Proceedings of the EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services, Budapest, 2001.

A Multichannel Data Acquisition and Analysis System based on off-the-shelf DSP Boards

Andre Folkers, Ulrich G. Hofmann
Institute for Signalprocessing
Medical University of Lübeck
Seelandstr. 1a, Geb. 5, 23569 Lübeck, Germany
Phone: +49 (451) 3909 556, Fax: +49 (451) 3909 555, Email: {folkers, hofmann}@isip.mu-luebeck.de

Abstract— **The EU-funded project VSAMUEL aims to develop a versatile system for advanced neuronal recordings with multisite microelectrodes. Within this project we are developing a data acquisition system for high channel counts. The system will be able to acquire and process data on 128 channels at a sampling rate of at least 32 kHz. We are utilizing multiple DSP boards to perform that task. Other project parts include development of multisite microelectrodes and respective pre- and main amplifier stages. Besides data acquisition the DSP boards are used to perform online analysis of the data, e.g. spike detection and spectrogram computation. Other project parts include development of multisite microelectrodes and respective pre- and main amplifier stages.**

Keywords— **Digital Signal Processors, Multichannel Data Acquisition**

I. Introduction

The principal goal of the project VSAMUEL sponsored by the European Commission is to provide a versatile data acquisition system to the neuroscience community which is able to acquire simultaneous recordings from hundreds of cells in the nervous system. The project includes the development of silicon based microelectrodes, amplifier stages, and data acquisition software and hardware, where the latter part is where we turn our attention in this paper. The data acquisition hardware is based on off-the-shelf DSP-boards in combination with a personal computer running under Windows NT. The software provides basic functionality like acquiring data on 32 channels, and streaming this raw data to disk. Furthermore, online visualization of a high channel count, and spectrogram of single channels is implemented.

II. Microelectrodes and Amplifier

The next two sections give a short description of the microelectrodes and the amplifier stages.

A. Mircoelectrodes

A key issue in understanding the nervous system is to make simultaneous observations of the activity of a large number of cells. Therefore, it is desired to have a large number of recording sites inserted in the neural tissue, while the insertion causes as smallest possible tissue damage. The design and construction of the microelectrodes is done by ACREO AB (Sweden) with respect to these constraints.

The microelectrodes are of fork shaped silicon with shafts of width 25 μm having a pointed end. The recording sites (Ir, 10 μm $\times$ 10 μm) are arranged in a two dimensional array (Fig. 1) at the front end of the shafts. They are connected with the contact pads (Au) by fine and narrowly spaced metal (Au) conductor traces. There are several designs, which differ in shaft length (4 mm to 15 mm) and shaft count (1 to 4), and also in recording site spacing (see also [1, 2]).

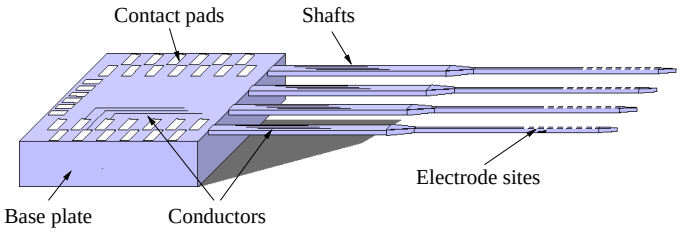

Fig. 1. Sketch of Microelectrode

B. Amplifiers

The amplifier rack consists of a precision low-noise pre-amplifier with gain factor of up to 20 and main amplifier providing gain factors of up to 2000. Both pre- and main-amplifier are newly designed by Thomas RECORDING (Germany) for 64 channels. The amplifier rack gain settings are controlled by a micro controller, and can be setup either manually or by software via a PC-serial port. All channels can have an individual gain settings. The electrical connection of microelectrodes and pre-amplifier is realized with a "zero insertion force" connector on the pre-amplifier side and a several centimeters long flexible PC-board at the probe site (Fig. 2).

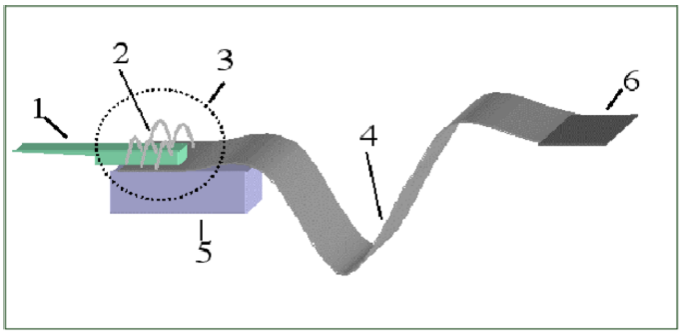

Fig. 2. Schematic drawing of a microelectrode (1), wire bonded (2) to a flexible PC-board (4) on a carrier (5). The bonded probe is epoxy sealed (3) and may be connected via the stiffened flex-board (6) to a zero insertion force connector on the pre-amplifier

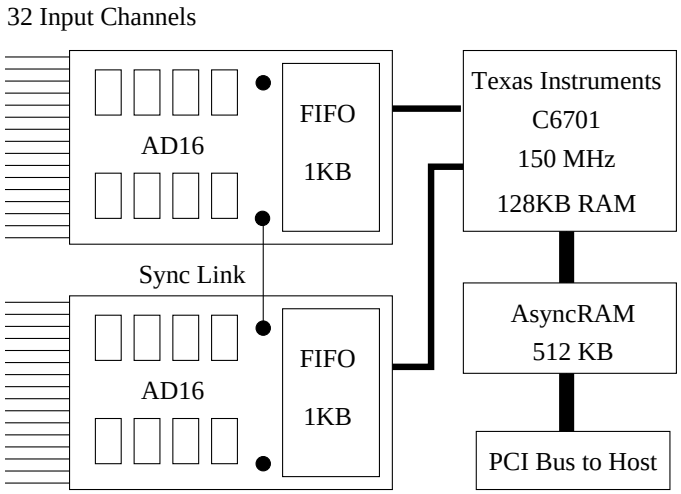

Fig. 3. Hardware arrangement and data flow of one DSP board

III. Data Acquisition Hardware

Our hardware consists of four DSP boards (M67, Innovative Integration, Thousand Oaks, CA, USA) which are combined with analog digital converter (ADC) modules (AD16 Omnibus modules, II). The M67 board has a single digital signal processor, i.e. a TMS3206701 processor (Texas Instruments, Dallas, TX, USA) clocked at 150 MHz with 128KB onchip memory. The CPU is provided with three different types of external memory: asynchronous SRAM (ASRAM, 512KB), synchronous DRAM (SDRAM, 16 MB), and synchronous burst SRAM (SBSRAM, 1MB). Each DSP board is equipped with two AD16 modules. Fig. 3 shows the arrangement of the hardware components and the data flow for one DSP board.

The AD16 module provides the M67 board with 16 channels of high speed 195 kHz, 16-bit resolution analog input to digital output conversion (A/D) per module site. There are 16 A/D converters for simultaneous conversion on all channels. Each of the 16 input channel consists of a high precision, DC accurate sigma-delta A/D converter (AD7722, Analog Devices, Norwood, MA, USA) with front end conditioning circuitry, which removes the need for multiplexers. The A/D converters are clocked either using a DDS timer of the M67 board or an external clock. Due to the 64 times oversampling performed by the A/D converters the clock rate is 64 times higher than the actual sampling rate. Conversion results are transferred into a FIFO which can store up to 512 16-bit samples. The AD16 triggers an interrupt when the FIFO contains a certain amount of samples. Usually this threshold is set to half of the samples which can be stored in total. This interrupt is serviced by a routine running on the DSP which fetches the data from the FIFO and stores it into the onchip memory using a DMA transfer.

A. Synchronization

Multiple AD16 modules may be synchronized by linking the synchronization signals between modules by a cable. Sync input and output signals allow software sync commands from one AD16 module to be shared with other AD16 modules on the same M67 board and also across multiple M67 boards. This allows a single sync command to synchronize multiple AD16 modules and causes simultaneous sampling across the converters present on those modules. The A/D converters of modules residing on the same board are exactly synchronized. If the AD16 modules reside on different M67 boards and do not use the same A/D clock, then the synchronization is not quite perfect. They are only synchronized within one A/D clock cycle. The length of such a cycle is quite small with e.g. 521 ns at a sampling rate of 30 kHz. This effect may be eliminated by providing the same clock to all AD16 module, e.g. by using a synchronization link between the M67 boards or an external clock source.

IV. Data Acquisition Software

The software can be divided into three parts, i.e. the program which runs on the DSP, the DSP application, a data acquisition server (DAQ server), and a data acquisition client (DAQ client). The DSP application performs the raw data acquisition, the transfer to the data acquisition server, and executes different online or off-line analysis modules, like for example compression, filtering, and spike sorting. The DAQ server provides a general interface to the DSP program for the DAQ clients. A user interacts with the system through the DAQ client. It configures experiments and executes them using the DAQ server. Results are sent back to the DAQ client which in turn visualizes them appropriately.

We are developing the software using Texas Instrument's "Code Composer Studio 1.20" and Borland's "C++ Builder 5.0".

A. DSP Application

The DSP application is organized in four parts, i.e. A/D conversion module, processing module, transfer module, and a control task. Fig. 4 illustrates how the different parts interact. The control task can send and receive messages to and from the DAQ server. A message consists of a receiver field, a command field, and a data field. The control task routes the messages with regard to the receiver field toward one of the other modules, or it handles the com-

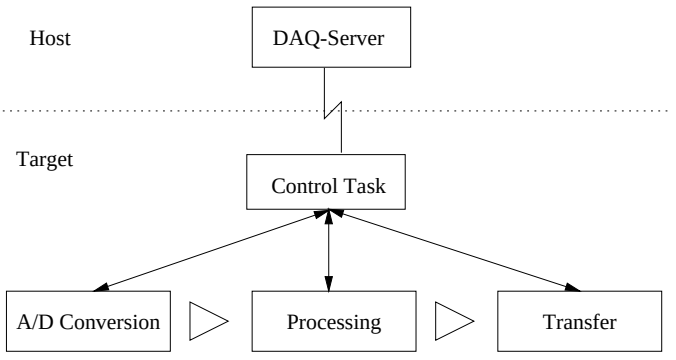

Fig. 4. Structure of the DSP application

mand directly if it is the receiver. The messages are used to configure the modules, e.g. setting the sampling rate in the A/D conversion module. A message can also request a certain parameter from a module. In this case the module sends a message back via the control task containing the requested parameter.

The data acquisition, processing and transfer is driven by interrupt events. The first interrupts in the cascade are triggered by the AD16 modules if their FIFO fill level reaches a certain threshold. Each AD16 module triggers on its own interrupt line. The FIFOs are independently read with DMA transfers into a frame in the onchip memory. Hereby the data is interleaved, such that the samples from all channels taken at the same time build a single block. The number and size of frames used is variable, but there must be at least two frames where each can hold the samples from the two FIFOs. This in turn depends on the FIFO fill level threshold of the FIFOs. The maximum number and size of frames is constrained by the size of the onchip memory.

After a frame has been filled it is put into the processing queue and a message is sent to the processing task which initiates the desired processing for this frame. During processing the raw data can either be replaced by the results or certain channels can be extracted into a separate buffer. For example given the case that only a certain frequency band is of interest raw data would be replaced by the result of the respective filtering. On the other hand if a spectrogram of a certain channel is requested a copy of this channel is created from the raw data and if enough sample points are collected the FFT is performed. After the processing is completed the frame is transferred into ASRAM and another interrupt is triggered to initiate the transfer via the PCI bus to the DAQ server.

The transfer from the DSP application to the DAQ server over the PCI bus is organized in packets. Each packet consists of a header and a data part. The header indicates which kind of data the packet contains in the data part. In order to keep the communication simple and fast the packets have a fixed length. One packet contains only one type of data, e.g. either raw data or Fourier coefficients not both. The packets are written into shared memory (shared between M67 board and Host PC), which is divided into two blocks. Each block can hold half of the packets which fit into the shared memory in total. The DAQ server is notified each time a block is filled with packets. This keeps the event rate for the DAQ server as low as possible.

B. Data Acquisition Server

The DAQ server provides services of the DSP application, like data acquisition or analysis, and other services like data retrieval to the DAQ client. For example the DAQ client creates an experiment object initialized with the experimental parameters and sends it to the DAQ server, which in turn uses it to configure the data acquisition hardware. The DAQ server has an entry queue for each type of data that can be sent from the DSP application. These queues are connected to queue transmitters. A queue transmitter has a list of references to queue receivers. Fig. 5 illustrates the relation of the respective classes. The association MainQueue refers to the entry queue of a certain type of data. The content of this queue is distributed to the queue receivers within the method Transmit() which copies a chunk from the queue into a buffer and calls the method Receive() of every queue receiver passing this buffer. The transmit method is periodically called by the DAQ server. A queue receiver can decimate the data which is passed to the received method according to the setting of block input frequency and block output frequency. This is e.g. used to keep the data volume for a visualization module below a specific rate. A visualization module would set the block output frequency to the number of sample points which can be displayed in one second. This feature of a queue receiver can also be turned off, e.g. streaming to disk is done without any decimation.

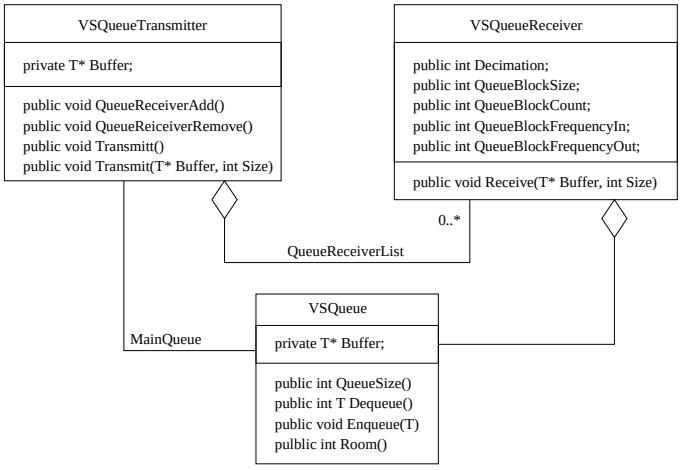

Fig. 5. Class diagram of queue receiver and queue transmitter

The data which needs to be stored is streamed to disk by the DAQ server. Only the data needed for visualization is sent to the DAQ client.

C. Data Acquisition Client

The DAQ client consists of dialogs to setup and control an experiment and several visualization components. Data acquisition and replay is controlled by a panel having buttons like a DAT player and recorder. Currently under development is an experiment planer which schedule is processed automatically. The following section shortly introduce the visualization components which include a virtual scope, a blueplot, and a spectrogram.

C.1 Virtual Scope

The virtual scope (Fig. 6) can be used to inspect the details of certain channels. Currently it supports up to eight traces. Each trace can have a different scaling, offset, color, and line width. The assignment of a recording site to a trace can be done either by name, or by selecting the recording site in a schematic drawing of the probe. The scope supports arbitrary zooming.

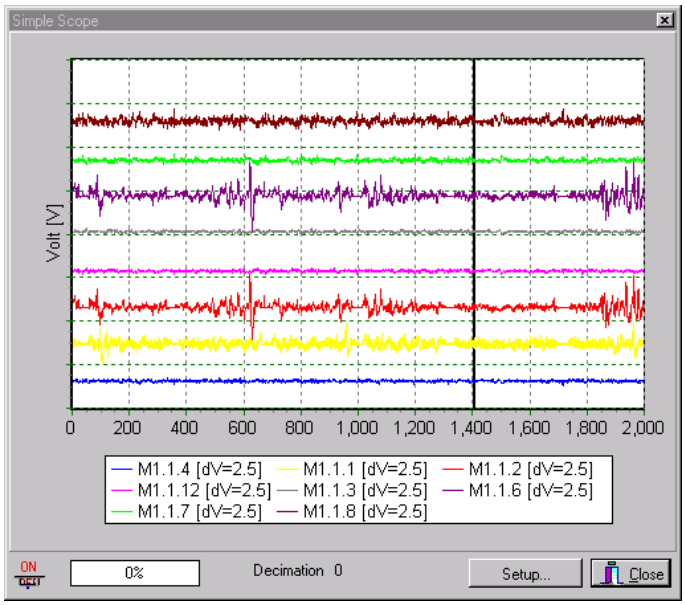

Fig. 6. Virtual scope

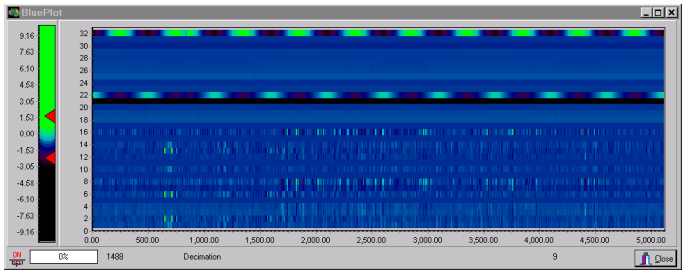

Fig. 7. Blueplot

C.2 Blueplot

The simultaneous visualization of a high count of channels in a scope does not give an appropriate overview of the acquired data. A better mode of visualization is provided by a so called blue plot (Fig. 7). For each channel a colored bar is shown, whereby the amplitude of the channel is coded into a color value. Small amplitudes in the order of the root mean square of the signal are mapped onto blue color values with mean intensity. High positive and negative amplitudes are mapped onto green with high intensity or onto red with low intensity, respectively. Due to the variation of intensity the blueplot can also be shown using gray colors whereby the information about the amplitudes (high negative or positive, mean) is preserved. The mapping of amplitudes can be either manually or automatically adjusted in the color bar. The user can also define a threshold

C.3 Spectrogram

One example for online processing of raw data and visualization is the spectrogram. The computation of the windowed FFT of a certain channel is done on the DSP. The log magnitudes of the Fourier coefficients are computed by the DAQ client and the result is mapped onto a color. The

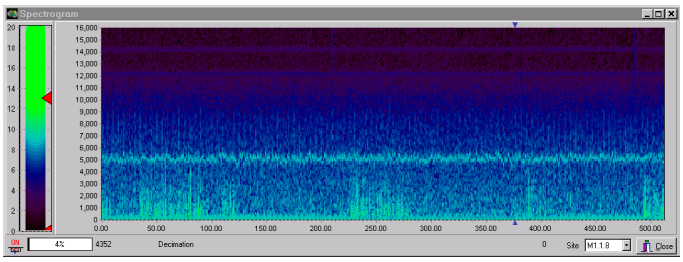

Fig. 8. Spectrogram

color mapping can be manually adjusted, such that only a range of interest is displayed (Fig. 8).

V. Concluding Remarks and Future Work

The described data acquisition system provides basic data acquisition functionality on a flexible hardware basis build from off-the-shelf components. The software provides a modular framework for future extensions. Already implemented is a windowed Fourier transform, which result is used to display the spectrogram. Future work will include development and implementation of methods for online and offline data processing like filtering with high-pass and lowpass FIR filters, compression based on wavelet decomposition [3], spike detection and spike sorting [4–7].

Acknowledgment

We grateful acknowledge the help of our project partners at ACREO AB (Kista, Sweden), Thomas RECORDING (Giessen, Germany), Center for Sensory-Motor Interaction at Aalborg University (Denmark), Instituto Nationale Neurologico "Carlo Besta" (Mailand, Italy), Theoretical Neurobiology at University of Antwerp – UIA (Belgium).

References

[1] Ulrich G. Hofmann, Erik De Schutter, Ken Yoshida, Marco De Curtis, Uwe Thomas, and Peter Norlin, "On the design of multisite microelectrodes for neural recordings," in *Proceedings of the MICRO.tec 2000*, P. M. Knoll, Ed., Berlin, 2000, vol. 1, pp. 283 – 288, VDE Verlag.

[2] Ken Yoshida, W. Jensen, Peter Norlin, M. Kindlundh, and Ulrich G. Hofmann, "Characterization of silicon microelectrodes from the EU VSAMUEL project," in *Biomedizinsche Technik*, J. Werner, Ed., Bochum, 2001, Schiele & Schön GmbH, to appear.

[3] B. Weber, T. Malina, K. Menne, A. Folkers, V. Metzler, and U. G. Hofmann, "Handling large files of multisite microelectrode recordings for the European VSAMUEL consortium," *Neurocomputing*, Jun 2001.

[4] M. S. Lewicki, "A review of methods for spike sorting: the detection and classification of neural action potentials.," *Network*, vol. 9, no. 4, pp. 53–78, Nov 1998.

[5] J. C. Letelier and P. P. Weber, "Spike sorting based on discrete wavelet transform coefficients," *Journal of Neuroscience Methods*, vol. 101, no. 2, pp. 93–106, Sep 2000.

[6] M. S. Fee, P. P. Mitra, and D. Kleinfeld, "Automatic sorting of multiple unit neuronal signals in the presence of anisotropic and non-gaussian variability," *Journal of Neuroscience Methods*, vol. 69, no. 2, pp. 175–188, Nov 1996.

[7] Maneesh Sahani, *Latent Variable Models for Neural Data Analysis*, Ph.D. thesis, California Institute of Technology, Pasadena, California, 1999.