DigitLabwork#2
MUX/DEMUX/Parity-Check 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)) the functions of MUXs, DEMUXs and parity-check circuits;

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 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;

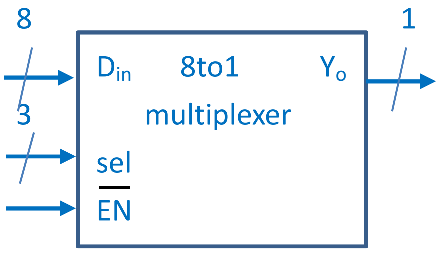
d)) Refer to “Development Context of Verilog Code” for operational details in running the developing system.

[3] Lab-task instruction

In digitlabwork#2, 3 tasks are assigned to every class members as given below.

[TASK1]

(1) Write a Verilog code for an 8to1 multiplexer with $\overline{EN}$ control and tri-state output in compliance with the requirements given below.



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

a)) coding specifications:

** write 3 modules of 8to1 multiplexer in gate-level, dataflow-level and behavior-level descriptions, respectively.

** the 3 modules will share the same inputs of data channels and controls, and each operates with respective 1-bit outputs.

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

b)) circuit specifications:

** I/O layout: 3 input sets $D_7D_6D_5D_4D_3D_2D_1D_0$,
sel_2-sel_0,
$\overline{EN}$;
output Y_o;

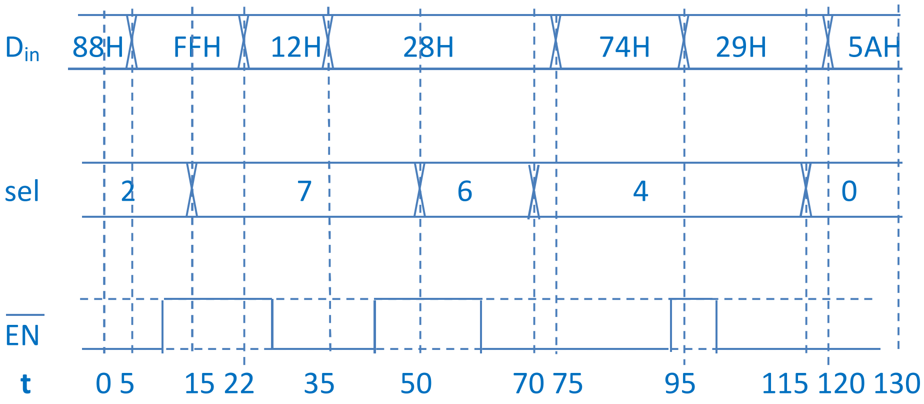
** Functional:

$\overline{EN}$	D_7~D_0	sel_2	sel_2	sel_0	Y_o
0	X	X			Z
1	X	0	0	0	D_0
		0	0	1	D_1
		0	1	0	D_2
		0	1	1	D_3
		1	0	0	D_4
		1	0	1	D_5
		1	1	0	D_6
		1	1	1	D_7

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

c)) testing data:

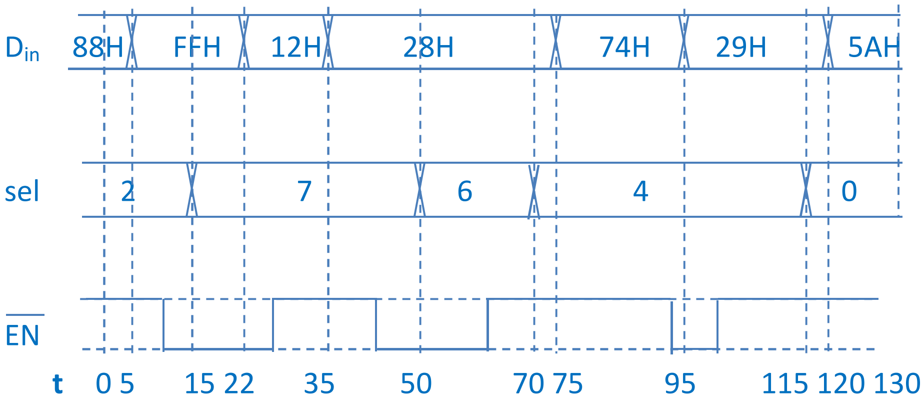
[test case 1]



** the testing stops at time-unit 130;

** determine the switching time instants for $\overline{EN}$ according to the waveform shown above.

[test case 2]

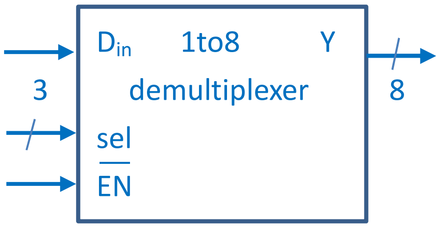


** the testing stops at time-unit 130;

** determine the switching time instants for $\overline{EN}$ according to the waveform shown above.

[TASK2]

(1) Write a Verilog code for an 1to8 demultiplexer with $\overline{EN}$ control and tri-state output in compliance with the requirements given below.



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

of output signals

a)) coding specifications:

** write 3 modules of 1to8 demultiplexer in gate-level, dataflow-level and behavior-level descriptions, respectively.

** the 3 modules will share the same inputs of data channel and controls, and each operates with respective 8-bit outputs.

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

b)) circuit specifications:

** I/O layout: 3 input sets D_{in},
sel_2-sel_0,
$\overline{EN}$;
output $Y_7Y_6Y_5Y_4Y_3Y_2Y_1Y_0$;

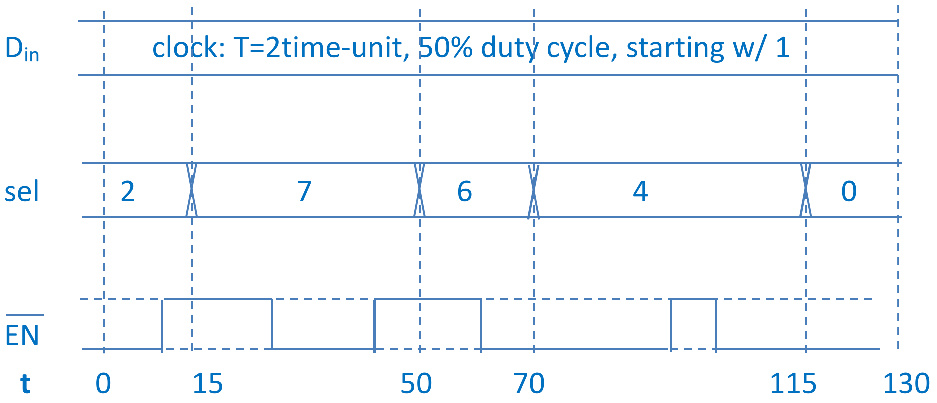
** Functional:

$\overline{EN}$	D_{in}	sel_2	sel_2	sel_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	X	X			Z	Z	Z	Z	Z	Z	Z	Z
1	X	0	0	0	Z	Z	Z	Z	Z	Z	Z	D_{in}
		0	0	1							D_{in}	Z
		0	1	0						D_{in}	Z	
		0	1	1					D_{in}	Z		
		1	0	0				D_{in}	Z			
		1	0	1			D_{in}	Z				
		1	1	0		D_{in}	Z					
		1	1	1	D_{in}	Z						

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

c)) testing data:

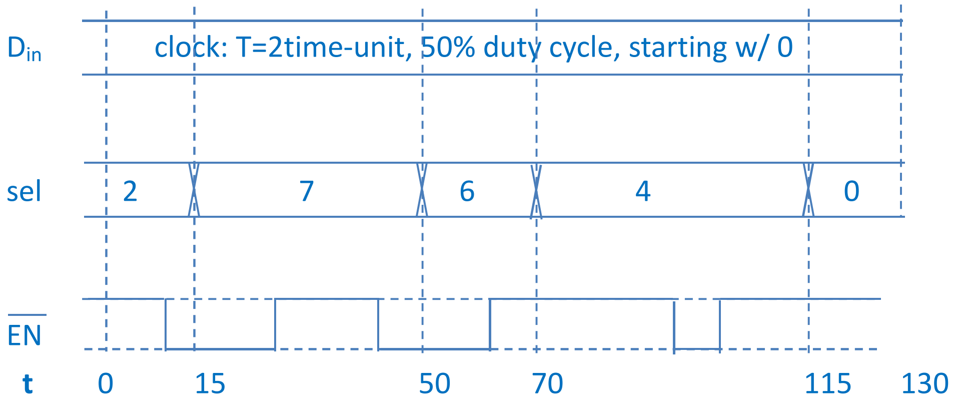
[test case 1]



** the testing stops at time-unit 130;

** determine the switching time instants for $\overline{EN}$ according to the waveform shown above.

[test case 2]

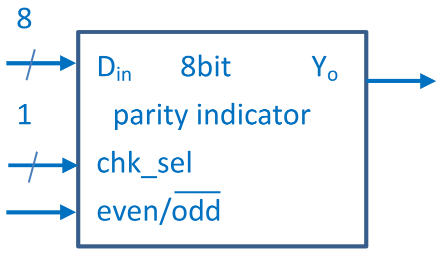


** the testing stops at time-unit 130;

** determine the switching time instants for $\overline{EN}$ according to the waveform shown above.

[TASK3]

(1) Write a Verilog code for an 8bit parity checking circuit in compliance with the requirements given below.



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

a)) coding specifications:

** write 3 modules of 8bit parity indicator in gate-level, dataflow-level and behavior-level descriptions, respectively.

** the 3 modules will share the same inputs of data channels and controls, and each operates with respective 1-bit outputs.

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

b)) circuit specifications:

** I/O layout: 3 input sets $D_7D_6D_5D_4D_3D_2D_1D_0$,
chk_1chk_0,
even/$\overline{odd}$;

output Y_o;

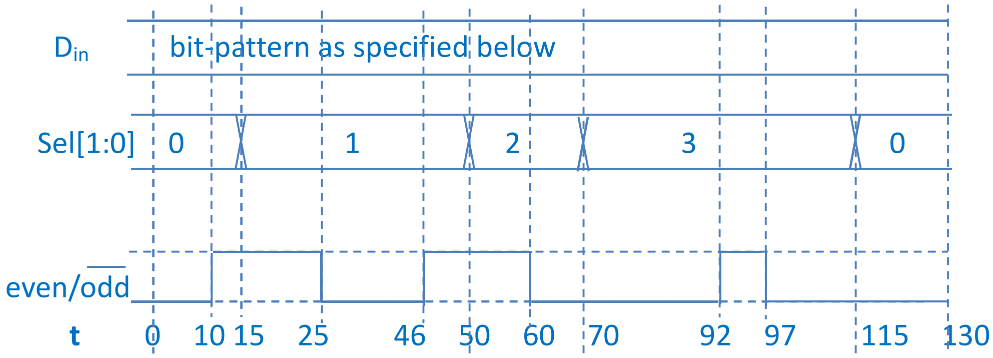
** Functional:

$\text{even}/\overline{\text{odd}}$	D_{in}	chk_1	chk_0	Y_o
0	X	0	0	L-half odd
		0	1	Even-bit odd
		1	0	Odd-bit odd
		1	1	full 8bit odd
1	X	0	0	L-half even
		0	1	Even-bit even
		1	0	Odd-bit even
		1	1	full 8bit even

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

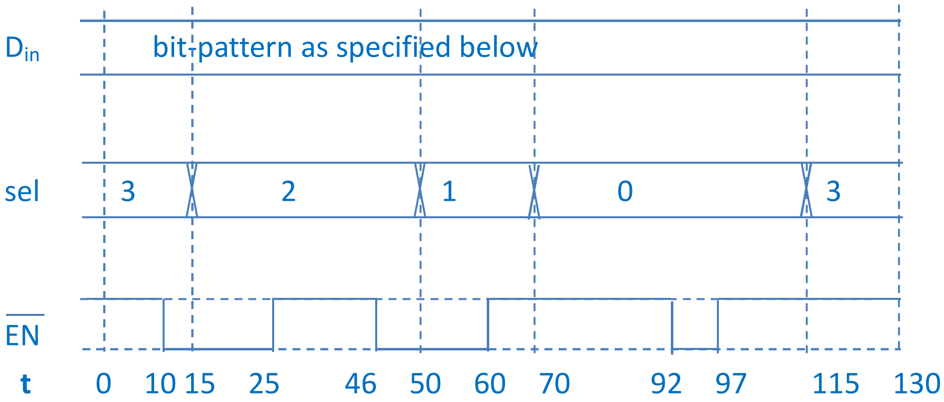
c)) testing data:

[test case 1]



D_{in}
bit-pattern as specified below
Sel[1:0]
0
1
2
3
0
$\text{even}/\overline{\text{odd}}$
t
0
10
15
25
46
50
60
70
92
97
115
130

[test case 2]



** D_{in}: repetitively varies from 00H to FFH, each pattern remains 1 time-unit;

** determine your own switching of even/odd and sel[1:0] after time-unit 130.

[4] Observations/Evaluation

a)) 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.

b)) It’s expected that, after DigitLabwork#2, one should feel confident, or at least comfortable, in dealing with circuit structures and operations of MUXs, DEMUXs, and parity-check circuits using Verilog. Hopefully this is the case of everyone in the class.