# Why new protocolks were added not to system but to user layer

Original project implements all protocols at the system layer. This is best solution from architecture point of view because user no need think about any system purposes. Everything is hidden in the low level modules. Also that modules are located in the system folders. When any new board is added, it will use that files.

But of course, we must pay for this architect. The payment is using of multiplexers. There are many AXI Multiplexers are present in the project for choose the way of data from layer to layer. That multiplexers are producing many wires. And wires are worst item for Lattice FPGA.

Lattice FPGA has no good solutions for inter-block communication. Most connections are going through trace resources. That is why router has much problems to optimize FMax. AS a result, any change of design can significantly changes FMax. Strange, but sometimes adding of extra wires improves FMax to 10..20 MHz. Sometimes remove of any modules drops FMax. That is wwhy we can say that FMax depends not only from Netlist, but also from router’s quality. And less wires => easier processing for router.

That is why if we can skip multiplexer for Lattice FPGA, better to do it. And that is wy we added all protocols to User Layer.

For now , all support files are also located in the Top Level Folder.Maybe later will be better to move them to any common place because now we must synchronize that files when any changes were added. From other hand, now we are providing many experiments with different boards and sometimes that differences are important. But in the future, all versions of the protocol support modules will be the same.

# Which buses we have at the User Layer

The User Layer has two global buses (x2 directions = 4 totally) for provide the functionality.

**rx\_udp\_.**.. This bus will be activated when detected UDP Datagram is came from the network. The receiver is processing datagrams in the promiscuous mode but it contains all addressing information such as Target IP, Target MAC and Target Port. User layer’s logic can use or ignore it.

**tx\_udp\_**.. This bus can be used for send UDP Datagram from User Layer to the network.

**rx\_ip\_**... All Ethernet Frames which were not served at Low Level and not detected as UDP Datagrams will be routed to this bus. We are using it for provide ICMP Protocol

**tx\_ip\_**... If we need send any non-UDP Packet, we must do it, using this bus

# Project’s items

Now project contains 4 subprojects.

1. Overclocked version for ColorLight\_5A-75B board
2. 100M version for ColorLight\_5A-75B board. System Bus has CLK= 104.167 MHz. Why not 100M? This value is got from the PLL Generator. We ordered primary clock 125 MHz because this is most important for RGMII. Of course, we ordered Secondary Frequency 100 MHz but got this value. Also Chinese Oscilloscope and Chinese Logic Analyzer shows that duty cycle of PLL Output is not 50%. Maybe this is problem of Chinese equipment at high frequency, maybe not.
3. Overclocked version for ColorLight\_i5 Board (DIMM Version)
4. 100M version for ColorLight\_i5 Board (DIMM Version) with System Clock 104.167 MHz

Why we still using two boards? The most problem of current DIMM solution is motherboard. We don’t see any input pins. But we need Input pins at least for force DHCP or AutoIP processing.

Описать четыре папки верхнего уровня. Обосновать, зачем нужны все четыре

# Project’s top hierarchy



The Top Level Module has name **fpga**. It is located in file **fpga.v**. This tradition is came from the original project. This file contains all FPGA Based items such as PLL, and some non-standard pins, if needed. That pins are needed for debug purposes, but not for real production. Also traditionally, this module contains LED Blinker because it also depends of hardware depended frequency divider. This is last module in ther hierarchy, which contains any physically depended information.

Next layer has name **fpga\_core**. This module is also came from the original project. Exacxtly this module is used for automatic tests in the original project. It has unified interface. This interface is same for Xilinx, Altera, Lattice etc.

The module **fpga\_logic** is added at this project. We separated it from fpga core for simplify simulation process in the ModelSim. This module receives Ethernet data already packed into AXI-Stream format. That is why we can use files, captured by WireShark without translate them into any more low level fromat and without thinking about any low level CRS. The mail FSM is located in this file.

# Project’s metafiles

## Makefile

Описать детали. В том числе, как формируется Yosys скрипт

## NextPNR Script

Описать зачем и как работает Питоновский скрипт

# Protocol Based Modules

## Module SpiFlashReader

Todo

## Module DHCPhelper

Todo

## Icmp

Todo

## Module mDNShelper

Todo

## Module prng

Todo

## Module lfsr

Todo

# Main FSM Logic