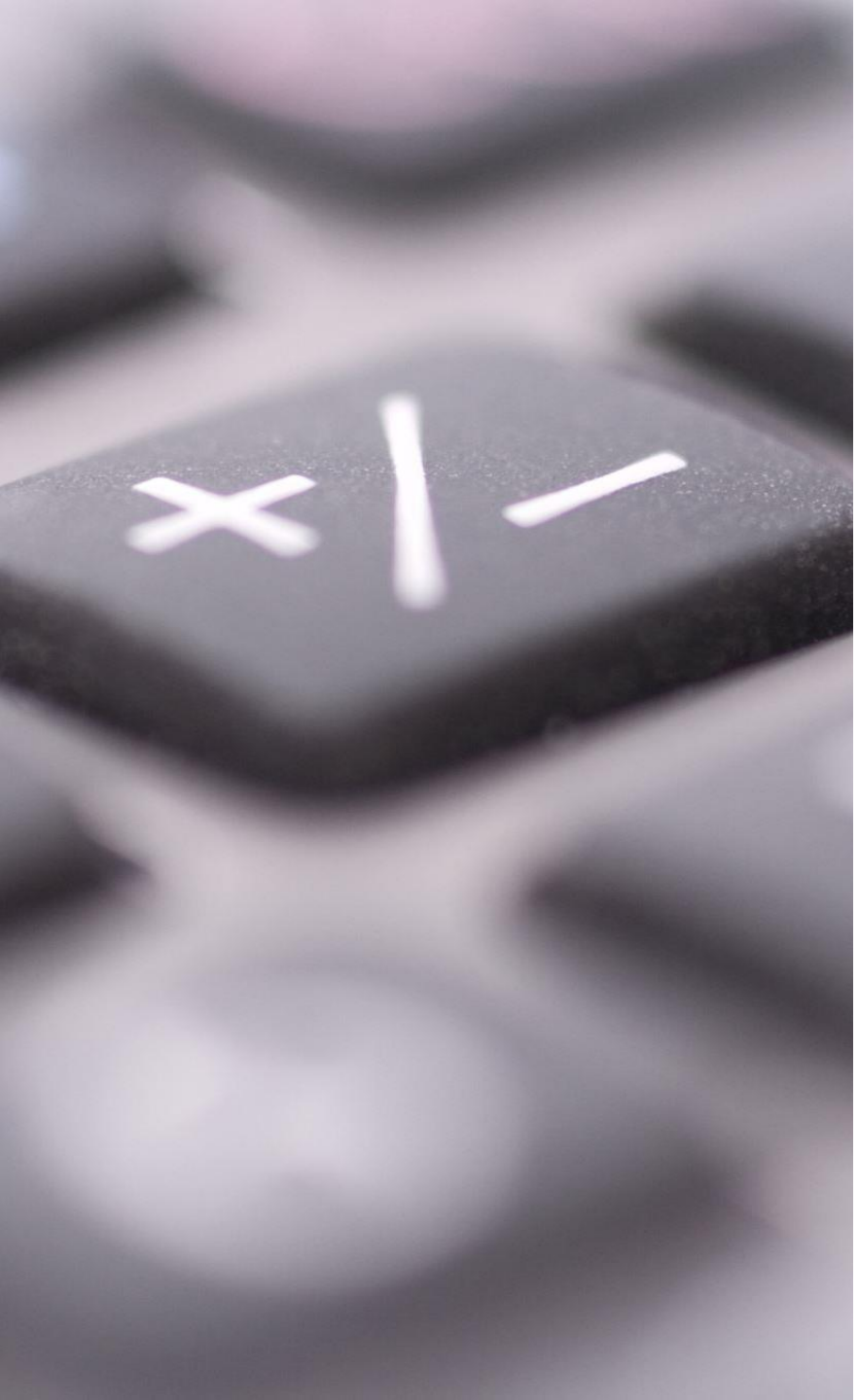
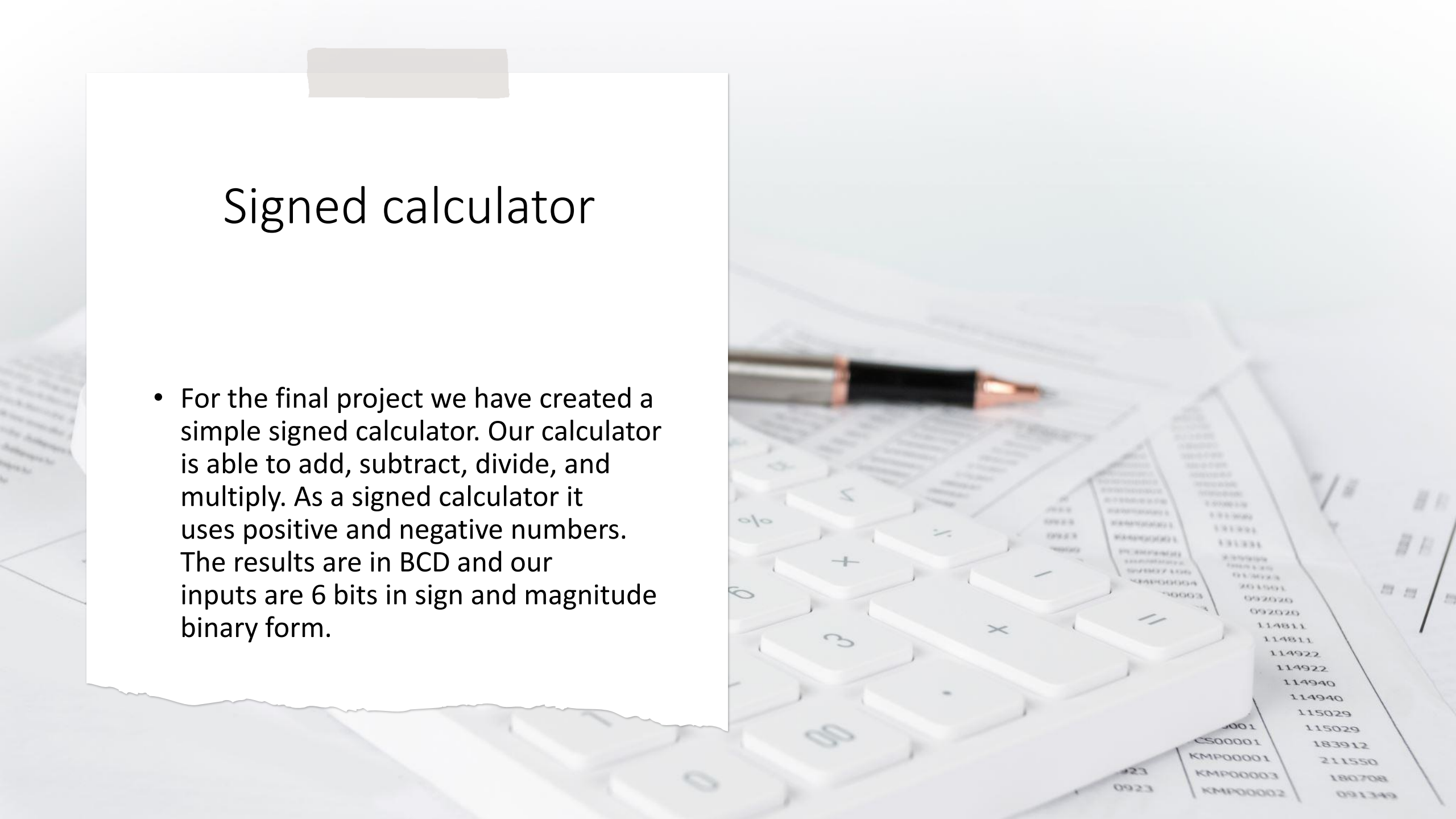


Signed calculator

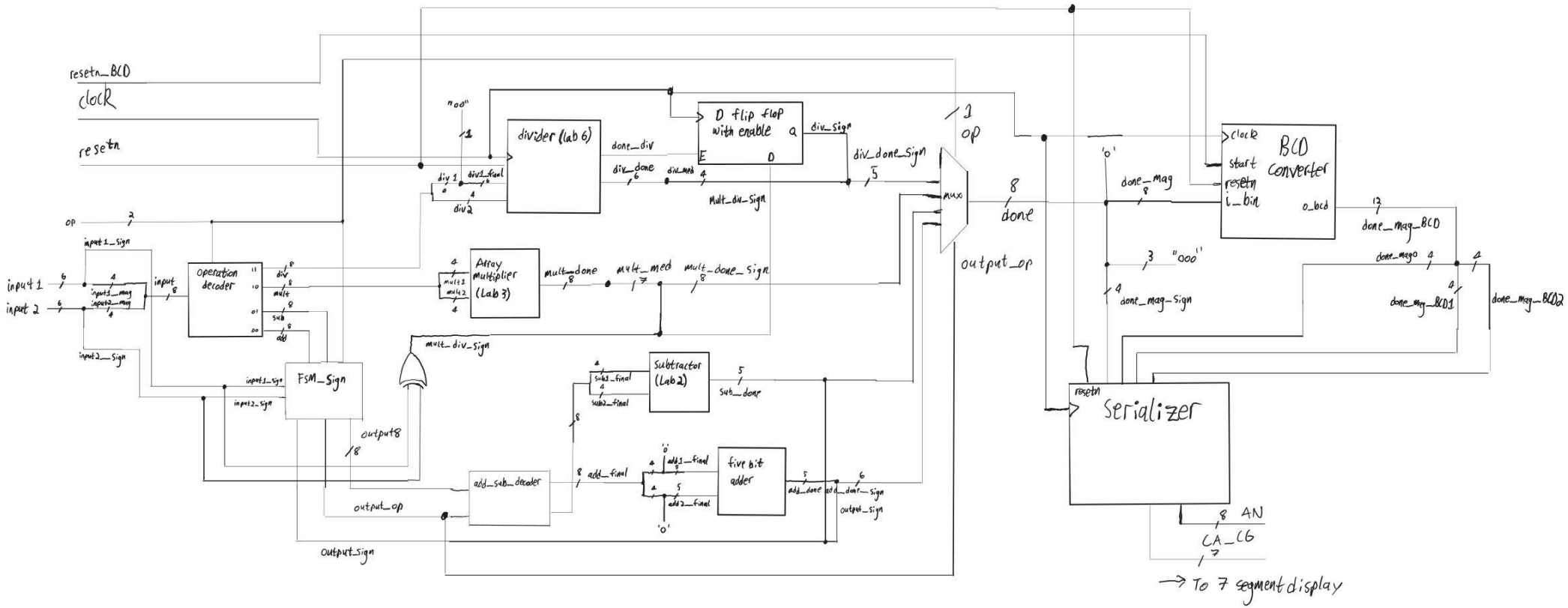
Fady Sto, Karam Matti, Jason Wend, Manuel Muresan

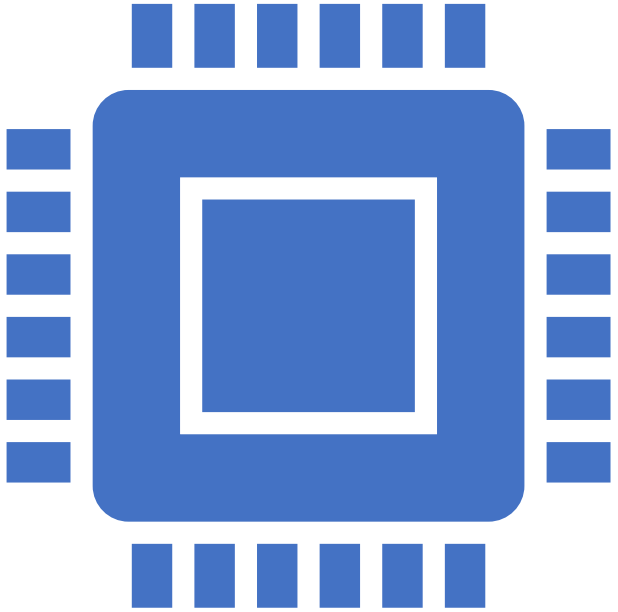
Signed calculator

- For the final project we have created a simple signed calculator. Our calculator is able to add, subtract, divide, and multiply. As a signed calculator it uses positive and negative numbers. The results are in BCD and our inputs are 6 bits in sign and magnitude binary form.



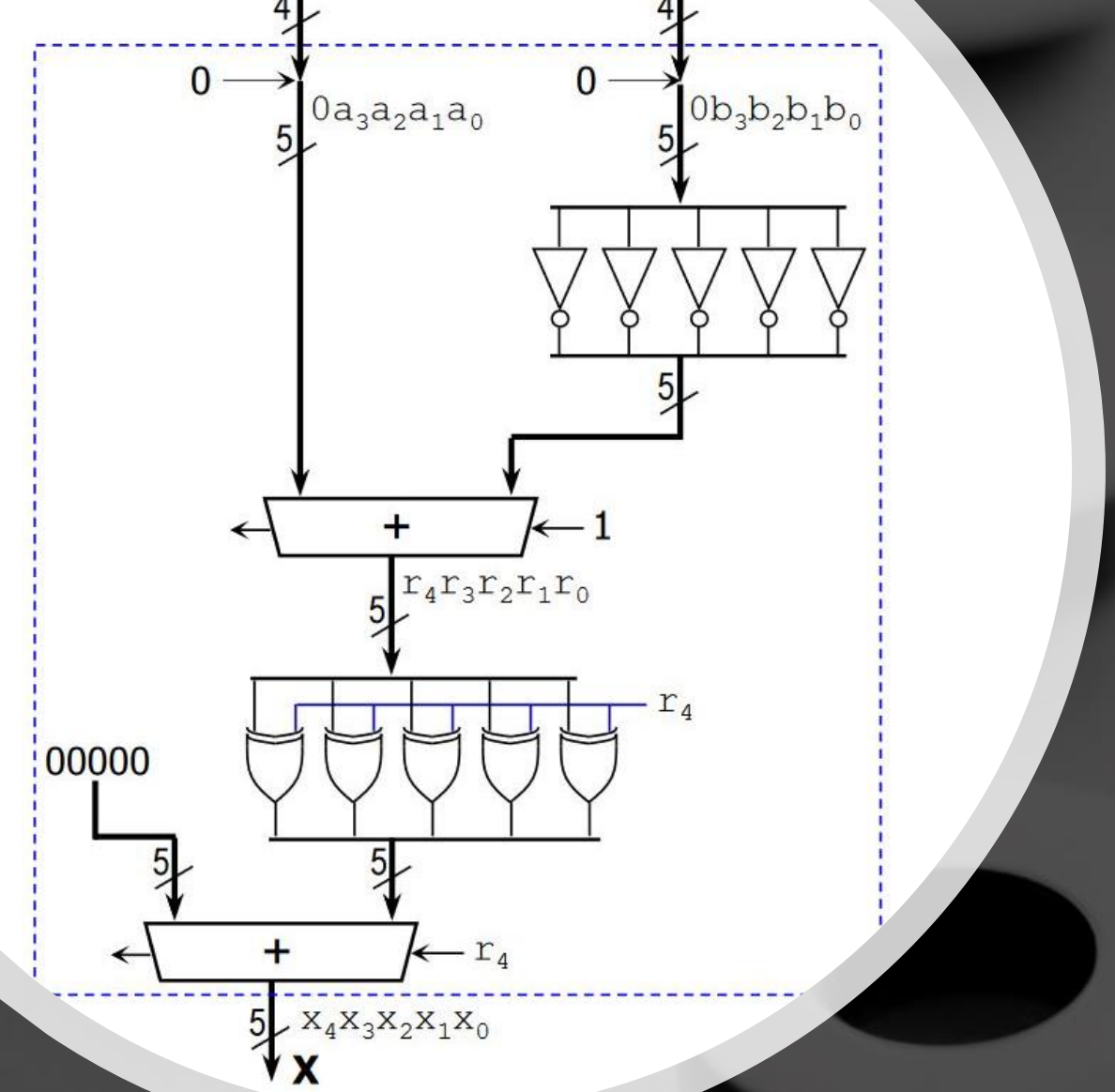
The design:


resetn_BCD
clock
resetn
op
input 1
input 2
input1_sign
input2_sign
input1_mag
input2_mag
Operation decoder
FSM_Sign
"00"
divider (lab 6)
div1
div2
div1_final
done_div
div_done
div_sign
D flip flop with enable
div_done_Sign
mult_div_Sign
Array multiplier (Lab 3)
mult1
mult2
mult_done
mult_med
mult_done_Sign
mult_div_sign
Subtractor (Lab 2)
sub1_final
sub2_final
sub_done
add_sub_decoder
five bit adder
add1_final
add2_final
add_done
add_done_sign
output_sign
output8
output_op
Output_Sign
mux
1
op
8
done
output_op
'0'
done_mag
3
"000"
4
done_mag_Sign
BCD converter
clock
start
resetn
i_bin
o_bcd
done_mag_BCD
done_mag0
done_mag_BCD1
done_mag_BCD2
resetn
Serializer
8
AN
CA_CG
7
To 7 segment display



Adder

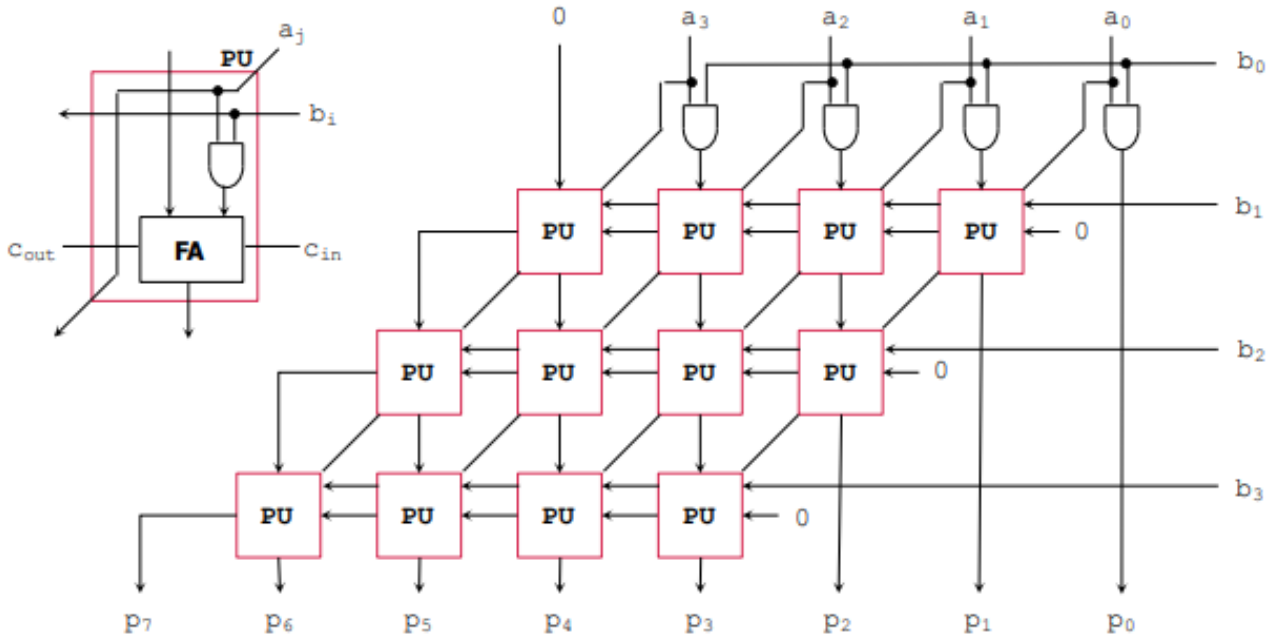
- One of the five bit adders from the subtractor (see subtractor a=on next slide)
- These five bit adders contain 5 full adders



Subtractor

- Uses two five bit adders and various other logic to compute the absolute value of input 1 minus input 2.
- Subtractor: There are two inputs that are 4 bits and one output that is 5 bits

Multiplier:

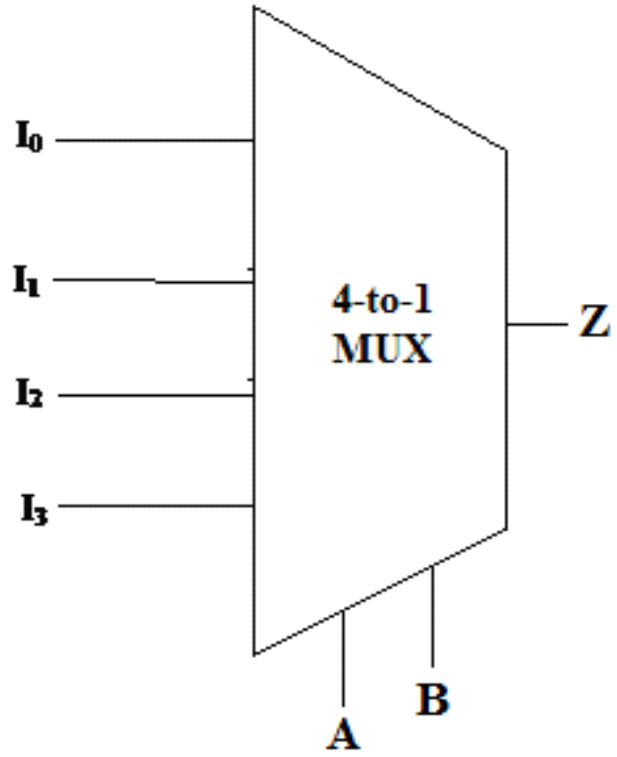


- A component that takes advantage of full adders.
- uses two 4-bit inputs and saves them as an 8-bit output (Lab 3)

Division:

- Unlike our previous asynchronous operations, the division operation was controlled by a clock since it used various shift registers, a counter, and an FSM (See Lab 6)
- Two of our inputs are vectors and three of our outputs are vectors.

Basic 4-to-1 MUX:

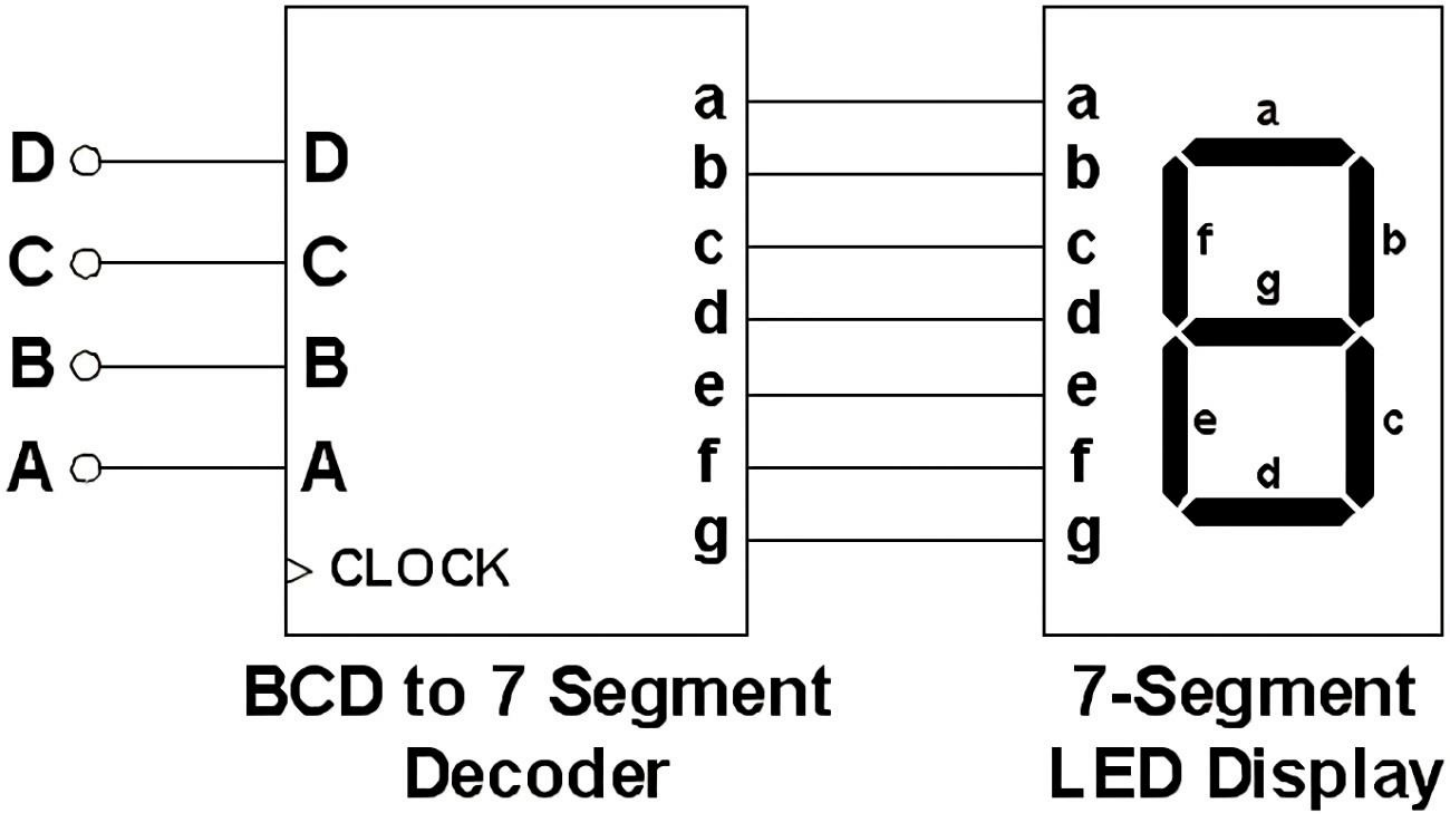


- main component of the output function of the calculator

- Receives the answer of all four operations and outputs only the desired answer.

7-Segment Decoder:

- Receives answer as a 16-bit Binary number.
- Converts to a 16-bit signed BCD and then segmented into four 4-bit numbers.
- Each display is turned on by a counter and state machine, which also updates the information once every millisecond.



Problems Faced:

● Organizing Files: When all the files were combined, they needed to be organized into groups of similar files that each member had created.

● Using several 7-segment displays at once

● Code that operates properly when used separately but not when combined

- Division being a synchronous process
- Did not have enough time to use a keyboard instead of the switches from the FPGA board

Thank you

Any Questions?