FPGA-accelerated Heterogeneous Hyperscale Server Architecture for Next-Generation Compute Clusters

René Griessl, Meysam Peykanu,
Jens Hagemeyer, Mario Porrmann
CITEC, Bielefeld University
Bielefeld, Germany
rgriessl, mpeykanu, jhagemeyer,
mporrmann@cit-ec.uni-bielefeld.de

Lars Kosmann, Patrick Knocke
OFFIS - Institute for Information Technology
Oldenburg, Germany
lars.kosmann@offis.de,
patrick.knocke@offis.de

Stefan Krupop, Micha vor dem Berge
christmann informationstechnik + medien GmbH
Ilsede, Germany
Stefan.Krupop@christmann.info,
Micha.vordemBerge@christmann.info

Michał Kierzynka, Ariel Oleksiak
Poznań Supercomputing and Networking Center
Poznań University of Technology
Poznań, Poland
michal.kierzynka@man.poznan.pl,
ariel@man.poznan.pl

ABSTRACT

Hyperscale servers tend to increase the resource efficiency in data centres by utilizing specialized and highly modular platforms. Instead of providing extreme high-performance nodes, scale-out is the main paradigm to increase the overall performance, i.e., adding additional energy-efficient compute nodes via high-bandwidth low-latency networks. In addition to this, heterogeneous architectures that can be tailored towards the specific needs of a particular application by integrating, e.g., FPGAs and GPUs, are a promising approach towards resource-efficient high-performance computing. In this paper, we present a novel highly-scalable, heterogeneous server architecture that seamlessly integrates arbitrary combinations of microservers based on general purpose CPUs, low power mobile CPUs, FPGAs and GPUs. Mobile CPUs based on the latest ARM Cortex-A15 devices with integrated GPUs are combined with FPGA-based reconfigurable SoCs, which can be used for application-specific hardware acceleration. A flexible multi-level interconnect enables communication between the microservers based on standard protocols like 10 Gigabit Ethernet. Additionally, serial high-speed links between the FPGA-based microservers are used as communication accelerators for high-bandwidth, low-latency data transmission. Control and fine-grained monitoring of all relevant system parameters is enabled by a dedicated monitoring network. The performance and energy efficiency of the platform is evaluated based on a set of synthetic benchmarks as well as a real-world sequence alignment application, which shows an increase in energy efficiency compared to CPU and GPU implementations.

Keywords

Heterogeneous Cluster Server, Hyperscale Server, Exascale, ARM-based Microserver, FPGA, High-Speed Serial Interconnect, Sequence Alignment, Needleman-Wunsch

1. INTRODUCTION

Hyperscale server clusters play an increasingly important role in data centres, providing specialized and highly modular architectures, which can be easily scaled-out for customers with specialized workloads. Therefore, hyperscale servers tend to highly optimize resource efficiency, i.e., initial cost, density, and energy efficiency are targeted in combination with high-bandwidth, low-latency networking. The high overall performance of these systems is typically achieved by the combination of large numbers of low-power server nodes rather than by extremely powerful (and energy-hungry) server nodes. In parallel, heterogeneous architectures, integrating GPUs, dedicated MPSoCS for floating-point acceleration, or FPGAs become more and more popular because of their high potential regarding power efficiency, one of the main challenges on the way to Exascale computing. Combining these two approaches leads to novel architectures that are highly scalable at a very fine-grained level and that can be optimized towards specific applications in terms of performance and power by using a dedicated combination of system components. Their high scalability makes them valuable not only for large-scale data centres but also for SMEs that want to maximize the efficiency of their in-house computing facilities.

While high-end CPU-based systems provide easy programming and fast deployment of applications, heterogeneous systems integrating GPGPUs for offloading data-parallel and homogeneous parts of the application require significant additional effort. Special programming environments like CUDA or OpenCL have to be used for application development [24], [23]. For many applications, FPGAs are a promising alternative to GPUs and have shown comparable or even higher performance and energy-efficiency [21]. Until recently, a major drawback of these architectures was

that they had to be programmed using special hardware description languages like VHDL or Verilog. Meanwhile, C-based high-level synthesis tools have significantly increased the productivity of the FPGA designers [12] but still a huge amount of hardware knowledge is required for realizing efficient FPGA implementations. Only recently, new toolflows have been realized by research and industry, making FPGAs easily usable by software engineers via OpenCL [6] or CUDA [8]. Unlike classical homogeneous reconfigurable devices, today's FPGAs integrate dedicated hardware blocks for calculation (e.g., DSP blocks), computation (e.g., embedded ARM cores), and communication [30, 1]. Here, communication is based on integrated high-speed serial links which can be configured to support standard protocols like PCIe, but which can also be used for high-speed communication with very low latency based on proprietary protocols [26].

In addition to increasing system efficiency by dedicated hardware accelerators, new CPU architectures are approaching on the server market. ARM-based servers are built based on SoCs from the embedded and mobile domain, promising very low power together with embedded hardware accelerators like GPUs. Analyses with respect to performance and energy efficiency of ARM Cortex-A8 and Cortex-A9 have been performed, e.g., by Padion et al. [22, 19] and Rajovic et al. [27]. Apart from the embedded processors discussed above, new ARM server processors are available, providing high performance at comparatively low power (e.g., Cavium ThunderX [9] and Applied Micro X-Gene [31]).

Optimizing a compute cluster with respect to energy efficiency requires not only the most suitable compute units but it also has to target communication and the remaining parts of the cluster including power distribution, power conversion and cooling. Both server hardware and mechanical design are subject to optimization in research and industry, as discussed, e.g., in [11]. Up to 38 percent power reduction have been achieved by Frachtenberg et al. [4] optimizing these aspects at Facebook data centers. In [14] it is shown that lower level of power usage effectiveness (PUE) can be achieved also with new methods of liquid cooling. Real-time power and temperature monitoring can be used to actively distribute the workload and to control the cooling system. Monitoring and control is especially useful to turn off or lower the power consumption of system components, currently not utilized [15].

In this paper we present a novel heterogeneous platform that seamlessly integrates x86 CPUs, low power mobile CPUs, GPUs, and FPGAs in a single enclosure. The basic concept and the architecture of the system, called RECS®|Box [7], are discussed in detail in the next section (Section 2). In Section 3 performance and energy efficiency figures of the system are discussed based on synthetic benchmarks as well as a real-world application from the field of sequence alignment in bioinformatics.

2. RECS SYSTEM ARCHITECTURE

The RECS®|Box system architecture is designed using a modular approach, resulting in a highly scalable system. The platform allows tight integration of general purpose processors, embedded processors, FPGAs, GPUs, and multi/many-core processors, thus enabling the realization of a truly heterogeneous hyperscale server architecture. This section describes the RECS®|Box system architecture in detail. After explaining the general concept (Section 2.1), the power (Section 2.2) and the architecture of the compute unit (Section 2.3 to Section 2.8) are discussed.

2.1 The RECS Concept

The architecture of the RECS®|Box system is a modular system architecture which splits into the building blocks discussed in the following. Every block consumes just 1 RU, so more than 40 of these units can be fitted into a single server rack, leading to a high integration density, essential for a hyperscale server approach.

- **RCU**: RECS®|Box Compute Unit — The Compute Unit hosts the different heterogeneous processing elements of the RECS®|Box platform, along with the communication and management infrastructure. Depending on the type of processing elements, a RCU can host up to 72 microservers. Details are provided in Section 2.3.
- **RPU**: RECS®|Box Power Unit — The Power Unit supplies the power to the RCUs. The power distribution system is based on 12 Volts. The RPU integrates 10 single PSUs, which are managed dynamically based on the current load conditions, as explained in section 2.2.
- **TOR-Master**: Top of RECS®|Box Rack — The TOR-Master acts as a single interface point to the user and/or manager of the RECS®|Box platform. It can be seen as a supervising and collecting node of the different distributed control systems inside the RPUs und RCUs in a particular RECS®|Box system. The TOR-Master can be a distinguished node in case of a big RECS®|Box installation, or realized inside the control infrastructure of one RCU, in case of a small/medium size installation.

The cooling concept of the RECS®|Box platform is constructed as a side-cooling solution. In contrast to conventional rackmount servers, which are constructed to allow an airflow from the front to the back of the chassis, the RECS®|Box platform uses a left to right airflow. This allows for an efficient cooling solution inside the server rack itself when combined with commercially available cooling systems like the Rittal LCP system [28] or the Lehmann SideCooler Rack [16].

Table 1 shows a comparison of the RECS®|Box system to commercially available hyperscale server systems like HP Moonshot [10]. The table assumes a standard 42 RU server rack filled with a RECS®|Box system or a HP Moonshot system. Both architectures offer low power (LP) as well as high performance (HP) microservers. While the Moonshot system integrates the power supply, the RECS®|Box system uses a dedicated power supply. This is reflected in the comparison by the rows reporting the effective (1 RU (eff.)) power figures and microserver numbers. As shown in the table, the RECS®|Box architecture allows about 25 % more microservers per rack.

Table 1: Comparison of RECS®|Box and the commercial available HP Moonshot [10]

		RECS	HP Moonshot
	Server Height	1 RU	4.3 RU
	Rack Depth	1200 mm (Sidecooler)	1200 mm
per chassis	#LP Server	72	180
	#HP Server	18	45
per rack (42 RU)	#LP Server	2016	1620
	#HP Server	504	405
1 RU (eff.)	#LP Server	48	38.57
	#HP Server	12	9.64
Power	1 Chassis	1.5 kW	4.8 kW
	1 RU (eff.)	1 kW	1.12 kW
	1 Rack	42 kW	47 kW

2.2 Architecture of the RPU

As discussed above (Section 2.1), the RECS®|Box architecture is split into computing units (RCU) and power units (RPU). The RECS system highly benefits from its split architecture, which makes it possible to specify and activate only the required power supply units (PSUs) via an intelligent power management system based on the current power budget. This is even more useful when using multiple RCUs connected to one single RPU. An RPU consists of ten power supply units, each supplying 12 Volts, connected to a common power rail. The units are controlled and monitored by a microcontroller supervising unit. A single RPU can serve up to nine low-power RCUs, depending on the respective RCU power requirements. The maximum power capability of a complete RPU is 3000 Watts/250 Amperes. The information collected by the microcontroller supervising unit is forwarded to the TOR Master which then activates a minimum number of power units sufficient for the supply of the attached RCUs. Thus, the management software is capable of applying different redundancy schemes, such as N + 1 redundancy by activating one more supply than actually required. When a microserver inside the RCU is switched on or off, the number of active power supplies is adjusted by the management software.

2.3 Architecture of the RCU

As depicted in Figure 1, the RECS®|Box integrates up to 18 compute boards, each equipped with up to four microserver boards in a one rack unit enclosure. The compute boards are connected to a central backplane, which is used as communication backbone as well as for management functions and power. The system is divided into three identical pieces, each aggregating six compute boards. Communication capabilities include switched Gigabit Ethernet as well as a multipurpose interconnect infrastructure for a bandwidth of up to

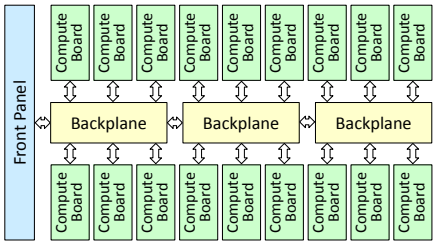

Figure 1: Structure of the RCU

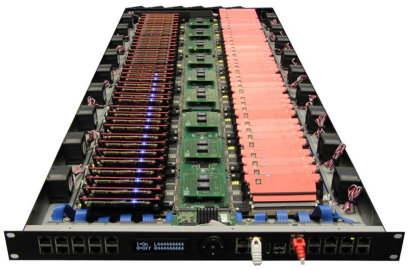

Figure 2: Example of a RCU, equipped with 72 ARM compute boards

40 Gbit/s per compute board. Microserver boards for Intel and AMD x86/x64 CPUs as well as for lower power mobile CPUs and FPGAs have been realized and can be flexibly combined to a heterogeneous multiprocessor system. Figure 2 shows an example of a RECS®|Box populated with 72 ARM microservers based on the Samsung Exynos 5250, each integrating a Cortex-A15 dual core.

2.4 RCU Communication Infrastructure

The communication infrastructure of the RECS®|Box system, provides three levels of interconnect between the integrated compute boards. A central backplane which is split into three identical sections implement the communication infrastructure. Each of the three sections connects up to six compute boards (Figure 1). The connection to the compute boards integrates all required interfaces, like power, PCIe, Ethernet and Management. The compute boards are hot pluggable/swappable for easy maintenance. All networks connected to the compute boards are provided to the user via a dedicated front panel (Figure 2). Dedicated net boards enable configuring the compute network to be either Gigabit Ethernet or 10 Gbit Ethernet in a modular fashion.

While the compute network is externally switched to provide maximum flexibility, the management network is a dedicated Gigabit Ethernet network, which is internally switched on the backplane. Distributed monitoring and control facilities are provided by dedicated microcontrollers on each compute board, connected via an I^2C-based system management bus. Furthermore, a KVM solution is integrated so that four USB ports and one HDMI port on the front panel can be switched to any of the microservers.

2.5 COM Express-based Compute Board for RCU

Compute boards based on the COM Express standard [25] are used to integrate high performance (HP) microserver in the RECS®|Box architecture. The COM-Express standard [25] is a computer-on-module standard that allows integration of a wide range of COTS-available x86-based microservers. Additionally, FPGA-based microservers utilizing Xilinx Zynq devices are available (Section 2.8) as COM Express compliant modules. The compute board provides the necessary connections for hosting one COM Express module of Type 6 and Type 2, supporting the "Basic" module size of 95 mm x 125 mm. In order to accommodate accelerator cards, such as Intel Xeon PHY or GPGPUs, a PEG slot with 16 PCI Express lanes has been integrated. It also provides

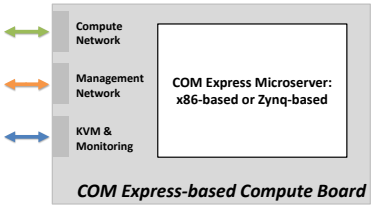

Figure 3: Architecture of the COM Express-based compute board

an mSATA connector to attach local storage for the microserver as well as two regular external SATA ports for further extensions. Standard connectors provide power for the PEG card and the SATA extensions. A microcontroller for monitoring and control is integrated on the compute board. It can be used to separately switch on/off the power of the COM Express module (regular as well as standby power), the PEG card, the SATA and mSATA connectors and the networking components on the baseboard. Additionally, it offers a fine-grained monitoring infrastructure for the various voltages, currents and temperatures of the module. Figure 3 shows the high level architecture of the COM Express compute board.

2.6 Apalis-based Compute Board for RCU

As a complement to the COM Express-based compute board, the Apalis-based compute board is used to integrate low power (LP) mircoservers in the RECS®|Box architecture. One Apalis-based compute board supports up to four low power (LP) microservers. The Apalis computer-on-module standard has been defined by Toradex [29] with ARM-based microservers in mind. In addition to the COTS-available modules based on e.g. Nivida Tegra or Freescale i.MX 6, a low power microserver based on Samsung Exynos (Section 2.7) is available. Network connections towards the microservers are provided by two independent Gigabit Ethernet switches. One switch connects the RECS management network to all four compute modules; the other switch connects the compute network to all four compute modules. Therefore, the only network interface towards the compute network that is currently supported for ARM baseboards is Gigabit Ethernet. As for the COM Express-based compute board, an integrated monitoring microcontroller is used to control the power to the microservers, the on-board Gigabit Ethernet switches, and the on-board KVM separately. The power consumption for each microserver, the fans, and the remaining parts of the baseboard can be monitored independently. The high level architecture of the Apalis-based compute board is shown in Figure 4.

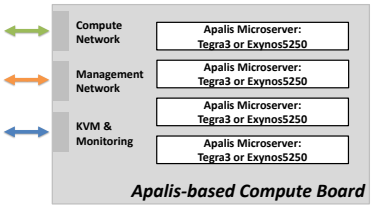

Figure 4: Architecture of the Apalis-based compute board

2.7 Exynos-based microserver

The Apalis-based compute boards, as discussed in section 2.6, can be populated with microservers compliant to the Apalis standard. Apart from the modules currently available from Toradex, a microserver based on the Samsung Exynos 5250 has been developed. It integrates an ARM

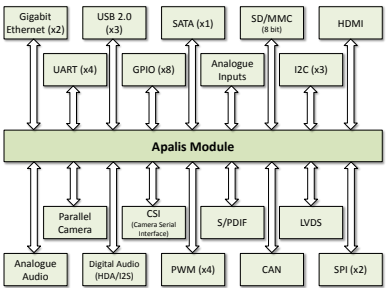

Figure 5: Interfaces of the RECS Exynos COM Compute Module

Cortex-A15 Dual Core running at 1.7 GHz and a MALI T604 MP4 GPU supporting OpenCL. The main memory comprises 4 GByte LPDDR3 memory directly accessible by the CPU and the GPU; additionally, 16 GByte eMMC are attached to the CPU as non-volatile storage. The module integrates all interfaces of the Apalis standard that are relevant for the integration into the RECS®|Box system as well as most of the remaining interfaces. Figure 5 gives an overview of the integrated interfaces. As an extension to the Apalis standard, the module integrates two Gigabit Ethernet interfaces, giving the possibility to easily access both, the management network as well as the compute network (Section 2.4) from the COM. A Linux Kernel from Linaro has been ported to the Exynos-based COM. The file system is built based on a Linaro Ubuntu server distribution [18].

2.8 Zynq-based microserver

The Zynq-based microserver is integrated into the RECS®|Box platform using the COM Express-based compute board. The Xilinx Zynq-7000 SoC (system on chip) was chosen for integration because it offers the processor-style interfaces necessary for integration in the COM Express standard alongside with a state of the art FPGA fabric. The Zynq-7000 family is based on the Xilinx all Programmable SoC architecture which integrates a dual-core ARM Cortex-A9 processing system and 28 nm Xilinx programmable logic in a single device. The tight integration of these two along with a high-speed AXI interconnect as the internal communication infrastructure ensures flexible, high-bandwidth and low-latency communication between different parts of the device. Furthermore, up to eight PCI Express lanes are available on the Zynq device, enabling fast and low-latency communication to other microservers in the RECS®|Box system (e.g. 10 Gigabit Ethernet). Three different pin-compatible Zynq devices can be populated on the Zynq-based COM Express microserver (XC7Z030/XC7Z35/XC7Z045), allowing selection of the optimal size of the FPGA fabric for a given application.

The Zynq-based COM Express microserver fully complies to the COM Express standard [25]. Apart from integration into the RECS®|Box system, the module can be integrated

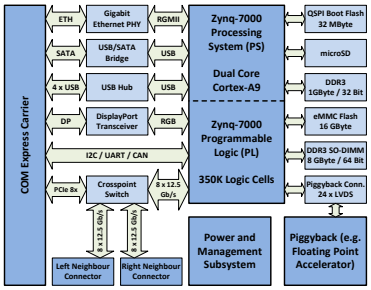

Figure 6: Architecture of the Zynq-based COM Express microserver

into all systems supporting the COM Express basic (95 mm x 125 mm) form factor. This makes the board interesting for use in other applications, as COM Express is widely used in embedded systems. In order to be compliant to the COM Express standard, the Zynq-7000 SoC was extended by additional I/O components to provide all required interfaces. Figure 6 gives an overview of the architecture of the module and its I/O structure.

In addition to the interfaces defined in the COM Express module standard, high-speed serial communication between the different Zynq-based microservers is supported by utilizing the serial high-speed transceivers of the Xilinx Zynq devices. Depending on the device, the bandwidth of these transceivers reaches up to 12.5 Gbit/s per channel, where each lane consists of an RX and a TX channel allowing full duplex communication. The Zynq-7000 is equipped with either 4 (XC7Z030) or 8 (XC7Z035/XC7Z045) lanes. These are connected via asynchronous crosspoint switches to enable highly flexible communication topologies between the different modules. The use of the serial high-speed transceivers enables a dedicated, high-speed (up to 100 GBit/s) low latency (300 ns) communication infrastructure between the Zynq-based microservers.

3. APPLICATIONS

To characterize the RECS®|Box platform, a number of benchmarks were conducted. In Section 3.1 performance and energy efficiency figures of the system are discussed based on synthetic benchmarks. The implementation, performance and energy efficiency of a real-world application from the field of sequence alignment in bioinformatics is analyzed in Section 3.2.

3.1 Synthetic benchmarks

The main characteristics of the microservers used in the RECS®|Box system are shown in Table 2. If applicable, their peak floating point compute power for single and double precision operations or the peak DSP performance is listed. The listed values are based on the maximum number of operations per cycle, i.e., a MAC operation is counted as 2 FLOPs, as it combines a multiplication and an addition. The values can be interpreted as the theoretical maximum peak performance of the architecture.

As described in detail in Sections 2.6 to Section 2.8, the RECS®|Box system supports x86, ARM and FPGA-based microservers. The ARM-based microserver uses the Exynos

Table 2: Main characterstics of the used microservers

ARM-based microserver (Dual Core)						
CPU	Exynos 5250	1.7 GHz	Cortex-A15 Dual Core	1 MByte L2 Cache	6.8 GFLOPS (DP/VFPv4)	27.2 GFLOPS (SP/NEON)
GPU	Mali-T604 MP4 Quad Core	533 MHz	Midgard (1st-gen)	256 kByte L2 Cache	21.32 GFLOPS (DP/5 FP64 per ALU)	72.488 GFLOPS (SP/17 FP32 per ALU)
Memory	DDR3L-1600	800 MHz	12.8 GByte/s	4 GByte		
x86-based microserver (Dual Core)						
CPU	i3-3120ME	2.4 GHz	Ivy-Bridge Dual Core	3 MByte L3 Cache	43.2 GFLOPS (DP/AVX)	86.4 GFLOPS (SP/AVX)
Memory	DDR3L-1600	800 MHz	25.6 GByte/s	4 GByte		
FPGA-based microserver (Dual Core)						
CPU	Zynq	800 MHz	Cortex-A9 Dual Core	512 KByte L2 Cache	3.2 GFLOPS (DP/VFPv3)	12.8 GFLOPS (SP/NEON)
CPU Memory	DDR3-1066	533 MHz	4.266 GByte/s	1 GByte		
FPGA	XC7Z045	350,000 Logic Cells	2,180 kByte BlockRAM	900 DSP Slices	1,334 GMACs	218,600 / 437,200 LUTs / FFs
FPGA Memory	DDR3-1333	667 MHz	10.666 GByte/s	4 GByte		

microserver described in Section 2.7. Apart from a Cortex-A15 Dual Core CPU it uses a Mail-T604 GPU. Using the CPU, a theoretical computing performance of 6.8 GFLOPS (double precision) can be achieved using the Vector Floating Point engines of both cores (the MAC instruction gives 2 FLOPS/cycle). Using the NEON SIMD extension of the CPU, 27.2 GFLOPS (single precision) can be obtained using 4 MAC operations per cycle. The same basic performance characteristics apply for the Cortex-A9 Dual Core CPU, however, the overall performance of the CPU core is reduced due to the lower clock frequency. The GPU of the Exynos (Mali T604 MP4) can deliver 5 double precision or 17 single precision FLOPS per cycle per ALU, leading a performance of 21.32 GFLOPS (double precision) or 72.488 GFLOPS (single precision). The ALU of the Mali-T604 is quite flexible, in addition to FP64 (double precision) and FP32 (single precision), it also supports FP16. For the x86-based compute boards, available COM Express modules are used. Utilizing the Ivy microarchitecture, AVX (Advanced Vector Extensions) instructions can be used. This instruction set extension offers vector instructions for addition and multiplication, leading to 8 FLOPs/cycle for double precision, or 16 FLOPs/cycle for single precision. Characterizing the peak performance of the Zynq FPGA fabric is only possible for the embedded DSP blocks, as the performance of the other components is highly application dependent.

In comparison to Table 2, Table 3 shows the performance of the microservers obtained from different synthetic benchmarks. To measure the floating point performance of the CPUs, the LINPACK benchmark was used. On the GPU

Table 3: Benchmark results of used microservers

Exynos	5250	1.5 GFLOPS / DP	Linpack	12 Watt
		7.3 GByte/s	Read BW	4.5 Watt
		8.6 GByte/s	Write BW	4.6 Watt
Mali	T604 MP4	14.2 GFLOPS / DP	SHOC	7.3 Watt
		43.2 GFLOPS / SP	SHOC	7.6 Watt
		6.1 GByte/s	Read BW	7.1 Watt
		7.2 GByte/s	Write BW	7.3 Watt
Intel i3	3120ME	14 GFLOPS / DP	Linpack	17 Watt
		14.2 GByte/s	Read BW	20 Watt
		16.8 GByte/s	Write BW	22 Watt
Xilinx	Zynq	8.99 GByte/s	Read BW	6.4 Watt
		8.35 GByte/s	Write BW	6.7 Watt

(Mail T604 MP4), the SHOC benchmark [2] was used. Apart from the floating point performance, also the read and write data transfer rate to and from the main memory was measured. Comparing the different architectures shows that the x86-based i3-3120ME gives the highest total floating point performance. This is not surprising, as it has a powerful instruction set extension for exactly this feature. While the ARM-based 5250 shows a low floating point performance, the T604 MP4, integrated alongside the 5250 on the Exynos provides a similar floating point performance at less than half the power, leading to a more energy-efficient calculation. This difference gets even more apparent if the single precision performance of the T604 MP4 is taken into account. Comparing the memory bandwidth utilization of the different architectures in Table 3 with respect to their theoretic maximum values in Table 2 shows that the FPGA has a noticeably higher bandwidth utilization.

3.2 G-DNA application

Sequence alignment is widely used in bioinformatics to compare two or more DNA, RNA or protein sequences. Unlike typical string comparison the algorithms used have to consider specific mutations that may have occurred between these sequences, i.e. insertions, deletions or substitutions of individual nucleotides. A common task is to find the global alignment of two sequences. Global alignment attempts to consider every nucleotide of the whole sequence. An efficient solution to this task is the Needleman-Wunsch algorithm [20].

GDNA [5] is an implementation of Needleman-Wunsch specialised for graphics processing units (GPUs). It is one of the benchmark applications in the FiPS project [3]. One of the goals of the project workflow [13] is the identification of significant parts within a larger application that can be separated and implemented on different hardware. A kernel was identified by applying the FiPS workflow on the GDNA implementation. In addition to a GPU implementation, the workflow suggested the implementation of this kernel on FPGA by analyzing the kernel structure [17]. The actual transformation was done by using the high level synthesis (HLS) tool Xilinx Vivado. Such HLS tools offer FPGA hardware implementations based on C, C++ or SystemC, thus providing a high abstraction level compared to traditional hardware description languages. Nevertheless, expert knowledge is needed to get an efficient hardware implementation. The C++ code derived from the GDNA application was used as a base to start off with code reformation in order to produce a synthesisable component. It was optimized in order to reduce memory accesses as well as memory utilization. After that, the single processing elements (PE) and the pipelining structure of the core was derived.

3.2.1 Processing Element and Pipelining

The design of the PE is derived from the implementation used in GDNA application. In order to reduce the number of memory accesses, GDNA encodes nucleotides as two bit and three bit values – depending on the type of sequences. The tuples are then chained and stored in memory as 32-bit unsigned integers. This allows to fetch subsequences of 10 to 16 nucleotides in a single memory access.

For this particular implementation, the two bit encoding was chosen. The PE resembles this memory access reduction method. Each PE takes two subsequences of 16 nucleotides length as input. This is contrary to implementations that can be found or obtained as IP. Most implementations follow the matrix strategy where one PE resembles one matrix cell. An advantage of single nucleotide PEs is the increased flexibility and the reduced synthesis time. Such implementations can build any sequence length desired.

In addition to the subsequences, each PE needs one horizontal and one vertical input and output vector. Those hand over calculated scores from adjacent PEs, immediately preceding a given PE. This is required for pipelining purposes.

The sequence alignment core assembles several PEs in order to allow the alignment of larger sequences. This is constrained by the PEs input encoding, which requires subsequence length of 16 nucleotides. Therefore, the whole sequence length is set to be a factor of 16. Additionally, the sequence length has to be known at synthesis time. Dynamic sequence structures cannot be pipelined efficiently as the synthesis tool does not know the required number of PE instances. Shorter sequences can be realized via placeholders at the end of the sequences that do not affect alignment scoring.

The loop structure which was used in GDNA implementation allowed the application to reduce its memory effort not only by bitwise data encoding, but also by reusing the array for scoring values. This type of implementation already resembles a systolic array structure, which was transfered to this FPGA implementation as well. In this process the loop structure had to be slightly changed to preserve the right order of executions. This effort results in the systolic array structure depicted in Figure 7. It shows an example presenting the pipeline stages within the systolic array and the first three alignment steps of two 64 nucleotide long sequences. In the first step only PE 1 is active. In the second step part of the sequence is handed to PE 2, additionally with the output vector from PE 1. Both PEs continue to work in parallel. This progresses until all submatrices have been processed. The implementation based on a systolic array benefits from the efficient pipelining structure in terms of speed. Additionally, the area effort is reduced because the number of necessary instances of PEs is reduced from quadratic effort to linear effort. For a sequence length of 112 nucleotides, this results in seven PEs instead of 49, each aligning 16 nucleotides.

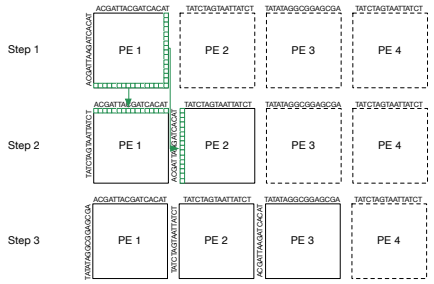

Figure 7: Example of pipeline stages within the systolic array

3.2.2 Integration on FPGA

With the described architecture, the area requirements of PEs and computation structure allow the integration of 11 individual cores into the Zynq XC7Z45FF676-2. The detailed area requirements of a single PE and a single core strongly depend on the overall integration with other cores and communication. In addition to the 11 cores, the FPGA part of the Zynq also implements the communication infrastructure. It is realised by an AXI bus which connects each core to its sequence buffer and to integrated ARM dual core processors. Additionally, the cores need internal Block RAM in order to have buffers for the additional computation arrays. The overall utilization is shown in Table 4. The AXI bus communication structure allows access to each individual alignment core.

Table 4: Programmable logic utilization for Zynq XC7Z045

	Used	Available	Utilization
Slice LUTs	209641	218600	95.90 %
LUT as Logic	208805	218600	95.51 %
LUT as Memory	836	70400	1.18 %
Slice Registers	171635	437200	39.25 %
Block RAM Tile	13	545	2.38 %

3.2.3 Performance

The FPGA implementation is compared to a CPU based implentation of Needleman-Wunsch running on an i5-4400E and the GDNA implementation running on a Tesla M2070 GPU (Fermi) and a newer Nvidia Tesla K40c (Kepler) GPU. The results were measured using the RECS®|Box infrastructure. In order to evaluate the performance, an additional performance counter was introduced which measures the Giga Cell Updates Per Second (GCUPS) for each architecture. The results of this measurement are shown in Table 5. It can be seen, that GDNAs performance on GPU is way above the average performance of CPU and FPGA. The CPU implementation is the slowest one. This is due to the fact, that CPUs do have limited capabilities of pipelining and consecutive execution. Both these abilities are crucial for the performance of the Needleman-Wunsch algorithm. GPU performance has increased over chip generations, which is visible in the much higher performance at much lower power of the Tesla K40 over the M2070. In direct competition with the GPU, the FPGA implementation achieves just 9 % to 15 % of the GCUPS performance. But this comes at a much lower power draw, resulting in a way better GCUPS per watt ratio. The performance per energy used is therefore the best in the case of FPGA, which was the main goal of this implementation.

Table 5: Performance of sequence alignment on different architectures. Efficiency gain is calculated relative to CPU implementation.

	GCUPS	Power [Watt]	GCUPS/ Watt	Efficiency gain
Intel i5-4400E	0.2	30	0.007	1
Tesla M2070	44.02	196.8	0.224	32
Tesla K40c	71.03	123.05	0.577	82.46
Zynq XC7Z045	7.08	8.7	0.814	116.26

4. CONCLUSION

In this paper a novel heterogeneous platform that seamlessly integrates x86 CPUs, low power mobile CPUs, GPUs, and FPGAs has been presented. The RECS®|Box system aims at providing a hyperscale server solution for scale-out server installations. The heterogeneous system integrating GPUs and FPGAs offers a promising approach towards resource-efficient high-performance computing for a wide range of applications. In addition to this, the integration of traditional PCIe-based hardware accelerators, e.g., Intel Xeon Phi, is possible as well. The architecture is based on microservers compliant to available computer-on-module standards, for integration in the RECS®|Box system, microservers based on ARM Cortex-A15 devices with integrated GPUs as well as FPGA-based reconfigurable SoCs have been designed. A flexible multi-level interconnect enables communication between the microservers based on standard protocols like 10 Gigabit Ethernet. Additionally, serial high-speed links between the FPGA-based microservers are used as communication accelerators for high-bandwidth, low-latency data transmission. Compared to commercially available homogeneous scale-out systems, the RECS®|Box architecture not only provides a heterogeneous solution, but also offers higher integration density. The performance and energy efficiency of the platform has been characterized based on a set of synthetic benchmarks. Futhermore, a real-world application from the field of sequence alignment in bioinformatics has been implemented on the FPGA-based microserver, which shows a drastic increase in performance and energy efficiency when compared to a CPU-based implementation, demonstrating the potential of heterogeneous hyperscale server systems. Compared to a GPU-based implementation, the FPGA still achieves 1.18 to 3.63 times better energy efficiency figures.

5. ACKNOWLEDGMENT

This research was supported by the EU Seventh Framework Programme FP7/2007–2013 under grant agreement no. FP7-ICT-2013-10 (609757). Webpage: www.fips-project.eu. This work was also funded as part of the Cluster of Excellence Cognitive Interaction Technology 'CITEC' (EXC 277), Bielefeld University, and supported by the PL-Grid Infrastructure in Poland.

6. REFERENCES

[1] Altera-SoCs. Dual-Core ARM Cortex-A9 MPCore Processor (SoC). *Available at* `http://www.altera.com/devices/processor/arm/cortex-a9/m-arm-cortex-a9.html`.

[2] A. Danalis, G. Marin, C. McCurdy, J. S. Meredith, P. C. Roth, K. Spafford, V. Tipparaju, and J. S. Vetter. The Scalable Heterogeneous Computing (SHOC) Benchmark Suite. In *Proc. of the 3rd Workshop on General-Purpose Computation on Graphics Processing Units*, GPGPU '10, pages 63–74, New York, NY, USA, 2010. ACM.

[3] FiPS. FiPS Project - Developing Hardware and Design Methodologies for Heterogeneous Low Power Field Programmable Servers. *Available at* `https://www.fips-project.eu/wordpress/`.

[4] E. Frachtenberg, A. Heydari, H. Li, A. Michael, J. Na, A. Nisbet, and P. Sarti. High-Efficiency Server Design. In *High Performance Computing, Networking, Storage and Analysis (SC), 2011 Int. Conf. for*, pages 1 – 11, November 2011.

[5] W. Frohmberg, M. Kierzynka, J. Blazewicz, P. Gawron, and P. Wojciechowski. G-DNA – a highly efficient multi-GPU/MPI tool for aligning nucleotide reads. *Bulletin of the Polish Academy of Sciences. Technical Sciences*, Vol. 61(4):989–992, 2013.

[6] S. Gao and J. Chritz. Characterization of OpenCL on a scalable FPGA architecture. In *ReConFigurable Computing and FPGAs (ReConFig), 2014 Int. Conf. on*, pages 1–6, Dec 2014.

[7] R. Griessl, M. Peykanu, J. Hagemeyer, M. Porrmann, S. Krupop, M. Berge, T. Kiesel, and W. Christmann. A scalable server architecture for next-generation heterogeneous compute clusters. In *12th IEEE Int. Conf. on Embedded and Ubiquitous Computing (EUC)*, pages 146–153, Aug 2014.

[8] S. Gurumani, J. Tolar, Y. Chen, Y. Liang, K. Rupnow, and D. Chen. Integrated CUDA-to-FPGA Synthesis with Network-on-Chip. In *Field-Programmable Custom Computing Machines (FCCM), 2014 IEEE 22nd Ann. Int. Symposium on*, pages 21–24, May 2014.

[9] L. Gwennap. ThunderX Rattles Server Market. *Microprocessor Report*, June 2014.

[10] HP. Moonshot. *Available at* `http://www8.hp.com/us/en/products/servers/moonshot/`.

[11] W. Huang, M. Allen-Ware, J. B. Carter, E. Elnozahy, H. Hamann, T. Keller, C. Lefurgy, J. Li, K. Rajamani, and J. Rubio. Tapo: Thermal-aware power optimization techniques for servers and data centers. *Int. Green Computing Conf. (IGCC)*, pages 1–8, 2011.

[12] K. Karras, M. Blott, and K. A. Vissers. High-Level Synthesis Case Study: Implementation of a Memcached Server. *CoRR*, abs/1408.5387, 2014.

[13] P. Knocke, R. Gorgen, J. Walter, D. Helms, and W. Nebel. Using early power and timing estimations of massively heterogeneous computation platforms to create optimized hpc applications. In *12th IEEE Int. Conf. on Embedded and Ubiquitous Computing (EUC)*, pages 162–169, Aug 2014.

[14] M. LaMonica. HP's Water-Cooled Supercomputer is Designed for the Hydrophobic. *Available at* `http://spectrum.ieee.org/tech-talk/computing/hardware/a-watercooled-supercomputer-for-the-hydrophobic-`.

[15] C. Lefurgy, K. Rajamani, F. Rawson, W. Felter, M. Kistler, and T. W. Keller. Energy management for commercial servers. *Computer*, 36(12):39–48, Dec. 2003.

[16] Lehmann. Lehmann SideCooler Rack. *Available at* `http://www.lehmann-it.eu/`.

[17] Y. Lhuillier, J. Philippe, A. Guerre, M. Kierzynka, and A. Oleksia. Parallel Architecture Benchmarking: From Embedded Computing to HPC, a FiPS Project Perspective. In *12th IEEE Int. Conf. on Embedded and Ubiquitous Computing (EUC)*, pages 154–161, Aug 2014.

[18] Linaro. Ubuntu server. *Available at* `http://www.linaro.org`.

[19] MONT-BLANC. European Approach Towards Energy Efficient High Performance. *Available at* `http://www.montblanc-project.eu/publications`.

[20] S. Needleman and C. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48(3):443–453, 1970.

[21] K. Ovtcharov, O. Ruwase, J.-Y. Kim, J. Fowers, K. Strauss, and E. S. Chung. Accelerating Deep Convolutional Neural Networks Using Specialized Hardware. *Available at* `http://research.microsoft.com/apps/pubs/default.aspx?id=240715`, February 2015.

[22] E. L. Padoin, D. A. de Oliveira, P. Velho, and P. O. Navaux. Evaluating Performance and Energy on ARM-based Clusters for High Performance Computing. *41st Int. Conf. on Parallel Processing Workshops*, pages 165–172, 2012.

[23] E. L. Padoin, L. L. Pilla, F. Z. Boito, R. V. Kassick, P. Velho, and P. O. A. Navaux. Evaluating application performance and energy consumption on hybrid CPU+GPU architecture. *Cluster Computing*, 16(3):511–525, 2013.

[24] Peter Kogge et. al. Exascale computing study: Technology challenges in achieving exascale systems. In *Darpa Report*, September 2008.

[25] PICMG. PICMG COM.0 R2.1 - Com Express Module Base Spec. *Available at* `http://www.picmg.org`.

[26] A. Putnam, A. Caulfield, E. Chung, D. Chiou, K. Constantinides, J. Demme, H. Esmaeilzadeh, J. Fowers, G. P. Gopal, J. Gray, M. Haselman, S. Hauck, S. Heil, A. Hormati, J.-Y. Kim, S. Lanka, J. Larus, E. Peterson, S. Pope, A. Smith, J. Thong, P. Y. Xiao, and D. Burger. A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services. In *41st Ann. Int. Symposium on Computer Architecture (ISCA)*, June 2014.

[27] N. Rajovic, L. Vilanova, C. Villavieja, N. Puzovic, and A. Ramirez. The low power architecture approach towards exascale computing. *Journal of Computational Science*, 4(6):439 – 443, 2013.

[28] Rittal. LCP - Liquid Cooling Packages. *Available at* `http://www.rittal.com/de-de/product/list.action?categoryPath=/PG0001/PG0800ITINFRA1/PGR1951ITINFRA1/PG1023ITINFRA1`.

[29] Toradex. Apalis Computer Module - Module Specification. *Available at* `http://developer.toradex.com/hardware-resources/arm-family/apalis-module-architecture`.

[30] Xilinx-SoCs. Zynq-7000 AP SoC. *Available at* `http://www.xilinx.com/products/silicon-devices/soc/zynq-7000/index.htm`.

[31] A. Yeung, H. Partovi, Q. Harvard, L. Ravezzi, J. Ngai, R. Homer, M. Ashcraft, and G. Favor. 5.8 A 3GHz 64b ARM v8 processor in 40nm bulk CMOS technology. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE Int.*, pages 110–111, Feb 2014.