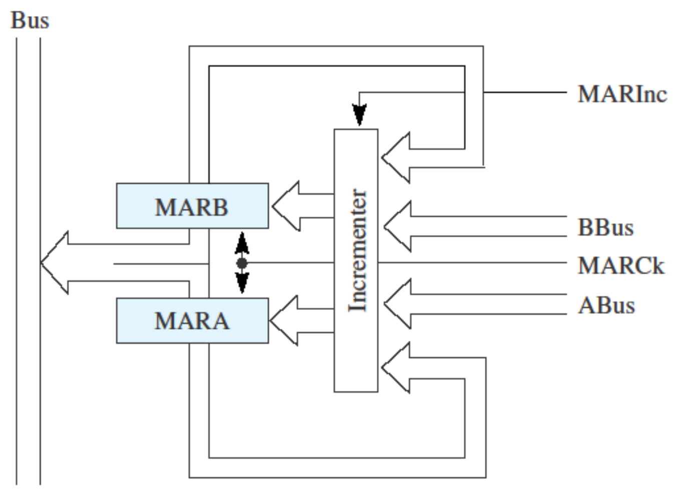
COMP262-01 LAB9, CH20 Digital Logic and Gates

NOTE: You may choose to use free hand drawing in preparing the requested Diagram and then 'scanning' and pasting into a WORD docx or converting it to a .pdf…

Specialized Hardware Units
When you coded the FO() steps for the Indirect address mode(MIC1), you probably noticed how many cycles were required to add 1 to an address because of the need to access bytes in memory at successive locations. It takes three cycles to (1) add 1 to the low-order byte of the address, (2) add a possible carry to the high-order byte, and (3) transfer the two-byte word to the MAR. The problem is that the A L U is a general-purpose device designed to do many operations on ONLY eight-bit quantities. To alleviate the observed bottleneck, you need a special-purpose circuit to add 1 to a 16-bit quantity (the memory address).
Figure 1(below) shows one possibility, done by inserting a combinational circuit between the A and B buses and the MAR, that includes feedback from MARA and MARB.
The control line labeled MARInc works as follows:
• MARInc = 1. The 16-bit sum MAR + 1 passes through.
• MARInc = 0. ABus and BBus pass through unchanged.

TO DO:
Design the **Incrementer** circuit of Figure 1.
Draw the Logic Diagram using:
Half-Adders to implement a **ripple-carry** addition for the increment and
Any combination of gates from the functionally complete set: AND/OR/NOT .
Please use ellipses(…) to shorten/simplify the size of the diagram, since it consists of 33 inputs and 16 outputs.

Figure 1 – The INCREMENTER circuit


Bus
MARInc
MARB
Incrementer
BBus
MARCk
ABus
MARA