Quantitative Evaluation of an FPGA based Wireless Vibration Monitoring System in relation to Different Sampling Rates

Khurram Shahzad, Bengt Oelmann
Department of Electronics Design
Mid Sweden University
Sundsvall, Sweden
{khurram.shahzad, bengt.oelmann}@miun.se

***Abstract*— In order to achieve the high-processing performance required in typical computationally intensive high-sample rate monitoring applications, a Field Programmable Gate Array (FPGA) is often used as a hardware accelerator. Given the design complexity, increased power consumption and additional cost of an FPGA, it is desirable to determine the sampling rates for which the use of an FPGA as hardware accelerator results in most effective solution. For this purpose, a computationally intensive application is realized on an FPGA based architecture so as to determine the sampling rates for which it achieves the highest performance and consumes the least amount of energy as compared to that of a micro-controller based architecture. Based on the measured performance and energy consumption for a computationally intensive application, tri-axes/three-channel vibration based condition monitoring, the results suggest that the FPGA based architecture is the most appropriate solution for sampling frequencies of 4 kHz and above.**

Keywords- wireless monitoring, FPGA, sampling rate, high-sample rate, energy consumption, hardware accelerator

I. INTRODUCTION

Built on low-power, low-cost and miniature size electronics, battery operated wireless monitoring systems are emerging as a cost effective alternative to traditional wire-based monitoring solutions, in many fields spanning from home automation to industrial process monitoring [1]-[3]. With the emphasis on achieving a long operational lifetime, these wireless systems are typically designed with low-power modules, that, in-turn, provide limited computational capability and communication bandwidth and are therefore, generally better suited to low-sample rate intermittent monitoring applications.

Realizing high-sample rate monitoring applications, such as those based on video, audio, image, and vibration data, the communication bandwidth of such a wireless system poses a challenge in transmitting large amounts of raw data, acquired in these applications. To overcome this problem, one possibility is to process raw data locally in the wireless system so as to generate small a amount of results that can be transmitted wirelessly. However, the processing speed, of low-power micro-controllers that are typically integrated in these systems, becomes a limiting factor in achieving the required performance, in order to process a large amount of raw data through a series of computationally intensive algorithms. Nevertheless, in some recent studies, carried out to investigate feasible architectures for high-sample rate applications, such as image and vibration based condition monitoring, the desired processing performance is achieved by means of a hardware accelerator, such as a Field Programmable Gate Array (FPGA) [4]-[5]. In addition, it is also shown that an operational lifetime time of up to several years is achievable by dynamically optimizing the power consumption and by operating the wireless monitoring system in a duty-cycle manner.

However, the above mentioned design exploration was performed for a fixed sample rate. For example, in the case of vibration based condition monitoring, a sampling frequency of 50 kHz was used to acquire and process tri-axes (i.e. three channels') vibration data. Given the design complexity, increased power consumption, and additional cost of integrating an FPGA into a wireless monitoring system, it is highly desirable to investigate the sampling rates for which an FPGA is an ideal processing solution in order to realize high-sample rate applications. To the best of the authors' knowledge, no such study can be found in the published literature and therefore, an investigative study is carried out and presented in this paper.

Among high-sample rate applications mentioned earlier, vibration based process condition monitoring is equally important for low, medium as well as high sampling rates, as the desired frequencies to be monitored vary among different machinery and their operating conditions. In addition, the vibration based condition monitoring is a widely accepted method to determine growing defects in rotating machinery [6]-[7]. Therefore, for the investigative study presented in this paper, a vibration based condition monitoring application, in which high-resolution tri-axes vibration data is processed, is realized so as to evaluate an FPGA based architecture in relation to different sampling rates. The FPGA based architecture is then compared with two micro-controller based architectures. In addition, different communication loads, the amount of results that are transmitted wirelessly after processing vibration data, are also used to observe the effect of wireless transmission in relation to different sampling rates. Based on the measured processing performance and energy consumption, obtained from real hardware implementation, the architectures are compared in relation to different sampling rates and communication loads.

The remainder of the paper is organized as follows. In section II, the details regarding different architectures investigated in this paper are provided. In section III, the vibration data processing algorithm implemented in the study is described. In section IV, the experimental setup is described. In section V, the results and the supporting discussion are presented. In section VI, the concluding remarks are given.

II. Architectures and their Evaluation Criteria

In order to evaluate the performance and energy consumption of an FPGA based wireless monitoring system, in addition to the FPGA based architecture, two micro-controller based architectures specified in the following are realized for comparison.

A. Architectures

Architecture I The majority of the low-power wireless systems used in monitoring applications are based on simple micro-controllers that typically operate at a clock frequency of a few MHz to 20 MHz. Therefore, it is interesting to investigate an FPGA based architecture in comparison with a micro-controller based architecture that operates within the above mentioned clock frequencies and, unlike the FPGA, processes data in a sequential manner. In this architecture, the data acquisition, processing and result transmission is performed with a micro-controller, operating at a clock frequency of 16 MHz.

Architecture II With a sequential processor, such as a micro-controller that supports higher operating frequencies such as 50 MHz or above, higher processing performance can be achieved at the cost of increased power consumption. In relation to evaluating architectures on the basis of performance and energy consumption for a given sampling frequency, it can be valuable to include such an architecture for comparison. Therefore, in this architecture, the data acquisition, data processing, and result transmission is performed with a micro-controller operating at a clock frequency of 60 MHz.

Architecture III In this architecture, an FPGA operating at a clock frequency of 100 MHz is used as a hardware accelerator to perform tri-axes vibration data processing. The data acquisition and result transmission is performed through a micro-controller operating at 16 MHz.

B. Scheduling

As both the processing performance and the energy consumption, for each of the above mentioned architectures, is highly dependent upon the way the processes are scheduled, and therefore, it is important to describe the corresponding scheduling.

Due to similar resources and the sequential nature of the processing in architectures I and II, their scheduling is identical as shown in Fig. 1(a). In order to off-load the processor from the data acquisition process, direct memory access (DMA) is used to acquire data. This allows the acquisition of a new data set while processing and transmitting results for a previously acquired data set . However, data processing and result transmission for a given data set is achieved in a sequential manner for each of the three axes.

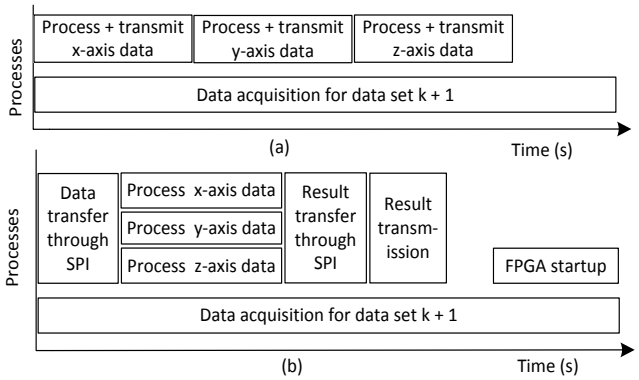

Fig. 1. Process scheduling for (a) architecture I and II (b) architecture III

The scheduling for architecture III is shown in Fig. 1(b). In a similar manner to that for architectures I and II, the data acquisition and the remainder of the processing is performed in parallel. In addition, actual vibration data processing for all three axes is also performed in parallel. The raw vibration data and results are communicated between the FPGA and the micro-controller through a serial peripheral interface (SPI). In this architecture, the FPGA is powered-on immediately before the data transfer and remains powered-on until the results are transferred to the micro-controller. After that, it is powered-off to conserve energy.

In all three architectures, during the data acquisition and processing, the micro-controller and the radio transceiver are switched to low-power modes if the wakeup time allows them to satisfy the performance requirements for a given sampling frequency. It should also be noted that the relative execution time regarding each process shown in the Fig. 1 is arbitrary, as the actual execution time depends upon many factors such as sampling frequency, number of samples to be processed, architecture etc.

C. Sampling rates and evaulation criteria

The aforementioned architectures are realized and evaluated in relation to a wide range of sampling rates starting from 0.5 kHz to 200 kHz. This wide range of sampling rates is chosen not only to ensure Nyquist sampling rates for typical low, medium, and high-frequency responses such as from few Hz to 50 kHz are covered in this study, but challenges and bottleneck associated with oversampling are also included in the study. The exact sampling rates and the corresponding number of samples processed as one data set are given in Table I. For sampling rates of up to 8 kHz, the data set size is selected to match with the maximum expected frequency component ( ) so as to achieve a resolution of 1 Hz/bin in the frequency spectrum. However, for sampling frequencies above 8 kHz, the number of samples in a data set is fixed to 4096, as further increasing the data set size poses a challenge in realizing a practical solution, in relation to typical wireless monitoring systems with limited processing and memory resources.

For a given sampling frequency, all three architectures are quantitatively evaluated in relation to the following two parameters.

1. **Real time processing performance** In relation to a vibration based condition monitoring system, it is important to analyze several consecutive data sets in order to be able to detect anomalies and to draw conclusion regarding the condition of machinery. Therefore, the architectures are evaluated for their ability to process high-resolution tri-axes vibration and transmit results in a real time manner.
2. **Energy consumption** For a battery operated wireless vibration monitoring system, deployed on a rotating and/or hard to access part of machine, minimum energy consumption is desirable, as it enables the possibility of achieving a long operational lifetime using limited amounts of energy resources. Therefore, the architectures are also evaluated in relation to energy consumption.

III. VIBRAION DATA PROCESSING ALGORITHM REALIZED FOR THE STUDY

In this section, the vibration data processing algorithm that is used to evaluate aforementioned architectures in relation to different sampling rates is described.

A. Data Acquisition

The 16-bit vibration data for each of the three axes is acquired at a sampling frequency that ensures the Nyquist's criteria for digitizing a analog signal from a vibration sensor with a given bandwidth. The acquired samples are then stored in three buffers, one for each axis, in the micro-controller. A new data set, comprising consecutive samples for each axis is then processed as soon as the processing for the previously acquired data set is completed.

B. Filtering

The raw vibration data is firstly filtered in order to remove noise and to select the desired bandwidth of the signal before further processing. In the experimental study performed for this paper, an equiripple direct-form Finite Impulse Response (FIR) filter with a filter order of 135 is realized. The given filter order is chosen as it enables a balance in filtering quality and computational complexity to be achieved, in relation to limited processing and storage resources available in a typical wireless monitoring system.

C. Windowing

In order to minimize the spectral leakage resulting from computing a Fast Fourier Transform (FFT) on a fixed length data set, a windowing operation is performed on the filtered data. The hamming window function is used in the experimental design, as it provides a high performance with relatively low computational complexity as compared to other windowing functions.

D. FFT

After performing the windowing operation, the time domain signal is transformed into frequency domain information by means of the FFT algorithm. As the acquired vibration data is comprised of real values, the imaginary input of the FFT is set to zero.

E. Spectrum Processing

In the analysis phase, the following three methods, which generate very low to very high processed information for wireless transmission, are used to investigate the effect of different payloads on the given sampling frequency.

TABLE I. SAMPLING FREQUENCIES AND CORESSPONDING DATA SET SIZES USED TO EVALUATE ARCHITECTURES

Sampling rate (kHz)	0.5	1	2	4	8/16//32/50/ 100/150/200
No. of samples in a data set	256	512	1024	2048	4096

No Load (NL) In NL, the amplitude of each frequency components is not only analyzed in relation to that of the stored spectrum but the amplitudes of the frequency components, representing the fundamental frequency and its harmonics are analyzed to assess the condition of machinery. If there is no detected anomaly to report, as is highly likely for machinery operating under normal conditions, no information is transmitted wirelessly.

One Byte Per Analysis (1BPA) In a similar manner to that for NL, in 1BPA the amplitude of each the frequency components is not only analyzed in relation to that of the stored spectrum but the amplitudes of the frequency components that represents the fundamental frequency and its harmonics are analyzed to assess the condition of the machinery. However, based on the analysis, a resultant code of one byte for each axis is transmitted for notification purposes.

Full Spectrum Per Analysis (FSPA) In FSPA, the high resolution frequency spectrum is transmitted to a central station where it is analyzed by experts. The main motivation to include this scheme was to present the worst case performance and power consumption that results in a large amount of wireless communication.

F. Result Transmission

In the case of spectrum processing of 1BPA and FSPA, the micro-controller transmits the results using an IEEE 802.15.4 compliant radio transceiver.

IV. EXPERIMENTAL SETUP

A. Hardware platform used to realize the architectures

In this study, a wireless embedded platform, the SENTIOF [8] is used to realize the architectures described in Section II. The SENTIOF integrates a low-power micro-controller to perform sequential processing tasks in addition to performing control specific operations such as power management. It also integrates a Spartan-6 FPGA to perform computationally intensive processing tasks. In order to store both the short term and long term data, it integrates a low-power 4 MB of SRAM and a 64 Mb Flash. In addition, the CC2520 RF transceiver that operates at the 2.4 GHz IEEE 802.15.4 band and supports a data throughput of 250 kpbs, is integrated to provide wireless connectivity.

Thanks to the flexible design of the SENTIOF that enables dynamically switching-off power to the FPGA and memories, architectures I and II are realized. In order to realize

architecture III, the power to the FPGA is turned-on by the micro-controller, which performs tasks such as data acquisition, result transmission and power management in this architecture.

The SENTIOF provides a generic interface to connect sensors for a given application. In relation to vibration monitoring, a custom designed sensor board is used to validate the architectures. However, the power consumption corresponding to vibration sensors is excluded in order to achieve a fair comparison, as power consumption for applicable sensors varies among sensors and with different sampling frequencies.

B. Performance and power consumption measurement

In order to obtain energy consumption of the architectures, both the execution time and power consumption were measured on real hardware by executing the vibration processing algorithm discussed in section III. During this execution time, the micro-controller, radio transceiver and the FPGA are switched to their respective low-power modes depending upon the idle time and the time required to switch these modules to back to active mode. The details regarding different power modes and the corresponding power consumptions can be found in [8]-[9].

The instantaneous current consumption for each architecture was recorded at a sampling frequency of 10 kHz using a digital multi-meter, Agilent 34410A. In order to obtain the average power consumption, the recorded instantaneous current consumption was averaged and multiplied by a supply voltage of 3.6 V, which is used to power the SENTIOF.

V. RESULTS/DISCUSSIONS

The results, related to the performance and energy consumption, measured by operating each architecture at different sampling rates, are organized according to the spectrum processing methods discussed in section III.E, as this allows for the evaluation of the architectures using equal processing and communication load.

A. NL

In relation to the spectrum processing method NL, the execution time required to process one set of vibration data on three architectures is shown in Fig. 2. As the number of samples in a data set increases from 256 to 4096, so does the execution time for each of the architectures. For both the architectures I and II, the percentage increase in the execution time is measured to be 150 % for sampling rates of 0.5 kHz to 1 kHz, and gradually decreases to around 100 % for sampling rates of 4 kHz to 8 kHz. For architecture III, the execution time increases at a constant rate of around 100 %.

Analyzing the execution time in relation to the time required to sample one set of data (as shown in Fig. 2), it can be observed that the real-time processing performance for architectures I, II and III is achieved for sampling rates of up to 4 kHz, 16 kHz, and 150 kHz, respectively.

The energy consumed in processing one set of vibration data for a spectrum processing method of NL is shown in Fig. 3. It should be noted that the reported energy consumption includes

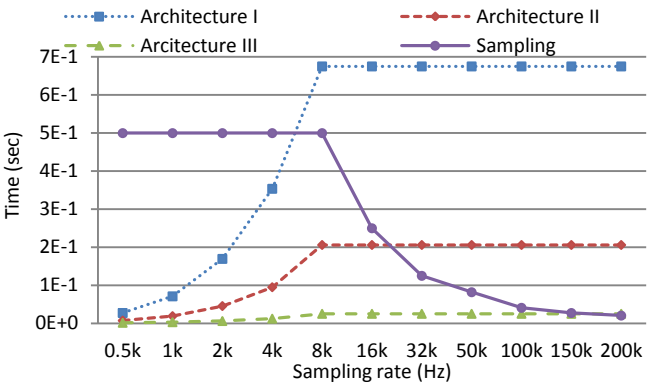

Fig. 2. Time required to process one data set using spectrum processing method NL for different sampling rates

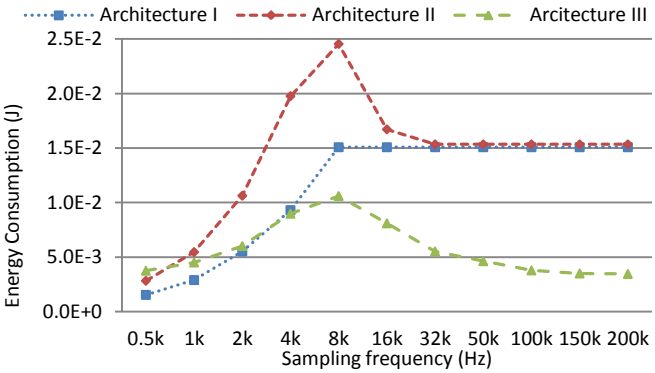

Fig. 3. Energy consumed to process one data set using spectrum processing method NL for different sampling rates

the energy consumed to acquire a full data set, in addition to the energy consumed in processing. In Fig. 2, we observed that the energy consumption, for each architecture increases, as the number of samples to process are increased from 256 to 4096. The rate of this increase, is almost identical for both architectures I and II, however, the energy consumption of architecture II is always greater than that of architecture I, for a given sampling rate. For sampling rates of up to 8 kHz, this is because the higher processing speed of architecture II is least useful in relation to its power consumption, when performing simple tasks such as sampling the data. On the other hand, for sampling frequencies above 8 kHz, the gain in processing performance of architecture II, is measured to be less than the corresponding increase in the power consumption, as compared to those of architecture I. Therefore, the energy consumption of architecture I for sampling rates above 8 kHz is less than that of architecture II.

The architecture III not only achieves real time processing performance for sampling rates of up to 150 kHz, but also consumes the least amount of energy for sampling rates above 4 kHz. For a sampling rate of 4 kHz, though, the energy consumed by architecture III is slightly (3.5 %) less than that of architecture I, given the simple design and low-cost of architecture I, it may be preferred over architecture III for a sampling rate of 4 kHz.

The results regarding the real time processing performance and energy consumption for spectrum processing NL are summarized in Table II. For sampling rates of up to 4 kHz,

architecture I stands ahead of the other architectures. On the other hand, for sampling rates of 8 kHz and above, architecture III achieves the highest performance with the least amount of energy consumption.

TABLE II. SUMMARY OF THE RESULTS FOR SPECTRUM PROCESSING NL

Sampling Frequency (kHz)	**0.5**		**1**		**2**		**4**		**8**		**16**		**32**		**50**		**100**		**150**		**200**	
Real Time Processing Performance (RPP) / Least Energy Consuming (LEC)	RPP	LEC	RPP	LEC	RPP	LEC	RPP	LEC	RPP	LEC	RPP	LEC	RPP	LEC	RPP	LEC	RPP	LEC	RPP	LEC	RPP	LEC
Architecture I	x	x	x	x	x	x	x															
Architecture II	x		x		x		x		x		x											
Architecture III	x		x		x			x	x	x	x	x	x	x	x	x	x	x	x	x		x
Most appropriate architecture	**I**		**I**		**I**		**I/III**		**III**		**III**		**III**		**III**		**III**		**III**		**III**	

B. 1BPA

The time required to process vibration data for the spectrum processing method 1BPA is shown in Fig. 4. In this method, after processing one data set, three bytes of results, one for each axis, are transmitted wirelessly, which increases the execution time shown in Fig. 2 by a constant amount of 1.3 ms. With this minor addition, the overall performance remains the same as that achieved using the spectrum processing NL.

The energy consumption of the architectures for spectrum processing 1BPA is shown in Fig. 5. With three bytes of wireless transmission, the maximum increase in the energy consumption of any architecture is less than 170 µJ, when compared with that of the spectrum processing NL. In other words, this small communication energy is less than 0.1 % of the total energy consumed for a given sampling rate. Based on the performance and energy consumption, the quantitative comparison of the three architectures, when 1BPA is used, is the same as that given in Table II. Therefore, it can be concluded that architecture I is the most appropriate solution for sampling rates of up to 4 kHz, when a light amount of results such as three bytes are wirelessly transmitted after processing each data set. On the other hand, for higher sampling rates, architecture III maintains its domination.

C. FSPA

The time required to process and transmit a full spectrum using the spectrum processing method of FSPA for different sampling rates is shown in Fig. 6. The execution time, in general and, especially for architectures II and III, is dominated by the time required to wirelessly transmit the full spectrum. For example, the minimum ratio of communication time to total execution time for architectures II and III are 70 % and 90 %, respectively. In such a case, the large transmission load becomes a limiting factor to achieve real time processing performance, even for architecture III.

The energy consumption of the three architectures when the spectrum processing method FSPA was used is shown in Fig. 7. For a given sampling rate, communication energy, which is the same for each architecture, dominates the total energy consumption. Therefore, total energy consumption differs by only a small amount among the architectures. For example, the

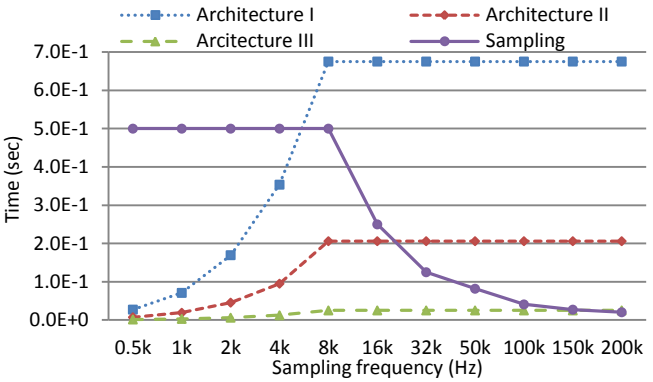

Fig. 4. Time required to process one data set using spectrum processing method 1BPA for different sampling rates

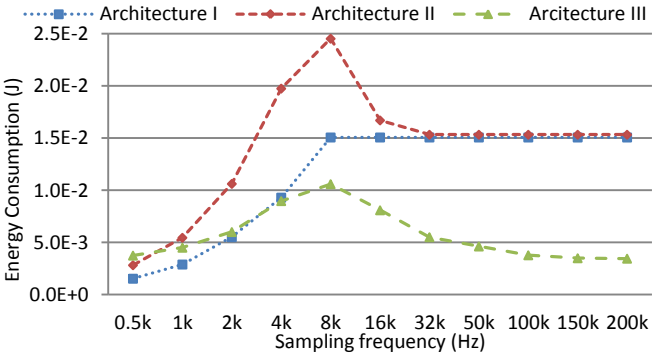

Fig. 5. Energy consumed to process one data set for using spectrum processing method 1BPA for different sampling rates

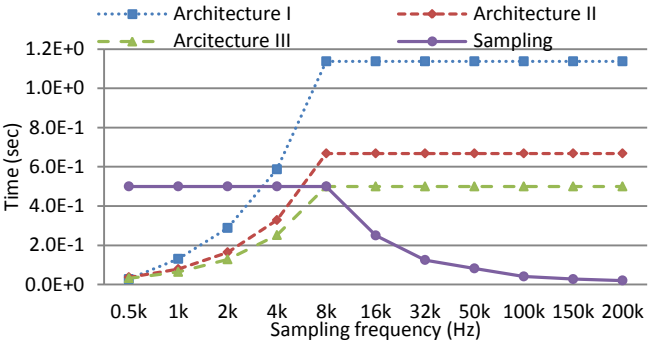

Fig. 6. Time required to process one data set using spectrum processing method FSPA for different sampling rates

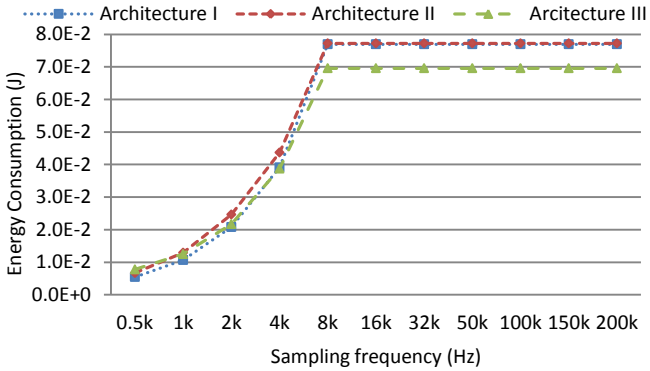

Fig. 7. Energy consumed to process one data set for using spectrum processing method FSPA for different sampling rates

TABLE III. SUMMARY OF THE RESULTS FOR SPECTRUM PROCESSING FSPA

Sampling Frequency (kHz)	0.5		1		2		4		8		16		32		50		100		150		200	
Real Time Performance (RTP) / Least Energy Consuming (LEC)	RTP	LEC	RTP	LEC	RTP	LEC	RTP	LEC	RTP	LEC	RTP	LEC	RTP	LEC	RTP	LEC	RTP	LEC	RTP	LEC	RTP	LEC
Architecture I	x	x	x	x	x	x																
Architecture II	x		x		x		x															
Architecture III	x		x		x		x	x	x	x		x		x		x		x		x		x
Most appropriate architecture	I		I		I		III		III		III		III		III		III		III		III	

energy consumption of architecture I is within 10 % of that of architecture III for sampling rates of 4 kHz and above.

Based on the processing performance and energy consumption, the results, when the spectrum processing method FSPA is used, are summarized in Table III. For sampling rates of 2 kHz and below, architecture I not only achieves real time processing performance but also results in the least energy consumption. On the other hand, for sampling rates above 2 kHz, the architecture III delivers the highest performance with the least energy consumption. It should be noted that none of architecture achieve real time processing performance for sampling rates above 8 kHz. Therefore, to achieve real time performance for high-sampling rates, small amount of results should be transmitted wirelessly.

D. Challenges

In this section, challenges and bottlenecks associated with modules that are used to realize these architecture are analyzed in relation to achieving real-time processing performance.

In case of architecture I and II, the execution time, , for a given sampling rate as represented by eq. 1, is the maximum of acquisition time, , or the sum of processing, , and communication time, .

(1)

For sampling rates of 4 kHz and below, the execution time is determined by the acquisition time. There is, however, an exception to architecture I when the spectrum processing method, FSPA, is used. In this case, the wireless communication creates a bottleneck in relation to achieving real time processing performance. On the other hand, for sampling rates above 8 kHz, irrespective of the time required to wirelessly transmit results, the processing time becomes a bottleneck in relation to achieving high-performance using architecture I. For architecture II, the same is true when sampling rates above 16 kHz are used.

(2)

The execution time for architecture III can be determined as shown in eq. 2. In relation to eq. 1, two additional terms, time required to transfer data from micro-controller to the FPGA, , and time required to transfer results from the FPGA to the micro-controller, are added to eq. 2. For sampling rates of 8 kHz and below, none of inter-communication (between the micro-controller and the FPGA), processing or wireless transmission time poses any challenge in achieving real-time performance. Irrespective of the wireless transmission time, becomes a bottleneck for sampling rates above 150 kHz. On the other hand, wireless transmission poses a challenge in achieving real-time performance when spectrum processing FSPGA is used.

TABLE IV. SUMMARY OF DIFFERENT CHALLENGES AND BOTTLENECKS FOR EACH ARCHITECTURE

		Architecture I	Architecture II	Architecture III
Data acquisition				
Data transfer from micro-controller to FPGA		N. A	N. A	> 150 kHz
Data processing		> 4 kHz	> 16 kHz	
Result transfer from FPGA to micro-controller		N. A	N. A	
Wireless transmission	**1BPA**			
	FSPA	> 2 kHz	> 4 kHz	> 8 kHz

Based on the time required to perform data acquisition, inter-module communication, processing and wireless transmission discussed above, the bottlenecks in achieving real-time performance for each of the architecture are summarized in Table IV.

VI. CONCLUSION

In this paper, an FPGA based architecture is evaluated in relation to different sampling rates for a computationally intensive application. Based on the measured real time processing performance and energy consumption, the FPGA based architecture is compared with two micro-controller based architectures so as to determine the sampling rates for which it achieves the highest performance with the least energy consumption.

The architecture I, in which a micro-controller is operated at a low speed, 16 MHz, to perform computationally intensive high-resolution tri-axes vibration data processing tasks, not only achieves real time processing performance but also consumes the least amount of energy for sampling rates of up to 2 kHz. For sampling rates of 4 kHz and above, architecture III, in which an FPGA is used to process vibration data, achieves the performance goals with the minimum energy consumption. On the other hand, architecture II, based on a high-speed micro-controller, is found to be the least appropriate solution for any sample rate in comparison to the other two architectures.

In relation to communication load, when it is increased from a few bytes to full spectrum, the real time performance goals, even for architecture III, are limited to sampling rates of up to 8 kHz. In addition, the associated energy consumption is increased by many orders of magnitude. Therefore, wireless transmission should be minimized in order to achieve the required performance for high-sampling rate monitoring while maintaining low-energy consumption.

REFERENCES

[1] T. Arampatzis, J. Lygeros, and S. Manesis,"A Survey of Applications of Wireless Sensors and Wireless Sensor Networks," *IEEE International*

Symposium on Intelligent Control, *Mediterrean Conference on Control and Automation*, pp.719,724, 27-29 June 2005.

[2] C. Gomez, and J. Paradells, "Wireless home automation networks: A survey of architectures and technologies," IEEE *Magazine on Communications*, vol.48, no.6, pp.92,101, June 2010.

[3] X. Xin, V. Sundararajan and W. P. Brithinee, "The application of wireless sensor networks for condition monitoring in three-phase induction motors", Electrical Insulation Conference and Electrical Manufacturing Expo, 2007, pp. 445-448, October 2007.

[4] K. Khursheed, M. Imran, A.W. Malik, M. O'Nils, N. Lawal, and B. rnberg, "Exploration of Tasks Partitioning between Hardware Software and Locality for a Wireless Camera Based Vision Sensor Node", 26th International Symposium on Parallel Computing in Electrical Engineering (PARELEC), pp.127-132, 3-7 April 2011.

[5] K. Shahzad, P. Cheng,B. and Oelmann, "Architecture exploration for a high-performance and low-power wireless vibration analyzer", IEEE Journal on Sensors, vol.13, no.2, pp.670-682, Feb. 2013.

[6] E. P. Carden, P. Fanning, "Vibration Based Condition Monitoring A Review" Journal on Structural Health Monitoring, vol. 3, issue 4, pp. 355-377, Dec. 2004.

[7] S. Edwards, A.W. Lees and M.I. Friswell, "Fault diagnosis of rotating machinery", Journal of Shock and Vibration Digest, vol. 30, issue 6, pp. 4–13, 1998.

[8] K. Shahzad, P. Cheng, and B. Oelmann, "SENTIOF: An FPGA-Based High-Performance and Low-Power Wireless Embedded Platform" Accepted for publication *in International Conference on Wireless Sensor Networks (WSN'13)*, 8-11 Sep. 2013.

[9] K. Shahzad and B. Oelmann, "Investigation of Energy Consumption of an SRAM-based FPGA for Duty-Cycle Applications", Accepted as conference publication in *ParaFPGA2013, Parallel Computing with FPGAs,* Munich Germany, 10-13 Sep. 2013.