There are 8 different operations that are done by the ALU. Since the adder block can be used as a subtractor, and different inversions of the original r and s inputs, when passed through nand, nor, and xor gates, can make simpler gates, it is possible to only use 4 lines connecting to a 4 to 1 mux. The mux inputs are reserved for 4 types of operations. Sum, nor, nand, and xor. The control unit determines which of these outputs are accepted. I added the choice to invert the r input in the logic block from a control input, just like the s, because it allows for R to be subtracted from S and a number of other operations.

Operation 000 is S+R:
S+R is just the xoring of the two inputs. Because of the inverters that were given in front of the inputs, I double inverted the inputs so that they would be the original signals for the sum block to add together. Goes to sum mux input

Operation 001 is S-R:
S-R is logically equivalent to S+R’+1. To do this, the R is controlled to not be inverted, and the S is. Remember, the inputs have a requisite inverter to pass through, so they need to be inverted again to be the original signal. Since there is a sum block already made, it is used to add whatever the Cin is to the output. If the Cin is low, then the arithmetic output is just S-R-1. Goes to sum mux input

Operation 010 is R-S:
R-S follows them same process as S-R, except the R is chosen to be inverted, and the S isn’t. Goes to sum mux input

Operation 011 is R or S:
R or S is logically equivalent to R’ nand S’, and since we already have a not gate, neither R nor S is chosen to be inverted, and the nand between the two of them is the or of the two. Goes to nand mux input.

Operation 100 is R and S: R and S is logically equivalent to R’ nor S’, which is also provided in the logic block. R invert and S invert are not high so they can be inverted by the not gates in the logic unit. Goes to nor mux input.

Operation 101 is R’ and S: This is logically equivalent to R nor S’. As stated, we can choose the inversion of the inputs via the control block, and a nor is provided in the logic block. Goes to nor mux input.

Operation 110 is R xor S: The positive values of R and S are brought through the xor logic gate and they go to the xor mux input.

Operation 111 is R xnor S: This is logically equivalent to a single one of the inputs being inverted when it’s xored with the other input. I went with allowing S to be inverted before passing through the xor gate and going to the xor port on the mux.