CS2100 Computer Organisation
Lab #7: 3-bit Majority Logic Circuit

Remember to bring this along to your lab. Prepare your report before attending the lab!

[This document is available on LumiNUS and module website http://www.comp.nus.edu.sg/~cs2100]

Name: _____ Student No: _____

Lab Group: _____

Objectives:

In this experiment, you will design, connect and test a *3-bit Majority Logic Circuit*. The design of this circuit MUST be prepared BEFORE your lab session or you may not have time to complete the experiment.

Please submit your report and leave the lab by latest 10 minutes before the hour.

IC chips:

1. One **74LS00** chip (QUAD 2-input NAND gates)
2. One **74LS20** chip (DUAL 4-input NAND gates)

The pin configurations for the chips are shown in step 5 below.

Introduction:

A **3-bit majority logic** accepts three input bits. When the number of 1 among these input bits is more than the number of 0, we say that 1 is a majority. The 3-bit majority logic is to output TRUE (1) if 1 is a majority; otherwise it outputs FALSE (0).

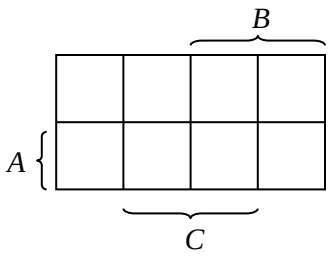
Procedure:

1. Complete the truth table below. The input bits are *A*, *B* and *C*. The output is *F*. For example, if *ABC* = 011, then *F* is 1.

A	*B*	*C*	*F*
0	0	0	
0	0	1	
0	1	0	
0	1	1	1
1	0	0	
1	0	1	
1	1	0	
1	1	1	

$F = \Sigma\, m($ _____ $)$

$F = \Pi\, M($ _____ _ $)$



Simplified SOP expression for *F*:

$F =$ _____

2. Write the **sum-of-minterms** expression in Σm notation and **product-of-maxterms** expression in ΠM notation for *F* above.
3. Fill in the **K-map** for *F* above and write the **simplified SOP expression** for *F*. Remember to write the dot symbol (.) for the AND operation.

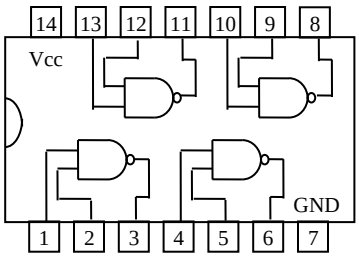
4. Draw the **logic diagrams** (neatly!) to implement *F* in each of the following circuits:

Using 2-level AND-OR circuit

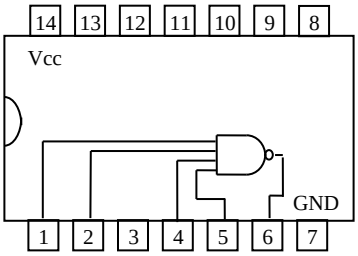
Using 2-level NAND circuit

5. We will implement the circuit using NAND gates only. A useful step before constructing your circuit on the logic trainer is to plan the wiring. Draw your wiring plan below.

74LS00

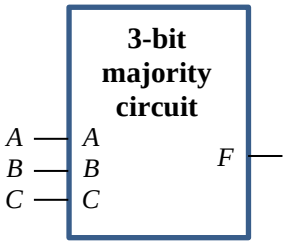


74LS20 (partial)



6. Construct your circuit on the logic trainer and **show it to your Lab TA.**
(If you encounter any problem with your circuit, use the logic probe to check it.)

7. You have implemented *F* above. Now, suppose you want to implement a 3-bit minority logic circuit with output *G* (that is, *G* is 1 when there are more 0s than 1s in the inputs, or 0 otherwise). How do you obtain *G* from *F* by using the fewest number of NAND gates and no other logic gate? Complete the diagram below, where the block diagram contains the circuit you drew in step 4 above.



Marking Scheme: Report (18 marks), Circuit (7 marks); Total: 25 marks.