PUC-Minas - Ciência da Computação ARQ1 – Guia 09

Período: 03-070/10/2011

Tema: Introdução à linguagem Verilog Atividade: Circuitos sequenciais – Flip-Flops

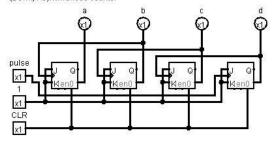
Todos os circuitos deverão ser simulados no Logisim.

- 01.) Projetar e descrever em Verilog um módulo, com portas e flip-flops tipo T apenas, para implementar um contador assíncrono decrescente. DICA: Ver modelo anexo.
- 02.) Projetar e descrever em Verilog um módulo com portas e flip-flops tipo T apenas, para implementar um contador assíncrono crescente
- 03.) Projetar e descrever em Verilog um módulo, com portas e flip-flops tipo T apenas, para implementar um contador decádico decrescente. DICA: Ver modelo anexo.
- 04.) Projetar e descrever em Verilog um módulo com portas e flip-flops tipo T apenas, para implementar um contador decádico crescente
- 05.) Projetar e descrever em Verilog um módulo, com portas e flip-flops tipo JK apenas, para implementar um contador módulo 6. DICA: Ver modelo anexo.

## Extras

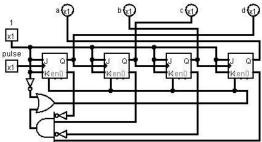
- 06.) Projetar e descrever em Verilog um módulo, com portas e flip-flops tipo JK apenas, para implementar um contador em anel. DICA: Ver modelo anexo.
- 07.) Projetar e descrever em Verilog um módulo com portas e flip-flops tipo JK apenas, para implementar um contador em anel torcido. DICA: Ver modelo anexo.

## (Down) Asynchronous counter

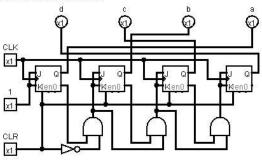


OPS: CLR - 1 - pulse

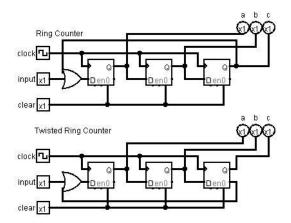
## (Down) Decade counter



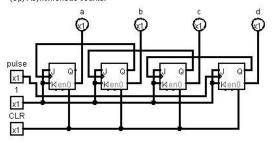
(Down) Synchronous counter



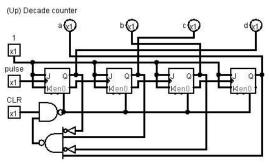
OPS: CLR - 1 - CLK



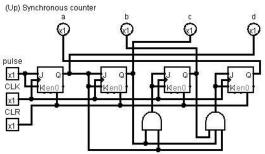
## (Up) Asynchronous counter



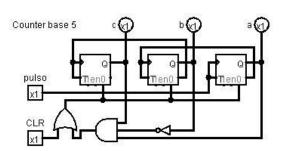
OPS: CLR - 1 - pulse



OPS: CLR - 1 - pulse



OPS: CLR - pulse - CLK



```
module dff ( output q, output qnot,
                                                         module srff (output q, output qnot,
            input d, input clk);
                                                                      input s, input r, input clk);
reg q, qnot;
                                                         reg q, qnot;
always @( posedge clk )
                                                         always @( posedge clk )
begin
                                                         begin
  q \le d;
                qnot <= ~q;
                                                           if (s & ~r)
end
                                                            begin
                                                            q <= 1;
                                                                         qnot \le 0;
endmodule // dff
                                                            end
module tff ( output q, output qnot,
                                                           else
            input t, input clk,
                                                             if (~s & r)
            input preset, input clear );
                                                             begin
                                                              q <= 0;
                                                                         qnot \ll 1;
reg q, qnot;
                                                             end
                                                             else
always @( posedge clk )
                                                              if (s&r)
begin
                                                               begin
if (~clear)
                                                                         qnot <= 0; // arbitrary
                                                               q <= 0;
begin
                                                               end
 q <= 0;
                qnot <= -q;
                                                         end
 end
else
                                                         endmodule // srff
 if (~preset)
 begin
                                                         module jkff (output q, output qnot,
  q <= 1;
                qnot <= -q;
                                                                      input j, input k, input clk);
 end
                                                         reg q, qnot;
 else
 begin
                                                         always @( posedge clk )
   if (t)
                                                         begin
   begin
                                                           if ( j & ~k )
    q <= ~q;
                qnot <= \sim q;
                                                            begin
   end
                                                             q <= 1;
                                                                          qnot \le 0;
 end
                                                            end
end
                                                            else
                                                             if (~j & k)
endmodule // tff
                                                             begin
                                                              q <= 0;
                                                                          qnot <= 1;
                                                             end
                                                             else
                                                              if (j&k)
                                                               begin
                                                               q \le q; q \le q 
                                                               end
                                                         end
                                                         endmodule // jkff
```