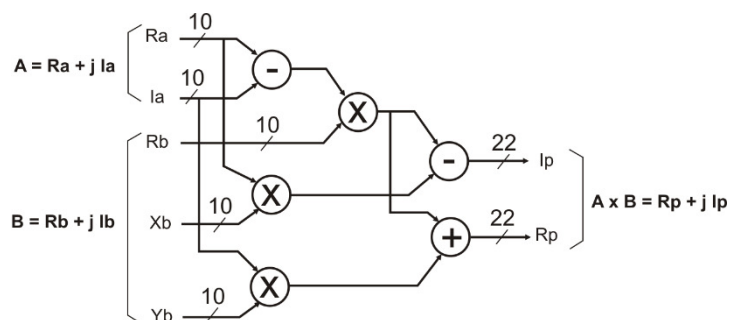
- a) Diga justificando quais os erros que identifica no código que pretende modelizar um circuito combinacional e explique como o deveria corrigir.

[2 valores]

- a) Diga justificando quais são os erros no código que pretende modelizar um circuito sequencial e diga como corrigir.

[4 valores]

- 4 - O diagrama da figura mostra uma implementação do produto de 2 números complexos **A** e **B** dados na forma $R + jI$. O operando $B = R_b + jI_b$ é representado pelas 3 componentes **Rb**, **Xb** = $R_b + I_b$ e **Yb** = $R_b - I_b$, o que permite implementar o produto complexo com apenas 3 multiplicadores em vez dos 4 que seriam necessários usando o cálculo direto de $(R_a + jI_a) \times (R_b + jI_b) = (R_a \times R_b - I_a \times I_b) + j(I_a \times R_b + R_a \times I_b)$. Os 5 operandos de entrada são inteiros com sinal de 10 bits e os resultados **Rp** e **Ip** (parte real e parte imaginária) são também inteiros com sinal de 22 bits.



[2.5 valores]

- a) Construa um módulo Verilog (sintetizável) que realize o cálculo representado pelo diagrama da figura, onde os operadores aritméticos são implementados como circuitos combinacionais. Devem ser criados 5 registos nas entradas, para os 5 operandos, e 2 registos para os dois resultados. O *datapath* deve ser *pipelined* de forma a que a frequência máxima de relógio seja condicionada pelo tempo máximo de propagação dos multiplicadores.

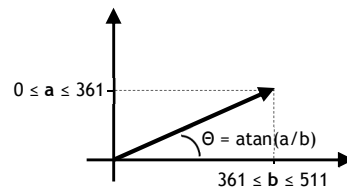
[1.5 valores]

- b) Depois de implementar esse circuito verificou-se que a frequência máxima de relógio não ultrapassava os 160 MHz (período de 6.3 ns), quando o circuito necessita de operar com uma frequência mínima de relógio de 200 MHz (período 5 ns). Para resolver isso será necessário acrescentar estágios de *pipeline* adicionais de forma a dividir o circuito lógico combinacional que implementa o multiplicador. Explique como implementaria essa solução recorrendo a diagramas ou código Verilog para suportar a sua explicação.

[4 valores]

5 - No projeto dum sistema digital para processar os sinais de um sensor de posição angular é necessário implementar um calculador da função $\text{atan}(a/b)$, apenas para o primeiro octante. O sistema em que esse projeto se insere destina-se a ser implementado como um circuito integrado de aplicação específica (ASIC) e é objetivo global do projeto a minimização da sua área.

Os operandos a e b são inteiros positivos de 9 bits e representam, respetivamente, as componentes segundo Y e X de um vetor cujo ângulo se pretende calcular e que tem módulo aproximadamente igual a 511. Assim, os valores de a e b são condicionados pelas relações seguintes: $0 \leq a \leq 361$ e $361 \leq b \leq 511$:



O bloco a desenvolver deverá ser capaz de processar dados apresentados continuamente ao ritmo de 100 kHz e poderá operar com um sinal de relógio com frequência máxima de 2 MHz. O resultado produzido deverá ser um inteiro de 9 bits representando o ângulo $\text{atan}(a/b)$ em unidades de décima de grau (entre 0 e 450). O atraso entre a apresentação dos operandos a e b e a geração do resultado não pode exceder 20μs (dois períodos de 100 kHz ou 40 períodos do relógio de 2 MHz).

Numa primeira reunião da equipa de projeto, 3 colegas propuseram implementações alternativas para este módulo que apresentaram da seguinte forma:

- 1) *A minha proposta é muito simples e escreve-se em apenas 2 linhas de código: primeiro calcula-se o quociente a/b com `assign x = a / b`; e depois obtém-se o valor da função $\text{atan}()$ recorrendo a uma tabela, escrevendo `assign atan = ROM_ATAN[x]`; onde `ROM_ATAN[]` é uma tabela implementada como uma memória pré-carregada com os valores de $\text{atan}(x)$.*
- 2) *Concordo com essa ideia geral [calcular primeiro o quociente a/b e usar depois uma tabela], mas como é possível usar até 40 ciclos de relógio para calcular o resultado tenho uma solução melhor que permite reduzir substancialmente a área/complexidade do circuito.*
- 3) *Não concordo com as vossas propostas. Há uma outra solução ainda mais simples do que de 2), que também faz uso de alguns dos 40 ciclos de relógio disponíveis entre cada dois conjunto de de operandos e que só necessita de uma memória pequena, com menos de 16 palavras.*

[1.5 valores]

a) Coloque-se no papel do projetista 2). Explique em que consistirá a solução que ele apresenta e diga porque razão é mais simples da que foi proposta por 1).

[1.5 valores]

b) Assuma agora o papel do projetista 3). Diga, justificando, qual será a solução a que se refere.

[1 valor]

c) Com o objetivo de procurar reduzir o consumo de energia, a frequência máxima de relógio foi redefinida para apenas 100 kHz e por isso o circuito terá de produzir um resultado em no máximo 2 ciclos de relógio. Explique se nestas condições poderá ser atingido o objetivo de reduzir o consumo de energia (apenas relativa a esse bloco), tendo em atenção que neste novo cenário as soluções propostas por 2) e 3) já não podem ser adotadas (mas note que isso não significa que tenha de ser seguida a solução proposta pelo 1) !)

◀= Fim =▶