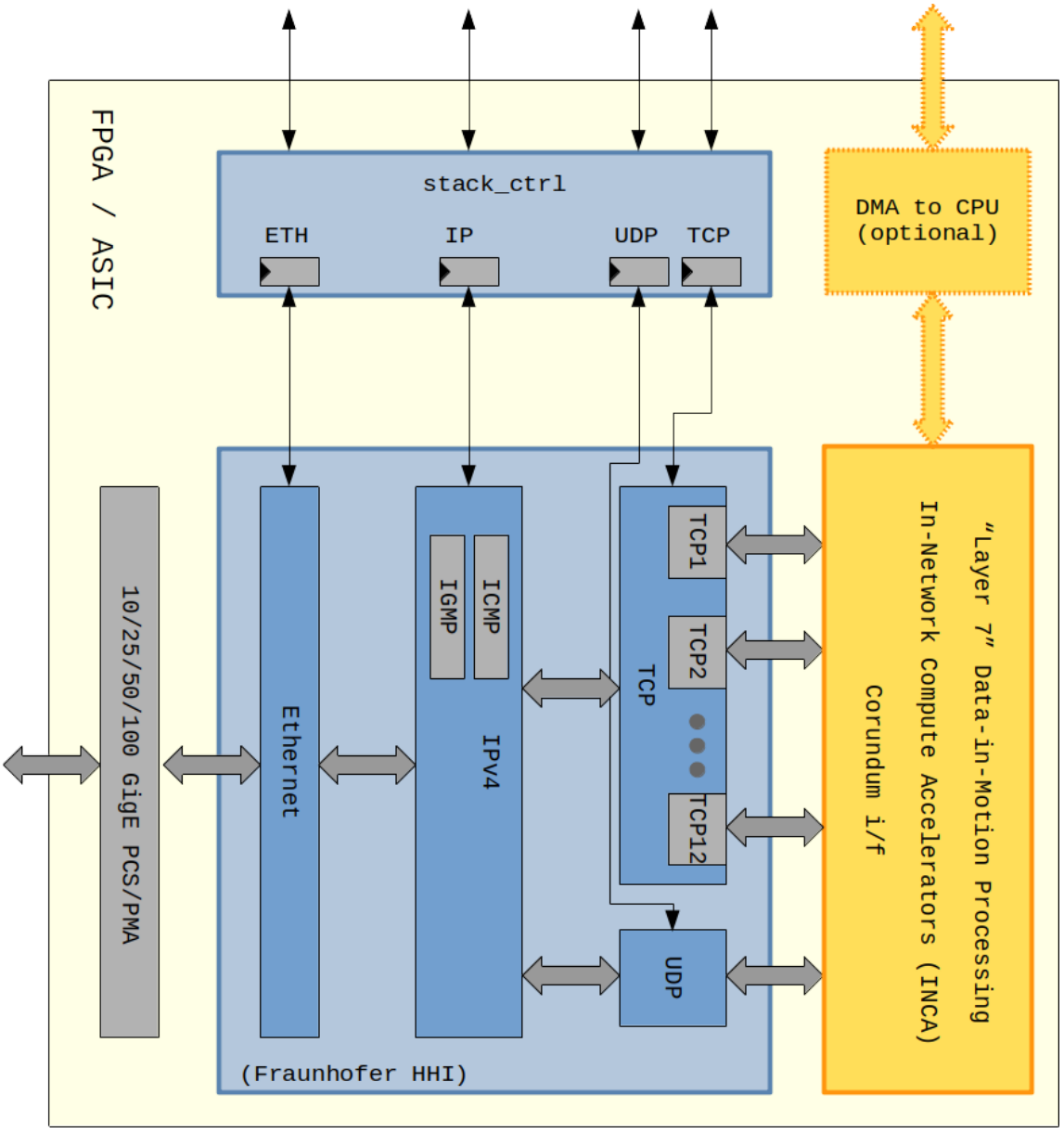
MLE Technical Brief 20220408
Network Protocol Accelerator Platform

A stand-alone TCP/UDP/IP Stack Full-Accelerator Subsystem allowing communication at full line rate and low latency



Version: 1.8.0.

Date: April 2022

Authors: Ulrich Langenbach, Andreas Schuler, Endric Schubert

Features

MLE's Network Protocol Accelerator Platform (NPAP) for 1/2.5/5/10/25/40/50/100 Gigabit Ethernet is a TCP/UDP/IP network protocol Full-Accelerator subsystem which instantiates the standalone 128 bit TCP/IP Stack technology from German Fraunhofer Heinrich-Hertz-Institute (HHI). This Fraunhofer HHI 10 GbE TCP/IP Stack was designed for embeddable FPGA and ASIC system solutions and offers the following features:

- Interface to 1 / 2.5 / 5 / 10 / 25 / 40 / 50 / 100 Gigabit Ethernet
- Bidirectional datapath width 128 bit each
- Full line rate up to 70 Gbps per individual TCP session in FPGA
- Full line rate >100 Gbps per individual TCP session in ASIC
- Low round trip time NPAP-to-NPAP 700 nanoseconds for 100 Bytes RTT

Designed for maximum flexibility, NPAP implements in programmable logic the most common network communication protocols:

IPv4 The core of the most standards-based networking protocols

TCP Reliable connectivity for direct secured connectivity

UDP Widespread protocol to enable simple direct or multicast communication

ICMP Diagnostic protocol to validate connections

IGMP Enables joining of multicast groups

Due to the modularity NPAP can easily be enhanced by application specific protocols.

Originally targeted to deliver close to the theoretical line-rate of 10 Gigabit Ethernet, a 128 bit wide datapath in combination with a pipelined architecture allows to scale throughput to line-rates of 50 GbE, and beyond, when using modern FPGA fabric.

Applications

NPAP enhances your real-time application with a leading fast data connectivity. The powerful architecture of the underlying TCP/UDP/IP Stack allows to transfer data at line-rate with low processing latency without using any CPUs in the data path. The widespread TCP/IP UDP communication protocol suite using industry standard network infrastructure addresses a wide-range of applications:

- FPGA-based SmartNICs
- In-Network Compute Acceleration (INCA)
- Hardware-only implementation of TCP/IP in FPGA
- PCIe Long Range Extension
- Networked storage, such as iSCSI
- Test & Measurement connectivity
- Automotive backbone connectivity based on open standards
- Video-over-IP for 3G / 6G / 12G transports
- Bring full TCP/UDP/IP connectivity to FPGAs
- High-speed sensor data acquisition:
  stream data out of FPGAs into Network-Attached Storage (NAS)
- High-speed robotics control and machine-to-machine:
  Stream data from servers via FPGA into actuators
- Hyper-converged computational storage acceleration for “over-Fabric” NVMe/TCP
- Deterministic low-latency, high-bandwidth alternative to lwIP or Linux on embedded CPU

Description

High performance programmable logic based, standalone TCP/IP stack featuring transparent handling of complete TCP/IP and UDP protocol tasks, e.g. packet encoding, packet decoding, acknowledge generation, link supervision, timeout detection, retransmissions and fault recovery. Complete automatic connection control including tear up and tear down. Transparent checksum generation and checksum checking, integrated flow control. RFC 793 compatibility (TCP/IP stack for Windows and Linux). Depending on the project's needs, deliverables can be:

- HDL source code or netlist
- Integrated FPGA system implementation
- Testbenches and scripts for real-life testing
- Comprehensive documentation and interfacing guide
- Development & design-in support

NPAP is optimized to ensure the best bandwidth-delay product performance for your application. The IP core described herein is easy to port to FPGA and ASIC target platforms.

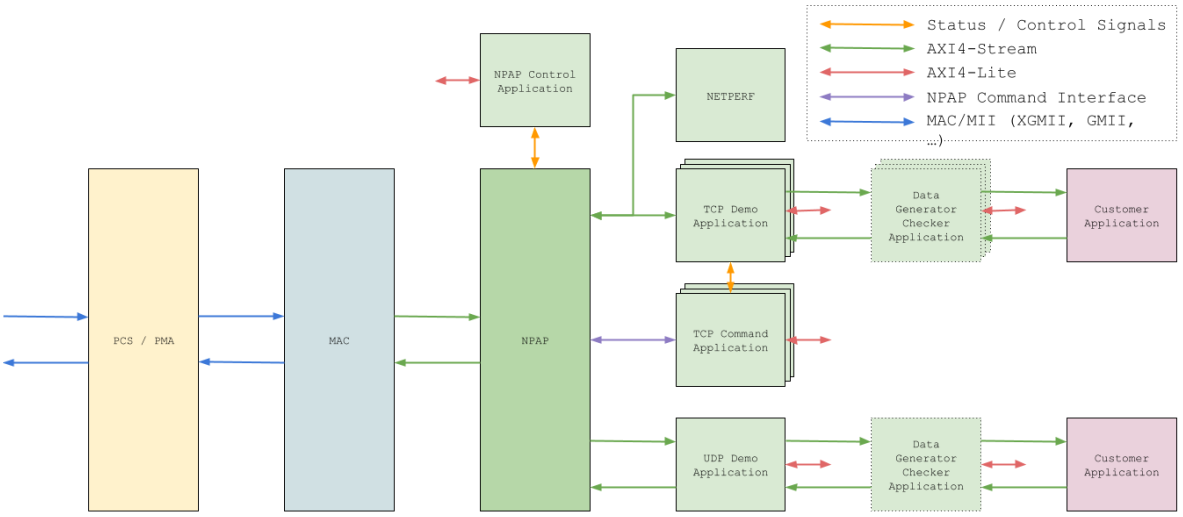
Technical Features

Feature	Specification
Supported Interfaces	GMII, RGMII, SGMII, PSC/PMA 128 bit wide AXI4-Stream
Ethernet Media Access Controller	Fraunhofer HHI 10G Low-Latency MAC, or Xilinx 10G/25G Ethernet Subsystem (PG210), or Xilinx 100G Ethernet Subsystem (PG165), or Intel 10G / 25G Ethernet FPGA IP
Supported protocols (Hardware based)	Ethernet, ARP, IPv4, ICMPv4, IGMPv4, UDP & TCP
Number of simultaneous connections	One per TCP engine instantiation
Interface to application	AXI4-Stream 128-bit, or custom TCP session control interface option
Supported FPGAs	Complete stack uses generic VHDL code
Tested FPGA series	Xilinx Virtex 4 to Virtex UltraScale+, Xilinx Spartan 6, Xilinx Zynq-7000, Xilinx Zynq UltraScale+ MPSoC, Xilinx Zynq UltraScale+ RFSoC, Altera Cyclone IV series, Intel Cyclone 10 GX series, Altera Stratix V, Intel Stratix 10 GX series, Intel Agilex F Series, MicroSemi Polarfire

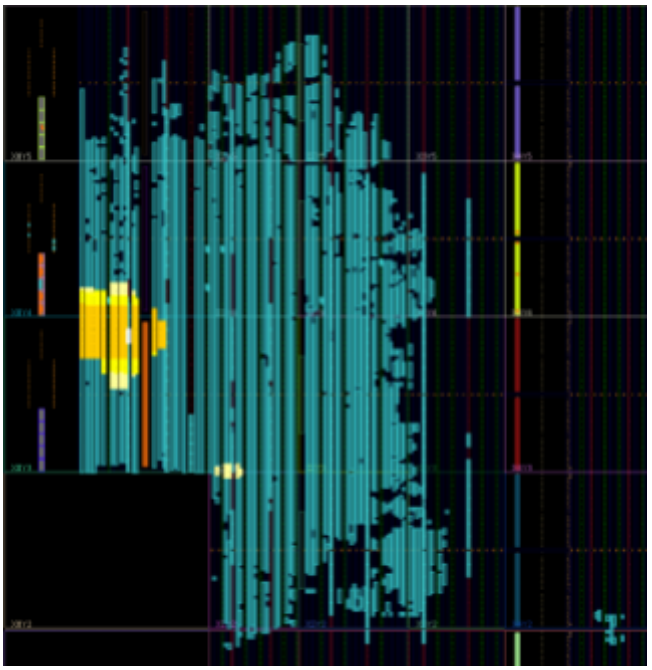
[1] Data in preparation

Implementation Details

Highly modular implementation using defacto standard AXI4 interfaces to support RTL synthesis flows for various FPGA vendors and ASIC RTL design flows.



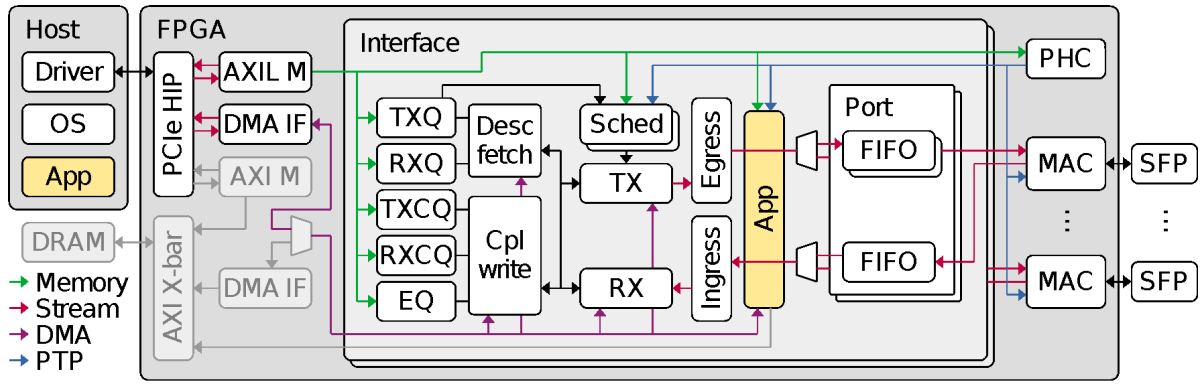
Starting with Release 1.7.1. RTL is optimized for FPGA pipelining such as Intel Hyperflex or Xilinx IMux.



Deliverables include IEEE 1685 IP-XACT packages including “helper IP blocks” plus non-IP-XACT FPGA reference design project for AMD/Xilinx Vivado 2021.2 or Intel Quartus 2021.2, or MicroSemi Libero 2021.1.

Network Interface Card Functionality

NIC functionality is supported by complementing NPAP with CORUNDUM.io, the open-source FPGA-based NIC.

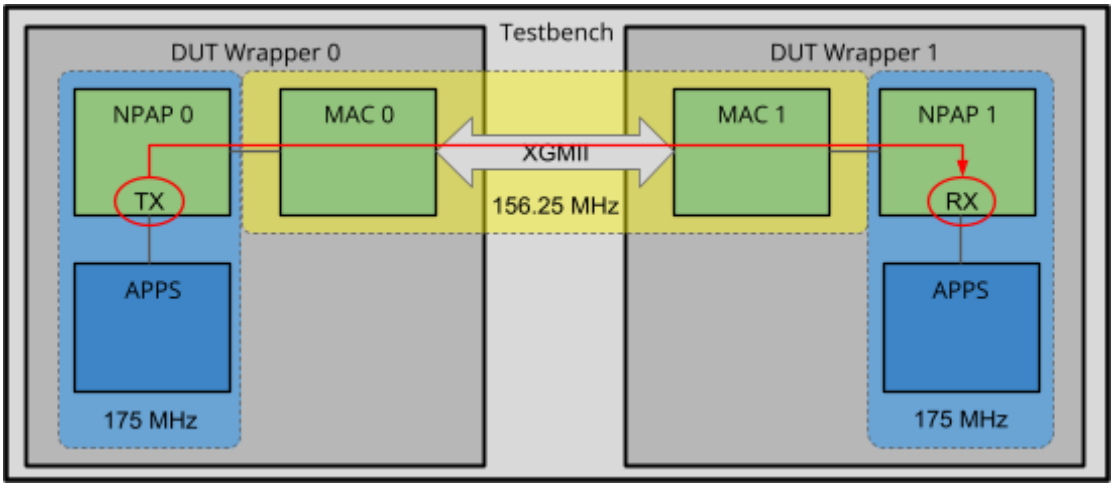


Latency Analysis Results

MLE analyzed processing latency using RTL simulation of two instances of NPAP (clocked at 175 MHz) connected via the 10G LL MAC via XGMII (clocked at 156.25 MHz).

TCP Payload Size [Byte]	Latency [ns]
1	462,8
32	485,8
64	520,0
160	656,0
448	1092,1
960	1868,9
1216	2251,3
1456	2622,7

Latency was measured “door-to-door”, i.e. we measured the time difference between sending payload data from one NPAP instance via TCP/IP until receiving that payload data at the other instance of NPAP, see the system-level block diagram:



Characteristics (when used with 10 GbE)

While the latency data above are from RTL simulation, the following table lists some key characteristics for NPAP when deployed (and measured with target hardware) with 10 GigE physical Ethernet on FPGA hardware.

Symbol	Parameter	Condition	Value	Units
RTT_{avg}	Average round-trip time	BL= 1 BL=10 BL=100 BL=1k	10,10 14,20 24,54 26,75	µs µs µs µs
TPR_{avg}	Average network throughput	BL = 1 BL = 10 BL = 100 BL = 1k	4,17 7,21 8,97 9,23	Gbps Gbps Gbps Gbps
BDP_{avg}	Average bandwidth-delay product	BL = 1 BL = 10 BL = 100 BL = 1k	44 101 255 246	kbit kbit kbit kbit

[1] Burst length BL=1 (=8 kByte)

[2] Measurement setup = HHI Stack-to-HHI Stack

[3] The values are valid for XILINX Virtex 5XC5VFX130T, speed grade - 2 FPGA. The reference test hardware is the HHI 10G EthEval board.

Evaluation Reference Design

For evaluating the functioning and the performance of NPAP MLE provides Evaluation Reference Designs (ERD) for several FPGA Development Kits:

- NPAP-10G on in Xilinx ZCU102 with Zynq Ultrascale+ MPSoC ZU9EG
- NPAP-25G on Xilinx ZCU111 with Zynq Ultrascale+ RFSoC ZU28EG
- NPAP-25G on Fidus Sidewinder 100 with Zynq Ultrascale+ MPSoC ZU19EG
- NPAP-10G Xilinx ZC706 with Zynq-7045 SoC
- NPAP-10G on Intel Cyclone 10 GX Development Kit
- NPAP-10G on Intel Stratix 10 GX Development Kit
- NPAP-10G on Microsemi PolarFire MPF300-EVAL-KIT)

Each ERD typically instantiates the full stack including MAC, Ethernet, IPv4 (with ICMP and IGMP), plus 3 TCP session instances, plus the UDP block, plus Netperf/Netserver implementation in programmable logic. The Netperf/Netserver block is compatible with open source Netperf/Netserver 2.6 and can be used for functionality analysis and for performance benchmarking.

For SoC-FPGAs such as Xilinx Zynq-7000 or Xilinx Zynq UltraScale+ MPSoC the ERD can run Linux, and Netperf/Netserver control commands can be set and results can be looked at by logging in via UART or SSH (RJ45).

Here some exemplary setups from MLE's NPAP Test Lab:

Resource Estimates for Xilinx Ultrascale+ Series

The following table shows resources for Xilinx Zynq Ultrascale+ MPSoC ZU19EG compiled with Xilinx Vivado 2018.3 - instantiating the following design features:

- 10 GigE Low-Latency MAC from Fraunhofer HHI
- Ethernet block
- IPv4 block
- UDP block
- 3 instances of TCP blocks

Instance	Module	Total LUTs	Logic LUTs	LUT RAMs	SRLs	FFs	RAMB36	RAMB18	DSP48 Blocks
npap_tcp_udp_wrapper_u0	npap_tcp_udp_wrapper	33277	31755	1506	16	35034	71	10	6
npap_tcp_udp_wrapper_u0	npap_tcp_udp_wrapper	31359	29837	1506	16	34053	71	10	6
(npap_tcp_udp_wrapper_u0)	npap_tcp_udp_wrapper	65	65	0	0	0	0	0	0
npap_tcp_udp_top_u0	npap_tcp_udp_top	31294	29772	1506	16	34053	71	10	6
(npap_tcp_udp_top_u0)	npap_tcp_udp_top	0	0	0	0	1	0	0	0
gen_hhi_to_axis_adapter[0].u	hhi_to_axis_adapter_5	8	8	0	0	0	0	0	0
gen_hhi_to_axis_adapter[1].u	hhi_to_axis_adapter_6	8	8	0	0	0	0	0	0
gen_hhi_to_axis_adapter[2].u	hhi_to_axis_adapter_7	8	8	0	0	0	0	0	0
wrapper_ll_ip_tcp_u0	Wrapper_LL_IP_TCP	31270	29748	1506	16	34052	71	10	6
(wrapper_ll_ip_tcp_u0)	Wrapper_LL_IP_TCP	42	42	0	0	203	0	0	0
GEN_MAC_NET_CONV_HHI.u	MacNetworkLayerConversion	370	290	80	0	554	0	0	0
g0.i_tcpTxMux	TcpTxMux	242	242	0	0	3	0	0	0
gen_WithUdp.i_udp	wrapper_udp	2226	2144	82	0	3243	19	1	0
gen_tcpConnections[0].u	Wrapper_TCP__xdcDup__1	8030	7598	432	0	8113	16	3	2
gen_tcpConnections[1].u	Wrapper_TCP__xdcDup__2	7962	7530	432	0	8117	16	3	2
gen_tcpConnections[2].u	Wrapper_TCP	8018	7586	432	0	8117	16	3	2
iBusScheduler8	BusScheduler8	85	85	0	0	53	0	0	0
i_internetLayer	Wrapper_IP	2286	2222	48	16	2908	2	0	0
i_networkLayer	Wrapper_Networklayer	2051	2051	0	0	2737	2	0	0
i_rxLinkResetSync	ResetSync__xdcDup__20	0	0	0	0	2	0	0	0
i_txLinkResetSync	ResetSync__xdcDup__21	0	0	0	0	2	0	0	0
mac_10_gbe_wrapper_u0	mac10gbe_wrapper	1918	1918	0	0	981	0	0	0
mac10gbe_top_u0	mac10gbe_top	1918	1918	0	0	981	0	0	0
mac10Gbe_struct_u0	mac10Gbe_struct	1918	1918	0	0	981	0	0	0

Resource Estimates for Xilinx 7-Series

The following table shows resources for Xilinx 7-Series Kintex fabric (XC7Z045-2) compiled with Xilinx Vivado 2014.4 - instantiating the following design features:

- 10 GigE Low-Latency MAC from Fraunhofer HHI
- Ethernet block
- IPv4 block
- UDP block
- 2 instances of TCP blocks

Instance	Module	Total LUTs	Logic LUTs	LUT RAMs	SRLs	FFs	RAMB36	RAMB18	DSP48 Blocks
wrapper_mac_10gStack	(top)	29459	28425	904	130	25727	50	6	4
(wrapper_mac_10gStack)	(top)	241	241	0	0	0	0	0	0
i_10GStack	Wrapper_LL_IP_TCP__parameterized0	27045	26013	904	128	24644	50	6	4
g0.i_tcpTxMux	TcpTxMux__parameterized0	0	0	0	0	3	0	0	0
gen_WithUdp.i_udp	wrapper_udp__parameterized0	3538	3350	92	96	3732	14	0	0
gen_tcpConnections[0].i	Wrapper_TCP__parameterized0	9347	8963	384	0	7938	16	3	2
gen_tcpConnections[1].i	Wrapper_TCP__parameterized0_1	9349	8965	384	0	7938	16	3	2
iBusScheduler8	BusScheduler8__parameterized0	568	568	0	0	31	0	0	0
i_internetLayer	Wrapper_IP__parameterized0	2989	2913	44	32	3437	2	0	0
i_netLayerConv	MacNetworkLayerConversion__parameterized0	139	139	0	0	72	0	0	0
i_networkLayer	Wrapper_Networklayer__parameterized0	1125	1125	0	0	1491	2	0	0
i_txLinkResetSync	ResetSync__parameterized0_2	0	0	0	0	2	0	0	0
i_ResetStretch_aux	ResetStretch__parameterized0	75	74	0	1	35	0	0	0
i_ResetStretch_stack	ResetStretch__parameterized2	75	74	0	1	34	0	0	0
i_clk_stretch	clk_stretch	1	1	0	0	29	0	0	0
i_gen100ms	GenClk100ms__parameterized0	69	69	0	0	30	0	0	0
i_mac10GbE	mac10Gbe_Wrapper__parameterized0	1957	1957	0	0	955	0	0	0
(i_mac10GbE)	mac10Gbe_Wrapper__parameterized0	0	0	0	0	1	0	0	0
i_mac10Gbe	mac10Gbe__parameterized0	1957	1957	0	0	954	0	0	0

Resource Estimates for Intel Stratix-10

The following table shows resources for compiled with Quartus Prime v19.3 for 1SX280HN2F43E2VG - instantiating the following design features:

- 10 GigE Low-Latency MAC from Fraunhofer HHI
- Ethernet block
- IPv4 block
- UDP block
- 3 instances of TCP blocks

Compilation Hierarchy Node	ALMs used in final placement	ALMs used for memory	Dedicated Combinational ALUTs	Logic Registers	Block Memory Bits	Block M20Ks	DSP Blocks
	35194.0 (4.2)	200.0 (0.0)	44839 (18)	34020 (0)	2112512	163	3
gen_loopback_tcp_interface_wrapper_top[0].u	100.2 (0.0)	0.0 (0.0)	57 (0)	172 (0)	0	0	0
gen_loopbackServer.i_loopbackServer	100.2 (100.2)	0.0 (0.0)	57 (57)	172 (172)	0	0	0
gen_loopback_tcp_interface_wrapper_top[1].u	99.1 (0.0)	0.0 (0.0)	57 (0)	172 (0)	0	0	0
gen_loopbackServer.i_loopbackServer	99.1 (99.1)	0.0 (0.0)	57 (57)	172 (172)	0	0	0
gen_loopback_tcp_interface_wrapper_top[2].u	101.4 (0.0)	0.0 (0.0)	57 (0)	172 (0)	0	0	0
gen_loopbackServer.i_loopbackServer	101.4 (101.4)	0.0 (0.0)	57 (57)	172 (172)	0	0	0
mac_10_gbe_wrapper_u0	2089.5 (0.0)	0.0 (0.0)	2926 (0)	1378 (0)	0	0	0
mac10gbe_top_u0	2089.5 (0.0)	0.0 (0.0)	2926 (0)	1378 (0)	0	0	0
npap_tcp_udp_wrapper_u0	32624.9 (0.0)	200.0 (0.0)	41429 (0)	31867 (0)	2112512	163	3
npap_tcp_udp_top_u0	32624.9 (2.2)	200.0 (0.0)	41429 (4)	31867 (1)	2112512	163	3
wrapper_ll_ip_tcp_u0	32622.7 (67.2)	200.0 (0.0)	41425 (1)	31866 (196)	2112512	163	3
GEN_MAC_NET_CONV_HHI.i_netLayerConv	1948.2 (9.2)	0.0 (0.0)	785 (15)	2655 (0)	0	0	0
gen_64bitAlign.i_rxAlign	74.2 (74.2)	0.0 (0.0)	13 (13)	145 (145)	0	0	0
gen_rxSyncFifo.i_fifoCDC	1855.4 (1855.4)	0.0 (0.0)	744 (744)	2507 (2507)	0	0	0
i_nlToMacMux	9.3 (9.3)	0.0 (0.0)	13 (13)	3 (3)	0	0	0
g0.i_tcpTxMux	1.7 (1.7)	0.0 (0.0)	3 (3)	3 (3)	0	0	0
gen_WithUdp.i_udp	485.7 (0.8)	0.0 (0.0)	626 (0)	888 (2)	278560	18	0
gen_tcpConnections[0].i_tcp	8480.2 (35.9)	60.0 (0.0)	11408 (12)	7936 (87)	593568	45	1
gen_tcpConnections[1].i_tcp	8460.5 (40.8)	60.0 (0.0)	11357 (12)	7600 (87)	593568	45	1
gen_tcpConnections[2].i_tcp	8497.9 (34.8)	60.0 (0.0)	11376 (12)	7801 (86)	593568	45	1
iBusScheduler8	86.2 (86.2)	0.0 (0.0)	136 (136)	55 (55)	0	0	0
i_internetLayer	2741.3 (6.2)	20.0 (0.0)	3560 (10)	2657 (5)	17920	6	0
i_networkLayer	1851.8 (0.0)	0.0 (0.0)	2173 (0)	2071 (0)	35328	4	0
i_rxLinkResetSync	1.0 (1.0)	0.0 (0.0)	0 (0)	2 (2)	0	0	0
i_txLinkResetSync	1.0 (1.0)	0.0 (0.0)	0 (0)	2 (2)	0	0	0

Resource Estimates for MicroSemi Polarfire

The following table shows resources synthesized for MicroSemi PolarFire MPF300TS-1FCG1152I using Libero 2021.1 - instantiating the following design features:

- Ethernet block
- IPv4 block
- UDP block
- 3 instances of TCP blocks

Instance	Fabric 4LUT	Fabric DFF	Interface 4LUT	Interface DFF	uSRAM 1K	LSRAM 18K	Math 18x18	Chip Global
npap_tcp_udp_wrapper_u0	60339	31116	6792	6792	122	146	2	12
npap_tcp_udp_top_u0	60339	31116	6792	6792	122	146	2	12
Primitives	13	1	0	0	0	0	0	0
interface_adapter (all)	57	1	0	0	0	0	0	0
wrapper_ll_ip_tcp_u0	60269	31115	6792	6792	122	146	2	12
Primitives	349	195	0	0	0	0	0	0
i_netLayerConv	249	206	0	0	0	0	0	0
i_tcpTxMux	123	2	0	0	0	0	0	0
gen_WithUdp.i_udp	5598	3997	1548	0	24	35	0	0
gen_tcpConnections[0].i	15560	7037	1740	1740	31	37	0	3
gen_tcpConnections[1].i	13287	6649	1068	1068	26	19	2	3
gen_tcpConnections[2].i	15707	7112	1752	1752	35	37	0	3
iBusScheduler8	106	34	0	0	0	0	0	0
i_internetLayer	3717	2995	216	216	6	4	0	1
i_networkLayer	5306	2665	504	504	0	14	0	0

Detailed protocol support according RFC1122 (excerpt)

Ethernet Layer

Feature	Section	Must	Must not	Implemented
Send Trailers by default without negotiation	2.3.1		x	x
ARP	2.3.2			
Flush out-of-date ARP cache entries	2.3.2.1	x		(x)
Prevent ARP floods	2.3.2.1	x		(x)
Ethernet and IEEE 802 Encapsulation	2.3.3			
Host able to:	2.3.3			
Send & receive RFC-894 encapsulation	2.3.3	x		x
Send K1=6 encapsulation	2.3.3		x	
Use ARP on Ethernet and IEEE 802 nets	2.3.3	x		x
Link layer report b'casts to IP layer	2.4	x		
IP layer pass TOS to link layer	2.4	x		
No ARP cache entry treated as Dest. Unreach.	2.4		x	x

IP & ICMP Layer

Feature	Section	Must	Must not	Implemented
Implement IP and ICMP	3.1	x		x
Handle remote multihoming in application layer	3.1	x		x
Meet gateway specs if forward datagrams	3.1	x		-
Silently discard Version != 4	3.2.1.1	x		x
Verify IP checksum, silently discard bad dgram	3.2.1.2	x		x
Addressing:				
Subnet addressing (RFC-950)	3.2.1.3	x		-
Src address must be host's own IP address	3.2.1.3	x		x
Silently discard datagram with bad dest addr	3.2.1.3	x		x
Silently discard datagram with bad src addr	3.2.1.3	x		x
Support reassembly	3.2.1.4	x		-
TOS:				
Allow transport layer to set TOS	3.2.1.6	x		-

TTL:				
Send packet with TTL of 0	3.2.1.7		x	x
Discard received packets with TTL > 2	3.2.1.7		x	-
Allow transport layer to set TTL	3.2.1.7	x		-
Fixed TTL is configurable	3.2.1.7	x		x
IP Options:				
Allow transport layer to send IP options	3.2.1.8	x		-
Pass all IP options rcvd to higher layer	3.2.1.8	x		-
IP layer silently ignore unknown options	3.2.1.8	x		x
Silently ignore Stream Identifier option	3.2.1.8b	x		x
Source Route Option:				
Originate & terminate Source Route options	3.2.1.8c	x		-
Datagram with completed SR passed up to TL	3.2.1.8c	x		-
Build correct (non-redundant) return route	3.2.1.8c	x		-
Send multiple SR options in one header	3.2.1.8c	x		-
ROUTING OUTBOUND DATAGRAMS:				
Use address mask in local/remote decision	3.3.1.1	x		x
Operate with no gateways on conn network	3.3.1.1	x		x
Maintain "route cache" of next-hop gateways	3.3.1.2	x		-
If no cache entry, use default gateway	3.3.1.2	x		x
Support multiple default gateways	3.3.1.2	x		-
Able to detect failure of next-hop gateway	3.3.1.4	x		-
Ping gateways continuously	3.3.1.4		x	-
Ping only when traffic being sent	3.4.1.4	x		-
Ping only when no positive indication	3.3.1.4	x		-
Switch from failed default g'way to another	3.3.1.5	x		-
Manual method of entering config info	3.3.1.6	x		-
REASSEMBLY and FRAGMENTATION:				
Able to reassemble incoming datagrams	3.3.2	x		-
Transport layer able to learn MMS_R	3.3.2	x		-
Send ICMP Time Exceeded on reassembly timeout	3.3.2	x		-
Pass MMS_S to higher layers	3.3.3	x		-
MULTIHOMING:				
Allow application to choose local IP addr	3.3.4.2	x		x
BROADCAST:				
Broadcast addr as IP source addr	3.2.1.3		x	-
Recognize all broadcast address formats	3.3.6	x		-
Use IP b'cast/m'cast addr in link-layer b'cast	3.3.6	x		-
INTERFACE:				

Allow transport layer to use all IP mechanisms	3.4	x		-
Pass interface ident up to transport layer	3.4	x		-
Pass all IP options up to transport layer	3.4	x		-
Transport layer can send certain ICMP messages	3.4	x		-
Pass spec'd ICMP messages up to transp. layer	3.4	x		-
Include IP hdr+8 octets or more from orig.	3.4	x		-
ICMP:				
Echo server	3.2.2.6	x		-
Echo client	3.2.2.6	x		x
Use specific-dest addr as Echo Reply src	3.2.2.6	x		x
Send same data in Echo Reply	3.2.2.6	x		x
Pass Echo Reply to higher layer	3.2.2.6	x		x
Reverse and reflect Source Route option	3.2.2.6	x		-
Use IP b'cast/m'cast addr in link-layer b'cast	3.3.6	x		-

TCP Layer

Feature	Section	Must	Must not	Implemented
Push flag				
ESEND call can specify PUSH	4.2.2.2			-
If cannot: sender buffer indefinitely	4.2.2.2		x	
If cannot: PSH last segment	4.2.2.2	x		x
Window				
Treat as unsigned number	4.2.2.3	x		x
Robust against shrinking window	4.2.2.16	x		-
Sender probe zero window	4.2.2.17	x		(x)
Allow window stay zero indefinitely	4.2.2.17	x		x
Sender timeout OK conn with zero wind	4.2.2.17		x	x
TCP Options				
Receive TCP option in any segment	4.2.2.5	x		x
Ignore unsupported options	4.2.2.5	x		x
Cope with illegal option length	4.2.2.5	x		-
Implement sending & receiving MSS option	4.2.2.6	x		x
Send-MSS default is 536	4.2.2.6	x		x
Calculate effective send seg size	4.2.2.6	x		x

TCP Checksums				
Sender compute checksum	4.2.2.7	x		x
Receiver check checksum	4.2.2.7	x		x
Use clock-driven ISN selection	4.2.2.9	x		x
Opening Connections				
Support simultaneous open attempts	4.2.2.10	x		-
SYN-RCVD remembers last state	4.2.2.11	x		-
Passive Open call interfere with others	4.2.2.18		x	-
Function: simultan. LISTENs for same port	4.2.2.18	x		
Ask IP for src address for SYN if necc.	4.2.3.7	x		x
Otherwise, use local addr of conn.	4.2.3.7	x		x
OPEN to broadcast/multicast IP Address	4.2.3.14		x	-
Silently discard seg to bcast/mcast addr	4.2.3.14	x		-
Closing Connections				
Inform application of aborted conn	4.2.2.13	x		x
In TIME-WAIT state for 2 x MSL seconds	4.2.2.13	x		x
Retransmissions				-
Jacobson Slow Start algorithm	4.2.2.15	x		-
Jacobson Congestion-Avoidance algorithm	4.2.2.15	x		-
Karn's algorithm	4.2.3.1	x		-
Jacobson's RTO estimation alg.	4.2.3.1	x		-
Exponential backoff	4.2.3.1			
Generating ACK's:				
Process all Q'd before send ACK	4.2.2.20	x		x
Receiver SWS-Avoidance Algorithm	4.2.3.3	x		-

Changelog

The following lists MLE's engineering changelog for NPAP:

- 1.8.0 (20220331)
  - ETHERNET
    - #4606 - add padding for frames smaller than 60 Bytes
  - TCP
    - #4609 - remove maximum TCP Session limit
  - UDP
    - #4557 - fix length assignment in UDP Interface Adapter
- 1.7.1 (20211220)
  - GENERAL
    - #3951 - add missing reset signal
    - #4412 - fix buffer generic ranges
    - #4416 - fix subnet mask assignment for no UDP setup
  - TCP
    - #4367 - fix tcp space available calculation for buffer sizes above 64KB
- 1.7.0 (20211101)
  - GENERAL
    - #3951 - remove unused altera_attribute
    - #4139 - allow set off RTO values in TCP session establish state
- 1.6.2 (20210913)
  - GENERAL
    - #3981 - remove unused altera_attribute
- 1.6.1 (20210801)
  - UDP
    - #3969 - fix use of ceil function
- 1.6.0 (20210701)
  - GENERAL
    - #3916 - rename / fix generic names
- 1.5.3 (20210301)
  - UDP
    - #3549 - fix bus scheduler bus handling
- 1.5.2 (20201101)
  - TCP
    - #3091 - fix ack command fifo interface handling
- 1.5.1 (20201001)
  - ARP
    - #2924 - fix handshake between ARP and bus scheduler
  - TCP

        - #2539 - fix possible wrong MAC usage in multi session configuration
        - #2923 - fix data acceptance criteria from application
- 1.5.0 (20200901)
    - GENERAL
        - #2254 - change 100ms clock not generated inside NPAP wrapper
        - #2848 - change application clocking by removing TCP and UDP application clock
    - IP
        - #2239 - change default configuration for IP filter, now enabled
    - DHCP
        - #2858 - change increase DHCP usability by adding try counter, valid signal and new timeout behaviour
        - #2737 - fix DHCP lease calculation
    - TCP
        - #2703 - fix payload length update behaviour in transmit controller
        - #2727 - fix acknowledgment number update behaviour in transmit controller
        - #2831 - fix FSM handshake in transmit controller which could lead to TCP session freeze
- 1.4.9 (20200702)
    - IGMP
        - #2704 - change disable IGMP per default
        - #2496 - fix compiler warning about latch implementation
    - TCP
        - #2706 - change TCP tx data path in asynchronous mode
        - #2627, #2688, #2693, #2701, #2702 - fix TCP tx splitter, rework after multiple bugs
        - #2692 - fix used TCP tx splitter generic
        - #2711 - fix transmit controller fsm reset generation
- 1.4.8 (20200430)
    - TCP
        - #2477 - fix TCP retransmission buffer delete handling
- 1.4.7 (20200331)
    - TCP
        - #2180 - change TCP tx splitter to work with byte granularity
        - #2097 - fix TCP multi session reset synchronization
        - #2339 - fix TCP application reset clock domain crossing
- 1.4.6 (20200303)
    - GENERAL
        - #2469 - fix default gateway IP address usage
    - TCP

        - #2468 - add register stage to TCP TX application interface to ease timing on Virtex 6
- 1.4.5 (20200220)
    - GENERAL
        - #2449 - fix Xilinx ISE 14.7 workflow
- 1.4.4 (20200213)
    - TCP
        - #2086 - fix retransmission lockup
        - #2112 - fix fsm lockup in transmit controller
- 1.4.3 (20200116)
    - ARP
        - #2179 - fix ARP cache ip address lookup
- 1.4.2 (20200113)
    - GENERAL
        - #2294 - change delivered IP XACT constraint file
- 1.4.1 (20200107)
    - TCP
        - fix data type and IP core GUI handling of TCP sequence number initialization
- 1.4.0 (20191126)
    - GENERAL
        - #1930 - add AXI4-Stream TCP application interface
        - #2151 - add customized block design configuration gui
        - #2166 - add example constrain file to IP XACT packaging
        - #2148 - fix block design gui NPAP name generic
    - TCP
        - #1939 - fix space available calculation which lead to duplicated data beat
        - #2181 - fix TCP tx splitter timeout
- 1.3.0 (20191002)
    - GENERAL
        - #1682 - add IP XACT TCP/UDP wrapper and packaging
- 1.2.0 (20190920)
    - GENERAL
        - #2075 - add packaging infrastructure for TCP source code release
- 1.1.0 (20190705)
    - ARP
        - #1719 - add new ARP cache size generic
        - #1698 - fix internal race condition during initialization
    - UDP
        - #1720 - fix retry mechanism on failed ARP lookup

- 1.0.0 (20181023)
  - GENERAL
    - #1320 - add NPAP to MLE Vivado build toolchain
    - #1326 - add IP XACT UDP wrapper and packaging
    - #1133 - fix bus scheduler grant timeout
  - ETHERNET
    - #1323 - add 64 and 128 bit AXI4-Stream interface option
  - IP
    - #1328 - fix IP header decoder data valid calculation for payloads of 1 to 3 byte
  - UDP
    - #1322 - add AXI4-Stream UDP application interface
    - #1324 - add new generic to disable UDP TX aligner
    - #1132 - fix UDP tx fifo write count calculation
    - #1133 - fix UDP header encoder fsm timeout for ARP
    - #1327 - fix UDP throughput bottleneck for payload sizes less than 100 byte
    - #1330 - fix corrupt UDP data multiplexing for zero TCP connections

Contact Info

Missing Link Electronics, Inc.
2880 Zanker Road, Suite 203
San Jose, CA 95134
Phone: +1-408-475-1490

Missing Link Electronics GmbH
Industriestraße 10
89231 Neu-Ulm
Tel. +49 731 141149-0

Email: sales-web@mlecorp.com

http://www.missinglinkelectronics.com

Fraunhofer HHI

Founded in 1949, the German Fraunhofer-Gesellschaft undertakes applied research of direct utility to private and public enterprise and of wide benefit to society. With a workforce of over 23,000, the Fraunhofer-Gesellschaft is Europe's biggest organization for applied research, and currently operates a total of 67 institutes and research units. The organization's core task is to carry out research of practical utility in close cooperation with its customers from industry and the public sector.

Fraunhofer HHI was founded in 1928 as "Heinrich-Hertz-Institut für Schwingungsforschung" and joined in 2003 the Fraunhofer-Gesellschaft as the "Fraunhofer Institute for Telecommunications, Heinrich-Hertz-Institut". Today it is the leading research institute for networking and telecommunications technology, "Driving the Gigabit Society" .