Configurable Port Processor Increases Flexibility in the Protocol Processing Area

Tomas Henriksson, Ulf Nordqvist, and Dake Liu
Dept. of Physics and Measurement Technology, Linköping University, SE-581 83 Linköping, Sweden
Phone: +46-13-28{8956, 5816, 1256}, Email: {tomhe, ulfnor, dake}@ifm.liu.se

Abstract

The limitation in networking is no longer only the physical transmission media but also the end equipment, which has to process the protocol control fields. In most end terminals this processing has been performed by the main processor, but different types of co-processor have lately appeared to relieve it from this task. These co-processors have high power consumption since they are based on a RISC core. Instead ASIC:s can be used, but they lack flexibility and are specific for only one single protocol. It is clear that a new approach is needed.

A new type of architecture for protocol processing has been specified. Our configurable port processor for protocol processing (CPP) could be situated in an end terminal or as part of a switch or router. The processor works with a non-von Neuman architecture to reduce the power consumption and keeps the flexibility by being heavily software reconfigurable. The configurations can be generated from a description language, for example SDL or C++. The CPP has the performance and power consumption similar to an ASIC synthesized by a protocol synthesizer, but has the observability and flexibility within the protocol processing area of a normal von Neuman processor. This makes the CPP suitable for System-on-Chip integration.

The CPP consists of two parts, see figure 1, a deep pipeline serial processor (DPSP) and a micro controller (μC). The protocol processing takes place in the DPSP. The μC only handles configuration and some interface functions, it never touches the main data stream. The DPSP is based on configurable functional pages (FP), which each take care of one small task such as checksum calculation or field extractions.

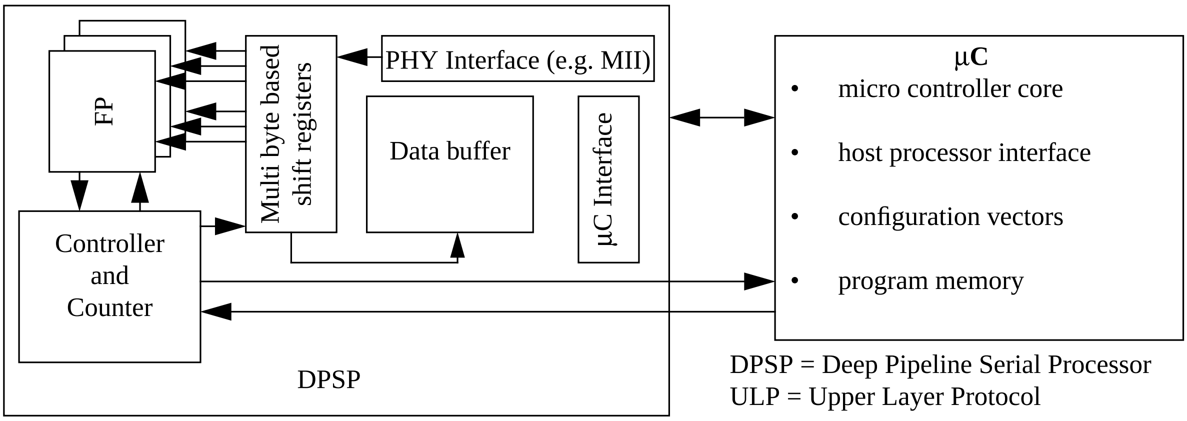

Figure 1:Overview of the architecture, the FP:s perform the actual protocol processing

Data is received via the PHY Interface and is then synchronized and parallelized and passed on to the multi byte based shift register. Data is then pipelined and each FP collects the protocol fields that it needs to perform its task. The FP:s are activated on control signals from the controller and counter (CC). The FP:s report to the CC by sending flags. The CC activates and shuts down FP:s according to the dataflow and configurations. In particular, when the CC receives a flag that indicates that the packet must be discarded the CC shuts down all FP:s to save power.

The CC and the FP:s are software reconfigurable. Software reconfiguration is controlled by the μC. Configuration can also be performed in the design phase. If the CPP is placed in a System-on-Chip where only specific protocols are used, the CPP can be optimized for that particular environment.

The CPP can perform protocol processing on layer 2, 3, and 4 of the ISO-OSI reference model (e.g. Ethernet MAC, IP, and TCP) at the same time. Since all three protocol layers are processed simultaneously the latency can be kept very low and the memory access can be minimum.