DigitLabwork#4

Shift-register and Counter in Verilog codes

[1] Training goals

[2] Preparations

[3] Lab-task instruction

[4] Observations/Evaluation

= =

[1] Training goals

Through the lab-work, class members are expected to get acquainted with the following matters.

a)) operation of left-/right-shift registers and counters;

b)) the use of Verilog for logic circuit description and simulation, including

** circuit module building up in Verilog,

** setting up of the test data;

c)) the interactive commands required in operating the Verilog-code development system:

** compilation for a syntax error-free Verilog description and test data set;

** simulation of the Verilog-coded circuit module;

[** synthesis of the Verilog-coded circuit module].

[2] Preparation

Every class member should get oneself prepared by studying the following materials prior to attending the lab-work sessions.

a)) Refer to “Digital Circuits 123”-section[1] and, better yet, datasheets from IC manufacturers such that functional/electrical/timing specifications regarding the circuit components targeted in the lab-work are understood;

b)) Refer to “Digital Circuits 123”-section [4], where examples of Verilog-coding of the circuit components targeted in the lab-work are offered;

c)) Refer to “Digital Circuits 123”-section [4] so as to know about how test data for circuit simulation should be arranged;

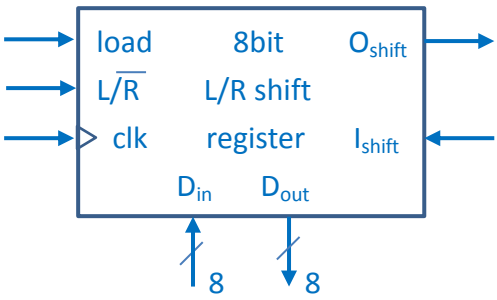
d)) Refer to “Development Context of Verilog Code” for operational details in running the developing system.

[3] Lab-task instruction

In digitlabwork#4, 2 tasks are assigned to every class members as given below.

[TASK1]

(1) Write a Verilog code for an 8bit shift register with general controls in compliance with the requirements given below.



(2) Put the codes under simulation; observe and interpret the waveforms of output signals

(3) Try drawing the circuit diagram of the 8bit shift register, using 1bit D-reg, logic gates and components one deems as necessary.

a)) coding specifications:

** write behavior-level descriptions for the 8bit L/R shift-register.

** the module has load, L/R, clk, D_{in}, and I_{shift} as inputs, and D_{out} and O_{shift} as outputs.

** write Verilog codes for generating test data sequence as requested.

b)) circuit specifications:

** I/O layout: 5 input sets D_{in}[7:0], load, L/R, clk, I_{shift};

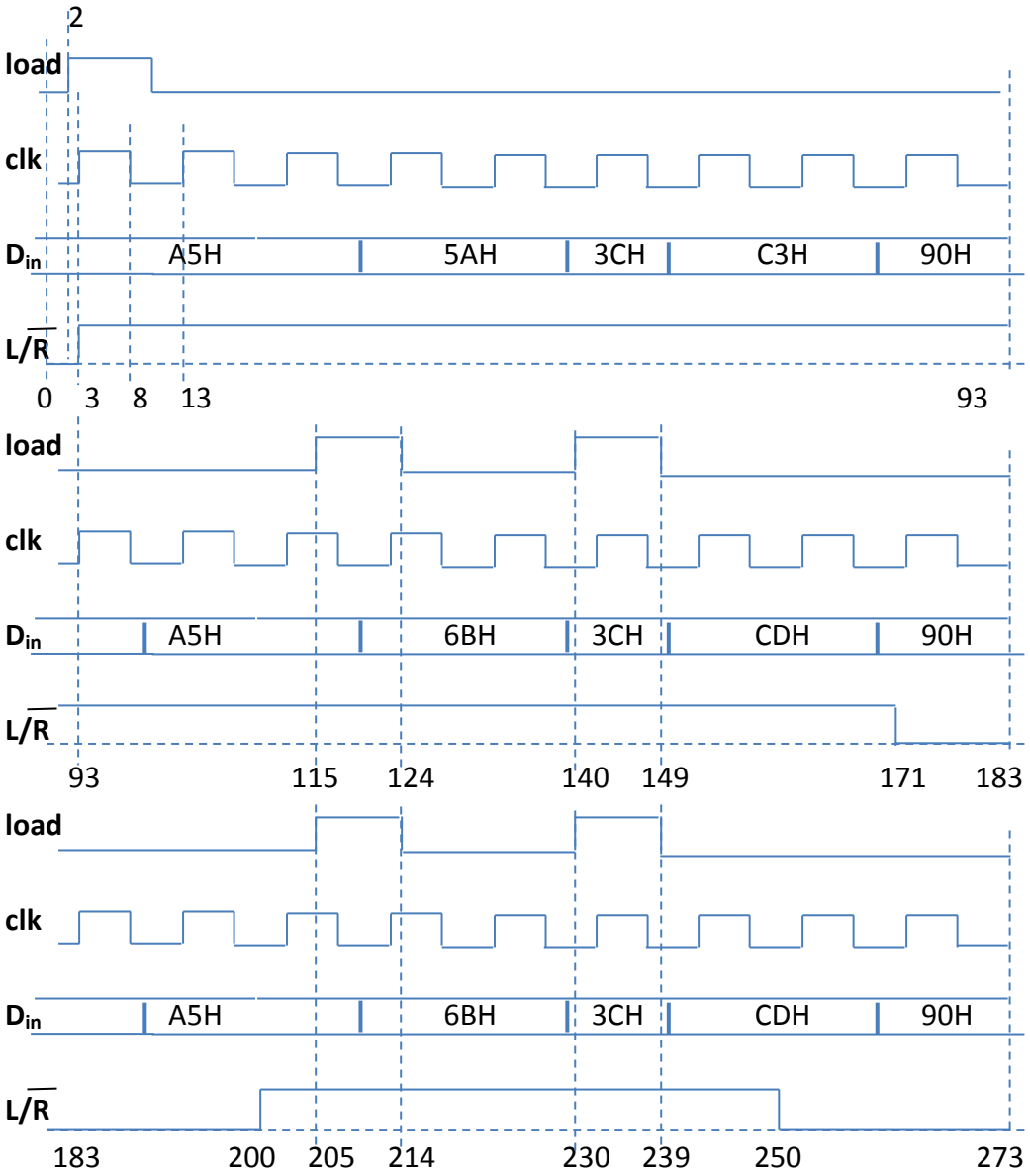
outputs D_{out}[7:0], O_{shift};

** Functional:

load	clk	L/$\overline{R}$	I_{shift}	D_{in}	O_{shift}	D_{out}
1	x	x	x	1s/0s	x	D_{in}
0	↑	1	1/0	x	D_{out}[7]	D_{out}[j]= D_{out}[j-1], j:7~1 D_{out}[0]=I_{shift}
		0	1/0	x	D_{out}[0]	D_{out}[j]= D_{out}[j+1], j:6~0 D_{out}[7]=I_{shift}

** Timing: no delay(s) to be considered in this task

c)) testing data:



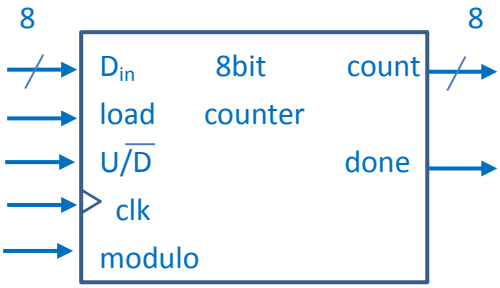
** the testing stops at time-unit 273;

** D_{in} changes at 2 time-unit after the falling edge of **clk**;

** derive unspecified timing instances using information revealed in the chart whenever necessary.

[TASK2]

(1) Write a Verilog code for an 8bit counter with general controls in compliance with the requirements given below.



(2) Put the codes under simulation; observe and interpret the waveforms of output signals

(3) Try drawing the circuit diagram of the 8bit shift register, using 1bit D-reg, logic gates and components one deems as necessary.

a)) coding specifications:

** write behavior-level descriptions for the 8bit counter in question.

** the module has D_{in}, load, U/$\overline{D}$, clk, modulo as inputs, and count and done as outputs.

** write Verilog codes for generating test data sequence as requested.

b)) circuit specifications:

** I/O layout: input sets D_{in}[7:0], load, U/$\overline{D}$, clk, modulo;

outputs count[7:0], done;

** Functional:

load	clk	U/$\overline{D}$*	D_{in}	mod*	c_end*	count	done
1	↑	1	1s/0s	0	0FFH	D_{in*}	0
1	↑	0	1s/0s	0	0H	D_{in*}	0
1	↑	1	1s/0s	1	D_{in*}	0	0
1	↑	0	1s/0s	1	0H	D_{in*}	0
0	↑	1	x	0	c_end*	count++	if count==c_end* { count=D_{in*}; done=1; } else done=0;

0	↑	0	x	0	c_end*	count--	if count==0
							{ count=D_{in*};
							done=1;
							}
							else done=0;

0	↑	1	x	1	c_end*	count++	if count==c_end*
							{ count=0;
							done=1;
							}
							else done=0;

0	↑	0	x	1	c_end*	count--	if count==c_end*
							{ count=D_{in*};
							done=1;
							}
							else done=0;

[note] ** modulo: 0 for binary counting

from D_{in} to 0FFH for $U/\overline{D}$==1

D_{in} to 0H for $U/\overline{D}$==0

1 for decimal counting

from 0 to D_{in} for U/D==1

D_{in} to 0H for $U/\overline{D}$==0

** mod*: internal copy of modulo

** $U/\overline{D}$*: internal copy of $U/\overline{D}$

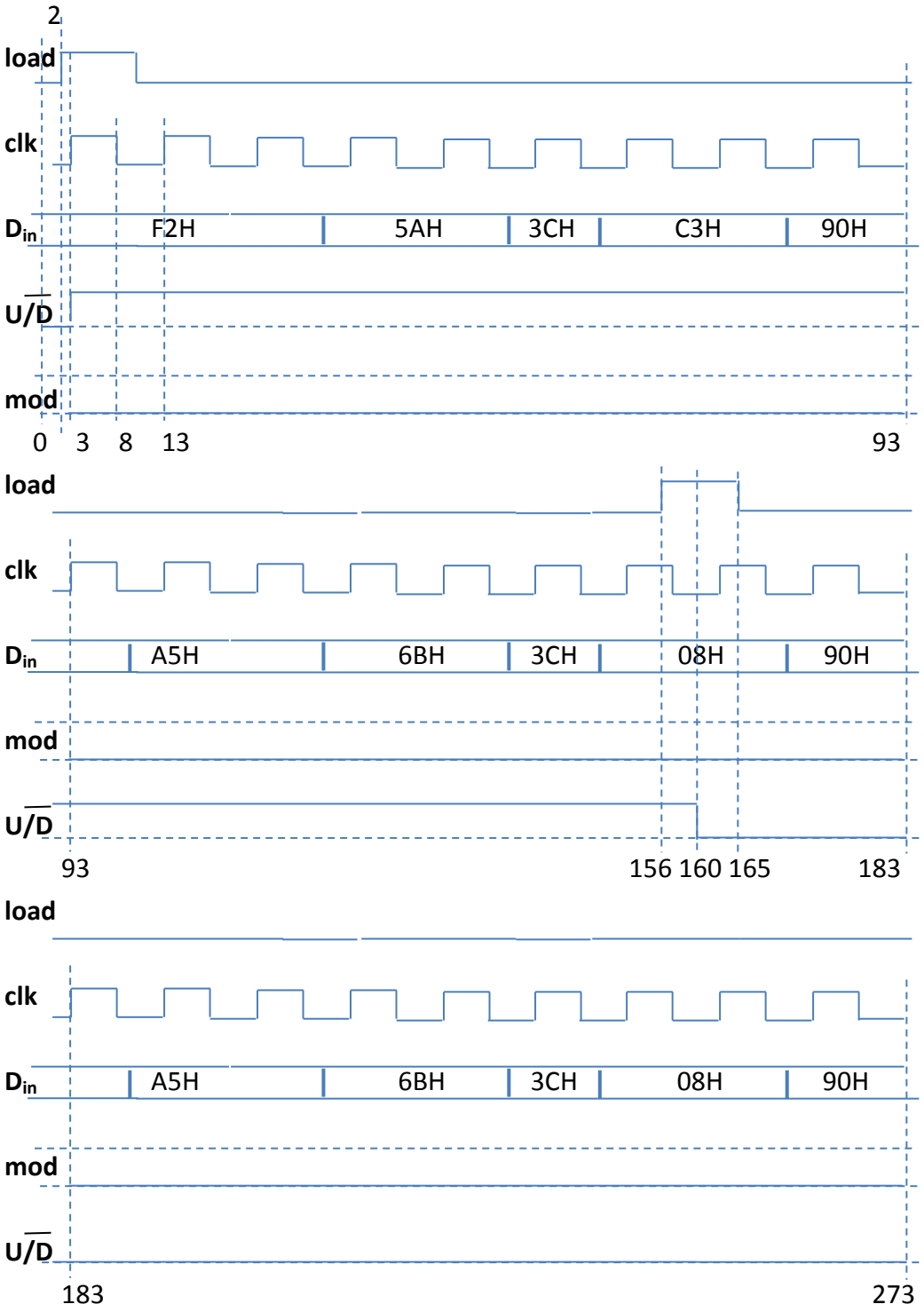
** D_{in*}: internal copy of D_{in} or 0 as the initial value of counting operation;

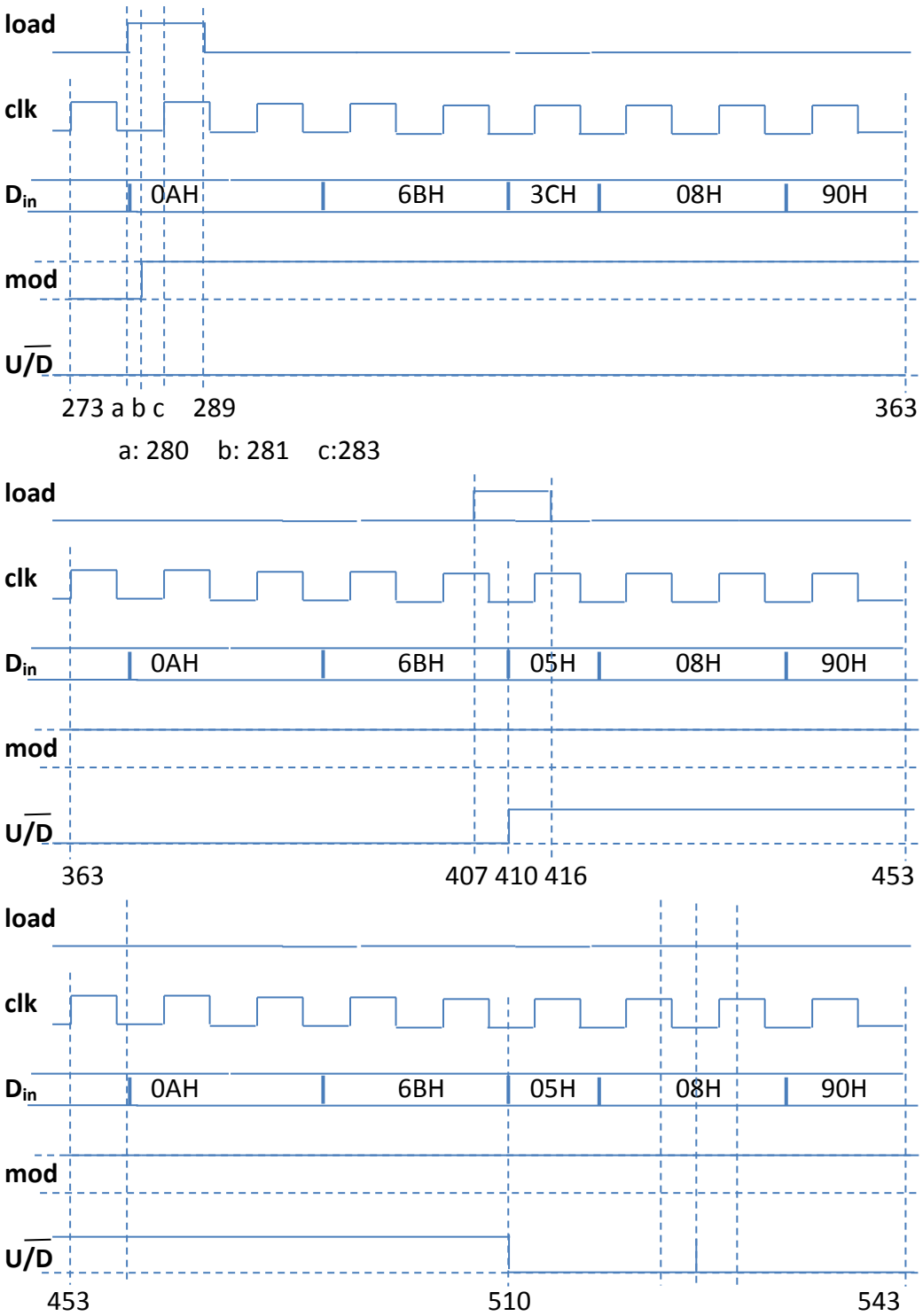
** c_end*: internal copy of 0H, 0FFH, or D_{in} as the ending value of counting;

** internal copies of input signals are set at clk rising edge.

** Timing: no delay(s) to be considered in this task

c)) testing data:





** the testing stops at time-unit 543;

** D_{in} changes at 2 time-unit after the falling edge of **clk**;

** derive unspecified timing instances using information revealed on the chart whenever necessary.

[4] Observations/Evaluation

a)) What would the shift register respond if D_{in} varied when **load** is high? (as indicated by the 2nd **load** in the testing sequence)

b)) What would be shift register respond if L/$\overline{R}$ switching occurred at shifting?

c)) In the function spec. of the 8bit counter, it is noted that inputs are strobed for internal copies at the rising edge of the clock when **load** is high.

** Could you think of any undesired situations which might occur at counting if there are no internal copies of $\overline{f}$ modulo, U/D, and D_{in} made available while **load** is in effect?

** While making internal copies of modulo, U/$\overline{D}$, and D_{in} at load-phase, the copying is triggered by clk↑ as noted in the function table. Is it possible making internal copies by using **load** alone without causing any undesired consequence?

d)) Difficulties encountered while working on the assigned tasks may more or less reflect one's deficiency in one or more aspects of the following:

** experiences on the circuit structures and operations of the three basic modules which are building blocks deployed in all digital systems;

** experiences on the mastery of Verilog and the tools of the developing environment;

** readiness for the lab-task at hand.

Inexperience won't be a problem, as it fades away eventually when one pours in time and dedication; whereas the third would present a serious issue. The so called "poor-luck-in-the-laboratory" would persist, by which one will always be haunted as long as the unsound attitude continues.

e)) It's expected that, after DigitLabwork#4, one should feel confident, or at least comfortable, in dealing with shifters and counters using Verilog. Hopefully this is the case of everyone in the class.