PMR3303 – Eletrônica Digital para Mecatrônica
Prof. Dr. Celso M. Furukawa – Lista 8 (rev. A) – Entrega até 29/4/2019

Os exercícios E1 a E6 a seguir se referem à Unidade Lógico-Aritmética descrita na apostila da experiência 4. Resolva os exercícios E1 a E4 usando mapas de Veitch-Karnaugh. Nos mapas, indique os "uns notáveis".
Para verificar a sintaxe dos seus módulos Verilog e simulá-los, sugerimos dois ambientes online Crie um login para poder salvar seus arquivos (gratuito).

- JDoodle: https://www.jdoodle.com/execute-verilog-online (módulos a simular e testbench devem ocupar um único arquivo)
- EDAplayground: https://www.edaplayground.com/home (permite salvar o *testbench* em um arquivo separado dos módulos a simular, mas estes últimos devem estar todos no mesmo arquivo). Nota: para salvar seu projeto com outro nome, clique em ***Share***; para recarregar um projeto, clique em ***Playground***.

E1. Determine uma expressão simplificada na forma de soma de produtos da saída x_i da célula básica LE1 do Extensor Lógico.

E2. O exercício anterior possui duas soluções de mesmo custo: ou seja, envolvem o mesmo número de produtos na soma, e os produtos possuem o mesmo número de termos. Determine a expressão alternativa de resposta ao exercício anterior.

E3. Determine uma expressão simplificada na forma de soma de produtos da saída y_i da célula básica AE1 do Extensor Aritmético.

E4. Determine uma expressão simplificada na forma de soma de produtos da saída c_{0E} do Extensor de *Carry-in*. Transcreva a expressão para Verilog, usando uma atribuição do tipo *assign*.

E5. Escreva um módulo em Verilog de nome LAE1 que combie as funções das células básicas LE1 e AE1. Ou seja, o módulo deve ter entradas M, s[1:0], ai e bi, e gerar as saídas xi e yi, conforme o símbolo mostrado na figura FE5. Use atribuições do tipo *assign*.

E6. Suponha que a ULA de 3 bits mostrada na figura FE6a tenha uma arquitetura que usa o somador completo mostrado na figura FE6a em configuração *ripple-carry*. Desenhe o diagrama lógico dessa ULA, usando também a célula básica LAE1 da figura FE5a e o extensor CE definido na apostila e descreva o circuito em um módulo Verilog ULA3.v que instancie os módulos LAE1, FA1 e CE.

E7. A figura FE7 mostra um decodificador 2:4 (2 bits de endereço, 4 saídas) com *enable* (_E) e saídas (_Y_i) ativas em zero: quando _E = 1, todas as saídas permanecem em 1, e com _E = 0, somente a saída _Y_i endereçada por A_1A_0 vai a 0. Usando descrição procedural (bloco "*always*") e estrutura do tipo *for*, implemente o decodificador em um módulo Verilog definido por:
Decod_2_4 (input _E, input [1:0] A, output [3:0] _Y)

E8. Repita o exercício anterior usando agora estrutura do tipo *case* para implementar o módulo Decod_2_4 (input _E, input [1:0] A, output [3:0] _Y)

E9. Implemente em Verilog um módulo Decod_2_4_sim( ) que instancie o decodificador da figura FE7 e que contenha um bloco ("*initial*") para simulá-lo, varrendo todas as 8 combinações possíveis de entrada.

E10. Instanciando 2 módulos da figura FE7, implemente um decodificador 3:8 com saídas e enable ativos em zero, em módulo definido por:
Decod_3_8(input _E, input [2:0] A, output [7:0] _Y).

E11. Instanciando 5 módulos da figura FE7, implemente um decodificador 4:16 com saídas e enable ativos em zero, em um módulo definido por:
Decod_4_16(input _E, input [3:0] A, output [15:0] _Y).

E12. Escreva um módulo *testbench* em Verilog para simular a ULA3 da figura FE6a por meio de um **vetor de teste** (ou seja, não simule todas as combinações possíveis de entrada – que seriam... 1024!). Teste alguns exemplos de operação da ULA – por exemplo, as operações indicadas na Atividade 6 da apostila da experiência 4.

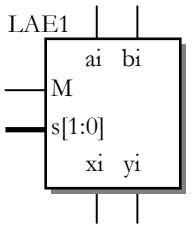

Figura FE5

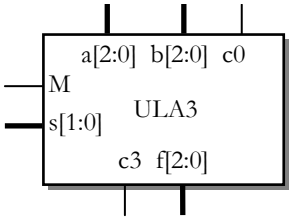

Figura FE6a

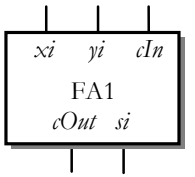

Figura FE6b

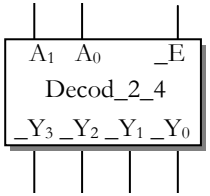

Figura FE7

Algumas respostas
E2) $M.a_i'.b_i.s_1$ pode ser substituído por $M.b_i.s_1.s_0'$