Embedded Self Organizing Systems (Vol 10. No 7. 2023) (pp.39-42)

Embedded Selforganizing Systems

Special Issue Topic: “Proceedings of ISCSET 2023“

Hardware Software Co-Design for Impact Localization using Hybrid Laminates

Stephan Lede
Chemnitz University of Technology
Department of Computer Engineering
Chemnitz, Germany
stephan.lede@informatik.tu-chemnitz.de

René Schmidt
Chemnitz University of Technology
Department of Computer Engineering
Chemnitz, Germany
rene.schmidt@informatik.tu-chemnitz.de

André Böhle
Chemnitz University of Technology
Department of Computer Engineering
Chemnitz, Germany
andre.boehle@informatik.tu-chemnitz.de

Wolfram Hardt
Chemnitz University of Technology
Department of Computer Engineering
Chemnitz, Germany
wolfram.hardt@informatik.tu-chemnitz.de

***Abstract*—Impact detection using piezoelectric sensors is an actual and widespread research field. The current work provides an approach for a real-time realization of an impact detection system using deep learning methods. For realization a hardware software co-design approach is used utilizing hardware acceleration by a continuous pipelining FPGA structure. The concept describes the hardware software partitioning of the underlying functions and the methodology for ensuring continuous data processing and the associated real-time capability. The behavior of the hardware is realized with the help of a finite state machine and thus the correctness of the data is ensured and the impact identification is realized. The results show the real-time capability as well as a reasonable resource utilization of the FPGA design.**

***Index Terms*—Signal processing, Machine learning, Signal processing systems, Real time**

I. Introduction

In the aerospace sector, components made of composite materials are used to save weight without compromising the stability of flying objects, and thus to comply with the increasingly stringent emission guidelines by reducing fuel consumption [1]. The problem with composites is that, in contrast to metallic components, impact damage is hardly visible from the outside [2]. Inside the hybrid laminates, these impacts can nevertheless lead to delamination of interlayers, fiber breaks and matrix cracks, which must be detected early on, especially in safety-critical applications [3], [4]. For this reason, piezoelectric sensors have been used for many years to detect and localize impacts on composite materials [5].

Another area of application for impact localization on hybrid laminates is as a multifunctional lightweight component in the automotive industry. As part of the MERGE federal cluster of excellence, research is being conducted on an intelligent center console that will detect inputs similar to a touch display and perform the corresponding functions, such as opening and closing windows or controlling the multimedia system [6], [7].

In addition to localize the impacts as precisely as possible, resource consumption and the necessary computation time are also relevant. Particularly when used as an input facility for, among other things, safety-critical functions, reliable and real-time capable localization of the inputs must be provided.

II. Related Work

To localize impacts on composite materials, deep learning approaches, among others, have been evaluated in various research projects. In 2000, Worden and Staszewski equipped a $530\,\mathrm{mm}$ wide and $400\,\mathrm{mm}$ long composite panel with 17 piezoceramic sensors [8]. As a result, this correctly predicted the impacts with an average difference of $23.1\,\mathrm{mm}$ in the x-direction and $25.7\,\mathrm{mm}$ in the y-direction [8]. Tabian et al. were also able to achieve accuracies of over $95\,\%$, and in most cases over $99.4\,\%$, on a composite $1\,150\,\mathrm{mm}$ long and $750\,\mathrm{mm}$ wide equipped with 12 piezoceramic lead zirconate titanate (PZT) sensors, using a CNN-based approach [1]. Damm et al. confirmed these results in 2020 on a $280\,\mathrm{mm}$ long square composite plate equipped with three PZT sensors [9]. They were able to determine the exact position with an accuracy of $99.55\,\%$ and an average deviation of $0.16\,\mathrm{mm}$ [9].

Based on the MERGE center console equipped with four piezoelectric sensors, various options for localizing the inputs have already been evaluated. Using a support vector machine (SVM) with a polynomial kernel, Schmidt et al. achieved an accuracy of 84 % with an energy consumption of approximately 1.7 W [7]. The time differences between the individual sensors, determined by thresholding, have been used as input data, resulting in an intensity dependency [7]. This dependency is impractical for use as a center console, so Böhle et al. considered signal features that guarantee intensity independence [10]. Overall, the results obtained are not satisfactory, since a maximum accuracy of about 70 % does not meet the requirements of a safety-critical system [10].

An improvement has been obtained by Lede et al. using feed forward neural networks (FFNN) and convolutional neural networks (CNN) [11]. A Digilent-branded Zynq XC7Z010 board with a 650 MHz ARM dual-core processor and FPGA has been used for data processing and prediction calculation [11]. The neural networks are trained by using the amplitude spectrum of the sensor data recorded by several subjects [11]. The average accuracy during testing of the learned FFNN is 99.26 %, while the CNN achieved 98.53 % [11]. In addition to the accuracy, the time required and the resource utilization have been also evaluated, whereby the FFNN, which required only 12ms for the calculation of the prediction, has been faster than the CNN with 38ms [11]. The maximum physical memory consumption is less than 17 MB and the maximum virtual memory consumption less than 29 MB for both networks.

As a result, neural networks were found to meet all requirements for impact localization on an automotive center console. So far, except for the analog-digital conversion of the sensor signals, all functionalities are performed on the processor. However, the current analyses include the processing time for classification and the necessary Fast Fourier Transformation (FFT) transformation only. The additional necessary identification of relevant measurement windows as input for the classification are not included. Therefore, the identification of a potential impact should be executed directly by the hardware and just relevant sensor data should be streamed to the software. This keeps resources free to enable further applications such as the Graphical User Interface or necessary interfaces.

III. Concept

To realize this, a hardware-software co-design is required, which can be implemented on the hardware mentioned in section II. For this purpose, the hardware has to convert the analog signals of the four piezoelectric sensors into digital signals and monitor if certain limit values are exceeded. In addition, it is necessary to ensure that the impact is fully contained in the data stream. The fundamental idea is to continuously fill a memory, which works according to the first in - first out principle (FIFO), with sensor data and forward the complete content to the software when an impact is detected.

The overall system concept is shown in Figure 1. The first step is to connect the piezoelectric sensors to the analog-to-digital converter (ADC) interface through an electrical circuit (cf. [7]). The hardware-software partitioning is done by distributing the functions to the FPGA as hardware and the ARM processor as software resource. The first step of the FPGA implementation is to provide the ADC interface converting the analog signals into digital values ensuring the availability for subsequent components. The impact detection is to be done via a threshold comparison, which is triggered as soon as one of the sensors exceeds the given limit. As soon as the FIFO is filled, the data is streamed to the software via the AXI4-Stream interface. Subsequently, the processor performs a FFT and starts classification using the selected neural network. The determined input can be sent to the interface via a desired medium, such as CAN, Ethernet or UART.

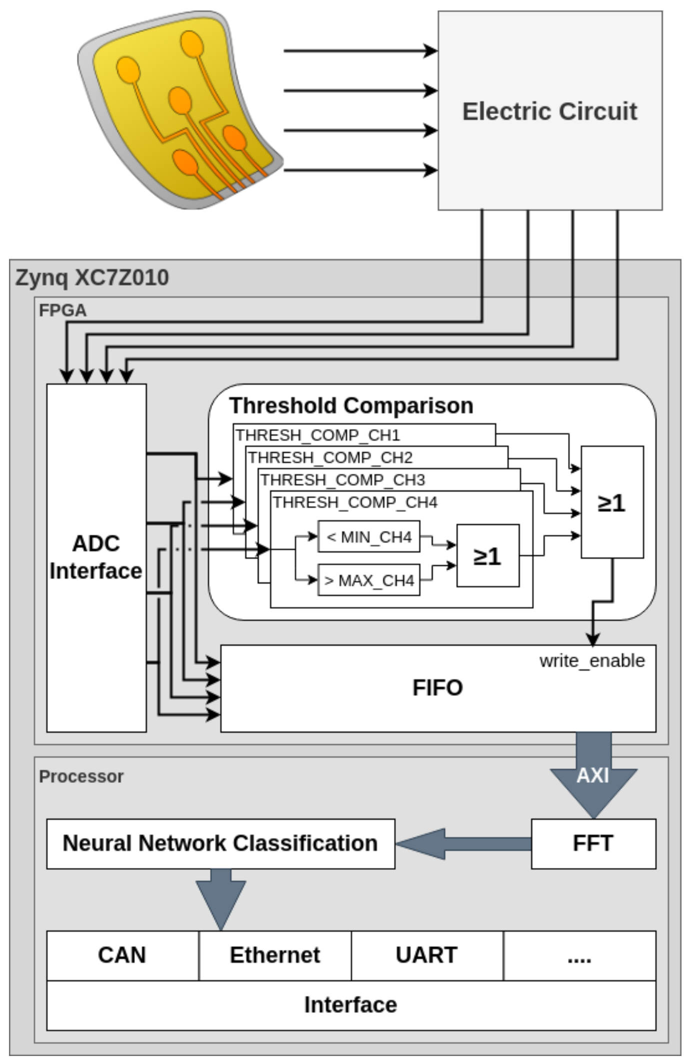

Fig. 1. Concept of system architecture for real-time processing based on Zynq SoC (XC7Z010).

Two options exist for classification, using either a FFNN or a CNN. In terms of accuracy, both networks are very similar, with the FFNN (99.26 %) performing slightly better than the CNN (98.53 %) when tested with previously unknown data [11]. On the other hand, for the loss function values, the CNN performs better, especially for training and validation [11]. For the functions performed by the software, the execution times are already known. The calculation of the amplitude spectrum (magnitude of the FFT) requires $10\,\mathrm{ms}$ and the classification using the CNN $38\,\mathrm{ms}$, respectively $12\,\mathrm{ms}$ for the FFNN [11]. This results in two options, either with slightly better accuracy using the CNNs or with faster computation time using the FFNNs. The time remaining to get from an impact to a result is determined by the time required to perform the FPGA's operations. Consequently, two valid options are available for the realization in the software part, which can be selected depending on the available resources and the required accuracy of the given application ensuring reusability of the Hardware implementation. By using a deterministic system, as well as the predictability of the hardware acceleration, a fixed deadline can be ensured, making the system real-time capable.

IV. Implementation

A detailed explanation of the implementation of the software part is omitted, as this has been described in preliminary work [11]. The FPGA design implements a continuous data flow, which consists of ADC interface, Threshold Comparison, and data storage in the FIFO. The data flow is regulated by an finite state machine (FSM) ensuring the necessary functionality and FIFO storage behavior. After the relevant impact data is stored, the software is informed to read the FIFO and perform the FFT and classification, accessing the data via an AXI DMA. In the following, the structure and function of the ADC interface just as the data processing are explained in more detail, whereby a detailed description of the FIFO and the DMA is omitted, since these are standard XILINX IP cores.

A. ADC Interface

In addition to the standard reset and clock inputs, the ADC interface has one positive and one negative input for each of the four sensors, whereby $500\,000$ $12\,\mathrm{bit}$ samples per second can be recorded. As output a $64\,\mathrm{bit}$ vector is used, which contains the data samples of one point in time of all four recorded ADC channels representing four sensor signal samples. Additionally, a validation signal is provided indicating whether the data in the vector is valid and can be processed further. The control of the interface is taken over by a FSM, whose states are explained in more detail below.

After initialization, the interface is in the idle state and waits until an analog-to-digital conversion is completed. It then determines which of the sensor values are currently available and stores them in the respective vector. In parallel, as soon as a sensor signal has been converted, its value is written to the corresponding position of the output vector. After all four sensor signals have been proceed and stored in the output vector, a confirmation is provided by setting the corresponding validation signal. Further data processing in the FIFO control unit takes place and the ADC interface can sample new data by returning to the idle state ensuring real time behavior and continuous data processing.

B. Data Processing for Impact Detection

The data processing necessary for impact detection is done by the FIFO control unit. This is implemented by a FSM controlling the overall data flow and it contains a FIFO with a write depth of $16\,384$ and a width of $64\,\mathrm{bit}$. In addition to the reset and clock inputs, it has inputs to determine whether the AXI DMA is ready to receive data, a data input from the ADC interface and the corresponding valid signal. The outputs of the control unit are the data from the FIFO and the corresponding valid signal.

As can be seen in Figure 2, the FSM starts in the INIT state, initializing necessary control signals. This is followed by the PREFILL state, in which the FIFO is filled with $1\,638$ values. This is necessary because the used classification in the software part has already received sensor data with this buffer during the training process. This ensures that the entire impact is evaluated and no important data for the classification is lost. Simultaneously, it is ensured that the comparability of the life system and the training data is given. As soon as the FIFO contains this amount of data, the FSM changes to the IDLE state in which the threshold comparison is performed. The minimum and maximum threshold values of the sensors are stored in registers, which allows adjustment by the software. As long as the values are in the idle range, new data is permanently stored in the FIFO and the corresponding number of oldest data is discarded. This ensures a continuous flow of data, which guarantees the actuality and correctness of the data. As soon as the respective threshold value has been undercut or exceeded, the FSM changes to the FILLING state. The new data, which now contains the impact, is written to the FIFO until it is full, whereupon the FIFO switches to the FULL state. Now it is checked whether the AXI DMA is ready to receive data and whether valid data is present at the data output of the FIFO. If this is the case, the data is read out and streamed to the software via the AXI DMA until the FIFO is empty, whereupon the FSM is switched back to the PREFILL state.

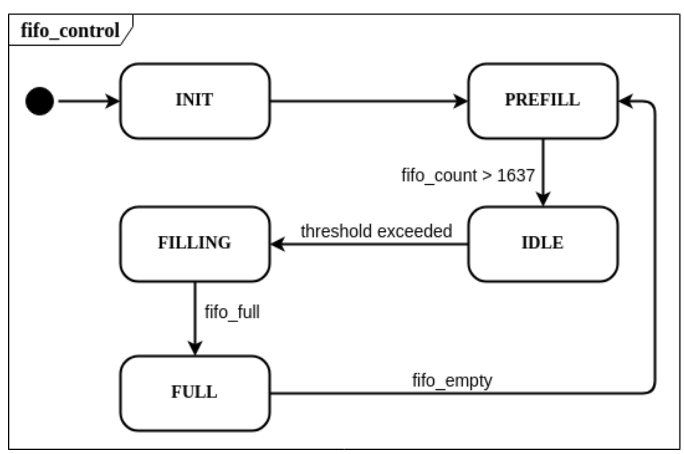

Fig. 2. Finite state machine of the FIFO control unit

V. Results

By synthesizing, implementing, and simulating this design, the following results can be achieved. In the Table I, the resource usage of the embedded system can be seen which shows that $18.91\,\%$ of the available Look Up Tables (LUT), $11.95\,\%$ of the Flip Flops (FF) and $55\,\%$ of the Block RAMs (BRAM) are used.

TABLE I
Resource utilization of the target system per implemented component

Resource	Component	Utilization	Available
LUT	Total	3329	17600
	AXI Interconnect	1954	
	AXI DMA	1062	
	FIFO Control	182	
	ADC	54	
	Miscellaneous	77	
FF	Total	4207	35200
	AXI Interconnect	2172	
	AXI DMA	1551	
	FIFO Control	83	
	ADC	121	
	Miscellaneous	280	
BRAM	Total	33	60
	AXI DMA	4	
	FIFO Control	29	

The simulation can also be used to determine the times required for the individual functions. This allows us to determine that at an FPGA clock rate of $100\,\mathrm{MHz}$, one sample of valid data from the four sensors takes $2\,080\,\mathrm{ns}$. Furthermore, the Table II shows which time the individual states of the FIFO Control FSM require, whereby all values are constant except for the IDLE state. This is due to the system waiting in the IDLE state until an impact is identified. As a result it can be derived that from the moment of the impact about $30.84\,\mathrm{ms}$ are needed until the data is streamed to the software.

TABLE II
Required time of the individual states of the FIFO control unit FSM determined by the simulation of an impact

State	Time
INIT	$250\,\mathrm{ns}$
PREFILL	$3.41912\,\mathrm{ms}$
IDLE	$2\,080\,\mathrm{ns}$
FILLING	$30.66961\,\mathrm{ms}$
FULL	$0.16385\,\mathrm{ms}$
TOTAL	$34.25491\,\mathrm{ms}$

VI. Conclusion and Future Work

In this paper a Hardware Software Co-Design has been presented realizing a real-time capable impact detection on a Zybo Zynq SoC board using a hybrid laminate equipped with piezoelectric sensors. The necessary classification has been realized with deep learning methods on the ARM processor and the necessary digital signal processing on the FPGA providing a continuous data flow controlled by a FSM. The results show a resource utilization of LUTs and FFs with approx. $12\,\%$ to $19\,\%$, which in turn can be considered as low. Contrary, the utilization of the BRAMs with $55\,\%$ is comparable high for the given functionality. This is reasoned by the observation time and corresponding measurement window size for one impact. A reduction of the measurement window size required for the classification process can be achieved by further research. The time required to sample an impact is $30.84\,\mathrm{ms}$ when the system is in the IDLE state. Adding $10\,\mathrm{ms}$ required for the FFT and $12\,\mathrm{ms}$ for classification with the FFNN gives a constant total time of about $52.84\,\mathrm{ms}$, proving real-time capability. In the future work, the necessary FFT calculation can also be shifted to the FPGA to further reduce the load on the ARM processor.

References

[1] I. Tabian, H. Fu, and Z. Sharif Khodaei, "A convolutional neural network for impact detection and characterization of complex composite structures," *Sensors*, vol. 19, no. 22, p. 4933, nov 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/22/4933

[2] A. H. Seno and M. F. Aliabadi, "Impact localisation in composite plates of different stiffness impactors under simulated environmental and operational conditions," *Sensors*, vol. 19, no. 17, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/17/3659

[3] M. Ghajari, Z. Sharif-Khodaei, and F. M. H. Aliabadi, "Impact detection using artificial neural networks," in *Advances in Fracture and Damage Mechanics X*, ser. Key Engineering Materials, vol. 488. Trans Tech Publications Ltd, 1 2012, pp. 767–770.

[4] R. Schmidt, A. Graf, R. Decker, M. Heinrich, V. Kräusel, L. Kroll, and W. Hardt, "Monitoring the structural health of glass fibre-reinforced hybrid laminates using novel piezoceramic film," *Sensors*, vol. 20, no. 18, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/18/5428

[5] W. J. Staszewski, K. Worden, R. Wardle, and G. R. Tomlinson, "Fail-safe sensor distributions for impact detection in composite materials," *Smart Materials and Structures*, vol. 9, no. 3, pp. 298–303, Jun. 2000. [Online]. Available: https://doi.org/10.1088/0964-1726/9/3/308

[6] F. Ullmann, W. Hardt, and V. Zhmud, "Machine learning algorithms for impact localization on formed piezo metal composites," in *2017 International Siberian Conference on Control and Communications (SIBCON)*, June 2017, pp. 1–5.

[7] R. Schmidt, A. Graf, R. Decker, V. Kräusel, W. Hardt, D. Landgrebe, and L. Kroll, "Hybrid laminate for haptic input device with integrated signal processing," *Applied Sciences*, vol. 8, no. 8, 2018. [Online]. Available: https://www.mdpi.com/2076-3417/8/8/1261

[8] K. Worden and W. Staszewski, "Impact location and quantification on a composite panel using neural networks and a genetic algorithm," *Strain*, vol. 36, no. 2, pp. 61–68, 2000. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1475-1305.2000.tb01175.x

[9] A. M. Damm, C. Spitzmüller, A. T. S. Raichle, A. Bühler, P. Weißgraeber, and P. Middendorf, "Deep learning for impact detection in composite plates with sparsely integrated sensors," *Smart Materials and Structures*, vol. 29, no. 12, p. 125014, nov 2020. [Online]. Available: https://doi.org/10.1088/1361-665x/abb644

[10] A. Böhle, "Evaluation of signal features used for impact localization on a center console with piezoceramic sensors," *Embedded Selforganising Systems*, vol. 6, no. 1, pp. 3–8, dec 2019.

[11] S. Lede, W. Hardt, and R. Schmidt, *Analyse des Ressourcenverbrauchs von Deep Learning Methoden zur Einschlagslokalisierung auf eingebetteten Systemen*, ser. Chemnitzer Informatik-Berichte. Technische Universität Chemnitz, Fakultät für Informatik, 2023. [Online]. Available: https://nbn-resolving.org/urn:nbn:de:bsz:ch1-qucosa2-840097