LMS Hardware Engineer Assignment

Message Extractor

1. Draw a diagram of your chosen design (feel free to use <https://asciiflow.com>)

Please find the diagram in <msg\_parser\doc\architecture.txt>.

1. Write an elegant, synthesizable solution for the message extractor in RTL/Verilog/SystemVerilog using the skeleton provided. And verify it against the given sample inputs.

Please find the solution developed in <msg\_parser\src> and <msg\_parser\sim>. If you have modelsim, you can launch the simulation using msg\_parser\msim\_compil.bat>.

1. What is the bottleneck of your design/code? (what can limit the maximum

frequency?)

1. Please explain how would your design change if the range of message lengths change from min=8B max=32B to:

a. min=1B ; max=32B

b. min=8B ; max=256B

This design using FIFOs should be easily scalable. For the a. option, it will change the tkeep counter which has one read for two bytes of data. As the data can be smaller, the tkeep counter will have to take into account that the data could be 1 byte. Apart from that, the system doesn’t need to change much more. For the b. option, where the msg\_len can goes up to 256 bytes, it will change the size of signals such as the payload signal which will need 2048bits.

1. What are the trade-offs for the chosen approach?

The chosen approach implies another input clock, which is 10 times faster to decode the data. It is the most simple and most scalable approach. As said in 4., this design should be easy to adapt for different message lengths, however this might not be the best approach for a latency optimized system. By using two parallel FIFO, we optimize high throughput but we are dependent of the FIFO latency.