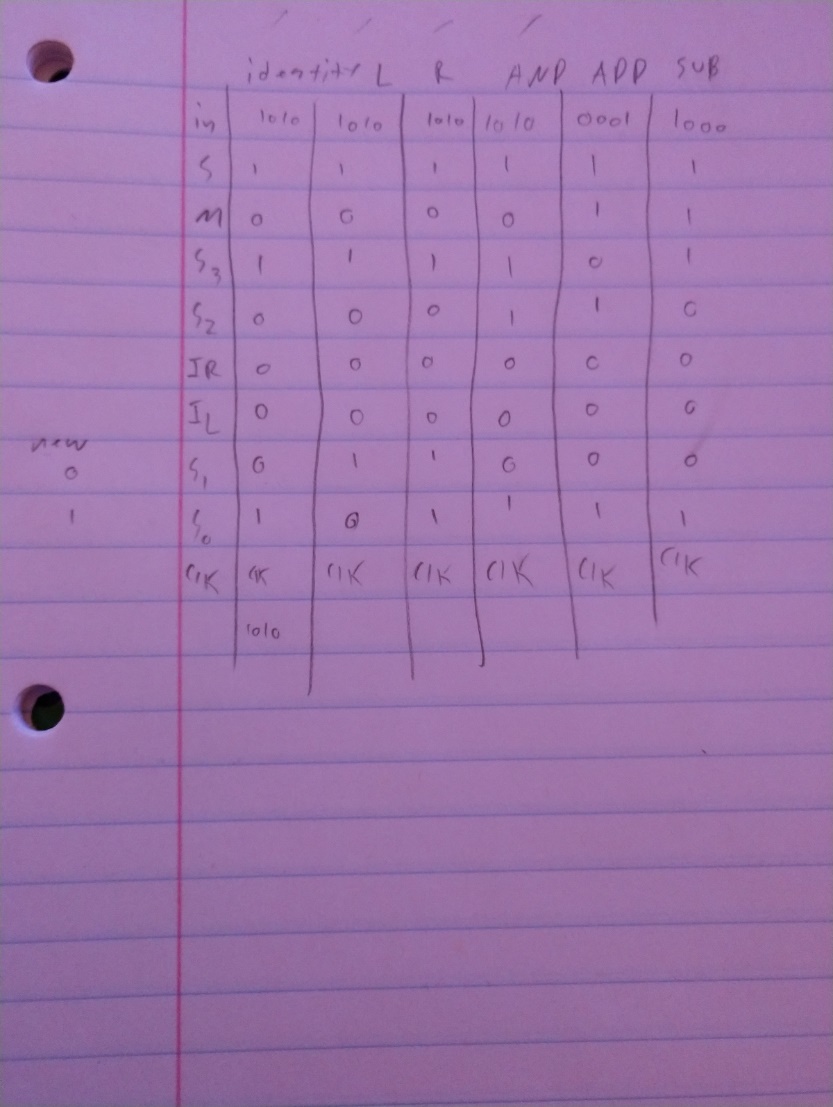
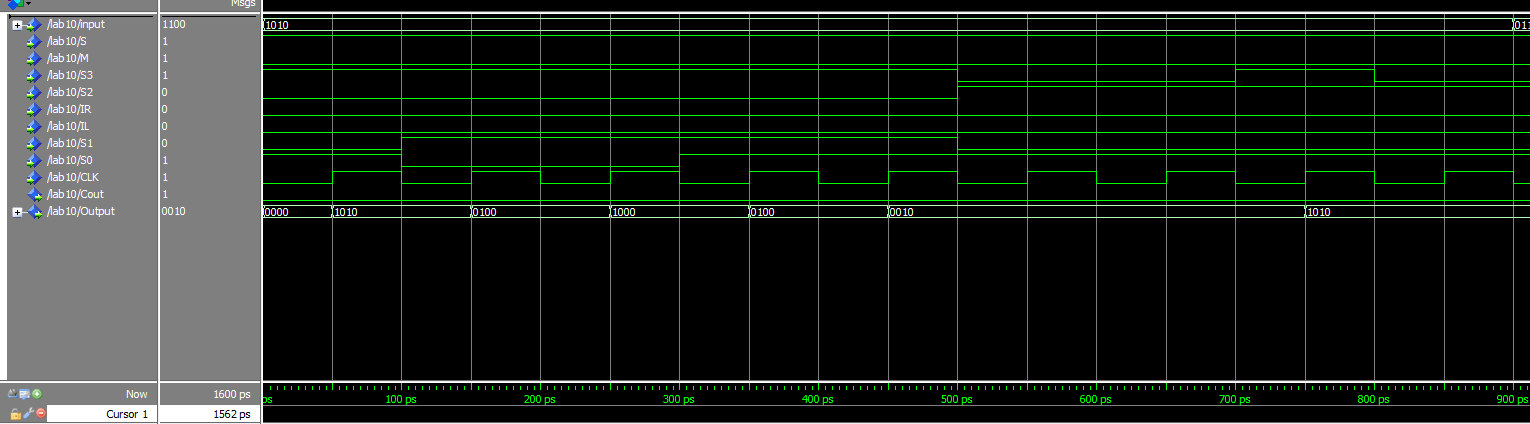
**Design Analysis**

* There were 8 total sub components that made up the registered ALU. Out of those 8, there were 3 main components, the selector, ALU and accumulator.
* The Selector was created using 4 2:1 MUXs. The 2 inputs for this MUX were a ground at 0 and a controlled input (I) at 1. The output of the MUX was determined by a controlled select line. When the select is 0 the value of ground is passed through, and while the select is 1 the value of I is passed to the ALU as A.
* The ALU is made up of an Arithmetic Extender (AE), Logic Extender (LE), and Full Adder (FA).
* The AE’s function is to perform one of 4 arithmetic operations in the form of (M,S3,S2) where M decides whether the AE or LE is used and S2-3 are select lines governing the operation used. Decrement(1,0,0) passes the value of A -1, Add(1,0,1) passes the value A + B, Subtract(1,1,0) passes the value of A + B’+1 (B’ is B’s complement i.e. inverting all of its bits) and Increment(1,1,1) passes the value of A+1.
* The LE is similar to the AE and perforems Compliment(0,0,0) which inverts the Bits of A, AND(0,0,1) which only passes a value if both A and B are a 1, Identity(0,1,0) which returns the value given, and OR(0,1,1) which passes B or A as it’s final value.
* The AE and LE feed into a full adder which is detailed in a prior experiment. The FA simply adds it’s A, B and Cin components and passes a 2,1 or 0.
* The ALU is completed by adding a set of 4 AE’s paired with 4 LE’s that together feed into 4 FA’s. M and S3 are also connected to an and gate in order to use the Cin function of the full adders for the Subtract (A+B’ +1) and increment (A+1) functions. S2, S3, and M are new controlled inputs while A is the value from the selector and B is the value from the Accumulator. Cout is an extra output of the whole registered ALU that only occurs when during the increment and add functions
* The Accumulator is made of 4 4:1 MUX’s that are then connected to 4 D flip flops. There are also 5 more inputs. S1, S0, CLK, IL(the value added to the left most bit if shifted right) and IR(the value added to the right most bit if shifted left)
* The 4:1 MUX takes in 4 values and two select lines, it simply passes one of the values through depending on the combination of the select lines.
* The outputs of the flip flops have the pattern of being the inputs for the 3rd input for the MUX preceding it, the 0th input of the corresponding MUX and the 2nd input for the MUX after it. In the cases where there is no prior or next MUX the flip flop only goes to two inputs. In the place of these missing inputs IR and IL act as the inputs to the least significant bit and most significant bit respectively. The 1st MUX input is always used by the corresponding value it has been passed by the ALU.
* The output for the Accumulator is the final output as well as being the value B that is passed back into the ALU.

**Figures and Attachments**

The list of Inputs required to test different functions

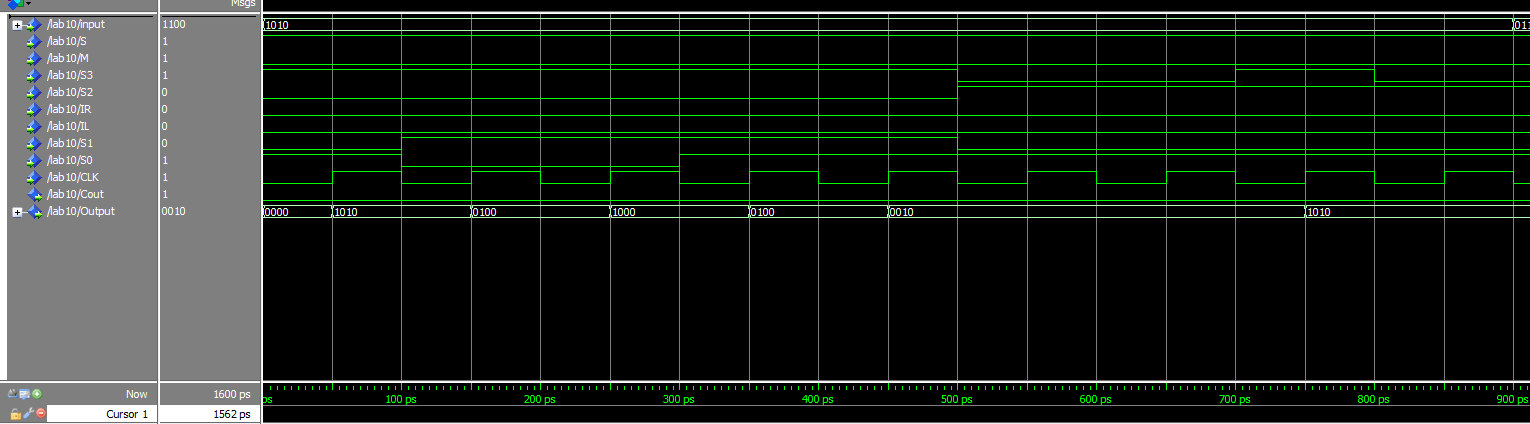




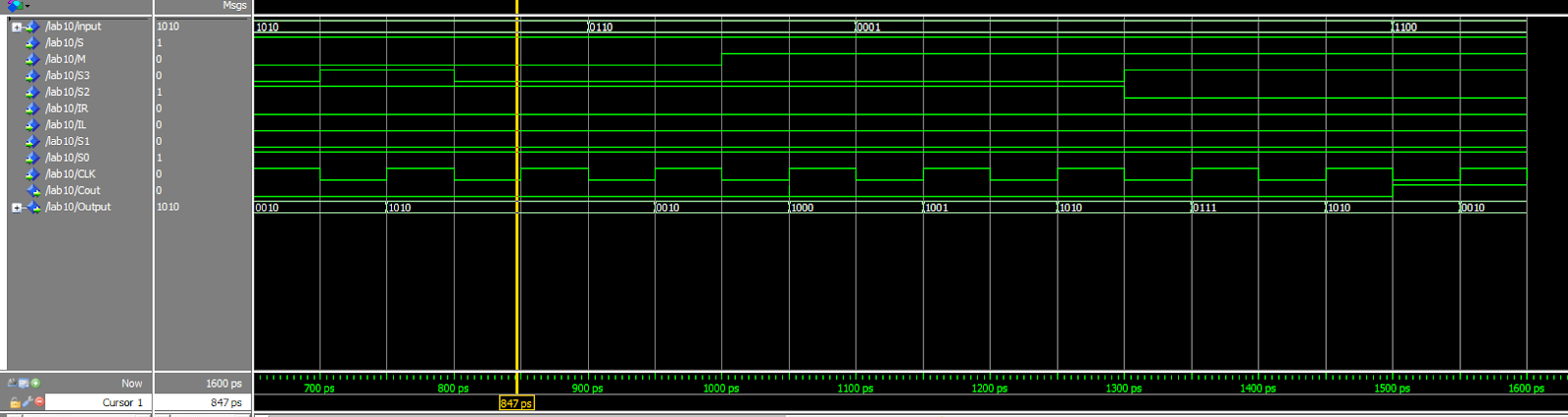
At 50ps the Identity function was activated

At 150ps the shift left function was activated

At 350 the shift right function was activated

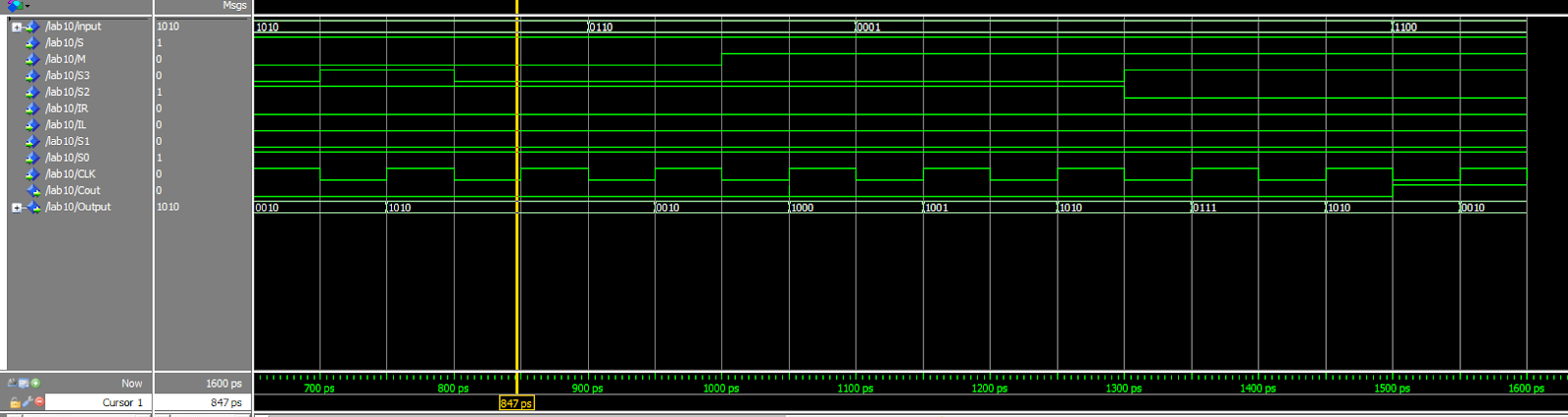


At 550 the AND function was activated (inconclusive 1st try)

~~\*~~

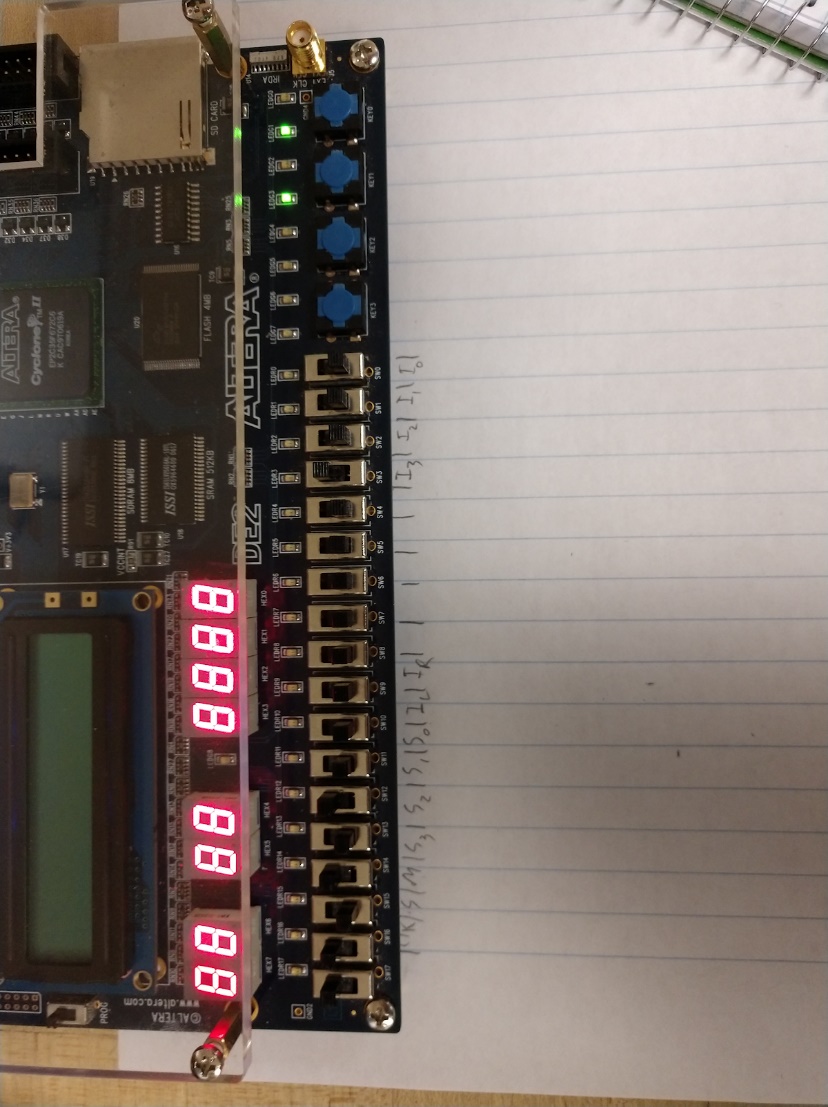
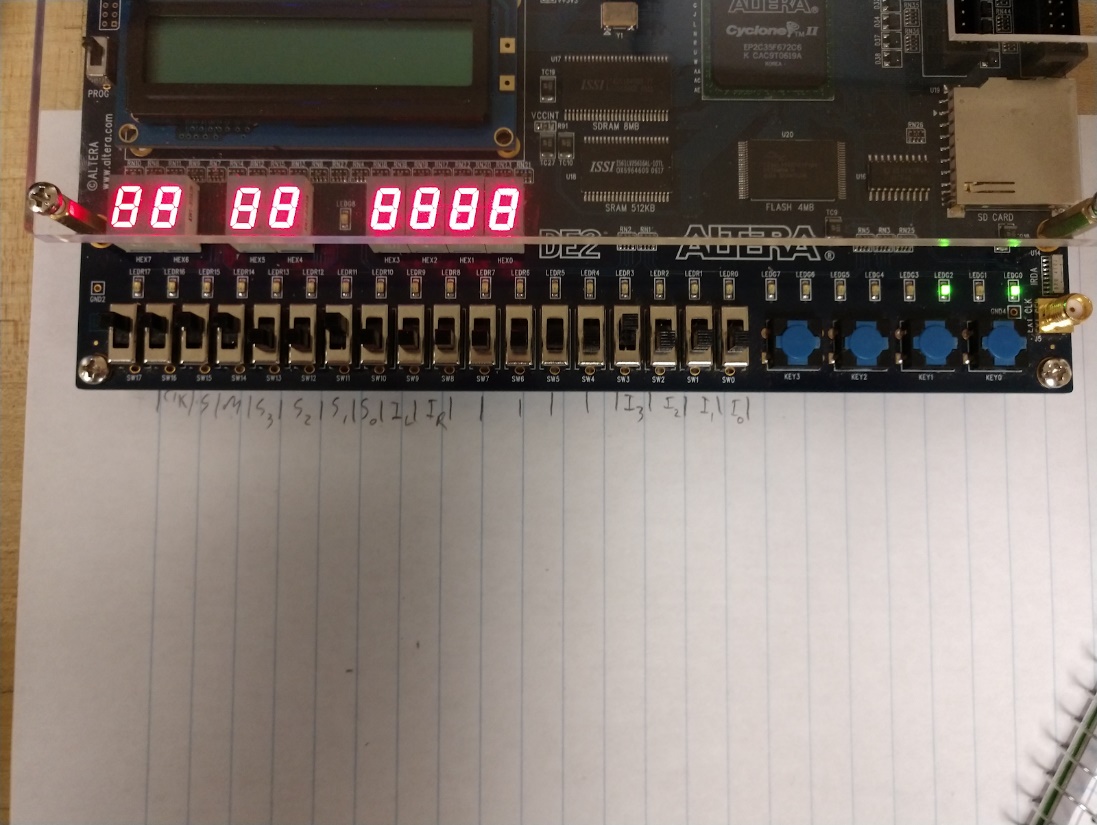
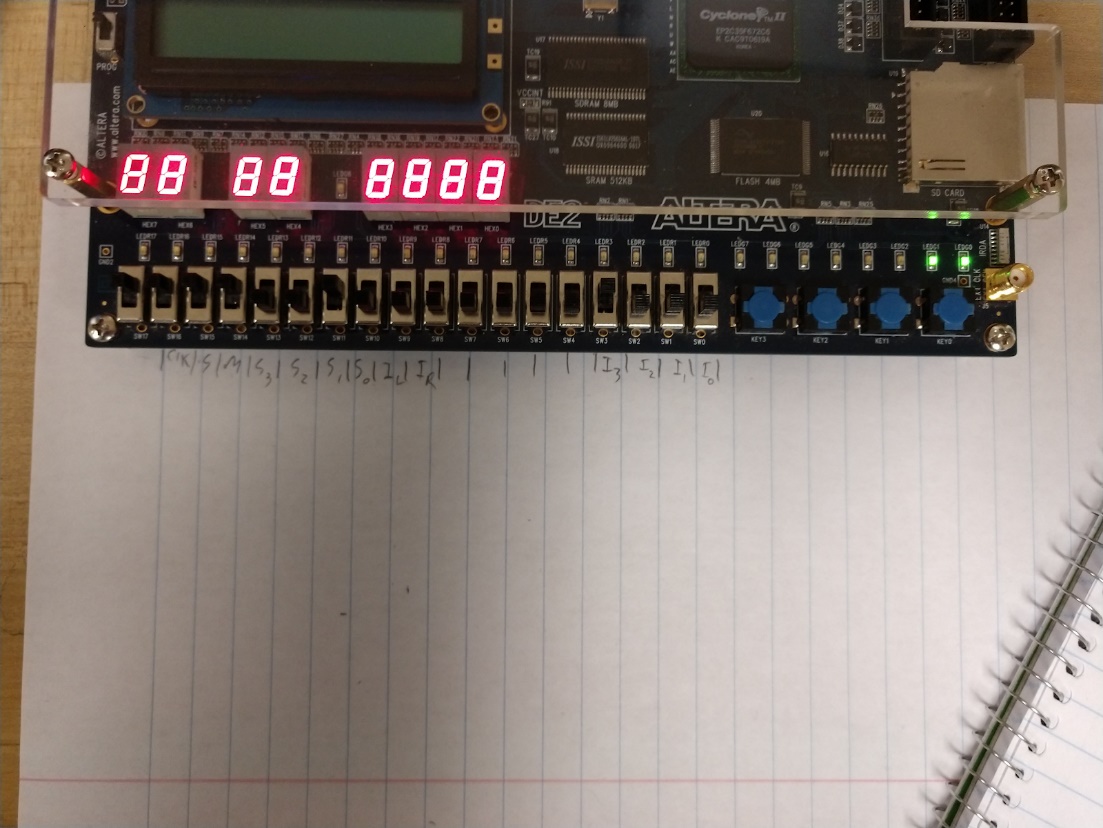
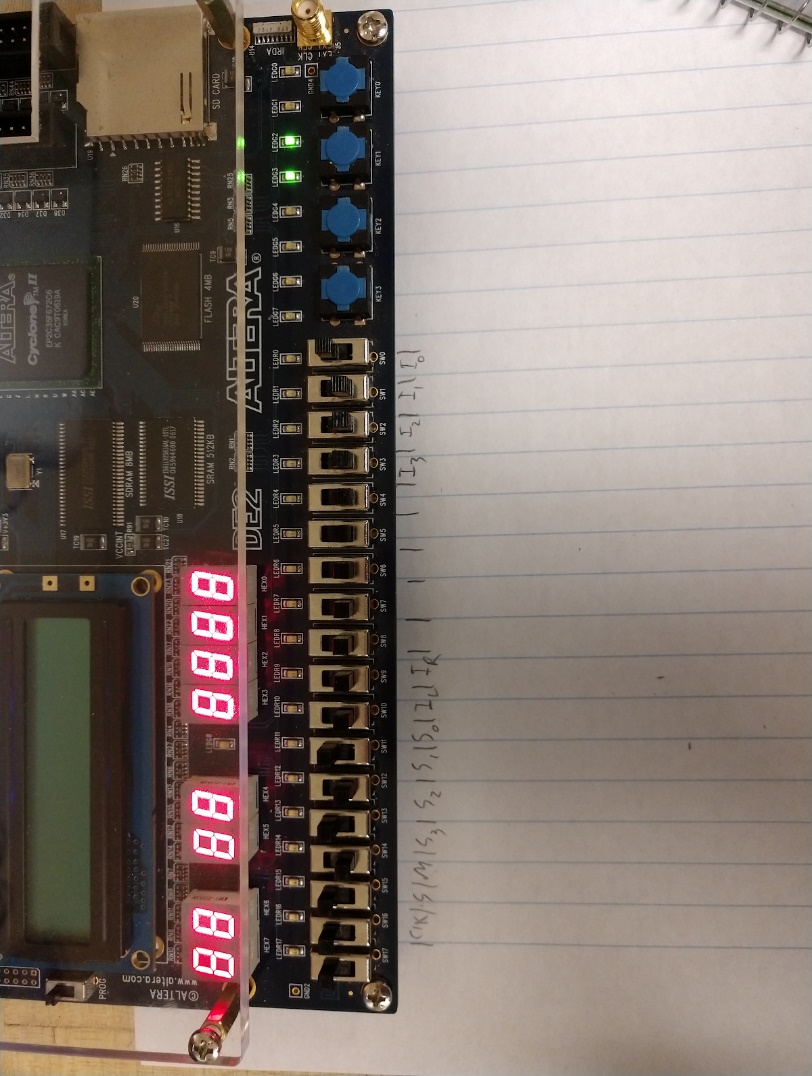
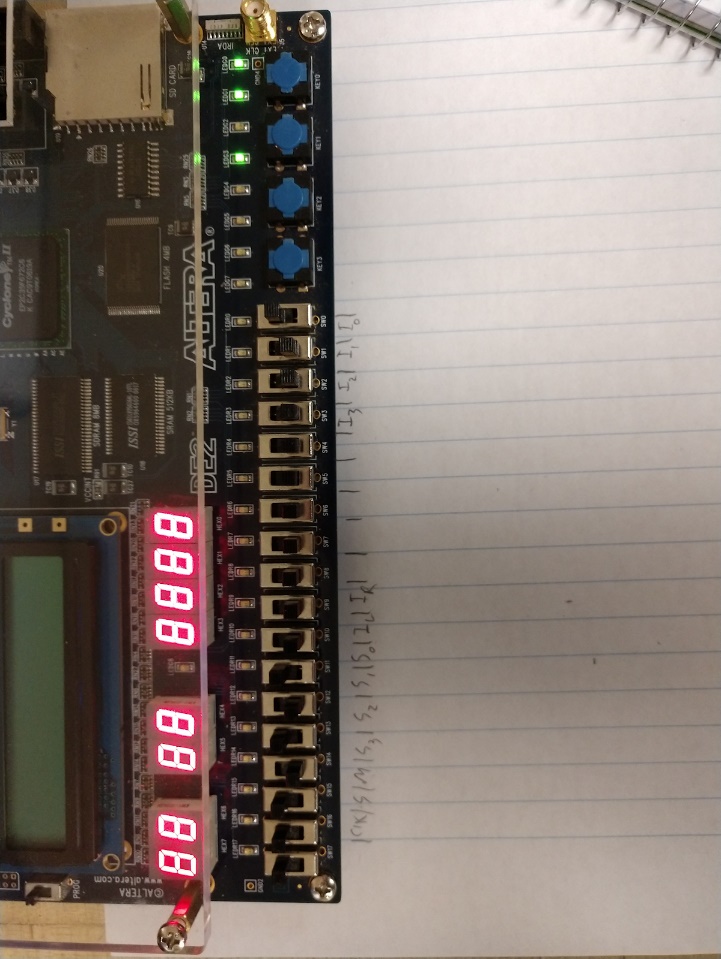
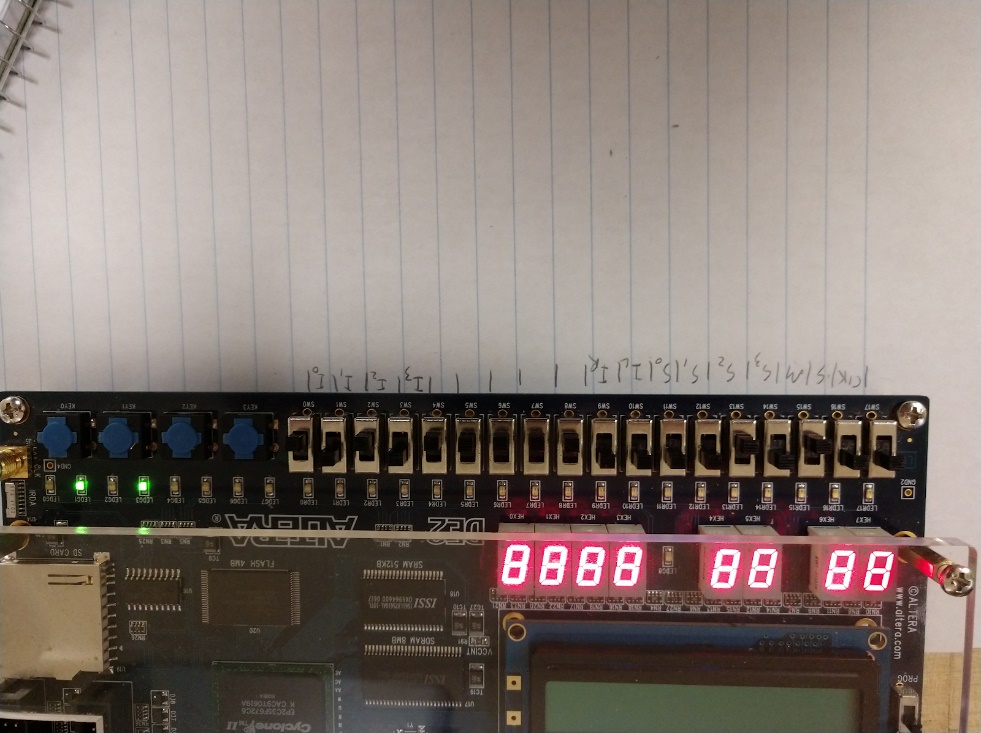
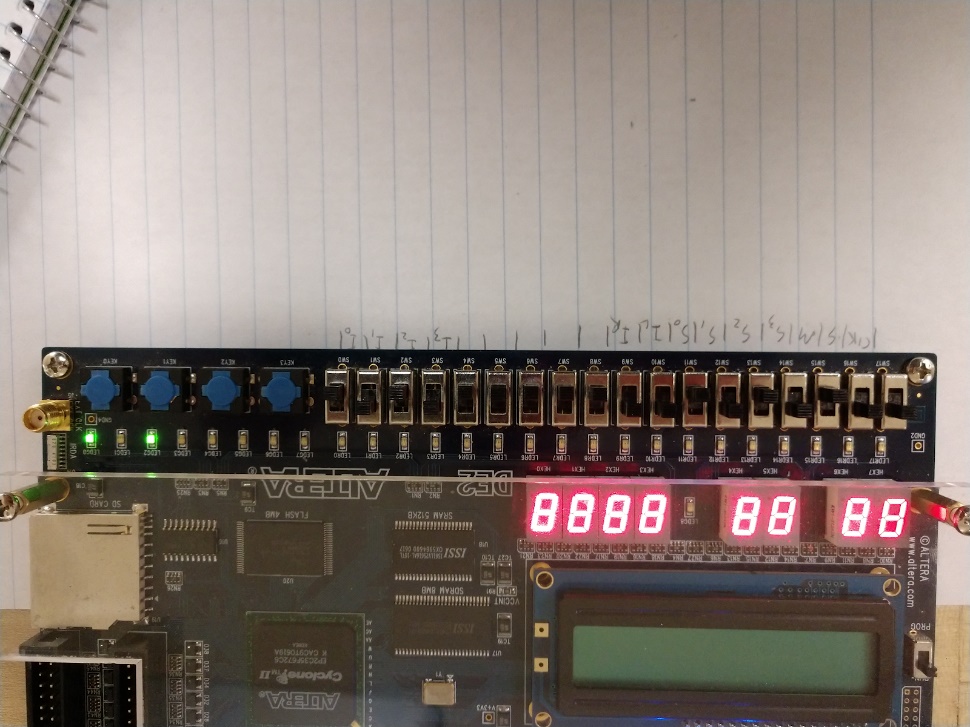
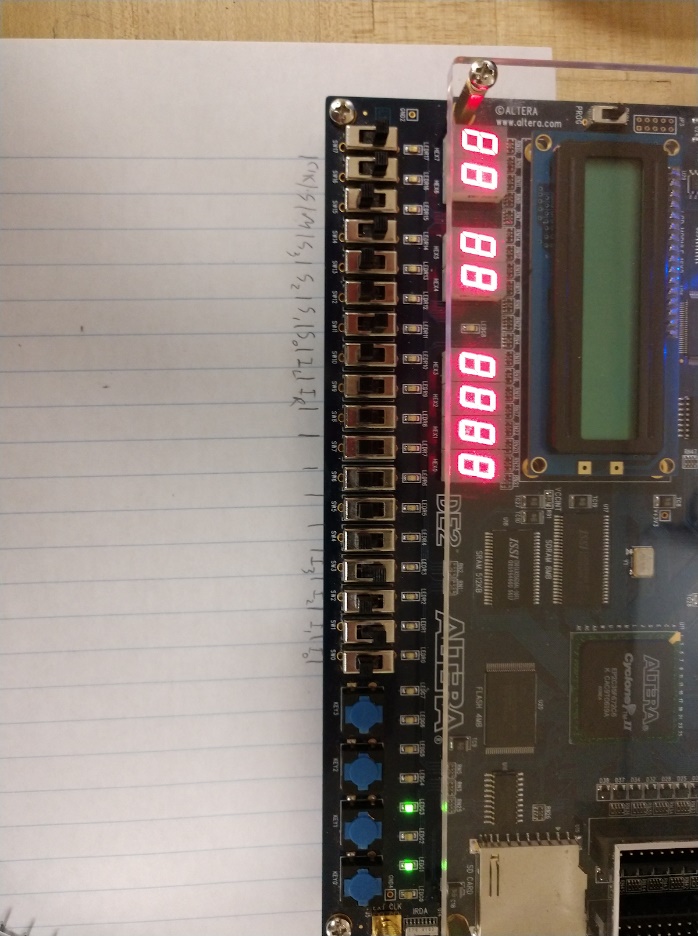
At 750 the OR function was activated

At 850 the AND function was reactivated(good)

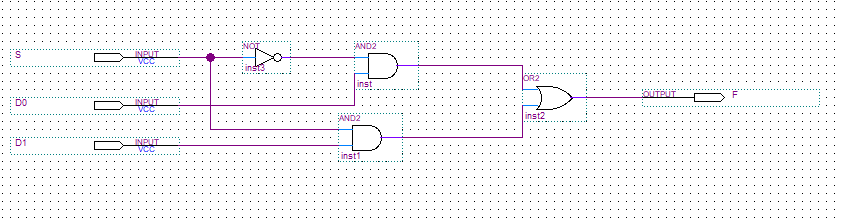


At 1050 subtract was activated

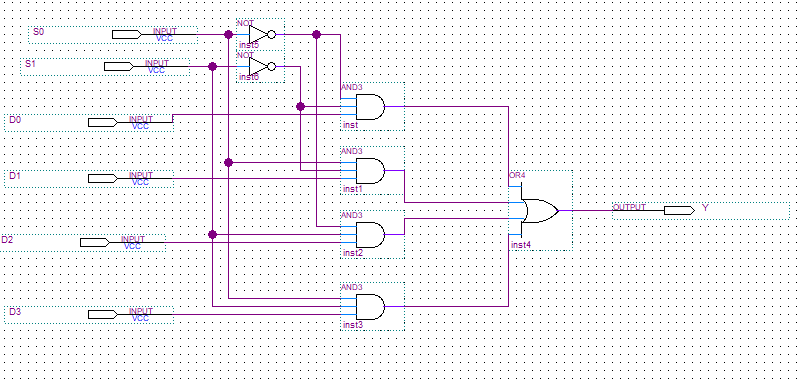
At 1350 ADD was activated



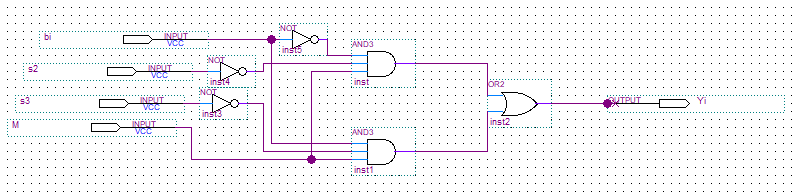
2:1 mux



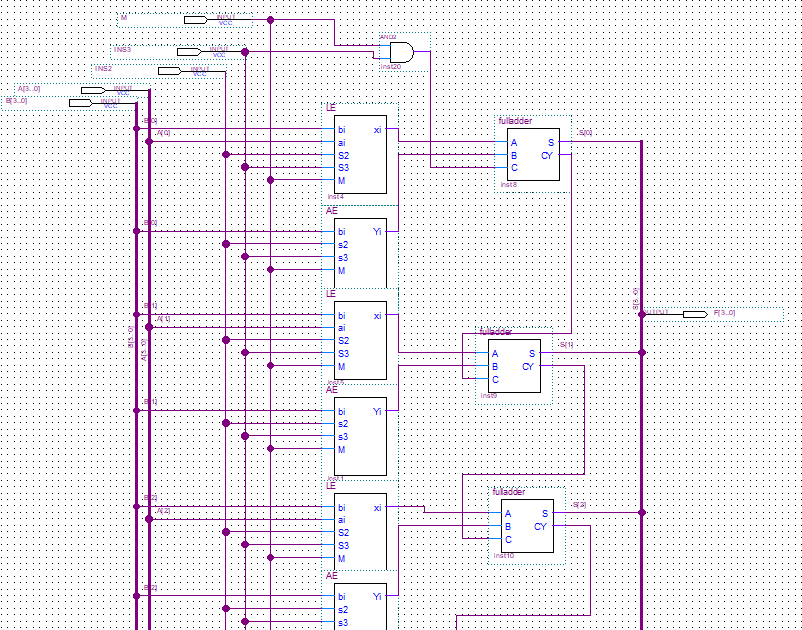
4:1 mux

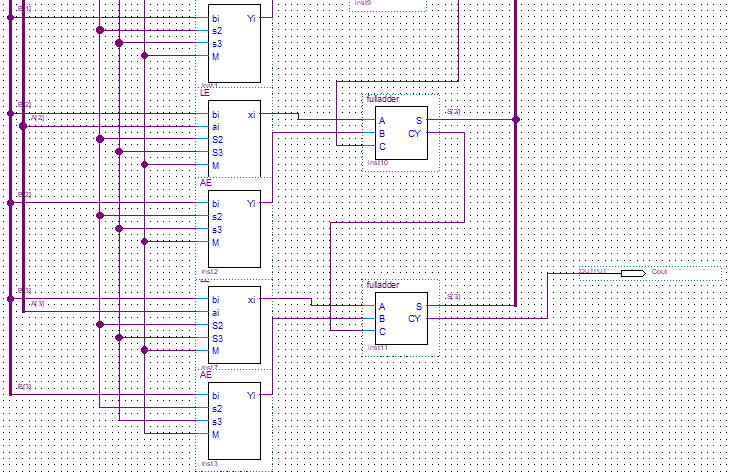


AE

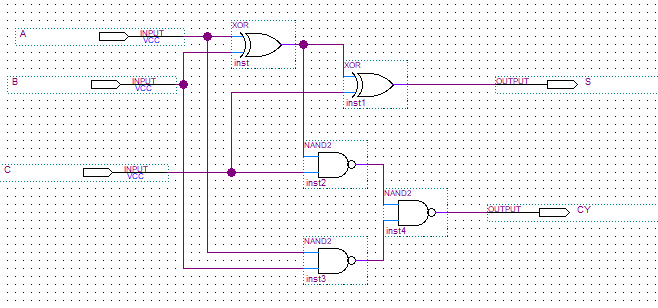


ALU 1

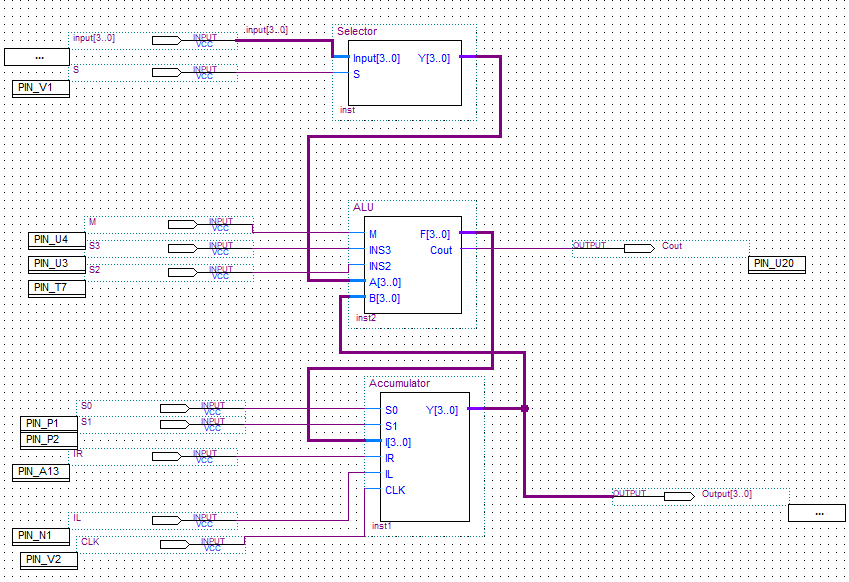




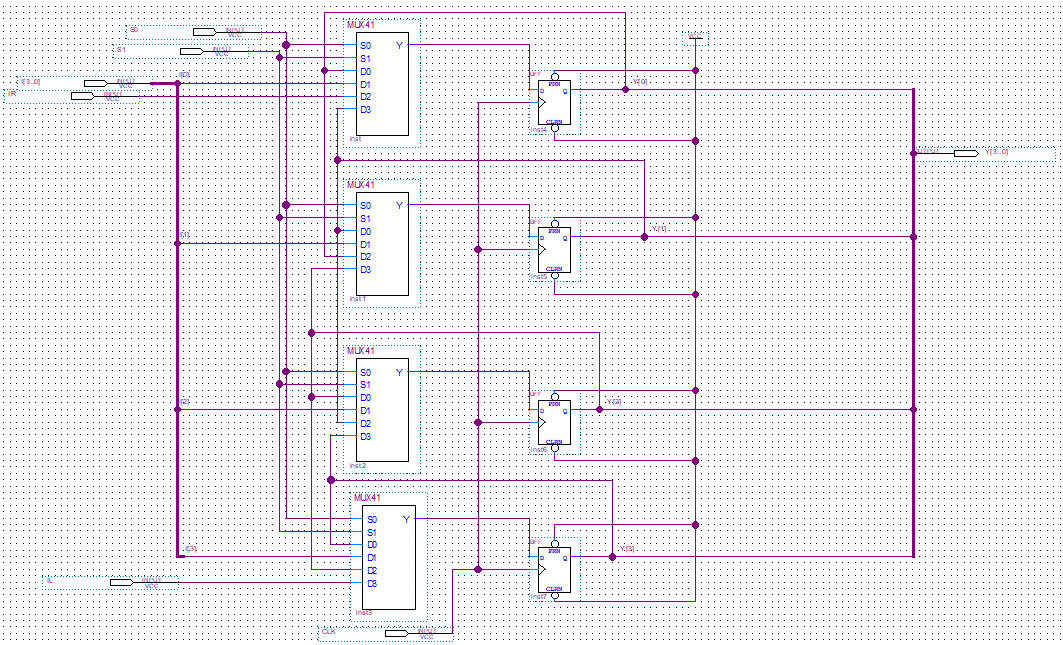
Full adder



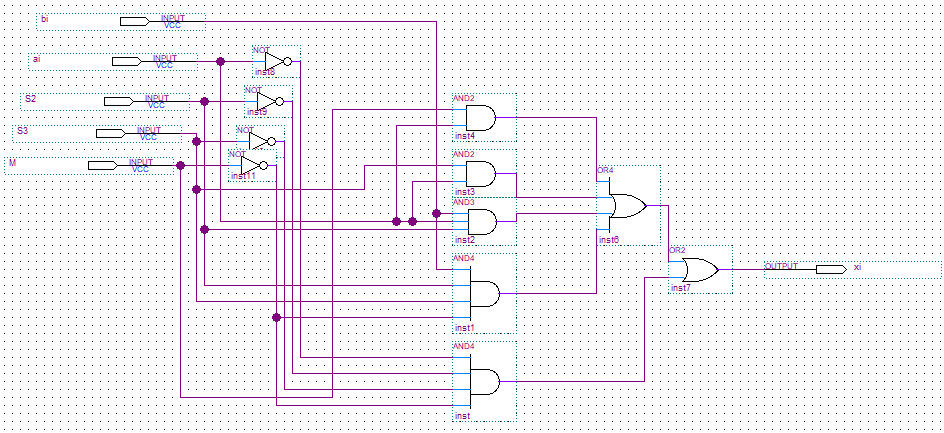
Lab 10



accumulator



LE



Selector

