Available online at www.sciencedirect.com

SciVerse ScienceDirect

APCBEE Procedia 00 (2013) 000–000

2013 3rd Conference on Biomedical Engineering and Technology

A Framework for High Performance Embedded Signal Processing and Classification of Psychophysiological Data

Hendrik Woehrle[a,*], Johannes Teiwes[b], Elsa Kirchner[a,b] and Frank Kirchner[a,b]

[a] *Robotics Innovation Center, German Research Center for Artificial Intelligence (DFKI GmbH), Robert-Hooke-Str. 5, 28359 Bremen, Germany*

[b]*Robotics Group, University of Bremen, Robert-Hooke-Str. 5, 28359 Bremen, Germany*

Abstract

We present a framework to perform and speed up signal processing and machine learning tasks of biomedical and psychophysiological data in mobile and wearable systems using field programmable gate arrays. We show the basic architecture and capabilities of the framework and demonstrate its usage to construct a mobile system for the detection of event related potentials in electroencephalographic data. The performance of the developed system is evaluated in a specific application: the single trial classification of the P300 in an operator surveillance setup.

Keywords: Brain Computer Interface, Brain Reading, Event Related Potentials, P300, Signal Processing, Machine Learning, FPGA

1. Introduction

An important current development in general and biomedical signal processing is the unification of signal processing and machine learning techniques, and the application of these techniques in scenarios with difficult problems like biomedical signal processing[1]. Since these methods usually have high computational requirements, but nevertheless shall be utilized in mobile devices, reconfigurable application specific processing architectures need to be employed to meet the temporal and quantitative performance constraints[2]. However, the design and implementation of signal processing systems is a difficult task, so that

* Corresponding author. Tel.: +49 (0)421 178 45-6559; fax: +49 (0)421 178 45-6613.
E-mail address: hendrik.woehrle@dfki.de.

tools and frameworks need to be established to support the development of such systems. In this paper, we present the reconfigurable Signal Processing And Classification Environment (reSPACE) for the development of Field Programmable Gate Array (FPGA) based signal processing systems. Furthermore, we show how reSPACE can be used in a difficult application case: the development of a single System on Chip (SoC) for the detection of the P300 event related potential (ERP) in Electroencephalographic (EEG) data.

1.1. Relation to other Work: Brain Computer Interfaces and Mobile Brain Computer Interfaces

Nowadays, nearly all BCI systems use machine learning techniques to learn patterns in the data that are correlated with certain mental states of the subject. In the case of EEG based BCIs, a challenging problem is single trial analysis of ERPs[3]. As a part of the data preprocessing, a number of different processing steps and feature reduction methods are carried out, such as normalization, frequency and spatial filtering, trend-adjustment, smoothing and decimation[4], in order to improve the signal-to-noise ratio and reduce the number of features for the learning algorithm. Due to this complex processing, there is currently only a small number of mobile BCI systems, which usually only support simple paradigms (e.g. SSVEP detection) or small number of channels[5]. Since FPGAs become popular for signal processing tasks, there are some attempts to develop FPGA-based BCI systems. In [6], a first FPGA-based SSVEP single channel BCI system was developed. A first P300 FPGA BCI system was presented in [7], but in that case only a simple filter was performed in the reconfigurable logic, while most of the processing was performed in softcore processors.

1.2. Current Developments in Reconfigurable Signal Processing using FPGAs

In the last couple of years, FPGAs became increasingly popular for signal processing. However, the development of FPGA based signal processing systems is very complex and differs in various aspects from the development of standard software based systems. Therefore, frameworks that support the development of complex signal processing systems are needed to overcome these problems.

1.3. Outline of the Paper

In section 2, we present reSPACE and discuss its features. In chapter 3, we show how it can be used to construct a mobile EEG processing and classification system. The usage of the system is demonstrated in section 4.

2. The reSPACE Design and Architecture

To overcome the mentioned problems, we developed reSPACE. It allows the generation of application specific data flows (ASDFs), which act as a hardware accelerator for the application. The advantage of this approach is, that computationally expensive parts of an application can be expressed as an ASDF, since most of these parts, like digital filters and matrix-vector-operations, are simple regarding their structure. Therefore, they can be realized in specialized hardware structures and do not require the generality of an instruction based CPU.

2.1. Features of reSPACE

The framework is realized by combining two different techniques: hardware description languages (i.e. the Very High Speed Integrated Circuit Hardware Description Language (VHDL)) and the Xilinx System

Generator (SG) for DSP. Two different system architectures can be realized with reSPACE: ASDFs that are used standalone and ASDFs that are used as coprocessors in System on Chips (SoCs) (see Figure 1a). In this case, a generic CPU is combined with the ASDF. The CPU can be used for application parts that should be realized as software modules, like high level configuration and communication tasks, while the ASDF is used for the computational expensive tasks of the application. VHDL allows to combine generated systems easily with other hardware systems like custom bus interfaces, communication peripherals or any other intellectual property (IP) core. Since VHDL is very cumbersome and not well suited for the high level development of complex signal processing systems, reSPACE combines VHDL with SG. SG allows to focus on the specific application, and to numerically simulate and investigate the results of the system using Matlab/Simulink.

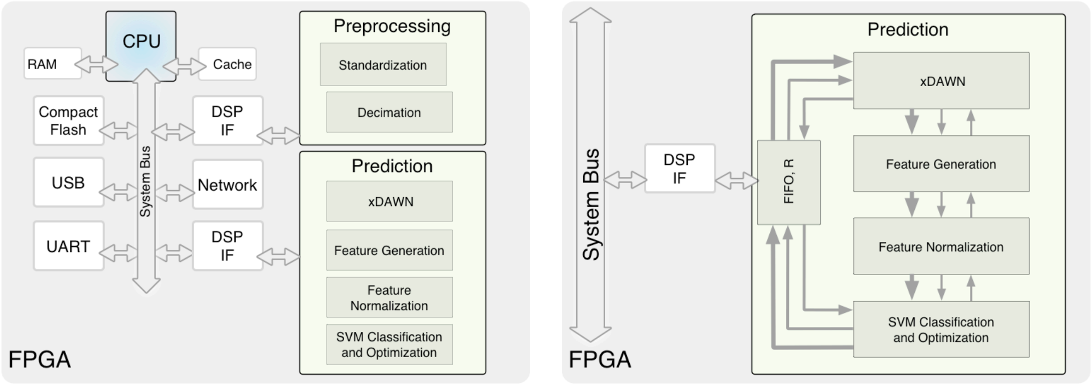

Fig. 1. (a) Example usage for the design of a SoC for the detection of the P300 ERP (b) Internal design of a ASDF for the prediction part of the P300 ERP. The ASDF is connected to the system bus with a FIFO buffer and configuration registers. The nodes are connected using a simple AXI-Stream like handshaking protocol.

Our framework is based on the synchronous dataflow model of computation (see Figure 1b). Therefore, the application specific accelerator is realized as a consecutive chain of nodes, where each node realizes a specific algorithm. The consecutive chain of nodes is called flow. A flow can be used independently, or connected to a system bus by a specific interface, the flow interface. The flow interface contains several registers to set and get parameters of the flow from outside, and input/output First-In-First-Out (FIFO) buffers for data transfer, which act as source and sink node, respectively. The nodes are connected using an AXI-Stream like protocol (i.e. data channel and enable/ready signals for handshaking). A major design goal of the reSPACE framework is generality. Therefore, all nodes provide a customizable high level configuration interface, that can be parameterized for the specific application. The available parameters depend on the specific algorithm, common examples are the input/output data fix point number format, number of multiplexed channels, or specific parameters, like filter coefficients or weight vector values. Currently, the framework is used with several different FPGA families: the Virtex-5, Spartan-6 and the Zynq processing system.

3. Using reSPACE for the Development of a Mobile Brain Computer Interface

We demonstrate the usage of reSPACE to construct a prototype mobile EEG classification system. The application of the developed system is the single trial detection of the P300 signal in electroencephalographic (EEG) data. This is a difficult signal processing task, since the EEG data has a low signal to noise ratio and therefore needs several different signal processing methods to detect the P300.

3.1. General System Architecture

For the construction of the mobile BCI system, we use a Xilix Zynq ZC7020 on a Zedboard evaluation platform. Zynq combines two ARM Cortex A9 processing cores with programmable logic, and is therefore well suited for our approach. The ARM cores allow the straightforward usage of standard software and operating systems. In our case, we use Linux (Kernel version 3.6.0 with a Linaro Ubuntu based root file system) as the operating system. This allows us to use the pySPACE framework for high level and user configuration. PySPACE is implemented using the Python scripting language and depends on some third party modules (Numpy, Scipy and pyYaml), which can be easily obtained using the package manager.

3.2. Realized Signal Processing Procedures

The data is passed through various different processing methods for signal enhancement and conditioning. These processing steps are performed one after another on each data segment. All these methods exist as software and hardware modules, and compared in section 5 regarding time and classification performance. In this application, we use two different ASDF: a preprocessing flow, which implements data independent signal processing methods, and a prediction flow with data dependent signal processing and classification methods. The data dependent parameters are obtained in a training session and computed in software.

3.3. Preprocessing flow

In this flow, we perform two operations: direct current (DC) offset removal and decimation. The DC offset of the data was removed using a notch infinite impulse response filter to remove the 0 Hz component. In the decimation node, the data was decimated from the initial 1000 Hz sampling rate to 25 Hz. To avoid aliasing effects, an anti-alias finite impulse response filter was applied before the sampling rate reduction.

3.4. Prediction flow

The first step in the prediction flow is the application of the XDAWN [8] spatial filter to reduce the number of 62 original channels to 8 signal channels. The next step is the Straight Line Description of the data, which is performed to further reduce the dimension in a descriptive way by fitting straight lines to the data using linear regression. All resulting values are arranged in a single feature vector, i.e. the channels are not considered in the subsequent nodes. The feature vector is standardized by dividing each single feature by the standard deviation and subtracting the mean of the corresponding features in the training data set and classified using a linear support vector classifier In this step, we do not map the actual output of the classifier to the corresponding class label, but pass the value to the next processing step. To cope with the different amounts of instances per class, we perform a threshold optimization. Therefore, we compare the continuous classification with a chosen threshold score, which we use as the decision boundary to assign the class label.

4. Evaluation of the System for the Single Trial Detection of Event Related Potentials in EEG Data

We evaluated our system on a P300 Oddball paradigm, which is described in the following.

4.1. Application Scenario

The experimental setup for the evaluation of the system is shown in Fig. 2. It allows the monitoring of the subjects EEG while it is under a high cognitive workload. The high cognitive workload is achieved because the subject has to perform a dual task: playing the BRIO® labyrinth game and react to certain optical stimuli at the same time. The setup of the scenario is as follows: the subject is sitting in front of a BRIO® which it has to actively control in this scenario.

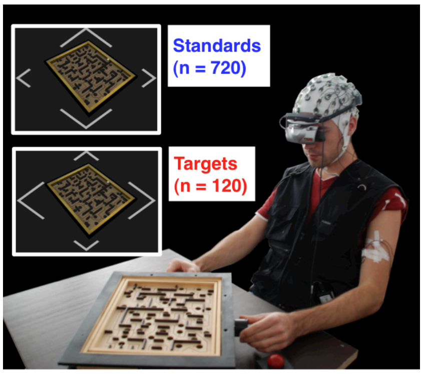

Fig. 2. The experimental setup. The subject has to react to given stimuli shown in the HUD while playing the BRIO Labyrinth game.

The subject wears a head mounted display (HMD), which displays a model of the game as well as certain symbols, which serve as visual stimuli for the subject. There are two kinds of stimuli: unimportant standard stimuli, which do not require a reaction of the subject, and different kinds of important target stimuli, which require him to press a buzzer that is placed next to the game. The target stimuli are shown infrequently among the standard stimuli in a fixed ratio of about 1:6. The inter-stimulus interval (ISI) was 1000ms with a random jitter of ±100ms. The used setup is conformant with the oddball paradigm, in which infrequent important stimuli evoke a P300 while frequent unimportant ones do not.

4.2. Experimental Procedures

Six subjects (males; mean age 27.5) took part in the experiments. The experiment was performed two times by each subject with at least one day of rest in between, generating two sessions per subject. In each session, each subject performed 5 runs with 120 *target* 1 stimuli (important information) and about 720 *standard* stimuli. While the subject was performing the task, the EEG was recorded (62 electrodes, extended 10-20 system with reference at FCz) using 62 channels of a 64 channel actiCap system (Brain Products GmbH, Munich, Germany). Impedance was kept below 5 kΩ. EEG signals were sampled at 1000 Hz, amplified by two 32 channel BrainAmp DC amplifiers and filtered with a low cut-off of 0.1 Hz. We used the first 3 runs of each session for the training of the signal processing methods and used the last 2 runs of the same session for testing, where we classified the correct perception of a *standard* versus a *target*.

4.3. Results

A comparison of the classification performance between the software based and FPGA based detection is shown in Fig. 3. The classification performance is measured as the balanced accuracy (BA), which is defined as 0.5 TPR + 0.5 TNR, where TPR and TNR denote the true positive rate and TNR rate, respectively. The mean time for the prediction part of data was 0.8 ms on the FPGA and 13.7 ms on the ARM processor. We achieved an average BA for all architectures of 0.879. The FPGA resource consumption regarding the number of consumed look up tables (LUT), block RAMs (BRAMs) and DSP48 slices is shown in Table 1.

Table 1. Resource usage of the FPGA system

Type of Hardware Device	LUTs	BRAMs	DSP48 slices
Preprocessing	6536	5	55
Prediction	3306	9	20
Total Used Amount of Resources	10289	14	75
Available Resources on Device	15960	140	220

5. Conclusions

We presented a powerful framework for the development of embedded signal processing systems and demonstrated its capabilities by implementing a SoC for single trial EEG classification. In comparison to the software modules, we achieve a high speedup of the processing time compared to the mobile ARM processor, but not in comparison to the Intel Core i7 desktop processor. The main amount of FPGA processing time is due to the transfer of the data from main memory to the processing flows, which reveals a future optimization point. The prediction accuracy is not affected by the used hardware architecture. Regarding the consumed FPGA resources, the Xilinx ZC702 is about 6-34% full, so that enough area for further extensions is available.

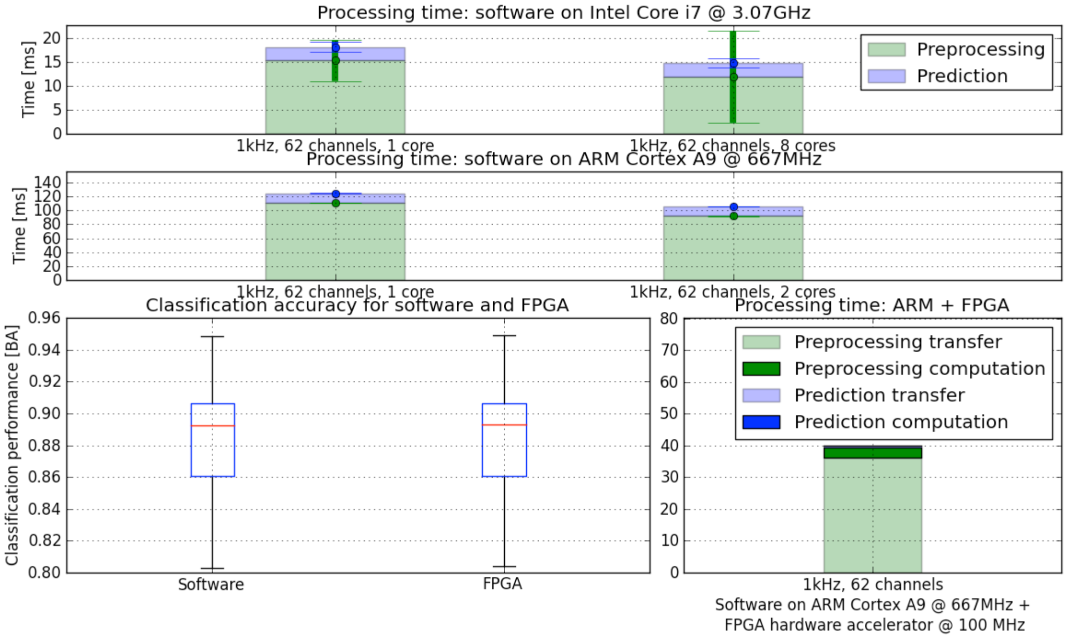

Fig. 3. Classification and run time performance of different hardware architectures for the P300 signal processing.

6. Future Work

In future, we will use our framework in different applications, e.g. real time prediction of movements and the processing of data from completely different sources, like robot sensor data. This type of architecture will be extremely valuable for robotic systems that need to simultaneously control a complex kinematical system while at the same time process sensor data from various channels to derive motoric actions[9][10][11][12].

Acknowledgements

Work was funded by the German Ministry of Economics and Technology (grant no. 50 RA 1011 and grant no. 50 RA 1012).

References

[1] Adali, T.; Miller, D.J.; Diamantaras, K.I.; Larsen, J.; , "Trends in Machine Learning for Signal Processing [In the Spotlight]," *Signal Processing Magazine, IEEE* , vol.28, no.6, pp.193-196, Nov. 2011

[2] Mansour, M.M.; Liang-Gee Chen; Wonyong Sung; , "Trends in Design and Implementation of Signal Processing Systems [In the Spotlight]," *Signal Processing Magazine, IEEE* , vol.28, no.6, pp.192-193, Nov. 2011

[3] Blankertz, B., S. Lemm, M. Treder, S. Haufe, und K. R. Müller. „Single-trial analysis and classification of ERP components–a tutorial“. *NeuroImage* 56, Nr. 2 (2011): 814–825.

[4] Bashashati, Ali, Mehrdad Fatourechi, Rabab K. Ward, und Gary E. Birch. „A survey of signal processing algorithms in brain computer interfaces based on electrical brain signals“. *Neural Engineering* 4, Nr. 2 (Juni 2007). doi:10.1088/1741-2560/4/2/R03.

[5] Lin, C.T., L.W. Ko, M.H. Chang, J.R. Duann, J.Y. Chen, T.P. Su, und T.P. Jung. „Review of Wireless and Wearable Electroencephalogram Systems and Brain-Computer Interfaces–A Mini-Review“. *Gerontology* (2009). doi:10.1159/000230807.

[6] Shyu, K. K., P. L. Lee, M. H. Lee, M. H. Lin, R. J. Lai, und Y. J. Chiu. „Development of a low-cost FPGA-based SSVEP BCI multimedia control system“. *Biomedical Circuits and Systems, IEEE Transactions on* 4, Nr. 2 (2010): 125–132.

[7] Khurana, K., P. Gupta, R.C. Panicker, und A. Kumar. „Development of an FPGA-based real-time P300 speller“. In *2012 22nd International Conference on Field Programmable Logic and Applications (FPL)*, 551 –554, 2012. doi:10.1109/FPL.2012.6339377.

[8] Rivet, B., A. Souloumiac, V. Attina, und G. Gibert. „xDAWN algorithm to enhance evoked potentials: application to brain–computer interface“. *Biomedical Engineering, IEEE Transactions on* 56, Nr. 8 (2009): 2035–2043.

[9] Bartsch, S; Birnschein, T; Cordes, F; Kuehn, D; Kampmann, P; Hilljegerdes, J; Planthaber, S; Roemmermann, M; Kirchner, F; , "SpaceClimber: Development of a Six-Legged Climbing Robot for Space Exploration," *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, vol., no., pp.1-8, 7-9 June 2010

[10] Spenneberg, D; Kirchner, F “Omnidirectional walking in an eight legged robot”, *Proc. Symp. on Robotics and Automation (ISRA), 2000*

[11] Kassahun, Y; Edgington, M; Metzen, JH; Sommer, G; Kirchner, F. “A common genetic encoding for both direct and indirect encodings of networks”, *Proceedings of the 9th annual conference on Genetic and evolutionary computation, 2007*

[12] Rommerman, M; Kuehn, D; Kirchner, F. “Robot design for space missions using evolutionary computation”, Evolutionary Computation, 2009. CEC'09. IEEE Congress on, 2098-2105