Faculty of Engineering and Technology
Department of Electrical and Computer Engineering
ENCS 211 Digital Electronics and Computer Organization Lab
Second Semester 2020-2021
Practical Final Exam Questions

Introduction:

Each student will need to implement the problem by him/herself and show understanding of what is being done, isolate errors and introduce correction when needed. Each student will be given a set of problem that includes problems from **Area1 (Basic Digital Design), Area2 (ALU) and Area3 (Verilog/Quartus).**

The result needs to be demonstrated to the instructor/teaching assistant and the student will be evaluated on his/her ability to understand, change and diagnose errors in the design.

Please note that the questions can be repeated for different sections.

Evaluation Points:

Students will be evaluated on the following items (out of 100):

1. Clear sketch of the circuit to be implemented (**15%**).
2. Using the correct equipment (kit) and selecting inputs/outputs (**15%**).
3. Implementation, testing and locating errors and fixing them (**30%**).
4. Ability to change the circuit to achieve a related functionality (**30%**).
5. Time management (10%)

Good luck

Problem Set 1:

a) Build a 2-to-4 decoder using gates. Then use a chip based 3-to-8 decoder to implement a basic function, say a full adder (3 inputs). Sketch the design before the implementation and show the designs to your instructor/TA.

b) Operate the ALU unit so that it performs the following Logic and Arithmetic operations, with all inputs and outputs are active High. Example operations: logic xor ($\oplus$), Arithmetic Addition/subtraction with/without carry.

c) Create schematic symbols as shown in the figure, Use : $Y_3Y_2Y_1Y_0 = 0111$. What does this circuit do? Use Verilog to implement the circuit and Run a meaningful simulation for this circuit.
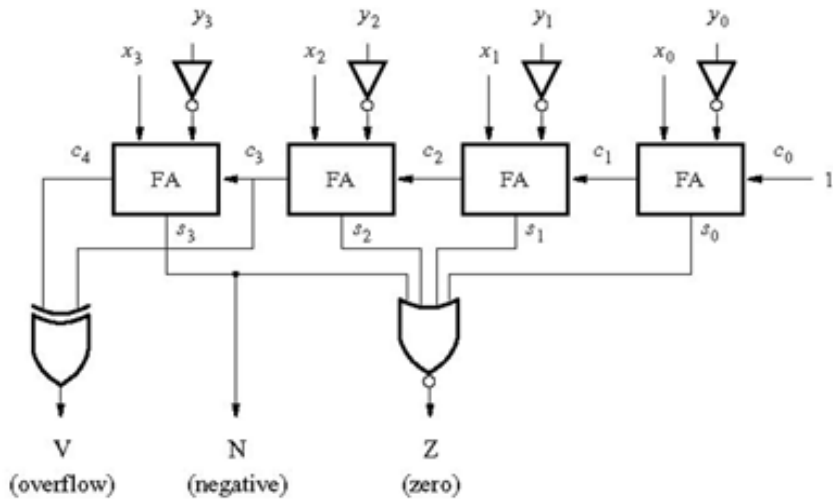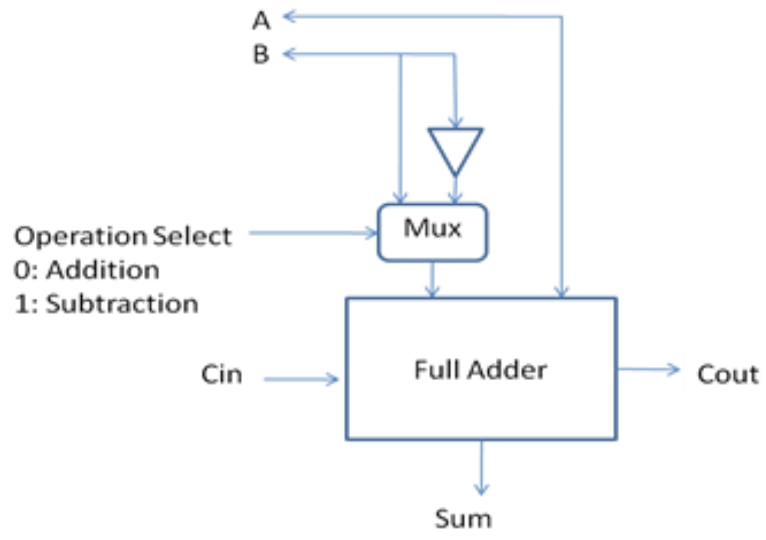

Problem Set 2:

a) Build a basic 2-to-1 MUX using gates. Then use a chip based 4-to-1 MUX to implement a basic function, say a full adder, or something similar.

b) Operate the ALU unit so that it performs the following Logic and Arithmetic operations, with all inputs and outputs are active High. Example operations: logic xor ($\oplus$), Arithmetic Addition/subtraction with/without carry.

c) Use Verilog HDL to implement a 2-to-1 MUX. Use Verilog HDL to implement a Full Adder. Create schematic symbols for both the MUX and the Full adder, then connect them as shown in the figure. Run a meaningful simulation for this circuit.

A

B

Operation Select
0: Addition
1: Subtraction

Mux

Cin → Full Adder → Cout

Sum

Problem Set 3:

a) Build a basic 2-bit parity checker using gates. Then use a chip based (8-bit) parity checker and connect it so that it checks parity for 5 inputs only. Sketch the design before the implementation and show the designs to your instructor/TA.
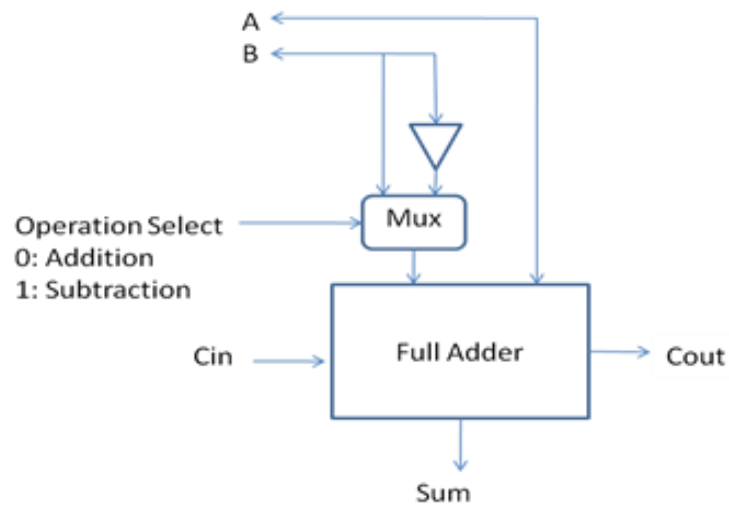
b) Operate the ALU unit so that it performs the following Logic and Arithmetic operations, with all inputs and outputs are active High. Example operations: logic xor ($\oplus$), Arithmetic Addition/subtraction with/without carry.

c) Create schematic symbols as shown in the figure, Use : $Y_3Y_2Y_1Y_0 = 0111$. What does this circuit do? Use Verilog to implement the circuit and Run a meaningful simulation for this circuit.
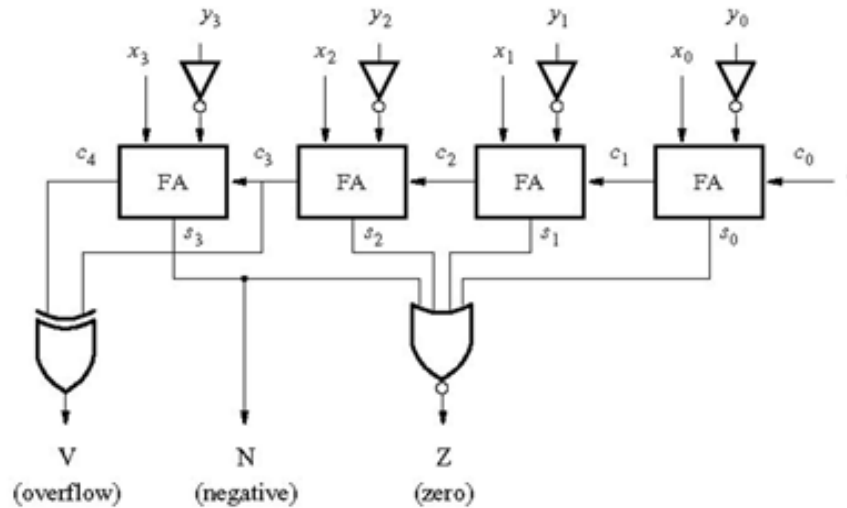
Problem Set 4:

a) Build a basic 2-bit full adder from gates. Use the result and 2 shift registers to design and implement a serial adder to a chip based parity checker and connect it so that it checks parity for 5 inputs only. Sketch the design before the implementation and show the designs to your instructor/TA.

b) Operate the ALU unit so that it performs the following Logic and Arithmetic operations, with all inputs and outputs are active High. Example operations: logic xor ($\oplus$), Arithmetic Addition/subtraction with/without carry.

c) Use Verilog HDL to implement a 2-to-1 MUX. Use Verilog HDL to implement a Full Adder. Create schematic symbols for both the MUX and the Full adder, then connect them as shown in the figure. Run a meaningful simulation for this circuit.
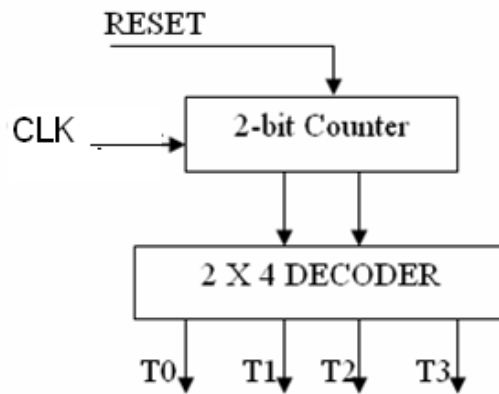
Problem Set 5:

a) Build a basic 2-bit parity checker from gates. Use the result and 2 shift registers to design and implement a serial parity checker for 4 bits. Sketch the design before the implementation and show the designs to your instructor/TA.
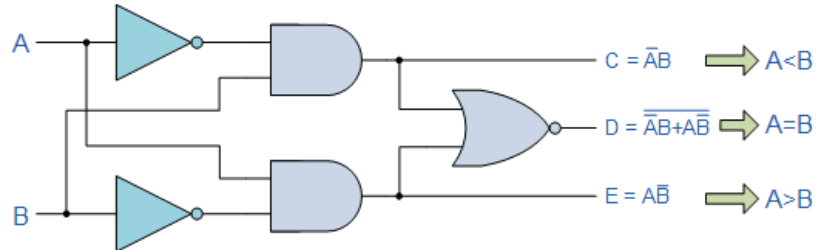
b) Operate the ALU unit so that it performs the following Logic and Arithmetic operations, with all inputs and outputs are active High. Example operations: logic xor ($\oplus$), Arithmetic Addition/subtraction with/without carry.

c) Use Verilog HDL to implement a 2-bit counter with direct reset input (RESET). Use Verilog HDL to implement a 2-to-4 Decoder. Create schematic symbols for both the counter and the decoder, then connect them as shown in figure below. Run a meaningful simulation for this circuit.
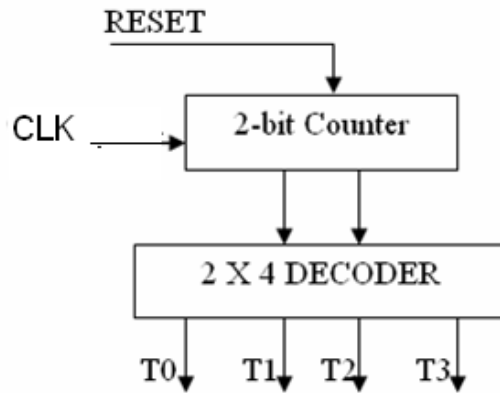
Problem Set 6:

a) Use the breadboard to implement the circuit below (2 inputs, 3 outputs). We will give you the chip and its datasheet and you have to do the rest. The circuit could be different for different students/sections. Sketch the design before the implementation and show the designs to your instructor/TA.
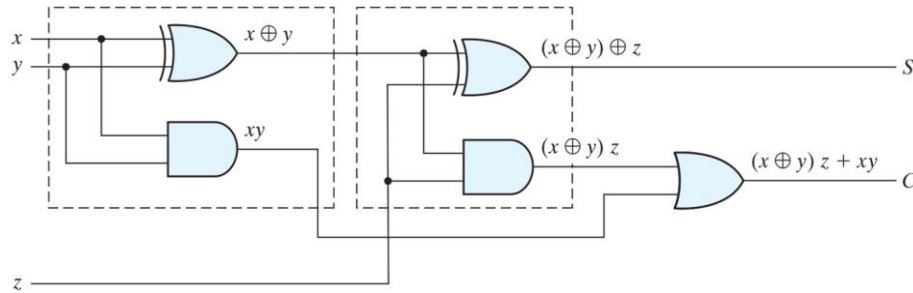

$C = \bar{A}B \Rightarrow A<B$

$D = \overline{\overline{A}B + A\bar{B}} \Rightarrow A=B$

$E = A\bar{B} \Rightarrow A>B$

b) Operate the ALU unit so that it performs the following Logic and Arithmetic operations, with all inputs and outputs are active High. Example operations: logic xor ($\oplus$), Arithmetic Addition/subtraction with/without carry.

c) Use Verilog HDL to implement a 2-bit counter with direct reset input (RESET). Use Verilog HDL to implement a 2-to-4 Decoder. Create schematic symbols for both the counter and the decoder, then connect them as shown in figure below. Run a meaningful simulation for this circuit.
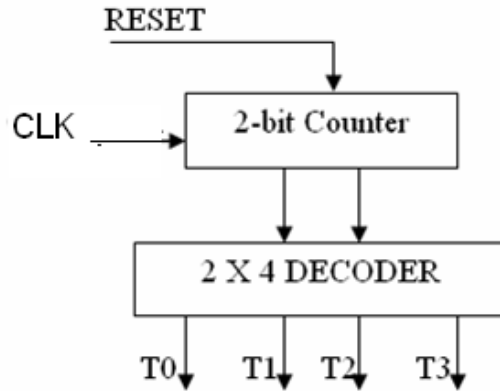
Problem Set 7:

a) Use the breadboard to implement the circuit below. We will give you the chip and its data and you have to do the rest. The circuit could be different for different students/sections.

b) Operate the ALU unit so that it performs the following Logic and Arithmetic operations, with all inputs and outputs are active High. Example operations: logic xor ($\oplus$), Arithmetic Addition/subtraction with/without carry.

c) Use Verilog HDL to implement a 2-bit counter with direct reset input (RESET). Use Verilog HDL to implement a 2-to-4 Decoder. Create schematic symbols for both the counter and the decoder, then connect them as shown in figure below. Run a meaningful simulation for this circuit.
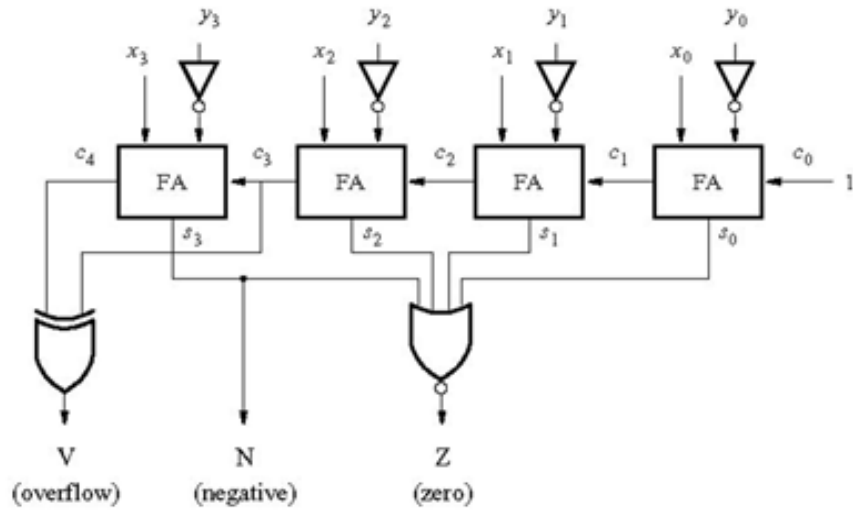
Problem Set 8:

a) Given $F(A, B, C) = \sum m(0,1,6,7)$, construct the truth table for the F. Then draw the logic circuit and implement F in the breadboard using the basic gates.
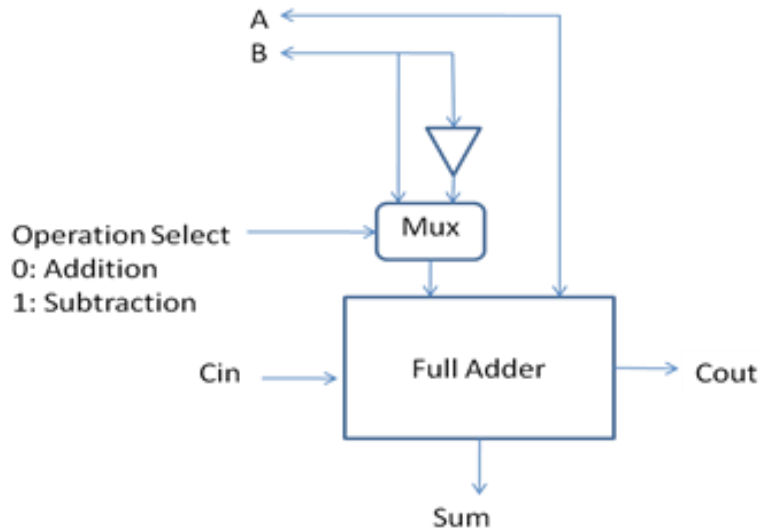
b) Operate the ALU unit so that it performs the following Logic and Arithmetic operations, with all inputs and outputs are active High. Example operations: logic xor ($\oplus$), Arithmetic Addition/subtraction with/without carry.

c) Create schematic symbols as shown in the figure, Use : $Y_3Y_2Y_1Y_0 = 0111$. What does this circuit do? Use Verilog to implement the circuit and Run a meaningful simulation for this circuit.
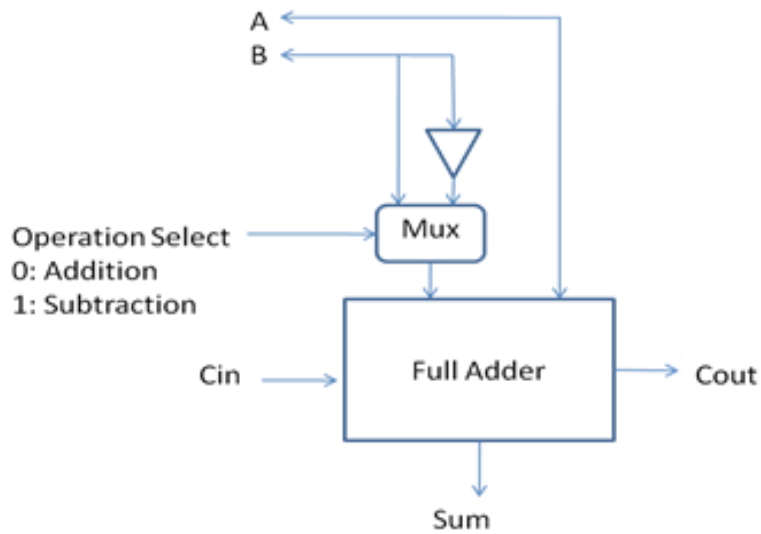
Problem Set 9:

a) Connect the BCD counter and show its operation on a 7-segment display. Modify the circuit so that it counts only to 3 then back to 0.

b) Operate the ALU unit so that it performs the following Logic and Arithmetic operations, with all inputs and outputs are active High. Example operations: logic xor ($\oplus$), Arithmetic Addition/subtraction with/without carry.

c) Use Verilog HDL to implement a 2-to-1 MUX. Use Verilog HDL to implement a Full Adder. Create schematic symbols for both the MUX and the Full adder, then connect them as shown in the figure. Run a meaningful simulation for this circuit.

Problem Set 10:

a) Construct a shift register (3 bits) from Flip_Flops and show its operation. Load the register with the values 101 then 010. Show how to operate it as a ring counter.

b) Operate the ALU unit so that it performs the following Logic and Arithmetic operations, with all inputs and outputs are active High. Example operations: logic xor ($\oplus$), Arithmetic Addition/subtraction with/without carry.

c) Use Verilog HDL to implement a 2-to-1 MUX. Use Verilog HDL to implement a Full Adder. Create schematic symbols for both the MUX and the Full adder, then connect them as shown in the figure. Run a meaningful simulation for this circuit.

Problem Set 11:

a) Construct a 4-bit odd parity generator using an 8x1 multiplexer and any basic gates if needed. Connect the four input bits to switches and Leds, also connect the output parity bit to Led.
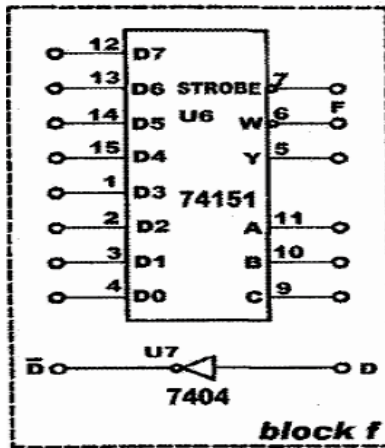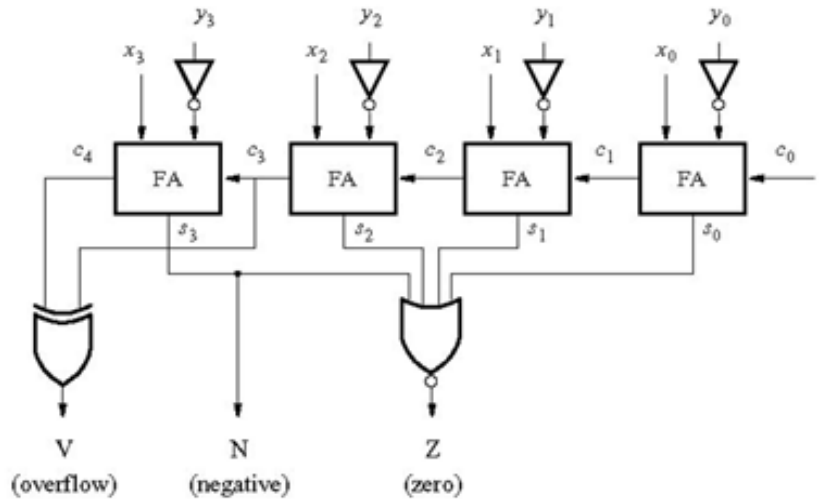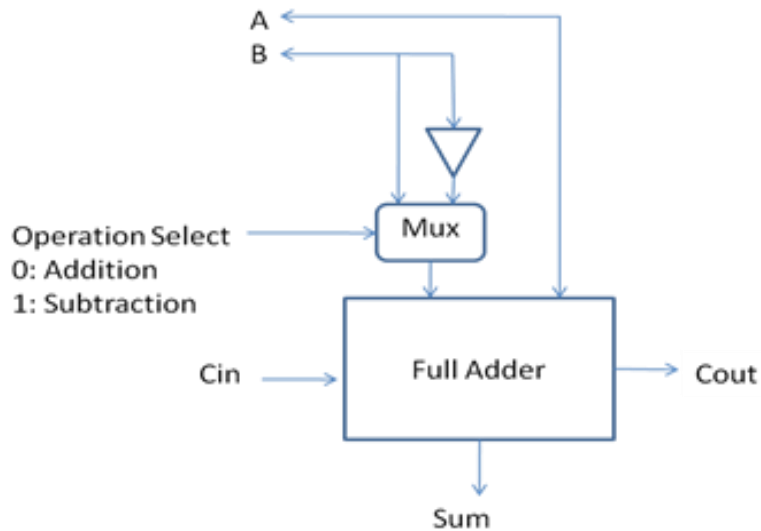

Figure.1: 8-to-1 MUX (KL-26004 block f)

b) Operate the ALU unit so that it performs the following Logic and Arithmetic operations, with all inputs and outputs are active High. Example operations: logic xor ($\oplus$), Arithmetic Addition/subtraction with/without carry.

c) Create schematic symbols as shown in the figure, Use : $Y_3Y_2Y_1Y_0 = 0111$. What does this circuit do? Use Verilog to implement the circuit and Run a meaningful simulation for this circuit.
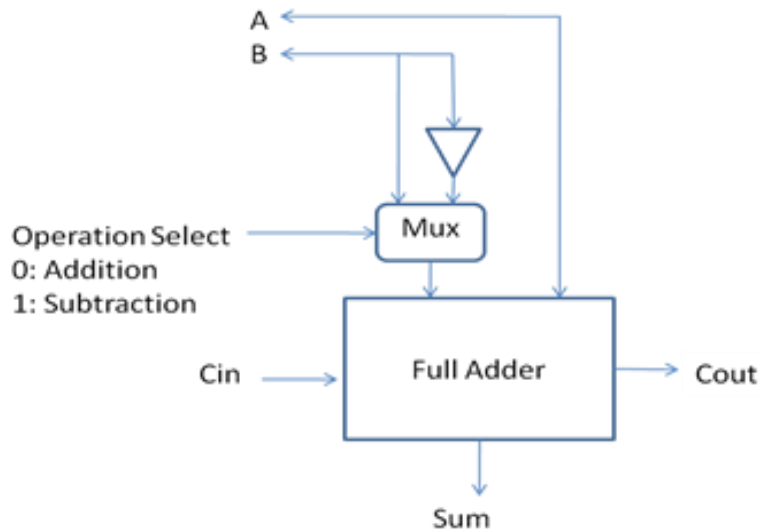
Problem Set 12:

a) Use JK Flip-Flops to construct a 2-bit counter.

b) Operate the ALU unit so that it performs the following Logic and Arithmetic operations, with all inputs and outputs are active High. Example operations: logic xor ($\oplus$), Arithmetic Addition/subtraction with/without carry.

c) Use Verilog HDL to implement a 2-to-1 MUX. Use Verilog HDL to implement a Full Adder. Create schematic symbols for both the MUX and the Full adder, then connect them as shown in the figure. Run a meaningful simulation for this circuit.

Problem Set 13:

a) Design and Implement one bit **active high** comparator using basic logic gates.

b) Operate the ALU unit so that it performs the following Logic and Arithmetic operations, with all inputs and outputs are active High. Example operations: logic xor ($\oplus$), Arithmetic Addition/subtraction with/without carry.

c) Use Verilog HDL to implement a 2-to-1 MUX. Use Verilog HDL to implement a Full Adder. Create schematic symbols for both the MUX and the Full adder, then connect them as shown in the figure. Run a meaningful simulation for this circuit.
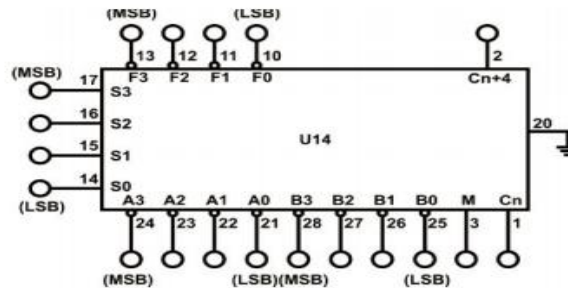
Problem Set 14:

a) Using the following, Show how you would implement the functions: $A + B, A - B, A.\bar{B}$. Can be different for different functions.
  1. IT-3000Basic Electricity Circuit Lab.
  2. IT-3003 Module.

b) Use the ALU 74181 in the figure below and the tables (one is enough; second maybe, need to check the new kit) to implement : $A + B, A - B, A.\bar{B}$.


And given that under M = 1 the circuit performs the following arithmetic and logic functions according to Table 11.1.

Input selection				M=H	Output			
S3	S2	S1	S0	Cn=L	F3	F2	F1	F0
0	0	1	0	A				
0	0	1	1	~A				
0	1	0	0	B				
0	1	0	1	~B				
0	1	1	0	A&B				
0	1	1	1	A×B				
1	0	0	0	A^B				
1	0	0	1	A×(~B)				
1	0	1	0	(~A)×B				
1	0	1	1	(~A)×(~B)				

SELECTION				ACTIVE-HIGH DATA		
				M = H	M = L; ARITHMETIC OPERATIONS	
S3	S2	S1	S0	LOGIC FUNCTIONS	$\bar{C}_n$ = H (no carry)	$\bar{C}_n$ = L (with carry)
L	L	L	L	F = $\bar{A}$	F = A	F = A PLUS 1
L	L	L	H	F = $\overline{A + B}$	F = A + B	F = (A + B) PLUS 1
L	L	H	L	F = $\bar{A}$B	F = A + $\bar{B}$	F = (A + $\bar{B}$) PLUS 1
L	L	H	H	F = 0	F = MINUS 1 (2's COMPL)	F = ZERO
L	H	L	L	F = $\overline{AB}$	F = A PLUS A$\bar{B}$	F = A PLUS A$\bar{B}$ PLUS 1
L	H	L	H	F = $\bar{B}$	F = (A + B) PLUS A$\bar{B}$	F = (A + B) PLUS A$\bar{B}$ PLUS 1
L	H	H	L	F = A ⊕ B	F = A MINUS B MINUS 1	F = A MINUS B
L	H	H	H	F = A$\bar{B}$	F = A$\bar{B}$ MINUS 1	F = A$\bar{B}$
H	L	L	L	F = $\bar{A}$ + B	F = A PLUS AB	F = A PLUS AB PLUS 1
H	L	L	H	F = $\overline{A ⊕ B}$	F = A PLUS B	F = A PLUS B PLUS 1
H	L	H	L	F = B	F = (A + $\bar{B}$) PLUS AB	F = (A + $\bar{B}$) PLUS AB PLUS 1
H	L	H	H	F = AB	F = AB MINUS 1	F = AB
H	H	L	L	F = 1	F = A PLUS A	F = A PLUS A PLUS 1
H	H	L	H	F = A + $\bar{B}$	F = (A + B) PLUS A	F = (A + B) PLUS A PLUS 1
H	H	H	L	F = A + B	F = (A + $\bar{B}$) PLUS A	F = (A + $\bar{B}$) PLUS A PLUS 1
H	H	H	H	F = A	F = A MINUS 1	F = A

Table 11.1

c) Create schematic symbols as shown in the figure, Use : $Y_3Y_2Y_1Y_0 = 0111$. What does this circuit do? Use Verilog to implement the circuit and Run a meaningful simulation for this circuit.
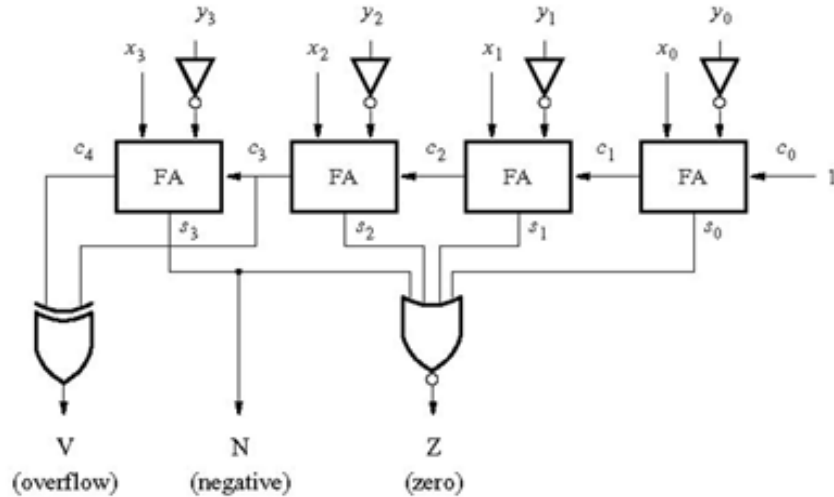
Evaluation Points:

6. Clear sketch of the circuit to be implemented (**10%**).
7. Using the correct equipment (kit) and selecting inputs/outputs (**10%**).
8. Implementation, testing and locating error and fixing them (**20%**).
9. Ability to change the circuit to do related functionality (**20%**).

Good luck