

# LÓGICA DIGITAL (1001351)

## EXPERIMENTO NR.7

### Circuitos Sequenciais: Atribuições [não-]blocantes <sup>1</sup>

---

## 1 Instruções Gerais

- Grupos definidos no AVA, só incluir os nomes de quem efetivamente participou;
- Ler atentamente todo o procedimento desta experiência antes de realizá-la;
- Construir um desenho esquemático completo de cada um dos quatro circuitos, incluindo nomes para os blocos e ligações, antes de codificar;

## 2 Objetivos da Prática

- Implementação de circuitos sequenciais que fazem uso de latches, flip-flops ou registradores;
- Comparação de comportamentos de circuitos combinacionais com circuitos sequenciais;
  - Observação de que os circuitos fornecidos não apresentam saídas diferentes com a mesma entrada, dependendo da sequência de entradas, já que não possuem realimentação (i.e. a mesma variável dos dois lados de uma expressão);
  - Modificação dos circuitos fornecidos para circuitos sequenciais;
- Demonstrar como as atribuições blocantes e não-blocantes são interpretadas pelas ferramentas de síntese de hardware;
- Demonstrar que as transições de valores nos circuitos sequenciais ocorrem sempre de forma síncrona com o clock;
- Simulação do projeto realizado através de um arquivo de *test bench* e análise de suas saídas ao longo do tempo (análise temporal);
- Opcionalmente, implementação do projeto no Kit de desenvolvimento Zybo Z7-20.

---

<sup>1</sup>Revisão 08/10/2021: Prof. Ricardo Menotti, Prof. Maurício Figueiredo

### 3 Procedimentos Experimentais

1. Para cada um dos códigos a seguir, forneça o diagrama esquemático correspondente (observe que `=` é diferente de `<=`):

<p>i) <pre> 1 module foo1 (a, b, c, clk, x, y, z); 2   input a, b, c, clk; 3   output reg x, y; 4   output z; 5   always @(posedge clk) 6   begin 7     x &lt;= ~c; 8     y &lt;= ~(x &amp; a &amp; b); 9   end 10  assign z = ~c; 11 endmodule </pre></p>	<p>ii) <pre> 1 module foo2 (a, b, c, clk, x, y, z); 2   input a, b, c, clk; 3   output reg x, y; 4   output z; 5   always @(posedge clk) 6   begin 7     x = ~c; 8     y = ~(x &amp; a &amp; b); 9   end 10  assign z = ~c; 11 endmodule </pre></p>
--	---

<p>I) <pre> 1 module bar1 (a, b, c, clk, x, y, z); 2   input a, b, c, clk; 3   output reg x, y, z; 4   always @(posedge clk) 5   begin 6     x &lt;= a ^ c; 7     y &lt;= ~(x &amp; c); 8     z &lt;= b   y; 9   end 10 endmodule </pre></p>	<p>II) <pre> 1 module bar2 (a, b, c, clk, x, y, z); 2   input a, b, c, clk; 3   output reg x, y, z; 4   always @(posedge clk) 5   begin 6     x = a ^ c; 7     y = ~(x &amp; c); 8     z = b   y; 9   end 10 endmodule </pre></p>
--	---

2. Observando que os pares de circuitos possuem as mesmas expressões, explique a diferença das atribuições blocantes e não-blocantes na geração do circuito;
3. Implemente um *test bench* que instancie os quatro circuitos e forneça as mesmas entradas para eles, observando e analisando as saídas do circuitos;
  - Ele deve conter um sinal de clock e as entradas fornecidas devem ser modificadas em sua **borda de descida**;
  - Isso é para que as entradas estejam estáveis no momento das transições de saída geradas pelo clock, já que os circuitos respondem à **borda de subida**;
4. No primeiro par de circuitos, `x` e `z` são complementos da entrada `c`. Explique por que suas saídas evoluem no tempo de forma diferente;
5. Modifique as expressões de um par de circuitos para que as saídas sejam usadas novamente como entradas das expressões;
  - Tente diferentes combinações até que consiga fazer com que o circuito apresente saídas diferentes a partir da mesma entrada;

6. Elaborar relatório simplificado consistindo de: página de rosto (com identificação da prática e integrantes entre outras informações, tal como no relatório padrão) e seção de resultados contendo: diagramas esquemáticos dos circuitos, link para a implementação no EDAPlayground, diagrama de formas de onda (*waveform*) da simulação completa, **captura de imagem da placa (opcional)** e comentários com as explicações solicitadas.
7. Envie o relatório em PDF no AVA.