

# Lógica Digital (1001351)



## Circuitos Sequenciais: Implementação em Verilog

---

Prof. Ricardo Menotti

[menotti@ufscar.br](mailto:menotti@ufscar.br)

Prof. Luciano de Oliveira Neris

[lneris@ufscar.br](mailto:lneris@ufscar.br)

Atualizado em: 1 de abril de 2024

**Departamento de Computação**

Centro de Ciências Exatas e de Tecnologia

Universidade Federal de São Carlos

# Circuitos Sequenciais

---

```
1 module D_latch (D, Clk, Q);  
2   input D, Clk;  
3   output reg Q;  
4  
5   always @(D, Clk)  
6     if (Clk)  
7       Q = D;  
8  
9 endmodule
```

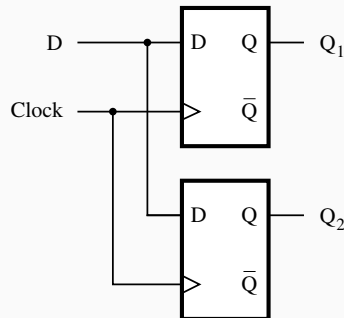
```
1 module flipflop (D, Clock, Q);  
2   input D, Clock;  
3   output reg Q;  
4  
5   always @(posedge Clock)  
6     Q = D;  
7  
8 endmodule
```

# Atribuições blocantes

```
1 module example5_3 (D, Clock, Q1, Q2);  
2     input D, Clock;  
3     output reg Q1, Q2;  
4  
5     always @(posedge Clock)  
6     begin  
7         Q1 = D;  
8         Q2 = Q1;  
9     end  
10  
11 endmodule
```

# Atribuições blocantes

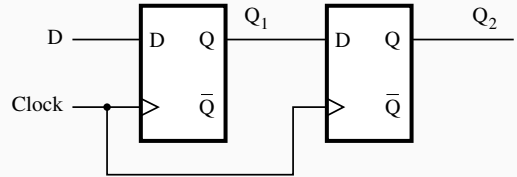
```
1 module example5_3 (D, Clock, Q1, Q2);  
2   input D, Clock;  
3   output reg Q1, Q2;  
4  
5   always @(posedge Clock)  
6   begin  
7     Q1 = D;  
8     Q2 = Q1;  
9   end  
10  
11 endmodule
```



**Figure 5.37** Circuit for Example 5.3.

# Atribuições não-blocantes

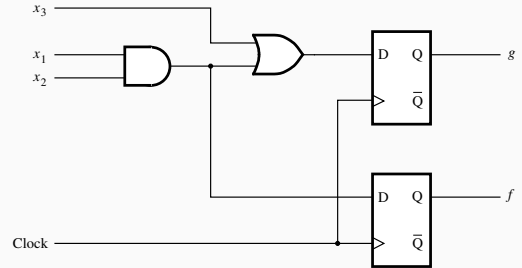
```
1 module example5_4 (D, Clock, Q1, Q2);  
2   input D, Clock;  
3   output reg Q1, Q2;  
4  
5   always @(posedge Clock)  
6   begin  
7     Q1 <= D;  
8     Q2 <= Q1;  
9   end  
10  
11 endmodule
```



**Figure 5.39** Circuit defined in Figure 5.38.

# Atribuições blocantes

```
1 module example5_5 (x1, x2, x3, Clock, f, g);
2   input x1, x2, x3, Clock;
3   output reg f, g;
4
5   always @(posedge Clock)
6   begin
7     f = x1 & x2;
8     g = f | x3;
9   end
10
11 endmodule
```

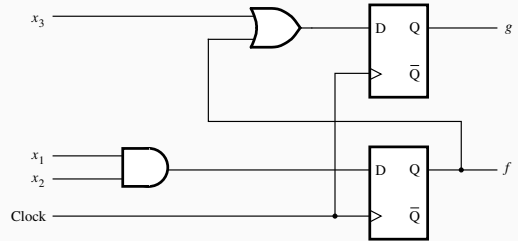


**Figure 5.41** Circuit for Example 5.5.



# Atribuições não-blocantes

```
1 module example5_6 (x1, x2, x3, Clock, f, g);  
2   input x1, x2, x3, Clock;  
3   output reg f, g;  
4  
5   always @(posedge Clock)  
6   begin  
7     f <= x1 & x2;  
8     g <= f | x3;  
9   end  
10  
11 endmodule
```



**Figure 5.43** Circuit for Example 5.6.

# Assíncrono vs Síncrono

```
1 module flipflop (D, Clock, Resetn, Q);
2   input D, Clock, Resetn;
3   output reg Q;
4
5   always @(negedge Resetn, posedge Clock)
6     if (!Resetn)
7       Q <= 0;
8     else
9       Q <= D;
10
11 endmodule
```

```
1 module flipflop (D, Clock, Resetn, Q);
2   input D, Clock, Resetn;
3   output reg Q;
4
5   always @(posedge Clock)
6     if (!Resetn)
7       Q <= 0;
8     else
9       Q <= D;
10
11 endmodule
```

# Contador up/down com carga e enable

```
1 module updowncount (R, Clock, L, E, up_down, Q);
2   parameter n = 8;
3   input [n-1:0] R;
4   input Clock, L, E, up_down;
5   output reg [n-1:0] Q;
6
7   always @(posedge Clock)
8     if (L)
9       Q <= R;
10    else if (E)
11      Q <= Q + (up_down ? 1 : -1);
12
13 endmodule
```

- Brown, S. & Vranesic, Z. - Fundamentals of Digital Logic with Verilog Design, 3rd Ed., Mc Graw Hill, 2009

# Lógica Digital (1001351)



## Circuitos Sequenciais: Implementação em Verilog

---

Prof. Ricardo Menotti

[menotti@ufscar.br](mailto:menotti@ufscar.br)

Prof. Luciano de Oliveira Neris

[lneris@ufscar.br](mailto:lneris@ufscar.br)

Atualizado em: 1 de abril de 2024

**Departamento de Computação**

Centro de Ciências Exatas e de Tecnologia

Universidade Federal de São Carlos