0 0 1 1	Qn 0 1 0 1	Q	(n+1) 0 0 1	No Re Se	ommer o chanç eset t	ge															
	0		$\overline{)} + \overline{K}$	•	Com	<b>nments</b> change															
0 0 0 1	0 1 1 0 0	1 0 1 0		1 0 0 1	Rese Rese Set	et															
, +1 <sup>=</sup>	1 1 - <b>type</b> = T · 0	Q(n)	$\overline{0}$ + $\overline{T}$	Q(r)		erse															
<b>T</b> 0 0 1 1	Qn 0 1 0 1		0 1 1 0	No No Inv	chang chang chang verse	ge															
	p_d_l begin	<b>of</b>	: pr	ocess	(en,d,	f the p	roce	ess p	p_d_	_lat	ch										
	е	q q lsif q q	<= ' _bar (en <= d _bar f;	<= '1'; = '1'; <= not	'; ) then t d;																
	 En  en_gen begin	able	gene	ration	eset a	ess	mulus	s pro	oces	sses	from	the	testk	ench	tb_	_d_la <sup>·</sup>	tch.v	/hd			
	s w s w end p	_en ait _en ait _en ait _en ait roce	<= '1 for 4 <= '0 for 7 <= '1 for 5 ss p	5ns; '; .5ns; '; ns; en_ge	n;																
tb_a	arst: begin s w s	pro _ars ait _ars ait	t <= for 2 t <= for 4	'0'; 0 ns; '1'; ns; _arst	;																
	stimul begin s w s w	us : _d < ait _d < ait _d <		ess ; @ns; ; ns;																	
	end p  check  begin  wait  wait  asser	roce : pr unti for t (s	1 ((s 0.4ns _q =	<pre>Ons; _stimu _d = ; '0');</pre>	'0') a	ind (s_e =q_bar DW)" sev	= '1')	)		s_arst	: = '1	'));									
	wait wait asser repor wait wait asser	unti for t (s t "R unti for t (s	1 ((s 0.4ns _q = eset 1 (ri 0.4ns _q =	_d = ;; '0'); fault sing_(''0');	'1') and (s. (D HIC) edge(s. and (s.	and (s_e =_q_bar ==GH)" se =_d) and =_q_bar	en = '0 = '1') everity (s_en = '1')	0') an ) y erro n = '(	or;	and (s	s_arst	: = '0	') and	(s_q	= '0')	);					
	wait wait asser repor wait wait	unti for t (s t "H unti for	l (fa 0.4ns _q = old f l ((s 0.4ns	<pre>lling; ; '1') a ault d _d = ;</pre>	_edge( and (s (Hold ) '1') a	when EN s_d) an s_q_bar when EN and (s_e	d (s_e = '0')   is LO   n = '1	en = ) DW)" : 1') a	'0') sever	and ( rity e s_arst	s_ars	t = '(	0') an	<b>d</b> (s_c	ן = '1'	));					
ame	end p	roce	ot W	_checl	k;	ated ti				ms		150.000	ns	200,0	100 ns	250.0	100 ns	300.	000 ns	350	.000 ns
Fl	arst q q_bar			DL cc	ode o	f the p	proce	esses	S												
	_ff_ar begin if q	st: (ars <= _bar	proce t = ' '0'; <= '	ss(cll 1') tl 1';	k,arst		p_d_	_ffa	arst												
	q q end p ent p	<= _bar if; roce	d; <= n ss p_ /HDL	ot d; d_ff_a	arst; of the	process	s p_d_	_ff_ı	rst												
o_d_	_ff_rs begin i	f ri i	sing_ f (rs q q_ lse q	edge(( t = ': <= '0 bar <: <= d; bar <:	clk) t	en															
	end p  Listing  k_ff_r  varia  begin	of V	f; ss p_ /HDL proce q_loc	d_ff_i  code (	of the k) std_log		; p_jk	k_ff <sub>-</sub>	_rst												
		f ri i e e	f (rs q_ lsif q_ lsif q_ lsif	t = ': local ((j = local (j = local (k = local		en '; ind(k = ot q_loc then '; then		then													
	end p	q q nd i roce	nd if <= q _bar f; sss p_	; _loca: <= not jk_ff_ code	l; t q_lo _rst; of the		s p_t_	_ff_!	rst												
١	begin	le q f ri if q els q end q <	_loca sing_ (rst _loca if (t _loca if; = q_l	<pre>l : st  edge(( = '1') l :=</pre>	td_log  clk) t  ) then  '0';  ') the  not q_	chen en local;															
	end p	nd i	f; ss p_	t_ff_i	ch pr	ocesso		lock	<b>k</b> , ।	rese	t,:	stim	ulus	1							
	p_clk begin s w	ock  gen _clk ait _clk	gener	ation cess 0'; 0ns; 1';	proce																
	end p Re Re p_res begin s w s	roce set et: _ars	ss p_  gener	clk_go  ation  ss '0'; 0 ns; '1';	en;  proce																
	end p St St p_sti begin s w s	roce imul mulud < aitd <	us  s: pr = '0' for 2 = '1'	reset cocess ; 5ns;																	
	S W S W S	_d < ait _d < ait _d < ait _d <	for 3 = '0' for 1 = '1' for 5 = '0' for 7 = '1' for 1 ess p_	; 5ns; ; 5ns; ; 5ns; ; 0ns;	lus;																
	w a r	egin ait ait sser epor	until for 0 t (s_ t "Re	((s_0 .4ns; q = '0 set fa ((s_0	0') and ault (	o') and  od (s_q_ D LOW)"	bar = sever	'1') rity (	error	r;											
	r w a r	epor ait ait sser epor	t "Re until for 0 t (s_ t "Ho until	(ris: .4ns; q = '( ld fac	ault (  ing_ed; 0') and ult (H	d (s_q_D HIGH)  lge(s_d)  d (s_q_ lold wheeledge(s_d)	" seve and n bar = n CLK	erity not(r: '1') is no	erro	g_edge	" sev	erity	error	;							
	a r w a a	sser epor ait ait sser	t "Ho until for 0 t (s_ t "Da	q = ': ld fau  ((s_c .4ns; q = ': ta fau	ult (Ho d = '1 1') and ult" so	d (s_q_ lold whe ') and d (s_q_ eeverity " sever	rising bar = error	is no g_edgo '0')	not ri ge(s_c	clk) a	and (s	_arst	= '0'								
2.2	Listing	of t	estbe	ation		<b>d_ff_r</b> :	st														
	w s w end p	_clk ait _clk ait roce  set	gener	Ons; 1'; Ons; clk_go  ation	en;  proce	 SS															
	W	_rst ait _rst ait roce  imul	us	0 ns; 1'; ns; reset																	
	W S W S W	_d < ait	= '0' for 2 = '1' for 3 = '0' for 1 = '1' for 5 = '0'	5ns; ; 5ns; ; 5ns; ; 5ns;																	
	end p  p_che b w	_d < rait roce ck : egin rait rait	proc until for 0	; 0ns; stimu: ess ((s_0, .4ns;	d = '0	'') and			. –	clk) <mark>a</mark>	ınd (s	_rst :	= '1')	);							
	w a r w w	ait sser epor	until for 0 t (s_ t "Re until for 0	((s_q, .4ns; q = 'q, set fa	<pre>d = '1 0') and ault () ing_ed;</pre>	D LOW)"  ') and  d (s_q_ D HIGH)  ge(s_d)	rising bar = " seve and n	g_edge '1') erity not(r:	ge(s_c	clk) a					'0') a	and (s_d	o' = F	'));			
	r w a r w	epor ait ait sser epor	t "Ho until for 0 t (s_ t "Ho until for 0	(fal4ns; q = ': ld fau  ((s_(.4ns;	ult (He ling_e  1') and ult (He  d = '1	old whe	bar = n CLK  bar = n CLK  rising	not() '0') is no	not ri	ng_edg ising) clk) a	ge(s_c " seventh seven	erity	error = '0')	_rst =	- '0')	and (s	_q = ':	L'));			
2.3 1	r r	epor epor nd p	t "Da t "Te roces	ta fau st con s p_cl	ult" somplete	d (s_q_ everity " sever	error	r;	tr	ranspa	rency	checl									
	p_clk begin s w	ockgenclk aitclk	gener : pro : <= ' for 1 : <= ' for 1	ation cess 0'; 0ns; 1'; 0ns;	proce																
	p_res begin s w s	et: _rst ait _rst	proce <pre></pre>	ss 0'; 0 ns; 1';	proce																
	S	imul mulu _j < _k <	us	ocess ; ; ; 5ns;																	
	S W S S S S S S	_k < ait _j < _k < ait _j < ait _j <	= '1' for 3 = '1' = '0' for 1 = '1' = '1' = '1'	; 5ns; ; 5ns; ; 5ns; ;																	
	end p  p_che	ait _j < _k < ait roce ck :	proc	5ns; ; ; 0ns; stimu:		ge(s_cl	k) and	d s_r	rst =	'1' a	and s_	q = '(°	ð');								
	w a r w a r	ait sser epor ait ait sser epor	for 0 t (s_ t "Re until for 0 t (s_ t "Re	.4ns; q = '('set fa') (ris: .4ns; q = '('set fa') (ris:	0') and ault (ding_ed) and ault (ding_ed) ing_ed ing_ed	d (s_q_ Q LOW)" ge(s_cl d (s_q_ Q HIGH)	bar = sever k) and bar = "seve	'1') rity ( d s_r: '1') erity	error	r; '1' a or;	and s_	.q = '.	L');	s_k	: '0'`						
	w a r w a a	ait sser epor ait ait sser	for 0 t (s_ t "J  until for 0 t (s_ t "K	.4ns; q = ': fault' (ris: .4ns; q = '(fault'	1') and seve ing_ed	dd (s_q_ erity er dge(s_cl dd (s_q_ erity er	bar = ror; k) and bar = ror;	'0') d s_r: '1')	rst =	'0' a	and s_	j = '(	ð' and	s_k =	· '1');		= 1	);			
	w a r w a a	ait sser epor ait ait sser	for 0 t (s_ t "Ho until for 0 t (s_ t "Ho	.4ns; q = '1 ld fau (ris: .4ns; q = '( ld fau	1') and ult (Q ing_ed; 0') and ult (Q	d (s_q_ ) HIGH)" dge(s_cl d (s_q_ ) LOW)"	bar = sever k) and bar = severi	'0') rity ( d s_r: '1') ity e	error est =	r; '0' a	and s_	j = '(	ð' and	s_k =	= '0'	and s_q	= '0'	);			
	w a r w a a	ait sser epor ait ait sser	for 0 t (s_ t "Ho until for 0 t (s_ t "Ho	.4ns; q = '( ld fau (ris: .4ns; q = ': ld fau	0') and ult (Q ing_ed; 1') and ult (Q	ge(s_cl d (s_q_ ) HIGH)" ge(s_cl d (s_q_ ) LOW)"	bar = sever k) and bar = severi	'1') rity ( d s_r: '0') ity e	error est =	r; '0' a	and s_	.j = ':	L' and	s_k =	= '1'	and s_q	= '0'	);			
	w a r w a a	ait sser epor ait ait sser	for 0 t (s_ t "cl  until for 0 t (s_ t "cl	.4ns; q = ': ock/sy (not .4ns; q = '( ock/sy	1') and ync far (rising 0') and ync far	ng_edge(  nd (s_q_  nult (as  ng_edge(  nd (s_q_  nult (as  nult (as	bar = ync ch s_clk) bar = ync ch	'0') nange )) and '1') nange	when	n Q HI	GH)" : _edge(:	sever:	ity er	ror; alling							
2.4	Listing	nd p	roces	s p_cl	or <b>p_1</b>	" sever		ote;													
	p_clk begin s w s	_clk ait _clk ait roce	: pro  : <= ' for 5 for 5 ss p_	Cess 0'; ns; 1'; ns; clk_ge																	
	p_res begin s w	et: _rst ait _rst ait	proce  <= ' for 2 <= ' for 5 ss p_	ss 0'; 4 ns; 1'; ns;	;	!SS															
	St p_sti begin s w s	imulu mulu _t < ait _t <	us s: pr = '0' for 2 = '1' for 3 = '0'	ocess; ; 5ns; ; 5ns;																	
	W S W S	ait _t < _t < ait _t < ait roce	for 1 = '1' for 1 = '0' for 7 = '1' for 9 ss p_	5ns;; 5ns;; 5ns;; ns;	lus;																
	b w a r w w	egin ait sser epor ait ait sser	until for 0 t (s_ t "Re until for 0 t (s_	(ris: .4ns; q = '( set fa (ris: .4ns; q = '(	0') and ault" ing_ed	<pre>def(s_cl d (s_q_ severit def(s_cl d (s_q_ ) High)"</pre>	bar = y erro k) and bar =	'1') or; d s_r: '1')	rst =	'0' a		t = '0	∂' and	s_q =	= '0');						
	r w a r w w	epor ait sser epor ait ait sser	t "Ho until for 0 t (s_ t "Ho until for 0 t (s_	(ris:   .4ns;   q = ':   ld fan   (ris:   .4ns;   q = ':	ult (Q ing_ed; 1') and ult (Q ing_ed; 1') and	<pre>PHigh)"  ge(s_cl  d (s_q_  Low)"  ge(s_cl  d (s_q_  )</pre>	sever k) and bar = severi k) and bar =	rity ( d s_r: '0') ity el d s_r: '0')	error est =  error; erst =	'0' a ; '0' a											
	r w a r w	epor ait ait sser epor	t "In until for 0 t (s_ t "In until for 0	(ris: .4ns; q = '( not' .4ns;	fault ing_ed; 0') and fault (risin;	(Q Low)  Ige(s_cl  Id (s_q_ (Q High  Ig_edge(	" seve k) and bar = )" sev s_clk)	erity d s_r: '1') verity )) and	<pre>/ erro  rst =  cy err  nd (ri</pre>	'0' a							) and	(rising	_edge	(s_t) <mark>o</mark>	r fall
	a r w w a r r	epor ait ait sser epor	t (s_ t "cl until for 0 t (s_ t "cl t "Te	<pre>q = '( ock/sy</pre>	0') and ync fan (rising 1') and ync fan mplete	nd (s_q_nult (as ng_edge( nd (s_q_nult (as " sever	ync ch s_clk) bar = ync ch	nange )) and '0') nange	e when	ising_	_edge(	s_rst	or f	alling	_edge(	s_rst)	) and	rising	_edge	(s_t)	r fal
3.1 ame	p_d_f	nsho	ots v	vith s		ated t	ime v		efor		499.999	0 ns	500.0	00 ns	1600.00	0 ns	799.999	15	300.000	5 90	10,000 ns
s_ar s_d s_q s_q s_q	p_d_1  Value  k 0  tt 0		rst	100.	000 ns	289, 900 n	300	),000 ns	40	00.000 ns	500.	.000 ns	600.00	0 ns	790.090 ns	800.0	00 ns	990.000 ns	1.0	00.999 ns	1,100.0
s_d s_q s_q s_clk s_rst	1 1 0 p_d_f				200.000	ns ns	1400.	.000 ns		600	.000 ns		300.00	) ns		000.000 ns		11,200.000		1,400.0	00 ns
as_k			rst				12.	.000 ns	200	0.000 ns				0 ns	350.000 ns	1490.0					550,800
5 s_q_ 5 s_q_l 6 s_clk 6 s_rst 6 s_t 6 s_q_	t 0 1 0		00 ns	50.00	00 ns	100.000 ns					250.	000 ns	300.00				00 ns	450.000 ns	500	.000 ns	

Assignment 7

1. Preparation tasks

Link to this Assignment Link to top of repository

1.1 D-type flip-flop