



Lomonosov Moscow
State University

SIMULATION MODEL FOR NETWORK PROCESSING UNIT



Stanislav Bezzubtsev
IPMCE

Moscow, Russia

stas.bezzubtsev@gmail.com

Vladislav Miroshnik
MSU

Moscow, Russia

miroshnikv@lvk.cs.msu.su

Yuliya Skobtsova
MSU

Moscow, Russia

xenerizes@lvk.cs.msu.su

Dmitry Volkanov
MSU

Moscow, Russia

volkanov@lvk.cs.msu.su

Shynar Zhailauova
MSU

Moscow, Russia

tshynar@arccn.ru

Introduction

Network processing unit (NPU) is a programmable embedded multi-core semiconductor system optimized for performing data operations at a speed appropriate to the data transmission medium.

Processing network data can be divided into several steps: parsing, classification and modification. In order to achieve high processing speed, one of the approaches is to use functionally specific stages for every step, designed for solving the corresponding problem. In this paper we evaluate applicability of general-purpose processing units for pipeline stages.

Imitation model description

The proposed simulation model of the NPU performs the following actions:

- incoming packet transfer to the network processor;
- storing the incoming packet in a local memory;
- packet header processing;
- packet forwarding decision based on the processing result;
- checking the network processing unit correctness.



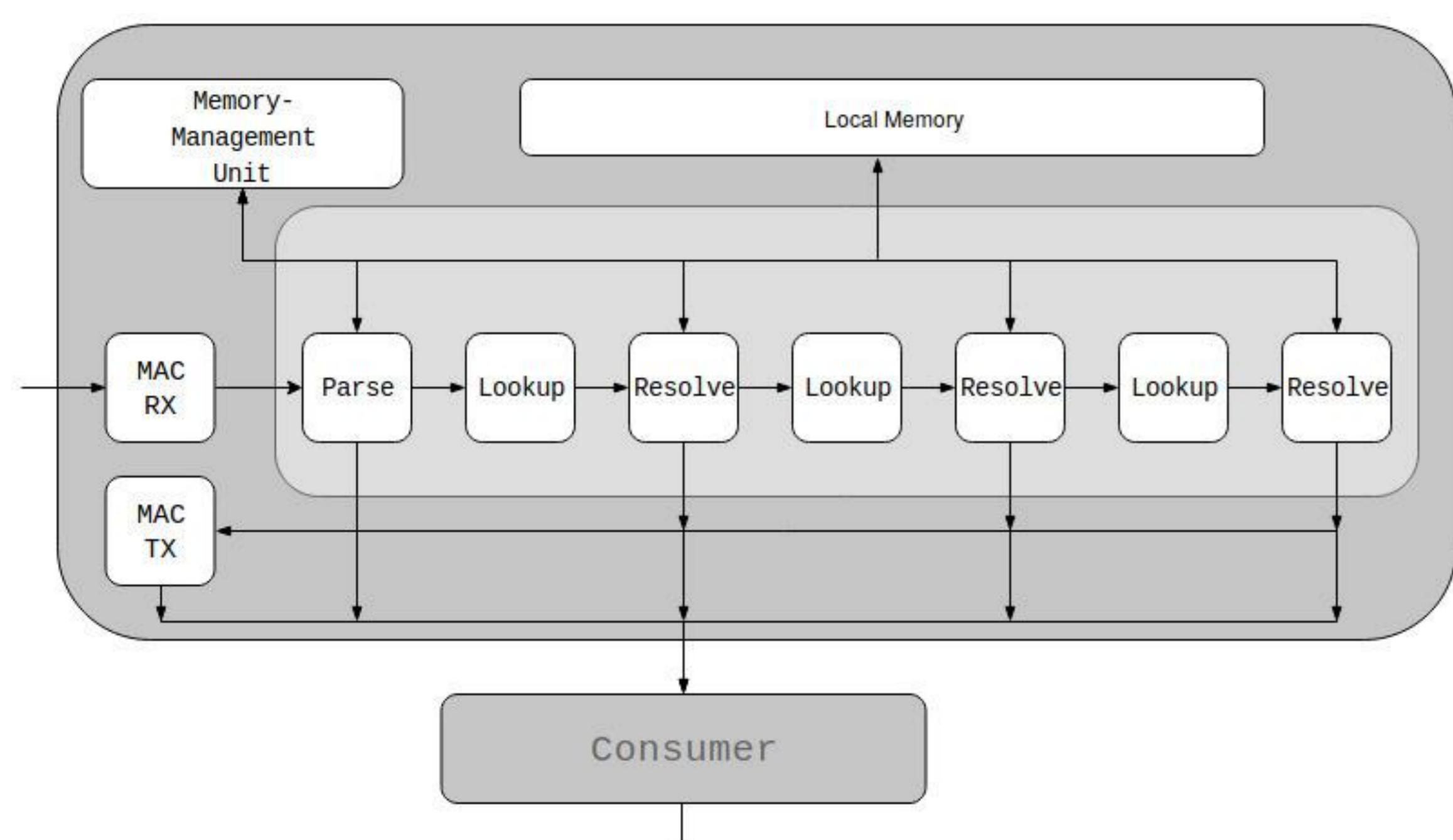
Figure above shows the structure of presented model. The model consists of three components:

- Provider,
- Device Under Test (DUT) NPU,
- Consumer.

DUT

The DUT module consists of the following components:

- MAC RX — Ethernet interface receiver;
- Data processing pipeline — 7-stage pipeline with functionally specialized stages of the following types:
 1. Parse — parses header defined by programmer for packet classification.
 2. Lookup — searches in specified classification table by header field (classification).
 3. Resolve — processes classification result: applies header modification and makes decision about further header processing.
- Memory-Management unit (MMU) — unit for managing; MAC TX — Ethernet interface transmitter.
- Main Memory — external (out-of-chip) memory used as packet body storage.
- Local Memory — internal (on-chip) memory block for storing packet contexts (i.e., headers and metadata), classification tables.



Model implementation

The simulation model of the network processor is implemented in C/C++ languages using SystemC, the open-source C++ library. Model components are represented by C++ classes. Communication between objects of these classes is performed using the SystemC TLM library. Objects are connected to each other by unidirectional FIFO channels from TLM library.

The model uses dynamic libraries containing the implementation of all Parse, Lookup and Resolve stages for each use case scenario. Also during startup, the following parameters can set as options: the clock period, the number of cycles during which the simulation takes place, the set of delays for each stage.

A set of software products developed by KM211 company were adapted to meet the needs of this research. The possibility of using the KMX32 family microcontroller (designed by KM211) in the NPU was also studied. To measure the processing unit operating time were used the KMX32 family of processors.

For performing functional testing of the model a set of packet processing use case scenarios are implemented in C. Use case scenarios can be classified into three groups (their reference names are in the parentheses):

1. Use case scenarios for SDN switch network applications (B2C-AR, B2C-DR, Multicast);
2. Use case scenarios for aggregating, queuing and redirecting traffic by SDN (Inband, LAG, LACP/LLDP/QoS);
3. Use case scenario for L2 switch with MAC learning (L2-switch VLAN).

Results

We evaluated the minimal pipeline throughput bound for the developed model in the case of the bottlenecks of pipeline. For each use case scenario test the bottleneck is a stage where packet processing time is the longest. The resulting throughput values for each scenario are shown below:

Сценарии	1 pipeline x 1 port, Mb/s	1 pipeline x 2 ports, Mb/s	1 pipeline x 24 ports, Gb/s	2 pipeline x 48 ports, Gb/s
B2C-AR	97,7	195,4	2,3	4,7
B2C-DR	97,7	195,4	2,3	4,7
Inband	101,9	203,8	2,4	4,5
LACP/LLDP/QoS	94,2	188,4	2,2	4,5
LAG	103,3	206,6	2,5	5
Mirror	93,3	186,6	2,2	4,5
Multicast	102,3	204,6	2,5	4,9
P2P	97,7	195,4	2,3	4,7
L2 switch + VLAN	98	196	2,3	4,7

Conclusion

The results show that in considered case, when traffic mainly consists of small packets, proposed architecture does not reach the minimal speed provided by existing NPUs. Thus, it is necessary to refine the available processor cores used by stages and improve the proposed architecture.

Bibliography

- [1] Orphanoudakis T., Perissakis S. Embedded Multi-Core Processing for Networking //Multi- Core Embedded Systems - 2010. - CRC Press - p. 399-463.
- [2] Black D. C., Donovan J., Bunton B., Keist A. SystemC : From the Ground Up - 2010. - Springer.
- [3] Kaushalram A. S., Budiu M., Kim C. Data-plane stateful processing units in packet processing pipelines, US Patent App 14864088. – 2017.
- [4] KMX32 family microcontrollers.
<http://km211.com/en/microcontrollerplatform> [In Russian]