CS 254 - Assignment 7

Rules:

1) ***Any style of code is permitted.***

2) Use the ieee.std_logic_1164 library data types. (Bit and bit vector are not permitted.)

Description of design:

We want to design a data compression circuit using run-length encoding. It replaces continuously repeated occurrences of a byte by a repeat count and the byte value. The circuit receives a fresh byte at every positive transition of an externally supplied clock. We shall use the 'ESC' character (code = 1BH) to signal the use of a repeat count. Therefore, if the 'ESC' character itself appears in the input stream, it has to be handled in a special way. The output is one byte wide and every data byte being output is signalled by a rising transition on a DataValid line.

- If any character 'c' repeats 'n' times in the input stream such that 2<n<16 then we output the three-byte sequence "ESC n c".
- If 'n' number of 'ESC' characters arrive contiguously in the input stream, we output the 3-byte sequence "ESC n ESC", where n can be from 0<n<16. Otherwise, we just output the received characters without any change.
- Notice that in both of the above cases since the output of your circuit is only 1 byte wide, it cannot output all the 3 bytes at once. Only 1 byte is output per clock cycle and whenever a byte is being output, the Data Valid output line must go from low to high.
- If the repeat count is more than 15, we handle the first 15 characters as above and treat the 16th occurrence onwards as if a new character has been received.
- This circuit reads from an input file and sends the output to an output file. The reading and writing operations are done in the test bench itself which will be shared with you. Modify this test bench according to your need.
- For the purpose of testing make your own input file with the following constraints.
    - The data should be present in binary ASCII format (Make use of the ASCII table that will be shared with you).
    - Only small letter alphabets can be used. Beside 'ESC' and 'SPACE', no other

special character is permitted.

    - There should be only one character per line in the file. (That is every line will have only 8 binary bits).
    - The file should not have more than 32 bytes. (i.e. not more than 32 characters).

- The testbench will read the output from your circuit and print it to an output file. It will print only one character (8 bits) per line.

Need for a buffer: Fresh data is input at every rising edge of the clock. Depending on whether or not there is repetition in the input there may or may not be a fresh valid byte on the output lines. Since the output data rate may be less than or equal to or greater than the input data rate over short periods, provision must be made for buffering the input data and provide a data valid output line which becomes 1 in a clock cycle in which valid data is being output. Decide on a safe buffer size based on the constraints of the input file size mentioned above (Another approach is that you can choose whatever buffer size you want but then there should be a mechanism in your circuit for handling the situation when the buffer gets full).

Things required in Submission:

1. All VHDL files of top-level and sub-components. (.vhd or .vhdl files)
2. Screenshot of Waveform.
3. A document should be submitted which should contain:
    a. Block diagram of your entire design including your RLE encoder, file reader and file writer.
    b. A detailed explanation of the design of each and every module.
    c. RTL netlist diagram obtained after successful compilation of your code.

Submission rules:

1. The VHDL files of each question should be kept in a **separate folder**.
2. All the VHDL files of one question (both Top-level & sub-components) should be kept in the **same folder**.
3. The final zip file that is submitted on moodle should be named in the following format: **group_<groupnumber>.zip**. For example, the zip file for group 8 should be named group_8.zip