

# FPGAs Help Characterize Massive-MIMO Channels

**by Patrick Murphy**

President  
Mango Communications  
[patrick@mangocomm.com](mailto:patrick@mangocomm.com)

**Clayton Shepard**

Graduate Student  
Rice University  
[cws@rice.edu](mailto:cws@rice.edu)

**Lin Zhong**

Associate Professor  
Rice University  
[lzhong@rice.edu](mailto:lzhong@rice.edu)

**Chris Dick**

Chief DSP Scientist  
Xilinx, Inc.  
[chris.dick@xilinx.com](mailto:chris.dick@xilinx.com)

**Ashutosh Sabharwal**

Professor  
Rice University  
[ashu@rice.edu](mailto:ashu@rice.edu)

A system built from 24 FPGAs, 96 antennas plus a custom 802.11 implementation enables the real-time study of multiuser-MIMO propagation environments.

**M**ultiuser MIMO (MU-MIMO) is a wireless communications technique that leverages multiple antennas at infrastructure nodes, like basestations and access points, to serve many clients simultaneously. MU-MIMO is an integral part of upcoming wireless standards and is expected to deliver significant improvements in the performance of busy networks.

It is envisioned that with each new generation of wireless systems, the number of antennas at basestations will continue to grow, eventually leading to “massive MIMO” systems. The massive-MIMO approach extends the number of antennas at MU-MIMO basestations to tens or hundreds, seeking to improve performance while possibly simplifying the basestation’s signal processing. One scalable massive-MIMO technique is known as conjugate beamforming [1]. An early implementation of this strategy shows the potential for real-world gains [2].

Multiuser-MIMO techniques rely on accurate knowledge of the wireless propagation environment. An MU-MIMO infrastructure node can serve multiple users simultaneously only if it has accurate and recent measurements of the wireless channel to each user. Gathering this channel information in real time is challenging, and the performance impact of stale or inaccurate channel information can be severe.

We have devised an integrated system for massive-MIMO channel characterization that enables researchers to study the dynamics of channels in real time. The system uses the Xilinx® FPGA-based WARP hardware platform and [Mango Communications’ 802.11 Reference Design](#) at its core, scaled up to 24 FPGAs connected to 96 antennas using the Rice University Argos platform [2]. A custom Python framework developed by Mango Communications controls and gathers data from every node in the array in real time. This combination of Mango- and Rice-developed tools provides deep visibility into the wireless stack, including the raw channel data necessary for characterizing massive MIMO.

A key feature of the custom 802.11 implementation by Mango Communications is its ability to stream low-level baseband parameters, like AGC gains, channel estimates and raw packet contents (even packets with errors), from all receiving antennas in real time. This feature of the reference design allows the Rice Argos array to act as a standards-compliant 802.11 access point (AP), serving Internet to commercial Wi-Fi devices (for example, smartphones, tablets or laptops) while simultaneously gathering channel data between the array antennas and each client, all in real time. The Xilinx FPGAs are crucial to enabling real-time

# Our massive-MIMO system, Argos, uses FPGAs to locally process the data in real time and significantly reduce the burden on the upstream processors.

processing at each antenna. They condense the data gathered from each antenna into per-client channel characteristics that a custom application can stream and analyze.

Let's take a closer look at the WARP hardware platform and the custom 802.11 implementation by Mango, as well as the conjugate beamforming strategy for MU-MIMO. Finally, we will also examine the characterization process, including the real-time collection of wireless channel measurements from Wi-Fi clients and the processing of channel data to estimate achievable MU-MIMO performance.

## SYSTEM COMPONENTS

The Wireless Open-Access Research Platform (WARP) is a scalable and extensible programmable wireless platform, built from the ground up to prototype advanced wireless networks. WARP combines high-performance programmable hardware with an open-source repository of reference designs and support materials.

The WARP Project, founded in 2006 by Rice University professor Ashu Sabharwal, was originally funded by the National Science Foundation with ongoing support from Xilinx. The project has since grown into a self-sustaining

open-source effort with users around the world. Mango Communications spun off from the Rice University WARP Project in 2008 with the initial goal of manufacturing and distributing the Rice WARP hardware. In 2012 Mango released the entirely redesigned WARP v3 hardware. Today, Mango engineers are the most active contributors to the WARP repository and forums, providing ongoing development and support of open-source WARP designs.

A central component in our system for measuring massive-MIMO channels is Mango Communications' WARP v3 hardware platform. WARP v3 is designed for rapid, real-time prototyping of novel wireless designs. The hardware integrates a high-performance Xilinx Virtex®-6 FPGA, two flexible RF interfaces and multiple peripherals, including DDR3 DRAM and two 1-Gbps Ethernet interfaces. The WARP v3 board can be extended to four RF interfaces with Mango's dual-radio FMC module. This hardware configuration, shown in Figure 1, provides four fully programmable RF interfaces with independent digital baseband connections to the FPGA.

To study massive-MIMO systems, it's necessary to co-locate multiple WARP v3 nodes with shared power, clocking and Ethernet connectivity. This requirement has been addressed by the [Rice University Argos project](#). The Argos v2 array is a collection of 24 four-antenna WARP v3 nodes, pictured in Figure 2. The Argos array is designed to support a wide variety of massive-MIMO experiments and is perfectly suited to gather-



Figure 1 – The WARP v3 hardware with dual-radio FMC module provides a large FPGA, four RF interfaces, memory and two Ethernet connections.



Figure 2 – The Rice University Argos v2 array combines 24 quad-radio WARP v3 nodes with shared clocking and Ethernet connectivity.

ing channel measurements simultaneously across all 96 array antennas.

The FPGA on each WARP v3 node in the Argos array provides significant processing power close to the RF interfaces. In a massive-MIMO configuration like Argos, there is a tremendous amount of data to process. For example, when receiving 40 MHz of band-

width, each RF interface on WARP v3 generates a 960-Mbps sample stream (dual 12-bit 40-Msample/second ADCs). The full Argos array generates 96 times this amount—far more than can be streamed to a PC and processed in real time. Instead, the system uses FPGAs to locally process the data in real time and significantly reduce the burden on

the upstream processors. For our massive-MIMO channel-characterization design, this real-time processing is very important as it allows the design to continually measure channels and reliably observe submillisecond variations in channel characteristics. The custom FPGA design that performs this processing is the Mango Communications 802.11 Reference Design for WARP v3.

This reference design is a real-time FPGA implementation of the 802.11a/g medium-access control (MAC) layer and physical (PHY) layer. The design can interoperate with standard Wi-Fi devices, acting as an AP (serving Wi-Fi clients), a client (accessing a Wi-Fi AP) or a monitor (a receive-only passive observer of network activity). You can customize both the MAC and the PHY to explore new variations to the standard. This combination of interoperability and extensibility enables a wide range of wireless communications and networking experiments. The full source for the 802.11 reference design is available to users of WARP v3 hardware at no cost.

Figure 3 illustrates the reference design architecture. The design uses two Xilinx MicroBlaze™ cores to implement the high- and low-level MAC protocol in software. The MAC connects to two FPGA cores, which implement the PHY transmitter and receiver. We implemented these PHY cores in Xilinx System Generator. The transmitter core implements a complete bytes-to-waveform pipeline. It reads a packet payload from the MAC, creates the OFDM waveform and drives the waveform to the RF interface DACs. This pipeline includes encoding, scrambling, interleaving, IFFT and preamble insertion. The MAC specifies the modulation and coding rates per packet; all eight data rates specified in 802.11a/g are supported.

The receiver design implements the full waveform-to-bytes pipeline, including AGC, synchronization, FFT, channel estimation, equalization, detection and decoding. The receiver configures its

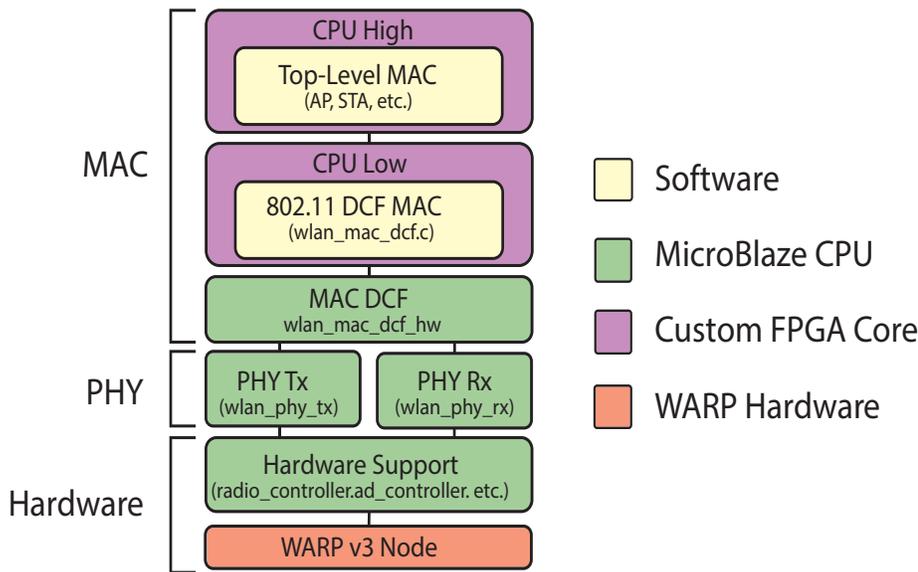


Figure 3 – The Mango 802.11 Reference Design architecture includes two Xilinx MicroBlaze CPUs for the MAC and custom System Generator cores for the PHY transmitter and receiver.

demodulation and decoding blocks automatically per packet using the RATE value in the packet's SIGNAL field. The receiver decodes packets of any rate fast enough to meet the standard's strict Rx-to-Tx turnaround requirements for transmitting an acknowledgement (ACK) in response to a reception.

One feature of our receiver design that is central to characterizing massive-MIMO channels is the channel-estimation subsystem. In a standard OFDM receiver, the channel estimator produces a complex channel coefficient per subcarrier. The equalizer uses these estimated coefficients to correct the channel's amplitude and phase degradations for each received data symbol. Our design additionally saves a copy of the channel estimates for every received packet to an on-chip memory area. The MAC treats these channel estimates as extra metadata about the received frame, along with standard information like Rx power, AGC gain selections, checksum status and antenna selection. The channel estimates are then copied to the higher-level MAC for further processing. Our

characterization platform gathers these estimates from every packet received by every node in the Argos array to assemble its real-time view of the massive-MIMO propagation environment.

### WARPNET EXPERIMENTAL FRAMEWORK

The final piece for our massive-MIMO characterization system is a framework, dubbed WARPnet, for running experiments with large networks of WARP nodes. WARPnet is a custom Python package that uses a dedicated control connection to multiple WARP nodes. The framework allows a Python script running on a PC to remotely configure experiment parameters and retrieve experimental data, all in real time. WARPnet interacts with the Mango 802.11 Reference Design via the secondary Ethernet connection on each WARP v3 board. The upper MicroBlaze device processes the WARPnet commands, giving the framework direct access to the node's high-level MAC state and all data passed up from the low-level MAC and PHY.

For our massive-MIMO channel-characterization design, the WARPnet framework maintains a connection to every node in the Argos array. Each node is configured as an 802.11 monitor, capturing channel estimates from every received packet and offloading these packets via Ethernet for further analysis.

The full Python source code for WARPnet is open-source in the WARP repository.

### UNDERSTANDING MULTIUSER MIMO

Basestations employing multiuser-MIMO techniques seek to create waveforms for many transmit antennas which, when combined by the wireless channel, deliver data to multiple users simultaneously. Creating multiuser waveforms requires sophisticated processing at the basestation. Many MU-MIMO techniques have been proposed. A common requirement of MU-MIMO designs is accurate knowledge of the wireless propagation characteristics from each basestation antenna to each client device.

One approach to MU-MIMO is known as zero-forcing, which in theory—and recently in practice [3]—has been shown to achieve significant performance gains over single-user techniques. The zero-forcing approach aims to maximize the signal-to-interference-plus-noise ratio (SINR) at each client's receive antenna. Maximizing SINR requires maximizing the power of the signal representing a user's payload (the "S") while minimizing the power of every other user's payload (the "I") in the waveform that arrives at the user's antenna. Zero-forcing requires very sophisticated processing at the basestation. With zero-forcing, the calculation of the transmit waveform for a given basestation antenna requires knowledge of the payload for every user and the wireless channel from every other antenna to every user. The computational complexity of this calculation increases significantly as the number of basestation antennas increases.

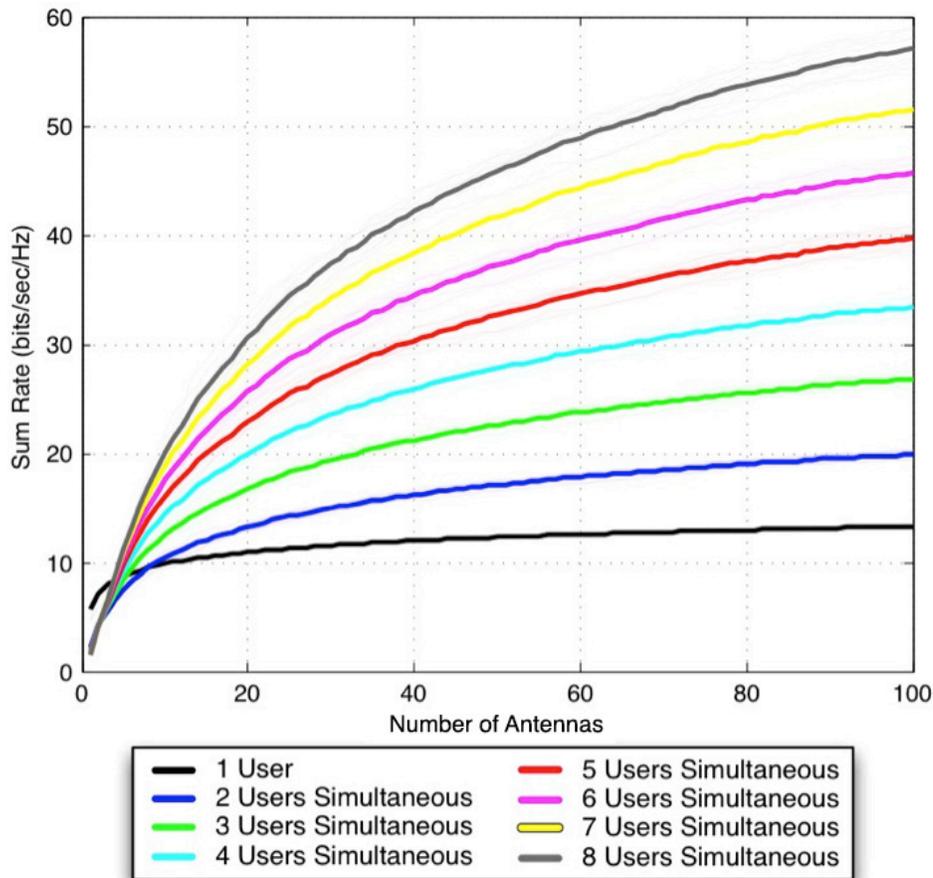


Figure 4 – Simulation of a multiuser-MIMO network shows significant rate improvements for many users when sufficient antennas are installed at the access point.

Conjugate beamforming [1] is an alternative MU-MIMO technique. In this approach, the basestation seeks to maximize the good signal power delivered to each client device without actively minimizing the interference power. In theory, the conjugate-beamforming approach of increasing SINR at each user by maximizing signal power (the “S” in SINR) while ignoring interference power (the “I” in SINR) improves with an increasing number of antennas. Further, the conjugate-beamforming calculation of each transmit antenna’s waveform does not require knowledge of any other antenna’s channel characteristics. Together these factors make conjugate beamforming well suited to massive-MIMO systems, where a basestation has many more antennas than users.

Consider the classic expression for Shannon channel capacity,  $C = \log(1+\text{SINR})$ . The capacity of the wire-

less channel, in bits/second/Hz, grows logarithmically with the SINR. Conjugate multiuser beamforming has two competing effects when the system adds more users and antennas. First, the presence of multiple antennas allows for an increase in received signal power, since each antenna can rotate its phase such that transmissions are constructively combined at the users’ receivers. Second, the presence of multiple transmissions to independent users creates an increase in interference power. The superimposed interference signals combine randomly. As the number of antennas increases, the growth in constructively combined signal power outruns the randomly combined interference power, increasing overall SINR.

The simulation results in Figure 4 illustrate the impact of increasing the number of basestation antennas on the overall network capacity when con-

jugate beamforming is employed. The simulation assumes a network with one basestation and eight users, with wireless channels modeled by independent and identically distributed Rayleigh fading. The simulation shows the overall network rate when one to eight users are served simultaneously vs. the number of antennas employed by the basestation. For a small number of antennas, we can see that conjugate beamforming to more than one user at a time is not beneficial. If basestations are limited to just a few antennas, traditional single-user beamforming with time-sharing may be superior to multiuser conjugate beamforming. As the number of antennas increases, more users can be supported, for substantial overall network rate improvements.

This simulation uses idealistic channel models to show that multiuser conjugate beamforming might be able to achieve performance gains. Whether these gains can be realized in a real system depends on the actual wireless channels between the basestation and client devices. Our MU-MIMO channel-characterization platform can measure the channels from the basestation to actual user devices in real time, providing a powerful tool to evaluate the real-world performance of MU-MIMO techniques.

## PUTTING IT ALL TOGETHER

Now that we’ve explored the motivations for measuring massive-MIMO channels and the tools provided by the Rice Argos array, WARP hardware and Mango 802.11 Reference Design, let’s take a look at how to assemble these elements into a complete real-time, massive-MIMO channel-characterization platform.

The 24 WARP v3 nodes in the Argos array are configured with a custom version of the Mango 802.11 Reference Design. This version operates in receive-only monitor mode, attempting to receive packets on all four of the node’s antennas. For every packet reception, the node estimates the complex chan-

nel coefficient for each subcarrier, decodes the packet and sends the packet header and channel estimates via Ethernet for analysis. This processing flow is implemented in all 24 nodes in the array, with all nodes operating in parallel.

To communicate with standard Wi-Fi devices, the channel-measurement platform must also implement a standard 802.11 access point. Another WARP v3 node is used for this purpose, running the Mango 802.11 Reference Design in AP mode. This AP node serves as the 25th node in the Argos array. The AP advertises an open Wi-Fi network, accepts associations from commercial Wi-Fi devices and serves Internet access via its primary Ethernet connection.

This is the standard behavior of the AP profile in the Mango 802.11 Reference Design. To enable real-time channel measurements, this AP implements one additional function. Using the secondary Ethernet connection on the WARP v3 board, the AP node sends an Ethernet packet every time a Wi-Fi client joins or leaves the wireless network. The channel analysis application (discussed below) uses these association updates to maintain its local table of active clients.

## CLIENT TRANSMISSIONS

One key challenge in gathering channel estimates from packets received from commercial Wi-Fi devices is ensuring the devices transmit often enough. Modern Wi-Fi devices often employ aggressive power-saving schemes, disabling their Wi-Fi radios when no application is requesting network access. The devices will periodically check in with the AP but possibly not frequently enough to ensure up-to-date channel estimates at the array.

We address the issue of infrequent client transmissions with two “hacks.” First, we modify the traffic indication map (TIM) field in beacons transmitted by the platform AP to inform all connected clients that new data packets are queued for them. The TIM field is normally used to support power saving at clients, allowing clients to briefly wake up in receive-only mode to receive the beacon, decode the TIM and resume low-power mode if no traffic is waiting. By listing every node in every beacon’s TIM field, nodes will sleep less often.

The second technique solicits client transmissions by using the ACK packets the client devices have transmitted. The array can extract channel esti-

mates from any packet a client transmits, including short ACKs. However, the 802.11 ACK packet only includes a destination MAC address, typically preventing the array from identifying the transmitting client.

We work around this issue by exploiting a quirk in the 802.11 MAC specification. The standard requires 802.11 devices to send positive acknowledgment packets after successfully receiving a unicast packet addressed to the client. The “must ACK” requirement applies even if the packet’s source address is not recognized. Thus, to trigger an ACK transmission by a client that contains an identifier unique to the client, the AP sends a data packet with a bogus (but unique) source address. Upon receipt, the client then transmits the ACK to the bogus (but unique) address the AP used. The array nodes receive this ACK and can unambiguously associate the resulting channel estimates with the transmitting client. This trick works remarkably well for triggering frequent updates of channel estimates at the array, and is only possible because of the full programmability of the Mango 802.11 Reference Design.

## REAL-TIME ANALYSIS

The final component of the massive-MIMO channel-measurement platform is a custom application that gathers the array-channel estimates, computes achievable multiuser capacities and displays the results in real time. We developed this application in Objective-C, using native UDP sockets to interface with the WARP v3 nodes in the array and OS X graphics frameworks for plotting results.

The application has two primary views. The first displays the raw channel magnitudes that each array antenna gathers for every subcarrier, 4,992 data points in all (52 subcarriers x 96 array antennas). This view is a raw display of the channel data gathered by the array and serves primarily to convey the wide range of channel values observed by individual array antennas.



Figure 5 – Channel-magnitude view in our custom MU-MIMO channel-analysis application. Each bar plot shows magnitudes per antenna, per subcarrier.

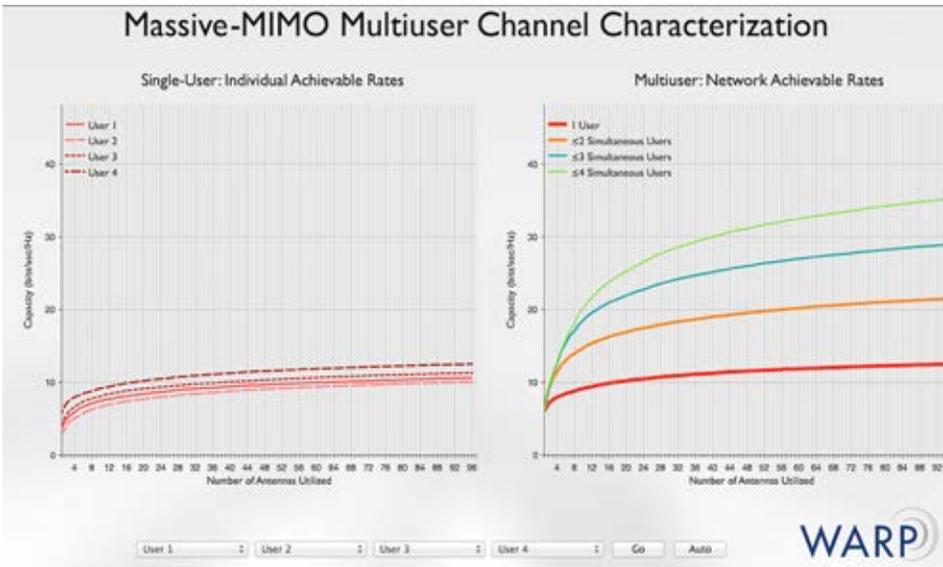


Figure 6 – Estimated network capacity for single-user and multiuser techniques were calculated from real MU-MIMO channel measurements the array had gathered, as displayed by our custom analysis application.

A screen shot of this view is shown in Figure 5. In practice, the view updates in real time (10 frames per second with active Wi-Fi clients).

The application’s second view displays results of capacity calculations based on the array’s channel estimates. This view is illustrated in Figure 6. Two capacity calculations are performed. The first plots the capacity to each user vs. the number of array antennas used. Each line on this plot approximates the achievable downlink capacity to a single user assuming the array used a subset of its antennas in a traditional, single-user beamforming configuration. The decreasing slope of each capacity curve with increasing antennas clearly demonstrates the diminishing benefit of many antennas with traditional, single-user wireless techniques.

The second plot shows the total network capacity if the array implements downlink multiuser beamforming techniques using a subset of its antennas. The trends on the four plots clearly highlight the benefit of additional antennas when multiuser techniques are employed. The increasing slope when

additional users are served highlights the “outside the log” (often labeled “pre-log” in MIMO literature) gain in network capacity with multiuser beamforming.

We demonstrated the massive-MIMO channel-measurement platform at the 2014 Xilinx Emerging Technology Symposium (ETS) in February. Details of this demonstration, including videos and links to supplementary material, are available at <http://warpproject.org/XilinxETS>.

**REFERENCES**

1. T.L. Marzetta, “Noncooperative Cellular Wireless with Unlimited Numbers of Base Station Antennas,” *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3590–3600, 2010
2. C. Shepard, H. Yu, N. Anand, E. Li, T. Marzetta, R. Yang and L. Zhong, “Argos: Practical Many-Antenna Base Stations,” *Proceedings of ACM MobiCom*, pp. 53–64, 2012
3. Q. Yang, X. Li, H. Yao, J. Fang, K. Tan, W. Hu, J. Zhang and Y. Zhang, “Bigstation: Enabling Scalable Real-time Signal Processing in Large MU-MIMO Systems,” *Proceedings of ACM SIGCOMM*, pp. 399–410, 2013

## Everything FPGA.

**1. MARS ZX3**  
Zynq™-7020 SoC Module

- Xilinx® Zynq-7020 SoC FPGA
- Up to 1 GB DDR3L SDRAM
- 16 MB QSPI flash
- 512 MB NAND flash
- USB 2.0
- Gigabit Ethernet
- 85,120 LUT4-eq
- 108 user I/Os
- 3.3 V single supply
- 67.6 × 30 mm SO-DIMM

**VxWorks**

---

**2. MERCURY KX1**  
Kintex™-7 FPGA Module

- Xilinx® Kintex™-7 FPGA
- Up to 2 GB DDR3L SDRAM
- 16 MB QSPI flash
- PCIe® x4 endpoint
- 4 × 6.6/10.3125 Gbps MGT
- USB 3.0 Device
- 2 × Gigabit Ethernet
- Up to 406,720 LUT4-eq
- 178 user I/Os
- 5-15 V single supply
- 72 × 54 mm

---

**3. FPGA MANAGER**  
IP Solution

→ USB 3.0

→ PCIe® Gen2

→ Gigabit Ethernet

Streaming,  
made simple.

One tool for all FPGA communications. Stream data from FPGA to host over USB 3.0, PCIe®, or Gigabit Ethernet – all with one simple API.

---

**4. PROFINET IP Core**

- Optimized for Xilinx FPGA and Zynq SoC
- IRT cycle times as low as 31.25 µs
- Hardware IEEE 1588 PTP implementation
- Isochronous traffic can bypass the software stack
- Co-developed by ZHAW IneS and Enclustra

## Enclustra. We speak FPGA.

Design Center • FPGA Modules  
Base Boards • IP Cores

ENCLUSTRA

FPGA SOLUTIONS

Fourth Quarter 2014

Xcell Journal

25