

# DEVELOPMENT OF AN FPGA-BASED REAL-TIME P300 SPELLER

Kanav Khurana, Pooja Gupta, Rajesh C. Panicker and Akash Kumar

Department of Electrical & Computer Engineering  
National University of Singapore, Singapore-117576  
Corresponding author email: akash@nus.edu.sg

## ABSTRACT

A Brain Computer Interface (BCI) is a system that allows direct communication between a computer and the human brain. Though the main application for BCIs is in rehabilitation of disabled patients, they are increasingly being used in other application scenarios as well. Most of the current BCI systems are based on personal computers. However, there is an increased interest in implementing BCIs for portable platforms as well, such as mobile phones and Field Programmable Gate Arrays (FPGAs) owing to low cost, power and portability. This paper proposes a low-cost FPGA based BCI speller application. The proposed system combines a stimulation panel, data acquisition and FPGA based real-time signal processing. The BCI system demonstrated here is a speller, which allows the user to use his/her brain signals to communicate directly with the application and spell out words by merely looking at the screen. The system achieves an accuracy of 65.37% when utilizing 2 rounds of data per character and an accuracy of 100% when utilizing 20 rounds of data per character.

*Index Terms*—Field-programmable gate array (FPGA), brain computer interface (BCI), P300, Real-time system.

## 1. INTRODUCTION

A Brain Computer Interface is a system that bypasses the body's normal neuromuscular pathways. Instead of depending on peripheral nerves and muscles, a BCI directly measures brain activity associated with the user's intent and translates the recorded brain activity into corresponding control signals for certain applications. A BCI system consists of a signal acquisition unit which records brain activity such as electroencephalogram (EEG), a stimuli presentation method (typically visual stimuli), a signal processing unit, and an application/prosthetic device. When the user attends to a stimulus, the intent is "captured" by an EEG recording system through electrodes placed on the user's scalp. The signals recorded by the system are processed and classified to recognize the intent of the user.

BCI systems have proven to be a boon for patients suffering from severe neuro-muscular disorders, such as Amyotrophic Lateral Sclerosis (ALS), Stroke, Spino-Cerebellar Ataxia (SCA) etc, who have difficulty in communicating

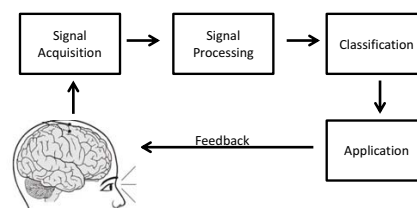


Fig. 1. Building Blocks of a Brain Computer Interface

vocally or through actions. Developing a system that enables the users to control or communicate with external devices without using their limbs reduces their dependence on a helper, and enables them to have a better quality of life.

BCI technology is being used in an increasing number of applications. For instance, Graimann *et al.* [1] developed "Roland III", a system which can help the user autonomously in a room. In another work, Valbuena *et al.* [2] demonstrated another robot, "FRIEND II". Disabled users could do activities such as pouring beverages into a glass using only intent and without any muscle movement. However, such systems require high-speed computers for signal processing and bulky monitors for displaying the stimulus, which limits their use and portability. In order to address these shortcomings, there has been an increased trend to move towards more portable and less expensive, power efficient and portable platforms such as mobile phones, FPGA's, etc [3]. Experiments have also been performed to implement signal processing algorithms and signal acquisition on mobile phones [4]. Although mobile devices ensure high portability, implementing these algorithms require use of expensive phones with high processing capability. Furthermore, mobile phones are usually not scalable and do not support custom refresh rates required for accurate stimulus presentation. FPGA's are also being explored as alternate platforms for implementing BCI systems. In [5], all the subsystems such as those for stimulus generation, signal acquisition and signal processing algorithms were implemented successfully on an FPGA. Some of the benefits of using an FPGA are that it can guarantee a scalable, low-power, standalone and cost effective system.

In a typical BCI system, certain brain activity patterns need to be elicited through external stimuli or through self-

modulation, which are then mapped to various commands. The various activity patterns include P300 (produced by a surprise stimuli), steady state visual evoked potentials (SSVEP, produced in response to flickering stimuli), mental imagery, slow cortical potentials etc [6]. The specific potential/activity pattern employed by the BCI dictates the hardware and software resources as well as the level of training required for the subject. In this paper, we propose a P300-based BCI Speller Application system implemented on a Xilinx Spartan 3E FPGA board. The application consists of a 6 x 6 grid with alphabets and characters. The main contributions of the proposed system are that the data acquisition, signal processing and stimuli generation are done on the same FPGA. Moreover, this BCI system is, to the best of our knowledge, the first FPGA-based P300 Speller application. While in the current setup data is sent from PC to simulate an on-line scenario, there are EEG systems available that are directly able to supply data over Ethernet.

In Section 2, we discuss some related works. Section 3 gives a short background of P300 event related potentials on which our system is based. Section 4 gives an overview of our system such as the experimental setup, system specifications in terms of resource requirements etc. Section 5 gives an in-depth view of the implementation of the system on the FPGA. This section details the basic building blocks of the system and the results of this experiment in terms of accuracy and response time. The paper concludes with Section 6 where we discuss future directions and possible enhancements.

## 2. RELATED WORK

There is a recent interest in realizing BCI systems which are more user friendly, cost effective, portable and efficient. In a study by Wang *et al.* [3], an SSVEP stimulus was realized on a mobile phone. The performance was evaluated by comparing the effectiveness of the stimulus with that on a tablet and a laptop. The setup consisted of a Bluetooth-enabled cell-phone and tablet, a mobile and wireless EEG device (in the study, a headband), and a computer screen. The stimulus consisted of a single flickering animation flashing at 11 Hz for a one-minute duration. The results indicated that though the tablet and the laptop were better than the cellphone, its performance was within acceptable norms. In another experiment, Wang *et al.* [4] explored the use of mobile phones in order to design truly portable and practical BCI systems. The study integrated a portable, wireless, low-cost EEG system and a cell-phone based signal processing platform to create a portable and practical SSVEP online BCI system. A wireless and battery-powered EEG headband was used to acquire and transmit EEG data of unconstrained subjects in real-world environments. The acquired EEG data was received by a regular cell phone through Bluetooth. The visual stimuli comprised of a 21 inch cathode ray tube (CRT) monitor (140 Hz refresh rate, 800 x 600 screen resolution) with a 4 x 3 stimuli matrix constituting a virtual telephone keypad which includes digits 0-9, BACKSPACE and EN-

TER. The stimuli frequencies ranged from 9 to 11.75 Hz with an interval of 0.25 Hz between two consecutive digits. The users were required to make a phone call by dialing an eleven digit number. Lee *et al.* [5] implemented a complete BCI on an FPGA. The stimulus consisted of a panel with four buttons each corresponding to particular tasks like “Play/Pause”, “Stop”, “Volume Up” and “Volume Down”. The study tested 7 subjects and required them to execute a sequence of commands by gazing at the stimuli. Experimental results verified the effectiveness of the proposed SSVEP-based BCI multimedia device control system as a low-cost SSVEP BCI “computer-free” system.

## 3. THE P300 EVENT RELATED POTENTIAL

P300 is a widely used brain activity pattern for BCIs. The user is asked to selectively look for the target/odd-ball stimulus amidst randomly sequenced stimuli flashing in succession. As an example, the subject can be told to look for a row/column containing the character ‘A’, in a sequence of randomly appearing rows/columns of alphabets. The occurrence of the target/task-relevant stimulus, i.e., the row/column containing ‘A’, elicits a positive deflection in the EEG after approximately 300 ms.

The fact that P300 can be evoked in nearly all subjects, and is relatively easy to elicit and detect simplifies interface design and permits greater usability [7]. Pattern recognition algorithms are used to classify the EEG data into recognizable commands. The classifier we use is FLDA, which projects the data linearly such that the projected means of the classes are far apart, while the spread of projected data is small [8].

## 4. SYSTEM OVERVIEW

The P300 BCI system is used to implement a speller operation that enables a user to type without using his hands. The stimulus is displayed on a monitor and consists of a 6 x 6 grid with alphabets A-Z and numbers 0-9.

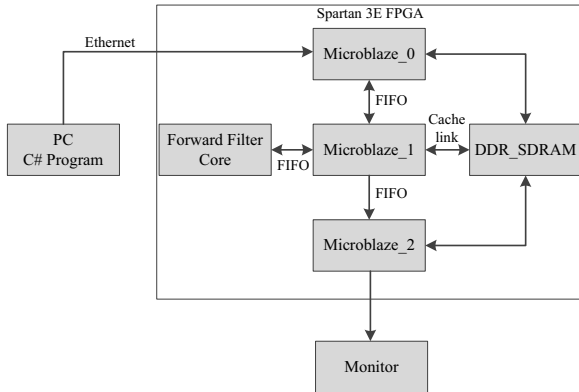
	7	8	9	10	11	12
1	A	B	C	D	E	F
2	G	H	I	J	K	L
3	M	N	O	P	Q	R
4	S	T	U	V	W	X
5	Y	Z	0	1	2	3
6	4	5	6	7	8	9

Fig. 2. 6 X 6 Grid

Each row and each column of the grid is highlighted in a pseudo-random sequence with an ISI (inter-stimulus interval) of around 187 ms. Whenever the character the user is focusing at is highlighted, a P300 response is evoked in the brain, which is detected in the EEG. Electrodes are placed at seven different areas on the scalp, viz, Cz, C3, C4, Pz, P3, P4 and Oz, according to the International 10-20 system [9].

#### 4.1. System Specification

The system is designed as a triple core system with three Microblaze processors. All three processors have shared access to the double data rate synchronous dynamic random-access memory (DDR SDRAM) and universal asynchronous receiver/transmitter (UART) peripherals. There are fast simplex links (FSL) for communication between the first Microblaze (Microblaze\_0) and the second Microblaze (Microblaze\_1), as well as the second Microblaze (Microblaze\_1) and the third Microblaze (Microblaze\_2).



**Fig. 3.** Block diagram of the developed system

The first Microblaze includes the Ethernet Medium Access Control (MAC) for high speed communication with the computer. The second Microblaze has cache memory. This processor has an FSL link for communication with the Forward Filter core. The third Microblaze has a TFT IP and displays output on a TFT monitor.

**Table 1.** FPGA UTILISATION

Resource	Slices	LUTs	BRAM
Microblaze_0	953	1823	0
Microblaze_1	1169	2268	4
Microblaze_2	852	1565	0
Filter Core	4019	5377	0

### 5. IMPLEMENTATION

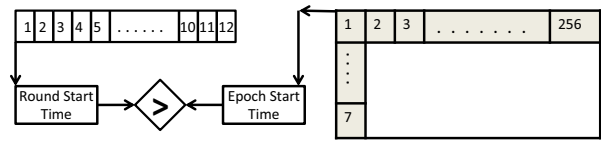
This section gives the implementation details of the system and the various building blocks involved.

#### 5.1. Algorithm

One “round” of recording is said to be complete when each row and each column of the 6 x 6 stimulus grid is highlighted once. This experiment is conducted for 720 such rounds, out of which the first 100 rounds are used to train the classifier, while the remaining 620 rounds are used to classify the subsequent brain signals into classes, or, in our case, predict which character is being gazed at. Furthermore, every 1 second of data recording, sampled at 256 Hz, from each of the seven EEG channels is marked as one “epoch”. Each epoch is time stamped by making a note of its start-time from a

fixed reference. The time of highlighting of each row and column is recorded as well. For all rounds, a target character is highlighted in white color. The user is supposed to focus on this character.

On the FPGA side, Microblaze\_2 (MB\_2) always runs in parallel and is responsible for the 6 x 6 stimulus. Microblaze\_0 (MB\_0) receives the first block of information from the PC in the form of TCP/IP packets. It compares the starting time of the epoch and the time for the first highlighting of the round as shown in Figure 4. If the starting time of the epoch is found to be less than that of the first highlighting, the FPGA discards that epoch. Conversely, if the starting time of the epoch is found to be greater than that of the first highlighting, MB\_0 passes both the structures, that is, “Signal\_Data” and “Order\_Data” to Microblaze\_1 (MB\_1) via the FSL\_0.1 Link.



**Fig. 4.** Order\_Data(left) and Signal\_Data(right) Comparison: Epoch Discarded

Hence, after passing the data for the first epoch, MB\_0 also passes the data for the next two epochs to MB\_1. In MB\_1, this received data is sent to the Forward Filter core via FSL\_FILT. The Filter Core returns the filtered values back to MB\_1. MB\_1 then down-samples these elements by a decimation factor of 8, down to 32 Hz from 256 Hz. Usually, data for a duration of 0.7 seconds from the start is considered to belong to a particular epoch. Hence, the data for the last 0.3 seconds is also discarded, further reducing the sample size by 0.7 times, or down to 23 samples.

#### 5.2. Data Transfer

The transfer of data between the PC and the Spartan 3E is managed by using the TCP/IP protocol. The PC is running an Intel i5 Processor at 2.3 GHz. A C# program makes use of the socket programming library ‘System.Net.Sockets’ to send/receive data. On the Spartan 3E board, the lightweight internet protocol (LWIP) stack is used for communication with the PC. The PC uses a flow control strategy of waiting for one acknowledgment between successive transmissions, to ensure that it does not overwhelm the FPGA.

#### 5.3. Forward Filter

The filter being used here is a forward Butterworth filter of order 3. The band-pass frequency range is between 0.5 Hz and 12 Hz. The data path for the VHDL filter is shown in Figure 5. Our implementation of this filter utilizes the FloPoCo [10] floating point arithmetic cores compliant with the single-precision IEEE-754 standard. The process of filtering begins with MB\_1 sending the epoch data to the filter core via the FSL\_FILT link. The filter’s transfer function

consists of 7 coefficients in both the numerator and the denominator.

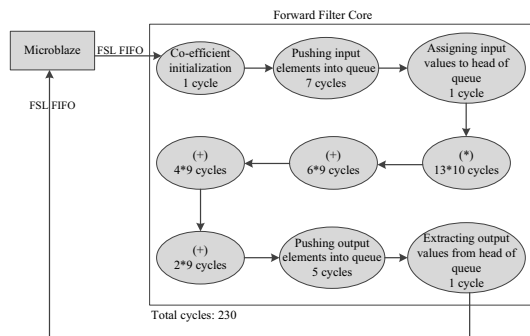


Fig. 5. Forward Filter Data Flow

#### 5.4. Matrix Inverse

The computation of FLDA involves the inverse of a 161 X 161 matrix. Due to low memory bandwidth and the recursive nature of the Determinant-Adjoint method, we have implemented the inverse using the Gauss-Jordan Algorithm. In this algorithm, the concerned square matrix is augmented with an identity matrix of the same dimension and then reduced to its ‘row echelon form’ yielding its inverse. This algorithm, for a large input square matrix, reduces the complexity of the operation from  $O(n!)$  to  $O(n^3)$ .

#### 5.5. Video Output

The TFT monitor displays a grid of 205 x 205 pixels with all characters displayed in black and the target character displayed in white. The rows are highlighted in pink color. Double buffering is implemented to tackle the problem of flickering when the rows or columns are being highlighted. The user is supposed to focus on the target character (the character which is highlighted in white color). This target is pre-determined for all the rounds. This setup is done in order to determine the accuracy of the system. The character predicted by the system is compared with the actual pre-determined target. Thereafter, the number of characters correctly identified are taken note of.

Table 2. Accuracy results for the P300 speller application

Matlab	Rounds/Char	2	4	5	10	20
Offline	Accuracy	65.37%	83.88%	93.66%	97.76%	100%
Matlab	Rounds/Char	2	4	5	10	20
Online	Accuracy	58.06%	69.03%	73.39%	82.26%	87.10%
FPGA	Rounds/Char	2	4	5	10	20
Online	Accuracy	58.06%	69.03%	73.39%	82.26%	87.10%

#### 5.6. Experimental Results

The classifier is tested for 1340 rounds, which implies that the classifier scores for 1340 characters are computed. In the experiment, the target character is same for 2 consecutive rounds and hence, 670 characters are detected by the system. By decreasing the number of rounds required to identify each character, the classification accuracy decreases. Conversely, on increasing the number of rounds, the accuracy

increases however, the time taken to predict each character increases. This is an unavoidable trade-off of this BCI system. The accuracy obtained by varying the number of rounds is tabulated in Table 2.

## 6. CONCLUSIONS AND FUTURE WORK

In this experiment, we implement a FPGA-based low-cost P300 speller. The system integrates the generation of a stimulus on a TFT monitor, signal acquisition and signal processing, all on an FPGA. The design allows online real-time processing of the P300 signal without a bulky personal computer. Finally, experimental results verify the effectiveness of the proposed P300-based BCI system through implementing the P300-based Speller application. The proposed system allows disabled patients to communicate more effectively with a computer/prosthetic device. Future directions include (1) getting the brain signals directly onto the FPGA from the EEG amplifier, and (2) offloading more pieces of code from C to VHDL to take advantage of the hardware acceleration. Certain computationally intensive functions like the matrix inverse and matrix multiplication may be executed more efficiently. This will reduce the time required for the system to detect and display the predicted character.

## 7. REFERENCES

- [1] B. Graimann, B. Allison, C. Mandel, T. Lüth, D. Valbuena, and A. Gräser, “Non-invasive brain-computer interfaces for semi-autonomous assistive devices,” *Robust Intel. Sys.*, pp. 113–137, 2008.
- [2] D. Valbuena, M. Cyriacks, O. Friman, I. Volosyak, and A. Graser, “Brain-computer interface for high-level control of rehabilitation robotic systems,” in *Proc. IEEE 10th Int. Conf. Rehab. Robotics, ICORR 2007*. IEEE, 2007, pp. 619–625.
- [3] Y. Wang, Y. Wang, and T. Jung, “A cell-phone-based brain-computer interface for communication in daily life,” *J. Neur. Eng.*, vol. 8, p. 025018, 2011.
- [4] —, “Developing stimulus presentation on mobile devices for a truly portable ssvep-based bci,” *Screen*, vol. 60, no. 59.975, pp. 55–575, 2010.
- [5] K. Shyu, P. Lee, M. Lee, M. Lin, R. Lai, and Y. Chiu, “Development of a low-cost FPGA-based SSVEP BCI multimedia control system,” *IEEE Trans. Biomed. Cir. and Sys.*, vol. 4, no. 2, pp. 125–132, 2010.
- [6] B. Graimann, B. Allison, and G. Pfurtscheller, *Brain-computer interfaces: A gentle introduction*. Springer, 2010.
- [7] X. Yang, J. Dai, H. Zhang, B. Wu, Y. Su, W. Chen, and X. Zheng, “P300 wave based person identification using lqv neural network,” *J. Conver. Info. Tech.*, vol. 6, no. 3, 2011.
- [8] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification II edition*. John-Wiley and Sons, 2001.
- [9] H. Jasper, “The ten-twenty system of the international federation,” *Electroenceph. Clin. Neurophysiol.*, vol. 10, pp. 371–375, 1958.
- [10] F. de Dinechin. *et al.*, “Flopoco, a generator of arithmetic cores for fpgas,” 2012. [Online]. Available: <http://flopoco.gforge.inria.fr/>