FPGA Design Assignment

Strike Technologies

December 11, 2014

Introduction

This assignment is designed to assess your ability to design a functional and synthesizeable hardware block targeted for FPGA deployment. Please submit both your functional code in a Hardware Description Language of your choice as well as an associated testbench.

The Fair Queuing Problem

A common problem in ethernet networking is line aggregation, the solution to this problem should address that issue.

Imagine you work for an internet service provider who owns a single bi-directional fiber-optic link between two popular internet data-centers. Until this point, only your own company's traffic travelled across this link freely and without contention. However, times are tough and your employer decides to lease bandwidth on this link to the other ISP's at the data-centers.

Your task as part of the engineering team is to design a hardware module, targeted for FPGA, that aggregates all of the traffic from the new ISP's **as fairly as possible** into the single fiber link for the long haul. Including your own company, there are a total of 8 ISP's at each data-center.

In order to ensure fairness across all of the ISP's, you will be implementing a simple fair queuing algorithm as described here: `http://en.wikipedia.org/wiki/Fair_queuing#Algorithm`. A more detailed description of the algorithm as well as a simple example can be seen in this video starting at 10:31; `https://www.youtube.com/watch?v=tEU9b5N53fM`.

Your module will input messages from an array of 8-byte wide first-in-first-out (FIFO) buffers. There will be one input buffer per ISP. The FIFO's function in "showahead" mode which means when there is data available it will be exposed at its output. The read request signal only acts as an acknowledgement to receipt of data and pops the exposed word out of the buffer. See figure 1 for the timing diagram of this functionality.

For the purpose of this assignment, you will not be processing actual network packets, instead you will be handling a simplified message format.

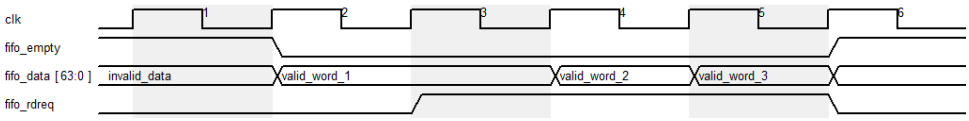

Figure 1: FIFO Timing

Each message will contain a single byte length field (the first byte of the message) which denotes the length of the message in **quantity of 8-byte words**. The rest of the message will be arbitrary payload data corresponding to the length field. See figure 2 for the message format.

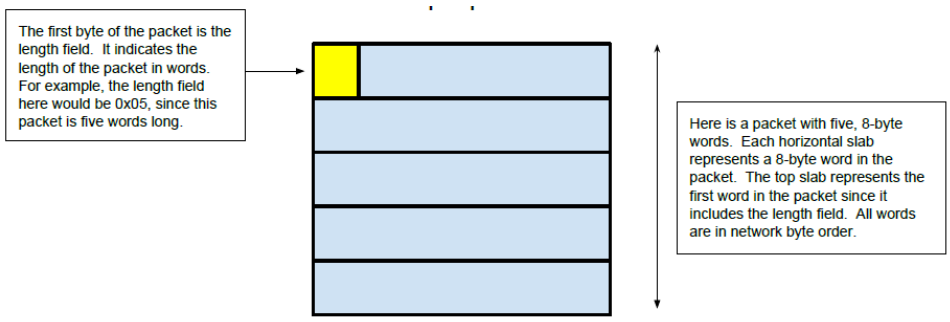

Figure 2: Message Format

At any point in time, there can be an arbitrary number of messages (including zero) in any of the input buffers. Each message can be 1 to 255 words long.

Messages will be forwarded from the ISP's to the output buffer with no changes to the format. i.e. the length field and associated data should hit the output in the same 8-byte words that were read on the input. The goal here is only to aggregate the inputs, no modifications are made to the actual data. See figure 3 for the block contract of your module. You are provided with a reset signal that will be asserted at startup as well as a 200MHz clock signal. Please be conscious of static timing concerns as you make design decisions.

Please submit your design files as well as any testbench hdl files you have used in the design and validation of this module. If you have any questions, you can contact either of us at msherman@striketechnologies.com or agoldstein@striketechnologies.com and we will be sure to promptly respond.

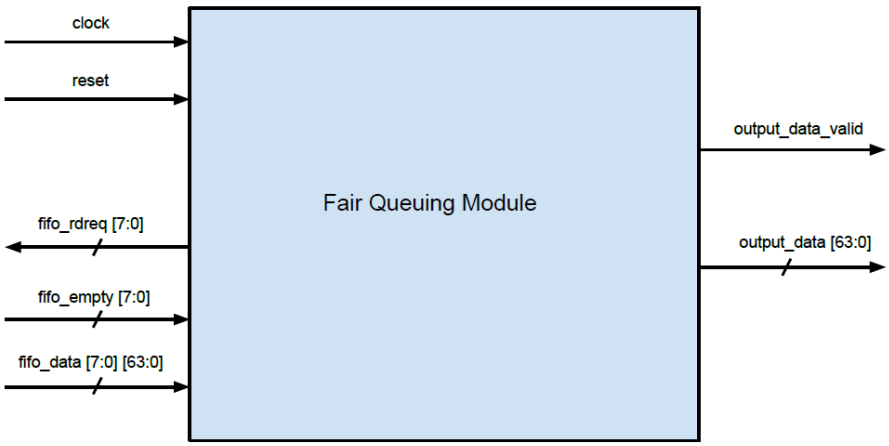

Figure 3: Block Contract of the Module