



## Projeto de Sistemas Digitais

4º ano - 1º semestre  
Recurso - 7 fevereiro 2014

Duração máxima: 2h30m sem consulta de apontamentos

[2 valores]

1 - Imagine que desenvolveu um modelo Verilog de um sistema digital sequencial síncrono, mas ao realizar a verificação funcional concluiu rapidamente que não produzia as saídas corretas. Apesar disso, e porque não estava muito confiante nos seus conhecimentos de Verilog, resolveu realizar a síntese RTL desse código para confirmar que era sintetizável e para obter uma primeira estimativa da área ocupada e da frequência máxima admitida para o sinal de relógio. A síntese correu sem erros, e como já tinha construído o modelo de verificação, realizou uma simulação do modelo pós-síntese e constatou que passou todas as verificações, mesmo depois de ter aplicado um conjunto muito completo de estímulos de simulação. Sem perceber bem o que se tinha passado, experimentou realizar então a implementação final e testar o circuito numa plataforma FPGA e pode concluir então que o sistema funcionava na perfeição! Relatou o sucedido ao seu chefe e ele disse logo que o mais provável era o código estar mal escrito, apesar de ter sintetizado e funcionado bem (e mandou-o para casa estudar...). Explique o que pode ter corrido mal, identificando e justificando pelo menos uma razão responsável por essa situação.

[2 valores]

2 - As ferramentas de síntese RTL, como o XST da XILINX usado nos trabalhos laboratoriais, realizam diversas otimizações ao longo do processo de transformação do código RTL (Verilog) para um circuito lógico. Mudando o objetivo de otimização de “*area*” para “*speed*”, é possível que o circuito sintetizado (admitindo que é um circuito sequencial) veja reduzido o número de ciclos de relógio necessários para realizar a sua operação? Justifique: se sim, explique como; se não, diga porquê e de que forma é então otimizada a rapidez.

[6 valores]

3 - O código Verilog seguinte descreve um circuito sequencial síncrono que calcula uma operação aritmética entre os valores aplicados nas 3 entradas x1, x2, x3.

```
module psdrecurso(clock, restart, x1, x2, x3, dout);
input clock, restart;
input [11:0] x1, x2, x3;
output [31:0] dout;
reg [31:0] dout, r1, r2, r3;
reg [11:0] xr1, xr2, xr3;
always @(posedge clock)
if (restart)
begin
xr1 <= 12'd0; xr2 <= 12'd0; xr3 <= 12'd0;
dout <= 32'd0; r1 <= 32'd0; r2 <= 32'd0; r3 <= 32'd0;
end
else
begin
xr1 <= x1; xr2 <= x2; xr3 <= x3;
r1 <= xr1 * xr2; r3 <= xr3;
r2 <= r1 + r3; dout <= dout + r2;
end
endmodule
```

[2 valores]

a) Desenhe um diagrama de blocos que mostre o circuito lógico representado por este modelo.

[2 valores]

b) Este circuito é usado para operar 3 vetores com 100 elementos que são aplicados em sequência nas entradas x1, x2 e x3 ao mesmo ritmo do sinal de relógio. Diga, justificando, quantos ciclos de relógio são necessários para completar o processamento daquele conjunto de dados.

[2 valores]

- c) Imagine que depois da implementação a frequência máxima estimada para o sinal de relógio é de 200 MHz. Sabendo que a saída deve ser apresentada num tempo inferior a 100 ns depois da aplicação das entradas, diga quais as alterações deveria fazer ao módulo apresentado e ao sistema em que ele será usado para obter uma solução com um consumo de energia mais baixo do que a solução apresentada.

[6 valores]

- 4 - No projeto de um sensor angular foi necessário desenvolver um módulo para calcular a função arco-tangente (  $\text{atan}()$  ). A entrada para esse sistema provém de um conjunto de sensores que produz dois valores X e Y, representando as componentes cartesianas de um vetor cujo ângulo se pretende determinar. O sistema de sensores também já produz o quociente Y/X, que será o argumento a usar para o cálculo da função  $\text{atan}()$  relativa apenas ao 1º octante. Para este caso esse quociente é um valor entre 0 e 1 e o resultado da função  $\text{atan}()$  entre 0 e 45º (o processo de cálculo para os outros octantes será semelhante).

[1,5 valores]

- a) Escreva um processo combinacional em Verilog ( `assign @* ...` ) que codifique em 3 bits o octante a que pertence o vetor, tendo por entrada os sinais **x** e **y**, representando valores inteiros em sinal e grandeza com 9 bits. A saída deverá ser representada no sinal **octant** de 3 bits que codifica o 1º octante como 000, o segundo como 001 e o 8º octante como 111. Deverá procurar minimizar a complexidade lógica.

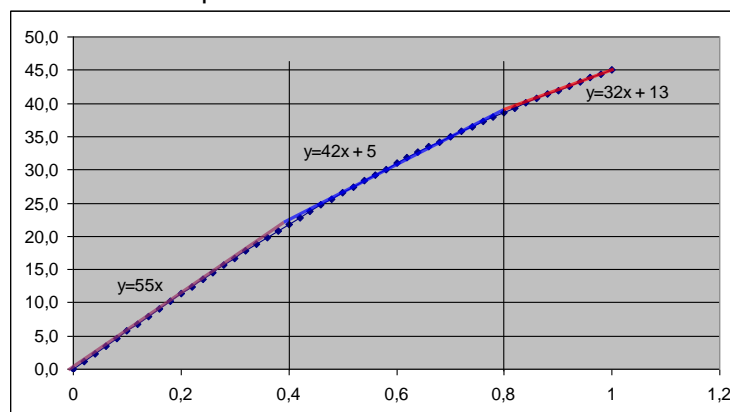
[3 valores]

- b) Projete agora um bloco que recebe na entrada o quociente Y/X (já calculado), relativo a um vetor no 1º octante tal como foi referido acima, e calcula o ângulo (em graus inteiros entre 0 e 45, como uma palavra de 6 bits) usando uma aproximação da função  $\text{atan}()$  que recorre a um conjunto de 3 segmentos lineares, da forma como se mostra na figura. Os pontos limites no domínio para cada segmento são 0.4 (em binário 0.011001100110011) e 0.8 (em binário 0.110011001100110). O argumento de entrada (Y/X) é um número de 16 bits positivo com 1 bit para a parte inteira e 15 bits para a parte fracionária. Os declives dos 3 segmentos de reta têm as representações binárias seguintes:

$$55 = 110111 \quad 42 = 101010 \quad 32 = 100000$$

Admite-se que a saída pode ter um erro absoluto de 1º e por isso o resultado do cálculo daquelas funções lineares pode ser truncado para valores inteiros, não sendo necessário implementar processos de arredondamento.

Desenvolva este módulo como um circuito combinacional, sem recorrer ao operador produto e procurando minimizar a complexidade lógica (custo área). As multiplicações por constantes devem ser implementadas de forma adequada e se o entender conveniente recorra à recodificação de Booth das constantes referidas para reduzir a complexidade do circuito.



Aproximação por troços lineares para a função  $\text{atan}()$  no domínio  $[0,1]$ .

[1,5 valores]

- c) Considere agora que o sistema de controlo do sensor tem de realizar aquele cálculo com uma frequência de 100 Hz (10 ms de período), dispondo para isso de um sinal de relógio de 1 MHz. Admitindo que o sistema pode agora ser realizado como um circuito sequencial, explique de que forma abordaria o problema com o objetivo de minimizar a área ocupada. Note que a questão se refere à implementação da aproximação de  $\text{atan}()$  por troços lineares que é descrita aqui e por isso a solução que se espera não é recorrer ao algoritmo CORDIC, que também poderia ser usado para calcular esta função mas seguindo um processo diferente.

---

[4 valores]

5 - Considere que numa certa tecnologia digital as células básicas *full-adder* têm um tempo de propagação igual a 500 ps, de qualquer entrada para qualquer saída.

[2 valores]

a) Quais são os tempos de propagação máximo e mínimo de um somador *ripple-carry* de 32 bits (sem considerar *carry-in* nem *carry-out*) construído com aqueles *full-adders*?

[2 valores]

b) Qual é o tempo de propagação máximo do circuito combinacional que é criado como resultado da síntese RTL da declaração Verilog `assign y=(a+b)+(c+d)` , considerando que todos os sinais envolvidos são de 32 bits?

: -)) - FIM - ((-: