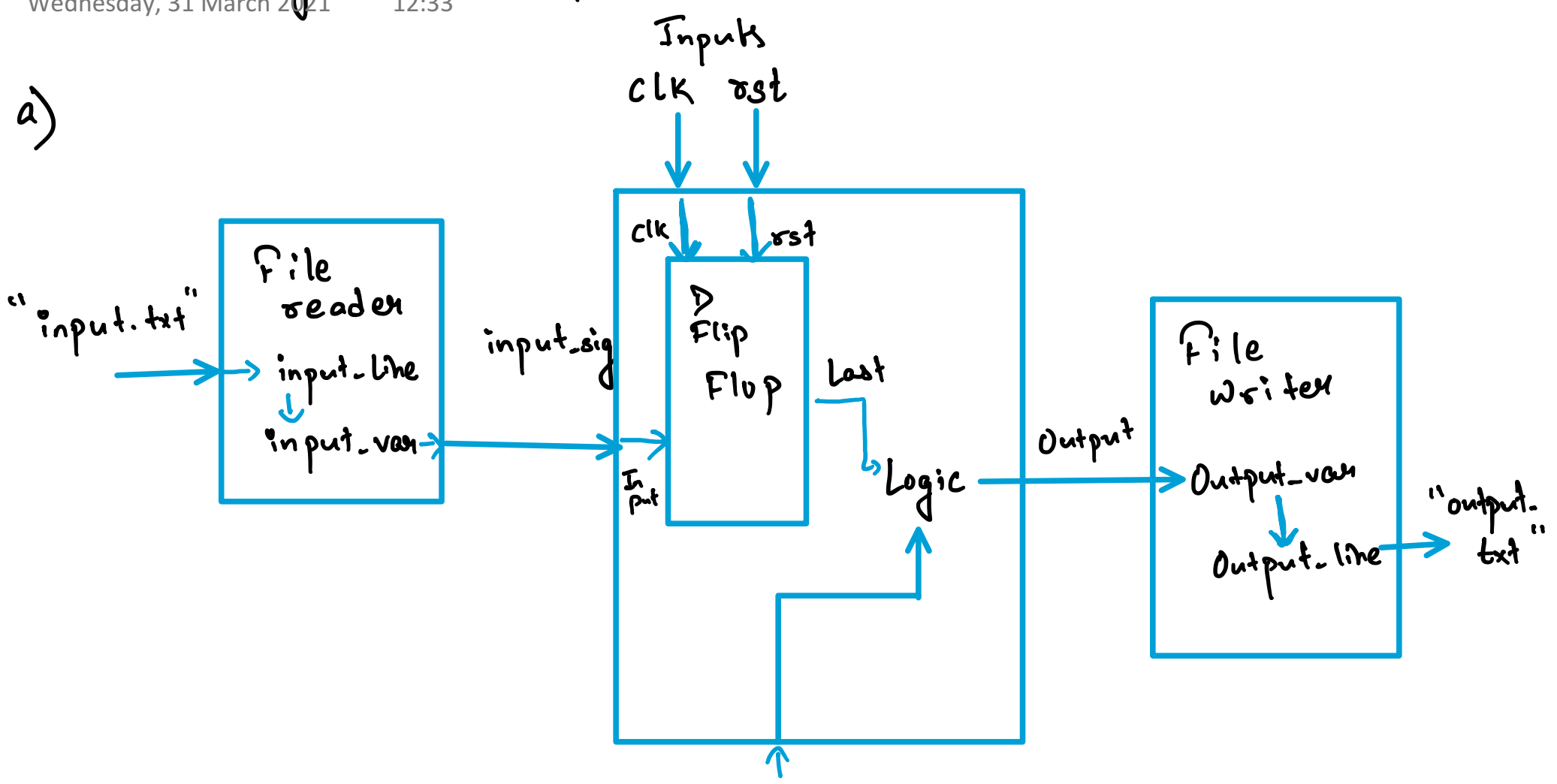
Block Diagram and Explanation

Wednesday, 31 March 2021 12:33

3 a)



Variables:
buff : Buffer array (0 to 63)
r = read pointer
w = write pointer
count = counter

b) There are 2 modules in use ie. RLE and DFlipFlop. ASCII_Read-test is the testbench that reads the input signals from input.txt and writes the output in output.txt. It uses RLE as a component and each input line is fed into RLE per rising clock edge.

RLE is the main entity in use, with inputs input, clk, rst and and outputs output and valid (Data valid line). Input is fed into the DFlipFlop to help sync with the clock and we use last as the output of DFlipFlop.

We define certain variables and constants. Buffer array of 64 bytes to store the output lines as we process the data, ESC is stored as a constant, and read and write pointers, r and w to denote the current location till which we have read/wrote. On each rising clock edge, if there is repetition of input and count is less than 15, then we increment count and move to next clockcycle. Else, we write into the buffer array the desired output. If there were continuous "ESC" then we write ESC, count, ESC as three terms in buffer array, else we take cases on count being 1, 2 or more than 2 and write the corresponding outputs into buffer array. Then we set count to 1 for the new character. For each clock cycle, we also need to output, if there is an output ready. Hence, we output a byte each time the r pointer is behind w pointer, and keep the valid bit 1, else output garbage value and valid bit 0. This output is processed by the testbench and written into output.txt.