

Pontifícia Universidade Católica de Minas Gerais  
Instituto de Ciências Exatas e Informática – ICEI  
Arquitetura de Computadores I

ARQ1 \_ Aula\_09

Tema: Introdução à linguagem Verilog

Links recomendados

<https://www.youtube.com/watch?v=o6kS7izbM7o>

[http://www.asic-world.com/verilog/art\\_testbench\\_writing2.html](http://www.asic-world.com/verilog/art_testbench_writing2.html)

[http://referencedesigner.com/tutorials/verilogexamples/verilog\\_ex\\_06.php](http://referencedesigner.com/tutorials/verilogexamples/verilog_ex_06.php)

[http://www.testbench.in/TB\\_08\\_CLOCK\\_GENERATOR.html](http://www.testbench.in/TB_08_CLOCK_GENERATOR.html)

Orientação geral:

Apresentar uma forma de solução em formato texto (.txt).

Programas em Verilog poderão ser entregues em formato (.v) com previsão de testes.

Os arquivos para simulação em Logisim (.circ) poderão ser entregues como complementação, deverão ser identificados internamente e entregues, acompanhados (ou não) de figuras equivalentes exportadas pela ferramenta.

Atividade: Circuitos sequenciais

01.) Projetar e descrever em Verilog um módulo gerador de **clock**.

O nome do arquivo deverá ser Exemplo\_0901.v, e poderá seguir o modelo descrito abaixo.

Incluir previsão de testes e verificação da carta de tempo usando GTKWave.

```
// -----  
// -- test clock generator (1)  
// -----
```

```
module clock ( output clk );  
    reg    clk;
```

```
    initial  
    begin  
        clk = 1'b0;  
    end
```

```
    always  
    begin  
        #12 clk = ~clk;  
    end
```

```
endmodule // clock ( )
```

```
module Exemplo_0901;

  wire clk;
  clock CLK1 ( clk );

  initial begin
    $dumpfile ( "Exemplo_0801.vcd" );
    $dumpvars;

    #120 $finish;
  end

endmodule // Exemplo_0901 ( )
```

- 02.) Projetar e descrever em Verilog módulos geradores de pulso (**pulse**) e gatilho (**trigger**).  
O nome do arquivo deverá ser Exemplo\_0902.v, e poderá seguir o modelo descrito abaixo.  
Incluir previsão de testes e verificação da carta de tempo usando GTKWave.

```
// -----  
// -- test clock generator (2)  
// -----  
  
module clock ( output clk );  
    reg    clk;  
  
    initial  
    begin  
        clk = 1'b0;  
    end  
  
    always  
    begin  
        #12 clk = ~clk;  
    end  
endmodule  
  
module pulse ( signal, clock );  
    input  clock;  
    output signal;  
    reg    signal;  
  
    always @ ( clock )  
    begin  
        signal = 1'b1;  
        #3 signal = 1'b0;  
        #3 signal = 1'b1;  
        #3 signal = 1'b0;  
    end  
endmodule // pulse  
  
module trigger ( signal, on, clock );  
    input  on, clock;  
    output signal;  
    reg    signal;  
  
    always @ ( posedge clock & on )  
    begin  
        #60 signal = 1'b1;  
        #60 signal = 1'b0;  
    end  
endmodule // trigger
```

```

module Exemplo_0902;

    wire clock;
    clock clk ( clock );

    reg p;

    wire p1,t1;

    pulse pulse1 ( p1, clock );
    trigger trigger1 ( t1, p, clock );

    initial begin
        p = 1'b0;
    end

    initial begin
        $dumpfile ( "Exemplo0802.vcd" );
        $dumpvars ( 1, clock, p1, p, t1 );

        #060 p = 1'b1;
        #120 p = 1'b0;
        #180 p = 1'b1;
        #240 p = 1'b0;
        #300 p = 1'b1;
        #360 p = 1'b0;
        #376 $finish;
    end

endmodule // Exemplo_0902

```

- 03.) Projetar e descrever em Verilog módulos geradores de pulso (**pulse**) com períodos diferentes. O nome do arquivo deverá ser Exemplo\_0903.v, e poderá seguir o modelo descrito a seguir. O gerador de **clock** do Exemplo0801.v deverá ser previamente isolado em um arquivo único cujo nome deverá ser **clock.v**, para uso posterior. Incluir previsão de testes e verificação da carta de tempo usando GTKWave.

```

// -----
// -- test clock generator (3)
// -----

`include "clock.v"

module pulse1 ( signal, clock );
input  clock;
output signal;
reg    signal;

always @ ( posedge clock )
begin
    signal = 1'b1;
    #4 signal = 1'b0;
    #4 signal = 1'b1;
    #4 signal = 1'b0;
    #4 signal = 1'b1;
    #4 signal = 1'b0;
end
endmodule // pulse

module pulse2 ( signal, clock );
input  clock;
output signal;
reg    signal;

always @ ( posedge clock )
begin
    signal = 1'b1;
    #5 signal = 1'b0;
end
endmodule // pulse

module pulse3 ( signal, clock );
input  clock;
output signal;
reg    signal;

always @ ( negedge clock )
begin
    signal = 1'b1;
    #15 signal = 1'b0;
    #15 signal = 1'b1;
end
endmodule // pulse

module pulse4 ( signal, clock );
input  clock;
output signal;
reg    signal;

always @ ( negedge clock )
begin
    signal = 1'b1;
    #20 signal = 1'b0;
    #20 signal = 1'b1;
    #20 signal = 1'b0;
end
endmodule // pulse

```

```

module Exemplo_0903;

    wire clock;
    clock clk ( clock );

    wire p1,p2,p3,p4;

    pulse1 pls1 ( p1, clock );
    pulse2 pls2 ( p2, clock );
    pulse3 pls3 ( p3, clock );
    pulse4 pls4 ( p4, clock );

    initial begin
        $dumpfile ( " Exemplo0803.vcd" );
        $dumpvars ( 1, clock, p1, p2, p3, p4 );

        #480 $finish;
    end

endmodule // Exemplo_0903

```

- 04.) Projetar e descrever em Verilog um módulo gerador de pulso (**pulse**) com frequência igual à dobro da frequência (metade do período) do gerador do Exemplo0901.v.  
O nome do arquivo deverá ser Exemplo\_0904.v.  
Incluir previsão de testes e verificação da carta de tempo usando GTKWave.
- 05.) Projetar e descrever em Verilog um módulo gerador de pulso (**pulse**) com frequência igual a um quarto da frequência (quatro vezes o período) do gerador do Exemplo0901.v.  
O nome do arquivo deverá ser Exemplo\_0905.v.  
Incluir previsão de testes e verificação da carta de tempo usando GTKWave.

## Extra

- 06.) Projetar e descrever em Verilog um módulo gerador de pulso (**pulse**) com marcação igual a 2 unidades de tempo, sincronizado com a borda de subida do gerador do Exemplo\_0901.v. O nome do arquivo deverá ser Exemplo\_0906.v. Incluir previsão de testes e verificação da carta de tempo usando GTKWave.  
DICA: Usar *always @(posedge clk)*.
- 07.) Projetar e descrever em Verilog um módulo gerador de pulso (**pulse**) com marcação igual a 2 unidades de tempo, sincronizado com a borda de descida do gerador do Exemplo\_0901.v. O nome do arquivo deverá ser Exemplo\_0907.v. Incluir previsão de testes e verificação da carta de tempo usando GTKWave.  
DICA: Usar *always @(negedge clk)*.
- 08.) Projetar e descrever em Verilog um módulo gerador de pulso (**pulse**) com marcação igual a 2 unidades de tempo, sincronizado com as bordas de subida e descida do gerador do Exemplo\_0901.v. O nome do arquivo deverá ser Exemplo\_0908.v. Incluir previsão de testes e verificação da carta de tempo usando GTKWave.  
DICA: Usar *always @(clk)*.

Instruções para ver as cartas de tempo no GTKWave:

- 01.) Abrir o módulo de visualização (GTKWave)
- 02.) Selecionar a pasta de trabalho:  
File  
Open  
Exemplo\_0901 (.vcd) (por exemplo)
- 03.) Selecionar os sinais desejados:  
clk (sinal a ser visto)  
clock (outro sinal a ser visto)  
(selecionar, arrastar e soltar na coluna à direita)