ARCHITECTURE DESIGN FOR A NEURAL SPIKE-BASED DATA REDUCTION PLATFORM PROCESSING THOUSANDS OF RECORDING CHANNELS

A Dissertation
Submitted to
the Temple University Graduate Board

In Partial Fulfillment
of the Requirements for the Degree
DOCTOR OF PHILOSOPHY
OF ENGINEERING

---

by
Nashwa Elaraby
May 2014

Examining Committee Members:

Dr. Iyad Obeid, Advisory Chair, ECE Department
Dr. Dennis Silage, ECE Department
Dr. Joseph Picone, ECE Department
Dr. Prawat Nagvajara, External Member, Drexel University

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

ABSTRACT

Simultaneous recordings of single and multi-unit neural signals from multiple cortical areas in the brain are a vital tool for gaining more understanding of the operating mechanism of the brain as well as for developing Brain Machine Interfaces. Monitoring the activity levels of hundreds or even thousands of neurons can lead to reliable decoding of brain signals for controlling prosthesis of multiple degrees of freedom and different functionalities. With the advancement of high density microelectrode arrays, the craving of neuroscience research to record the activity of thousands of neurons is achievable. Recently CMOS-based Micro-electrode Arrays MEAs featuring high spatial and temporal resolution have been reported.
The augmentation in the number of recording sites carries different challenges to the neural signal processing system. The primary challenge is the massive increase in the incoming data that needs to be transmitted and processed in real time. Data reduction based on the sparse nature of the neural signals with respect to time becomes essential.

The dissertation presents the design of a neural spike-based data reduction platform that can handle a few thousands of channels on Field Programmable Gate Arrays (FPGAs), making use of their massive parallel processing capabilities and reconfigurability. For Standalone implementation the spike detector core uses Finite State Machines (FSMs) to control the interface with the data acquisition as well as sending the spike waveforms to a common output FIFO. The designed neural signal processing platform integrates the application of high-speed serial Multi-Gigabit transceivers on FPGAs to allow massive data transmission in real time. It also provides a design for autonomous threshold setting for each channel.

To

Tamer, Billal and Eyad

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor Dr. Iyad Obeid. Dr. Obeid has helped me tirelessly through the long journey of my PhD work. His constant motivation, understanding, encouragement and guidance are greatly appreciated. He sets an example of a highly motivated successful professor and it was an honor working with him on my PhD research.

I would also like to extend my gratefulness to Dr. Dennis Silage. He has been very generous sharing his expertise and knowledge that seem to have no boundaries. I am very thankful to Dr. Joseph Picone for his insightful comments and valuable input. I really appreciate your time and constructive advice.

I would like to thank Dr. Alessandro Maccione from IIT for supplying me with the high-density MEA neural data recordings. His contribution was an asset to the dissertation work.

I am blessed with two great parents, Soheir Nour and Nabil Elaraby, who have supported and encouraged me and never lost faith that I can do it.

I could not have completed my PhD work without the love, support and patience of my small family. I would like to express my heart-felt gratitude to my husband, Tamer who has been a constant source of support and strength all these years. To you, Billal and Eyad, I dedicate this dissertation.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

What beauty is shown in the preparations obtained by the precipitation of silver dichromate deposited exclusively onto the nervous elements! But, on the other hand, what dense forests are revealed, in which it is difficult to discover the terminal endings of its intricate branching… Given that the adult jungle is impenetrable and indefinable, why not study the young forest, as we would say in its nursery stage.

Santiago Ramón y Cajal (1852-1934)

Information processing in the brain is carried out by large groups of interconnected neurons. Neurons are the cells responsible for encoding, transmitting, and integrating signals originating inside or outside the nervous system. The transmission of information within and between neurons involves changes in the resting membrane potential, when compared to the extracellular space. The inputs one neuron receives at the synapses from other neurons cause transient changes in its resting membrane potential, called postsynaptic potentials. These changes in potential are mediated by the flux of ions between the intracellular and extracellular space. The flux of ions is made possible through ion channels present in the membrane. The ion channels open or close depending on the membrane potential and on substances released by the neurons, namely neurotransmitters, which bind to receptors on the cell's membrane and hyperpolarize or depolarize the cell. When the postsynaptic potential reaches a threshold, the neuron produces an impulse. The impulses or spikes, called action potentials, are characterized by a certain amplitude and duration and are the units of information transmission at the

interneuronal level [1]. The discovery of the neuron was a milestone in brain research and paved the way for modern neuroscience, but the brain is yet to yield the vast majority of its secrets.

Current neuroscience research operates at two separate levels: The macro- and microscopic levels. The macroscopic level uses imaging techniques like functional Magnetic Resonance Imaging (fMRI) and Magnetoencephalography (MEG) to measure regional changes in metabolism and blood flow associated with changes in brain activity. It captures whole brain activity patterns that allow the mapping of brain regions associated with a particular behavior or task. These techniques lack single-cell details and the requisite temporal resolution to permit detection of neuronal firing patterns. The microscopic level is concerned with investigating how individual nerve cells work, studying their response to stimulation and monitoring the firing rates associated with a certain behavioral output, mental state or motor activity. This can be done using implanted electrodes to record the rates and timing of action potentials. The sparse sampling of neuronal activity monitoring tens to few hundreds of neurons does not give the global view of signaling in neural circuits that can involve millions of neurons.

There is a gap between the two levels, that is believed to entail an answer to the question of how neuron cells collaborate to process information. To fill in the gap, we need a static anatomical map of the brain circuitry describing the synaptic connections within any given brain area, as well as a dynamic map revealing the patterns and sequences of neuronal firing by all neurons over time scales on which behavioral outputs or mental states occur. Hence the aspiration is not only to map the "impenetrable jungle"

that Cajal referred to but also to map the dynamical traffic within the jungle and analyze it. Research efforts are conducted to approach that ultimate goal, and along the hard path to achieve it, technological breakthroughs evolved and more are bound to arise. New technologies may include new optical techniques to image in 3D, new capabilities for storage and manipulation of massive data sets, new clinically viable brain-machine interfaces to help paralyzed patients and development of biologically inspired computational devices [2].

Focusing on the microscopic level, two of the research fields concerned with recordings of the spiking activity of neurons using microelectrode arrays are: The Brain-Machine Interface (BMI) and brain in a dish research fields.

Brain-Machine Interface:

Extracting motor control signals from the firing patterns of populations of neurons and using these control signals to reproduce motor behaviors in artificial actuators are the key operations of a brain-machine interface [3,43]. The typical neural signal processing pathway as shown in Fig.1.1 is designed to measure the instantaneous frequency of neural action potentials, or spikes. Since any given electrode may sense spikes from multiple neurons, it is typically necessary to sort all detected spikes by wave shape (i.e. by neuron). Firing rates of sorted spikes are typically measured by moving average; these rates can then be used by "decoding" algorithms which use statistical models to correlate spiking activity with behavioral or motor activity in the subject.

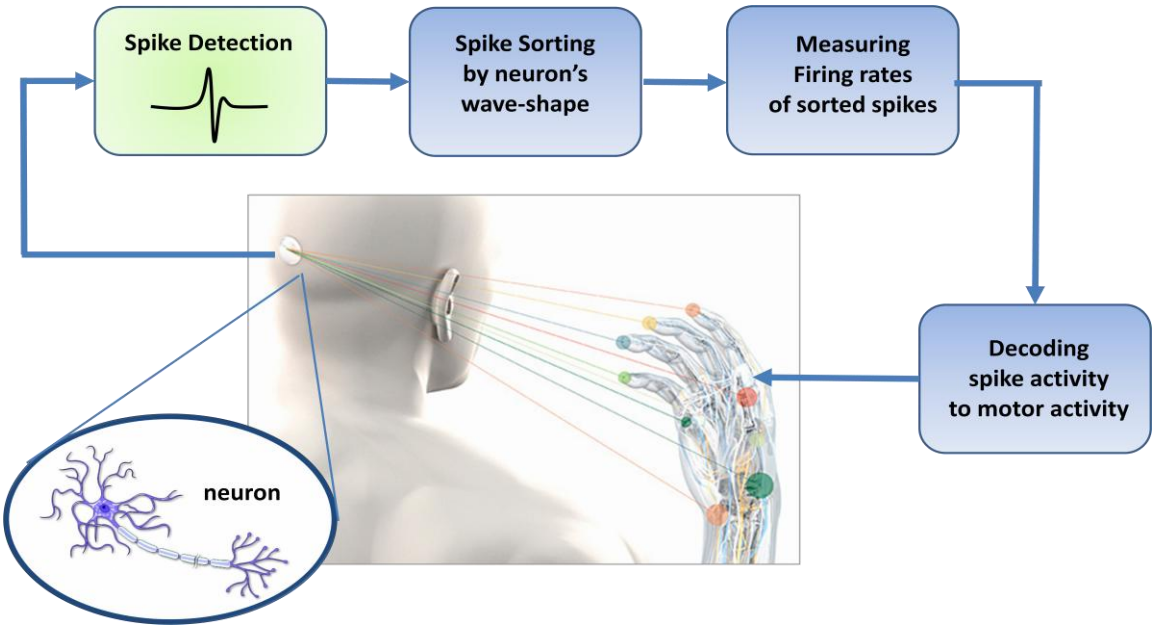


Fig.1.1 Block diagram of the typical pathway of brain machine interface

Hence invasive BMIs rely on the physiological property of individual cortical neurons to modulate their spiking activity in association with movements [3,53-56]. These modulations are found to be highly variable from neuron to neuron and from trial to trial. Yet averaging across many trials reveals fairly consistent firing patterns. Based on the hypothesis that the function of neural circuits is an emergent property that arises from the coordinated activity of large numbers of neurons, this phenomenon can be explained. Individual neurons generally form synaptic connections with thousands of other neurons. In distributed circuits, the larger the connectivity matrix the greater the redundancy within the network. Given their distributed connections and their plasticity, neurons are likely to be subject to continuous dynamic rearrangement, participating at different times in different active ensembles [2]. Accordingly both accuracy and

reliability of predictions of motor activity improve considerably with increasing the number of simultaneously recorded neurons and decreasing the errors due to individual neuron firing variability. Pursuing this motivation, the number of simultaneously recorded neurons has been approximately doubling every 7 years since 1950's [4]. Standard recording techniques using 704 implantable micro wire arrays have been reported in literature [5]. Recently Nicolelis Lab at Duke University announced their achievement to simultaneously record the electrical activity produced by a population of 1,874 interconnected single neurons at work in a primate.

Brain in a Dish:

At present, the prime methodology for studying neuronal extracellular activity under in vitro conditions is by using substrate-integrated microelectrode arrays (MEAs). This methodology permits simultaneous, long-term recordings (i.e. of up to several weeks) of extracellular field potentials. Correlating MEA recordings with microscopic imaging and stimulations is widely used to study the circuit-connectivity, dynamics and propagation effects in neuron assemblies. It is also used to investigate population coding, activity patterns, plasticity and pharmacological testing on either dissociated neuronal cultures or brain slices of embryonic rats, i.e. the young forests as Cajal described them. Commercially available MEA systems integrate typically 60–120 microelectrodes of 10–30 μm in diameter with pitches on the order of hundreds of micrometers. Typical neuron soma dimension in vertebrates is few micrometers long and the typical neuronal networks have 10000–50000 neurons, the limited number of electrodes and their rather large pitch

results in a substantial spatial undersampling of the overall network activity [6] as shown in Fig2.2.

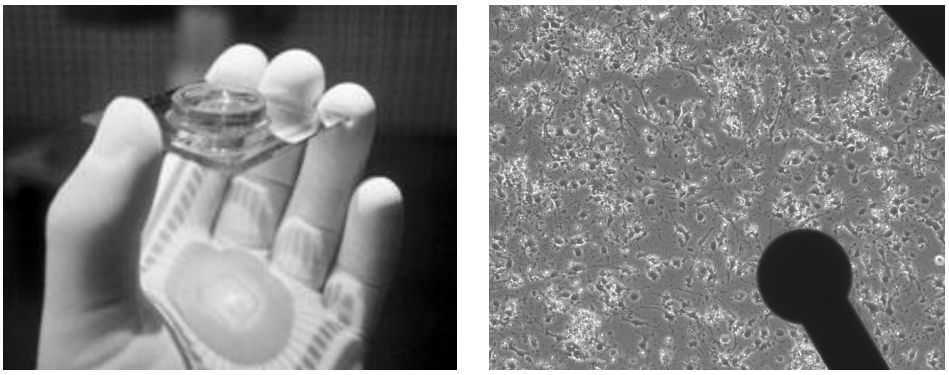


Fig. 2.2: Substrate-integrated MEA dish. The microscopic image of the electrode (black) and neurons. Neural Instrumentation Lab, Temple University [57]

The development of higher spatial and temporal resolution at low noise levels are prerequisites for opening the perspective to access the network electrical activity at the global and cell levels. Recently, CMOS-based high-density MEAs were developed featuring switching techniques to manage a large number of electrode channels interconnections, multiplexing, amplification, and filtering. Active Pixel Sensor based MEA platform providing 4096 microelectrodes at 21μm inter-electrode separation and 7.7KHz sampling rate has been documented [6].

Considering the ultimate goal of Brain Activity Map [2], the current neuroscience in vivo and in vitro research states and the advancement of high density microelectrode arrays, the migration to monitoring thousands of recording channels at high temporal resolution is achievable.

1.1 Increasing the number of Recording Channels:

More is Different - The behavior of large and complex aggregates of elementary particles, it turns out, is not to be understood in terms of a simple extrapolation of the properties of a few particles. Instead at each level of complexity entirely new properties appear.

Philip Warren Anderson

The augmentation of the number of recording channels carries different challenges to the neural signal processing system. The primary challenge is the massive increase in recorded data that needs proactive strategies for data transfer, reduction, management and analysis. The implementation of real-time signal processing becomes essential to alleviate huge data storage requirements. The access to a more detailed view of neuronal networks might reveal new properties and challenges pushing for the development of new analyzing tools.

With the continuous advancement of data acquisition systems featuring high-count recording channels, there exists a clear need for a test bed to develop and investigate a more suitable new generation of Neural Signal Processing (NSP) algorithms and computational tools. The platform has to offer programmable flexibility to allow the trial of different new strategies and novel computational techniques as well as rigorous testing for evaluation.

A plausible NSP platform that can handle thousands of recording channels has to provide means of high data transfer. As a numerical example, a NSP platform handling 2560 channels sampled at 31.25 KH at a sample precision of 16-bits must be capable of

managing an input data stream of 1.28Gbps. The data transfer interface has to be compatible with high-density neural data acquisition systems [7].

Data reduction based on the sparse nature of the neural signal with respect to time and the redundancy perceived across multiple electrode recordings becomes essential. Spike detection is the essential first step building block that allows the system to deliver only the action potential waveforms, their respective occurrence times and channel ID instead of the entire raw signal. The AP waveforms are then used by an autonomous spike sorter to first distinguish true spikes from false detections, then, to associate each spike to its generating neuron in case of multi-unit recordings. Depending on the performance and inter-electrode spacing, the AP waveforms might be necessary to identify redundancy over multiple recording channels.

The spike detection settings for each channel is independent from the settings of other channels, and hence spike detection over different sites can run in parallel. Applying parallel processing whenever possible limits the overall latency and assists in achieving real time implementation.

The NSP platform has to be fully autonomous and functional under expected Signal-to-Noise Ratios delivered by the data acquisition system. The system must be adaptive to varying noise levels over different channels and over time.

The main objective of the dissertation is to design an experimental test bed that can facilitate dealing with a large number of recorded neurons in real time. It also presents an architecture that performs spike-based data reduction.

1.2. Why Consider FPGA?

Ross Freeman (1944-1989) established the leading FPGA developer Xilinx in 1984 and invented a year later the first Field Programmable Gate Array (FPGA). **FPGAs** are programmable semiconductor devices that are based around a matrix of Configurable Logic Blocks (CLBs) connected through programmable interconnects. FPGAs can be configured to implement custom hardware applications and functionalities. Since their invention, FPGAs have evolved far beyond the basic capabilities present in their predecessors, and incorporate hard Application Specific Integrated Blocks of commonly used functionality such as RAM, clock management, and DSP.

FPGAs are parallel in nature, so different processing operations do not have to compete for the same resources. Each independent processing task is assigned to a dedicated section of the chip and can function autonomously without any influence from other logic blocks.

As integrated circuits grew smaller and maximum toggle rates increased the need for input/output bandwidth exploded. With more hardware resources and faster clock speeds, conventional I/O resources became the bottleneck to FPGA performance [52]. In 2002, Xilinx embedded high-speed serial Multi-gigabit transceivers (MGTs) on their FPGAs and introduced them commercially under the name Rocket I/O. MGTs are Serializers/Deserilizers (SERDES) that allow serial data transmission over differential pairs at speeds of up to 28.05Gbps per lane (see Fig. 1.3). Alternatively, multiple MGTs can be bonded together to form a higher bandwidth interface. Multiple MGTs are integrated above and below the Block RAM columns providing close availability for

ingress and egress FIFOs. Rocket IO serial transceivers are compliant with standard gigabit communication protocols.

FPGAs offer massive parallel processing performance, reconfigurable flexibility and superior capabilty of streaming data, and therefore present an appealing hardware implementation solution for a NSP testbed that can handle a large number of similarly strutured parallel channels in real time.

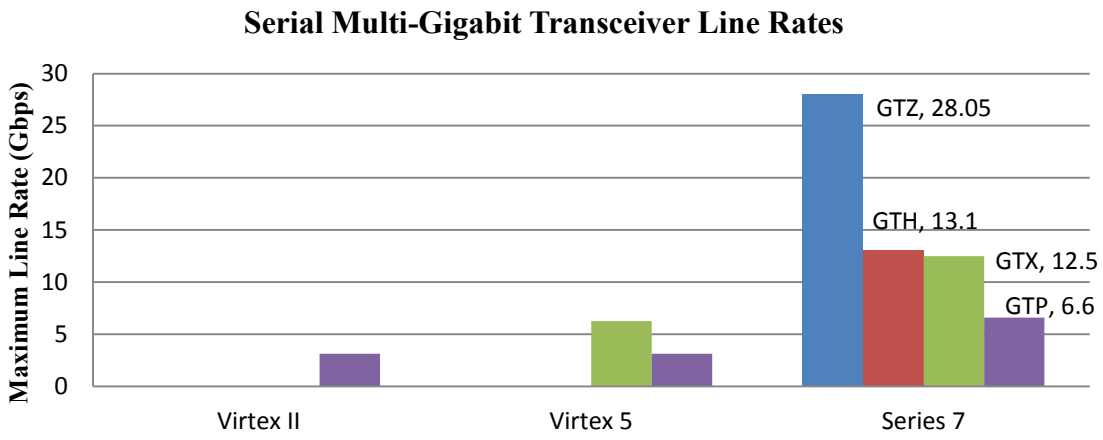


Fig. 1.3. Bar graph presenting the available serial Multi-Gigabit Transceiver line rates.

CHAPTER 2

REVIEW OF LITERATURE

Monitoring the interplay of neuronal ensembles in the brain is important for understanding mechanisms underlying memory, learning and behavior. Recently a group of neuroscientists have proposed launching a large-scale, international public effort called "Brain Activity Map" (BAM) Project, aimed at reconstructing the full record of neural activity across complete neural circuits [2]. They describe the neural circuit function as being emergent, meaning that it arises from complex interactions among millions of neurons and that the circuit state is not predictable from responses of individual sparsely sampled cells. They propose the dynamical mapping of the "functional connectome", the patterns and sequences of neuronal firing by all neurons. Correlating this firing activity with both the connectivity of the circuit and its functional or behavioral output could enable the understanding of neuronal codes and the regulation of behavior and mental states. Some of the mental illnesses that could not be understood using single-level analysis, such as autism and schizophrenia, may be possible to explain on an emergent level analysis. Clearly, the benefits of getting the full dynamical picture of the brain will be invaluable to address many questions in neuroscience, but to achieve this vision there is a clear need to develop novel technologies and significant innovations in systems engineering.

At present, population coding is studied either by monitoring the spiking activity of a few hundreds of individual neurons working with intact, living animals or by

studying the basics of distributed information processing using cultured neuronal networks. Cultured neuronal networks lack many features of real brain, but they retain others such as developing synaptic connections and exhibiting different patterns of electrical activity [8]. The neural activity cannot be correlated to a behavioral or mental output as *in vivo*, but it can be correlated to a structural connectome and to stimulation patterns. Advancement in micro-electrode array technology and multi-photon microscopy, has made it possible that every cell in a cultured monolayer network of dissociated neurons can be observed, monitored, stimulated and manipulated with temporal resolution in the sub-millisecond range, and spatial resolution in the submicron range, in a non-destructive manner [8]. Currently, such detailed analysis is not feasible in living animals, or even brain slices, but it remains an open question however, whether any of the processing done by cultured neurons is relevant to that carried out by intact brain.

This chapter serves to present efforts from a number of research groups to upgrade the recording capabilities of neuronal activity to higher spatial and temporal resolution across a large-scale neuronal ensemble to approach the model of *in vivo* brain. It will review some of these efforts reported on the data acquisition level. With the increasing number of recording sites, the chapter also discusses architecture design considerations at the spike detection level.

2.1. Multi-electrode Arrays:

Multielectrode arrays or microelectrode arrays are data acquisition devices that contain multiple plates or shanks through which neural signals are acquired, basically serving as neural interfaces that connect neurons to electronic circuitry. The signal then passes through amplification and filtering to remove some of the background noise. MEAs can be classified into two groups: implantable MEAs, used *in vivo*, and non-implantable MEAs, used *in vitro*. Using advances in multisite microelectrode array fabrication techniques varying shape and recording capacity of the electrodes, it is possible to record the activity of tens to hundreds of neurons in parallel [9]. Integrated microelectronic circuits were applied to enable the transition to even higher recording capacities [10]. Development of *in vivo* and *in vitro* multi-electrode probes share many of the same hardware and data analysis problems and mutually contribute to the advancement of the state of the art.

2.1.1 *In Vitro* Micro-Electrode Arrays:

Multi-electrode array culture dishes allow simultaneous recoding from and stimulation of neurons. These wired Petri dishes are also called planar electrode arrays [2]. Early microelectrode developments by Gross [11], Wise, Meister and others paved the way for enabling chronic multi-single-cell recording. They were able to record neural spike potentials with good fidelity from a few tens of neurons.

MEA's have become commercially available just within the last decade. MEA systems capable of recording at least 60 electrodes are produced by MultiChannel Systems of Germany, and Panasonic of Japan. Guenter Gross supplies MEAs that can be

used with multi-electrode processing hardware and software made by Plexon Inc [8]. MEAs typically consist of less than 100 planar metal electrodes on an insulating glass substrate with a diameter > 30µm and a pitch >100µm. For commercially available MEAs, amplification and filtering are realized by discrete off-chip components [6].

Considering the dimensions of neurons, which range from below 10µm for vertebrates up to 100µm for invertebrates, the development of high-density arrays was needed to acquire more details from cell-based biological experiments on brain slices and to elucidate the contribution of individual cells to collective network. An advanced multi-electrode array system has been developed to study how the retina processes and encodes visual images. This system can simultaneously record the extracellular electrical activity from hundreds of retinal output neurons and consists of 512 planar microelectrodes with a sensitive area of 1.7 mm² and a noise level of a few µV [13]. However, some brain structures, such as hippocampus or cerebral cortex, extend over distances of many millimeters [14]. To record from these larger structures, an increased density of electrodes and a larger array would be required in order to fully analyze all the neurons of interest.

CMOS-based devices presents several advantages for managing a large number of electrode channels' interconnections, multiplexing, amplification and filtering. They have been initially implemented for *in vivo* neural probe recordings [15]. Later they have been used for in vitro devices at a larger scale to overcome the connectivity limitation by making use of on-chip signal multiplexing [12]. A number of voltage recording microelectrode array devices have been developed with significantly higher electrode

densities and larger areas. Due to hardware bandwidth limitations, these devices all make some compromise between speed, electrode count, multiplexed sampling, and noise [14].

A high-density 128x128 biosensor array CMOS chip was designed featuring a frame rate of 2K frames per second, and a pitch of 7.8μmx7.8μm over 1mm² extent [12]. The device has a very high spatial resolution recording of small areas of tissue, but was reported to have noise levels in the range of 250μVrms, which could make recording smaller extracellular spike signals (20-100μV) a challenge [14]. The simultaneous recording from all electrodes required the front-end amplifiers being placed in each recording site, which, due to area constraints, entailed the high noise levels.

A switch-matrix-based high-density microelectrode array [16] was developed as a hybrid between low electrode count and high resolution arrays. The device has only 126 output channels but these could be digitally selected from among 11,000 electrodes, separated by a pitch of 18μm, using a reconfigurable electrode/readout-channel routing. The device has very low noise levels of 7-9μV, since the front-end circuitry were placed outside the array, where sufficient area for low-noise circuit implementation is available.

Imfeld and coworkers developed an electrode multiplexing , 4096 pixel recording array with a 42 μm pitch and a 2.7mmx2.7mm extent that can record the full frame at a rate of 8KHz. The device has high spatial resolution, a relatively good temporal resolution and a wide extent of ~7mm². The data recording has a hardware implementation inspired by image/video processing concepts. It implements an Active Pixel Sensor (APS) concept CMOS design, acquiring the data as a time sequence of images [17]. Basic amplification was performed underneath each electrode, and a

tradeoff between spatial resolution and noise dictated the inter-electrode spacing. The noise level is in the range of ~26μV rms. The complete architecture of the acquisition system is shown in Fig. 2.1. Control and timing of the APS-MEA as well as the bank of the Analog to Digital Converters (ADC) is performed by an FPGA. Filtering the 4096 channels in real time is also carried out on the same FPGA.

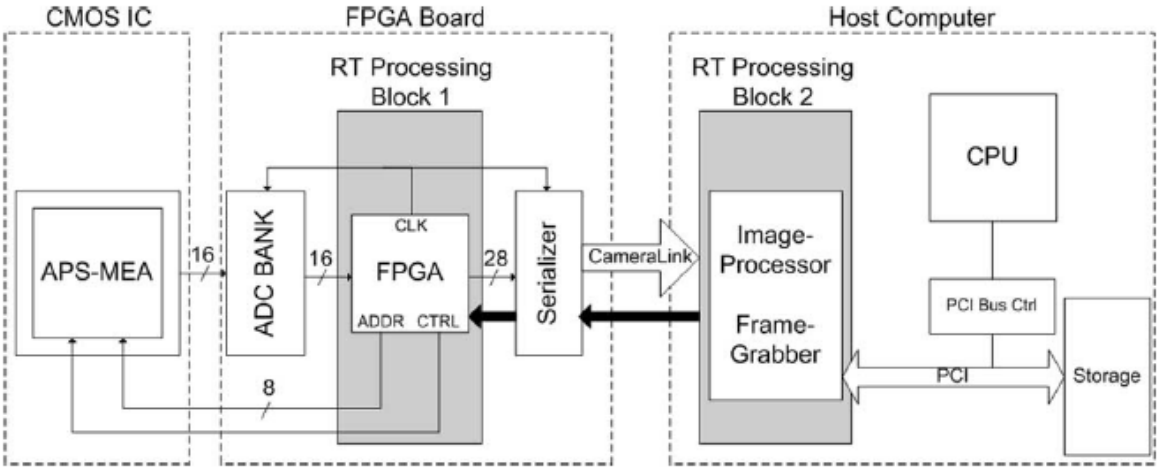


Fig 2.1 Block diagram of the acquisition platform. [17]

Recently a high-electrode count Pico-current Imaging Array (PIA), based on an 81,920 pixel readout integrated circuit camera chip was developed. While originally designed for interfacing to infrared photo-detector arrays, it was adapted for neuron recording by bonding it to microwire glass. The full frame of an area of 9.6mm by 7.7mm can be recorded at 100Hz. [14]

2.1.2 *In Vivo* Micro-Electrode Arrays:

Implantable MEA research considers more requirements and restrictions for acute and chronic implantation. Some research areas focus on the fabrication process, insertion techniques, chronic response of tissue on the implant, wireless implant design and power issues. In this section the main focus will be only on presenting a few of the research efforts on increasing the number of recording sites of neural signals. Some Labs are mainly interested in monitoring more neurons in different cortical areas of the brain [18], while others are interested in changing the microstructure of the neural probes to increase the spatial resolution [19-21].

Researchers at the Duke University lab published a paradigm for recording the activity of single cortical neurons from awake, behaving monkeys [5]. They implanted high-density microwire arrays, developed at Duke University, totaling up to 704 microwires per subject in five cortical areas. Early this year the lab announced that they were able to simultaneously record the firing patterns of close to 2000 neurons. Four multielectrode arrays with 448 electrodes were inserted in rhesus monkey motor and sensory cortices of both hemispheres. There are no publications yet explaining the detailed instrumentation used.

The microwire and similarly structured silicon-based arrays feature one recording site per wire, which limits the capability of the array to capture dense neuronal activity in 3-dimensional setting. Alternatively in 1985 the planar microelectrode array was introduced, using multiple electrodes arranged on implantable silicon shafts [20]. The planar microelectrodes increased the recording spatial precision. It was later modified by

proposing double-sided electrodes [22]. These devices contain electrodes on two parallel planes separated by the thickness of the implantable shaft, presenting a building block for a 3-dimensional recording geometry.

Du and coworkers at the California Institute of Technology have fabricated a dual-side electrode array by patterning recording sites at the front and back of an implantable microstructure. They proposed stacking several two-dimensional multishank arrays into three dimensional probe arrays, to access 3-D neuronal structures as shown in Fig. 2.2.

The nano-probe design presents a potential for hundreds or thousands of recording sites, but it holds a high risk of brain tissue damage. To minimize the disruptive interface between the silicon electrodes and the brain, the nano-probes will pass through more testing and evaluation to determine the optimal shaft size and shaft spacing.

It is evident that there are several efforts aiming to increase the number of recording channels *in vivo* as well as *in vitro* and *in situ*, which leads us to the next section of presenting the available signal processing tools and their capability of handling the resulting high amount of recorded data.

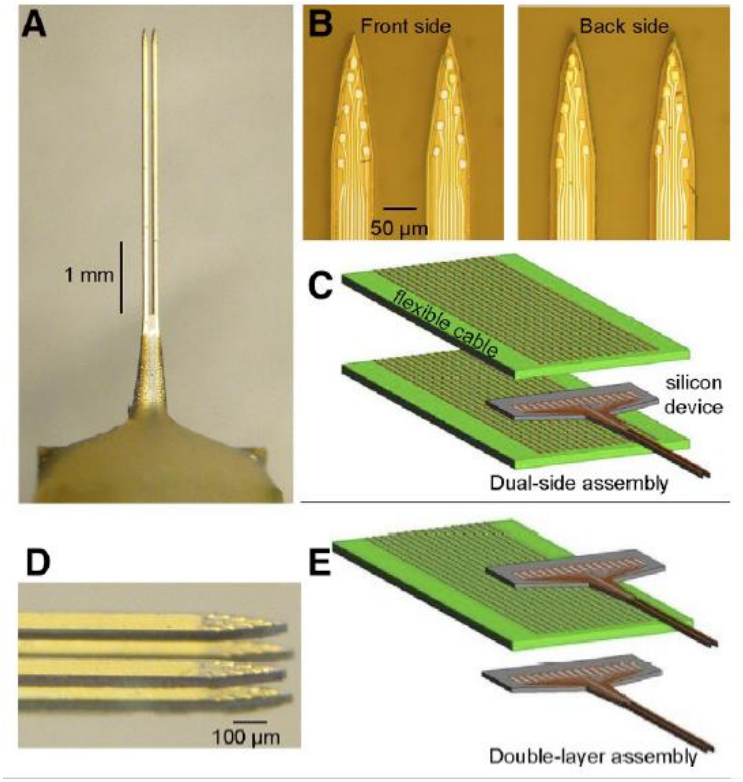


Fig. 2.2. Dual-side and double-layer microelectrode arrays were built on thin silicon shafts. *A*: front view of the device. The shaft dimensions are 4 mm x70 µm x50 µm (*l* x *w* x *t*). *B*: expanded view of the front and back sides of the dual-side array. The recording sites have a geometric area of 100µm². *C*: layers involved in connecting dual-side arrays to flexible printed circuit boards (PCBs, green), one board for each side. Electrical connections were made via low-profile flip-chip bonds. *D*: view of the tip of a 2 x 2 shaft, double-layer array. *E*: a modular assembly scheme used to make the multilayer structure. Note that the PCB contained conducting leads on both sides and thus the same board connected to the upper recording sites on the bottom layer and the lower sites on the top layer. [21]

2.2. Neural Signal Processing Systems:

Recordings of extracellular neural activity are used in many research studies and clinical applications. Usually, these signals are analyzed as a point process, and spike detection is used to estimate the times at which action potentials from one or more neurons occurred. Recordings from high-density MEAs and low-impedance microelectrodes often have a low signal-to-noise ratio (SNR < 10) and contain action potentials from more than one neuron. Hence, spike detection is often followed by spike sorting, that involves clustering, to assign each event to separate neurons based on AP waveforms.

2.2.1 Spike Detection Algorithms:

The main challenge in detecting spikes is the interference due to background noise. Various spike detection algorithms with different levels of complexity and performance have been presented [23,48]. The absolute threshold method is widely used as it requires the least computations, but it is highly sensitive to background noise. Various techniques have been proposed for autonomously selecting the threshold based on the statistical characteristics of the recorded signal, while others set the threshold based on a visual inspection of the detected spikes. A different type of algorithms is based on template matching. These algorithms scan the recorded signal for instances, where segments of the signal are similar to templates of spike waveforms. In this case a priori knowledge of the spike waveforms is required and the user should supply a threshold for similarity measures. A different approach suggests using a preprocessors, such as the Nonlinear Energy Operator NEO to give emphasis to the spikes relative to the noise before applying the absolute threshold, consequently improving the spike detection performance.

2.2.2 Overview on Existing Neural Signal Processing Systems:

Existing commercial recording systems are limited to a few hundred channels and rely on multiple sequential logic processors connected in parallel. While functional, such systems are difficult to manage, and do not scale well to larger channel counts. The paradigm described by researchers at Duke University [5] for acquiring neural signals from monkeys incorporated the multichannel acquisition processor MAP by Plexon. The MAP recorded all the events that crossed the voltage threshold, set by the user, for subsequent offline spike sorting analysis. Each MAP processor can handle up to 128 channels. For their experiments, they used a custom made MAP cluster, formed by three 128-channel MAPs connected in parallel and synchronized by a common 2MHz clock signal. The initial step in all recording sessions required the experimenter to manually set the voltage threshold for each of the MAP channels connected to an implanted microwire [5]. The threshold was set based on visual inspection of the original analog signals displayed in an oscilloscope as well as the digital signal displayed on the screen of the computer controlling the MAP. With the increasing number of recording channels, it becomes impractical to require the user to tune the spike detection algorithm to the signal properties visualized on each channel. Currently, Plexon is offering an upgraded version of the MAP called OmniPlex® D Neural Data Acquisition System. The system can handle up to 256 channels sampled at 40KHz with a sample precision of 16 bits.

With the rising demand to process a large number of similarly structured parallel channels in real time, there has been an emerging interest in hardware implementation over sequential processors. FPGAs offer massive parallel processing performance and

reconfigurable flexibility, which makes them an attractive alternative for real-time signal processing.

The data acquisition systems integrated with the high-density MEAs presented in section 2.1. perform signal conditioning in terms of amplification and filtering, and then send the complete signal to a host PC for storage, off-line spike detection and clustering. [17]. As high-density MEA platform produce data streams in the range of hundreds or thousands of Megabits, the amount of data storage required increases drastically with longer recording times. Real-time spike detection and data compression become vital to limit the amount of data storage.

2.3. Spike-Based Data Reduction:

The idea of data reduction has been addressed mainly in wireless implantable devices for Brain-Machine-Interfaces. Several efforts have been proposed to implement on-line hardware spike detection and send only the spike waveforms while disregarding the interspike samples. The spike waveforms are the only information needed for successive spike sorting. With a limited telemetry bandwidth, it was essential to consider spike-based data compression algorithms to reduce the amount of sent data. With power restrictions of implantable devices, there was also a need to avoid high power consumption associated with the continuous transmission of raw data. The proposed schemes aimed at providing an efficient use of the available transmission bandwidth and an increase of the device throughput. Based on the sparse nature of the neural signal with respect to time, and the average neuron firing rates, the amount of sent data can be reduced to approximately ~2.25% of the total amount of raw data [24].

With a focus on telemetry transmission, Bossetti et al [24] raised an important design consideration for spike-based data reduction in real-time. They demonstrated that although the spike-based compression might be very appealing from the point of view of average bandwidth, it is subject to telemetry bottlenecks during periods of multichannel neuron bursting causing queuing-based transmission delays at the output buffer. They drew the attention to the relation between the ratio of the output to average input bandwidth and transmission latency, the number of samples per spike waveform, the mean firing rate MFR, and the needed queue depth of the output buffer memory. Bottlenecks and latencies are mainly a consequence of accumulating the input data samples over short periods of time before their transmission at the output, waiting for the AP waveform to complete at the output queue. The research paper has concentrated mainly on the transmission delay. The hardware implementation delay is the time between the arrival of the spike waveform at the input buffer and its appearance on the output buffer. The method of spike detection employed will dictate the size and temporal pattern of spike data arriving at the output buffer. These patterns could impact the timing significantly. Aside from the delay depending on the scheme control and data handling between the input and output buffers, there are other delays related to the computational overhead and memory read/write times, that depend on the system clock. The performance of the spike detector will also affect the required output bandwidth. A high false detection rate will increase the overall MFR and change the system design.

2.4. Spike Detector Design Schemes:

The design of the data flow in the spike detection hardware-implementation defines the system latency and memory requirements. With the increasing demand to monitor thousands of recording channels, the efficient use of hardware resources, especially memory blocks on the FPGA becomes vital. Only a few literature have presented detailed patterns and sequences of the data flow on their spike-based compression architectures. This section presents two examples of spike detection architectures with different data flow sequences, and discusses their possible application on high channel-counts. The first example is a spike detection scheme designed for an implantable data acquisition system for BMI application [26]. The second example is an architecture of a Neural Spike Detection platform NSP [25].

2.4.1. Spike Detection Architecture for Implantable Application: [26]

A spike-detection based data reduction scheme described in literature [26] handles the time division multiplexed data recorded from 16 channels. In this design the 64 most recent samples from each channel are stored in the input data storage buffer memory. Once a spike has been detected on a channel, the hardware waits until an additional set of 34 samples, representing the spike waveform refractory period, from the same channel has been acquired. After the 45 samples of the AP waveform are completed in the input buffer memory, the spike waveform waits for its turn in a queue for detected spikes to be written out to the FIFO buffer, where it is held until the embedded PC and wireless card transmit them to the host station.

The scheme worked fine with 16 channels, but if the design is used for higher channel counts then some modifications must be considered. For example, if the spike detection unit handles many channels in a time division multiplexed approach, then the system must extend the memory space assigned for each channel to ensure that the detected spike waveform samples are copied from the input buffer to the output FIFO before they are overwritten by new samples. Another solution would be increasing the clock ratio between the reading and writing clock rates of the input buffer. As the copying process from the input buffers is queuing based, the more channels sharing the same queue, the slower would be the route. To avoid high memory usage on the hardware, when increasing the number of recording channels handled, a different design sequence might be considered. For example, copying the AP waveform in single samples, as they arrive at the input buffer, or in small groups of samples to the output buffer, in order to decrease the memory space assigned for each channel in the input buffer.

Another implementation approach might be to replicate the 16-channel spike detection unit and use an intermediate FIFO for each unit to store the spikes before sending them to the common output FIFO. This latter transmission may be controlled by a queuing-based scheme. In this case the queuing based delay must be monitored closely as the AP waveforms will passes through queuing-based transmission three times. Once to be copied from the input buffer to the intermediate unit FIFO, then from the intermediate FIFO to an output FIFO common to all spike detection units and finally queued in the output FIFO for transmission to the host. The delay is expected to increase in case of neuron bursting across the channels.

2.4.2. Spike Detection Architecture on NSP platform:

A Neural Signal Processing (NSP) platform was designed by [26]. The platform incorporates a spike detection and a spike sorting p-cores controlled by two Microblaze processors. The central processors were connected to the firmware layer via the LMBs. Communications between the processors and other subsequent layers were channelized through the PLBs, where the processors were connected as masters while all other peripherals and p-cores connected as slaves.

Focusing on the scope of the dissertation research, only the spike detection p-core was investigated. The spike detection p-core handled the spike detection process while the central processor managed only the transfer of input data to the p-core as well as monitoring the spike detection process.

(a) MicroBlaze Interrrupt Latency: The interrupt latency occupies a significant share of the processor cycles limiting the maximum operational frequency of the p-core. That s why the p-core was set to work at 10MHz, a ten times lower speed than its maximum possible operating frequency defined by the routing critical path. Assuming that the neural signal data is pipelined through the spike detector, and that the sampling frequecy is 31.25 KHz, the maximum number of channels that can be handled by the platform is limited to ~300 channels [26].

The hardware implementation advantages were restrained by the dependency on the MicroBlaze processor to control the operation sequence. If the p-core was to be implemented as standalone module, it can operate at around its maximum operating frequency, defined by the critical routing path. The alternative design solution presented

in the dissertation features a standalone implementation of a spike detector using Finite State Machines (FSMs) to control the interface between the data acquisition and the spike detection core as well as the interface between the spike detector and the output. The use of a processor with a higher clock rate may be another alternative to reduce the interrupt latencies.

(b) Input Data format: The data processing architecture was based on receiving the neural data as a stream of frames of 32 successive samples recorded from one channel and preceded by their channel ID. Simultaneous MEA data acquisition systems incorporate a Time Division Multiplexer (TDM). The rearrangement of the data in the flow scheme required adds control and storage burdens as well as data skewing to the interface between the data acquisition and the platform. As the system is required to extract valid spike waveforms, the platform has to deal with action potentials split between two data frames.

(c) The Threshold Comparator and Threshold Selection: The threshold comparator compares the neural data from the preprocessor, based on the nonlinear energy operator, to a user-defined threshold to detect spikes. The threshold was a fixed value for all the channels. The signals recorded by different electrodes may vary markedly in their SNR, and on the same channel SNR may fluctuate over time. With different SNRs the threshold has to be set adequetly for each channel. Dealing with massive number of recording channels threshold selection has to run autonomously without user interference, as manual channel settings become impractical.

2.5. Data Acquisition High Speed Serial Interface:

The typical neural signal processing pathway starts with a data acquisition system that records extracellular potentials from an MEA. The data acquisition provides amplification, filtering, time division multiplexing and A/D conversion of data read from the different electrodes. This thesis focuses on the spike-based data reduction module and is thus concerned with the interface between the ADC of the data acquisition system.

As the spike-detection based data reduction systems was designed to handle thousands of recording channels, it has to offer enough bandwidth to receive the massive amount of neural data from the data acquisition system in real time. Multi-Gigabit transceivers integrated on the FPGA were the primary choice for providing the needed high transmission rates in the range of a few Gigabits per second. Satisfying this requirement on the FPGA side, it was important to investigate the interfacing options to real data acquisition systems, and whether the high data transmission rates can be achieved by A/D converters.

Low voltage differential signaling (LVDS) is the traditional method of interfacing data converters with FPGAs. LVDS was introduced in 1994 with the objective of providing higher bandwidth and lower power dissipation than the existing differential transmission standards. The rapid increase in the resolution and speed of converters created several system design challenges related to connecting ADCs with conventional parallel CMOS/LVDS outputs to FPGAs or DSPs. The need for an extensive number of high bandwidth PCB interconnects increased the PCB complexity. The large number of traces with the restriction of being of the same length raised the routing difficulty. In

some applications, the data converter interface was the limiting factor in achieving the desired system performance in bandwidth demanding applications.

2.5.1 An Overview on JESD204 Data Converter Serial Interface Standard:

The JESD204 data converter serial interface standard was introduced in 2006 by the JEDEC Solid State Technology Association with the aspiration to avoid the limitations of LVDS connectivity and to provide a higher speed serial interface for data converters. The standard aimed at increasing the bandwidth and reducing the number of digital inputs and outputs between high speed data converters and processing devices. It applies 8b/10b encoding that eliminates the need for a frame clock and a data clock, enabling single line pair communication at speeds up to 3.125 Gbps [29]. JESD204 allowed the connection of the converters to the SerDes ports offered by many FPGAs as shown in Fig 2.3 [29].

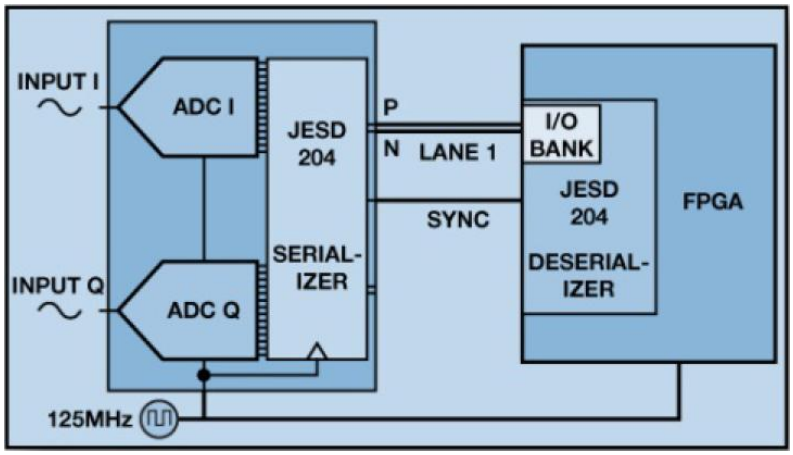


Fig. 2.3 A block diagram of the serial data link between one converter or more and the FPGA receiver, as defined by the original JESD204. It consists of a single lane with a data rate defined between 312.5 Mbps and 3.125 Gbps. The lane is a physical differential pair of connectors. [29]

The reduced number of connecting traces reduced the routing complexity. It offered reconfigurable resolution without hardware modification. The JESD204 standard went through two revisions. JESD204A [30] was released in 2008 and added support for multiple time-aligned data lanes and lane synchronization. This modification provided synchronization support of multiple devices. JESD204B, published in 2011, increased the maximum lane rate to 12.5 Gbps. It also added deterministic latency, which is achieved by communicating synchronization status between the receiver and the transmitter using a 'Sync' signal. Harmonic clocking was also introduced by JESD204B, making it possible to obtain a high speed data converter clock from a lower speed input clock with defined phasing. [31]

2.5.2 Implementation of JESD204B for high density MEA data acquisition:

Increasing the number of recording sites of neuronal signals is expected to involve wide bandwidth multichannel converters that are sensitive to deterministic latency across all lanes and channels. Hence the JESD204B might be the protocol of choice for converters used in future neuronal data acquisition systems.

Some of the recently developed ADCs have integrated SerDes and are compliant with the JESD204 standards. They can be connected to FPGAs using a high speed serial differential pair lane. As an example, AD9644 by Analog Devices® offers high sampling speeds of 80MSps or 155MSps which are in the same range as the operating frequency suggested in the dissertation presented designs.

2.6. Overview on Data Transmission from FPGA to a Host PC using PCIe:

PCI Express® (Peripheral Component Interconnect Express), abbreviated as PCIe®, is a high-speed, general-purpose interconnect architecture, designed for a wide range of computing and communicating platforms. It is a packet-based, point-to-point serial interface. The PCIe® protocol is divided into three layers: the Transaction Layer, the Data Link Layer, and the Physical Layer. These layers interact with the Configuration Space. Xilinx® provides scalable integrated PCIe Endpoint blocks on their FPGAs. Connections and control of the physical interfaces of the integrated Endpoint block are contained within the Endpoint Block Plus wrapper for PCI Express, available from the Xilinx® CORE Generator GUI. For more information on the Xilinx Endpoint PCI express solutions and guidance on how to estimate the performance of PCI express systems, the reader is directed to references [32-34]

Xilinx PCIe IP core was connected to the application design via a Xillybus IP core [35], and standard FIFOs. Xillybus provides end-to-end stream pipes solutions for application data transport. It conducts the data traffic between FPGA and host, by supplying a Direct Memory Access (DMA) hardware design along with a kernel mode driver for both Windows and Linux. The host driver generates device files which behave like named pipes. The application on the computer can read the data sent using file descriptors. A file descriptor is an index for an entry in a kernel-resident array data structure.

At driver load, DMA buffers are allocated in the host's memory space, and their addresses are saved on the FPGA. The number of DMA buffers and their size are

hardcoded parameters in the FPGA IP core for a given configuration. They are retrieved by the host during the detection process. A handshake protocol between the FPGA and the host ensures efficient utilization of the DMA buffers, while maintaining responsiveness for short segments of data.

2.6.1 Xillybus IP core Connection Overview:

A Xillybus stream can be configured to behave synchronously or asynchronously. An asynchronous stream fills the host's DMA buffers whenever possible, i.e. when the file is open, data is available and there is free space in the DMA buffers. In a synchronous stream setting, the IP core logic will not fetch data from the user application logic on FPGA unless the application on the host issues a request to read the data from the file descriptor. Asynchronous streams are preferred for high-bandwidth applications, as they allow a "background" flow of data while the application on the host is preempted or busy with other tasks. Xillybus can conduct bidirectional data traffic between FPGA and host, but for integration with the spike-based data reduction platform, the focus is on the upstream direction. Fig. 2.4 shows a simplified diagram describing the data and control signals, which establish the link between the FPGA design and the host PC.

The following is a description of the Xillybus IP core signals for FPGA-to-host transmission:

(a) **user_r_devicefile_data:** The width of the data input signal can be set to be 8, 16 or 32 bits during the device configuration.

(b) **user_r_devicefile_rden:** This core output signal is a read enable signal to the FIFO.

(c) **user_r_devicefile_empty:** When this input signal is asserted, it temporarily assures that no read cycles occur as long as the FIFO has no data.

(d) **user_r_devicefile_eof:** The end-of-file (eof) input signal triggers the core to generate an eof event, indicating that all data has been transmitted. Once asserted, the core will not issue any more read cycles. The application reading from the file descriptor on the host receives a notification that 'the file has reached eof'.

(e) **user_r_devicefile_open:** This core output signal is asserted high, when the respective device file in the host is open for read. This signal was used to reset the FIFO.

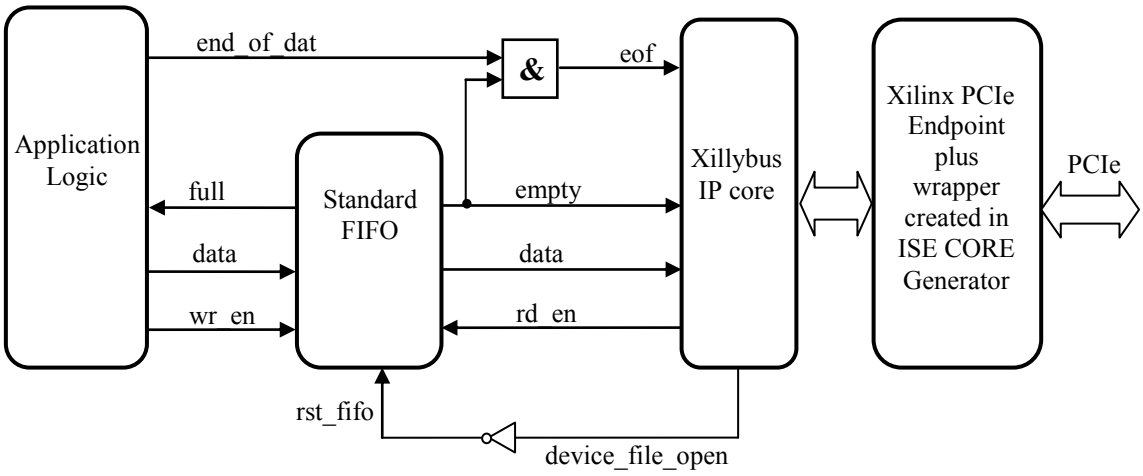


Fig. 2.4 Simplified block diagram describing a data stream flow from FPGA to host. [35]

CHAPTER 3

PLATFORM DESIGN

3.1 System Overview:

The Neural Spike Detection platform receives time division multiplexed serial samples from a high number of neural recording channels at the multi gigabit receiver port of the FPGA. The receiver performs deserialization of the data and ensures correct sample-word alignment. The system affiliates each sample to its source channel and performs spike detection. If a spike is detected the spike waveform along with its time stamp and channel ID are passed to an output buffer for further spike sorting or data analysis. Fig. 3.1 presents the integration of the spike detection platform in a typical neural signal processing system.

The typical neural signal processing pathway starts with a data acquisition system that records extracellular potentials from an MEA. The data acquisition provides amplification, filtering, time division multiplexing and A/D conversion of data read from the different electrodes. Then the signal passes through spike detection followed by spike sorting, spike binning and analysis. The dissertation work focuses on the spike-based data reduction module and is thus concerned with the interface between the ADC of the data acquisition system and the interface with the spike sorting on FPGA or sending the data to a host PC for further analysis.

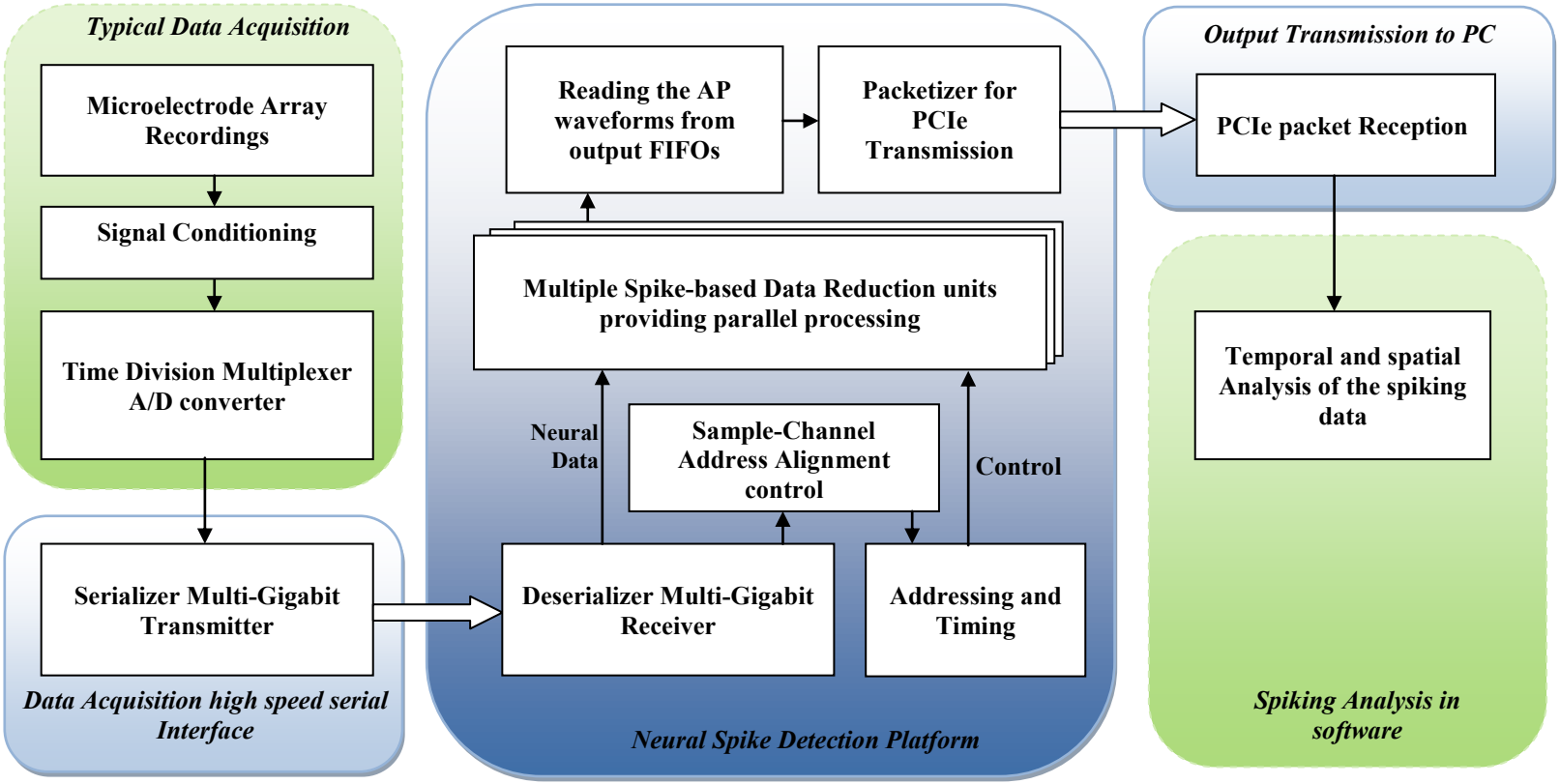


Fig 3.1: A block diagram of the Neural Spike Detection platform and its integration in a Neural Signal Processing system. The center block (dark blue) presents the Neural Spike Detection (NSD) platform performing spike-based data reduction. The blocks (light blue) connected to the NSD platform on the left and right sides present the interface required to embed the platform into a NSP system. The upper left and bottom right (green) building blocks present typical neural data acquisition and spiking analysis on a host PC, respectively. These are not part of the dissertation work.

The detection platform performs spike-based data reduction where:

$$\text{Average Reduction Ratio} = \frac{\text{MFR/electrode} \cdot \text{Number of samples per AP waveform}}{\text{Neural Signal Sampling frequency}} \quad (3.1)$$

where MFR = Mean Firing Rate. For example, for a MFR of 18 spikes/s/electrode, 50 samples per AP waveform, and a sampling frequency of 40 KHz the reduction ratio = 0.025.

As the system is designed to handle thousands of recording channels, it has to offer enough bandwidth to receive the massive amount of neural data from the data acquisition system in real time. For example for a 2560 channels sampled at 31.25 KSps, and a precision of 16-bits per sample, the data rate has to be 1.28 Gbps. Consequently, the platform architecture integrates the application of high-speed serial transceivers to allow for the required input data transmissions.

Although, the amount of data is significantly reduced, the system needs to integrate a high-speed communication link to transfer the AP waveforms to the host PC, accounting for transmission bottlenecks during periods of multi-channel neuron bursting [24]. A PCI express link is integrated to minimize queuing-based transmission latencies and performance degradation when the output data overwhelms the transmission bandwidth of the device.

3.2 Spike-based Data Reduction Unit:

The main building block of the design architecture is a spike-based data reduction unit that handles 128 channels. This unit can be replicated to process a higher number of recording sites. A block diagram of the spike detection module is shown in Fig.3.2. The spike detection unit receives time division multiplexed 16-bit sample data from 128 channels; it tests the samples for possible spikes, and then sends the complete Action Potential (AP) waveform of a detected spike preceded by the time stamp and the channel ID to the output buffer memory. This section presents the main building blocks of the unit and indicates how the design parameters were selected based on the spike detection algorithm applied on the platform. The main building blocks of the spike detection units are:

3.2.1 The Spike Detector

3.2.2 The output Buffer

3.2.3 The Input BRAM

3.2.4 The Channel Status Module

3.2.5 The BRAM Read Control

3.2.6 The Operation Management FSM

3.2.7 The Autonomous Threshold Selection

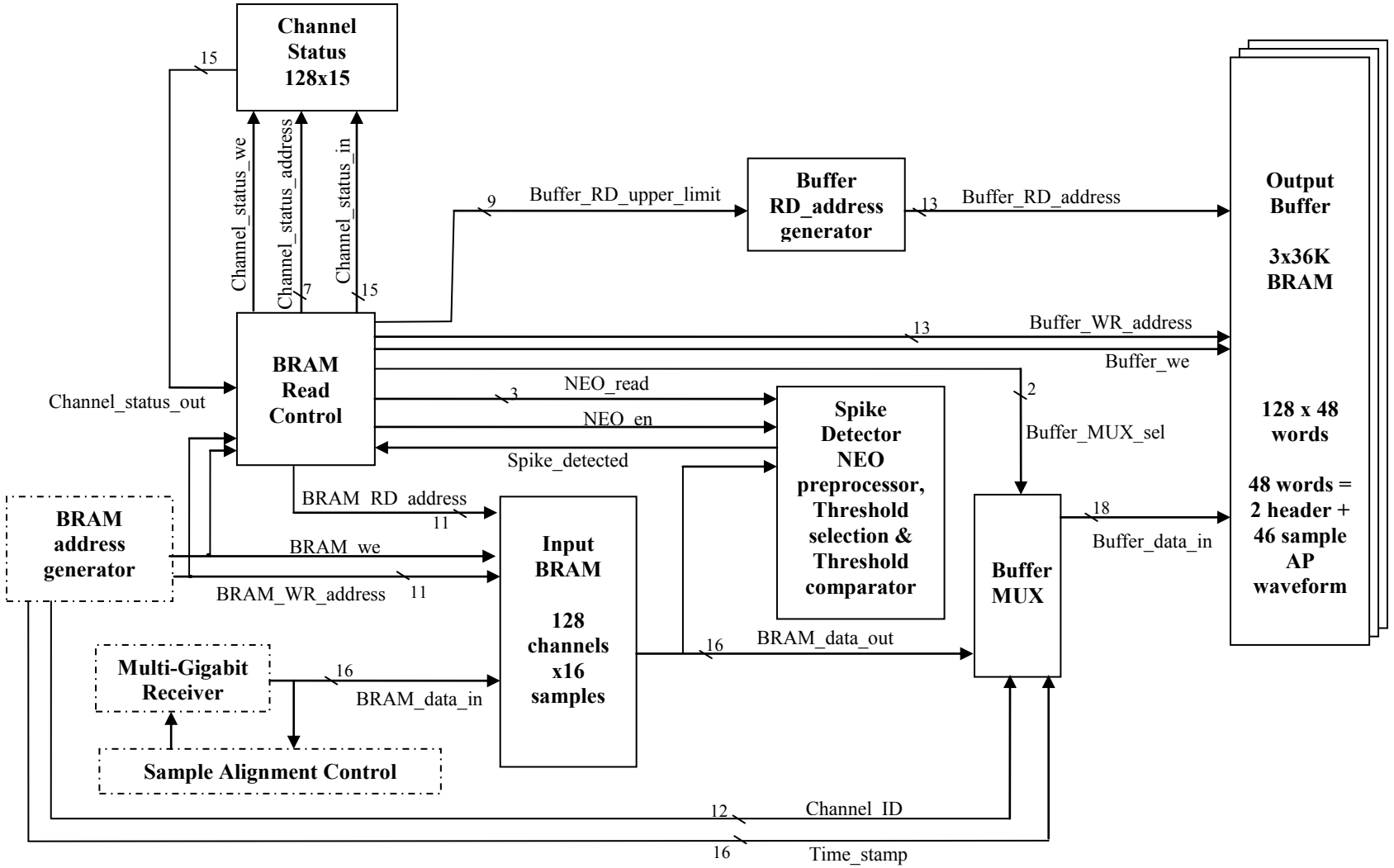


Fig 3.2: A block diagram describing the spike detection process. The spike detection unit is designed to detect neural spikes over 128 neural signal recording channels.

3.2.1 The Spike Detector:

The Spike detector block holds the hardware implementation of the spike detection algorithm. Various spike detection algorithms with different levels of complexity and performance have been presented in literature [2, 3] and can be applied on the designed platform with proper modifications of the system design parameters. As an example, the design model applies spike detection based on the absolute threshold after passing the signal through a Nonlinear Energy Operator (NEO) preprocessor eq.3.2 in order to give emphasis to the spikes relative to the noise and consequently, improve the spike detection performance.

$$\text{NEO}\ [n] = x^2[n] - x[n-\delta]\ x[n+\delta] \quad \text{where} \quad 1 \le \delta \le 4 \tag{3.2}$$

where x[n] is the neural data sample at any instance n .

The threshold for a given channel is set to a multiple of an estimate of the noise level on that channel. The detailed Threshold selection method and block diagram is presented in section 3.2.7.

3.2.2 The Output Buffer

A neural AP has duration of ~ 1.5ms on average. Considering sampling rates in the range of 30 KHz and based on the wave-shape, a full AP waveform was assumed to have 10 pre-spike samples, 1 spike sample and 35 samples representing the spike refractory period. This assumption was optimum for organizing the FIFO memory and address assignment. The output FIFO memory 3x36K can hold up to 128 spike waveforms at a time, counting

for the worst case scenario if firing neurons are detected on all channels at the same time. When the unit receives a sample from one of the channels it is written in the input memory.

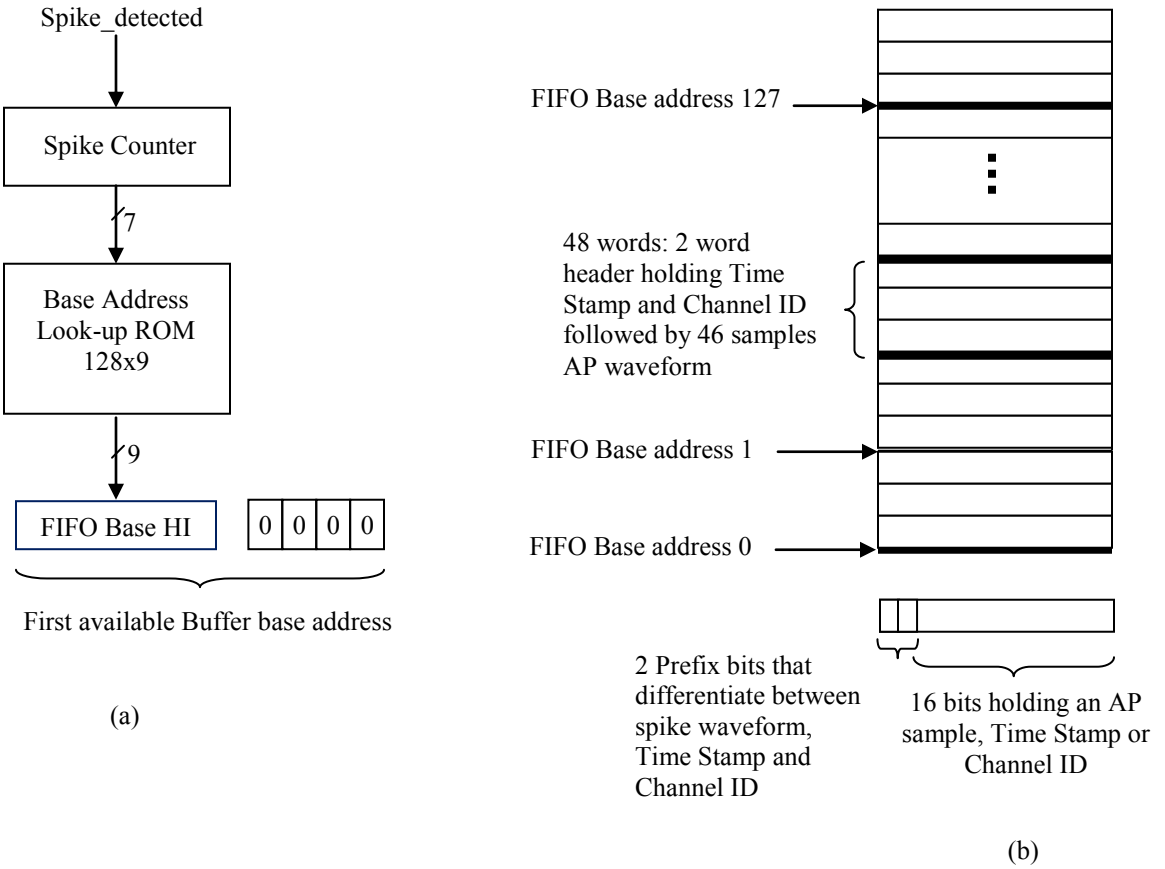


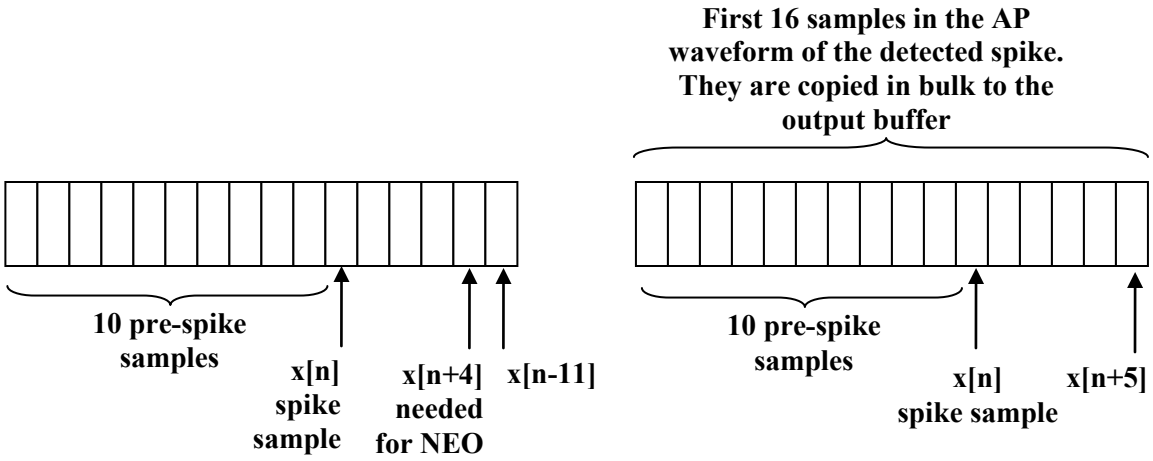
Fig. 3.3 (a) Spike counter and Base address look-up ROM used to determine the first available memory space in the output buffer to store a detected spike AP.
(b) Organization of the output buffer.

3.2.3 The Input BRAM:

For spike detection consecutive samples are needed to identify a spike. Each channel is assigned a memory space on the input BRAM to hold the most recent 16 samples. The depth of the memory space assigned to each channel was chosen to hold enough sample history to acquire the ten pre-spike samples, the spike sample x[n] and five post-spike samples. Four of the post-spike samples are the "future" samples held to reach x[n+4] needed for the NEO computation, and x[n+5] is added for timing control, as would be explained in the operation management section. The design does not copy the AP waveform as a bulk to the output buffer, instead it copies the first 16 samples, and then sends the refractory period sample by sample as they arrive at the input BRAM. This scheme has minimized the memory space depth needed for each channel, saving on total memory usage. An example of the arrangement of samples in the input BRAM space assigned to one channel is shown in Fig. 3.4.

3.2.4 Channel Status:

Switching between multiple time multiplexed channels with different statuses requires holding the status of each channel to determine the operation to be applied on the respective incoming input sample. The channel_status memory holds 128 words describing the status of each channel handled by the spike detection unit. Each word has fifteen bits. Two bits describe the state of the channel, and 13 bits hold the FIFO address needed to copy the AP samples at the right location and space assigned for it on the output buffer in case a spike was detected. The channel status bits and the cases they represent are shown in Table 3.1.



(a) **BRAM memory space assigned to a channel *k* at instance (n+4), at which the spike is detected.**

(b) BRAM memory space assigned to a channel *k* at instance (n+5), at which the first 16-channels of AP are copied to output buffer

Fig 3.4 An example of the arrangement of samples in the input BRAM space assigned to a channel k, when a spike is detected and when the initial part of corresponding AP waveform is copied to the output buffer.

Channel-status bits	**Channel-status description**
00	The channel has no detected spikes
01	The channel has a detected spike, time-stamp and channel ID were saved on output buffer. The first 16 samples need to be copied as a complete portion to the output buffer
10	AP samples 17 to 30 are being read sample by sample upon their arrival at the input BRAM
11	AP samples 31 to 46 are being read sample by sample upon their arrival at the input BRAM

Table 3.1: Channel-status-bits and the corresponding status description

3.2.5. The BRAM Read Control:

When the unit receives a sample from one of the channels it is written in the input memory. The BRAM read control checks the status of the channel being updated and plans the reading procedure accordingly. The channel_status word can indicate 3 possible cases:

(1) **The channel has currently no detected spikes:**

In this case the incoming sample is sent to the NEO module and threshold comparator for testing. If a spike is detected, a memory block space of 48 words is saved in the buffer to hold the corresponding AP waveform. The spike detector unit has a spike counter that is used along with a look up ROM to determine the first output buffer memory space available for AP waveform storage as shown in Fig. 3.3. If a spike is detected, the counter is incremented, and the time stamp and channel ID of the detected spike are copied into the lower first available buffer address indicated by the look up ROM. The channel_status word is updated to save the block base address that saves a space on the output buffer to hold the AP waveform. This case is represented by a channel-status = 00.

(2) **The channel has a detected spike and a saved memory space in the buffer:**

In this case the reading control copies the first 16 samples of the AP waveform available in the input BRAM to the output buffer memory. (10 pre-spike samples, 1 spike sample, 4 post-spike samples required for the NEO and the incoming sample) This is the longest cycle of the copying process. It takes a total of 19 clock cycles to complete. The unit has to complete it before reading a new sample. This case is described by a channel-status = 01.

(3) **The refractory period of the AP waveform is being completed:**

The incoming sample is copied directly to the output buffer. The 35 samples of the refractory period are each copied upon arrival at the input BRAM to the output buffer. This step is repeated 35 times to complete the refractory period. At each cycle the channel_status is updated with the buffer address that will hold the next incoming sample in the refractory period. Once a spike waveform is completely copied to the output buffer, the BRAM reading control updates the upper-limit for the buffer emptying process. The two states (10 and 11) were split into two states to apply an address counter for the lower 4 bits of the buffer address only, instead of applying an address counter for the whole 13 address bits. The 9 most significant address bits are updated the when the channel moves from state 10 to state 11.

The AP refractory period arrives in single samples at the output buffer. Once the last sample arrives at the input BRAM, it is directly transmitted to the output buffer and the complete waveform becomes available for further processing or transmission to a host PC. The design avoids queuing-based transmission, that arise from copying the AP waveforms as a whole to the output buffer. The memory space assigned for each channel on the input buffer memory is also reduced. The spike detection module and output buffer have access to read data samples from input BRAM.

3.2.6 Operation Management:

To control the sequence and timing of operations, a controller employing a finite state machine is used. Figure 5.4 presents an overview of the BRAM read control state diagram. The channel status word has two bits describing the spike copying stage. They are used to decide whether input stream should be passed through the NEO detection module or copied directly to the output FIFO.

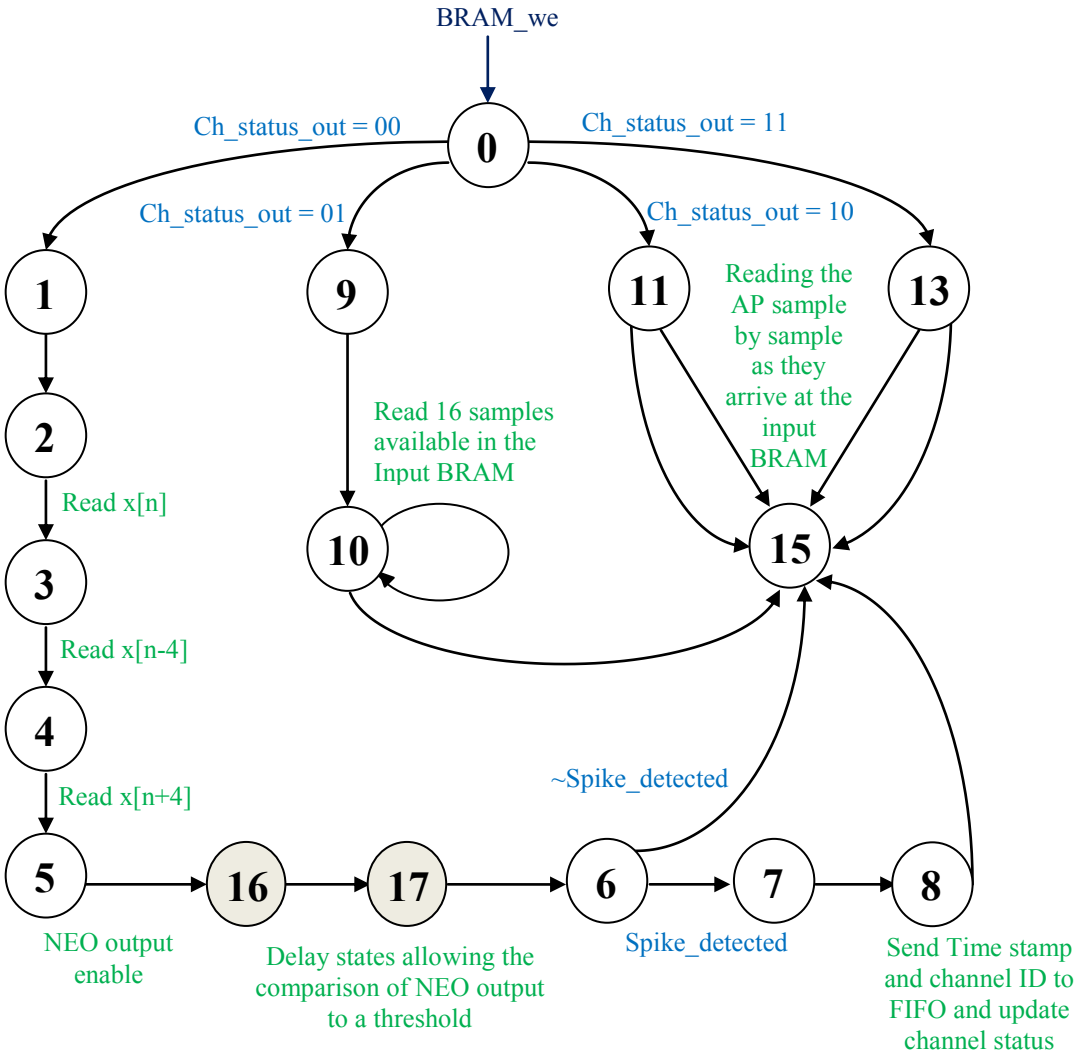


Figure 3.5: Overview of the state diagram describing the controller operation

3.2.7 Autonomous Threshold selection:

With the high channel count automatic threshold selection for each channel is vital. After reset, the system starts computing the threshold for each channel as a multiple of the Mean Deviation MD of a window of its incoming data. The channels are disabled until their thresholds are calculated, and saved on a threshold RAM. Fig.3.6 describes the details of the NEO preprocessing, threshold comparator operation and threshold computation.

In the normal operation, the samples are passed through the NEO module, the computed output is compared to the threshold of the corresponding channel. In the case of threshold computation, the output of the NEO is passed to a MD computation (eq.3.2),

$$MD = \frac{1}{N} \sum_{N} |NEO[n]| \tag{3.2}$$

where N is the window size of the data being used to measure the background noise.
N is chosen to be a power of 2, so that the division by N can be performed by right shifting of the dividend. Based on the threshold selection guidance provided in literature [4]the multiplier is chosen to be 16.

Each channel is affiliated with two bits in the enable-disable queue register that determine the state of the threshold computation. The enable-disable queue is used to determine whether the channel is disabled (11) as it still does not have a computed threshold yet, or whether the channel is undergoing a threshold computation (10), or if the channel has a valid threshold and is enabled for spike detection (00). The register is shifted 2 bits to the left whenever one channel has finished the threshold selection.

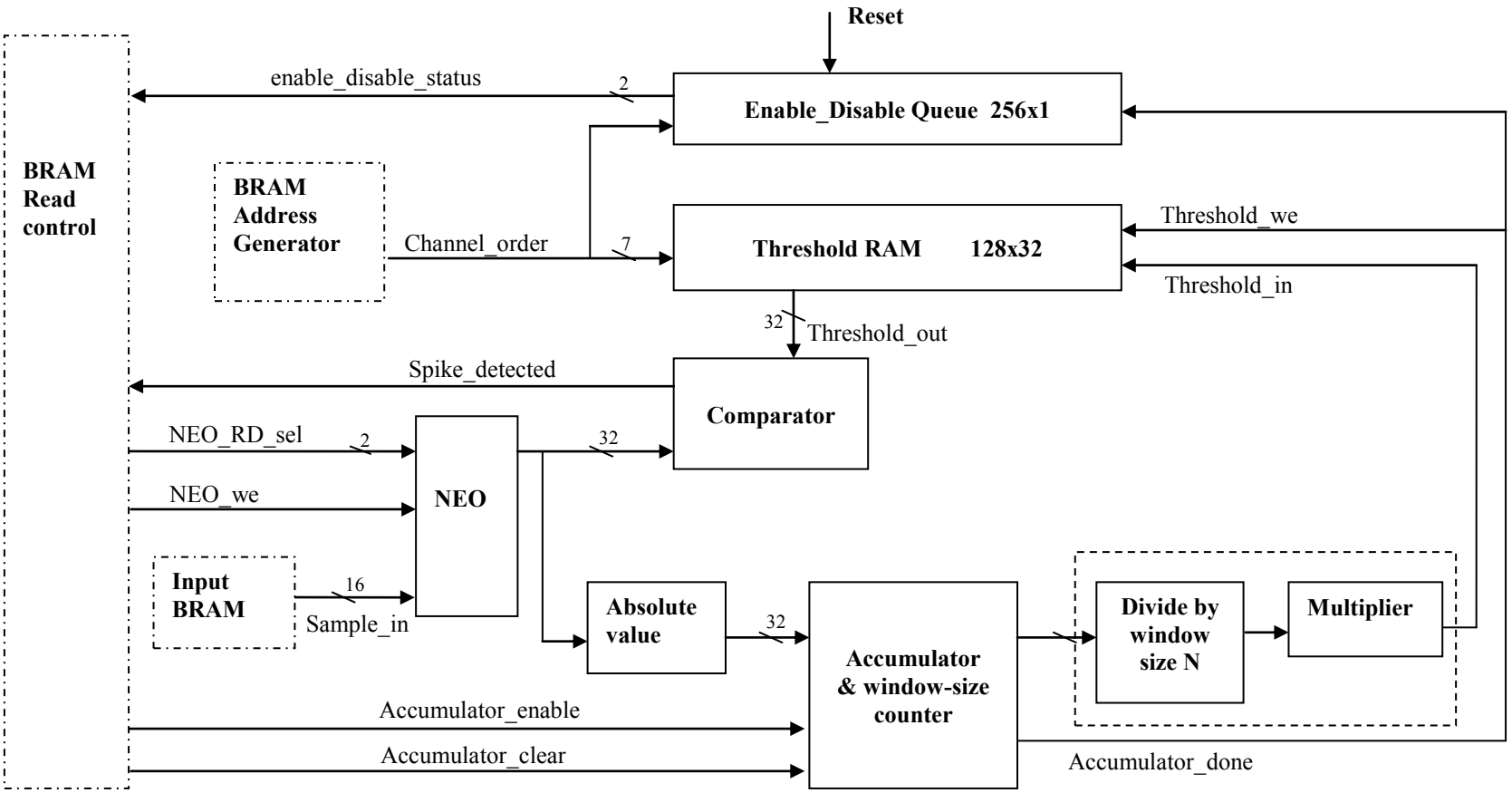


Fig.3.6: Block diagram describing the NEO preprocessing, threshold comparator and threshold computation

3.3. Integration of Several Spike Detection Units:

The total number of channels to be processed is reconfigurable. According to the neural signal processing algorithm used, the longest process applied after sample reading was to copy the first 16 samples of an AP. This procedure required nineteen clock cycles. To have an optimum hardware usage, twenty spike-based reduction units were integrated, so that channels on other units can be updated with their respective sample inputs while this longest procedure is being completed, and before that same unit receives a new incoming sample. Fig.3.7 presents the initial integration of twenty spike detection units to handle a total of 2560 channels. The detailed implementation of the integraton of twenty spike detection units on FPGA and the copying of the AP waveforms to a common output FIFO is explained in Chapter 4.

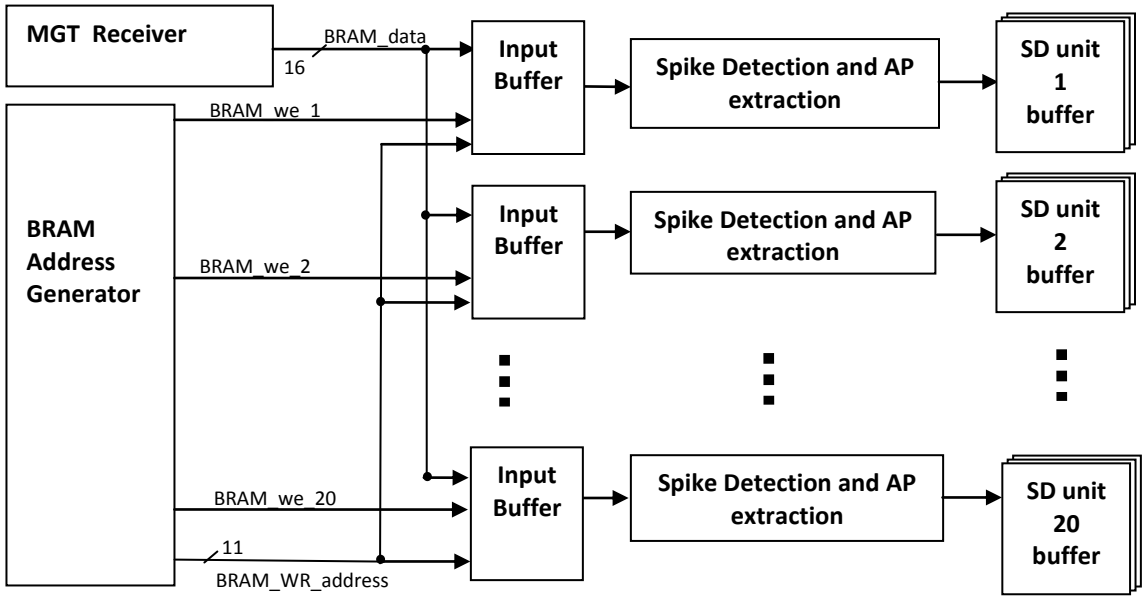


Fig. 3.7: The integration of twenty spike detection units to handle a total of 2560 channels.

3.4. Addressing and Timing:

The BRAM assignment has been chosen so that the BRAM_address can provide direct information on the channel order on the input BRAM and the sample number as shown in Fig.3.8. The write address generator constructs the BRAM write address to rearrange the sample data in preparation for a structured processing. It concatenates the output of three counters to write each sample data in the corresponding channel location.

The BRAM address generator operates at a frequency f, where:

f = sampling frequency per channel x number of channels

For the example of integrating twenty SD units, the BRAM address concatenates the output of three counters:

(a) a 5-bit counter presenting the Input BRAM ID (20 input BRAMs)

(b) a 7-bit counter presenting the channel order on the BRAM (128 channels per BRAM)

(c) a 4-bit counter presenting the sample number. (16-sample space per channel)

Counter (a) is the fastest changing at every clock cycle. Counter (b) is incremented after (a) reaches a full count cycle of 20 and then is reset. Counter (c) is the slowest counter, that only increments at the full count of counter (b).

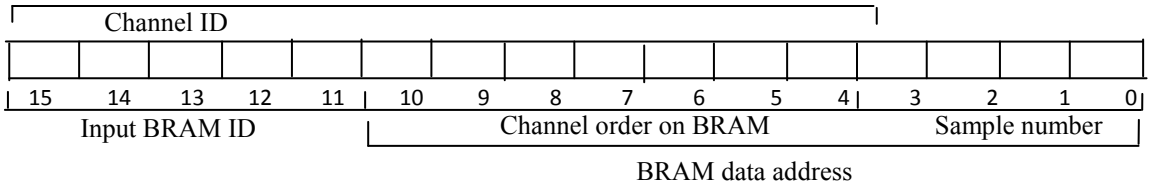


Fig.3.8 BRAM write address structure generated by the write-address-generator block.

3.5. Transmitting the APs from the Output Buffers to a Host PC:

The design structure can be extended to integrate spike sorting blocks. In this case the spike sorter will be reading the AP waveforms from the output buffers in their complete format. The dissertation work does not include a spike sorter, and the AP waveforms were sent to a host PC for system evaluation. The data were transmitted using PCI express (Peripheral Component Interconnect express) to a host PC. The data transmission performance was closely examined to make sure that the transmission latencies meet the system requirements and that there is enough hardware resources to cover the expected transmission queue depths. The system was tested for performance integrity assuring that no data was dropped.

Real-time hardware-implemented neuronal spike-based data reduction schemes are an attractive method to alleviate the bandwidth requirements for raw data transmission, and to increase the data acquisition throughput. The idea of data reduction was to send only the spike waveforms while disregarding the inter-spike samples. The spike waveforms are the only information needed for successive spike sorting. Based on the sparse nature of the neural signal with respect to time, and the average neuron firing rates, the amount of sent data can be reduced to approximately ~2.25% of the total amount of raw data [24].

The transmission from several output buffers corresponding to multiple Spike Detection units required the use of an intermediate FIFO to copy the AP waveforms to before transmission to the host PC. The copying process from multiple buffers was scheduled using queuing based control as explained later in Chapter 4.

3.5.1 Transmission Latencies:

With a focus on telemetry transmission, Bossetti et al [24] raised an important design consideration for spike-based data reduction in real-time. It was demonstrated that although the spike-based compression might be very appealing from the point of view of average bandwidth, it is subject to transmission bottlenecks during periods of multichannel neuron bursting causing queuing-based delays at the output buffer. They drew the attention to the relation between the ratio of the output to average input bandwidth and transmission latency, the number of samples per spike waveform, the mean firing rate MFR, and the needed queue depth of the output buffer memory. Bottlenecks and latencies are mainly a consequence of accumulating the input data samples over short periods of time before their transmission at the output. Based on statistical data performed on a 32-neuron system with an average neuron firing rate of 8.93 spikes/s, it was concluded that the output bandwidth had to be 3-5 times the overall average input firing rate to reduce the average maximum delays to less than the recommended limits of 10ms [24].

The model that they used relied on finding the average Firing Rates FR over 1ms time intervals and calculating the corresponding accumulation of AP waveforms in the output queue at different transmission rates. Their model neglected the reading and writing delays and assumed that the spikes were sent in bulk to the output FIFO. It was worth investigating if their model based on 32-neurons can be applied with the same binning parameters can be applied on a high-channel count system, and if the same transmission to FR ratio requirements would still apply to limit transmission latencies.

3.5.2 Overview on Bursting:

Burst is a term used in literature to describe a neuron's firing in a clustered pattern. Each such burst is followed by a period of quiescence. Burst synchronization refers to the alignment of bursting and quiescent periods in interconnected neurons. Burst synchronization is the phenomenon that causes the longest queuing based transmission delays. Neocortical neurons can be classified into different types according to their pattern of spiking and bursting. All excitatory cortical cells are divided into three main classes as shown in Fig 3.9. and they are: Regular Spiking (RS) neurons, intrinsically bursting (IB) neurons and chattering (CH) neurons. [42,44]

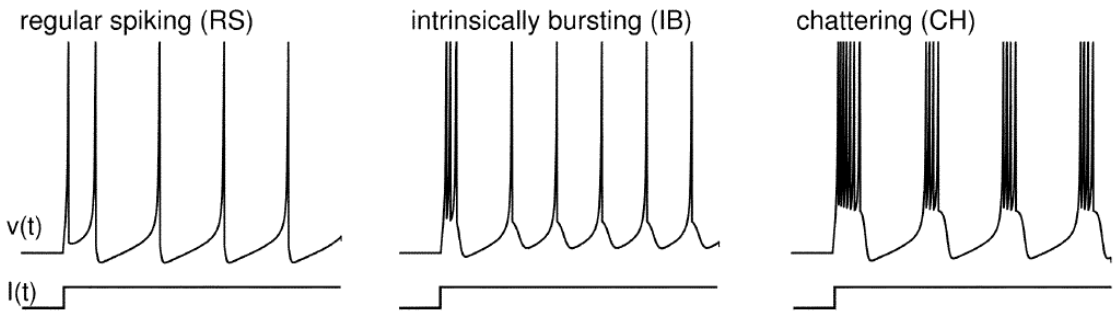


Fig. 3.9 Typical spiking patterns of cortical excitatory RS, IB, and CH neurons. This figure is reproduced with permission from www.izhikevich.com. (Electronic version of the figure and reproduction permissions are freely available at www.izhikevich.com.)

- **Regular Spiking (RS) neurons:** RS neurons are the most commonly encountered neurons in the cortex. When stimulated at threshold, an RS neuron generates only one spike. As the stimulus amplitude increases, the neurons respond with an initial high-frequency spike output, then they exhibit obvious frequency adaptation. A neuron might produce clusters of spikes in response to synaptic input. Some literature have reported a starting frequency spike output response of 320Hz, which declined to a much lower sustained frequency (< 100 Hz) within less than 50msec. [36]

- **Intrinsically Bursting (IB) neurons:** IB neurons fire a stereotypical burst followed by repetitive spikes. Bursts are often the minimal response to a threshold stimulus. A burst can consist of few spikes firing at high frequencies in the range of 300 Hz and then followed by individual spikes firing at 15-20 Hz.

- **Chattering (CH) neurons:** CH neurons can generate rhythmic stereotypical bursts of closely spaced spikes. The typical inter-burst frequency is in the range of 5-15 Hz [5]but can also be as high as 40 Hz.

In general, if a network of bursting neurons is linked, it will eventually synchronize for most types of bursting. Synchronization can also appear in circuits containing no intrinsically bursting neurons; however its appearance and stability improves if the network includes intrinsically bursting cells. Some literature described multichannel bursting as Neuronal Avalanches. Spiking activity propagates as individual neurons trigger action potential firing in subsequent neurons. They initiate a cascade that spreads through the neuronal network [37].

3.5.3. Super-Bursting:

High frequency network-wide bursting has been reported in research monitoring neural activity using MEA. This "super-bursting" was documented as a phenomenon of early plasticity that is ultimately refined into mature stable neural network behavior. Developmental super-bursting is thought to accompany transient states of heightened plasticity both in culture preparations as well as across brain regions.[38].

3.5.4. A Model to estimate the required Transmission Rate:

Designing a platform that should handle hundreds to a few thousands of recording channels, it was essential to test if the output/input bandwidth ratio values recommended by previous literature, based on a limited number of monitored neurons, holds for a larger number of neurons. A transmission model was created in MATLAB to carry out simulations on the neuronal activity recordings of 2550 channels over 2.5 seconds. The model was constructed to detect spikes using the NEO operator. The threshold was set at 10 times the mean deviation over the complete 2.5 seconds of recording time. Each channel was handled separately and the spike times were saved.

Then simulations were carried out using a windowing format [24]. In this case, spikes across the 2550 channels were collected over a 1 milliseconds period, rounding the recorded spike times to the nearest 1 millisecond. The queue depth based on the estimated transmission rate was found along the recording time. The spike times and transmission rate were used to calculate maximum latencies and queue depths.

At each rounded spike time, the corresponding detected spikes were added to the queue. The transmission rate determined how much of that data could be transmitted

before the next load of binned spikes arrived, as well as the time required to remove the data from the queue. If spikes arrived before the queue was empty, the new data was added to the queue, increasing its depth. Latency was calculated from the queue size and represented the total amount of time required to remove all of the data from the queue at the estimated transmission rate. Following the recommended ranges of bandwidth ratios,[8] the average firing rate was measured for the recorded data set used for testing and the transmission rate was set to be 5 times the MFR.

3.5.5. Data sets used for testing:

To test the model four in vitro data sets were used. The neural signals recorded using high-density MEA from 3Brain (www.3Brain.com) have been supplied by the NetS3 Lab in the Neuroscience Department of the Instituto Italiano di Tecnologia (IIT), Italy. Two sets were recorded using dissociated rat hippocampal cells (22 days in *vitro*) and two sets were taken from rat cortical cultures (21 days in *vitro*). The hippocampal and cortical neurons have a different dynamic firing pattern that was interesting to observe and analyze using this model (Fig 3.10-13). The hippocampal neurons tend to have a more synchronized firing behavior showing clear bursting events followed by relatively silent intervals. It was expected that they may represent a more critical case for the designed model in terms of the queue depths. The cortical neurons tend to be less synchronous and more spread from a spatial point of view. The validation of the model on cortical neurons was important since they are the mostly recorded type of neurons especially in vivo.

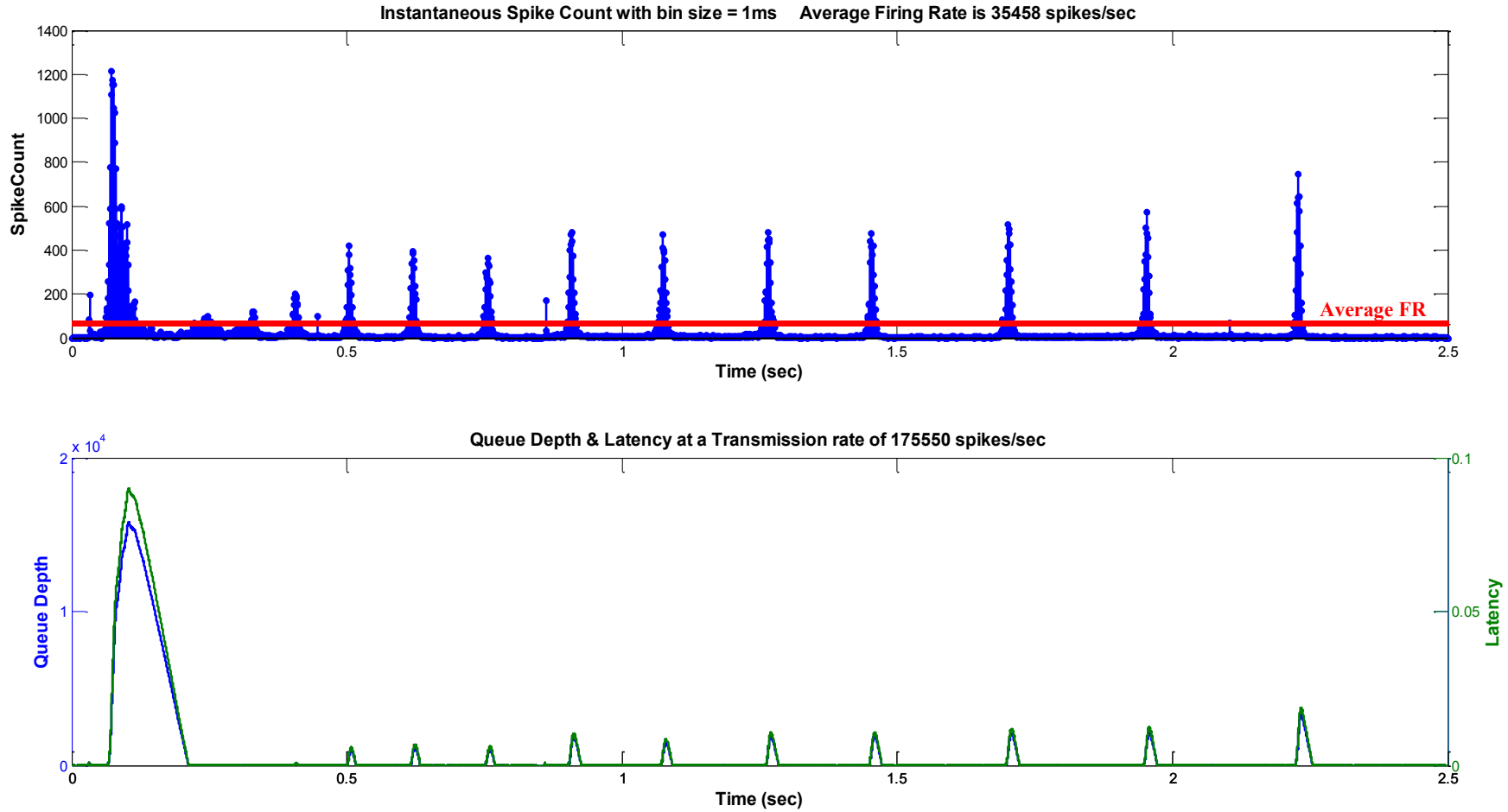


Fig. 3.10 Simulation results based on data recorded from dissociated rat hippocampus cell in vitro. In the upper figure, the average of the instantaneous firing rate based on 1ms bins was ~35Kspikes/sec. The lower graph shows the queue depth and corresponding latency in sec when the transmission rate is set to 5 times the average firing rate ~ 175Kspikes/sec.

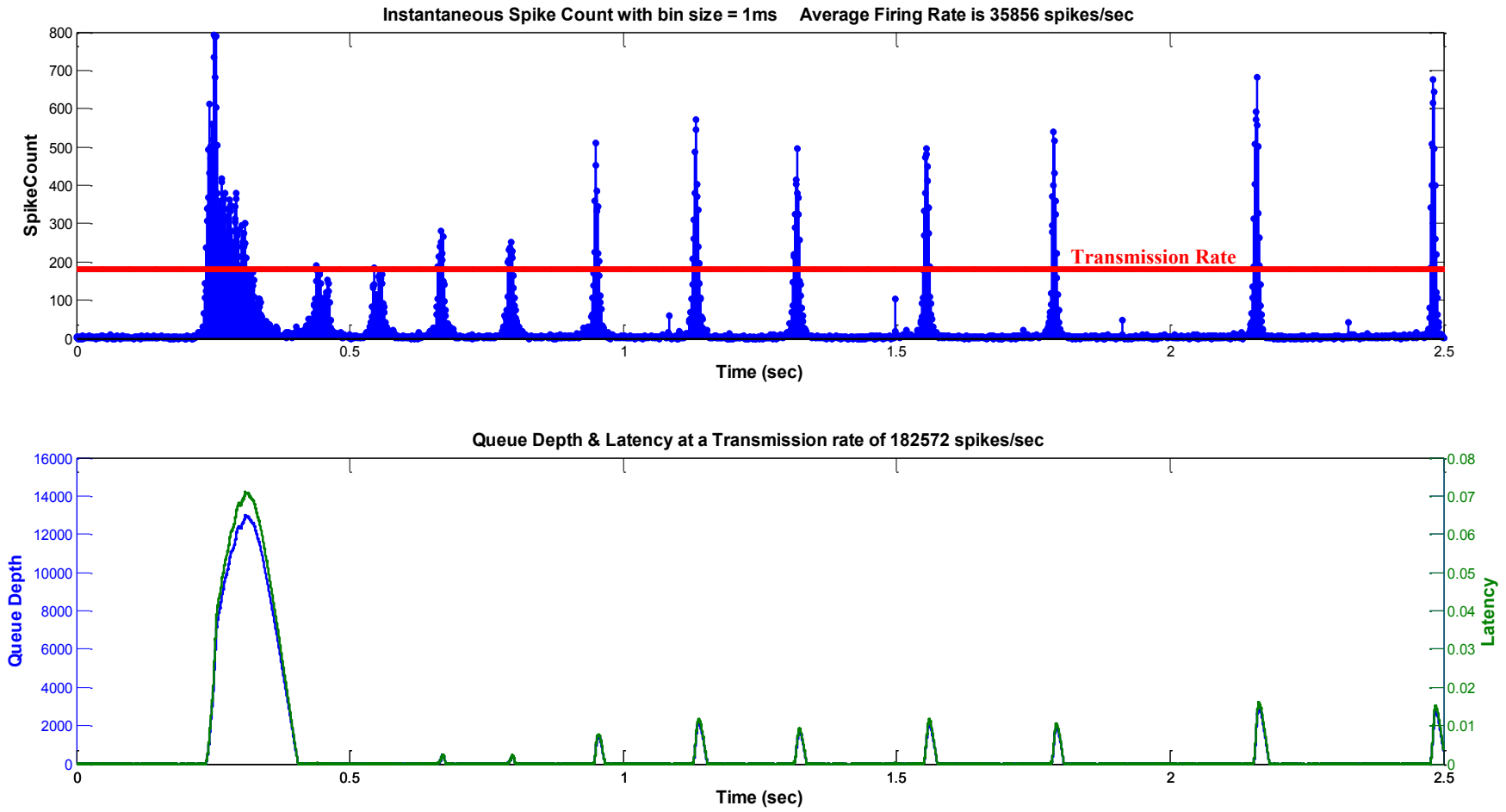


Fig. 3.11 Simulation results based on data recorded from dissociated rat hippocampus cell in vitro. In the upper figure, the average of the instantaneous firing rate based on 1ms bins was ~36 K spikes/sec. The lower graph shows the queue depth and corresponding latency in sec when the transmission rate is set to 5 times the average firing rate ~ 183Kspikes/sec.

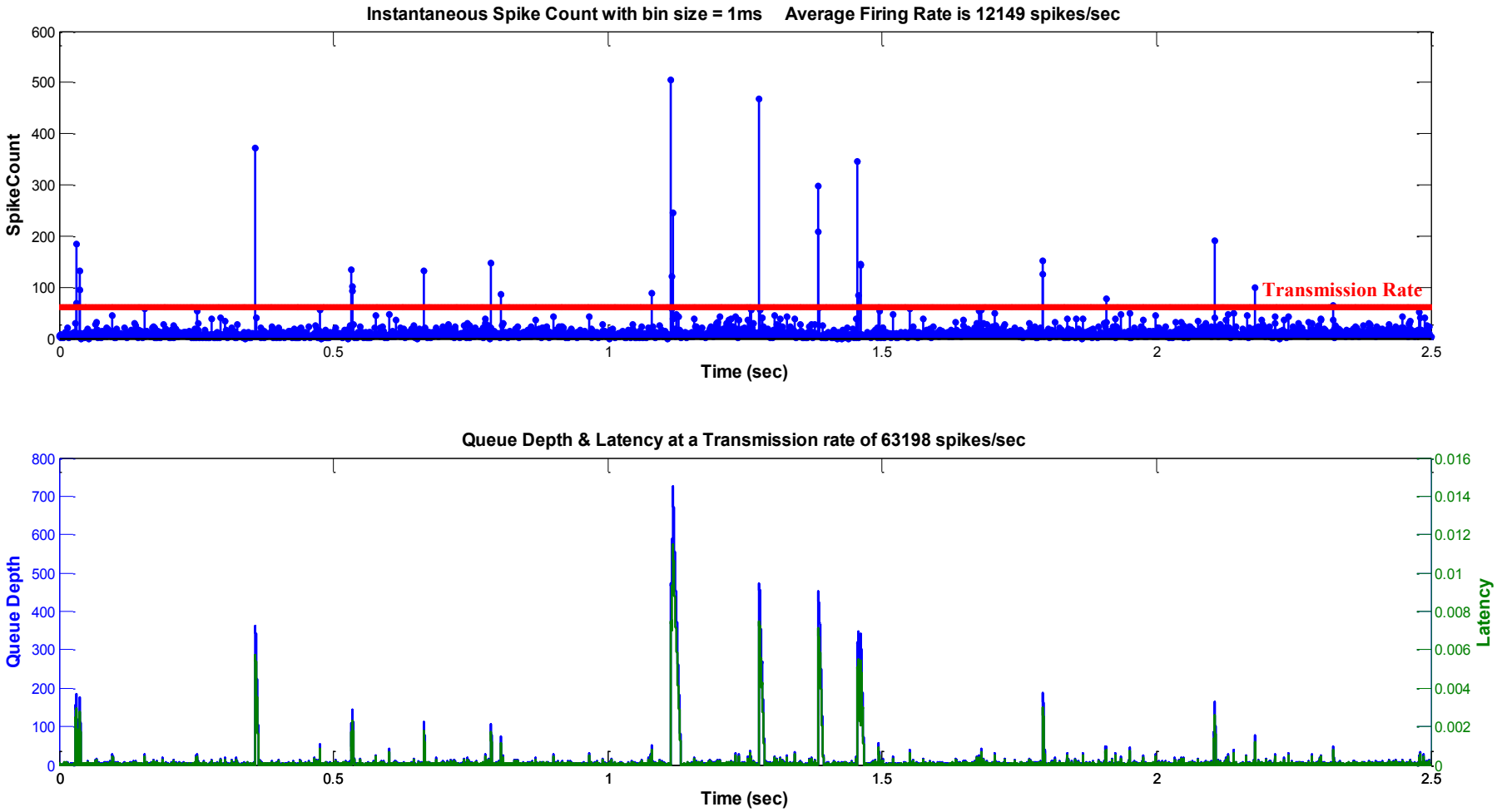


Fig. 3.12 Simulation results based on data recorded from dissociated rat cortex neurons in vitro. In the upper figure, the average of the instantaneous firing rate based on 1ms bins was ~12Kspikes/sec. The lower graph shows the queue depth and corresponding latency in sec when the transmission rate is set to 5 times the average firing rate ~ 64Kspikes/sec.

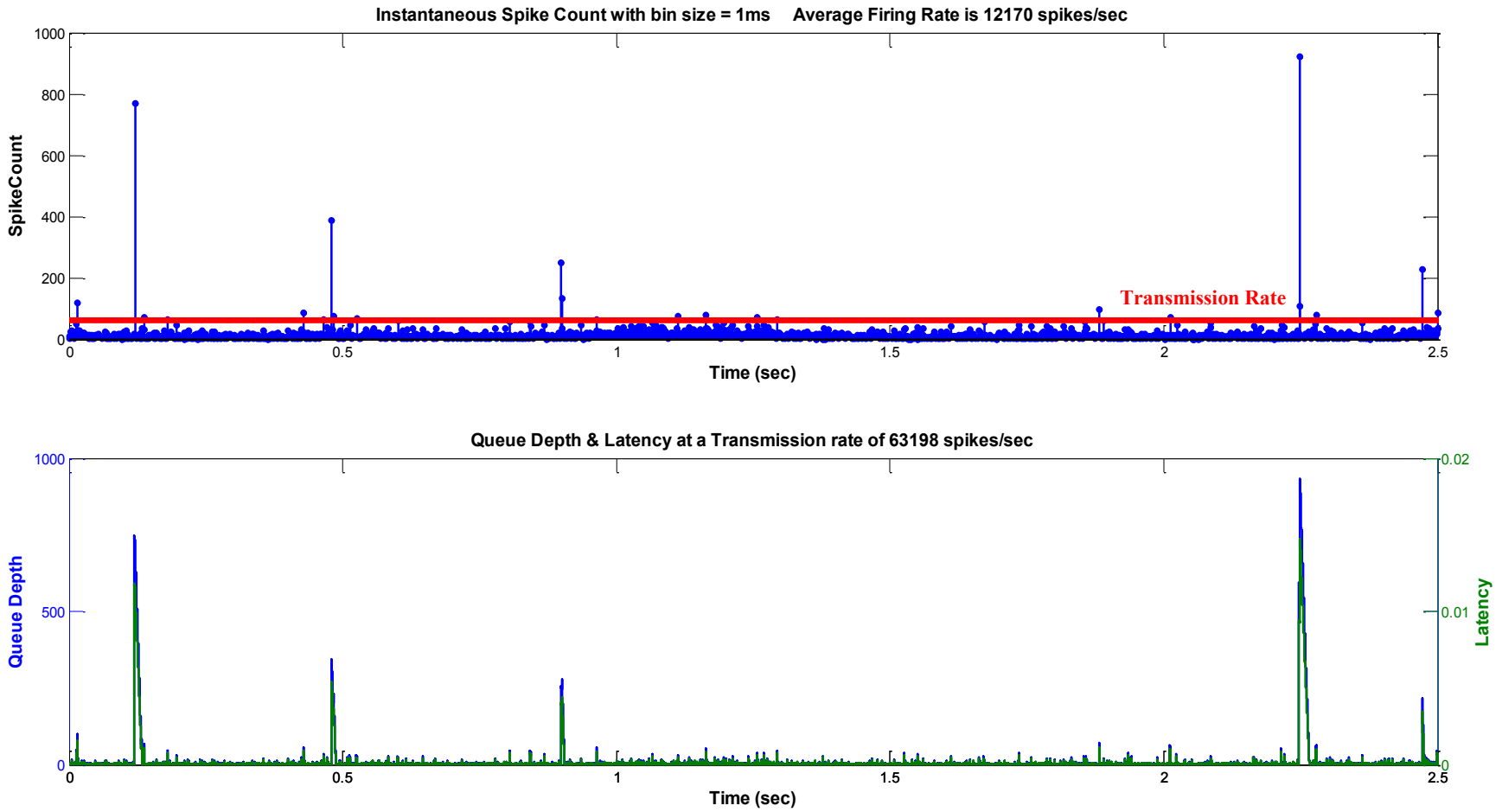


Fig. 3.13 Simulation results based on data recorded from dissociated rat cortex neurons in vitro. In the upper figure, the average instantaneous firing rate based on 1ms bins was ~12Kspikes/sec. The lower graph shows the queue depth and corresponding latency in sec, when the transmission rate is set to 5 times the average firing rate ~63Kspikes /sec.

3.5.6 Approximation approach for computing the transmission rate:

The instantaneous Firing Rate is the neuronal firing averaged over temporal bins. The accumulation of spikes in the queue occurs when the instantaneous FR is greater than the transmission rate. This can result from spike synchronization over multiple channels or multichannel neuron bursting. To set an approximation model, the recording time was divided into intervals where either the TR was higher or lower than the instantaneous FR. The average FR over each interval was calculated as shown in Fig. 3.14. During the bursting intervals, spikes accumulate in the queue and the queue reaches its maximum at the end of the bursting time. After the bursting event, during the following quiescent time or reduced neuronal activity, the queue is gradually emptied.

The maximum queue depth can be obtained by integrating the accumulated spikes in the queue over the bursting time as given in eq. 3.1. The accumulated spikes result from the difference between the average FR during the bursting activity and the TR.

$$\text{Max\_QueueDepth} = (\text{BurstFR} - \text{TR}) \cdot \tau_{\text{burst}} \tag{3.1}$$

Where Max_QueueDepth is the maximum queue depth, BurstFR is the average firing rate over the bursting interval; TR is the transmission rate and τ_{busrt} is the bursting time.

The corresponding latency would be:

$$\text{Max Latency} = \frac{\text{Max QueueDepth}}{\text{TR}} \tag{3.2}$$

The queue depth calculated using the approximation model and the exact queue depth are compared in Fig. 3.14.

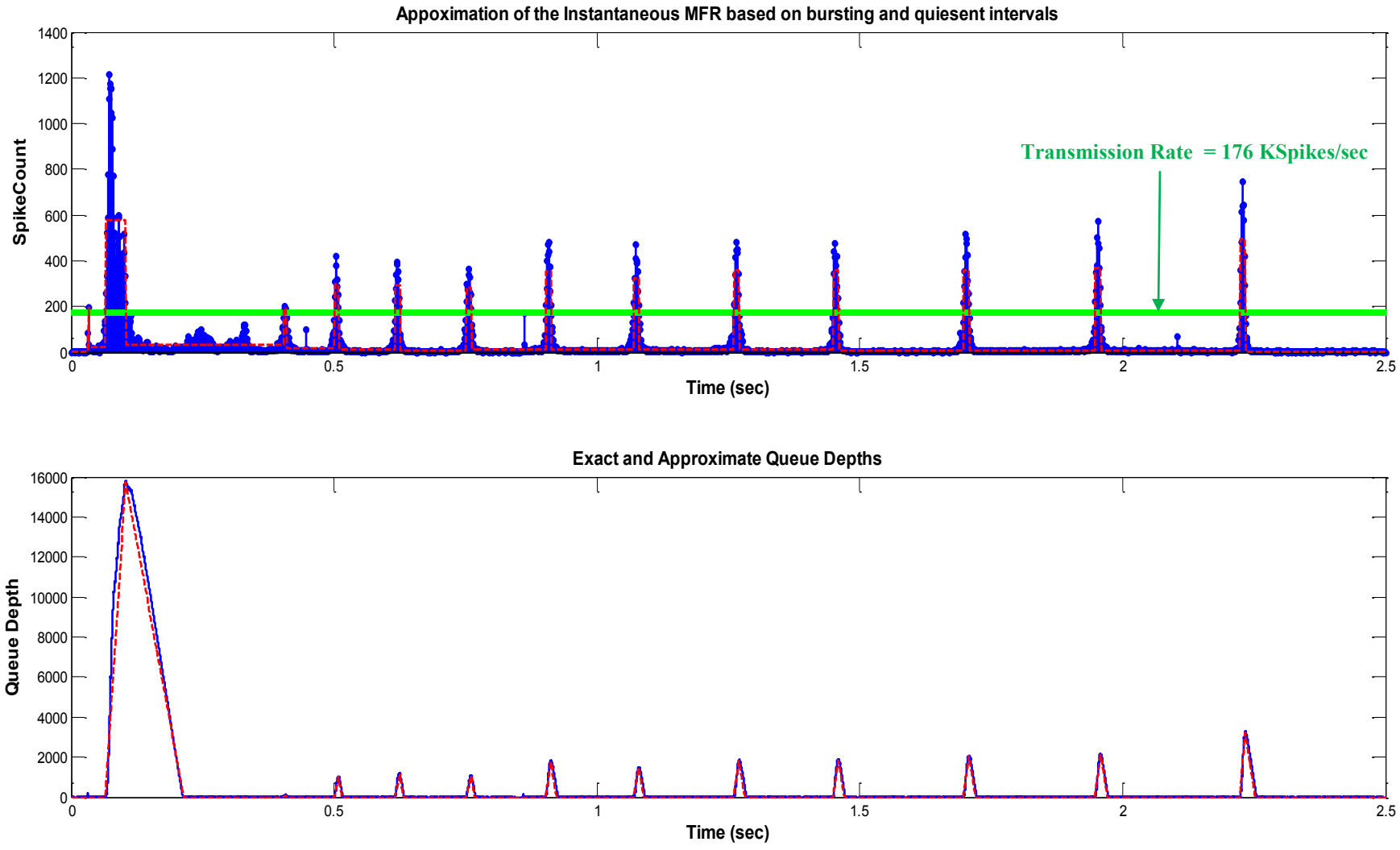


Fig.3.14 Approximation of the instantaneous firing rates of the hippocampus neuronal data recordings to obtain a closed formula for estimating a transmission rate for sending the spike waveforms to a host PC. The dotted line presents the approximate queue depth and the straight line is the exact queue depth obtained from the MATLAB model.

3.5.7 Simulation Results:

The simulation results of the neuronal firing data transmission model show that the bursting and super-bursting times present the most critical intervals for the system. In BMI applications, for example the queuing-based transmission delay must fall below 10 milliseconds. Setting a value for the transmission rate will not only depend on the limits for queuing-based transmission latency, but also on the memory resources available to save the queue on the hardware used.

The data sets recorded from the hippocampus neurons showed a synchronous multichannel bursting, and had the maximum queue depth requirements based on the simulations in MATLAB. If the maximum queue depth is 15,000 spikes and a spike waveform holds 50 samples with each sample being 2 bytes long, then the design requires a memory of approximately 1500 Kbytes just to save the queue. A memory space of 1500 Kbytes would translate to approximately 370 BRAMs (36Kbits each) on the FPGA. Following the literature recommendation [24], the transmission rate was set at five times the MFR. Increasing the data transmission rate would ease the memory burden on the FPGA resources, as some FPGA models do not have this amount of BRAMs. For example in the hippocampal recordings the TR was about 176Kspikes/sec. Considering 50 samples per spike and 2 bytes per sample we the TR was modeled at 17.6 Mbytes/s. Considering the data transmission options from an FPGA to a host, such as Ethernet and PCIe, there is some room for transmission rate increase. With the Ethernet and PCIe offering transmission rates in the range of at least few hundreds of Megabytes per second.

The recordings acquired from the cortical neurons demonstrated a more uniform firing rate, and a much lower average value. This fact was reflected on the results obtained for the queue depths and associated latencies as shown in Fig. 3.12 and 3.13.

The hippocampus is a main component of the brains of vertebrates. It is located under the cerebral cortex and belongs to the limbic system. It plays a major role in fusing the information from short-term memory to long-term memory and in spatial navigation. The subiculum, a component of the hippocampal formation, is thought to perform relaying of signals originating in the hippocampus to many other parts of the brain . In order to perform this function, it uses intrinsically bursting neurons to convert promising single stimuli into longer lasting burst patterns as a way to better focus attention on new stimuli and activate important processing circuits [39]. The detailed explanation of the firing dynamics of neurons from different parts of the brain is beyond the scope of the dissertation, but it was appealing to search for an explanation for the reason behind the similarity between the results obtained from the hippocampal data set used in testing and the firing patterns of Intrinsically Bursting IB neurons.

3.5.8 Limitations of the model:

(1) The model, designed in MATLAB, assumed that the spikes within a 1msec bin are sent to the output FIFO as a block at the same instant. The data used was recorded at a sampling rate of 7.022 KHz, meaning that the bin collected the spikes occurring across a time approximately equivalent to seven sampling periods. With a high count of channels and high transmission rates the binning size of 1msec may be relatively large.

For a TR of 50 Msamples/sec, and 48 samples per spike, the queue can empty 148 spikes within a sampling period, i.e. before any new samples arrive.

(2) In real implementation, the channels are recorded using TDM, so exact synchrony will not be faced.

(3) At high channel counts, the transmission rate values increase to limit hardware memory. With the TR values approaching the range of the clock frequency on the FPGA, the bulk transmission assumption will not be accurate, and the read and write times as well as the sample by sample transmission to the FIFO have to be considered.

CHAPTER 4

HARDWARE IMPLEMENTATION AND SYSTEM EVALUATION

This chapter details the hardware implementation of the platform design, and how the testing of each building block was performed. The spike detection processing modules were designed using Verilog HDL code. They were simulated using Xilinx® ISim for functional verification. The Xilinx® Core generator was used to configure the integrated blocks on the FPGA such as BRAMs, FIFOs and the Multi-gigabit transceivers. The modules were synthesized and implemented using ISE Design Suite 13.1. For design verification in hardware and as a proof of concept, the design architecture was implemented on a Xilinx® Virtex-5 XUPV5-LX110T FPGA evaluation board. Internal signals were monitored using Xilinx ChipScope.

This chapter covers:

- Data Acquisition High Speed Serial Interface
- Testing the Spike-Based Data Reduction Unit
- Implementation of Xillybus IPcore
- Testing Data Transmission using Real Data Recordings
- Integration of Multiple Spike-Detection units

4.1. Testing Data Transmission Using MGT Transceivers:

In lieu of interfacing the FPGA to a high speed multichannel analog to digital acquisition system, test vectors have been stored on BRAMs on the FPGA. To model the data acquisition process, the test data went through serial transmission using MGT transceivers before reaching the spike detection units.

The Xilinx board, used for hardware implementation provides access to a GTP transceiver through four SMA connectors. The transmitter pair was connected to the receiver pair using two differential copper cables to form an external serial link, as shown in Fig. 4.1.

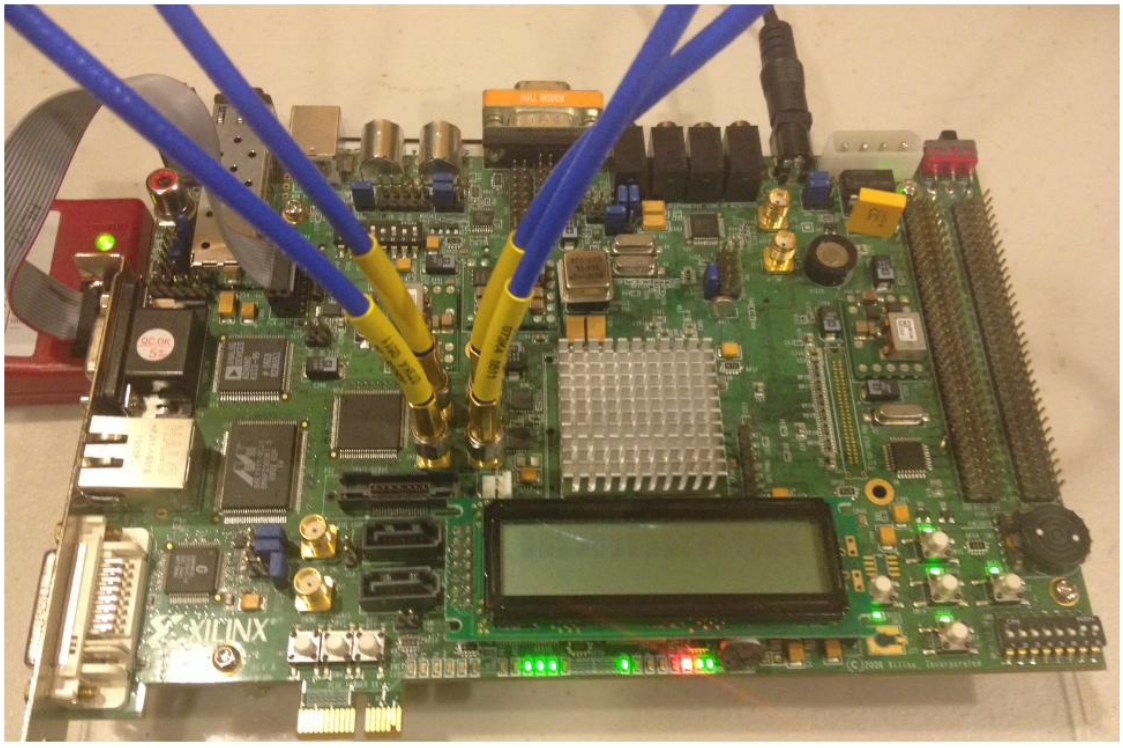


Fig. 4.1 Differential copper cables ASP1-024-ASP1-S402 form an external serial link, connecting transmitter and receiver pairs of GTP1 on dual tile GTP_X0Y5 integrated on the Virtex 5 FPGA on Xilinx XUPV5-LX110t board.

The RocketIO wrapper was created using the Xilinx ISE design tool CORE Generator®. The RocketIO offers useful features to support a wide variety of interface applications and transmission protocols. RocketIO has built in Physical Code Sub-layer features such as 8B/10B encoding, comma alignment and clock correction.

The comma detection and alignment circuit was activated to properly align 16-bit input data during the initialization of the data transmission process. Serial data must be aligned to symbol boundaries before it can be used as parallel data. To make alignment achievable, transmitters send a recognizable sequence, defined as a comma during device configuration. The receiver looks for that predefined comma in the incoming serial data. When it detects it, it shifts the comma to a byte boundary, so that the received parallel words match the transmitted ones. The GTP transceiver includes an alignment block that can be set to align specific commas, or to manually align data using bit-by-bit sliding.

The 8B/10B encoding includes special characters (K characters) that are often used for control functions. To transmit TXDATA as a K character instead of regular data, the TXCHARISK port must be driven high. If TXDATA is not a valid K character, the encoder activates an error signal. At the receiver end, RXCHARISK is asserted when RXDATA is an 8B/10B K character. This feature is not defined for bytes that bypass 8B/10B encoding. To mark the beginning of the valid data stream in the testing process, a K28.5 (10111100 = BC) character was sent. The K character was recognized by the receiver, and the RXCHARISK signal was set high. This control signal was used to trigger an address generator of the spike detection block, in order to assure correct address-data alignment. For transmission testing, the signals were monitored using ChipScope as shown in Fig.4.2.

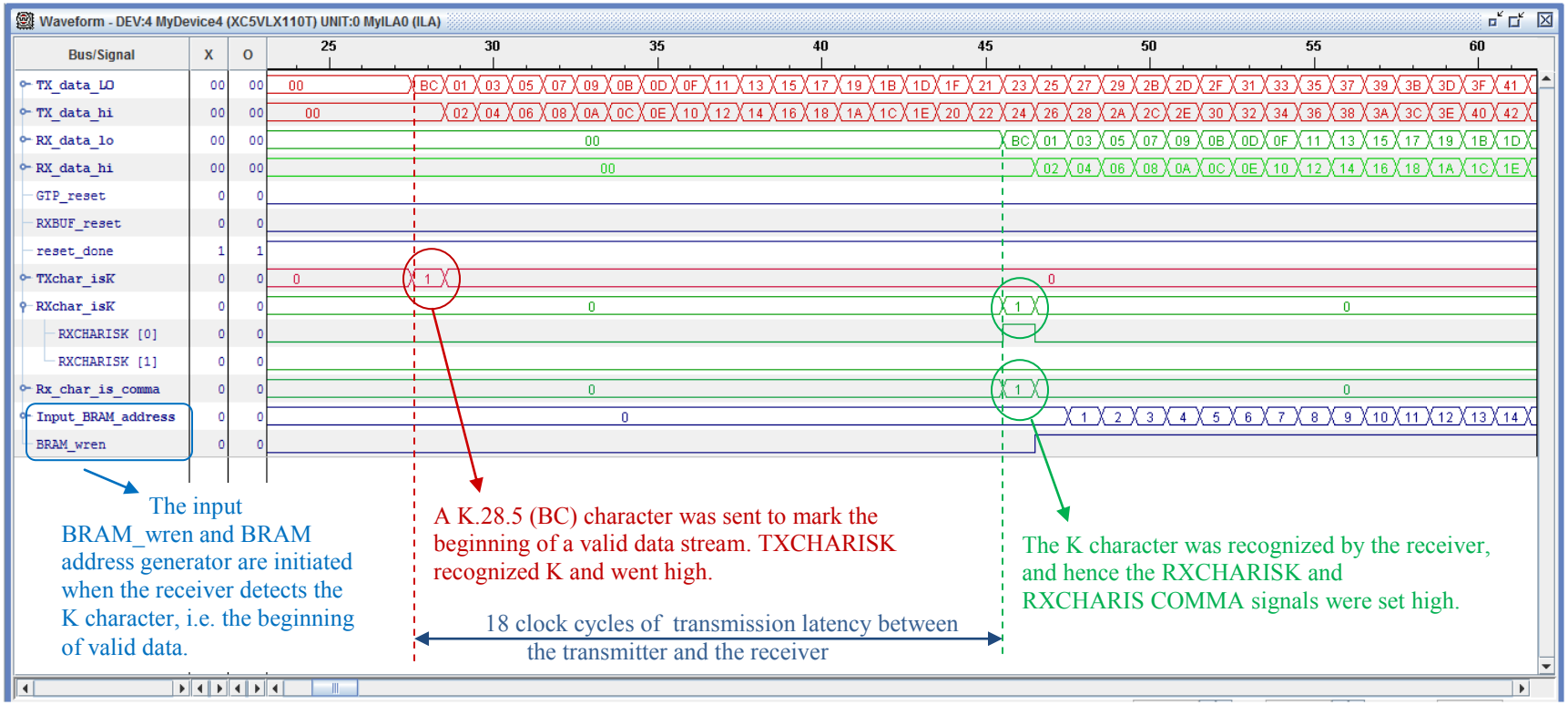


Fig.4.2. A screenshot of the ChipScope waveform window, showing transmitter and receiver 16-bit data split into lower and higher bytes (upper red signals). No data is available on the receiver end before it completes the reset operation and pulls 'reset done' high. TXCHARISK is a 2-bit input control signal at the transmitter end. TXCHARISK [1] corresponds to TXDATA [15:8] and TXCHARISK [2] corresponds to TXDATA [7:0]. TXCHARISK should only be asserted for TXDATA values defined by 8B/10B encoding as K-characters. At the receiver end, RXCHARISK is a 2-bit output signal that is asserted when RXDATA is an 8B/10B K character. Bit 0 corresponds to the lower byte of RXDATA, and bit 1 corresponds to the upper byte. The latency between sending the data at the transmitter data port, and receiving it as a parallel word was 18 clock cycles at a 125 MHz clock, and the target line rate was 2 Gbps.

4.2. Design Verification of the Spike-Based Data Reduction Unit:

The Spike-Based Data Reduction unit, handling 128 channels, was first simulated using Xilinx® ISim. After design simulation, it was implemented in hardware, and tested using ChipScope®. Modeling the data acquisition process, test neuronal data were saved on a BRAM, and then transmitted serially using MGT transceivers to the spike detection units.

The design verification objective was to make sure that the spikes have been detected and that their AP waveforms are copied to the output FIFO with the correct alignment required, correct time-stamp and channel ID.

For this test a window of 256 samples of neural signals recorded at 31.25KHz from, containing only one spike were stored on distributed ROMs and read in a cyclic mode. Using a multiplexer data was sent to selected channels in order to be able to perceive the correct channel IDs at the output buffers as shown in Fig. 4.3.

The design of the spike-detection unit is detailed in Fig 3.2. To save the spike waveforms on the ChipScope memory, a ‘READ’ signal was generated by the FIFO_RD_address_generator module, to indicate when spike waveforms were available for reading in the spike detection buffer. When complete spike waveforms were copied onto the output buffer of a spike detection unit, the upper-limit of the reading address of the buffer was updated. If the reading pointer was below the upper limit of the buffer, the "READ" signal was set high and the FIFO_RD_Address_generator incremented the reading address pointer. The ChipScope read the FIFO_data_out only when the ‘READ” signal was set high. At quiescent intervals, when the "READ" signal was low, the output data of the FIFO was not sampled for efficient use of the ChipScope memory.

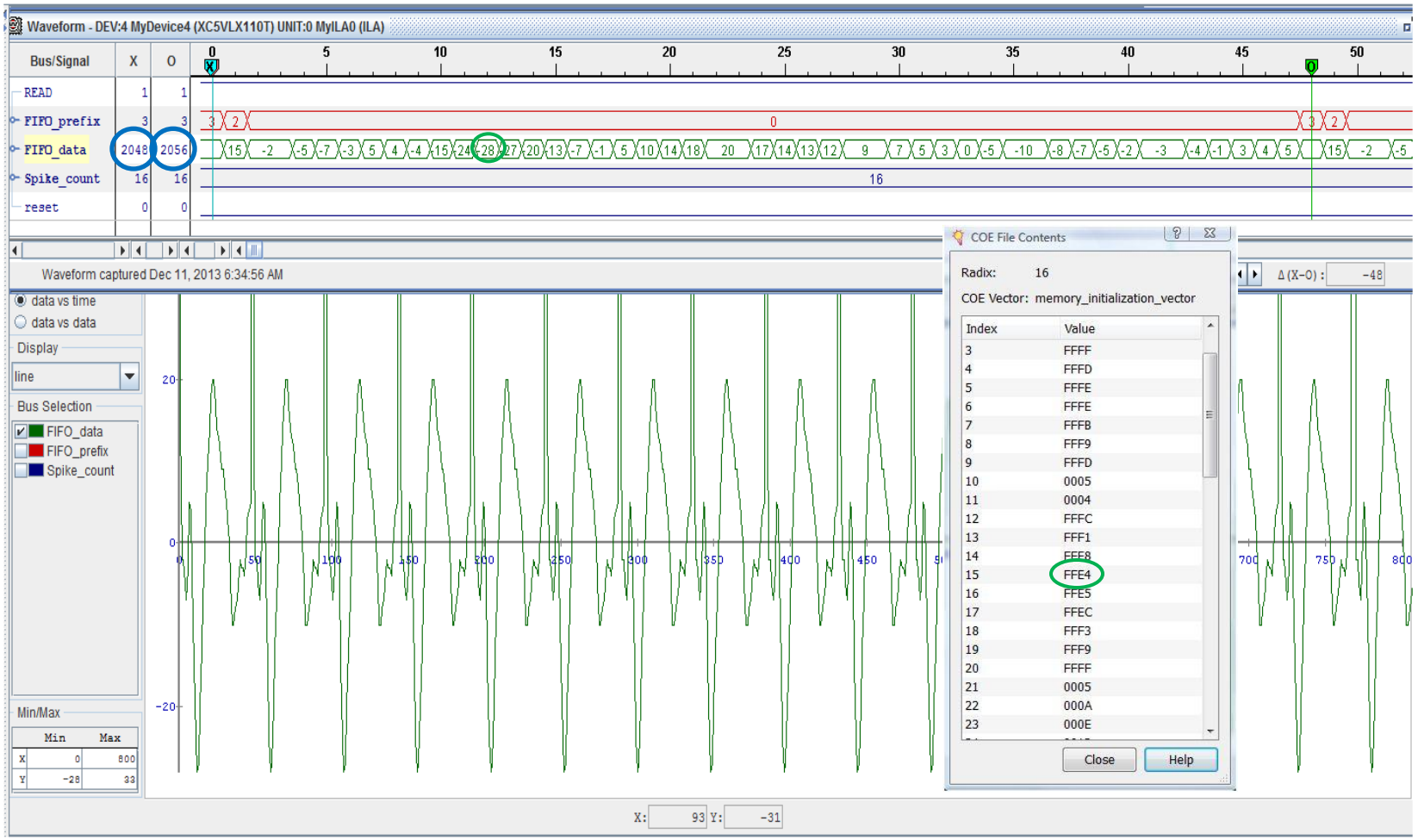


Fig. 4.3. A screenshot from ChipScope® monitoring the data sent to the output buffer of the spike-detection based data reduction unit. The data width is 18bits. The two higher bits are prefix data, indicating whether the lower 16 bits represent: '00' a spike waveform sample, '10' a time stamp or '11' a channel ID. The figure shows two spikes detected on channel 2048 and channel 2056. A spike was detected at time stamp 15. The inset shows the .coe file used to initialize the ROM on the FPGA that stored the neuronal data for this test. The spike waveform carries 46 samples: 10 pre-spike, 1 spike and 35 post-spike samples. The data was sent to ChipScope® BRAM when 'READ' signal was high.

4.3 Xillybus IP Core Implementation:

The Xillybus IP core was implemented on a Virtex-5 FPGA on a Xilinx xupv5lx110t board. The core ports were connected to ChipScope integrated logic analyzer for on-chip testing. Before integrating the Xillybus IP core into the spike-based data reduction platform, its performance was first tested by transmitting predefined data read from a ROM on the FPGA to the host PC. This test was set up to mimic the Xillybus operation in the spike-based data reduction platform. The spike waveform data, to be transmitted to the PC was saved on intermediate buffers then read by the FIFO before transmission using the Xillybus core. The testing data was created using MATLAB® and stored in a .coe file to initialize the ROM having the same size as the spike detection buffer 18x6144. The implementation setting is illustrated in Fig. 4.4.

4.3.1 On the FPGA side:

The design was implemented on a Xilinx xupv5-lx110t board. The implementation user constraint file was modified accordingly. The transceiver block GTP0 was used on the GTP tile assigned for PCIe transmission GTP_DUAL_X0Y2. The integrated endpoint block differential clock pair PCIE_REFCLK_P and PCIE_REFCLK_N is locked to AF4 and AF3 respectively. The pair is driven by an external PCIe source through the PCIe edge connector, and not driven internally. The clock frequency is 100MHz. The integrated endpoint block reset signal PCIE_PERST_B_LS is available on a CPLD and was locked to W10. The 100 MHz clock provided by the PCI Express connector is connected directly to the Virtex-5 FPGA to clock the PCI Express Endpoint Block Plus LogiCORE. It can be used to clock the internal logic on the FPGA or scaled to match the timing restrictions and latency requirements of the design used.

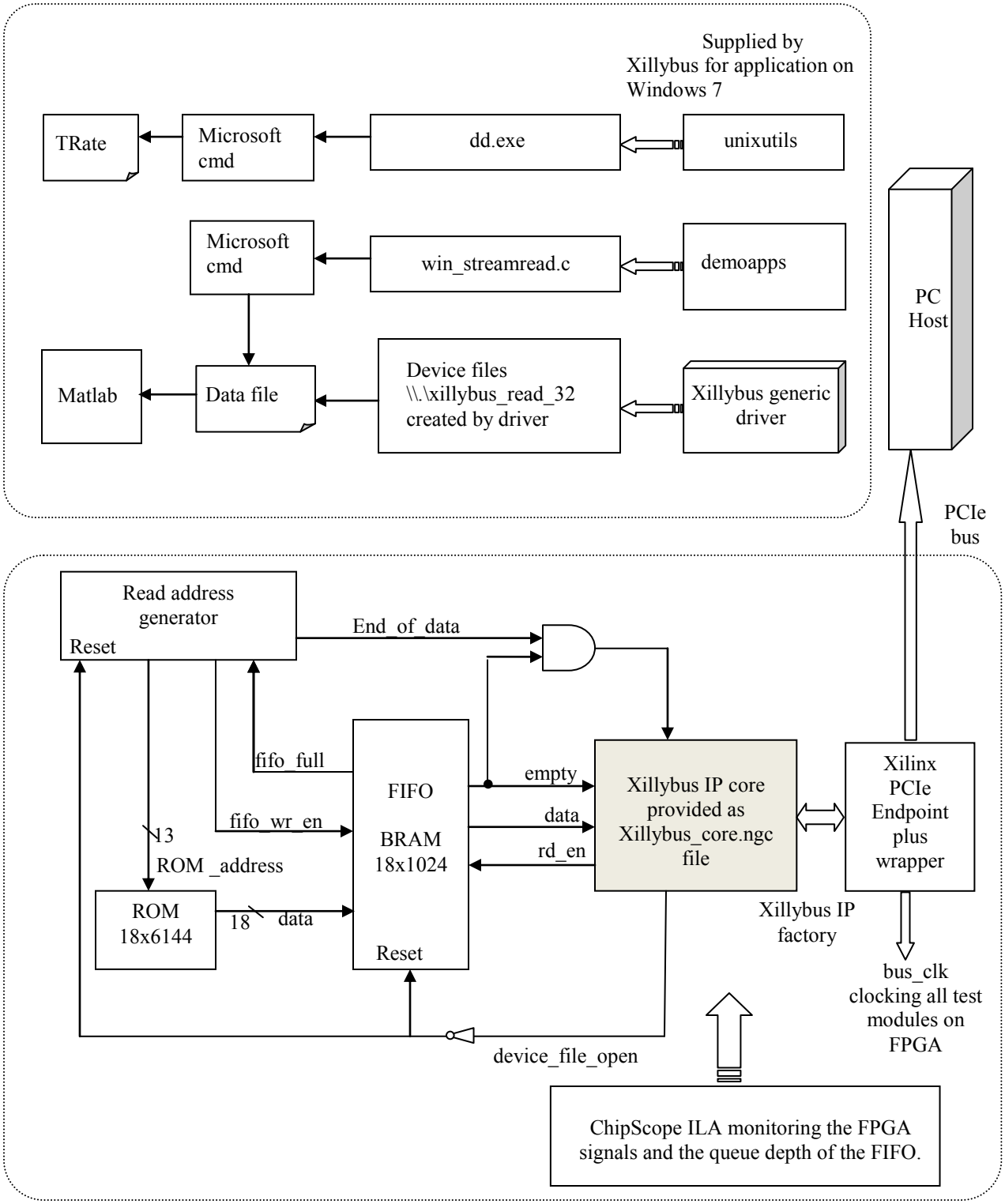


Fig. 4.4 Xillybus implementation and evaluation setting. The lower box includes the modules on the FPGA end. The upper block includes the software used on the PC host end to retrieve the transmitted data and measure the transmission rate.

4.3.2 On the host side:

For any Xillybus IP core configuration, the streams and their attributes are detected by the Xillybus as it is loaded into the host's operating system, and device files are created accordingly. In the testing setting designed for transmitting 18-bit words from FPGA to host, the data width option on the Xillybus IP core was 32 bits, and the corresponding port assignments and attributes were used, and the 14 most significant bits were set to 0. Correspondingly the driver creates the device file \\.\xillybus_read_32.

As sample host applications, Xillybus supplies C command line programs that were used in the evaluation setting. The application 'winstreamread.c' reads the streaming data from the device file and sends it to standard output. For proper operation, the translation mode was modified to binary mode, to suppress the LF (line feed 0A) character translation to CR-LF (carriage return-line feed combinations 0D 0A), that was observed in the data file.

Unfortunately the Xillybus driver does not offer any time stamping options to be able to track the exact transmission rate. It supplies a 'dd.exe' application file which copies data blocks from the device file and then indicates the corresponding transmission rate. For identifying the transmission overhead, sequential data was continuously read from the ROM at the same clock rate (bus_clk), used by the Xillybus IPcore and supplied by the PCIe Endpoint Plus Wrapper LogiCore of 100MHz. The internal signals were monitored using ChipScope and screenshots are shown in Fig 4.5a and Fig 4.5b to describe the transmission flow of data. The user_r_read_32_rden signal is set low when the PCIe is sending the overhead of the transmission layer packet (TLP), and during this internal, data is accumulated in the queue.

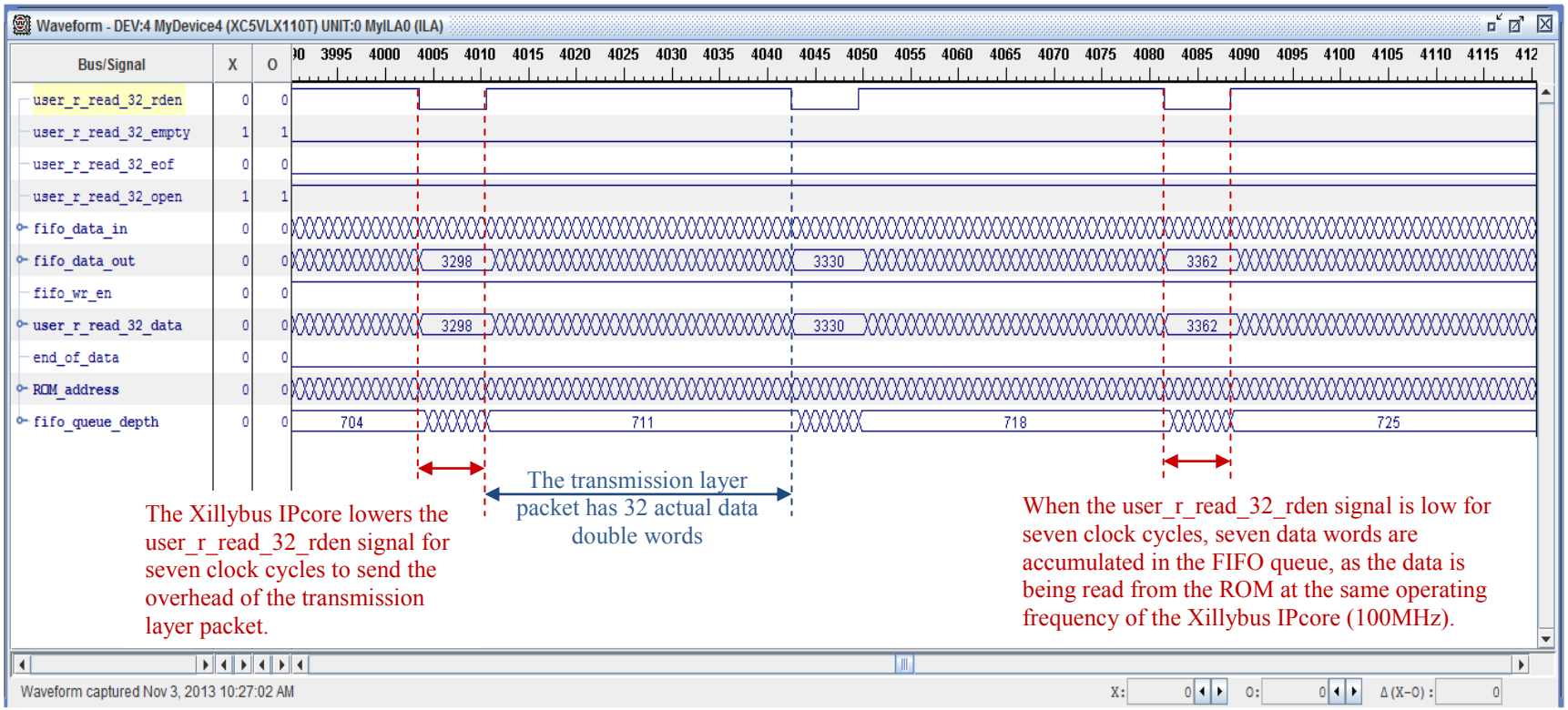


Fig. 4.5a The Transaction Layer packet (TLP) includes 32 double words of data, and an overhead of seven double words. During the transmission of the overhead, the Xillybus sets the read enable signal of the FIFO to '0', which caused the accumulation in the FIFO queue as shown in figure. Hence, the actual reading rate of data words is $(32/(32+7))\cdot 100\text{MHz} = 82\text{MHz}$. In case of a continuous data writing to the FIFO, accumulation can be prevented if the writing data rate is set to be equal to or less than the reading data rate.

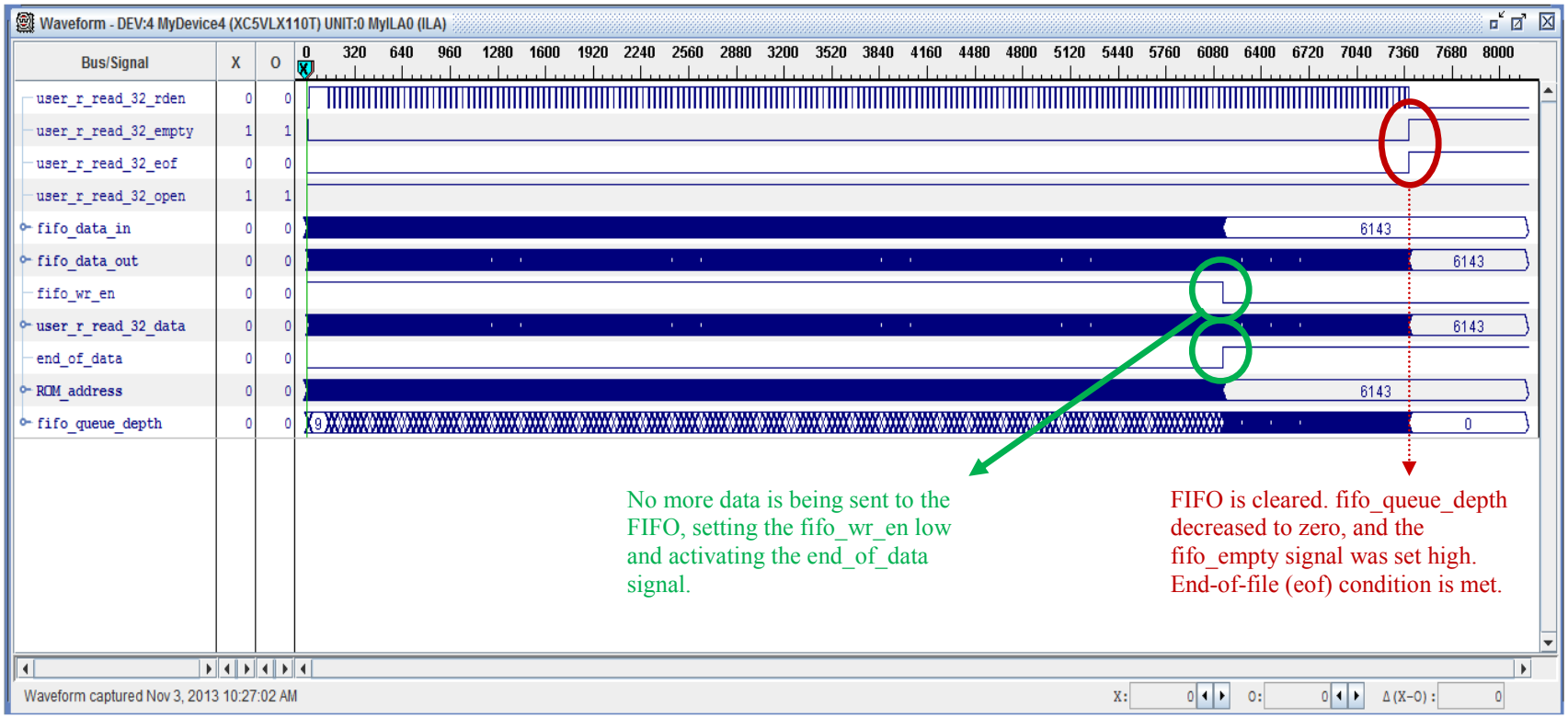


Fig. 4.5b When the application data source stops sending new data and sets the end-of-data signal high, the FIFO queue decreases gradually. When it is totally cleared and the FIFO empty signal is set high, the eof condition is met, as described in fig. 4.3, marking the end of the descriptor file sent to the PC. After reading the stream of data from the file descriptor, it was saved on a data file. The file was opened in MATLAB to check the transmission of the complete data set and the signal integrity.

4.4. Testing Data Transmission using Real Data Recordings:

A modeling of the data transmission process using real data recordings from 2550 channels at an approximate PCIe transmission rate was presented in Chapter 3. After testing the data transmission from FPGA to the host, using continuous incrementing counter data, it was desirable to evaluate the data transmission and queue depths needed when the system is handling real neuronal firing rates on hardware.

As the spike detection platform is not connected to real data acquisition system, the data was saved on the FPGA block memories. With a limit of 148 BRAMs of 36Kbits capacity each on the Virtex 5 FPGA XUPV5lx110t, a reduced version of the main design has been tested. The test focused on the transmission key players, which involve the queuing-based transfer of 48 samples for each detected spike to an output FIFO connected to Xillybus IPcore. It also examined the queue depths needed to prevent any spike dropping before transmission, while considering the reading cycles on hardware. The block diagram of the test setting is shown in Fig. 4.6.

Simulations were run on neuronal data recordings using MATLAB, as described in Chapter 3. The spike detection results were reduced to the spike times and the corresponding channel ID. The data was presorted based on the spike times first then the channel order based on the Time Division Multiplexing. The created data file was used to initialize a ROM on the FPGA, which served as the source of spike timing in the transmission test. As a numerical figure, for 88928 spikes detected in a 2.5 sec recording time, there was a need for 73 BRAMs to save the results on FPGA. A Time Frame Generator was used to determine when the spikes are sent to the transmission queue.

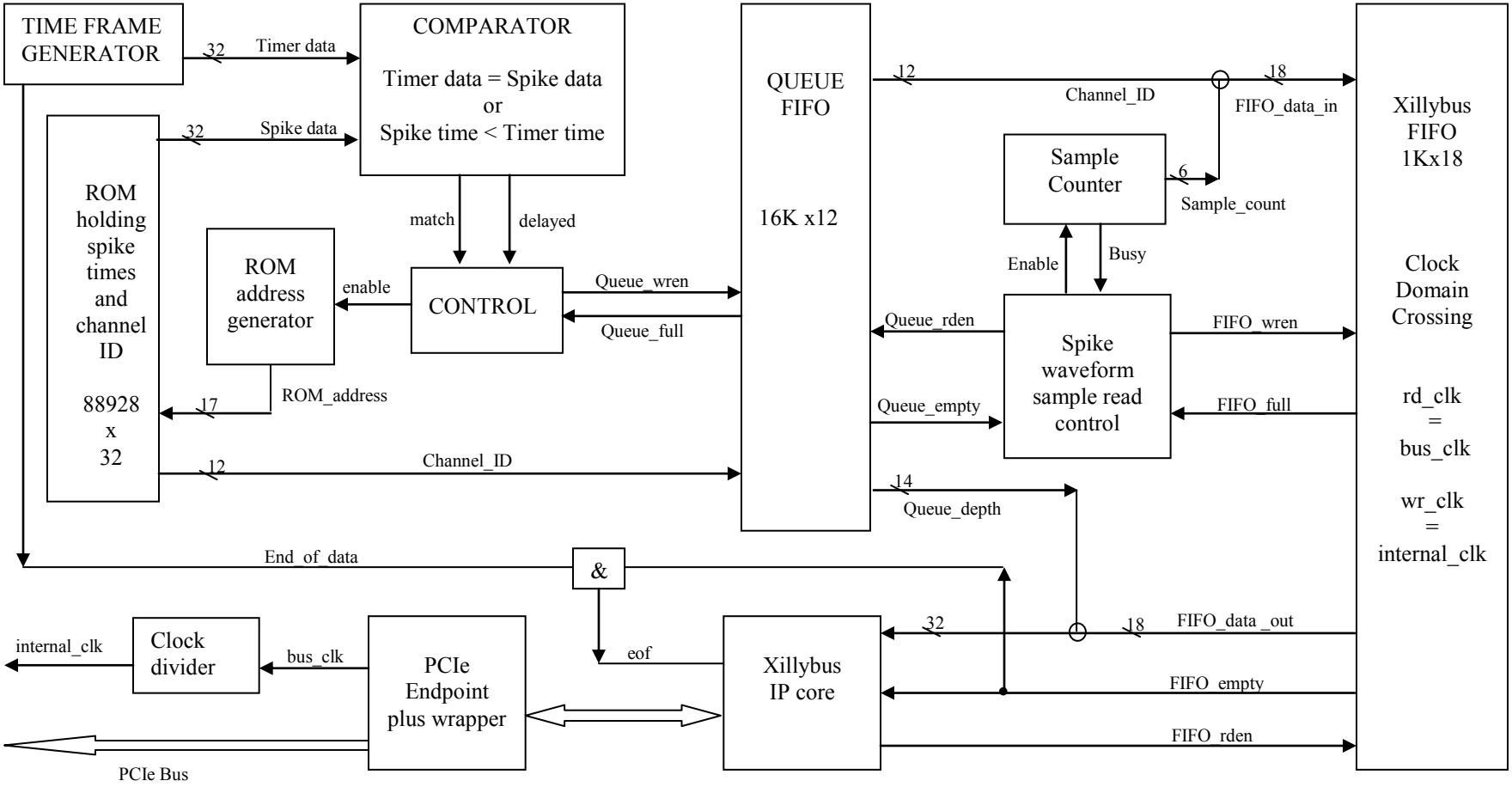


Fig. 4.6 A hardware design to test the data transmission of the detected spike wave shapes from the FPGA to the host PC based on spike timings obtained from real neuronal recordings.

When the time frame matches the saved spike timestamp and channel ID, the spike information is sent to the queue, and the ROM_address is incremented to read the next spike time. In case of synchronous firing, the timer may pass the next spike timestamp during the comparison and ROM reading cycles. If the time of the timer generator is greater than the spike time read on the ROM, the comparator activates a 'delayed' signal, and the controller sends the spike to the queue. The queue follows the temporal sequence of the detected spikes, and in the actual design, it holds the location of their waveforms in the output buffers of the spike detection units. In the reduced design used for testing, the 48 samples were generated using a sample counter, and they were concatenated with the channel_ID, and then sent to the output Xillybus FIFO. The FIFO input data is 18 bits long (12 bits for channel ID and 6 bits for sample order). The Xillybus IP core is designed to handle 32-bit words, so the 14 extra bits were used to send the queue-depth. The signals were monitored using ChipScope, and the data sent was evaluated using MATLAB.

The testing design incorporates two controllers: One manages the timely flow of spike data from the ROM to the queue, as explained above, and the second controller manages sending the spikes read from the queue to the Xillybus FIFO after attaching 48 samples to each spike. The spike waveform sample read controller was designed using a FSM as shown in Fig 4.7. The sequence of sending the spike samples to the output FIFO starts by enabling the sample counter. The counter increments gradually, to represent the 48 spike wave-shape samples. When started it activates a busy signal that is set back to low after completing the count. The controller sets the counter to a pause mode if the FIFO is full. When the busy signal is deactivated, the controller generates a queue read enable signal if the queue is not empty. The Xillybus IP core handles the reading control.

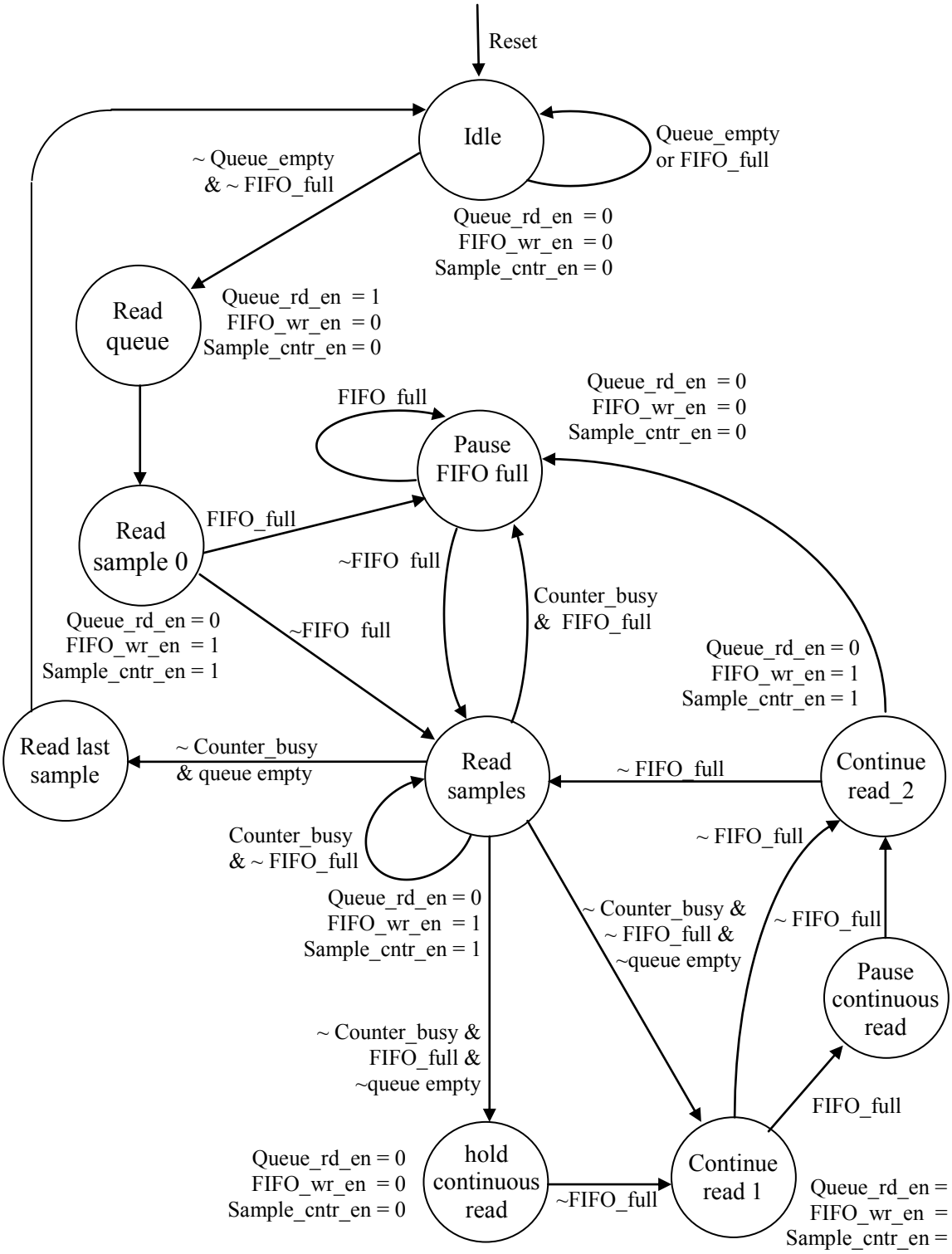


Fig. 4.7 A description of the read sample controller FSM in the reduced design testing the transmission rate of spike waveforms via PCIe from the FPGA to a host PC.

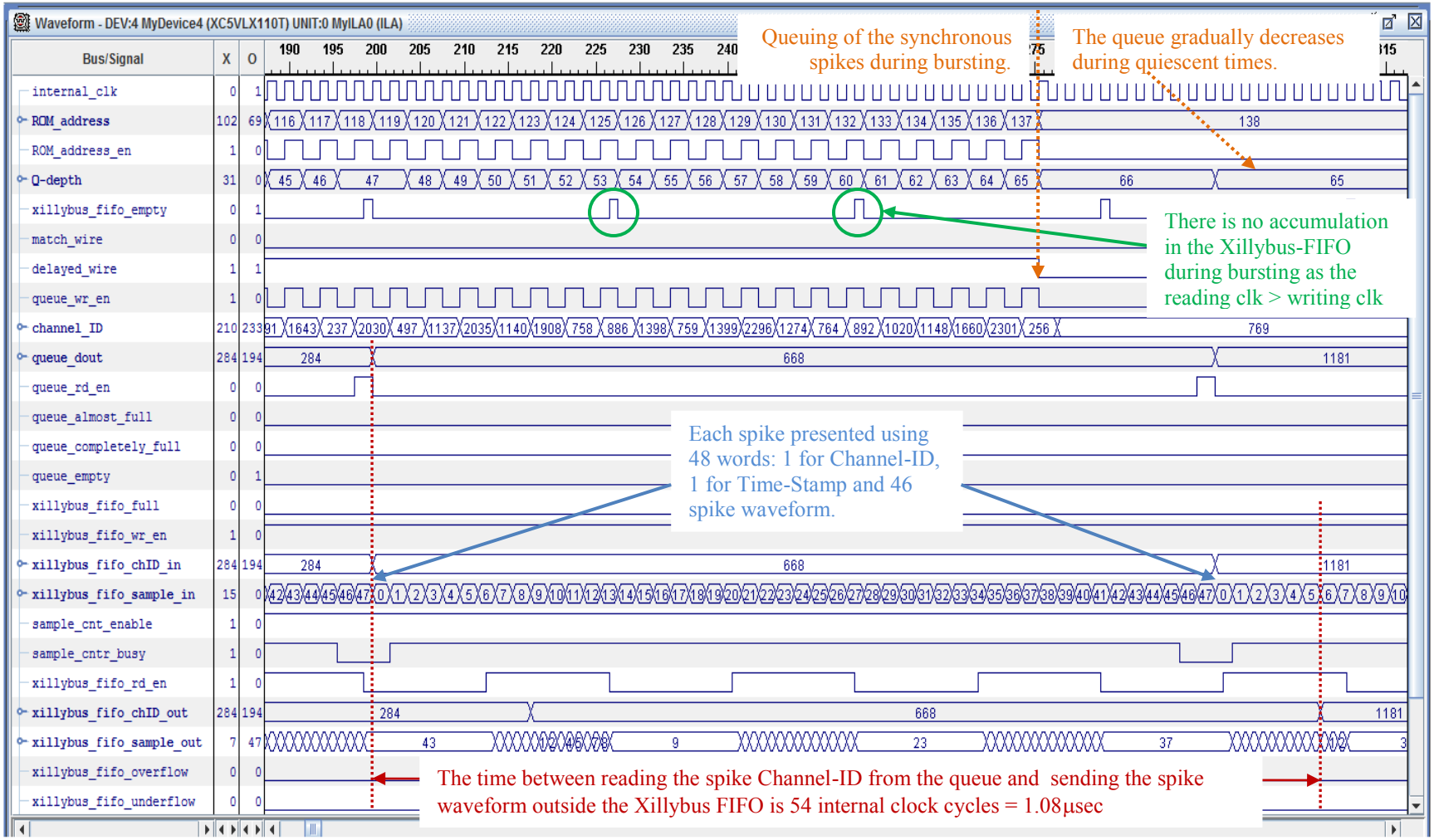


Fig. 4.8 Design verification was tested using signal monitoring in ChipScope®. ChipScope was used to examine correct sample alignments, and validate read and write cycles. The integrated logic analyzer was clocked by the bus-clk running at 100MHz. Internal clock was 50MHz. The time between reading the spike from the queue to sending the waveform outside the Xillybus FIFO is 54 internal clock cycles = 1.08µsec. No accumulation on the Xillybus FIFO.

4.4.1 Timing and Clocking:

Based on the timing summary generated by Xilinx® ISE Project Navigator, the maximum frequency, according to the critical path, is 65.811MHz. The 100 MHz clock provided by the PCI Express connector was connected directly to the Virtex-5 FPGA to clock the PCI Express Endpoint Block and PCI Express Endpoint Block Plus LogiCORE. The Xillybus IPcore and the reading clock of the Xillybus FIFO were supplied by the 100MHz clock denoted by bus_clk. Using a counter, an internal clock was generated, operating at 50MHz to regulate the rest of the design modules on the FPGA. The clock domain crossing was at the Xillybus FIFO, with the writing clock equal to 50MHz and the reading clock equal to 100MHz. In the complete design, if the TDM of for example 2500 channels, would be connected to the same internal clock of 50 MHz the design would allow a sampling frequency of 20 KHz per channel.

4.4.2 Device Utilization Summary:

The following is a table detailing the hardware usage to implement the transmission rate testing design. Table 4.1 utilization is based on the xupv5lx110t FPGA.

Slice Logic Utilization	***Used***	***Available***	***Utilization***
Number of slice registers	6,380	69,120	9%
Number of slice LUTs	5,708	69,120	8%
Number of occupied slices	2,896	17,280	16%
Number of Block-RAMs	144	148	97%
Number of bonded IOBs	9	640	1%
Number of BUFGs	6	32	18%
Number of GTP Duals	1	8	12%

Table 4.1 Device utilization summary to implement the transmission rate testing design for real neuronal data firing rates.

4.4.3 Data Used in the Test:

The data used in the test were recorded from dissociated rat hippocampal cells (2 days in vitro) using high-density MEA from 3Brain (www.3Brain.com). They have been supplied by the NetS3 Lab in the Neuroscience department of the *Instituto Italiano di Tecnologia* (IIT). The sampling frequency was 7.022 samples per second. The recording duration was 2.5 seconds. Total number of spikes detected across 2550 channels during the 2.5 sec recording time was 88928 spikes. Hence the Mean Firing Rate (MFR) was:

$$\text{MFR} = \frac{\text{Total \# of Spikes}}{\text{Recording Time}} = \frac{88928}{2.5} = 35{,}571.2 \text{ spikes/sec}$$

4.4.4 Queue Depth Implementation Results:

The queue depth signal was sent along with the spike data for testing purposes as shown in Fig. 4.6. The instantaneous queue-depths were extracted from the data words received at the host and are presented in Fig. 4.9. The maximum TR of the spike samples to the output FIFO is determined by the reading clock of the Xillybus-FIFO. As the internal frequency was set at 50MHz, the Xillybus FIFO can read 50 MSamples/sec. With 48 data words per spike, the internal queue can be cleared at a rate of 1042,667 spikes per sec.

$$\text{Queue reading rate} = \frac{50 \text{ MSamples/s}}{48 \text{ Samples/spike}} = 1042{,}667 \text{ spikes/s}$$

Applying the queue reading rate to the test data MFR and sampling frequency, the following can be concluded:

(1) The internal queue is read at a rate equal to 29.3 times the MFR.

(2) The time-stamp is based on the sampling frequency of the neural recording channels. According to the testing design, no spikes can be detected between successive time

stamps. With a sampling frequency of 7.022 KHz and a queue reading rate equal to 1042,667 spikes/sec, 148 spikes can be removed from the queue before any new spikes are added to it as shown in the inset in Fig.4.9. The maximum queue depth due to synchronized spikes was 184 spikes. Hence, the accumulation of spikes in the queue from one time-stamp to the next was limited to a few tens of spikes, if more than 148 spikes were detected at the same time-stamp. Removing 148 spikes from the queue means that 7104 words (148 spikes x 48 data-words/ spike) were read by the output FIFO. The difference between the two instances marked by the data-tips in the inset of Fig. 4.9 validates this statement.

The maximum queue depth in hardware implementation was 184 spikes as shown in Fig 4.9, while the maximum queue depth in the MATLAB model was 780 spikes. This difference was caused by the binning of spikes into 1msec intervals in the MATLAB model. The binning accumulated the spikes read across seven sampling periods (1msec bin/sampling period). The bin size choice of 1msec was relatively large with respect to the MFR of a few thousands of bursting neurons. A bin size equal to the sampling period (conforming to the time-stamp rate) is expected to match the hardware implementation results. Fig.9 is more dense than Fig.10 because of the fact that the quiescent intervals with clear empty queue were not monitored by the host as the PCIe transmission was idle during these times.

The MATLAB model, with a 1ms bin size, was run on a PC featuring an AMD Phenom™ II X6 1090T Processor 3.20GHz and a 64-bit operating system. The calculations of the queue-depth took approximately 12 hours to complete. The hardware implementation was much faster, taking less than a second.

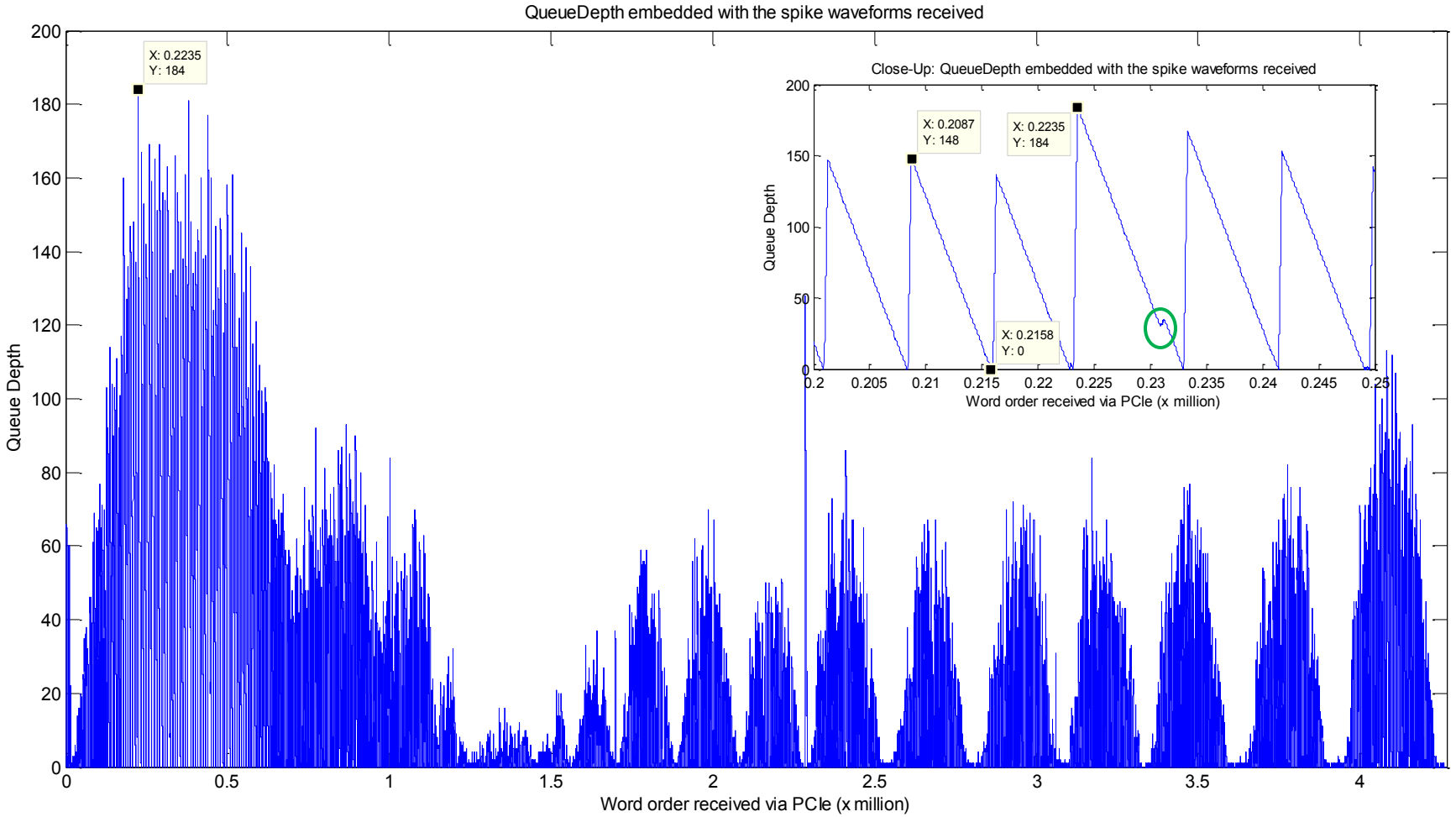


Fig. 4.9 The figure displays the queue depths after being extracted from the data sent via PCIe. For a total of 88,928 spikes, 4,268,544 (88,928 x 48) data words have been received. The inset shows how the design module clears 148 spikes from the queue between successive synchronized firing time-stamps. The recording sampling frequency was 7.022KHz.

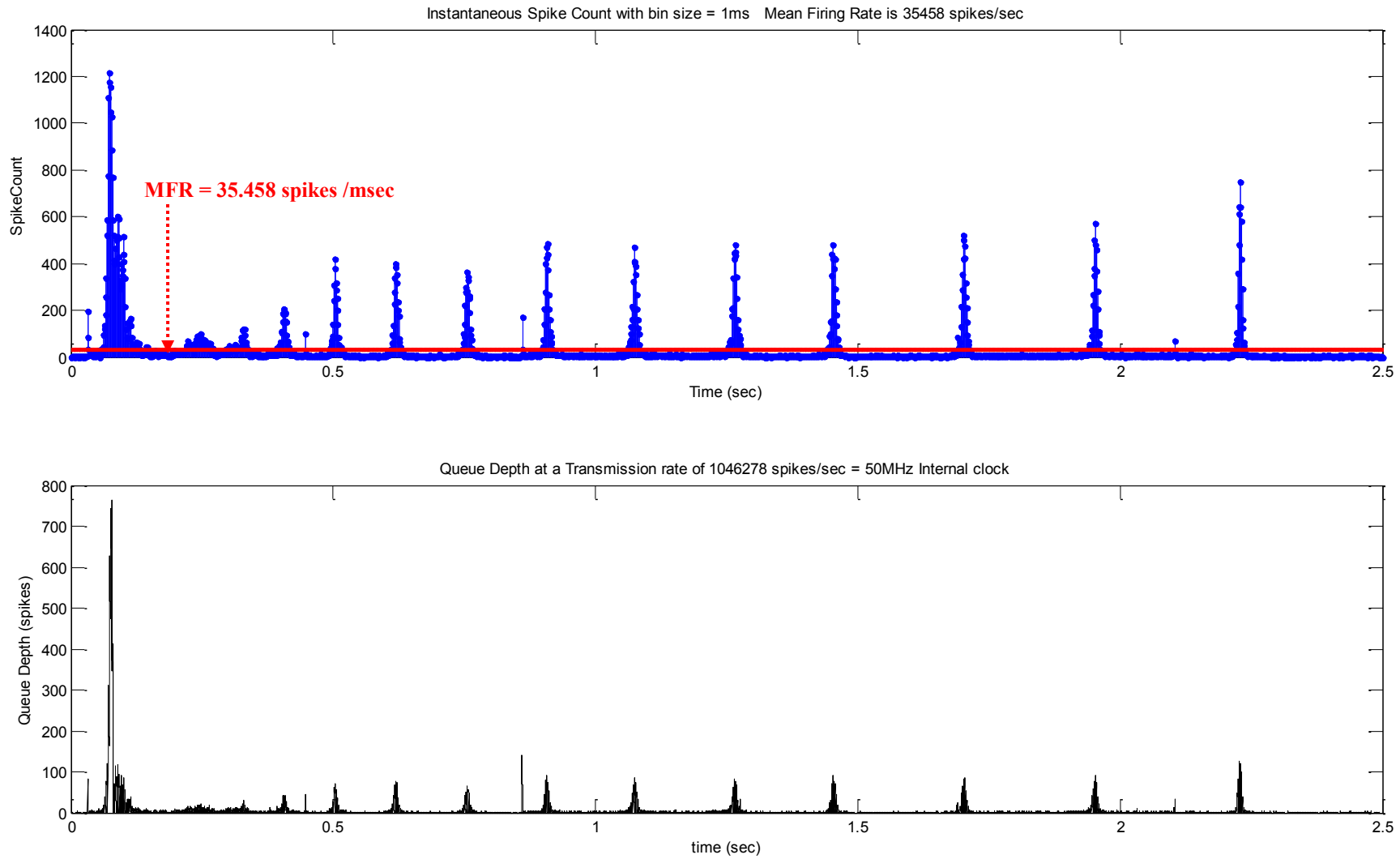


Fig. 4.10 The simulation results run in MATLAB on the same data recordings obtained from rat hippocampus dissociated neurons in vitro. The instantaneous queue depth was based on a bin size of 1msec, collecting spikes read across seven sampling periods of 0.1424 ms, in other words (1msec / 0.1424 ms) = 7 successive time-stamps.

4.5. Testing The Integration of Twenty Spike-Detection Units with PCIe Transmission:

In section 4.4 the PCIe data transmission was tested using real neuronal recordings from 2550 channels. The spike times were stored on BRAMs and a model was designed to mimic the Spike Detection Unit function. The design affiliated 48 words to every spike detected, and sent it via PCIe to the host PC. Testing the signal integrity and transmission operation on real neuronal data with typical bursting rates were the main objectives of the test. In this test, the main goal is to validate the design of integrating 20 spike-detection units and sending the spike detection waveforms using PCIe. The scheduling of data transmission between spikes detected across the 2560 channels, and the queue depths are monitored in this test.

The worst case scenario is having all channels recording synchronized spikes exactly at the same time stamp. Although this case was not witnessed in the real data recordings that were examined, this test serves to determine the capabilities of the system. The goal of the test was to test the functionality of the design as well as determine the maximum synchronous bursting rate that the system can handle before starting to skip spikes. The performance of the system is governed by the memory capacity and clock rates. It was also important to determine the queuing based delay in the described worst case scenario.

A block diagram of the complete test setting is shown in Fig. 4.11. The following sections present a description of each block.

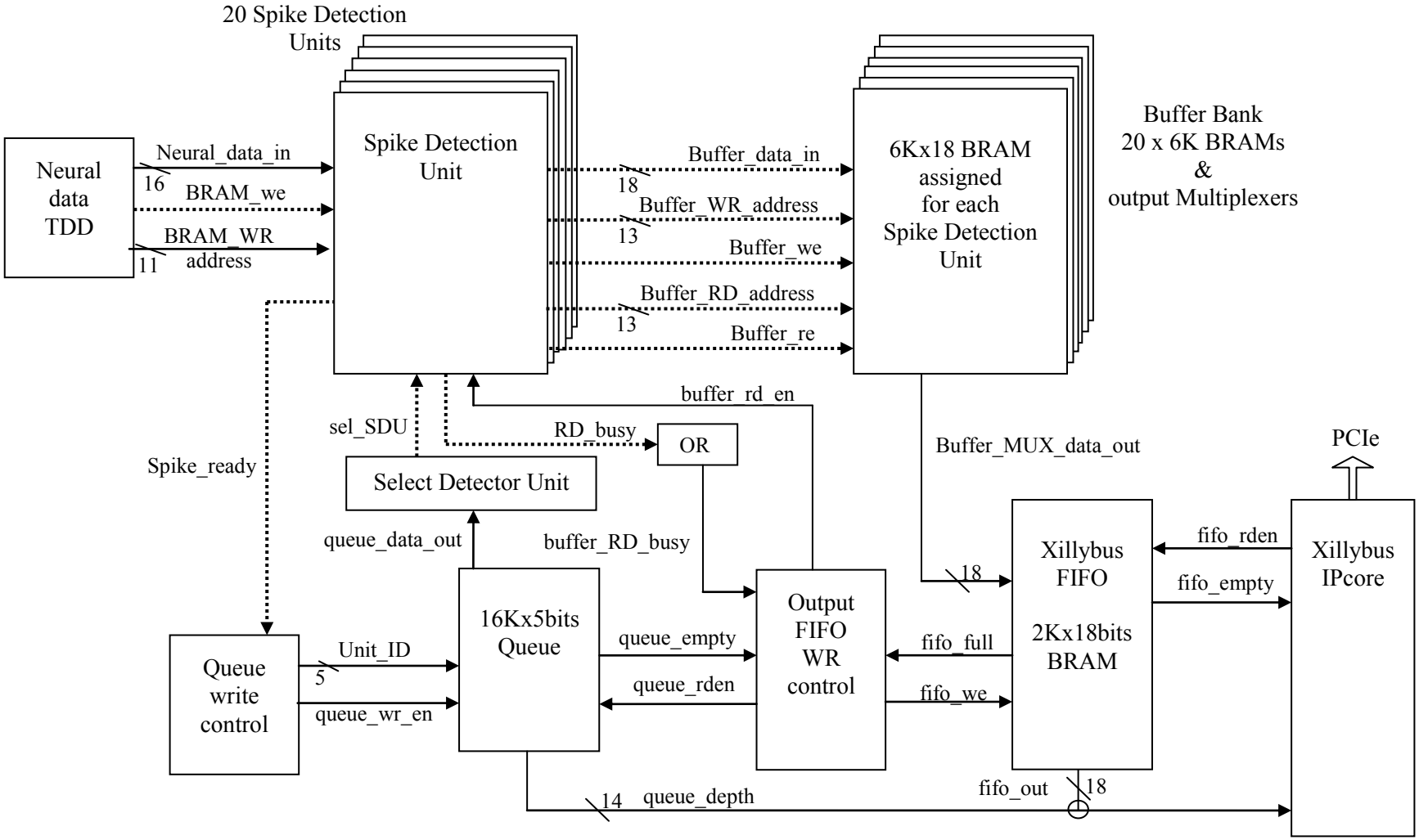


Fig. 4.11 Testing the integration of 20 Spike Detection Units (SDUs) on FPGA and using PCIe transmission to transfer detected spike waveforms to the host PC. The dotted arrows indicate that there are 20 replicates of similar internal signals each connected to one SDU. The solid arrows represent common or control signals.

4.5.1 Neural Data Time Division Demultiplexer:

The neural data TDD block has a ROM with stored neural data used for testing. It also has an address generator module that generates the time-stamp, the neural data ROM address, the channel ID, input BRAM_WR_address and BRAM_we. The virtual sampling rate Fs, at which the channels are updated with neural data is equal to the address generator clock F_{TDD} divided by the total number of channels N_{ch}.

The ROM has a short window of neural data containing only one spike that is being read in a cyclic mode. Controlling the width of the data window determines the firing rate of the signal. A window of N_{window} samples having one spike and being read at cyclic mode at a rate F_{TDD} has a firing rate FR of:

$$FR = \frac{F_s}{N_{window}} = \frac{F_{TDD}}{N_{ch} \cdot N_{window}} \tag{4.1}$$

For example if:

$$F_{TDD} = TDD\,clock = 50MHz$$
$$N_{ch} = Total\,number\,of\,channels = 2550$$
$$N_{window} = Number\,of\,data\,samples\,read\,in\,cyclic\,mode = 256$$

$$Virtual\ channel\ FR = \frac{F_{TDD}}{N_{ch} \cdot N_{window}} = \frac{50{,}000{,}000}{2560 \cdot 256} = 76.3\,spikes/sec$$

The block has a data multiplexer to control the input data to each SDU. In this test, the multiplexer supplies the SDUs by either the data stored on the ROM or a zero signal. The total synchronous bursting rate was controlled by the number of channels supplied by the ROM data. The worst case scenario is modeled by supplying the same neural data to all channels handled in the system.

4.5.2 Queue Write Control:

The system has a queue FIFO that saves the temporal sequence of the detected spikes to schedule reading the spike waveforms from the output buffer bank accordingly. When a SDU completes saving a spike waveform on the buffer, it set a "spike_ready" signal high. The queue write control block uses a FSM to scan the SDUs for any completed spike signals. If a "spike_ready" signal is set high, it writes the corresponding Unit_ID in the queue to schedule a timeslot for copying the completed spike waveform from the buffer affiliated with that SDU. The "spike_ready" signal was added to the SDU design described in Chapter 3 to serve the integration with multiple SDU and the data transmission to the host PC.

4.5.3 Spike Detection Unit:

The spike detection has three main operations in this test, namely:

(1) The spike detection using NEO operator

(2) Saving the spike waveform in the affiliated buffer

(3) Generating the Buffer_RD_address to send the data to the output Xillybus FIFO, when it is time to read from the SDU, following the queue schedule.

The first and second operation have been explained in Chapter 3. They are managed by a SDU_FSM. In this design a "spike_ready" signal was added to mark the completion of copying a spike waveform onto the buffer. As this test involves a cyclic repetition of one spike, the autonomous threshold selection was not implemented and instead a fixed threshold was predetermined and used in the design.

The Buffer_RD_address generator features a pointer that saves the address of the last word read from its buffer. The SDU is selected to send a spike waveform to the

output FIFO when its Unit_ID appears on the queue data output. When it is selected and the Output FIFO WR control activates the buffer_rd_en signal, the buffer_RD_address is gradually incremented until the 48 words of the spike waveform are copied to the FIFO.

While reading the spike waveform, the Buffer_RD_address generator sets a RD_busy signal high. This signal is connected to the Output FIFO WR Control module that manages the process of reading the spike locations from the queue. When the busy signal is activated by any of the SDUs, the queue_rden signal is set low.

4.5.4 The Queue:

The queue FIFO stores the ID of the SDU when it activates its "spike_ready" signal. The writing operation is managed by the "Queue Write Control" control module. The reading process is handled by the "Output FIFO WR control" block. The data output of the queue determines, which SDU is selected to transfer a spike waveform from its buffer to the output FIFO. The SDU keeps track of the last transmitted spike location on the buffer and hence this information does not need to be saved in the queue. The "Select Detector Unit" block activates the "sel_SDU" signal of the corresponding SDU defined by the queue data output.

The maximum queue depth was used to calculate the maximum transmission delay of the spikes to the Xillybus IPcore. The number of clock cycles that the system takes to copy a complete spike waveform to the Xillybus IPcore were determined using the ChipScope results. To monitor the queue depth, the data count on the FIFO was sent along with the spike waveform data via PCIe to the host.

The queue FIFO was created using Xilinx ISE core generator. The size of the queue was 16Kx5bits.

4.5.5 The Output FIFO WR Controller:

The output-FIFO-WR-controller manages copying the spike waveforms from the output buffer bank to the common output FIFO. It controls the reading process from the queue to decide which SDU should be enabled for a reading. If the queue is empty, or the system is busy reading a spike waveform from the buffer bank, or if the output FIFO is full, then the "queue_rden" signal is stays low. When the system is ready to read the next spike waveform, the "queue_rden" is activated for one clock cycle. Once the corresponding SDU is selected, the output-FIFO_WR-controller sets "buffer_rd_en" signal high to start the reading cycle of 48 spike waveform words. The output-FIFO-WR-controller manages the "output_FIFO_we" taking into account the reading and multiplexer delays between activating the buffer_rd_en signal and the data availability on the Buffer bank output.

4.5.6 The Buffer Bank:

The buffer bank has twenty 36K BRAM buffers, each assigned to one Spike Detection Unit. The outputs of the twenty buffers are connected via multiplexers to the buffer output. The "Buffer_MUX_data_out" is connected to the data input of the output buffer. The selection route of the multiplexers is determined by the Unit-ID that the queue outputs.

4.5.7 The Output FIFO:

The output FIFO has different reading and writing clocks. The writing clock is 100MHz supplied by the PCIe, and the writing clock is 50MHz. The slower clock was obtained by applying a counter supplied by the external 100MHz clock. The reading process from the output FIFO is controlled by the Xillybus IPcore as shown in Fig. 4.12.

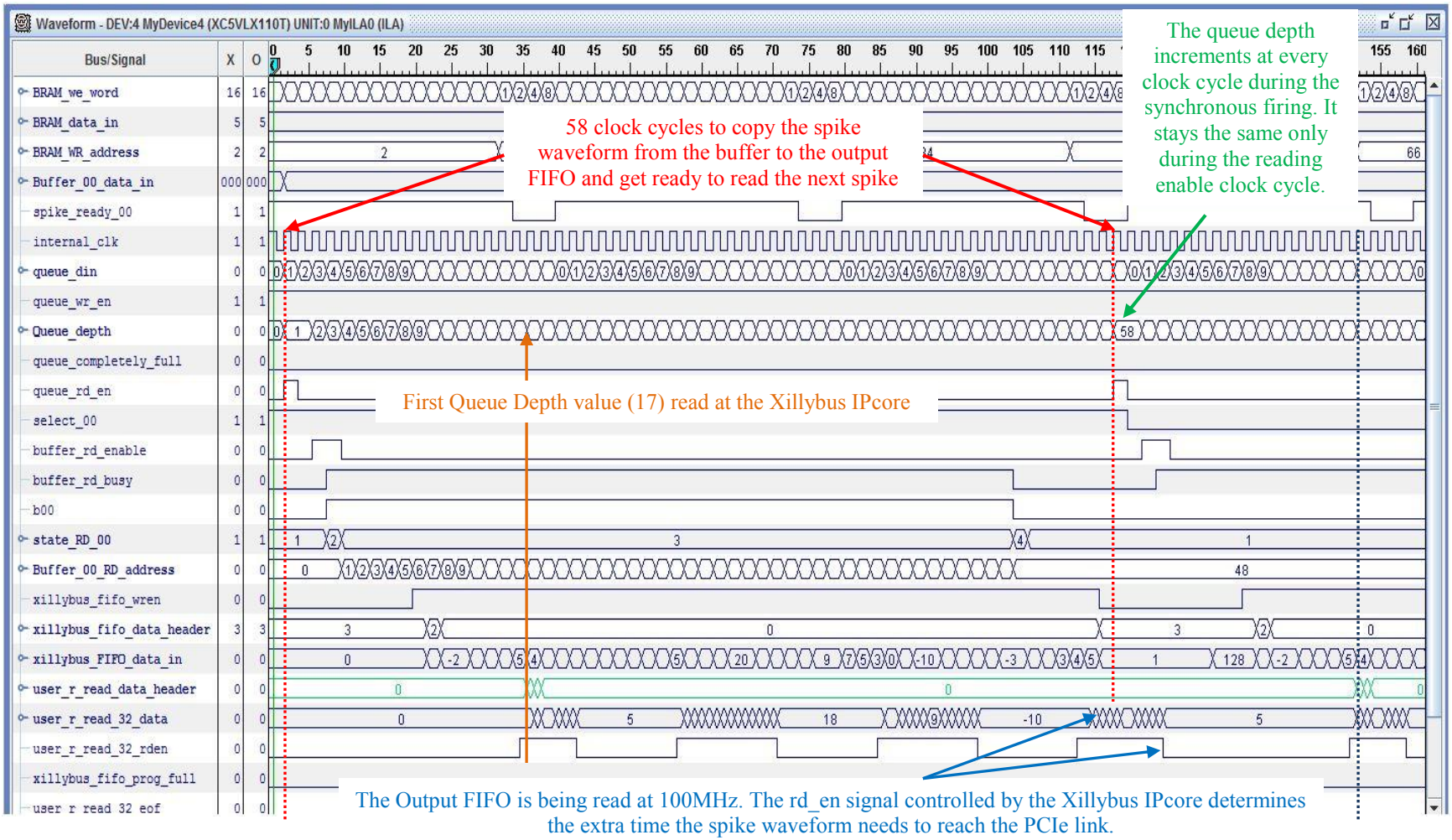


Fig 4.12. A screen shot from ChipScope, describing the copying process of spike waveforms from the buffer bank of SDU_00 onto the output Xillybus FIFO, to the Xillybus IPcore and then to the PCIe link.

4.5.8 Test Results:

Monitoring the signals using ChipScope, it was observed that 58 internal clock cycles are needed to copy a complete spike waveform from the buffer bank into the output FIFO and get ready to read the next AP waveform. In other words the time between two successive queue read enable signals is equal to 58 clock cycles.

Each buffer associated with a spike detection unit can hold up to 128 spike waveforms at a time, counting for the case when all channel s have detected spikes at the same time stamp. When a spike is detected a 48-word block of memory is reserved in the buffer, and the samples of the refractory period are copied to the buffer as they arrive to the input BRAM one by one. Hence in case of the perfect synchronous firing over all the channels, all the spikes need to be transmitted to the output FIFO before the next synchronized event occurs to prevent the dropping of any data. Assuming that the output FIFO will not be full at any of the transmission intervals, the maximum theoretical FR per channel that the system design can handle is:

$$\text{Maximum synchronous FR} = \frac{50\ \text{MHz clock}}{58\ \text{clock cycles} \cdot 2560\ \text{channels}} \approx 336\ \text{spikes/sec per channel}$$

The testing spike had 256 samples that were read in a cyclic mode, simulating a synchronous firing rate of ~76 spikes/sec per channel. The ChipScope screen shot in Fig.4.13 graphs the queue depth sent along with the spike data to the host. The figure shows that the queue is completely emptied before new spikes are detected on the Spike detection unit. The maximum delay, as shown in Fig.4.13 and Fig. 4.14, would be:

$$\text{Max\_latency} = 2515 \cdot 58 \cdot \frac{1}{50\text{MHz}} \approx 3\text{ms}$$

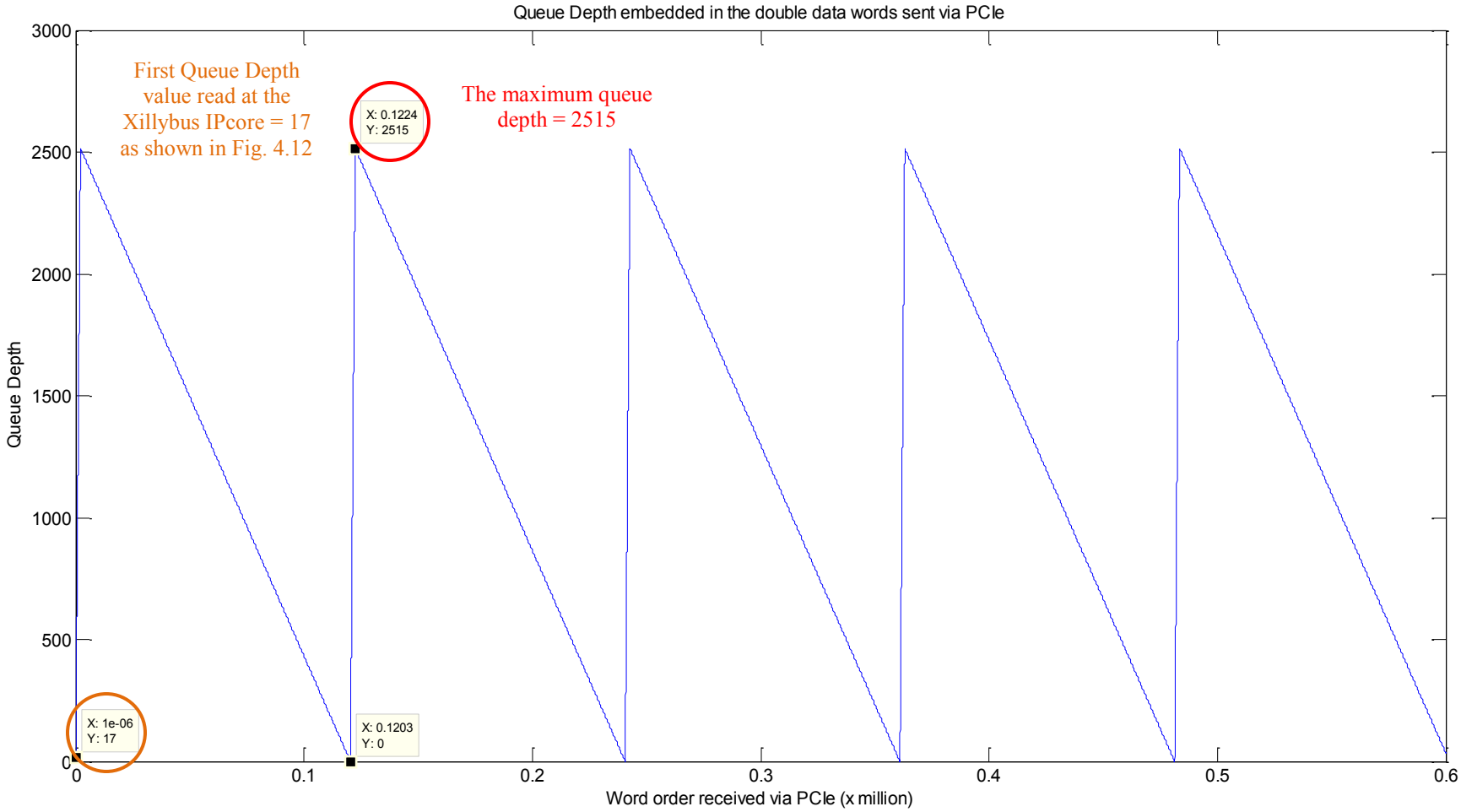


Fig. 4.13 In this test synchronous spike detection over all the channels was modeled, hence the queue depth is incrementing at every internal clock cycle during the spike time stamp. The design module can read a new spike waveform every 58 clock cycles. While a total number of 2560 channels are "reserving a turn" in the queue, the first 45 (~ 2560/58 + 1) spike waveforms are copied to the output buffer. That is why the maximum queue depth is 2515 (2560-45).

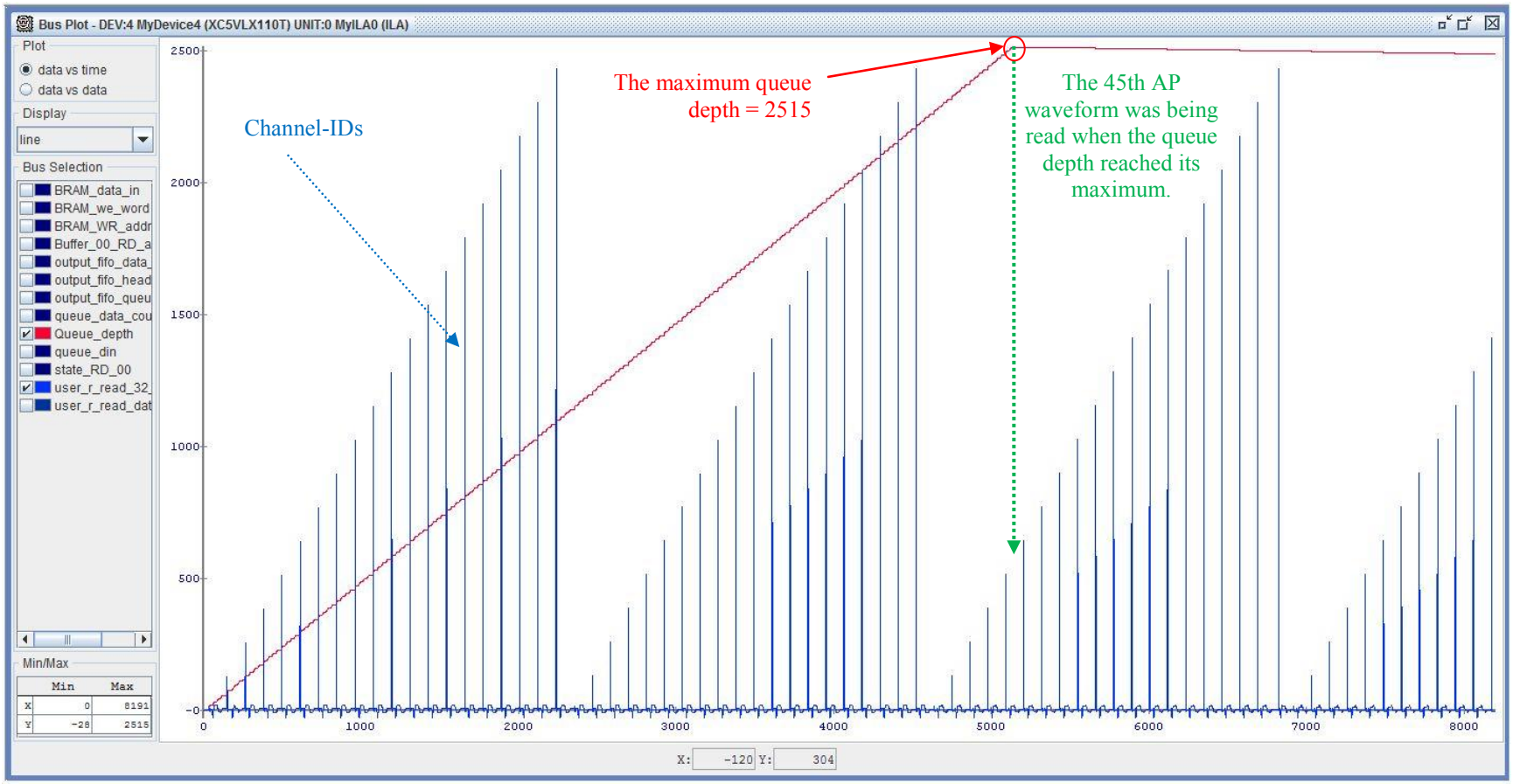


Fig. 4.14 ChipScope bus plot of the 32-bit data words transmitted to the Xillybus IPcore. The higher 14 bits represent the queue depth while the lower 18 bits have a two bit header and 16 bits of either AP waveform data, a time stamp or channel ID.

Table 4.1 and table 4.2 present the device utilization and timing summary respectively:

Slice Logic Utilization	Used	Available	Utilization
Number of slice registers	13,076	69,120	18%
Number of BRAM/FIFO	136	148	91%
Total memory used in KB	4,878	5,328	91%
Number of DSP48Es	20	64	31%
Number of GTP_Duals	1	8	12%
Number of PCIEs	1	1	100%
Number of PLL_ADVs	1	6	16%
Number of BFUGs	13	32	40%

Table 4.1 Device utilization summary for the hardware implementation of twenty spike detection units integrated with PCIe transmission.

Minimum period	13.332 ns
Maximum frequency	75.007 MHz
Maximum path delay from/to any node	3.315 ns

Table 4.2 Timing summary of the same design setting.

CHAPTER 5

DISCUSSION

The research presented in this dissertation was motivated by a long term goal of monitoring the electrical activity of thousands of neurons, in an effort to decipher the brain activity. Recording thousands of neural signals may provide some insight in what Santiago Ramón y Cajal, the father of modern neuroscience, called "the impenetrable jungle where many investigators have lost themselves." Monitoring the dynamic signals of an enormous number of neurons is a breakthrough that might bridge the gap between the firing of neurons and motion, perception or even decision making. Increasing the number of recording channels is a common demand among different research areas. The development of reliable BMI with multiple degrees of freedom, to help paralyzed patients and amputees restore their independent mobility, requires monitoring the firing patterns of hundreds or even thousands of neurons. Decoding the exact patterns of brain dynamics that underlie thinking and behavior will provide essential insight into what happens when neural circuitry malfunctions in neural and psychiatric disorders. In vitro neuronal network research also requires a high density MEA data acquisition to enable studying the correlation between the static and dynamic maps of the neurons.

The real-time neural signal processing will be an essential requirement for any system dealing with a massive number of recording channels, even if it is not a closed loop system. Even systems with offline data analysis will require at least real-time data reduction to limit the data storage needs.

Increasing the number of recording channels carries many challenges at every step of the neural processing pathway from data acquisition to data analysis. The work presented in this dissertation has attempted to find solutions to some of the problems related to designing a real-time neuronal data reduction platform that can handle a few thousands of recording channels. Along the research work, more questions were raised uncovering areas of further future work potentials. The hardware architecture designs developed can serve as a testing platform for new approaches to process neuronal signals.

5.1. Integration of the Platform with a Data Acquisition System:

One of the major questions that were investigated in the dissertation work was how to handle the massive input data that is expected to result from an augmentation in the number of recording channels. The application of the Multi-Gigabit transceivers (MGTs) was suggested to get the neural data into and out of the FPGA as fast as the device can process it. Simple solutions were suggested for the alignment of data words as well as the reassignment of input data to their respective channel IDs. The comma detection and comma alignment circuits of the MGT were applied. The next research step would be examining the system interface to Analog to Digital Converters that present the final stage of a neural signal acquisition system. Starting in 2006, JEDEC introduced a series of standards allowing ADCs to connect to SerDes interfaces on FPGAs. The latest version JESD204B released in 2012 features a high maximum lane rate (up to 12.5 Gbps per channel), support for deterministic latency, and support for harmonic frame clocking. The series of standards have set a common language between fast high performance ADC

and FPGAs making use of the high bandwidths SerDes can provide. Analog Devices has lately released an ADC with high-speed serial interfaces, the AD9250 dual, 14-bit, 250MSPS ADC supporting the JESD204B standard.

Theoretically speaking, this ADC can handle 10,000 recording channels sampled at 25 KSPS. Examining the practical implementation of this ADC to a neural data acquisition system and possible switching circuits that can multiplex different channels to the same ADC at this rate is a potential future research point.

5.2 The Autonomous Design Architecture:

One of the motivations of the dissertation work was to design an autonomous spike-based data reduction system, that is fully controlled by FSMs. No processors were used in the system control in order to avoid interrupt latencies that may degrade the performance of the overall design. FSM controllers were designed to handle different parts of the design, namely:

(a) The input data allocation between multiple spike detection units,

(b) The spike detection unit control and the copying process of the spike waveforms from the input BRAM into the output buffers.

(c) Autonomous threshold selection for the spike detection unit.

(d) Managing the transmission of the AP waveforms from the unit buffers to the output FIFO shared by all the units.

(e) A test-bed for the transmission of real neuronal data.

The design architecture and FSM designs can be implemented to test new neural signal processing approaches. As a proof of concept, the spike detector used threshold

comparison using the NEO operator. This is a classical approach that has been used for a long time in neuronal spike detection. The architecture design can implement other spike detection techniques such as the discrete wavelet transform. [40,41,46]

5.3 The hardware implementation:

The hardware designs presented in the dissertation work were implemented for evaluation and proof of concept on a Xilinx XUPV5-LX110t board. Virtex-7 FPGAs are expected to have lower utilization percentages, and faster speed. This will allow giving more room for design expansion to handle more channels. The design BRAM utilization showed the highest percentage of 91%. The complete design used a total of 136 x 36K BRAMs ~ 5Mb. The Virtex 7 FPGA families integrate on average 68Mb BRAMs. A rough estimate can conclude that the hardware design described in the dissertation can be replicated ~ 13 times to handle a total of more than 33 thousand channels. A definite channel count value cannot be given before synthesizing the design and running the placement and routing to ensure that the timing constraints will be met. The Virtex7 FPGA integrates 96 MGTs, each working at 28.05 Gbps. Considering the integration of 33 thousand channels and each channel recording neural signals at 30 KHz, then a total bandwidth of 990 MSPS and at a sample precision of 16 bits/Sample, the input bandwidth requirement is 15.84 Gbps. Hence it is not expected that the input data transmission would be a factor of design limitation. The design bottleneck will be the transmission through PCIe to a host PC. Further reduction will be needed to decrease the output data for example by implementing spike sorting in hardware as well [47]. With more DSP slices integrated, the implementation of more complex spike detection and

spike sorting algorithms will be feasible. Concrete values can only be determined when the design is implemented on hardware. This is another implementation project to be considered in future work.

Recently, Xilinx has released the Zynq®-7000 family. A series of products based on the Xilinx All Programmable System-on-Chip architecture, that integrates a dual-core ARM® Cortex™-A9 based processing system and 28 nm Xilinx programmable logic in a single device. Implementing the spike-based data reduction platform on the Zync FPGA may allow adding more features to the design capabilities.

5.4 PCI Express Transmission:

The PCIe transmission using Xillybus IPcore was relatively a straightforward solution for the transmission from the FPGA to the host PC. The Xillybus IPcore provides the necessary DMA-based design and the software driver to handle the data reception at the host. It was convenient for observing the data processed by the hardware design and evaluating it. On the other hand, the p-core is available as a bit-file with little insight into the internal design and limited flexibility for custom modifications.

One of the main drawbacks of using the Xillybus IPcore was the fact that it is hard to predict the behavior of the read-enable signal. It is controlled by many factors that are opaque to the user. Some of the factors may be: The operation of the PCIe core on FPGA, the response of the host to interrupts and the motherboard's packet switching. At some points of the transmission process it was observed that the read enable signal of the Xillybus IPcore was idle for longer intervals of time (across a ChipScope window of

8192 clock cycles at 100MHz) causing accumulation of the data in the output buffer and consequently in the queue.

5.5 Design Parameters:

The number of channels that can be handled by the spike-based data reduction platform depends on several parameters related to the hardware resources, the processing clock on the FPGA, the type of neurons, and the transmission link bandwidth to the host PC. Based on the design simulations and hardware implementation, the following formula summarizes the design parameters to define the scaling boundaries of the system.

5.5.1 Memory Usage:

There are three buffering stations in the design, namely the input buffer, the spike detection unit intermediate output buffer, and the common output FIFO where the spike waveforms from all the spike detection units are queued to be transmitted to the host PC. The input buffer has 16 sample words of 16 bits each assigned for each channel. The intermediate output buffer has 48 word block of 18 bits per word for each spike detected waveform. The number of 48 word blocks is equal to the number of channels to account for the worst case scenario of full bursting synchronization across all channels. The queue depth of the output FIFO depends on the transmission rate to the host PC.

Memory usage per channel for the first two buffering stations is:

Input buffer = 16x16 = 256 bits

Intermediate buffer = 48x18 = 864 bits

The total memory required for buffering one channel at the first two stages is 1120 bits.

To assure that no spike waveforms will be dropped, the system must be able to copy the spike waveforms to the common output FIFO before the intermediate buffers are filled up again with new spike waveforms. In other words, the maximum bursting rate may not be greater than the rate at which the spike waveforms are copied to the output FIFO.

$$\text{Maximum bursting Rate} = \frac{\text{Internal clock}}{58\ \text{clock cycles} \cdot \cdot \#\ \text{spike waveform blocks in the intermediate buffer}}$$

In order to keep the memory usage per channel as described above, the internal clock must be adjusted adequately to complement the maximum bursting rate of the neuronal culture recorded.

5.5.2. Transmission Rate and Queue Depth:

An approximation formula was derived in section 3.5.6 to relate the transmission rate to the queue depth, bursting rate and number of samples per spike.

$$\begin{aligned} \text{SpkAcc} &= (\text{Average BFR - TR}) \cdot \tau_{\text{burst}} \\ \text{Queue Depth} &= \text{SpkAcc} \cdot \text{samples / spike} \\ \text{AverageBFR} &= \text{Average bursting Rate per channel} \cdot \ \#\text{of channels} \end{aligned}$$

where SpkAcc = spike accumulation in the queue, Average BFR is the Firing rate during bursting activities, TR is the transmission rate to the host PC and τ-burst is the bursting time. Knowing the type of neurons that will be monitored, the average bursting firing rate can be estimated. Based on the transmission rate and the hardware memory resources, the number of channels can be determined. Or knowing the number of channels, the transmission rate and the queue depth can be designed. There is a number of algorithms

that have been developed to accurately detect burst occurrences and durations both in vivo and in vitro [48-50].

References

[1] Deco G, Jirsa VK, Robinson PA, Breakspear M, Friston K. The dynamic brain: From spiking neurons to neural masses and cortical fields. PLoS Comput Biol. 2008 Aug 29;4(8):e1000092.

[2] Alivisatos AP, Chun M, Church GM, Greenspan RJ, Roukes ML, Yuste R. The brain activity map project and the challenge of functional connectomics. Neuron. 2012 Jun 21;74(6):970-4.

[3] Lebedev MA, Nicolelis MA. Brain-machine interfaces: Past, present and future. Trends Neurosci. 2006 Sep;29(9):536-46.

[4] Stevenson IH, Kording KP. How advances in neural recording affect data analysis. Nat Neurosci. 2011 Feb;14(2):139-42.

[5] Nicolelis MA, Dimitrov D, Carmena JM, Crist R, Lehew G, Kralik JD, et al. Chronic, multisite, multielectrode recordings in macaque monkeys. Proc Natl Acad Sci U S A. 2003 Sep 16;100(19):11041-6

[6] Maccione A, Gandolfo M, Tedesco M, Nieus T, Imfeld K, Martinoia S, et al. Experimental investigation on spontaneously active hippocampal cultures recorded by means of high-density MEAs: Analysis of the spatial resolution effects. Front Neuroeng. 2010 May 10;3:4.

[7] Elaraby, N.; Obeid, I. "A Model Design of a 2560-Channel Spike Detection Platform" IEEE ReConFig 2012

[8] Potter SM. Distributed processing in cultured neuronal networks. Prog Brain Res. 2001;130:49-62.

[9] Buzsaki G. Large-scale recording of neuronal ensembles. Nat Neurosci. 2004 May;7(5):446-51.

[10] Olsson RH,3rd, Buhl DL, Sirota AM, Buzsaki G, Wise KD. Band-tunable and multiplexed integrated circuits for simultaneous recording and stimulation with microelectrode arrays. IEEE Trans Biomed Eng. 2005 Jul;52(7):1303-11.

[11] Gross GW, Rieske E, Kreutzberg GW, Meyer A. A new fixed-array multi-microelectrode system designed for long-term monitoring of extracellular single unit neuronal activity in vitro. Neurosci Lett. 1977 Nov;6(2-3):101-5.

[12] Eversmann B, Jenkner M, Hofmann F, Paulus C, Brederlow R, Holzapfl B, et al. A 128 × 128 CMOS biosensor array for extracellular recording of neural activity. Solid-State Circuits, IEEE Journal of. 2003;38(12):2306-17.

[13] Litke AM, Bezayiff N, Chichilnisky EJ, Cunningham W, Dabrowski W, Grillo AA, et al. What does the eye tell the brain?: Development of a system for the large-scale recording of retinal output activity. Nuclear Science, IEEE Transactions on. 2004;51(4):1434-40.

[14] Johnson LJ, Cohen E, Ilg D, Klein R, Skeath P, Scribner DA. A novel high electrode count spike recording array using an 81,920 pixel transimpedance amplifier-based imaging chip. J Neurosci Methods. 2012 4/15;205(2):223-32.

[15] Najafi K, Wise KD. An implantable multielectrode array with on-chip signal processing. Solid-State Circuits, IEEE Journal of. 1986;21(6):1035-44.

[16] Frey U, Sedivy J, Heer F, Pedron R, Ballini M, Mueller J, et al. Switch-matrix-based high-density microelectrode array in CMOS technology. Solid-State Circuits, IEEE Journal of. 2010;45(2):467-82.

[17] Imfeld K, Garenne A, Neukom S, Maccione A, Martinoia S, Koudelka-Hep M, et al. High-resolution MEA platform for in-vitro electrogenic cell networks imaging. Engineering in medicine and biology society, 2007. EMBS 2007. 29th annual international conference of the IEEE; ; 2007.

[18] Nicolelis MA, Ghazanfar AA, Faggin BM, Votaw S, Oliveira LM. Reconstructing the engram: Simultaneous, multisite, many single neuron recordings. Neuron. 1997 Apr;18(4):529-37.

[19] Najafi K, Wise KD, Mochizuki T. A high-yield IC-compatible multichannel recording array. Electron Devices, IEEE Transactions on. 1985;32(7):1206-11.

[20] Wise KD, Sodagar AM, Ying Yao, Gulari MN, Perlin GE, Najafi K. Microelectrodes, microelectronics, and implantable neural microsystems. Proceedings of the IEEE. 2008;96(7):1184-202.

[21] Du J, Riedel-Kruse IH, Nawroth JC, Roukes ML, Laurent G, Masmanidis SC. High-resolution three-dimensional extracellular recording of neuronal activity with microfabricated electrode arrays. J Neurophysiol. 2009 Mar;101(3):1671-8.

[22] Perlin GE, Wise KD. The effect of the substrate on the extracellular neural activity recorded micromachined silicon microprobes. Engineering in medicine and biology society, 2004. IEMBS '04. 26th annual international conference of the IEEE; ; 2004.

[23] Obeid I, Wolf PD. Evaluation of spike-detection algorithms fora brain-machine interface application. Biomedical Engineering, IEEE Transactions on. 2004;51(6):905-11.

[24] Bossetti CA, Carmena JM, Nicolelis MAL, Wolf PD. Transmission latencies in a telemetry-linked brain-machine interface. Biomedical Engineering, IEEE Transactions on. 2004;51(6):919-24.

[25] Obeid, I. A wireless multichannel neural recording platform for real time brain machine interface. [Ph.D dissertation]. Durham: Duke University; 2004.

[26] Balasubramanian, K. Reconfigurable system-on-chip architecture for neural signal processing [Ph.Ddissertation]. Philadelphia: Temple University; 2011

[27] Zhang F, Aghagolzadeh M, Oweiss K. A fully implantable, programmable and multimodal neuroprocessor for wireless, cortically controlled brain-machine interface applications. J Signal Process Syst. 2012 Dec 1;69(3):351-61.

[28] Jake Wiltgen and John Ayer. Bus Master DMA Performance Demonstration Reference Design for the Xilinx Endpoint PCI Express Solutions.Xilinx, December 2009. XAPP1052 (v2.5).

[29] JEDEC Standard No. 204A (JESD204A) Serial Interface for Data Converters http://www.jedec.org/download/search/JESD204A.pdf

[30] Marc Defossez Virtex-5 FPGA Interface to a JESD204A Compliant ADC XAPP876 (v1.0.1) February 22, 2010

[31] http://www.analog.com/static/imported-files/tech_articles/JESD204B-Survival-Guide.pdf

[32] Jake Wiltgen and John Ayer, xapp1052 September 2011, “Bus Master Performance Demonstration Reference Design for the Xilinx Endpoint PCI express Solutions”.

[33] Alex Goldhammer and John Ayer, WP350 September 2008, "Understanding the Performance of PCI express systems"

[34] Virtex-5 FPGA Integrated Endpoint Block for PCI express Designs, User Guide 2009

[35] *www.xillybus.com*

[36] Y. Chagnac-Amitai and B. W. Connors. Synchronized excitation and inhibition driven by intrinsically bursting neurons in neocortex. *J. Neurophysiol. 62(5),* pp. 1149-1162. 1989.

[37] J. M. Beggs and D. Plenz. Neuronal avalanches in neocortical circuits. *J. Neurosci. 23(35),* pp. 11167-11177. 2003. . DOI: 23/35/11167 [pii].

[38] C. L. Stephens, H. Toda, T. D. Palmer, T. B. DeMarse and B. K. Ormerod. Adult neural progenitor cells reactivate superbursting in mature neural networks. *Exp. Neurol. 234(1),* pp. 20-30. 2012. . DOI: http://dx.doi.org/10.1016/j.expneurol.2011.12.009.

[39] D. C. Cooper. The significance of action potential bursting in the brain reward circuit. *Neurochem. Int. 41(5),* pp. 333-340. 2002. . DOI: http://dx.doi.org/10.1016/S0197-0186(02)00068-2.

[40] K. G. Oweiss, A. Mason, Y. Suhail, A. M. Kamboh and K. E. Thomson. A scalable wavelet transform VLSI architecture for real-time signal processing in high-density intra-cortical implants. *Circuits and Systems I: Regular Papers, IEEE Transactions on 54(6),* pp. 1266-1278. 2007.

[41] Imfeld, K., Maccione, A., Gandolfo, M., Martinoia, S., Farine, P.-A., Koudelka-Hep, M. and Berdondini, L. (2009), Real-time signal processing for high-density microelectrode array systems. Int. J. Adapt. Control Signal Process., 23: 983–998. doi: 10.1002/acs.1077

[42] B. W. Connors and M. J. Gutnick. Intrinsic firing patterns of diverse neocortical neurons. *Trends Neurosci. 13(3),* pp. 99-104. 1990. . DOI: 0166-2236(90)90185-D [pii].

[43] J. P. Donoghue. Connecting cortex to machines: Recent advances in brain interfaces. *Nature Neuroscience JID - 9809671* 1204.

[44] E. M. Izhikevich. Simple model of spiking neurons. *Neural Networks, IEEE Transactions on 14(6),* pp. 1569-1572. 2003. . DOI: 10.1109/TNN.2003.820440.

[46] S. Kim and J. McNames. Automatic spike detection based on adaptive template matching for extracellular neural recordings. *J. Neurosci. Methods 165(2),* pp. 165-174. 2007.

[47] K. G. Oweiss. A systems approach for data compression and latency reduction in cortically controlled brain machine interfaces. *Biomedical Engineering, IEEE Transactions on 53(7),* pp. 1364-1377. 2006. . DOI: 10.1109/TBME.2006.873749.

[48] M. Aghagozadeh, A. Mohebi and K. Oweiss. Sorting and tracking neuronal spikes via simple thresholding. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on PP(99),* pp. 1-1. 2014. . DOI: 10.1109/TNSRE.2013.2289918.

[49] D.J. Bakkum, M. Radivojevic and H.Takahashi. Parameters for burst detection. Frontiers in Computational Neuroscience 2013

[50] Legendy C. R., Salcman M.. Bursts and recurrences of bursts in the spike trains of spontaneously active striate cortex neurons. Journal of Neurophysiology1985. 53, 926-939.

[51] Cocatre-Zilgien J.H., Delcomyn F. Identification of bursts in spike trains. Journal of Neuroscience Methods 41, 19-30.10.1016/0165-0270(92)9012-3. 1992

[52] A. Athavale, and C. Christensen. High-Speed Serial I/O Made Simple. A Designer's Guide with FPGA Applications. April 2005. http://www.xilinx.com/publications/archives/books/serialio.pdf

[53] Stephen H. Scott. Neuroscience: Converting thoughts into actions. *Nature* 442, 141-142 (13 July 2006) | doi:10.1038/442141a; Published online 12 July 2006

[54] Ifft PJ, Shokur S, Li Z, Lebedev MA, Nicolelis MA. A brain-machine interface enables bimanual arm movements in monkeys. Sci Transl Med. 2013 Nov 6;5(210):210ra154. doi: 10.1126/scitranslmed.3006159.

[55] Hochberg LR[1], Serruya MD, Friehs GM, Mukand JA, Saleh M, Caplan AH, Branner A, Chen D, Penn RD, Donoghue JP. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. Nature. 2006 Jul 13;442(7099):164-71.

[56] Hochberg LR, Bacher D, Jarosiewicz B, Masse NY, Simeral JD, Vogel J, Haddadin S, Liu J, Cash SS, van der Smagt P, et al. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. Nature. 2012 May 16; 485(7398):372-5

[57] Napoli A. Xie J, Obeid I. Understanding the temporal evolution of neuronal connectivity in cultured networks using statistical analysis. BMC Neuroscience. 2014Jan21;15:17. doi: 10.1186/1471-2202-15-17.

[58] Pasquale V, Martinoia S, Chiappalone M. A self-adapting approach for the detection of bursts and network bursts in neuronal cultures. J Comput Neurosci. 2010 Aug 28