Live Demonstration: A Scalable 32-Channel Neural Recording and Real-time FPGA Based Spike Sorting System

Ian Williams*, Song Luan*, Andrew Jackson† and Timothy G. Constandinou*

*Centre for Bio-Inspired Technology, Imperial College London, †Institute of Neuroscience, Newcastle University
Email: {i.williams10, s.luan, t.constandinou}@imperial.ac.uk, andrew.jackson@ncl.ac.uk

Abstract—This demo presents a scalable a 32-channel neural recording platform with real-time, on-node spike sorting capability. The hardware consists of: an Intan RHD2132 neural amplifier; a low power Igloo® nano FPGA; and an FX3 USB 3.0 controller. Graphical User Interfaces for controlling the system, displaying real-time data, and template generation with a modified form of WaveClus are demonstrated.

I. INTRODUCTION

This demo presents a scalable neural recording and real-time spike sorting system implementing a two stage hybrid approach to spike sorting [1] that enables the spike sorting to be performed on a low-power Field Programmable Gate Array (FPGA). During the first stage, the raw spike train is streamed to a PC which performs high performance, offline clustering of the data. The results of the clustering are used to generate templates that are uploaded to an FPGA. In the second stage, the FPGA can then perform a computationally much simpler (and lower power) template matching task for spike sorting.

II. DEMONSTRATION SETUP

This demonstration (Fig. 1) consists of: (1) a Neural Interface Board (NIB) housing an Intan Technologies RHD2132 neural amplifier and a low power Igloo® FPGA for online spike detection and template matching; (2) a data interface board (DIB) together with a Cypress® FX3 USB 3.0 controller for transmitting the data to a computer over USB 3.0. (3) Graphical User Interfaces (GUI) for controlling the system, displaying real-time raw neural signal & template matched spike events, and for performing template generation with WaveClus [2]. Another version of the system using Low Voltage Differential Signalling (LVDS) will also be presented, to enable visitors to compare performance.

The system is demonstrated using synthetic neural data (generated using the Neurocube package [3]) that is injected with a neural signal simulator consisting of a microcontroller evaluation board and a multi-channel DAC (AD5380) combined with ~ 30 dB attenuation per channel. The NIB and DIB boards can be powered 3 different ways (to demonstrate the noise impact on performance): (1) NIB on battery, DIB on isolated DC/DC power derived from FX3 evaluation board; (2) NIB and DIB on battery; (3) NIB and DIB on the isolated DC power. An ammeter will show the current in the demo.

To demonstrate scalability, a second DIB board can be stacked onto the first DIB for 64-channel operation. The system can be expanded to at least 1,024 channels.

Fig. 1. Setup for demonstrating real-time FPGA based spike sorting system with recorded neural signal playback. NIB: neural interface board. DIB: data interface board.

III. VISITOR EXPERIENCE

The visitor will experience a walkthrough of the whole flow of operation using a neural signal generator, including:

- 1) Locating a neuron (mimicked by inactive/active channels) aided by a clicking sound
- 2) Recording and viewing neural signals
- 3) Choosing spike thresholds, generating templates and choosing template thresholds
- 4) Configuring the amplifier and FPGA to different bandwidth and templates
- 5) Comparing the performance between standard CMOS signalling and LVDS based system
- 6) Operating the boards on different power source and observing the power consumption
- 7) Scaling the system from 32 to 64 channels

In addition, other supporting devices, e.g. the in-house designed microcontroller based multi-channel neural signal simulator, are also available for visitors to discuss.

REFERENCES

- [1] J. Navajas, et al., “Minimum requirements for accurate and efficient real-time on-chip spike sorting,” *Journal of Neuroscience Methods*, vol. 230, pp. 51–64, 2014.
- [2] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, “Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering,” *Neural computation*, vol. 16, no. 8, pp. 1661–1687, 2004.
- [3] L. A. Camuñas-Mesa and R. Q. Quiroga, “A detailed and fast model of extracellular recordings,” *Neural computation*, vol. 25, no. 5, pp. 1191–1212, 2013.

A Scalable 32 Channel Neural Recording and Real-time FPGA Based Spike Sorting System

Ian Williams*, Song Luan*, Andrew Jackson† and Timothy G. Constandinou*

*Centre for Bio-Inspired Technology, Imperial College London, †Institute of Neuroscience, Newcastle University
Email: {i.williams10, s.luan, t.constandinou}@imperial.ac.uk, andrew.jackson@ncl.ac.uk

Abstract—This paper presents a scalable 32-channel neural recording platform with real-time, on-node spike sorting capability. The hardware consists of: (1) an Intan Technologies RHD2132 neural amplifier; (2) a low power Igloo® nano Field Programmable Gate Array (FPGA) for spike detection and template matching; and (3) an EZ-USB FX3™ peripheral controller for transmitting the data to a computer over USB 3.0. The system is implemented in 2 parts, the first board (headstage, 51 mm × 33 mm) contains the RHD2132 and FPGA and is intended to be in close proximity to the animal, while the second (80 mm × 56 mm) is a combination of an FX3 development board with a custom interface shield. Graphical User Interfaces (GUI) for controlling the system, displaying real-time raw & template matched data, and for template generation with WaveClus. Noise performance of below 1 LSB ($\sim 3 \mu\text{V}$) was measured and system power consumption (excluding FX3) of 15.1 mW while idle, 34 mW while outputting raw data and 34 mW while outputting template matched results.

I. INTRODUCTION

Neural recording is a key enabling technology for neuroscience and in emerging neuroprosthetics applications. The capability to identify which neurons are firing, their pattern of firing, and their order of firing, enables behavioural descriptions of individual neurons to be generated and linked to wider subject or neural-network activity.

Neural recording technology often imposes limitations on the conduct of these experiments, including: the number of electrodes (and hence neurons) that can be recorded from; the freedom of motion of the subject (due to bulky headstages or non-portable rackmount equipment); and the latency with which neural activity information is available to the experimenter (due to the requirement to de-interleave neural spike trains offline – known as spike sorting). To address these issues there is research into miniaturised high channel count devices and automated spike processing [1], [2].

Micro-Electrode Arrays (MEAs) with hundreds or even thousands of electrodes are becoming available [1], [3] and front end neural amplifiers are tracking this trend [3]. However, scaling these aspects of the system brings major challenges in terms of increased data rate and power consumption (affecting thermal dissipation and battery life). Low power spike sorting co-located with the front end amplifier has the potential to dramatically reduce data transmission bandwidth requirements and power consumption [4], [5] as well as providing low latency neural activity data to the experimenter [2].

High performance automatic spike sorting algorithms exist [6], [7], but are not currently suitable for implementation

on standard low power microprocessors or FPGAs. Previous published approaches for neural spike-sorting have therefore demonstrated that (up to 64 channels) can be processed using custom ASIC designs, novel clustering algorithms and a variety of techniques to efficiently use resources [8], [9]. However, a novel 2 stage hybrid approach leveraging all the performance of standard algorithms was described in [2] and is implemented here in low power commercial off the shelf hardware. The system presented is targeted at 32 channels, but the architecture is scalable to at least 1,024 channels.

The remainder of this paper is organised as follows: Section II describes the hybrid approach, the system architecture and outlines the top level function of the system components; Section III describes the function and design of the system in more detail; Section IV presents example recorded data, and noise & power measurements; and finally Section V summarises the achieved system.

II. SYSTEM OVERVIEW

A. Hybrid Approach

Fig. 1. Spike sorting operation. Raw data (a) is filtered, amplified and digitised by a neural amplifier. The filtered data (b) is then digitally processed to detect spikes which are aligned by their peaks (c) before being matched (d) against templates created in WaveClus [6]

De-interleaving a spike train typically involves a computationally demanding clustering process and has been a major challenge for previous online spike sorting systems. Here a two stage hybrid approach to spike sorting [2] is utilised to circumvent this problem. In stage 1, filtered neural signal is streamed to a PC which performs a high performance, offline clustering of the data. The results of the clustering are used to generate templates that are uploaded to an FPGA. In stage 2 the FPGA can then perform a computationally much simpler (and lower power) template matching for spike sorting as shown in Fig. 1.

B. Outline Architecture

The system (see Fig. 2) consists of 2 custom PCBs, combined with a commercial FX3 SuperSpeed Explore Kit and

Fig. 2. (a) System hardware overview, (b) Data flow through the system.

a PC – the functions of each are described below. Standard HDMI and USB3 cables are used to link the boards.

1) *Neural interface board (NIB)*: The NIB is designed to sit close to the experimental subject (e.g. as part of a headstage) and either streams digitised neural signal (stage 1) or performs spike sorting and outputs spike events (stage 2).

2) *Data interface board (DIB)*: The DIB isolates the NIB during *in-vivo* trials for safety and performance reasons. The board also provides physical interface connections and is capable of providing power to the NIB (using an isolated DC/DC converter).

3) *FX3 Evaluation board*: In the presented system the FX3 simply performs as an SPI to USB interface. However, this device is a key aspect of the system’s scalability as it has 32 GPIOs which can be configured as 32 serial data input lines – enabling 1,024 channels to be connected simultaneously. This board is powered by the computer.

4) *Computer*: The computer is used for data storing and visualisation, offline spike sorting and system configuration.

III. SYSTEM DESIGN

A. Neural Interface Board

Fig. 3. Neural Interface Board. Intan technologies RHD2132 neural amplifier and Igloo[®] FPGA on separate sides. Size: 50.9 mm × 32.5 mm

The neural interface board (Fig. 3) is implemented in a 4-layer PCB board. Miniaturisation was desirable, but was limited by: 1) use of a 40-way standard pitch header (for easy testing); 2) large form factor FPGA (see Section III-C); and conservative separation of analogue and digital components for noise minimisation. The top layer of the board houses the neural amplifier, the connector and a low noise LDO; the second layer contains an analogue ground plane; the third layer

is a solid digital ground plane; and the bottom of the board houses the FPGA and supporting circuitry.

The board can be either powered by the battery (3.7 V) or via an isolated DC/DC converter on the DIB through the HDMI cable. When powered on battery, the analogue power supply (3.3 V) is from a low noise LDO while the FPGA core (1.5 V) and digital I/O supplies are from a dual DC/DC converter.

B. Data Interface Board

Fig. 4. Data Interface Board. Designed as a shield of the FX3 evaluation board. Size: 80.3 mm × 56.0 mm

The data interface board (Fig. 4) is also a 4-layer board with isolated power and ground domains to ensure safety during *in-vivo* experiments. Two HDMI connectors provide single ended communication and Low Voltage Differential Signalling (LVDS) communication respectively. The board can be directly mounted on the Cypress[®] FX3 SuperSpeed Explorer Kit-003.

The FX3 acts as SPI master with each of its 32 GPIOs capable of acting as MISO for a separate NIB, enabling it to read 32 sets of 16-bit words in parallel. Only one NIB and hence one MISO pin is used for this 32 channel system. The incoming data is ping-pong buffered so that the FPGA output buffer does not overflow, and Direct Memory Access (DMA) is used to transmit the data over USB bulk transfer.

C. FPGA Design

Fig. 5. The FPGA functional blocks. Colours indicate main data flows in different modes: green dots for configuration, blue dashes for pass through and red solid for template matching. Shaded blocks are RAM modules.

The system was initially implemented on an Igloo[®] Nano FPGA AGLN250v2, which was chosen for its low power and small form factor (minimum 5 mm × 5 mm). However, it only supports single ended I/O, whereas LVDS is recommended for best noise performance of the neural amplifier. To investigate the noise impact of the communication physical layer a second version of the system was created using the pin compatible

Igloo AGL250v2 (N.B. not Nano) which does support LVDS. To maintain commonality across the two boards it was necessary to use a larger form factor IC (14 mm × 14 mm).

The FPGA operates in 3 distinct modes (see Fig. 5): 1) a configuration mode; 2) a pass through readout mode; and 3) a template matching readout mode. A state machine coordinates the switching between these 3 modes (directing flows of data between the amplifier, FPGA modules and FX3). Due to limited memory on the FPGA and to increase the efficiency of operation (with minimal impact on performance [2]), the 16-bit data samples were truncated to 9 bits (to fit 9-bit word RAM modules). The truncated sample size decreases the input range (to ± 0.8 mV) and the resolution (to ~ 3 μ V), however, the input referred noise of RHD2132 is ~ 2.5 μ V and for our signals of interest the input range is sufficient, so these were considered to be acceptable compromises.

In the configuration mode, the FPGA either re-transmits incoming data to the RHD2132 to fill configuration registers, or sets: channel specific spike thresholds (288 bits); spike templates (4, channel specific, 16 sample spike waveform templates – 18,432 bits); and template thresholds (template specific sum of absolute error values for determining if a spike and template are a match – 1,792 bits).

In the readout modes the system is configured to sequentially query each of the 32 channels. Timestamping is implemented as a 5-bit count that increments once all 32 channels have been queried. At the sampling frequency of 15 kHz used here, this gives a resolution of 66.7 μ s.

In the pass through mode, the FPGA processing simply truncates the 16-bit sample, prepends the channel number (a 5-bit word) and transmits the data out over the SPI to the FX3.

In the template matching mode, incoming data is first truncated and then stored in a rolling buffer (32 samples per channel). This data is then passed to a channel specific spike detection state machine which determines whether a spike has been detected (configurable as between 1 and 4 samples above channel specific threshold), where the spike peak is and how many samples after the peak have been stored. Once a spike has been detected and sufficient samples stored (8 after the peak), the state machine updates a read pointer (setting it to 7 samples before the peak giving 16 samples total), stores the current timestamp and enqueues the channel number to the 32 depth 5-bit Channel FIFO (Fig. 5).

If there is data in the Channel FIFO the template matching module reads in the channel number and sequentially accesses the appropriate buffered samples. Each spike is compared sample by sample to 4 stored templates and the sum of absolute difference for each template is calculated. The sum for the best matching template (i.e. lowest total difference) is then compared to the template threshold. If it is below threshold it is considered a match and the channel (5-bit), timestamp (5-bit) and template (2-bit) are transmitted to the FX3 as part of a 16-bit SPI word. If it is above threshold it is discarded. There is one template matching module on the FPGA and it takes 10 μ s to process each spike (making it capable of processing 100,000 spikes/s), however, spike

windowing (requiring a peak followed by 8 samples) mean that the maximum spiking rate per channel is $1/9$ th of the sampling frequency, which in this example gives a total maximum throughput of just over 53,000 spikes/s for all 32 channels.

Time synchronisation and output buffer underflow is dealt with by transmission of a reserved data word and timestamp whenever there is no data to send, as well as a second reserved data word when the timestamp rolls over.

The system was implemented on the FPGA using 3 clock domains, the majority of the processing is carried out at 2 MHz, while the main state machine and writing to configuration RAM blocks operates at 5 MHz, the third clock domain is for the SPI interfaces which are driven by the SPI clock (8 MHz).

D. PC host software

The main PC application is implemented in C# using a USB 3.0 driver provided by Cypress®. Fig. 6 shows the GUIs. The configuration GUI shows programmed thresholds and templates overlapped with a spike scope (designed for use during electrode insertion) which beeps and shows detected spikes when a threshold crossing occurs on the selected channel.

Fig. 6. Snapshots of the GUIs in operation. From top: 1) Spike sorting GUI; 2) Configuration GUI; and 3) Signal live stream. All generated with synthetic signals for demo purposes.

The critical part of the application is the live streaming of the FPGA data. Different threads are used for different critical tasks to avoid losing data packets. The simplified program flowchart is shown in Fig. 7.

The data acquisition thread handles the communication with the FX3 and its output is passed to the data processing threads. Together these threads implement a data processing pipeline. A computationally significant part of this pipeline is the transposition of a 16×32 -bit matrix into 32 sets of 16-bit words. This results from the 32 parallel SPI input streams (for a 1,024 channel system with 32 NIBs) which are read in as words to make efficient use of the FX3s 32-bit GPIF data bus.

Fig. 7. PC host application flow chart. “[]” shows optional tasks.

For this 32-channel system the data from the active device is decoded and the extracted information is written to the display FIFO and stored onto disk. The display thread plots the data in real-time, using GPU acceleration to minimise the display delay and alleviate CPU pressure.

The spike sorting GUI is based on WaveClus [6] and modified to process 32-channels of data in fully automatic mode, but also providing the user full control of the key spike detection and clustering parameters. The output of this software is a configuration file that contains spike thresholds, templates and template thresholds to be loaded into the FPGA.

IV. RESULTS

Fig. 8. Example of recorded synthetic data and close up of a single spike.

For testing purposes a DAC and ~ 30 dB attenuator were used to playback simulated data into the input of the amplifier. Fig. 8 shows an example of the data recorded using this setup and illustrates how little digital and thermal noise is present. Noise measurements were carried out by connecting the RHD2132 inputs to ground, recording data and calculating RMS variation. Measurements conducted both in an EM-shielded anechoic chamber and office environment showed RMS noise of below 1-LSB ($\sim 3 \mu\text{V}$ which approaches the $2.5 \mu\text{V}$ input referred noise quoted for the RHD2132). The performance of both single ended and LVDS boards was not significantly different. Supplying power through the isolated DC/DC converter also did not significantly affect the noise.

The LVDS system, however, has a major drawback in terms of its power consumption (almost entirely due to the communication protocol, but also due to the higher power consumption of the LVDS capable FPGA). While the single ended solution (NIB+DIB) consumed 15.2, 34 and 34 mW during standby, pass-through and template matching respectively, the equivalent figures for the LVDS solution were 130, 220 and 220 mW. Calculations indicate that this difference is almost exclusively due to the static current draw of the LVDS transmitting and receiving circuits. The power consumed by the FX3 USB controller is drawn from USB and not included.

TABLE I
SYSTEM SPECIFICATIONS

Parameter	Value
Dimensions, Weight	51 mm $\times$ 33 mm, 12g
Num. of channels (per headstage)	32 (Intan RD2132 front-end)
Power (standby, recording, sorting)	15 mW, 34 mW, 34 mW
Max. latency (spike sorted output)	0.24 ms
Communication protocol	SPI (internal) and USB 3.0
SPIKE SORTING METHOD	
Clustering and offline training	WaveClus, template building
Online classification	Template matching
Spike detection, alignment	Single threshold, peak
TEMPLATE MATCHING	
Template memory, organisation	18.4 kbit, 4 templates/channel
Template length	16×9 -bit words
Matching quantifier	Sum of absolute difference

Simulated latency for spike sorting on the FPGA was $\sim 15 \mu\text{s}$, however, as the number of spikes increases so does the latency, reaching a peak of $240 \mu\text{s}$ (although due to the timestamping and relatively slow frame rate of monitors these latencies are not visible in the GUI or in the recorded data).

V. CONCLUSION

The system presented here is an efficient 32 channel spike sorting system using a recently proposed hybrid approach. The system demonstrated the feasibility of low power FPGA based spike sorting in real time and showed good noise performance. On the basis of these results we expect a 1,024 channel system to be demonstrated *in-vivo* in the near future. The main system specifications are summarised in Table I.

ACKNOWLEDGEMENT

This research has been funded by the Engineering and Physical Sciences Research Council (EPSRC) grant refs: EP/I000569/1 and EP/K015060/1.

REFERENCES

- [1] D. A. Schwarz, et al., “Chronic, wireless recordings of large-scale brain activity in freely moving rhesus monkeys,” *Nature Methods*, vol. 11, no. 6, pp. 670–676, 2014.
- [2] J. Navajas, et al., “Minimum requirements for accurate and efficient real-time on-chip spike sorting,” *Journal of Neuroscience Methods*, vol. 230, pp. 51–64, 2014.
- [3] P. M. Ballini, et al., “A 1024-channel cmos microelectrode array with 26,400 electrodes for recording and stimulation of electrogenic cells in vitro,” *IEEE J. Solid-State Circuits*, vol. 49, no. 11, pp. 2705–2719, 2014.
- [4] A. Eftekhari, S. E. Paraskevopoulou, and T. G. Constantinou, “Towards a next generation neural interface: Optimizing power, bandwidth and data quality,” *Proc. IEEE BioCAS*, pp. 122–125, 2010.
- [5] D. Y. Barsakcioglu, et al., “An analogue front-end model for developing neural spike sorting systems,” *IEEE Trans. BioCAS*, vol. 8, no. 2, pp. 216–227, 2014.
- [6] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, “Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering,” *Neural computation*, vol. 16, no. 8, pp. 1661–1687, 2004.
- [7] S. N. Kadir, D. F. Goodman, and K. D. Harris, “High-dimensional cluster analysis with the masked em algorithm,” *Neural Computation*, 2014.
- [8] V. Karkare, S. Gibson, and D. Markovic, “A 75- μW , 16-channel neural spike-sorting processor with unsupervised clustering,” *IEEE J. Solid-State Circuits*, vol. 48, no. 9, pp. 2230–2238, 2013.
- [9] M. Saeed and A. M. Kamboh, “Hardware architecture for on-chip unsupervised online neural spike sorting,” *Proc. IEEE/EMBS Conf. Neural Engineering (NER)*, pp. 1319–1322, 2013.