ERI Design: IDEA and POSH Ethernet Controller Open Source IP Cores

CHINH LE
LeWiz Communications, Inc.

IDEA and POSH Phase I Integration Exercise
Detroit, MI

17 Jul 2019 – 19 Jul 2019

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions, and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government

Open Session Slides

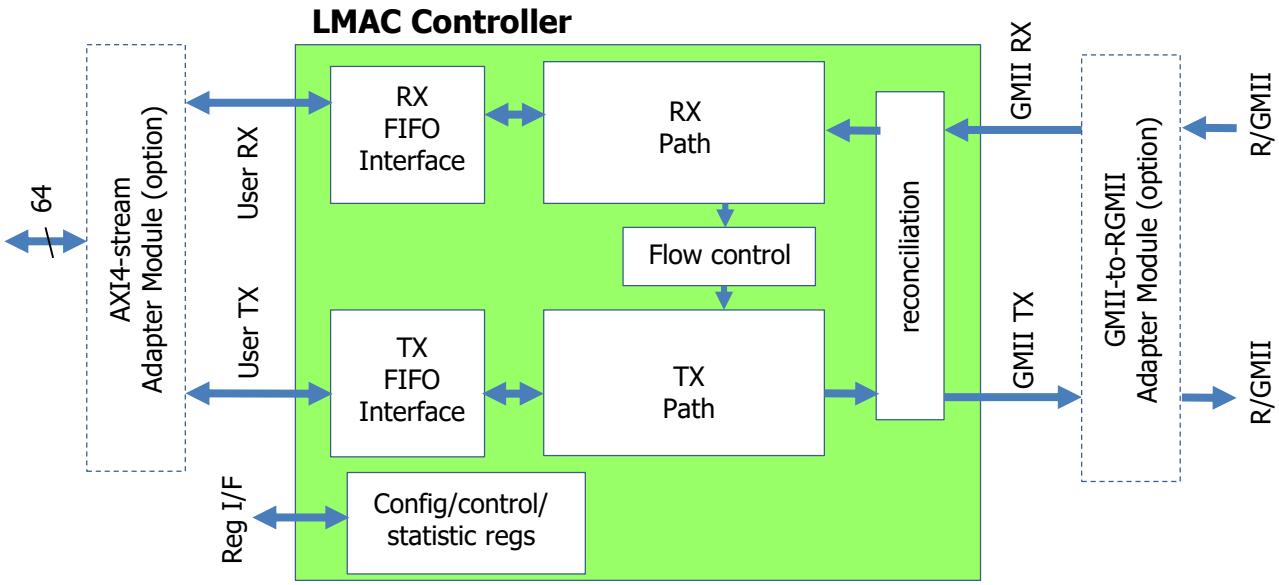
Program Overview (Open Session)

Development of 3 Open Source Ethernet controller (MAC) IP cores:

- Core 1: 10/100/1000Mbps
- Core 2: 10G/5G/2.5G/1Gbps
- Core 3: 100G/50G/40G/25G/10Gbps
- Designed for easy migration to SoC from FPGA emulation
- Unified architecture: Simpler interface, software driver, easy to migrate to different speed, different PHYs, suitable for small IoT to large applications
- Open source: but complete code, simulation bench, FPGA project, drivers, documentation, and support

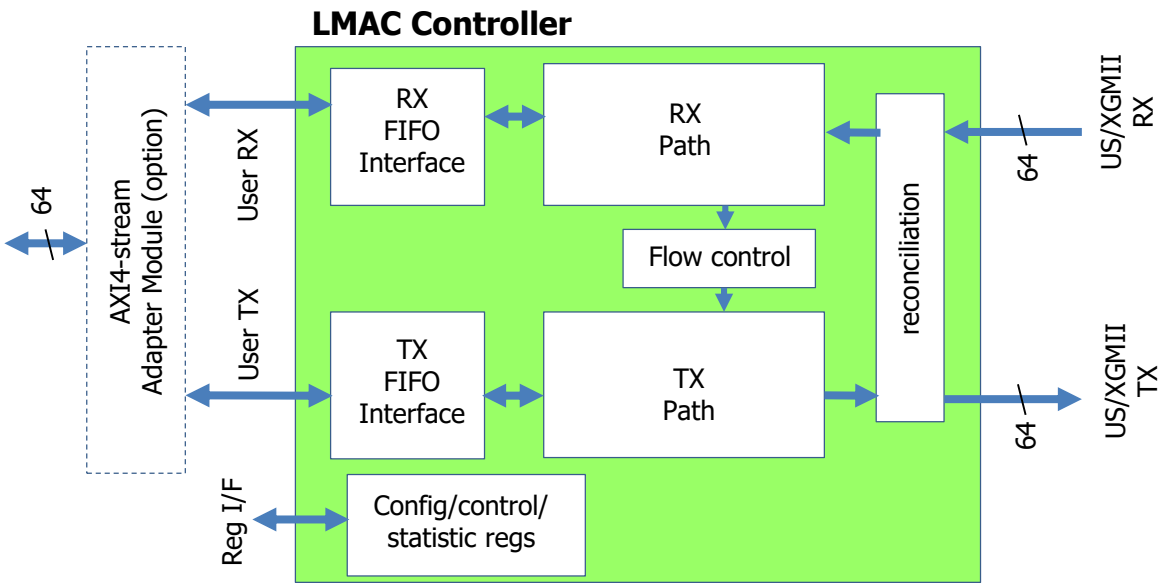
Technical Approach (Open Session)

CORE 1: 10/100/1000Mbps



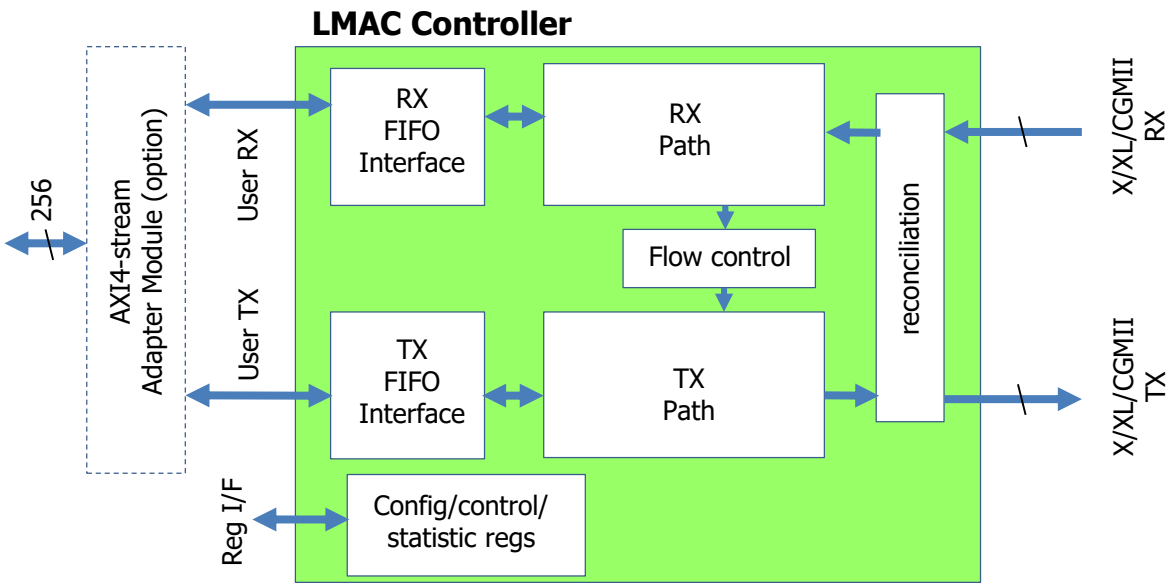
Technical Approach (Open Session)

CORE 2: 10G/5G/2.5G/1Gbps



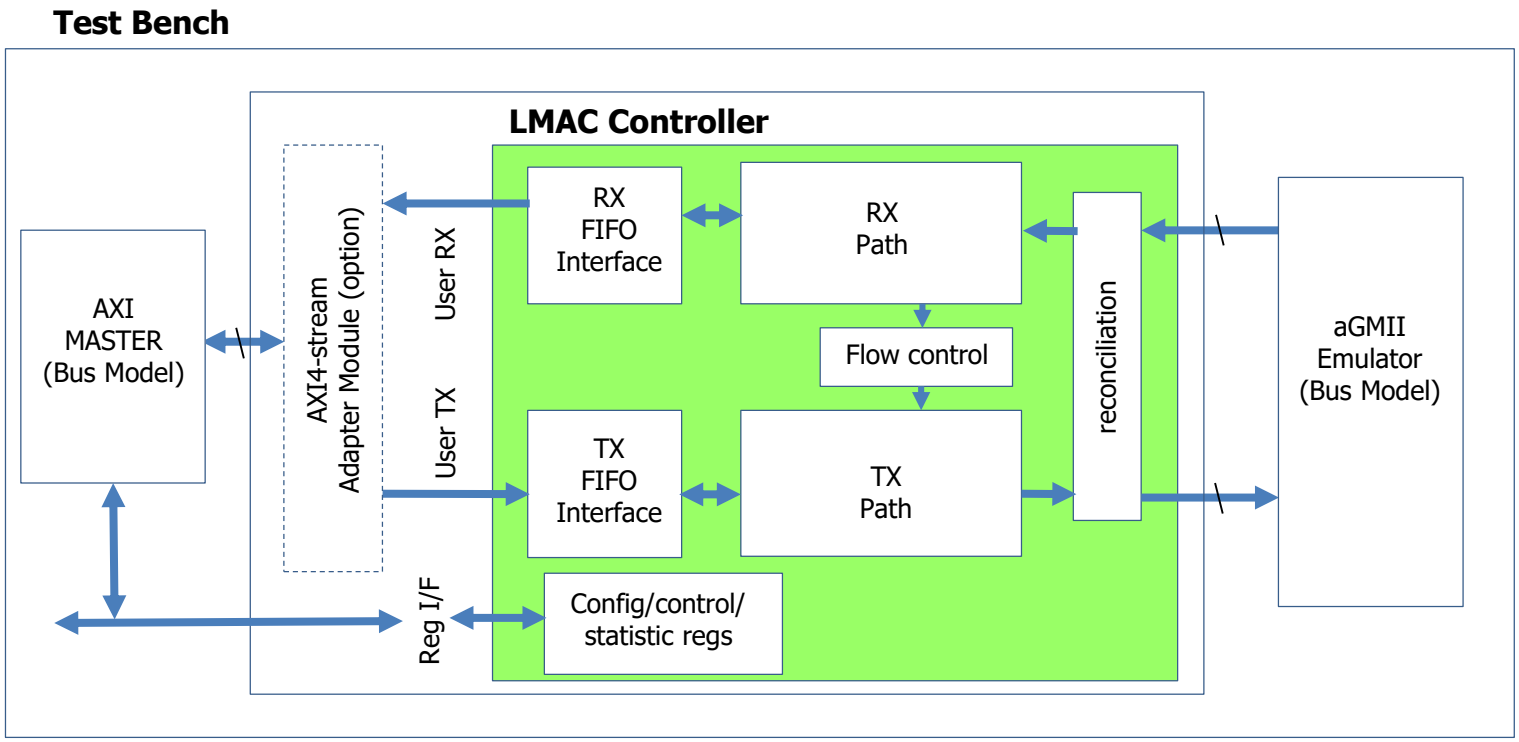
Technical Approach (Open Session)

CORE 3: 100G/50G/40G/25G/10Gbps



Technical Approach (Open Session)

Verification: Test bench (complex and most time consuming)



NOTES:
- Bus models allows packet level testing
- Transmit and receive flows
- No vendor specific or technology dependent

Technical Approach (Open Session)

Status

- APPROVED FOR RELEASE TO GITHUB
  - Released Core 1 and 2
  - Documentation (specs and usage), Instructions
  - Tests and results used in verification of IP cores in simulation, test benches
  - Github.com → search for LMAC CORE
  - You should see LMAC_CORE1 and LMAC_CORE2
- LICENSING - open source hardware ecosystem
  - Access model – open source (Github)
  - License using Lesser GPL 2.1

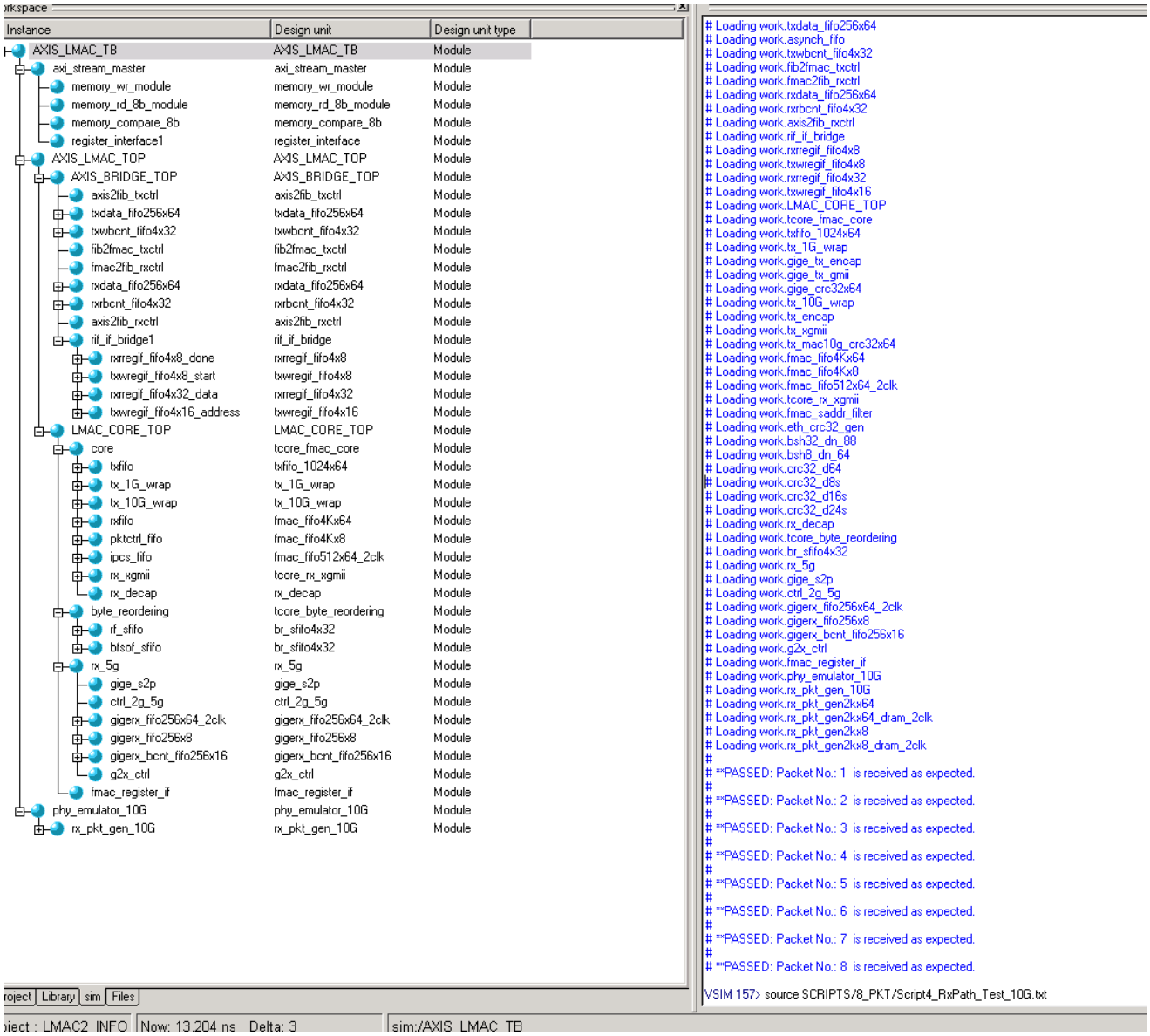
Future Plans (Open Session)

- Near term plan:
  - 3 months – Core 3 architecture review and initial code release
  - 6 months – Final Core 3 release with docs, simulation bench

Screenshot of Sim Script

```
##
## Copyright (C) 2018 LeWiz Communications, Inc.
##
## This library is free software; you can redistribute it and/or
## modify it under the terms of the GNU Lesser General Public
## License as published by the Free Software Foundation; either
## version 2.1 of the License, or (at your option) any later version.
##
## This library is distributed in the hope that it will be useful,
## but WITHOUT ANY WARRANTY; without even the implied warranty of
## MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
## Lesser General Public License for more details.
##
## You should have received a copy of the GNU Lesser General Public
## License along with this library release; if not, write to the Free Software
## Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA
##
## LeWiz can be contacted at:  support@lewiz.com
## or address:
## PO Box 9276
## San Jose, CA 95157-9276
## www.lewiz.com
##
##    Author: LeWiz Communications, Inc.
##    Language: Verilog
##

# source SCRIPTS/8_PKT/Script4_RxPath_Test_10G.txt
# Script to test LMAC: test rx path, 64-bit mode (XGMII).
# fmac_speed: 000 = 10 Gig mode
#             101 = 5 Gig mode
#             010 = 2.5 Gig mode
#             001 = 1 Gig mode
# rx_pkt_gen_sel selects the source of the data on RxD and RxC. If 1 = data is from the rx_pkt_gen. If 0 = loopback data (TxD and TxC is sent to RxD and RxC).


vsim work.axis_lmac_tb

view wave
do E:/LMAC2_INFO/waveforms/rx_path_test

view -new wave
add wave -r /*

mem load -i E:/LMAC2_INFO/AXIS_MASTER/memory_wr_data.txt -format hex /AXIS_LMAC_TB/axi_stream_master/memory_wr_module/memory_wr_data
mem load -i E:/LMAC2_INFO/AXIS_MASTER/memory_wr_ctrl.txt -format hex /AXIS_LMAC_TB/axi_stream_master/memory_wr_module/memory_wr_ctrl

mem load -i E:/LMAC2_INFO/SCRIPTS/8_PKT/rx_pkt_gen_data.mem -format hex /AXIS_LMAC_TB/phy_emulator_10G/rx_pkt_gen_10G/rx_pkt_gen2kx64_inst/dram_data/mem
mem load -i E:/LMAC2_INFO/SCRIPTS/8_PKT/rx_pkt_gen_ctrl.mem -format hex /AXIS_LMAC_TB/phy_emulator_10G/rx_pkt_gen_10G/rx_pkt_gen2kx8_inst/dram_ctrl/mem

force -freeze sim:/AXIS_LMAC_TB/fmac_speed 3'd0 0

force -freeze sim:/AXIS_LMAC_TB/phy_emulator_10G/rx_pkt_gen_start_addr 64'd00 0
force -freeze sim:/AXIS_LMAC_TB/phy_emulator_10G/rx_pkt_gen_read_cnt 11'd0 0


# Select the rx_pkt_gen memory data
run 200ns
```

Screenshot of Modelsim – exec & results



Screenshot of Waveform Results