Energy-Efficient Inertial Sensor Fusion on Heterogeneous FPGA-Fabric / RISC System on Chip

H.-P. Brückner, C. Spindeldreier, H. Blume
Leibniz Universität Hannover
Institute of Microelectronic Systems
Hannover, Germany
{brueckner, spindeldreier, blume}@ims.uni-hannover.de

***Abstract*—Energy efficiency is a major design goal for mobile and wearable devices. These kind of devices most often comprise System-on-Chip processor cores and further hardware accelerators. A novel heterogeneous hardware architecture introduced by various FPGA manufacturers consists of a programmable FPGA like structure and a common RISC processor core. For system designers this commercial architecture enables enhanced flexibility in partitioning of algorithmic tasks. The hardware demonstrator for auditory feedback of movements (sonification) captured by multiple inertial measurement units proposed in this paper bases on a heterogeneous Xilinx Zynq System-on-Chip processing core and a custom hardware accelerator. Energy efficiency is enhanced by utilizing the hardware accelerator for orientation estimation based on a Kalman filter algorithm. The evaluation furthermore explores the usability of High Level Synthesis tools based on a fixed-point software implementation. Moreover, the area and power consumption of hardware accelerator ASIC implementations based on a 40 nm TSMC library are evaluated.**

Hardware accelerator; inertial sensor fusion; energy efficient; system on chip; FPGA; High Level Synthesis;

I. Introduction

Mobile motion capturing is a mandatory requirement in various applications in medical rehabilitation [1] or virtual reality [2]. Highly reliable camera-based systems require a complex stationary setup and require a permanent line of sight between the tracked objects and multiple cameras. In contrast, mobile and wearable inertial sensors can be easily attached to the human body [2]. This is essential for human motion capturing in sport training sessions or medical applications.

In human motion capturing, common algorithmic approaches for orientation estimation like integration, vector observation, complementary filtering, or Kalman filtering are applied. These algorithms perform orientation estimation based on RAW data acquired by tri-axial gyroscopes, accelerometers and magnetometers. A previous study showed the accurate, and long-term drift-free orientation estimation based on IMU data of Kalman filter based sensor fusion techniques [3].

The target application of the proposed heterogeneous System-on-Chip (SoC) is auditory movements feedback captured by wearable, wireless inertial measurement units (IMU) in rehabilitation sessions [4]. Recent research showed that patients in stroke rehabilitation remarkably benefit from this so called sonification of movements [5]. In common stroke rehabilitation therapy training tasks are for example drinking from a water glass to regain a maximum independency within the daily live of the patients. Wearing multiple IMUs enables capturing of complex movements. Parameters like angles between body segments, positions or velocities can be computed based on a connected rigid chain body model. To enable home based rehabilitation the sonification demonstrator has to be battery powered and wearable. Therefore the reduction of overall power consumption is a major design goal.

Exemplarily the heterogeneous Xilinx Zynq FPGA fabric / RISC SoC is used to demonstrate the system integration of an energy efficient custom hardware accelerator for the highly computational demanding orientation estimation. The architecture of the sonification demonstrator is shown in Figure 1.

A basic requirement for the hardware accelerator design is a fixed-point implementation of the applied Kalman filter algorithm according to Lee and Park [6]. The proposed Application Specific Instructionset Processor (ASIP) in [7] achieves low execution times, but does not consider energy consumption. Therefore, an energy efficient hardware accelerator using a 32 bit fixed-point number representation is designed based on the results of a comprehensive fixed-point analysis [8].

An emerging approach for the design of hardware accelerators is the use of High Level Synthesis (HLS) tools like Calypto Catapult C or Xilinx Vivado HLS. Based on a C or C++ fixed point implementation these tools generate a hardware architecture considering design goals like area or speed.

A picture of the utilized hardware platform with the attached wireless IMUs is shown in Figure 2.

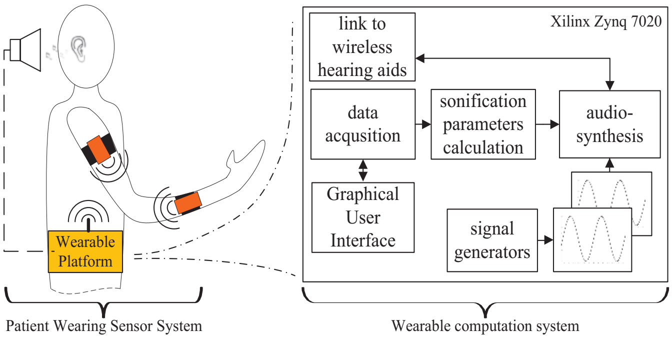
Xilinx Zynq 7020
link to wireless hearing aids
data acqusition
sonification parameters calculation
audio-synthesis
Graphical User Interface
signal generators
Wearable Platform
Patient Wearing Sensor System
Wearable computation system

Figure 1. Movement sonification demonstrator architecture

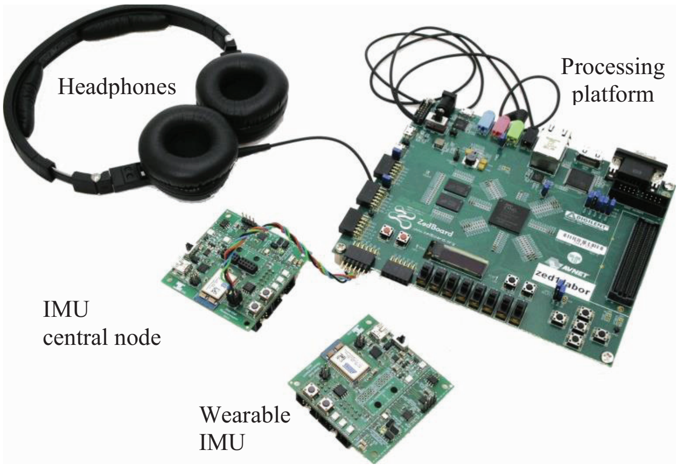

Figure 2. Picture of the Xilinx Zynq hardware platform with attached IMUs

The paper is organized as follows: Section II introduces movement sonification. Section III presents High Level Synthesis tools design concepts. Section IV explains the transformation of a floating-point based number representation to fixed-point. Section V highlights the hardware platform architecture. The proposed hardware accelerator architecture is illustrated in Section VI. Section VII presents evaluation results. Conclusions are drawn in Section VIII.

II. Interactive Movement Sonification

The goal of interactive movement sonification in stroke rehabilitation is to substitute the impaired proprioceptive sense of stroke patients by auditory feedback. A major design goal for real-time, interactive movement sonification is overall system latency below 30 ms. Higher latency values result in recognizable differences in visual and proprioceptive sense and thus the feedback gets annoying for humans [9]. In general, movement sonification arises from sports science applications to enhance the motion learning, like auditory feedback of [10] jump heights and real-time feedback in rowing [11].

Main processing tasks in interactive movement sonification are orientation estimation to provide reliable movement data, mapping of computed movement parameters to auditory feedback features, and sound synthesis for auditory feedback.

For example, the sonification displays the wrist position in relation to the patient's shoulder joint based on customizable parameter mappings. The polar angle of the wrist position in spherical coordinates influences the sound generators frequency. The left and right channel volume is modulated based on the azimuth angle. The overall volume is scaled dependent on the radius. Different basic sound synthesis toolkit (STK) [12] sound generators, like sine generators and artificial instruments, are used for sonification. Further movement parameters like wrist velocity are used to fade down the loudness when the patient is in a resting state.

Dependent on the chosen sound generator the pleasantness, intuitive understandability, and computational demands of the audio sample generation change and an individual adaption to the patients needs is achieved. For application in stroke rehabilitation orientation estimation accuracy comparable to human perception is targeted. According to [13] differences of below 2° to 3° are not noticeable for unimpaired people.

To allow home based rehabilitation and out of lab training sessions a fully mobile and portable system is desired. Therefore, a reliable mobile movement capturing and an energy efficient, low-power computational platform are mandatory.

III. High Level Synthesis

High Level Synthesis (HLS) is an emerging technique to combine the flexibility and short design cycles of software development and the energy efficiency and data throughput of dedicated hardware accelerators.

Calypto Catapult C and Xilinx Vivado HLS are two commercial state-of-the-art tools. Both software suites enable the generation of hardware accelerators based on C or C++ implementations. While Vivado HLS is limited to netlist generation for Xilinx FPGAs, Catapult C generates a register transfer level hardware description. Therefore, Catapult C provides more flexibility and especially the ability to perform an ASIC synthesis based on the generated HDL description.

The feasibility and efficiency of software based hardware accelerator generation for complex algorithms like a Kalman filter for IMU data fusion is analyzed in this paper. Therefore, the resulting resource and energy consumption of the custom hardware accelerator are compared to the results of both HLS software tools.

IV. Fixed-Point Transformation

For evaluation of the required minimal total bit-width and the number of integer and fraction bits for each Kalman filter variable a template based C++ framework, described and applied in [14], is used. The framework enables code re-usage at data-type level by abstracting the data-type.

Therefore, the floating-point data type (e.g. float or double) has to be replaced for analysis by the frameworks template data type. Setting the template data-type to float or double results in a floating-point reference implementation. For fixed-point analysis the template data-type has to be replaced by a hybrid floating-point or integer type (e.g. int32_t, int64_t) and parameters specifying total bit-width and fixed-point position.

An iterative design space exploration is performed, varying total bit-width and fixed-point position. An additional data-type implemented in the framework enables accuracy assessment and a comparison to floating-point reference implementation.

V. Hardware Architecture

In general, the Xilinx Zynq SoC comprises an FPGA fabric called **Programmable Logic** (PL) and a RISC processor part called **Processor Subsystem** (PS). The proposed processing platform utilizes a Xilinx Zynq 7020 SoC comprising an Xilinx Series-7 FPGA fabric and an ARM Cortex A9 dual core RISC processor at 667 MHz. The FPGA fabric provides 53,200 lookup tables (LUT), 106,400 Registers, 140 RAM blocks of 36 kBit (RAMB36) and 220 DSP slices. Providing rather as much resources as the largest Artix-7 SL or SLT FPGA.

The proposed hardware demonstrator based on the Zynq SoC features motion capturing and sonification utilizing the

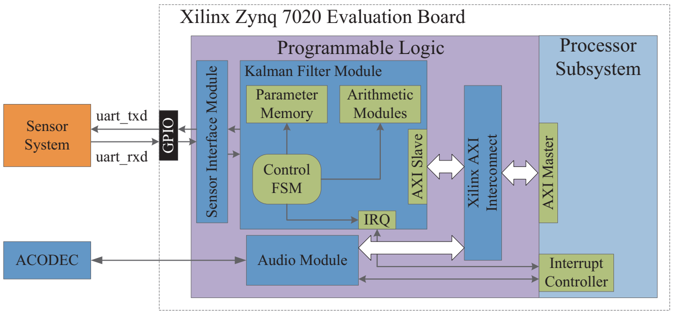

Figure 3. Hardware demonstrator architecture

Zedboard [15] development kit. Thereby, the feasibility of system integration and enhancing energy efficiency by dedicated hardware accelerators is demonstrated.

In general, the hardware demonstrator can be split up in three parts: the IMU **sensor system**, **hardware accelerators** residing in the FPGA fabric for communication and orientation estimation and software based processing for sound synthesis and movement feature calculation on the **processor subsystem**. Figure 3 shows the utilization of the Xilinx Zynq SoC within the movement sonification demonstrator, the on-board audio codec, and the attached external IMU sensors.

A. *Sensor System*

Commercial IMUs lack platform independent, low-level protocols and interfaces. Therefore, a custom IMU featuring a low-level TDMA based protocol and an RS-232 serial interface of the sensor master was designed. The IMUs constitute a star network topology whereby the master device acts as central node with an interface to the computation platform.

To achieve highest sampling rates while preserving low-power wearable IMU nodes, sensor fusion has to be performed on the processing platform. The custom IMU enables RAW data sampling rates of up to 512 Hz.

Utilizing ten IMUs, the processing platform has to perform orientation estimation at up to 5,120 samples per second.

B. *Xilinx Zynq Development Kit*

The ZedBoard development platform comprises a Xilinx Zynq 7020 SoC and a variety of external interfaces like HDMI, GPIO headers, and an audio codec.

Within the sonification demonstrator design, the IMU master node is attached via the GPIO headers. The onboard audio codec enables auditory feedback via stereo speakers.

Xilinx Zynq Programmable Logic

The AXI Bus interconnect enables communication between the Programmable Logic (PL) and the Processor Subsystem (PS). Intended application of the Zynq SoC PL subsystem is the design and integration of custom hardware modules to enhance throughput or energy efficiency or to reduce computational latency of execution time critical tasks.

Xilinx Zynq Processor Subsystem

The Processor Subsystem (PS) comprises an ARM Cortex A9 dual core RISC processor. In general, software based processing of algorithms achieves increased flexibility and short redesign cycles compared to custom hardware accelerators.

VI. Hardware Accelerator

A. *Architecture*

The application demands real-time processing of up to ten IMUs sampled at 512 Hz. Therefore, the processing time of a single Kalman filter iteration must not exceed 195.3 µs. This requirement results in a maximum processing time of the custom hardware accelerator module, residing in the Zynq SoC PL subsystem, of 19,530 cycles at 100 MHz.

According to the block diagram in [6] the custom hardware accelerator is subdivided into six sequential steps, realized as algorithmic sub-modules. Thereby, complexity of design and verification is reduced. Concerning the applications execution time requirements resource optimized design was focused. In adaption to increased throughput requirements a pipelined design can be established by introducing pipeline stages between the sub-modules. Figure 4 highlights the architecture of the custom Kalman filter hardware accelerator.

Resource minimization is achieved by sequential execution of the algorithm and transferring resource intensive algorithmic operations into shared dedicated modules. Therefore, modules for ADD, SUB, MUL and inverse square root are introduced.

The necessary Kalman filter parameters reside in a 36 Kbit dual port Block-RAM of the Zynq PL subsystem. One port is connected to the AXI Bus, while the other is used by the sub-modules to load and store required parameters. The Sensor Interface, realized as a separate hardware-module copies sensor RAW data into the Block-RAM using the AXI Bus interface. The resulting processed orientation estimation for each sensor is also written into the Block-RAM. The Zynq PS AXI Bus master is connected to the modules AXI interface and enables access to the Block-RAM data.

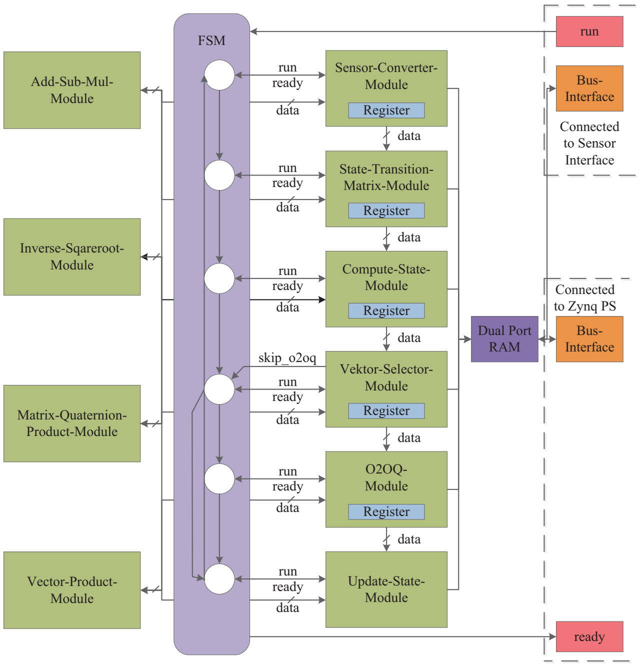

Figure 4. Hardware accelerator architecture

978-1-4673-5221-5/13/$31.00 ©2013 IEEE

The global dataflow is controlled by a Finite State Machine (FSM), also handling the dataflow between the algorithmic sub-modules by start / stop handshake signals. Due to the separation of arithmetic operations into sub-modules the state machine furthermore controls the allocation of the arithmetic sub-modules. The access to the algorithmic sub-modules is realized by multiplexing the input and output signals of the sub-modules. A global ready signal indicates a completed processing. This signal is also used as interrupt signal for the Zynq PS subsystem.

FSMs within the algorithmic sub-modules are used to implement the data-flow of the Kalman filter algorithm. Output registers are used to store the results of each arithmetic sub-module. Furthermore, these registers can be used as pipeline-registers, if demanded to achieve increased throughput.

B. *Dataflow*

Within the dataflow of the Kalman filter algorithm first a conversion of raw sensor data into the internally used fixed-point number representation is performed. This task is done by the **Sensor-Converter-Module**.

In the following **State-Transition-Matrix-Module** the state-transition-matrix based on the actual gyroscope data is computed.

This matrix is used in the **Compute-State-Module** to compute the a priori orientation estimation by multiplying the state-transition matrix with the previous orientation quaternion. The resulting normalized quaternion represents the actual orientation estimation based on the gyroscope data.

According to a range check of the input data, an adjustment of accelerometer and magnetic field vector is performed in the **Vector-Selector-Module**, to prevent the o2oq algorithm from unreliable data. The reliability of the data from accelerometer and magnetic field sensor is determined by threshold values according to [6]. If unreliable input data is detected, an approximation based on reference vector and the a priori orientation estimation is used as o2oq input.

The **O2OQ-Module** computes the orientation estimation based on accelerometer and magnetic field sensor data. In order to do this, the optimal two-observation quaternion estimation method called O2OQ [16] is used.

In the **Update-State-Module** the a posteriori orientation estimate is determined. The a priori orientation estimation, computed by the Compute-State-Module is corrected with the O2OQ based orientation estimation using the pre-computed Kalman gain matrix.

VII. Evaluation

The presented evaluation results compare the custom hardware accelerator to software based processing on the Zynq PS subsystem and hardware accelerators generated by the commercial HLS tools Calypto Catapult C and Xilinx Vivado HLS.

Evaluation objectives are the achievable orientation estimation accuracy, the execution time, resource consumption, and energy efficiency.

A. *Orientation Estimation Accuracy*

A reduction of computational complexity of the Kalman filter by using pre-computed a posteriori and a priori error covariance matrices is proposed in [7], [17]. The presented evaluation results show a negligible influence on the orientation estimation accuracy of the filter modification.

Using pre-computed matrices results in less required operations enabled by knowledge about constant sensor noise characteristics. The main effect is achieved by constant Kalman gain matrix due to the pre-computed error covariance matrices The constant Kalman gain matrix achieves a high reduction of the computational demands, as the computation of a 4x4 matrix inverse in each filter step is needless.

TABLE I. gives a comparison of floating-point operations required for a single Kalman filter step of the modified Kalman filter and the initial filter algorithm.

TABLE I. Number Of Floating-Point Operations Per Filter Step

Operation		‘+’, ‘-‘	‘*’	‘/’	Arc Cos
O2OQ		147	197	31	1
Kalman Filter	Original	579	524	46	0
	Modified	60	37	20	0

The orientation estimation of the modified filter differs 0.08 °RMS, 0.01 °RMS and 0.19 °RMS (roll, pitch, and yaw) from the original algorithm. Providing an absolute accuracy of 2.36 °RMS, 2.50 °RMS and 6.43 °RMS (roll, pitch, and yaw) compared to data captured using a Qulisys Oqus optical motion capturing system. In general, the influence on accuracy due to the algorithmic modification is negligible while highly reducing the number of operations required.

For verification of the custom hardware accelerator and the HLS tools generated accelerators a bit-true software model is used. In contrast to the fixed-point implementation of the algorithm used by the hardware accelerators, the Zynq PS subsystem performs orientation estimation based on a floating-point algorithm, as this is most suitable for this processor.

TABLE II. compares the accuracy of the floating-point based algorithm to the accuracy of the fixed-point version. The floating-point version achieves higher accuracy, while the fixed-point version still meets the application requirements.

Due to the 32 bit data representation the resource consumption is minimized, resulting in a slightly increased estimation error. Furthermore, the bit true fixed-point software was used to generate hardware accelerators by the HLS tools.

TABLE II. Kalman Filter Accuracy

Orientation Error / [ºRMS]	Roll	Pitch	Yaw	Average
Floating-point reference	0.96	0.84	4.53	2.11
Fixed-point implementation	2.04	3.57	3.14	2.92

978-1-4673-5221-5/13/$31.00 ©2013 IEEE

B. Custom Hardware Accelerator Ressources

A major design aspect for hardware accelerators is resource consumption. Therefore, FPGA synthesis results of the custom hardware accelerator and the results of the HLS tools are compared in TABLE III.

TABLE III. COMPARISON OF FPGA RESOURCES CONSUMPTION AND LATENCY AT 100 MHZ SYSTEM FREQUENCY

Resources	Registers	LUTs	DSP slices	RAMB36	Execution time / cycles
Catapult C	10,532	14,550	38	0	829
Vivado HLS	12,663	20,104	227	3	1,344
Custom module	4,395	7,035	1	1	1,623

The application latency requirements, are a processing of a single Kalman filter cycle within 19,530 clock cycles, are exceeded by all implementation. Therefore, resource minimization is focused. The HLS tools were optimized for reducing resource consumption by modifying the default settings. Within the proposed custom hardware accelerator resource minimization was achieved by performing a sequential processing and shared arithmetic operation modules.

The custom module achieves minimal resource consumption, while meeting the execution time requirements. Comparing the HLS tool results, Catapult C behaves best, as resources and execution time are less than the Vivado results.

C. Kalman Filter Energy Efficiency

The objective of a hardware accelerator for Kalman filter based IMU sensor fusion is increased energy efficiency. Therefore, TABLE IV. compares the power dissipation, and execution time of different hardware architectures.

The processor cores Intel Core i5 and ARM Cortex A9 achieve lowest latency, while having maximum power consumption. For comparison, a low-power Atmel 32 bit microcontroller having the maximum execution time is listed.

TABLE IV. COMPARISON OF KALMAN FILTER ENERGY CONSUMPTION

Processing Core	Power Dissipation / [mW]	Execution Time / [µs]	Energy per Filter Cycle / [nJ]
Intel Core i5 760	95,000.00	0.47	44,650
ARM Cortex A9	711.0	4.11	2,922
Atmel AT32UC3A	70.00	853.56	59,750
Custom Module (Zynq 7020 / Programmable Logic)	157.00	16.23	2,550
Catapult C Module (ASIC Synthesis / 40 nm TSMC)	91.94	8.29	760
Custom Module (ASIC Synthesis / 40 nm TSMC)	1.71	16.23	30

Performing an ASIC synthesis based on a 40 nm TSMC library the HLS tool Catapult C generated hardware accelerator results in an area of 0.144 mm^2 and a power dissipation of 91.94 mW. The custom accelerator area is 0.146 mm^2 and the power dissipation 1.71 mW. Compared to Catapult C based accelerator the custom accelerator is about 25 x more energy efficient while nearly preserving the area consumption.

The major goal within the custom hardware accelerator design is increased energy efficiency. Therefore, a measure considering the different clock frequencies, processor instruct-tion-sets and silicon technologies is energy per Kalman filter cycle. Energy consumption is calculated according to the power dissipation and execution time shown in TABLE IV. Figure 5 shows the comparison among the different processing architectures normalized to the proposed Kalman filter hardware accelerator.

Utilizing the PL subsystem of Zynq SoC improves Kalman filter based orientation estimation energy efficiency is 12.7 %, as shown in Figure 5. Due to the excessive resource consumption of the HLS tool generated accelerators, these modules were not considered in the energy efficiency comparison.

Furthermore, Figure 5 shows the improved energy efficiency of the ARM Cortex A9 RISC processor in comparison to the Intel Core i5 general purpose processor.

In general, a dedicated silicon implementation of the hardware accelerator achieves highest efficiency. Still the HLS tool Catapult C shows a non optimal result compared to the custom hardware accelerator.

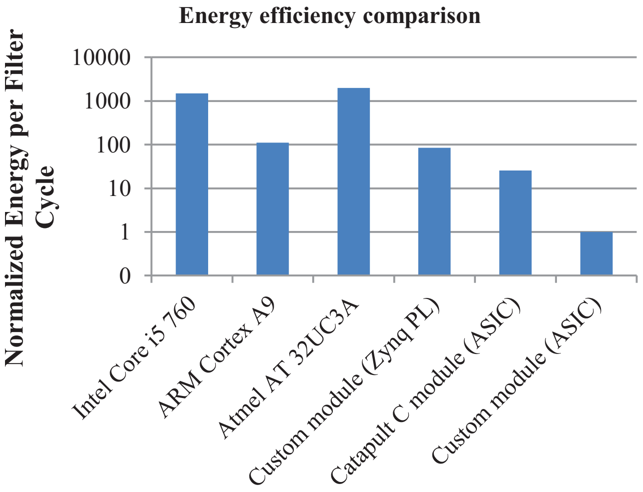

Figure 5. Energy efficiency comparison

In summary, the custom hardware accelerator residing in the Zynq PL subsystem improves energy efficiency by a factor of 1.14 compared to the ARM Cortex A9 RISC processor and moreover a reduction of the processor cores computational requirements is achieved. Compared to the general purpose processor Intel Core i5 energy efficiency is improved by a factor of 17.5.

The results of the ASIC synthesis clarify the potential of a fully customized heterogeneous SoC utilizing an accelerator for orientation estimation. Compared to the ARM Cortex A9 the accelerator improves energy efficiency by a factor of 111.

VIII. Conclusions

The proposed Kalman filter hardware accelerator highlights the design options of heterogeneous SoCs comprising FPGA fabrics and processor cores. Emerging commercial devices like the Xilinx Zynq and the Altera SoC FPGA family provide extended design partitioning flexibility for a wide range of applications. Furthermore, the energy efficiency improvement is mostly transferable to heterogeneous architectures comprising softcore processors and custom hardware accelerators.

Although HLS tools allow short development times for hardware accelerators, the presented results show the current inapplicability of these tools in generating highly efficient accelerators. Furthermore, performing a fixed point analysis by external tools is still required, increasing the overall development time.

The FPGA based hardware accelerator implementation increases energy efficiency by a factor of 1.14 compared to the ARM Cortex A9 processor core, additionally offloading computational requirements from the processor core. A dedicated ASIC implementation based on a 40 nm TSMC library could increase energy efficiency by a factor of 111.

Furthermore, the architecture of the proposed hardware accelerator combined with the heterogeneous SoC provides enhanced flexibility and scalability. Throughput can be increased by a factor of six by introducing pipeline stages. Even higher throughput rate requirements can be fulfilled by multiple instantiations of the hardware accelerator within the FPGA fabric of the SoC.

In general, the flexibility of heterogeneous FPGA fabric / RISC processor core SoC provides an increment value for system designers. The applicability in interactive movement sonification for stroke rehabilitation is demonstrated by the proposed hardware demonstrator design.

The Xilinx Zynq 7020 power dissipation within the proposed interactive sonificiation application is 1.5 W, estimated by the Xilinx Power Estimation spreadsheet. Therefore, such SoCs fulfill the power consumption requirements for mobile usability in a wide range of applications.

References

[1] E. R. Bachmann, R. B. McGhee, X. Yun and M. J. Zyda, "Inertial and magnetic posture tracking for inserting humans into networked virtual environments," *VRST '01: Proceedings of the ACM symposium on Virtual reality software and technology,* 2001.

[2] H. Zheng, N. Black and N. Harris, "Position-sensing technologies for movement analysis in stroke rehabilitation," *Position-sensing technologies for movement analysis in stroke rehabilitation,* 2005.

[3] H.-P. Brückner, C. Spindeldreier, H. Blume, E. Schoonderwaldt and E. Altenmüller, "Evaluation of Inertial Sensor Fusion Algorithms in Grasping Tasks Using Real Input Data," *Wearable and Implantable Body Sensor Networks (BSN), Ninth International Conference on,* pp. 189-194, 2012.

[4] H.-P. Brückner, M. Wielage and H. Blume, "Intuitive and Interactive Movement Sonification on a Heterogeneous RISC / DSP Platform," *The 18th Annual International Conference on Auditory Display,* pp. 75-82, 2012.

[5] I. Wallis, T. Ingalls, T. Rikakis, L. Olson, Y. Chen, W. Xu and H. Sundaram, "Real-Time Sonification of Movement for an Immersive Stroke Rehabilitation Environment," *The 13th International Conference on Auditory Display,* 2007.

[6] J. K. Lee and E. Park, "A minimum-order kalman filter for ambulatory real-time human body orientation tracking," *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on,* 2009.

[7] S. Sabatelli, M. Galgani, L. Fanucci and A. Rocchi, "A Double-Stage Kalman Filter for Orientation Tracking With an Integrated Processor in 9-D IMU," *Instrumentation and Measurement, IEEE Transactions on,* vol. 62, no. 3, pp. 590-598, 2013.

[8] H.-P. Brückner, C. Spindeldreier and H. Blume, "Modification and fixed-point analysis of a Kalman filter for orientation estimation based on 9D inertial measurement unit data," *Engineering in Medicine and Biology Conference,* 2013.

[9] D. Levitin, K. MacLean, M. Mathews, L. Chu and E. Jensen, "The perception of cross-modal simultaneity," *International Journal of Computing Anticipatory Systems,* 2000.

[10] A. O. Effenberg, "Movement sonification: Effects on perception and action," *IEEE Multimedia,* vol. 12, no. 2, pp. 53-59, 2005.

[11] N. Schaffert and K. Mattes, "Positive influence of acoustic feedback for adaptive elite athletes in rowing," *18th annual Congress of the European College of Sports Science,* p. 348, 2013.

[12] G. Scavone and P. Cook, "RtMidi, RtAudio, and a synthesis toolkit (STK) update," *In Proceedings of the International Computer Music Conference,* 2005.

[13] H. Tan, J. Radcliffe and et.al., Human Factors For The Design Of Force-Reflecting Haptic Interfaces, Dynamic Systems and Control, DSC, Vol. 55, no. 1, 1994.

[14] M. Pfitzner, F. Cholewa, P. Pirsch and H. Blume, "Close-to-hardware error analysis for real-time wavenumber domain processing," *RADAR 2012, 7th International Conference on Radar,* 2012.

[15] ZedBoard Development Kit, [Online]. Available: http://www.zedboard.org/.

[16] F. Markley, "Fast quaternion attitude estimation from two vector measurements," *Journal of Guidance, Control, and Dynamics,* vol. 25, pp. 411-414, 202.

[17] S. Sabatelli, F. Sechi, L. Fanucci and A. Rocchi, "A sensor fusion algorithm for an integrated angular position estimation with inertial measurement units," *Design, Automation Test in Europe Conference Exhibition (DATE),* 2011.

978-1-4673-5221-5/13/$31.00 ©2013 IEEE