DigitLabwork#3

Addition/Subtraction/Multiplication/Division in Verilog codes

= =

[1] Training goals

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

a)) operation of circuits doing addition, subtraction, multiplication and division;

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;

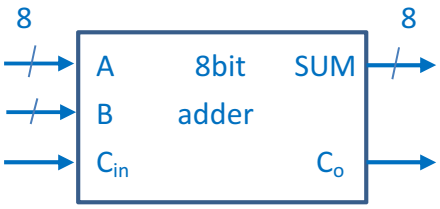
d)) Refer to “Development Context of Verilog Code” for operational details in running the developing system.

[3] Lab-task instruction

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

[TASK1]

(1) Write a Verilog code for an 8bit adder 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 adders in gate-level, dataflow-level and behavior-level descriptions, respectively.

** the 3 modules will share the same A,B and C_{in} inputs, and each generates respective SUM and C_o outputs.

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

b)) circuit specifications:

** I/O layout: 3 input sets A[7:0], B[7:0], C_{in};

outputs SUM[7:0], C_o;

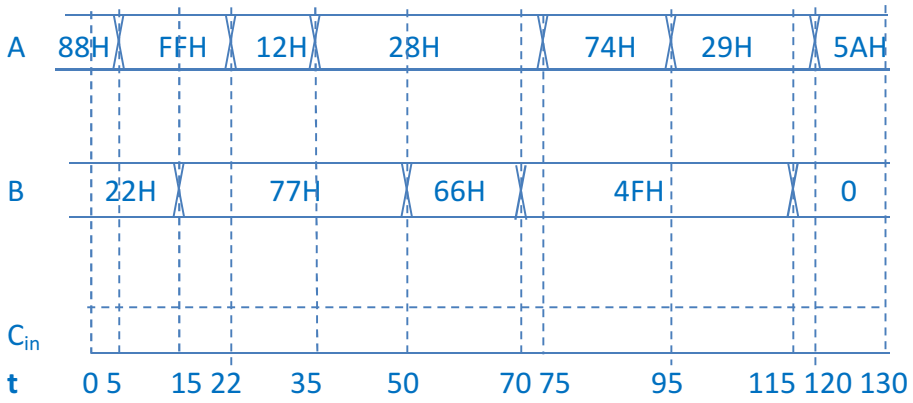
** Functional:

A	B	C_{in}	SUM	C_o
1s/0s	1s/0s	1/0	A+B+C_{in}	(A[7]&B[7])|(A[7]&C_6)|(B[7]&C6)

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

c)) testing data:

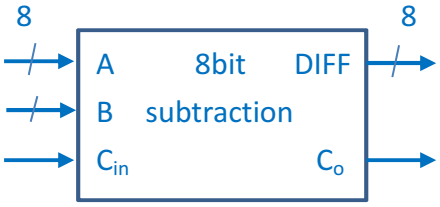
[test case 1]



** the testing stops at time-unit 130;

[TASK2]

(1) Write a Verilog code for a circuit capable of doing 8bit subtraction in compliance with the requirements given below.



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

a)) coding specifications:

** B is the data to be subtracted from A, A-B that is,

** using an 8bit adder as the underlying component is recommended,

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

** the 3 modules will share the same A, B and carry inputs, and each generates respective difference and carry outputs.

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

b)) circuit specifications:

** I/O layout: 3 input sets A[7:0], B[7:0],

C_{in};

output DIFF[7:0],

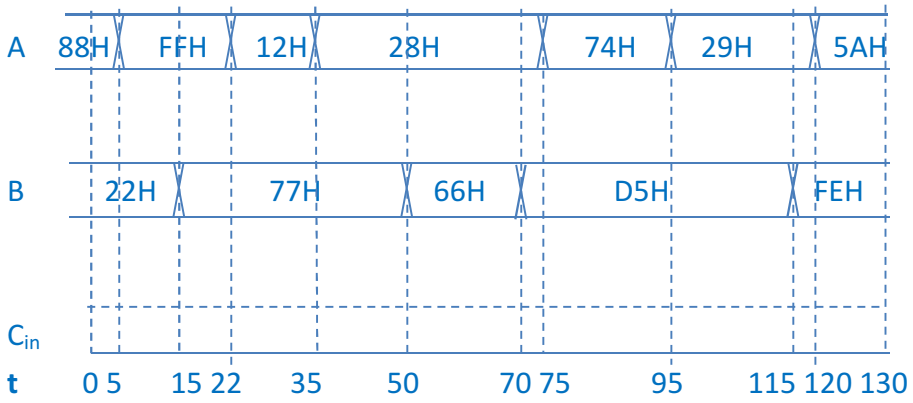
C_o;

** Functional:

			B2C=~B+1	
A	B	C_{in}	DIFF	C_o
1s/0s	1s/0s	1/0	A+B2C+C_{in}	(A[7]&B2C[7])| (A[7]&,C_6)| (B2C[7]&C6)

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

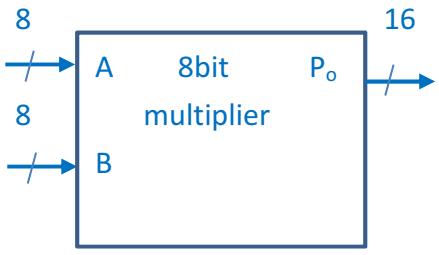
c)) testing data:



** the testing stops at time-unit 130;

[TASK3]

(1) Write a Verilog code for 8bit x 8bit unsigned multiplication in compliance with the requirements given below.



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

a)) coding specifications:

** multiplication done either by iterative/successive additions or by

Wallace-tree will suffice; however, realization by any better approaches known to you is encouraged,

** try not using a statement like Po=A*B in behavior-level description; coding in a way that more or less shows the procedure about how the multiplication being done, instead,

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

** the 3 modules will share the same A,B inputs, and each comes out with respective product.

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

b)) circuit specifications:

** I/O layout: 2 input sets A[7:0], B[7:0];

output P[15:0];

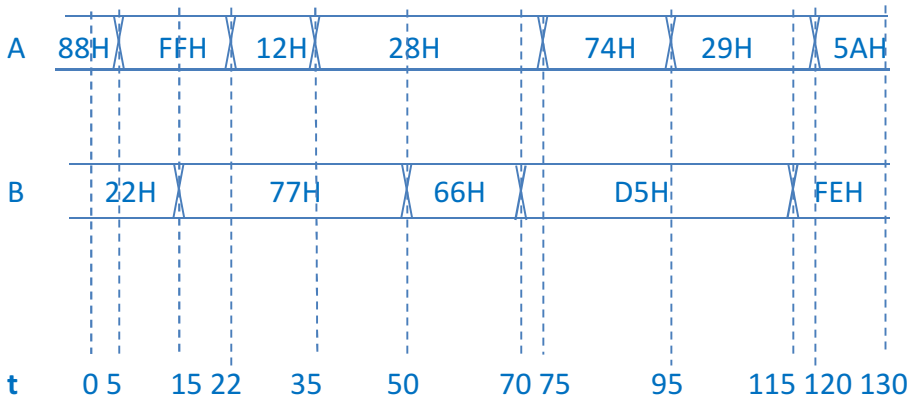
** Functional:

A	B	P
1s/0s	1s/0s	AxB

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

c)) testing data:

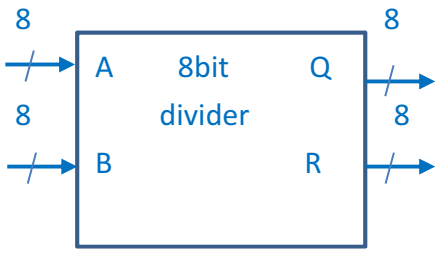
[test case 1]



** the testing stops at time-unit 130;

[TASK4]

(1) Write a Verilog code for 8bit/8bit unsigned division in compliance with the requirements given below.



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

a)) coding specifications:

** use iterative/successive subtraction as the underlying dividing mechanism, unless you know otherwise how the division could be done more elegantly,

** try not using a statement like Q=A/B in behavior-level description; coding in a way that more or less shows the procedure about how the division being done instead,

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

** the 3 modules will share the same A,B inputs, and each comes out respective quotient and remainder.

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

b)) circuit specifications:

** I/O layout: 2 input sets A[7:0], B[7:0];
output Q[7:0]; R[7:0]

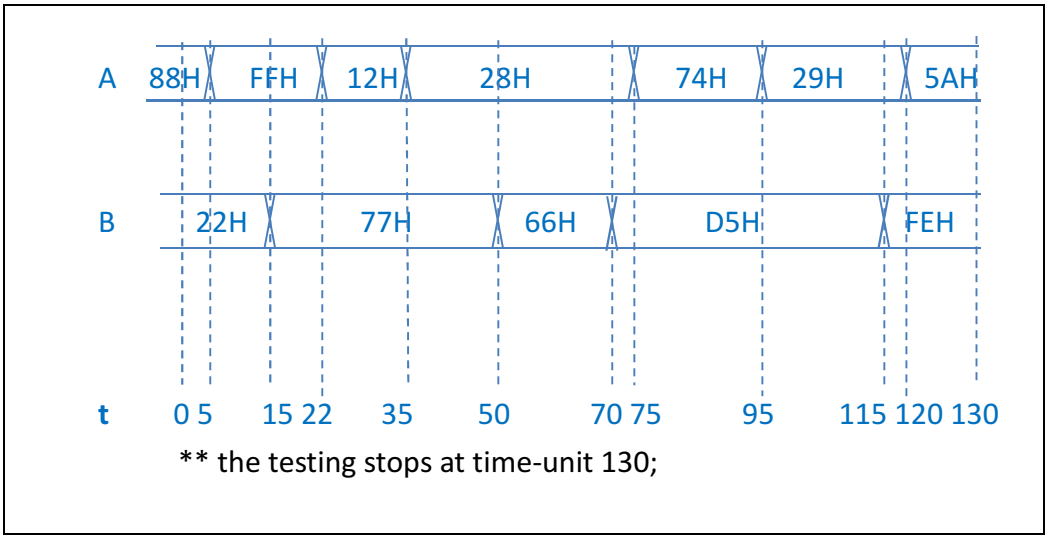
** Functional:

A	B	Q,R
1s/0s	1s/0s	A=BxQ+R

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

c)) testing data:

[test case 1]



** the testing stops at time-unit 130;

[4] Observations/Evaluation

a)) Try deriving fast carry C_6, C_5, C_4, C_3, C_2, C_1, C_0 in terms of A_6, A_5, A_4, A_3, A_2, A_1, A_0 and B_6, B_5, B_4, B_3, B_2, B_1, B_0, and then ask oneself:

** for any C_j, which part of A and B are involved if logic-level of the carry generating circuit is to be limited under, say, no more than 3?

** what does one see regarding the logic gates required for C_j generation as j getting higher?

b)) While observing the operation of subtracting circuit, tell about the implication of seeing and not seeing C_o generation, respectively.

c)) If, besides DIFF and C_o, a third output $O_{verflow}$ is needed while using 2 8bit subtraction modules for 16bit subtraction, what should be done?

d)) Dealing with iterative addition for adding up partial products during multiplication would seem require a sequential module. So is the multiplier combinational or sequential? What about the divider if iterative-subtraction being deployed inside?

e)) What more are to be done if multiplier/divider for signed integers is desired?

f)) 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.

g)) It’s expected that, after DigitLabwork#3, one should feel confident, or at least comfortable, in dealing with circuit doing addition, subtraction, multiplication and division using Verilog. Hopefully this is the case of everyone in the class.