Problem Statement: Implement a **4-bit ALU** that performs the operations indicated in the table below. The accumulator register bits (AQ[3:0]) should be routed to TOPRED[3:0] and the condition code flags (CF,ZF,NF,VF) should be routed to TOPRED[7:4]. Right pushbutton S1 should be used to clock the ALU (a bounceless switch module for it is provided). DIP[3:0] should be used to provide data inputs and routed to BOTRED[3:0]. DIP[7] should be used to provide the ALU enable signal (ALE) and routed to BOTRED[7]. DIP[6:5] should be used to provide function select signals (ALX, ALY) and routed to BOTRED[6:5]. Left pushbutton S2 should be used to asynchronously reset the accumulator and condition code registers. The ALU state (accumulator and condition code registers) should not change when ALE=0. The *function mnemonic* selected (LdA, Add, Sub, And) should be displayed on 7-segment displays DIS3..DIS1 (blank if ALE=0). Finally, the adder/subtractor should be based on a CLA design.

ALE	ALX	ALY	*Function Performed*	CF	ZF	NF	VF
1	0	0	LDA: AQ[3:0]←DIP[3:0]	0	⇳	⇳	0
1	0	1	ADD: AQ[3:0]←AQ[3:0]+DIP[3:0]	⇳	⇳	⇳	⇳
1	1	0	SUB: AQ[3:0]←AQ[3:0]–DIP[3:0]	⇳	⇳	⇳	⇳
1	1	1	AND: AQ[3:0]←AQ[3:0]∩DIP[3:0]	-	⇳	⇳	-
0	d	d	(*no operation – retain state*)	-	-	-	-

Follow the steps outlined below in completing this lab practical design problem:

1. [2 pts] Route DIP[3:0] to BOTRED[3:0] and DIP[7:5] to BOTRED[7:5].
2. [8 pts] Display the function mnemonic selected on DIS3..DIS1.
3. [12 pts] Display the AQ[3:0] results for each of the four functions on TOPRED[3:0].
4. [8 pts] Display the condition code register results (CF,ZF,NF,VF) for each of the four ALU functions on TOPRED[7:4], respectively – note that the state of a flag that is *unaffected* by a given operation (denoted as "-" in the table above) should be maintained.