

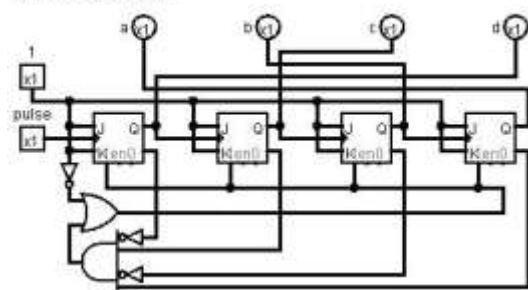
Tema: Introdução à linguagem Verilog
Atividade: Circuitos sequenciais – Flip-Flops
Todos os circuitos deverão ser simulados no Logisim.

- 01.) Projetar e descrever em Verilog um módulo, com portas e flip-flops tipo JK apenas, para implementar um contador assíncrono decrescente com 5 bits de comprimento.
DICA: Ver modelo anexo.
- 02.) Projetar e descrever em Verilog um módulo com portas e flip-flops tipo JK apenas, para implementar um contador assíncrono crescente com 5 bits de comprimento.
- 03.) Projetar e descrever em Verilog um módulo, com portas lógicas e flip-flops tipo JK apenas, para implementar um contador decádico crescente com 6 bits de comprimento.
DICA: Ver modelo anexo.
- 04.) Projetar e descrever em Verilog um módulo com portas e flip-flops tipo JK apenas, para implementar um contador decádico crescente com 6 bits de comprimento.
- 05.) Projetar e descrever em Verilog um módulo, com portas e flip-flops tipo T apenas, para implementar um contador módulo 10.
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 com 6 bits de comprimento.
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 com 6 bits de comprimento.
DICA: Ver modelo anexo.

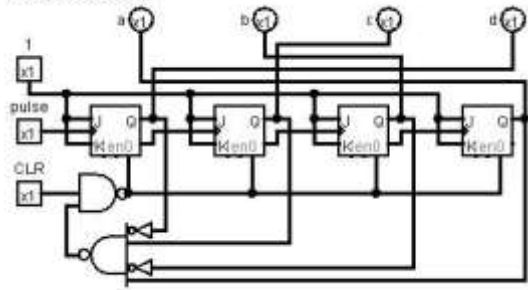
Down) Decade counter



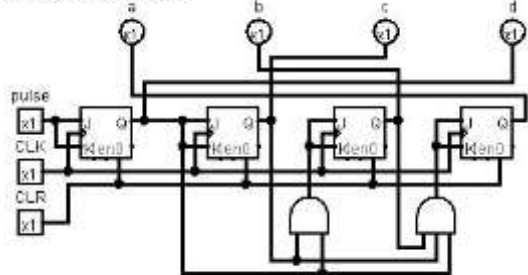
The image contains two circuit diagrams. The top diagram is labeled "Ring Counter" and the bottom diagram is labeled "Twisted Ring Counter". Both diagrams show a clock input, an input x1, and a clear input x1. The Ring Counter uses three D flip-flops (labeled Den0) connected in a ring configuration. The Twisted Ring Counter uses three D flip-flops (labeled Den0) connected in a ring configuration, but with a twist in the feedback path. Both diagrams have three outputs labeled a, b, and c, which are connected to three 7-segment displays labeled x1, x1, and x1.

[illegible]

(Up) Decade counter



(Ug) Synchronous counter



```

module dff ( output q, output qnot,
             input  d, input  clk );
reg q, qnot;

always @( posedge clk )
begin
    q <= d;      qnot <= ~d;
end

endmodule // dff

module tff ( output q, output qnot,
             input  t, input  clk,
             input  preset, input clear );

reg q, qnot;

always @( posedge clk )
begin
    if ( ~clear )
    begin
        q <= 0;      qnot <= 1;
    end
    else
    if ( ~preset )
    begin
        q <= 1;      qnot <= 0;
    end
    else
    begin
        if ( t )
        begin
            q <= ~q;  qnot <= ~qnot;
        end
    end
end

endmodule // tff

```

```

module srff ( output q, output qnot,
              input  s, input r, input clk );
reg q, qnot;

always @( posedge clk )
begin
    if ( s & ~r )
    begin
        q <= 1;      qnot <= 0;
    end
    else
    if ( ~s & r )
    begin
        q <= 0;      qnot <= 1;
    end
    else
    if ( s & r )
    begin
        q <= 0;      qnot <= 0; // arbitrary
    end
end

endmodule // srff

module jkff ( output q, output qnot,
              input  j, input  k, input clk );
reg q, qnot;

always @( posedge clk )
begin
    if ( j & ~k )
    begin
        q <= 1;      qnot <= 0;
    end
    else
    if ( ~j & k )
    begin
        q <= 0;      qnot <= 1;
    end
    else
    if ( j & k )
    begin
        q <= ~q;      qnot <= ~qnot;
    end
end

endmodule // jkff

```