

ARQ1 _ Aula_13

Tema: Introdução à linguagem Verilog e simulação em Logisim

Orientação geral:

Apresentar todas as soluções em apenas um arquivo com formato texto (.txt).
As implementações e testes dos exemplos em Verilog (.v) fornecidos como pontos de partida, também fazem parte da atividade e deverão ser entregues os códigos fontes separadamente.
As saídas de resultados, opcionalmente, poderão ser copiadas ao final do código, como comentários.

Os *layouts* de circuitos deverão ser entregues no formato (.circ), identificados internamente. Figuras exportadas pela ferramenta serão aceitas como arquivos para visualização, mas não terão validade para fins de avaliação. Separar as versões completas (a) das simplificadas (b).

Arquivos em formato (.pdf), fotos, cópias de tela ou soluções manuscritas serão aceitas como recursos suplementares para visualização e não terão validade para fins de avaliação.

Atividade: Circuitos sequenciais – Flip-Flops

Todos os circuitos deverão ser simulados no Logisim.

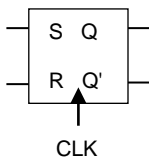
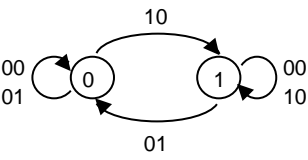
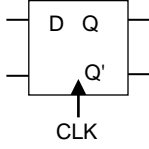
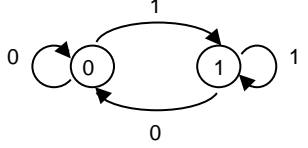
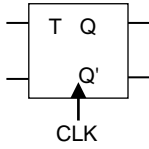
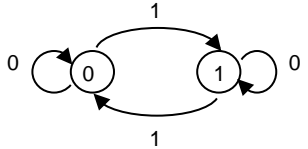
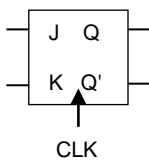
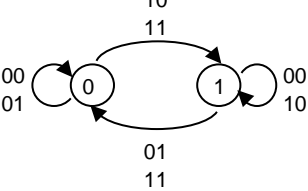
- 01.) Projetar e descrever em Logisim e Verilog um módulo, com portas e flip-flops tipo JK apenas, para implementar um contador assíncrono decrescente com 6 bits de comprimento.
DICA: Ver modelo anexo.
- 02.) Projetar e descrever em Logisim e Verilog um módulo com portas e flip-flops tipo JK apenas,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

- 05.) Projetar e descrever em Logisim e Verilog um módulo, com portas e flip-flops tipo T apenas, para implementar um contador síncrono módulo 9.
DICA: Ver modelo anexo.

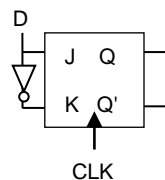
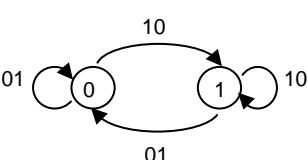
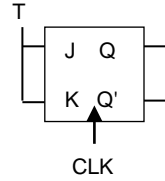
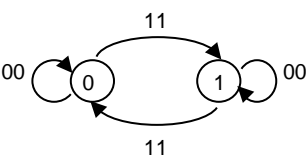
Extras

- 06.) Projetar e descrever em Logisim e 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 Logisim e 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.

Flip-flops

Flip-flop	Estados	Característica	Transição	Equação																																								
		<table><tr><th>S</th><th>R</th><th>Q_{t+1}</th><th>Q'_{t+1}</th></tr><tr><td>0</td><td>0</td><td>Q_t</td><td>Q'_t</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>?</td><td>?</td></tr></table>	S	R	Q_{t+1}	Q'_{t+1}	0	0	Q_t	Q'_t	0	1	0	1	1	0	1	0	1	1	?	?	<table><tr><th>Q_t</th><th>Q_{t+1}</th><th>S</th><th>R</th></tr><tr><td>0</td><td>0</td><td>0</td><td>X</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>X</td><td>0</td></tr></table>	Q_t	Q_{t+1}	S	R	0	0	0	X	0	1	1	0	1	0	0	1	1	1	X	0	$Q_{t+1}=S+R'.Q_t$
S	R	Q_{t+1}	Q'_{t+1}																																									
0	0	Q_t	Q'_t																																									
0	1	0	1																																									
1	0	1	0																																									
1	1	?	?																																									
Q_t	Q_{t+1}	S	R																																									
0	0	0	X																																									
0	1	1	0																																									
1	0	0	1																																									
1	1	X	0																																									
		<table><tr><th>D</th><th>Q_{t+1}</th><th>Q'_{t+1}</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	D	Q_{t+1}	Q'_{t+1}	0	0	1	1	1	0	<table><tr><th>Q_t</th><th>Q_{t+1}</th><th>D</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	Q_t	Q_{t+1}	D	0	0	0	0	1	1	1	0	0	1	1	1	$Q_{t+1} = D$																
D	Q_{t+1}	Q'_{t+1}																																										
0	0	1																																										
1	1	0																																										
Q_t	Q_{t+1}	D																																										
0	0	0																																										
0	1	1																																										
1	0	0																																										
1	1	1																																										
		<table><tr><th>T</th><th>Q_{t+1}</th><th>Q'_{t+1}</th></tr><tr><td>0</td><td>Q_t</td><td>Q'_t</td></tr><tr><td>1</td><td>Q'_t</td><td>Q_t</td></tr></table>	T	Q_{t+1}	Q'_{t+1}	0	Q_t	Q'_t	1	Q'_t	Q_t	<table><tr><th>Q_t</th><th>Q_{t+1}</th><th>T</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	Q_t	Q_{t+1}	T	0	0	0	0	1	1	1	0	1	1	1	0	$Q_{t+1} = T \oplus Q_t$																
T	Q_{t+1}	Q'_{t+1}																																										
0	Q_t	Q'_t																																										
1	Q'_t	Q_t																																										
Q_t	Q_{t+1}	T																																										
0	0	0																																										
0	1	1																																										
1	0	1																																										
1	1	0																																										
		<table><tr><th>J</th><th>K</th><th>Q_{t+1}</th><th>Q'_{t+1}</th></tr><tr><td>0</td><td>0</td><td>Q_t</td><td>Q'_t</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>Q'_t</td><td>Q_t</td></tr></table>	J	K	Q_{t+1}	Q'_{t+1}	0	0	Q_t	Q'_t	0	1	0	1	1	0	1	0	1	1	Q'_t	Q_t	<table><tr><th>Q_t</th><th>Q_{t+1}</th><th>J</th><th>K</th></tr><tr><td>0</td><td>0</td><td>0</td><td>X</td></tr><tr><td>0</td><td>1</td><td>1</td><td>X</td></tr><tr><td>1</td><td>0</td><td>X</td><td>1</td></tr><tr><td>1</td><td>1</td><td>X</td><td>0</td></tr></table>	Q_t	Q_{t+1}	J	K	0	0	0	X	0	1	1	X	1	0	X	1	1	1	X	0	$Q_{t+1}=J.Q'_t+K'.Q_t$
J	K	Q_{t+1}	Q'_{t+1}																																									
0	0	Q_t	Q'_t																																									
0	1	0	1																																									
1	0	1	0																																									
1	1	Q'_t	Q_t																																									
Q_t	Q_{t+1}	J	K																																									
0	0	0	X																																									
0	1	1	X																																									
1	0	X	1																																									
1	1	X	0																																									

Configurações especiais

Flip-flop	Estados	Característica	Transição	Equação																																								
		<table><tr><th>J</th><th>K</th><th>Q_{t+1}</th><th>Q'_{t+1}</th></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td></td><td></td><td></td><td></td></tr></table>	J	K	Q _{t+1}	Q' _{t+1}					0	1	0	1	1	0	1	0					<table><tr><th>Q_t</th><th>Q_{t+1}</th><th>J/D</th><th>K/D'</th></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td></tr></table>	Q _t	Q _{t+1}	J/D	K/D'	0	0	0	1	0	1	1	0	1	0	0	1	1	1	1	0	$Q_{t+1}=1.Q_t'+0'.Q_t$ $Q_{t+1}=1$ $Q_{t+1}=0.Q_t'+1'.Q_t$ $Q_{t+1}=0$
J	K	Q _{t+1}	Q' _{t+1}																																									
0	1	0	1																																									
1	0	1	0																																									
Q _t	Q _{t+1}	J/D	K/D'																																									
0	0	0	1																																									
0	1	1	0																																									
1	0	0	1																																									
1	1	1	0																																									
		<table><tr><th>J</th><th>K</th><th>Q_{t+1}</th><th>Q'_{t+1}</th></tr><tr><td>0</td><td>0</td><td>Q_t</td><td>Q'_t</td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td>1</td><td>Q_t'</td><td>Q_t</td></tr></table>	J	K	Q _{t+1}	Q' _{t+1}	0	0	Q _t	Q' _t									1	1	Q _t '	Q _t	<table><tr><th>Q_t</th><th>Q_{t+1}</th><th>J=T</th><th>K=T</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td></tr></table>	Q _t	Q _{t+1}	J=T	K=T	0	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	$Q_{t+1}=0.Q_t'+0'.Q_t$ $Q_{t+1}=0'.Q_t = Q_t$ $Q_{t+1}=1.Q_t'+1'.Q_t$ $Q_{t+1}=1.Q_t' = Q_t'$
J	K	Q _{t+1}	Q' _{t+1}																																									
0	0	Q _t	Q' _t																																									
1	1	Q _t '	Q _t																																									
Q _t	Q _{t+1}	J=T	K=T																																									
0	0	0	0																																									
0	1	1	1																																									
1	0	1	1																																									
1	1	0	0																																									

```

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 jkff ( output q, output qnot,
             input  j, input k,
             input clk, input preset, input clear );

reg  q, qnot;

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

```

```

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

reg q, qnot;

always @( posedge clk or ~preset or ~clear)
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

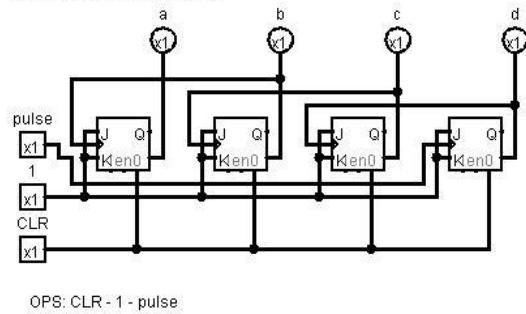
module srff ( output q, output qnot,
             input  s, input  r, input clk );
reg q, qnot;

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

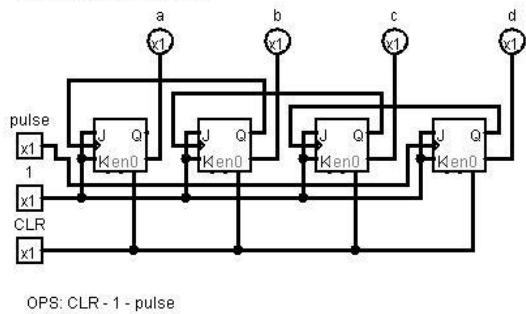
endmodule // srff

```

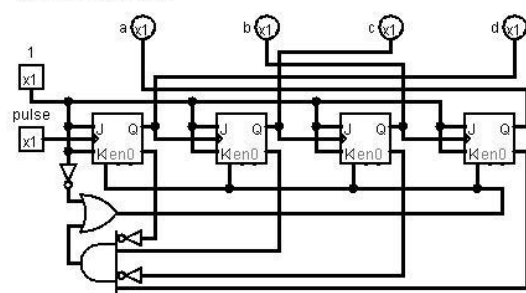
(Down) Asynchronous counter



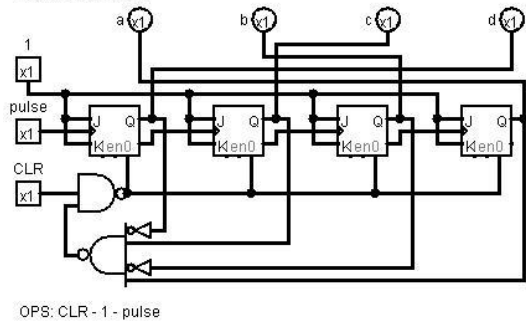
(Up) Asynchronous counter



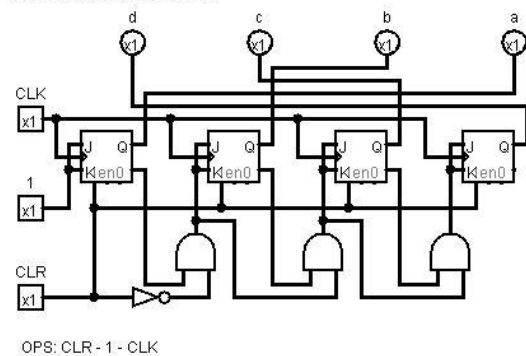
(Down) Decade counter



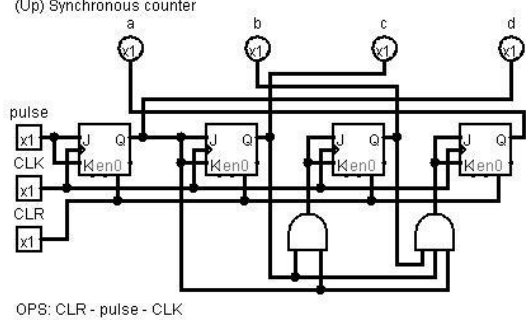
(Up) Decade counter



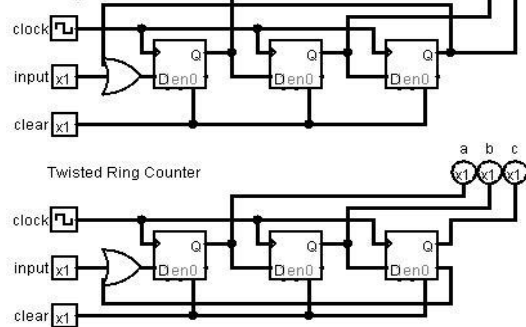
(Down) Synchronous counter



(Up) Synchronous counter



Ring Counter



Counter base 5

