Packet Manipulation System Documentation

Marco Spaziani Brunella

University of Rome Tor Vergata

March 2017

Index

1 Introduction

The *Packet Manipulation System*[1] is a synthesizable, VHDL-descripted IP Core oriented to the manipulation of network packets in a datapath of a switch.
The target device is a Xilinx Virtex-7 FPGA, since the system has been developed on a NetFPGA SUME.

It is composed by these 6 elements:

- **Packet Manipulator Processor**[2]; is the CPU of the system
- **Instruction Memory**; holds the instructions to be executed by the PMP
- **Data Memory**; holds the data accessed by the PMP
- **TUSER Memory**; holds TUSER from the input AXI Stream system and provides width conversions for the PMP
- **TDATA Memory**; holds TDATA from the input AXI Stream system and provides width conversions for the PMP
- **AXI Stream Width Converter**; provides width conversions from the PMP to the output AXI Stream system

The PMP is an Harvard, static, 8-issue VLIW processor. It can process up to 8 32bit *syllable*, which are packed up in a wider 256bit *instruction*.
The PMP has a 32bit data bus with 32bit addresses wich provides connection to the TUSER memory, TDATA memory, Data memory and the AXI Stream width converter. These peripherals are memory mapped.
The PMP can perform 4 parallel Load/Store instruction, although only Store on different devices can be parallelized, since all the rams are synthesized as BRAM. An architectural overview of PMS is given in Fig.1.

[1] Referred as PMS from now on

[2] Referred as PMP from now on

The AXI Stream input system can be a single 10G Ethernet interface or the input arbiter of the NetFPGA.
Likewise, the AXI Stream output system can be a 10G Ethernet interface or the BRAM Output queues of the NetFPGA.
An implementation environment for the PMS is given in in Fig.2.

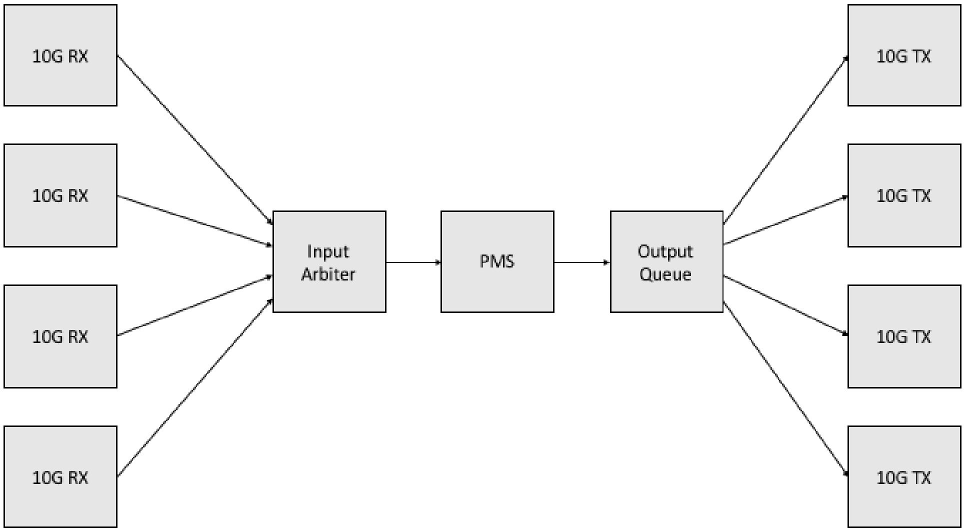

Figure 1: Implementation environment for the PMS