

Embedded Parallel Computing Accelerators for Smart Control Units of Frequency Converters

Steffen Vaas, Marc Reichenbach, Johannes Hofmann, Thomas Stadelmayer and Dietmar Fey

Department of Computer Science, Chair of Computer Architecture

Friedrich-Alexander-University Erlangen-Nürnberg (FAU), Germany

{Steffen.Vaas, Marc.Reichenbach, Johannes.Hofmann, Thomas.Stadelmayer, Dietmar.Fey}@fau.de

Abstract—

Classical frequency converters are designed as embedded devices optimized for a specific application-field. But in times of *Industry 4.0* simple frequency converters change to smart control units and become more intelligent with analysis and reporting functions to build up smart grids in automation systems for reducing maintenance costs and increasing productivity. To realize these new functions, an evaluation is needed, which kind of computer architectures should be used for these new devices. Due to more complex algorithms, classical microcontrollers are not sufficient anymore. Therefore, we show in this paper, if and how microprocessors in smart control units can benefit from highly parallel hardware accelerators. Consequently, we propose to increase the performance of an *ARM Cortex-A9* processor by using an *Epiphany III E16* many-core processor as hardware accelerator for complex analysis tasks. Our results show, that a speedup of 1.78 can be achieved, while the power consumption is increased by only 9%.

Index Terms—Smart Sensors, Many-Core Processor, Heterogeneous Processor Architecture, Epiphany, Parallella, Embedded Preprocessing

I. INTRODUCTION

For the control of most electrical engines frequency converters are needed. Therefore, the standard solution is to use embedded microcontrollers executing simple PID regulators. Depending on the application area of the electrical engines, there are different requirements for the control algorithms regarding performance, latency and power. Especially the embedded hardware of control units produced in quantity should be just powerful enough to meet the minimum requirements of a stable regulator. In other application-fields like power generator systems, they must be highly accurate to guarantee optimal control. Otherwise power would only be fed inefficiently to the grid. Moreover, robot arms need a low latency for fast movements. Beside controlling of electrical engines, frequency converters can also monitor them. To fulfill the requirements of the *Industry 4.0* a paradigm shift can be seen from simple frequency converters to more powerful smart control units with advanced analysis functions. Simple functions include the determine whether an engine is running or the analysis of different voltage levels. A more complex example is the spectrum analysis of the voltage and current waveforms, for example to detect failures of engines in a very early state, as it was shown in [1] to monitor electrical engines on a ropeway. Thus, maintenance costs can be reduced by additional monitoring information. This

way, parts in an automation system that otherwise become defective and interfere with production, can be identified and replaced in time. Furthermore, monitoring information can be provided actively by the frequency converters itself within big automation systems. Another optimization, which can be done by power analysis is feature extraction. Instead of transmitting all acquired raw samples for monitoring, the amount of data to transmit can be reduced significantly by preprocessing the raw values already in the sensor. Thus, only the state of the according electrical engines has to be transmitted. Such *smart sensors*, which combine data acquisition and data preprocessing, can be the solution to realize big, complex systems. To execute all these additional tasks on the power controller, more computing power will be needed.

To gain more computing power, different approaches from the view of computing architectures are possible. Because most frequency converters uses only simple microcontrollers (e.g. ARM Cortex A9), a replacement with a more powerful device (e.g. ARM Cortex A57) is possible. Although this way seems to work, the new device becomes more cost intensive and is also “overpowered” if only a legacy frequency controller is needed. Therefore, we propose in this paper the usage of parallel hardware accelerators. Using this methodology, it is possible to optionally put computational power to a existing device to make it more intelligent or even smarter. In summary, a possible approach is to use a common platform with minimum required hardware resources to provide the main functionalities, like PID controls, which are always needed for every application. Furthermore, there are free slots, where additional accelerator cores can be mounted. These can be necessary to implement more complex filter operations, reducing latency by processing multiple channels in parallel, executing more complex regulator algorithms and the concurrent signal analysis of several channels. Thereby, it is possible to cover a wider application area and reducing the development, as well as the production costs. An abstract view how the described system could look like is shown in Fig. 1.

The goal of this paper is to determine if and how hardware accelerators can help to put more smartness into power control units. As starting point of this work, we have chosen the Epiphany-III many core processor with 16 RISC cores. Due many different channels which have to be processed in power control units, this chip seems to be a good choice for processing acceleration. Moreover it has a low power consumption

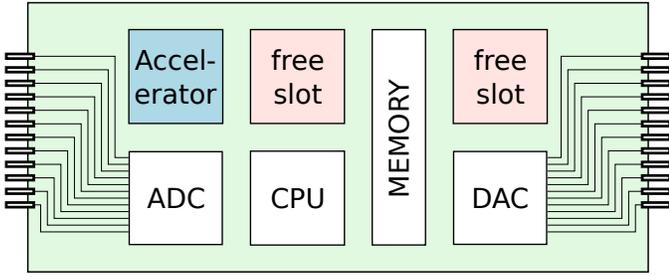


Fig. 1. Platform for an extendable frequency converter control unit.

and due to the Network-on-Chip architecture with off-chip connections, the computation power could be easily increased by putting multiple devices together. Finally, the Epiphany processor is programmable with high-level languages like C/C++ and does not require detailed hardware design knowledge as it is needed when using for example FPGAs as accelerators.

This paper is structured as follows. Section 1 discussed the need of smart control units and explains the need of new architectures for these devices. In the next section some work is presented, which relates to this topic. In Section 3 the algorithms, which run on such a smart control unit are analyzed in more detail. The implementation is described in Section 4. Section 5 presents an evaluation when the usage of hardware accelerators is useful. We will finish this paper with a conclusion and a short outlook.

II. RELATED WORK

Most controllers of frequency converters were designed as highly optimized solutions to reach the needed performance. In [2], for example, the system is optimized for a small scaled robot, a controller for quadcopters was shown in [3] and a PID using fuzzy logic was presented in [4]. Thus, it is difficult to find a common platform to save time and costs in development.

To gain more flexibility and cover a wider application area, a framework was presented in [5]. Another approach utilizing *Simulink* was shown in [6]. In [7] an application-specific instruction set processor (ASIP) was developed to increase flexibility. However, all these proposals are only useful for the same kind of algorithms, like PID controls.

However, there are also control algorithms, which differ drastically from another. For fuzzy logic controllers [8], there are completely different requirements on the hardware than on self-tuning regulators [9]. Moreover, for supporting multiple channels an ability to execute those channels in parallel is needed. In [10] a robot arm with six degrees of freedom was presented, multiple channels have to be controlled in parallel. In [11] even a combination of self-organizing regulators using multiple channels was presented. The presented frameworks and tools are not flexible enough to cover such wide application-fields.

One approach to execute nearly all applications is to utilize FPGAs. As shown in [12], configurable hardware offers large

flexibility, so the hardware can be reused. In [13] a combination of a microcontroller and a FPGA was presented to increase the performance by an accelerator core. However, a single architecture, can either be designed for high performance or low power application-fields. Moreover, applying a hardware description language, which is needed to configure FPGAs, increases the development time.

Requirements on the embedded can strongly differ. Beside applications using small scaled low-energy microcontrollers like in [14] and [15], there are other ones requiring more performant hardware, as in [16]. There a DSP serves as processing unit for high performance motor drives.

To cover also these different types of applications, a simple programmable and extendable architecture is required. So our approach is to use a standard embedded architecture, like an ARM processor as main control unit and many-core architectures as accelerators, which can be mounted optionally. A suitable many-core processor is the *Epiphany III E16* from *Adapteva*. *Zain Ul-Abdin* demonstrated the performance of the *Epiphany* core by executing radar processing algorithms on it in real-time [17], [18]. Moreover, *Reichenbach et al.* used the *Epiphany* core to execute correlation functions in real-time [19]. So, this architecture will provide the needed performance to serve as optional mountable accelerator core to speedup control and analysis algorithms requiring more performance.

III. ALGORITHMS FOR CONTROLLING AND ANALYSIS

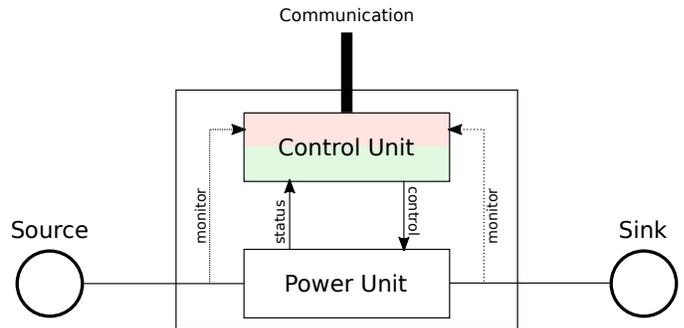


Fig. 2. Illustration of a frequency converter.

In Figure 2 a frequency converter is shown, converting frequency and voltage amplitude of the source signal. Thereby the frequency converter can be divided into a power electronics part, the *Power Unit* and into an embedded hardware controller, illustrated as *Control Unit*. To test an accelerator for a control unit, a reference design for a three phase current was implemented. This reference is presented in Figure 3. On the control unit two main applications are executed: the controlling, which processes the blocks in green and a monitoring application, executing the red marked block.

After sampling the voltage and current values of all 3 phases, filtering is needed at the control application, to achieve a stable regulation. To execute a PID algorithm, the gathered information of all channels has to be transformed into space vectors. Afterwards, another space vector modulation

is required to transform the data back. Then a pulse-width modulation (PWM), which is implemented as a peripheral of the microcontroller, is used to control the *Power Unit*.

For a simultaneous analysis while controlling the frequency converter, a *Monitoring* unit is implemented. There a compute-intensive FFT calculation is needed to get the spectrum of all sampled signals. This spectrum gets analyzed to detect failures in the system, such as damaged bearings of electric engines. These errors can then be reported by a communication interface.

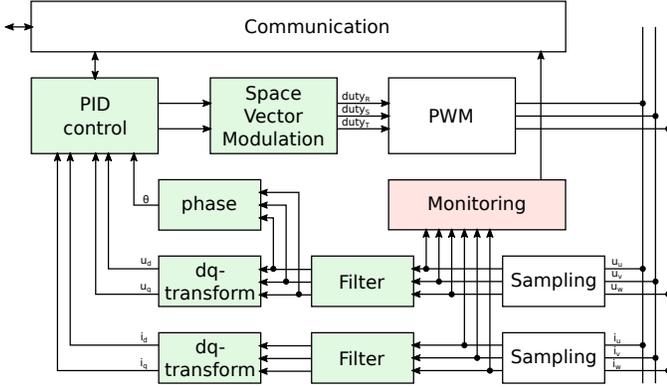


Fig. 3. Realized data flow within the control unit for frequency converters.

IV. RESULTS

To evaluate the algorithms on different architectures, the *Parallella* board from *Adapteva* was used as reference platform. The board owns an *Epiphany III E16* many-core processor and a *Zynq* system-on-chip core from *Xilinx* with an integrated *ARM Cortex-A9* dual core processor. To validate the approach of utilizing hardware accelerators for frequency converter control units, firstly all functionalities were implemented on the *ARM* processor. Then parts of the algorithms were executed on the *Epiphany* accelerator to ascertain if there is a speedup.

A. Speedup of the Execution time

Since the control algorithms of this reference design do not need much processing power and the code is only poor parallelizable, even on the two cores of the *ARM Cortex-A9*, there was no need to speedup the calculations by using an accelerator.

For the monitoring algorithm a FFT is required, which is a highly compute-intensive task. Thus, it was outsourced to the *Epiphany* accelerator for relieving the main CPU. To determine the speedup by an accelerator core, the algorithm was implemented for FFT calculations of 1, 4 and 8 channels in parallel. Thereby two different amounts of input values were considered, with 4096 and 16384 input values. The results are illustrated in Figure 4. For the implementation on the *ARM* core the *FFTW 3.2.2 ARM* [20] library was used. To increase the performance of the monitoring algorithm, the FFT and the peak detection was calculated on the accelerator

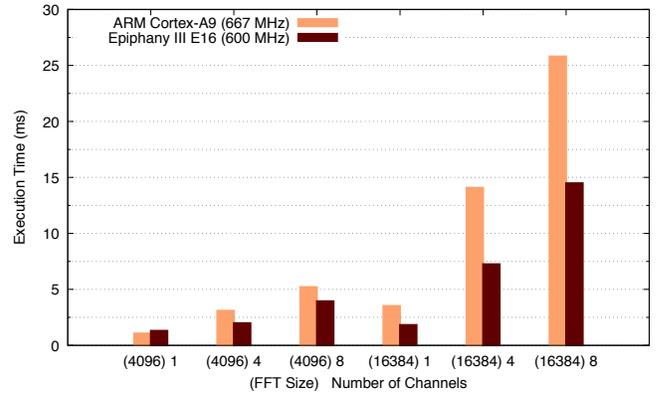


Fig. 4. Measured execution times of different monitoring algorithm configurations.

processor. On the 4096 FFT variant, 4 of 16 *Epiphany* cores were allocated for one FFT. So the monitoring algorithm for 1 channel utilizes the *Epiphany* only up to 25%, what resulted in a longer execution time. Only for the monitoring of 4 or 8 channels the core is utilized completely. On the FFT variant with 16384 input values, all 16 *Epiphany* cores were used for 1 FFT calculation. So the whole accelerator was already utilized completely for monitoring 1 channel. Monitoring multiple channels was also executed consecutively as on the *ARM* core. In this scenario, a speedup of up to 1.78 was reached.

B. Communication between Processor and Accelerator

With the *Epiphany* an even higher speedup could be reached theoretically, but measurements have shown, that the data transmission to and from the accelerator is the bottleneck of the system. The fastest way to transfer data between the *Cortex-A9* and the *Epiphany* is to use the shared memory of the *Parallella*. The memory read and write access times of both processors are shown in Figure 5 and Figure 6.

However, most tasks executed on the *Epiphany* are memory bound. Only computing-intensive tasks are suited to be executed on the accelerator keeping the amount of communication transfers between the processors low. Another way to increase the performance is to use a DMA controller, which moves raw data from input peripherals directly into the shared memory, as assumed in IV-A. Thus, the data transmission time can be reduced and the *ARM* processor gets relieved.

Due to high transmission times, for the reference design a benefit in performance is only given by outsourcing the compute-intensive parts of the analysis application. For the controlling task the execution time would rise, if the parts of the calculations would have to be transferred to the accelerator firstly. This was also estimated by a roofline model, which is shown in Figure 7.

C. Power Consumption

Apart from performance measurements, on embedded devices power consumption also has to be taken into account.

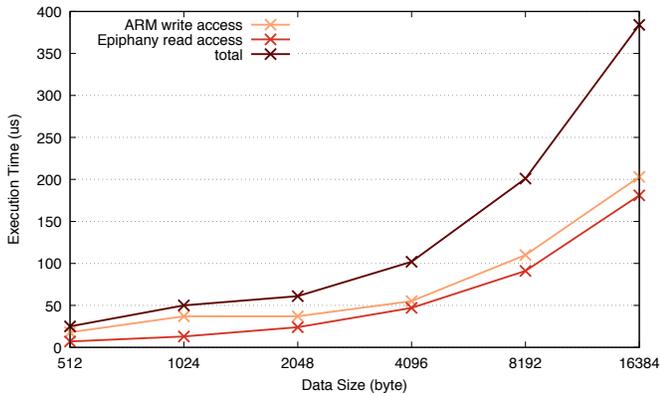


Fig. 5. Execution times to transfer data from *ARM Cortex-A9* to the *Epiphany* on the *Parallella* board.

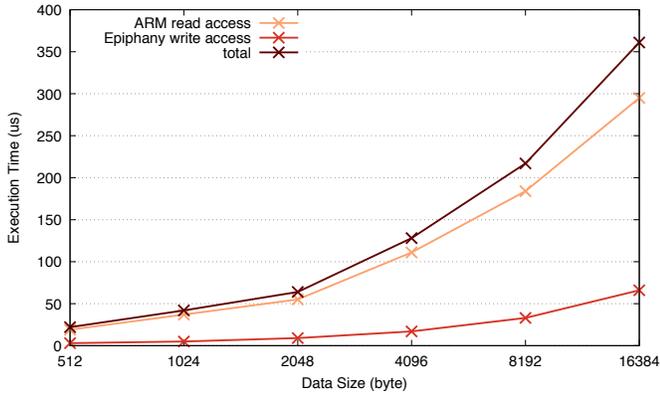


Fig. 6. Execution times to transfer data from the *Epiphany* to the *ARM Cortex-A9* on the *Parallella* board.

Therefore, the power consumption was measured during execution of all algorithms on the *ARM* processor and while the *Epiphany* was activated executing the monitoring algorithms. As illustrated in Figure 8 the consumption rose by 0.4 W. The results show, that the whole evaluation board needs 9% more power if the *Epiphany* is activated and the performance increases by 79%. Thus, only 60% of the energy is needed for the same calculation using the accelerator core.

V. CONCLUSION

To keep up with the requirements of *Industry 4.0*, power control units needs to become more intelligent with internal analysis functions. Therefore, new architecture concepts are necessary to fulfill these requirements. For a flexible and scalable solution, we have shown how hardware accelerators can be used to speedup calculation and allow complex data analysis already at the sensor.

In this paper we presented two applications, a PID controller as well as FFT for power monitoring which have to be executed simultaneously at a smart power control unit. The results show, that for a PID controller a standard *ARM 9* core is sufficient, while for more complex analysis operations a hardware accelerator is required. Using the *Epiphany-III* chip

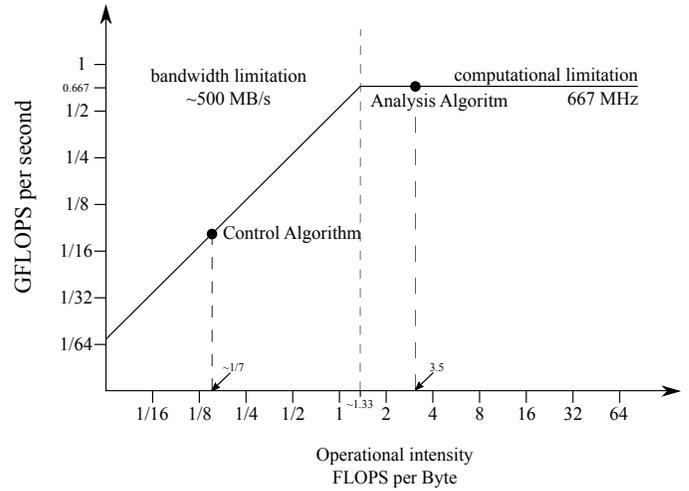


Fig. 7. Roofline Model

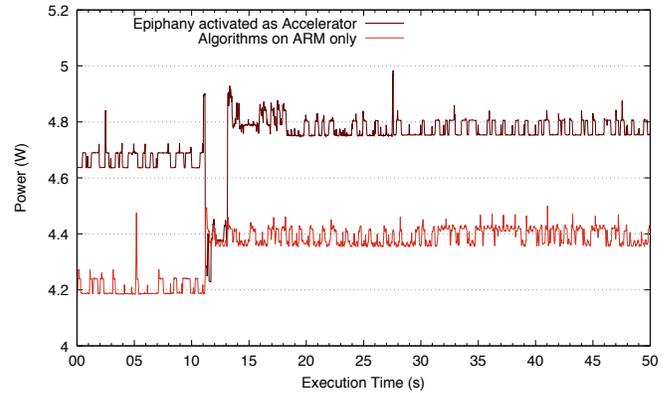


Fig. 8. Power measurement while executing all tasks on the *ARM* processor and enabling the *Epiphany* accelerator.

from *Adapteva*, a powerful high parallel embedded computing architecture is available which can do the FFT processing in real time and therefore free computational resources at the *ARM* core in favor of the PID control executed on it.

In this work, we first evaluated which architectures could be used for constructing new smart power control units. Using these results and the presented concept of a flexible and expendable architecture our next step is to realize an own PCB using an *ARM 9* core with the capabilities to put several hardware accelerators in this system.

ACKNOWLEDGMENT

The authors would like to thank *Andreas Gröger* and *Dr. Marvin Tannhäuser* from the *Siemens AG* in *Erlangen* for the meaningful discussions their valuable inputs.

REFERENCES

- [1] Schaeffler Technologies AG & Co. KG, "Ina deutschland — mediathek — mobile zukunft — seilbahnüberwachung mit dem fag smartcheck," <http://www.ina.de/content/ina.de/de/mediathek/library/library-detail-language.jsp?id=63364608&ref=rss>.

- [2] W. Zhao, B. H. Kim, A. Larson, and R. Voyles, "FPGA implementation of closed-loop control system for small-scale robot," in *12th International Conference on Advanced Robotics, 2005. ICAR '05. Proceedings*, Jul. 2005, pp. 70–77.
- [3] M. N. Duc, T. N. Trong, and Y. S. Xuan, "The quadrotor MAV system using PID control," in *2015 IEEE International Conference on Mechatronics and Automation (ICMA)*, Aug. 2015, pp. 506–510.
- [4] J. Guo, G. Wu, and S. Guo, "Fuzzy PID algorithm-based motion control for the spherical amphibious robot," in *2015 IEEE International Conference on Mechatronics and Automation (ICMA)*, Aug. 2015, pp. 1583–1588.
- [5] R. Frijns, A. Kamp, S. Stuijk, J. Voeten, M. Bontekoe, K. Gemei, and H. Corporaal, "Dataflow-Based Multi-ASIP Platform Approach for Digital Control Applications," in *2013 Euromicro Conference on Digital System Design (DSD)*, Sep. 2013, pp. 811–814.
- [6] J. Lazaro, A. Astarloa, J. Arias, U. Bidarte, and A. Zuloaga, "Simulink/Modelsim Simulabel VHDL PID Core for Industrial SoPC Multiaxis Controllers," in *IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics*, Nov. 2006, pp. 3007–3011.
- [7] D. Abramovitch, "A unified framework for analog and digital PID controllers," in *2015 IEEE Conference on Control Applications (CCA)*, Sep. 2015, pp. 1492–1497.
- [8] R. Kazemi, R. Vesilo, and E. Dutkiewicz, "A Novel Genetic-Fuzzy Power Controller with Feedback for Interference Mitigation in Wireless Body Area Networks," in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, May 2011, pp. 1–5.
- [9] X.-h. Liu and L. Xu, "Research on tension control system based on fuzzy self-tuning PID control," in *Control and Decision Conference (CCDC), 2010 Chinese*, May 2010, pp. 3385–3390.
- [10] J.-S. Kim, H.-W. Jeon, and S. Jung, "Hardware implementation of nonlinear PID controller with FPGA based on floating point operation for 6-DOF manipulator robot arm," in *International Conference on Control, Automation and Systems, 2007. ICCAS '07*, Oct. 2007, pp. 1066–1071.
- [11] H. Kazemian, "The SOF-PID controller for the control of a MIMO robot arm," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 4, pp. 523–532, Aug. 2002.
- [12] L. Barreto, P. Praca, C. Cruz, and R. Bascope, "PID Digital Control Using Microcontroller and FPGA Applied to a Single-Phase Three-Level Inverter," in *APEC 2007 - Twenty Second Annual IEEE Applied Power Electronics Conference*, Feb. 2007, pp. 1443–1446.
- [13] R. Alba-Flores, F. Rios-Gutierrez, and C. Jeanniton, "Qualitative evaluation of a PID controller for autonomous mobile robot navigation implemented in an FPGA card," in *2011 Seventh International Conference on Natural Computation (ICNC)*, vol. 3, Jul. 2011, pp. 1753–1757.
- [14] S. Sarin, H. Hindersah, and A. Prihatmanto, "Fuzzy PID controllers using 8-Bit microcontroller for U-Board speed control," in *2012 International Conference on System Engineering and Technology (ICSET)*, Sep. 2012, pp. 1–6.
- [15] C. Xu, D. Huang, Y. Huang, and S. Gong, "Digital PID controller for Brushless DC motor based on AVR microcontroller," in *IEEE International Conference on Mechatronics and Automation, 2008. ICMA 2008*, Aug. 2008, pp. 247–252.
- [16] A. Rubaai, M. Castro-Sitiriche, and A. Ofoli, "DSP-Based Implementation of Fuzzy-PID Controller Using Genetic Optimization for High Performance Motor Drives," in *Conference Record of the 2007 IEEE Industry Applications Conference, 2007. 42nd IAS Annual Meeting*, Sep. 2007, pp. 1649–1656.
- [17] Zain-ul-Abdin, A. Ahlander, and B. Svensson, "Energy-Efficient Synthetic-Aperture Radar Processing on a Manycore Architecture," in *2013 42nd International Conference on Parallel Processing (ICPP)*, Oct. 2013, pp. 330–338.
- [18] Z. Ul-Abdin and M. Yangt, "Dataflow programming of real-time radar signal processing on manycores," in *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec. 2014, pp. 15–19.
- [19] M. Reichenbach, M. Kasparek, M. Alawieh, K. Haublein, and D. Fey, "Real-time correlation for locating systems utilizing heterogeneous computing architectures," in *2015 Conference on Design and Architectures for Signal and Image Processing (DASIP)*, Sep. 2015, pp. 1–8.
- [20] Vesperix Corporation, "FFTW 3.2.2 ARM," <http://www.vesperix.com/arm/fftw-arm/>.