Flexible Demonstrator Platform for Cooperative Joint Transmission and Detection in Next Generation Wireless MIMO-OFDM Networks

Andreas Ibing, David Kühling, Marcel Kuszak, Clemens v. Helmolt, Volker Jungnickel
Fraunhofer Institute for Telecommunications
Heinrich-Hertz-Institut
Einsteinufer 37, 10587 Berlin, Germany

ABSTRACT

This paper presents a flexible demonstrator platform for cooperative joint transmission and detection in next generation wireless networks, especially cellular networks like 3GPP LTE+. The platform is shown capable of handling computational load as well as network load of jointly processing three maximum bandwidth LTE sectors with four antennas each. The platform uses a Workstation mainboard as PCI Express backplane. Joint signal processing is done on a Cell processor, protocol handling on an IXP2350 network processor. For baseband sample exchange a 10GBit/s optical network interface card is used. First results show the Cell processor capable of computing the 12x12 modified LTE MIMO-OFDM processing concurrently for precoding (downlink) and postprocessing (uplink) with a MIMO matrix update interval of 1ms (high mobility).

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*

General Terms

Experimentation

Keywords

Testbed, cooperative MIMO-OFDM, Software defined radio

1. INTRODUCTION

Modern wireless communication systems like 3GPP LTE [1] use orthogonal multicarrier modulation to avoid intra-cell interference. The remaining factor limiting system capacity is inter-cell interference, against which base station cooperation has gained interest [17] [19]. The idea is to cancel intercell interference by treating the antennas of neighbouring base stations jointly as one large array. High-bandwidth low-delay connections between base stations are assumed. An architecture proposal for network-wide deployment with piecewise central processing is described in [7].

TRIDENTCOM 2008, 17th – 20th Mar 2008, Innsbruck, Austria.
 ISBN 978-963-9799-24-0
DOI 10.4108/tridentcom.2008.3127

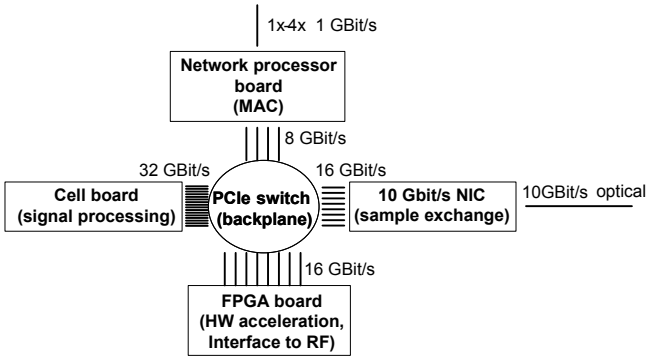

Figure 1: System overview with main components.

There are four main challenges in realizing such a system:

- Computational load of joint signal processing: grows cubically with the number of jointly processed antenna signals.
- Network load of baseband sample streaming between base stations.
- Network delay: for joint transmission it must be in the order of channel coherence time.
- Synchronization: coherent transmitters are necessary.

An FPGA-centric communication system demonstrator platform has been presented e.g. in [12], while a testbed based on the AdvancedTCA form factor is described in [11]. Our platform takes a different approach and uses off-the-shelf products based on the common PCI Express form factor, which reduces costs while nevertheless providing competitive high performance and upgradeability. Details of the implementation of MIMO-OFDM signal processing algorithms on the Cell processor are described in [14].

The next section gives an overview of the platform, section 3 describes the components in detail. Communication between platform components is depicted in section 4. First implementation results are given in section 5, before conclusions are drawn in section 6.

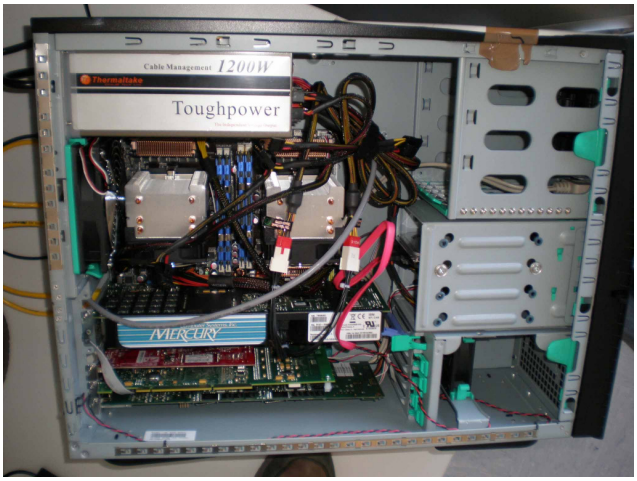

Figure 2: Photo of the platform.

2. PLATFORM OVERVIEW

A block diagram of the main components is shown in Fig. 1. A Workstation mainboard [4] is used as PCI Express backplane. PCI Express 2.0 is a (switched) serial bus with 2.5GBit/s per lane, offering a raw data rate of 2GBit/s bidirectional per lane [5]. Lanes can be aggregated for higher throughput. The four plug-in cards used are:

- Cell accelerator board [15]
- Network processor board (IXP2350) [10]
- 10GBit Ethernet network interface card [16]
- FPGA board (Virtex5) [18]

A base station should have only one cable connecting it to its router. The demonstrator platform uses two cables to be able to handle the high network load. A network processor like IXP28xx would be able to handle the complete load, but unfortunately it is currently not available on a PCI Express board.

2.1 Design decisions:

The choice of a standard bus enables usage of boards which are produced in higher volumes (and are therefore comparatively cheap) and makes the system upgradeable for future needs.

2.1.1 Comparison to FPGA-centric design

A base station demonstrator platform based on FPGAs (with DSPs for channel estimation and equalizer computation) has been presented e.g. in [12]. In our platform a PCI Express based system is chosen mainly for two reasons: first the high computational requirements of joint signal processing can be well met with the Cell processor, which is available as PCI Express coprocessor board. Second, protocol handling in FPGA is tedious and inflexible compared to software. Therefore network layer and Medium Access Control (MAC) layer protocol processing is done on a network processor, which is also available as PCI Express coprocessor board.

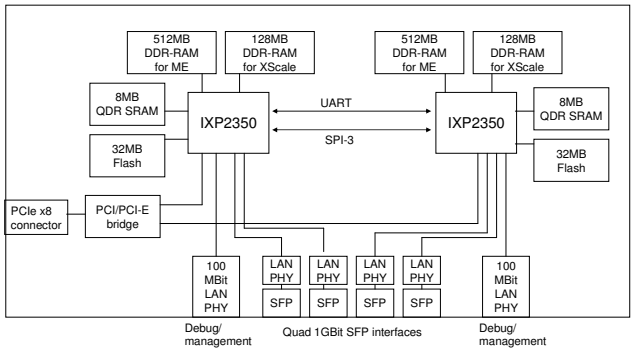

Figure 3: Block diagram of the network processor board [10].

2.1.2 Comparison to AdvancedTCA form factor

A testbed based on the AdvancedTCA (ATCA) form factor is presented e.g. in [11]. An ATCA backplane offers more slots and can be used with different fabrics like e.g. PCI Express, Ethernet or Infiniband. ATCA has not been chosen for our demonstrator platform because it would be considerably more expensive.

3. DETAILED DESCRIPTION OF PLATFORM HARDWARE COMPONENTS

3.1 Workstation mainboard as PCI Express backplane

The mainboard used is an Asus L1N64-SLI WS [4]. It has four highspeed PCIe slots: 2 times x16 (16 lanes, i.e. 32GBit/s bidirectional) and 2 times x8 (16GBit/s). In the demonstrator platform, the two host processors (dual core Athlon 64 FX each) on the mainboard are only used for management and to provide boot images from the local hard disc to the cell and network processors.

3.2 Protocol implementation on Network Processor

For protocol processing the 'Double Espresso' board [10] is used. A block diagram of the board is shown in Fig. 3. The board contains two IXP2350 network processors, each capa-

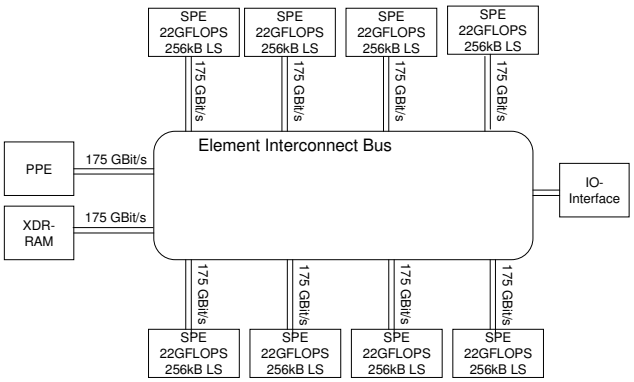

Figure 4: Basic structure and capabilities of a Cell CPU running at 2.8GHz [13].

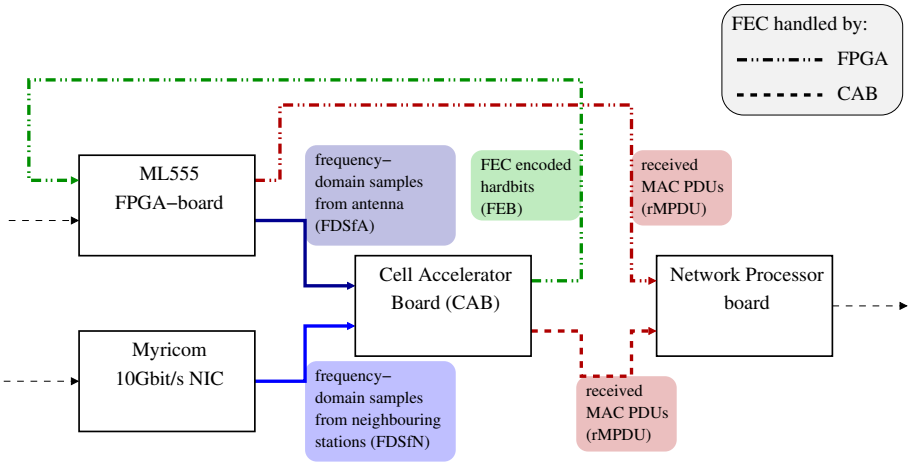

Figure 5: Dataflow through platform for receive.

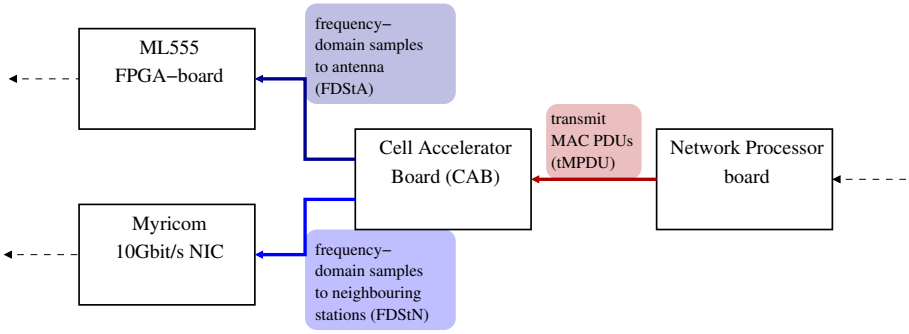

Figure 6: Dataflow through platform components for transmit.

Flow	Rationale	Throughput
FDStA	33.6 Msample/s per antenna, $2 \cdot 16$ bit/sample, 4 transmit antennas	4.3 Gbit/s
FDStN	same as FDStA but for 8 transmit antennas	8.6 Gbit/s
tMPDU	max. 3 bit/sample assuming QAM64 and code-rate 1/2	1.2 Gbit/s
FDSfA	see as FDStA	4.3 Gbit/s
FDSfN	see FDStN	8.6 Gbit/s
rMPDU	max. 3 bit/sample assuming QAM64 and code-rate 1/2	1.2 Gbit/s
FEB	max. 6 bit/sample assuming QAM64	2.42 Gbit/s

Table 1: Throughput of dataflow items in figures 5 and 6 for a 12×12 MIMO OFDM systems with LTE parameters and 1200 subcarriers, FEC decoding in Cell.

ble of processing 2.5GBit/s traffic. The board has an x4 PCI Express connector (8GBit/s) and four 1 GBit/s Ethernet interfaces (SFP). Each of the IXP2350 consists of a 900MHz XScale processor for the control plane and four 900MHz Microengines for data plane processing. The board is used for data exchange between the base station and its router (no baseband sample streaming). Typical functionality is packet classification, segmentation, reassembly, packing, addressing and queueing. On the XScale we run a Linux 2.4 kernel, development for the microengines is done with the free Intel IXA SDK.

3.3 Signal processing on Cell Processor

The Cell processor [8] is used for signal processing. It has a considerably higher performance (up to 210 GFLOP/S) than currently available DSPs. It consists of a general purpose power architecture core (PPE) and eight coprocessors (SPE) optimized for single-precision floating point SIMD processing. The Cell processor is easier to program and synchronize than a cluster of DSPs. It is supported by recent Linux kernels and the GNU C Compiler (gcc). A block diagram of the Cell CPU is shown in figure 4. The Cell accelerator board has an x16 PCIe connector (32GBit/s), the processor is running at 2.8GHz.

3.4 10GBit Ethernet inter-base-station link for baseband sample exchange

Joint signal processing means exchanging baseband samples (complex numbers) between base stations. Assume that network protocol overhead is neglected and that samples are exchanged with 12bit inphase and 12bit quadrature information. The LTE system uses 1200 subcarriers and a length-2048 FFT. The data rate for one antenna in time domain is then 740MBit/s, in frequency domain (without guard band) 430MBit/s. The network load increases linearly with the

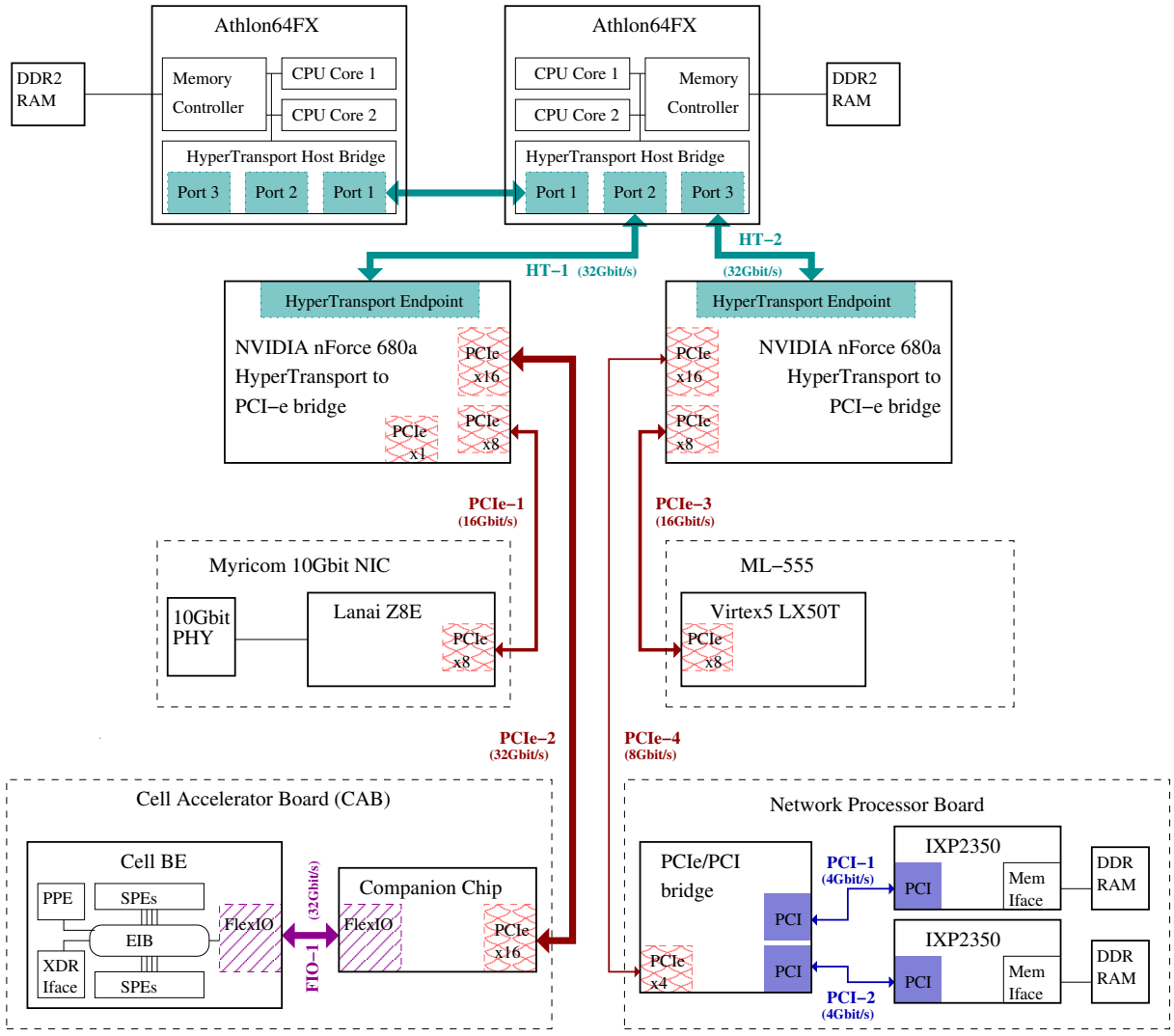

Figure 7: Bus topology and available raw data bandwidths for our platform [3, 4, 6, 10, 15, 16, 18]

number of jointly processed antennas. Therefore a 10GBit network interface card is used for baseband sample exchange between base stations. The used Myricom board [16] has an x8 PCI Express connector (16GBit/s).

3.5 Time and frequency synchronization between base stations using GPS locked Rubidium oscillator

For joint transmission, coherent transmitters are necessary. We use a commercially available GPS locked atomic oscillator (Rubidium) [2]. Its 10MHz reference frequency output is connected to the RF, the time output to the FPGA board.

3.6 FPGA board

Interfacing to the radio frequency hardware (RF) is done using the ML-555 board [18]. It hosts a Xilinx Virtex5 LXT FPGA, which contains a PCI Express interface as hard macro. The RF hardware has a digital interface, which is connected over cable to the FPGA board's LVDS connectors. For very high throughput applications, the FPGA can also be used as hardware accelerator (e.g. for FEC decoding). The Virtex5 supports up to 8 lanes PCIe and the board has an x8 connector (16GBit/s).

4. COMMUNICATION BETWEEN COMPONENTS

4.1 Data flow through the system

Fig. 5 and 6 depict how frequency domain sample data and MAC PDUs flow between platform components. This is under the assumption that FFT is handled by the FPGA board whereas channel equalization, demodulation and FEC decoding is handled by the Cell CPU. Fig. 5 also shows an alternative dataflow, where FEC decoding is offloaded to the FPGA. A third case (not shown) would be to add another FPGA board to the remaining free PCIe-x1 slot of the platform.

Fig. 7 shows physical busses and their available raw data bandwidths, connecting platform components in a topology remotely resembling a fat tree. For the worst-case scenario

Bus	Carried Flows		Throughput		Utilization	
	Up	Down	Up	Down	Up	Down
HT-1	FDStA, FEB	FDSfA, tMPDU	6.7 Gbit/s	5.5 Gbit/s	21%	17%
HT-2	FDSfA, tMPDU	FDStA, FEB	5.5 Gbit/s	6.7 Gbit/s	17%	21%
PCIe-1	FDSfN	FDStN	8.6 Gbit/s	8.6 Gbit/s	54%	54%
PCIe-2	FDStA, FDStN, FEB	FDSfA, FDSfN, tMPDU	15.3 Gbit/s	14.1 Gbit/s	47%	44%
PCIe-3	FDSfA, rMPDU	FDStA, FEB	5.5 Gbit/s	6.7 Gbit/s	34%	41%
FIO-1	FDSfA, FDSfN, tMPDU	FDStA, FDStN, FEB	14.1 Gbit/s	15.3 Gbit/s	44%	47%

Table 2: Utilization of busses for dataflow with FEC decoding performed by FPGA board, as depicted in fig. 5.

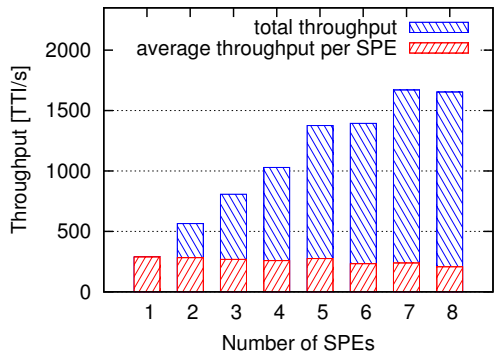

Figure 8: Parallelization speedup: throughput of MIMO-OFDM processing scales almost linearly with number of utilized SPEs, only limited by the small number of work units, which cannot be evenly distributed to many SPEs.

we assume a 12×12 MIMO system (3 cooperating basestations with 4 transmit and receive antennas each, centrally computed), requiring the bandwidths listed in table 1. This workload would utilize the data busses as summarized in table 2, with some busses being loaded with up to 54% of their available raw data rate.

4.2 System synchronisation and data exchange

Communication between components is exclusively performed using DMA block transfers to and from the recipient's RAM, cache or I/O areas. A multi-buffering or ring-buffer scheme will ensure that participants can perform data processing concurrently with transfer of data. Since all communication performed is strictly peer-to-peer (no broadcasts), no sophisticated synchronization schemes need to be implemented. The 2 notifications that need to be handled for a one-directional pipe would be request-to-send and ready-to-read. These can either be communicated via atomic writes and polled reads to locations in RAM, or using DMA writes to event notification registers for hardware components that support that mode of synchronization (such as FPGA and Cell SPE).

5. FIRST RESULTS

5.1 Joint MIMO-OFDM processing for three LTE sectors:

MIMO-OFDM processing for a 12×12 system with LTE parameters has been implemented on the Cell processor clocked at 2.8 GHz. It includes channel interpolation by Wiener filtering for each of the 144 transmit-receive antenna pairs, computation of the 1200 12×12 MMSE equalizer matrices with an update interval of 1 ms, and estimation of common phase error and SINR per resource block of 12 subcarriers. For parallelization, the computational load is divided into work units of 48 subcarriers covering 1 ms (1 TTI) of data, which are then distributed to the Cell CPU's SPEs. Data transfer in and out of the SPEs is synchronized at the TTI level with all SPEs using the *stop and signal* instruction [8,9] once per TTI to notify the host of completion of their work units, and all SPEs being synchronously restarted for the next TTI.

Our MIMO Cell implementation was compiled using Cell SDK 2.1 with GCC 4.3-20070713 as replacement for the included SPE C compiler. Fig. 8 highlights the computational throughput of our code utilizing up to 8 SPEs on the 2.8 GHz Cell CPU of the CAB. Scaling to more SPEs is limited mostly by the small number of 25 work units per TTI, which cannot be evenly destributed among SPEs. As can be seen, only 4 SPEs — equaling half a Cell processor — are required to achieve real-time MIMO processing of 1000 TTI/s. 30.2 KiB of free storage in the SPE local memory give room for future improvements to the algorithms. Also, reducing the size of work units to less than 48 subcarriers can free up parts of the 187.5 KiB currently used by data buffers.

5.2 Measured bus bandwidths

We used a test setup with CAB and the ML-555 board to gain insight into performance of the platform's bus architecture. Since the test code we loaded into the FPGA is not yet tuned for performance, we restricted the FPGA PCIe interface to 4-lanes (the Virtex5 PCIe macro supports up to 8-lanes). The test setup and available raw data rates are shown on the right side of Fig. 9, with the corresponding measured performance numbers on the left. For data transfers utilizing the PCIe bus, write transactions initiated by the data source are preferable over read transactions initiated by the sink. This is due to the asymmetry of PCIe writes versus reads, where reads consist of a two-stage request-response transaction, whereas writes are simply transmitted as unconfirmed messages. For this reason we only present results for writes. For write transactions in the opposite direction (FPGA to Cell) the FPGA code needs to support bus-master DMA (which our code not yet does).

6. CONCLUSION

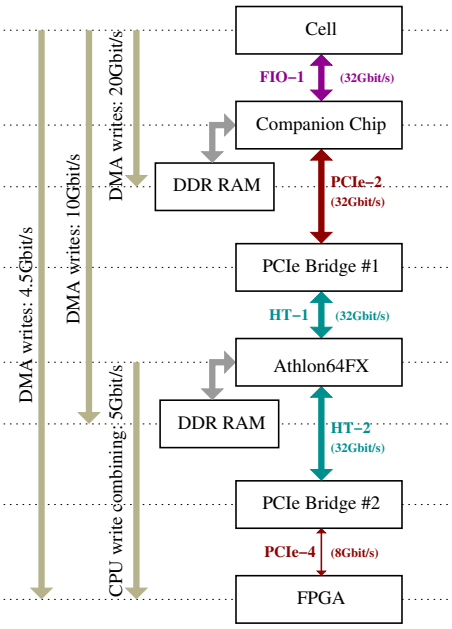

Figure 9: Results of performance measurements of data transfers between platform components.

A comparatively cheap high-performance demonstrator platform with off-the-shelf components has been presented and shown to be able to compute 12x12 MIMO-OFDM (SDMA) processing with LTE parameters. The platform is intended for cooperative joint transmission and detection demonstrations for 3GPP LTE+, but it can also be used for other systems like future evolution of WiMax and WLAN. First implementation results regarding signal processing performance and bus bandwidths are promising. Other mainboards using only a single bridge will be available in the near future and offer further room for improvement.

7. REFERENCES

[1] 3GPP. *3GPP TSG RAN: TS36.300v8.1.0 E-UTRA and E-UTRAN; Overall Description; Stage 2*, June 2007. [Online] Available: http://www.3gpp.org/ftp/Specs/html-info/36300.htm.

[2] AccuBeat. *GPS-Rubidium Clock AR70A Series*. [Online] Available: http://www.accubeat.com.

[3] AMD. *AMD Athlon(TM) 64 FX Processor Product Data Sheet*. [Online]. Available: http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/30431.pdf.

[4] Asus. *L1N64-SLI WS*. [Online]. Available: http://www.asus.com/products4.aspx?modelmenu=2&model=1530&l1=3&l2=136&l3=486&l4=0.

[5] R. Budruk, D. Anderson, and T. Shanley. *PCI Express System Architecture*. Mindshare, 2004.

[6] HyperTransport Consortium. *HyperTransport Specification 2.0 Revision B*. [Online] Available: http://www.hypertransport.org/docucontrol/HTC20031217-0036-0009.pdf.

[7] A. Ibing and V. Jungnickel. Joint transmission and detection in hexagonal grid for 3gpp lte. *International Conference on Information Networking (ICOIN)*, Jan. 2008.

[8] IBM, Sony, Toshiba. *Cell Broadband Engine Architecture (v1.01)*, October 2006. [Online] Available: http://www.ibm.com/chips/techlib/techlib.nsf/techdocs/1AEEE1270EA2776387257060006E61BA.

[9] IBM, Sony, Toshiba. *SPE Runtime Management Library for SDK 2.1*, March 2007. [Online] Available: http://www.ibm.com/chips/techlib/techlib.nsf/techdocs/1DFEF31B32111125872572420078 83F3.

[10] IP Fabrics. *Double Espresso Board*. [Online] Available: http://www.ipfabrics.com/products/de.php.

[11] B. Johansson and T. Sundin. *LTE test bed*. Ericsson. [Online] Available: http://www.ericsson.com/ericsson/corpinfo/publications/review/2007_01/files/2_lte_web.pdf.

[12] V. Jungnickel, A. Forck, T. Haustein, S. Schiffermüller, C. von Helmolt, F. Luhn, M. Pollock, C. Juchems, M. Lampe, W. Zirwas, J. Eichinger, and E. Schulz. 1Gbit/s MIMO-OFDM Transmission Experiments. In *Proc. 62nd IEEE Semiannual Vehicular Techn. Conf. (VTC 2005)* , volume 2, pages 861–866, Dallas, USA, Sept. 2005.

[13] J. A. Kahle, M. N. Day, H. P. Hofstee, C. R. Johns, T. R. Maeurer, and D. Shippy. Introduction to the cell multiprocessor. *IBM J. Res. Dev.*, 49(4/5):589–604, 2005.

[14] D. Kühling, A. Ibing, and V. Jungnickel. 12x12 mimo-ofdm realtime implementation for 3gpp lte+ on a cell processor. unpublished.

[15] Mercury Computer Systems Inc. *Cell Accelerator Board*. [Online]. Available: http://www.mc.com/products/productdetail.aspx?id=2590.

[16] Myricom. *10GBase-R*. [Online]. Available: http://www.myri.com/Myri-10G/NIC/10G-PCIE-8A-R.html.

[17] T. Weber, I. Maniatis, A. Sklavos, Y. Liu, E. Costa, H. Haas, and E. Schulz. Joint transmission and detection integrated network (joint), a generic proposal for beyond 3g systems. In *9th International Conference on Telecommunications (ICT'02)*, volume 3, pages 479–483, Beijing, China, June 2002.

[18] Xilinx. *Virtex-5 ML555 Development Kit for PCI and PCI Express Designs, User Guide*. [Online] Available: http://www.xilinx.com/bvdocs/userguides/ug201.pdf.

[19] H. Zhang and H. Dai. Cochannel interference mitigation and cooperative processing in downlink multicell multiuser mimo networks. *Eurasip Journal on Wireless Communications and Networking*, 2:222–235, 2004.