

PUC-Minas - Ciência da Computação  
ARQ1 – Guia 06  
Período: 19-23/03/2012

Tema: Introdução à linguagem Verilog  
Atividade: Circuitos sequenciais

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

O nome do arquivo deverá ser Exemplo0041.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 ( clk );  
    output clk;  
    reg    clk;
```

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

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

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

```
module Exemplo0041;
```

```
    wire clk;  
    clock CLK1 ( clk );
```

```
    initial begin  
        $dumpfile ( "Exemplo041.vcd" );  
        $dumpvars;
```

```
        #120 $finish;  
    end
```

```
endmodule // Exemplo041 ( )
```

- 02.) Projetar e descrever em Verilog módulos geradores de pulso (**pulse**) e gatilho (**trigger**). O nome do arquivo deverá ser Exemplo0042.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 ( clk );  
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 Exemplo0042;

    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 ( "Exemplo0042.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 // Exemplo0042

```

- 03.) Projetar e descrever em Verilog módulos geradores de pulso (**pulse**) com períodos diferentes. O nome do arquivo deverá ser Exemplo0043.v, e poderá seguir o modelo descrito a seguir. O gerador de **clock** do Exemplo0041.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 Exemplo0043;

    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 ( " Exemplo0043.vcd" );
        $dumpvars ( 1, clock, p1, p2, p3, p4 );

        #480 $finish;
    end

endmodule // Exemplo0043

```

- 04.) Projetar e descrever em Verilog um módulo gerador de pulso (**pulse**) com frequência igual a um terço da frequência (triplo do período) do gerador do Exemplo0041.v.  
O nome do arquivo deverá ser Exemplo0044.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 ao dobro da frequência (metade do período) do gerador do Exemplo0041.v.  
O nome do arquivo deverá ser Exemplo0045.v.  
Incluir previsão de testes e verificação da carta de tempo usando GTKWave.
- 06.) Projetar e descrever em Verilog um módulo gerador de pulso (**pulse**) com frequência igual a 5 unidades de tempo, sincronizado com a borda de subida do gerador do Exemplo0041.v.  
O nome do arquivo deverá ser Exemplo0046.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 frequência igual a 5 unidades de tempo, sincronizado com a borda de descida do gerador do Exemplo0041.v. O nome do arquivo deverá ser Exemplo0047.v. Incluir previsão de testes e verificação da carta de tempo usando GTKWave.
- 07.) Projetar e descrever em Verilog um módulo gerador de pulso (**pulse**) com frequência igual a 5 unidades de tempo, sincronizado com as bordas de subida e descida do gerador do Exemplo0041.v. O nome do arquivo deverá ser Exemplo0048.v. Incluir previsão de testes e verificação da carta de tempo usando GTKWave.

Instruções para ver as cartas de tempo no GTKWave:

- 01.) Abrir o módulo de visualização (GTKWave)
- 02.) Selecionar a pasta de trabalho:  
File  
Open  
Exemplo0041 (.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)