**Development of a Scalable High Performance Reconfigurable Real-Time Signal Processing Platform for Dynamic Data-Driven Neural Simulations and Modeling**

**Requested Amount**: $36,000

**Requested Starting Date**: 01/01/01

**Check Appropriate Box(es) if This Proposal Includes Any of the Items Listed Below**
- Beginning Investigator (GPG 1.A.3)
- Disclosure of Lobbying Activities (GPG II.D.1)
- Proprietary & Privileged Information (GPG II.D.10)
- National Environmental Policy Act (GPG II.D.10)
- Historic Places (GPG II.D.10)
- Small Grant for Explor. Research (SGER) (GPG II.D.12)
- Vertebrate Animals (GPG II.D.12) IACUC App. Date
- Human Subjects (GPG II.D.12) Exemption Subsection or IRB App. Date
- International Cooperative Activities: Country/Countries
- Facilitation for Scientists/Engineers with Disabilities (GPG V.G.)
- Research Opportunity Award (GPG V.H.)
Project Summary

A high performance reconfigurable signal-processing platform will be built with the goal of performing real-time on-line analysis of large-scale multi-channel data streams for data-driven simulations and modeling. The computational architecture will be a distributed real-time system of modular design consisting of computational nodes that will form a three-dimensional mesh. A computational node will include a floating-point digital signal processor (DSP), a field programmable gate array (FPGA), and local memory. The DSP used will be the ADSP-21160 SHARC from Analog Devices that can easily be configured in three-dimensions by using its six link ports to connect to neighboring DSPs. The FPGAs used will be based on Xilinx's Virtex family that have high speed I/O capability and will be configured to connect to the DSP and neighboring FPGAs. This will allow the system to be reconfigurable where algorithms can be directly implemented in hardware. The FPGAs will also be configured as communication processors, allowing significant bandwidth for communication between computational nodes. Configuring the system as a three-dimensional mesh will allow the system to scale to any number of computational nodes required to process an arbitrary number of real-time I/O data streams.

The research test-bed application will be the analysis of signal processing in a simple nervous system. Specifically, the platform will be used to discover the cooperative neural encoding schemes through which sensory information is represented and transmitted within a nervous system. The system will enable real-time decoding of the neural information streams, and will enable experimental perturbation of the encoded information while the neural signals are in transit between peripheral and central processing stages. This will provide an unprecedented degree of interactive control in the analysis of neural function, and could lead to major insights into the biological basis of neural computation. It will allow a new paradigm in experimental and computational neuroscience where experimental and theoretical neuroscientists can work together to test hypotheses of neural function within in vivo neural systems.
<table>
<thead>
<tr>
<th>Section</th>
<th>Total No. of Pages in Section</th>
<th>Page No.* (Optional)*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cover Sheet (NSF Form 1207) (Submit Page 2 with original proposal only)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A Project Summary (not to exceed 1 page)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>B Table of Contents (NSF Form 1359)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>C Project Description (plus Results from Prior NSF Support) (not to exceed 15 pages) \  \ (Exceed only if allowed by a specific program announcement/solicitation or if approved in advance by the appropriate NSF Assistant Director or designee)</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>D References Cited</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>E Biographical Sketches (Not to exceed 2 pages each)</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>F Budget \  \ (NSF Form 1030, plus up to 3 pages of budget justification)</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>G Current and Pending Support \  \ (NSF Form 1239)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>H Facilities, Equipment and Other Resources (NSF Form 1363)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>I Special Information/Supplementary Documentation</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>J Appendix (List below.) \  \ (Include only if allowed by a specific program announcement/solicitation or if approved in advance by the appropriate NSF Assistant Director or designee)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Appendix Items:

*Proposers may select any numbering mechanism for the proposal. The entire proposal however, must be paginated. Complete both columns only if the proposal is numbered consecutively.

NSF Form 1359 (10/99)
Project Description

Results from Prior Support

The PI has been trained in electrical engineering and has had extensive experience performing electrophysiological experiments in both the visual and auditory cortex. As a result, the PI has had experience developing and refining data acquisition systems in order to record the necessary neurophysiological data. This experience has included the development of spike-sorting algorithms needed to acquire data in order to answer neural coding questions. As a result, the PI understands the scientific and technical problems associated with acquiring and analyzing neural data.

In the visual cortex the PI has examined the influences of the temporal firing patterns of presynaptic cat visual cortical cells on spike generation by postsynaptic cells. (Snider et al., 1998). Multi-unit recordings were dissected into the activity of individual neurons within the recorded group. Cross-correlation analysis was then used to identify directly coupled neuron pairs. Burst activity, defined by groups of two or more spikes with intervals of less than 8 ms from any single neuron, was analyzed in terms of its effectiveness in eliciting a spike from a second, driven neuron. Effectiveness was defined as the percentage of spikes from the driving neuron that are time related to spikes of the driven neuron. The effectiveness of bursts (of any length) in eliciting a time-related response spike averaged 18.5% across all measurements as compared with the effectiveness of single spikes, which averaged 9.5%. Longer bursts were more effective than shorter ones. Effectiveness was reduced with spatially nonoptimal, as opposed to optimal, stimuli. The effectiveness of both bursts and single spikes decreased by the same amount across measurements with nonoptimal orientations, spatial frequencies and contrasts. At similar firing rates and burst lengths, the decrease was more pronounced for nonoptimal orientations than for lower contrasts, suggesting the existence of a mechanism that reduces effectiveness at nonoptimal orientations. This experimental result supports the hypothesis that neural information can be emphasized via instantaneous rate coding that is not preserved over long intervals or over trials. It is consistent with the integrate and fire model, where bursts participate in temporal integration.

The PI has also been actively involved in developing spike sorting algorithms (Snider and Bonds, 1998). Although a number of methods have been proposed for classification of individual action potentials embedded in multi-unit activity, they have been challenged by non-stationarity. The waveform shapes of action potentials can change rapidly over time as a result of shifts in membrane conductances during extended burst firing sequences and more slowly over time due to electrode drift. An algorithm was developed for waveform identification that makes no assumptions on the distribution of these shapes other than the change in waveform shape for a particular neuron should not be discontinuous. The algorithm was applied to the resolution of multi-unit neural signals recorded in cat visual cortex and was compared to a spike sorting method based on the Bayesian likelihood of a spike fitting a particular model.

The PI using a digital signal processor, a host PC, and a SUN workstation implemented the spike-sorting algorithm at the Visual Information Processing Laboratory at Vanderbilt University. The PI wrote the software for the spike-sorting algorithm and developed a user interface running on the SUN workstation. The system is currently being used in the laboratory. An integrated data acquisition system was also developed for auditory neurophysiology at Johns Hopkins University. This involved developing a recording and stimulus generation software environment. The software was used to integrate various hardware components into a user-friendly system. The system is in current use by the Laboratory of Auditory Neurophysiology at Johns Hopkins University. The PI has also had experience developing speech recognition algorithms (Deller and Snider, 1990).

The PI is currently a faculty member in the electrical and computer engineering department at Montana State University and teaches computer engineering related courses as well as digital signal processing. The PI is therefore qualified to develop a real-time architecture for the real-time analysis of neural data.
Development Plan and Research Program

Currently, most neurophysiological experiments are done using relatively few electrodes. The experimental data is typically recorded to disk and then analyzed off-line. This is done to characterize single neurons in different areas of the nervous system. However, in order to determine what neural computations are taking place in a neural network requires recording the responses of many neurons. Recording from a single neuron is analogous to an individual trying to determine the inner workings of a supercomputer using just a single logic probe. Digital system designers know that they will need a logic analyzer with more than a hundred input channels to determine the logical function of today's complex computer systems. In a similar manner, neurophysiologists need a system where hundreds, if not thousands, of neural signals are recorded and processed in real-time.

The ability to interact in real-time with neural systems will be an important tool to develop computational models of neural processing. Computer simulations of complex neural processing are only as good as the implemented models, which for complex neural systems have a significant number of unknown parameters. What is needed is a system that can gather as much information as possible about the current state of a neural system, process this information in real-time, and then interact with the system at a later stage in the neural processing stream. This will allow hypotheses of neural processing to be tested without modeling every aspect of the nervous system, which is intractable if modeling is to be done realistically for sensory systems at the biophysical level.

Technological Challenges

As in many scientific and engineering disciplines, there are fundamental technical problems that must be solved in order to interact in real-time with neural systems. The research, for which this platform will be first applied, will need up to 100 electrodes to be sampled at rates of up to 50,000 samples (at 16 bit resolution) per electrode per second. This represents a composite data rate of approximately 10 megabytes per second. Although conventional systems currently allow the collection and storage of data at these rates for short-duration experiments, these systems do not provide the capability for deciphering the “meaning” imbedded within the neural signals on a timescale that will allow interactive control of the preparation. The goal is to record and decipher the coded information in the data stream at the sensory periphery rapidly enough that activity in the sensory axons can be modified centrally, before reaching the next processing stage, to test theories of neural encoding.

Several fundamental technological challenges must be met before such experiments are possible. Devices must be developed to allow real-time on-line analysis of such massive data streams, with the potential to scale up to thousands of electrodes. The on-line analysis will need to be extremely sophisticated, and provide for extraction and recognition of complex signal features. For example, advanced statistical techniques (including information theory) must be implemented to discover the intrinsic symbols within the raw data stream signals; only the event times and/or amplitudes of these features would need to be stored and be used for experimental control. This will provide researchers with the means to predict the evolving state of the system with reasonable accuracy. This would allow a huge reduction in the amount of data to be stored, reducing post-processing tasks, and would also allow for levels of interactive control never before achieved.
System Specifications

The general specifications for a system that will be able to process a large number of channels in real-time are the following:

- The system must be scalable to allow an arbitrary number of input/output (I/O) channels. It also must be scalable to the number of processing elements required to process these I/O channels, and it must be scalable in terms of the communication bandwidth required between the processing elements.
- Data acquisition, spike separation and pattern classification will need to be fast enough in this system to allow real-time experimental manipulation of the biological information stream.
- Any processor must be able to communicate with any other processor, under real-time constraints.
- A processor must be able to implement signal-processing algorithms efficiently.
- The system should be reconfigurable, allowing signal processing and/or communication algorithms to be directly implemented in hardware.

The specific applications for which the device will be used include the following:

- The system will be used to discriminate and classify neuronal action potentials (spikes) in real time. Each input channel will be the waveform from a single recording electrode from a multi-electrode array. The signal on each electrode will appear as a complex continuous waveform, formed by the superposition of the electrical activity from multiple nerve cells. After correlative processing with the algorithms discussed later, each output channel will correspond to the activity pattern generated by an individual nerve cell. A large transient pulse on any output channel will correspond to the occurrence of an action potential in the corresponding nerve cell.
- Coherent patterns of spikes across multiple nerve cells will be identified and classified in real time, through a filter-based comparison to a set of multi-unit template patterns. These templates can be pre-defined and/or derived through identification of recurrent patterns in the data set itself.
- The coherence between these ensemble spike patterns and the sensory stimulus signal will be computed on-the-fly, and used to calculate the mutual information between stimulus and response in these neural systems.

System Architecture

The architecture proposed to meet these technological challenges will be a distributed real-time system. This will allow processing to be done that is constrained by time. The system will have a modular design, where the processing elements (computational nodes) will form a three-dimension mesh. Each computational node will be constructed so that it will be able to connect to and communicate with another node on any of its six sides. Thus a computational node will have six communications ports that will allow it to communicate in 3-dimensional space as seen in figure 1.

![Figure 1. A Computational Node. The communication ports of the computational node will be configured to communicate with neighboring nodes in 3-dimension space.](image-url)
The computational nodes will be constructed so that they will easily form a 3-dimensional mesh as seen in figure 2. This will allow the system to be scalable to an arbitrary number of I/O channels and processing elements.

![Figure 2. An illustrative example of a 3-dimensional mesh comprised of 27 computational nodes in a 3x3x3 matrix.](image)

To process the necessary input/output channels in real-time, multiprocessor capable DSPs and FPGAs will be used. The DSPs to be used will be the new generation of Analog Devices SHARC Processors (ADSP-21160's) and the FPGAs used will be the new generation of Xilinx's FPGAs (Virtex FPGAs). The reason for using both a DSP and FPGA in a computational node is that the SHARC DSPs are optimized for floating point DSP operations while the FPGAs are reconfigurable, which will allow computational algorithms to be directly implemented in hardware. The reconfigurable FPGAs will either be used as coprocessors, separate processors, or communication engines. The particular configuration a FPGA will take will depend on the computational needs of the algorithm being run on a particular computational node. A diagram of the computational node can be seen in figure 3.

The ADSP-21160 SHARC from Analog Devices is a 32-bit processor that is optimized for high performance floating-point DSP applications being executed in a multiprocessor configuration. The DSP, running at 100 MHz, has 6 external link ports and an on-chip DMA controller that allows the processor to handle 600 Mbytes/sec of streaming data without processor intervention. This means that the processor can execute algorithms at full speed (600 MFLOPS peak, 400 MFLOPS sustained) while the DMA controller handles data being sent to and from the processor. The six 4-bit link ports allow the DSP processors to be configured in a 3-dimensional mesh, which will allow communication with neighboring SHARC DSPs. Each DSP has 4 Mbits of internal SRAM, which can be configured in varying combinations of 32-bit data and 48-bit instructions.

The 64-bit bus of the SHARC DSP will be interfaced to the FPGA allowing the FPGA to function as a memory mapped 64-bit coprocessor. The SHARC bus will also be connected to the Virtex SelectMAP™ port of the FPGA, which is an 8-bit interface that allows high-speed (400Mb/s) reconfigurations of the device. As a result, a large capacity FPGA can be configured in approximately 20 msec. This will allow the SHARC DSP to control the functionality of the FPGA for signal processing cooperation. In addition, the SHARC bus will be connected to SDRAM memory allowing standard SODIMM memory modules to be used. This will allow standard memory capacities of 128 or 256 MBytes to be implemented economically. The FPGA will also function as a SDRAM controller for the SHARC DSP. EPROM will be used to boot the SHARC DSP and load a real-time operating system kernel. The EPROM will also store several default configurations for the FPGA. FLASH memory will be used to store variable configuration data and other parameters that need to be maintained during system power down.
The FPGA to be used will be based on Xilinx's Virtex-E series. These FPGAs have densities ranging from 50,000 to 3.2 million system gates and support 20 I/O standards including low voltage differential signaling (LVDS). Using LVDS, a Virtex FPGA can have up to 344 differential pairs that operate at 311 Mb/s for an aggregate I/O bandwidth of 13.4 GB/s. The FPGAs will be configured to implement these high performance communication channels. Using Xilinx's SelectLink™ technology, these channels are easily designed providing configurable internal and external bus widths and are implemented with FIFO data buffers in the transmitter and receiver modules.

The FPGAs will be designed to communicate with six of their nearest neighbors. The SelectLink™ ports will be configured to have a fixed 16-bit external bus. Since these ports transmit data in only one direction, two of these ports will be connected to each of the six neighboring FPGAs allowing bi-directional communication. Internally, the data bus width can be configured to the size of the data type being processed, highlighting the configurable nature of these FPGAs.

A systolic array architecture can then be implemented on the three dimensional mesh of FPGAs where the FPGAs will be configured as both processors and communication channels. This will allow the development of reconfigurable parallel hardware that supports the specific computational needs of algorithms being run on the distributed real-time system.
There is substantial industrial support for the Virtex series of FPGAs. Xilinx has a CORE generator that will automatically generate hardware description language (HDL) code to implement DSP building blocks such as FIR filters, a 1024 point FFT, a Sine/Cosine look up table, multipliers, a divider, and an accumulator. Xilinx and Mathworks have formed a partnership to develop the Xilinx System Generator for Mathwork's Simulink, which will be available third quarter 2000. This will enable designers to build and verify an entire DSP system and then automatically generate a HDL representation compatible with Xilinx FPGAs. The tool will automatically map the DSP design to the Xilinx LogiCORE building blocks for optimal implementation, highest performance, and lowest silicon cost. Thus DSP designs can be prototyped in Mathwork's MATLAB/Simulink environment, which is widely used in industry/academia engineering environments, and then implemented in hardware via FPGA's. Further industrial support is highlighted by the Virtex roadmap that includes the Virtex-II architecture where Xilinx is partnering with IBM to embed the IBM PowerPC processor core. This will provide the future potential for even greater floating-point performance for the computational nodes.

The performance capability of a computational node is listed in Table 1. The SHARC DSP has the advantage of being optimized for single cycle floating-point operations while the FPGA has the advantage of being reconfigurable, which will allow fixed point DSP operations to be implemented at very high processing speeds.

<table>
<thead>
<tr>
<th>Device</th>
<th>SHARC (100 MHz)</th>
<th>FPGA (155.5 MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advantage</td>
<td>Floating point processing</td>
<td>Reconfigurable Computing</td>
</tr>
<tr>
<td>Performance</td>
<td>600 MFLOPS(^1) Peak</td>
<td>32 Billion MACs(^2) (fixed point)</td>
</tr>
<tr>
<td></td>
<td>400 MFLOPS Sustained</td>
<td></td>
</tr>
<tr>
<td>I/O Bandwidth</td>
<td>100 MBytes/Sec/Link Port</td>
<td>622 MBytes/Sec/16-bit SelectLink Port</td>
</tr>
<tr>
<td></td>
<td>600 MBytes/Sec Total Bandwidth</td>
<td>7.46 GBytes/Sec Total Bandwidth</td>
</tr>
</tbody>
</table>

1. MFLOPS = Million Floating Point Operations/Sec
2. MACs = Multiply and Accumulate Operations/Sec

Table 1. Summary of the Computational and I/O capabilities of a Computational Node.

The modular computational nodes will have board-to-board physical connectors on each of their six sides allowing them to be connected together in a 3-dimensional mesh. This will allow the system to scale to any number of processors. The system will also need to scale to any number of input/output channels. Since the input/output channels will be integral to dynamic data-driven simulations, several variations of the basic computational node (as illustrated in figure 3) will be designed. These variations will be implemented via small mezzanine boards that will be attached to the computational nodes. One variation will be a computational node that contains analog-to-digital (A/D) converters. These nodes will be used to acquire real-time neural data and will be placed on the outside surface of the 3-dimensional mesh. Similarly, computational nodes will be designed with digital-to-analog (D/A) converters that will interface with stimulating electrodes to provide real-time stimuli.

In order to visualize the inputs, outputs, and computations taking place, a computational node variation will be designed with a 24-bit RAMDAC that will allow color video output. This is necessary in order to assess the recording quality of the real-time neural data and to visualize the computations that are occurring. An external computer interface will also be implemented to control the system and provide data storage.
As the number of computational nodes increases, the communication time between the furthest nodes (e.g., corner nodes) will also increase. The longest communication time is proportional to the cube root of the number of processors and depends on computational nodes passing messages between these nodes. In order to reduce this communication time, a subset of computational nodes will be developed with multiple fiber optic interconnections. This will allow the formation of a partial hypercube topology that will substantially reduce communication times between computational nodes. For example, this will allow edge nodes to communicate with nodes on the opposite side of the 3-dimensional mesh.

The initial DSP based system will have 27 computational nodes implemented in a 3x3x3 three-dimensional mesh. This will provide 10.8 GFLOPS of sustained floating point-computations and 0.9 teraMACs of fixed-point computations. The initial system will be connected to approximately 30 neural recording channels and 10 stimulation channels. The projected data processing rate achievable with this system will be much higher than the conduction velocity of action potentials through the peripheral sensory processing stages, allowing input signals can be analyzed and manipulated experimentally before the information stream reaches subsequent processing stages. The overall block diagram of the hardware system can be seen in figure 4. Note: only two dimensions are shown.

System Level Issues

The dynamic data driven neural simulation platform is a distributed real-time system. The characteristic of a distributed real-time system is that many processes are running on different processors that have timing constraints (Stankovic & Ramamritham, 1988). The timing constraint is based on the fact that the system must react to neural events that are occurring and processing this information fast enough to provide an output to interact with the neural system under study.

Real-time systems can be classified as either hard real-time systems or soft real-time systems. Hard real-time systems must meet time constraints in order to avoid disastrous consequences. Soft real-time systems are still considered functionally correct if the time constraints are not seriously violated. Since a simulation paradigm is to function as a neural processing layer, replacing a layer in an in vivo neural system, the proposed system will have a time constraint as to when stimulation should occur at the next layer. Thus, the proposed system will be considered a hard real-time system, since if timing constraints are violated, the interactive paradigm would not be valid.

For hard real-time systems, there are several aspects of the system that must be accomplished that are related to time (Burns & Wellings, 1990). These include a mechanism to keep track of time, a way to delay tasks, a way to handle time-outs, and a manner to schedule tasks to meet specified deadlines. Clearly, a system must have access to a clock to keep track of time. This is typically done by a processor incrementing a counter in response to a periodic interrupt that is based on the processor's own clock. In a distributed real-time system each processor will typically have its own clock and thus it is difficult to determine a precise measure of "global" time in order to determine the global state of the system (Tsai et al., 1996). Thus debugging a distributed real-time system in order to find timing constraint violations is a more difficult task than with non-distributed real-time systems.

The ability to delay a task is also important, because if the task is waiting for a neural event to occur, another task can be run instead of using up CPU cycles in a busy wait. If a task waits too long for some event, it may violate a timing constraint and thus there must be a mechanism to handle time-outs. These issues are addressed by a real-time operating system and thus a real-time operating system will be used on the SHARC DSPs. The choice of a real-time operating system vender will be chosen in regard to the support of the SHARC link ports and the debugging capabilities they support since this will be crucial for developing a distributed real-time system for this architecture.
Figure 4. An Illustration of the Hardware System. The basic computational node will have optional circuitry such as A/D converters, D/A converters, fiber optic links, video output, & external computer interfacing.
Software Development Objectives

Since the purpose of the real-time system is to provide the capability for interactive data-driven neural simulations and modeling, specific software must be developed and implemented to provide the necessary functionality. New algorithms will be developed for acquiring, analyzing, visualizing, interpreting and storing relevant aspects of massive multi-channel data streams. The specific functions and software necessary for the DSP based distributed real-time system are listed below. The relationships between these software elements can be seen in figure 4.

- A real-time operating system on each computational node.
- Routing software to send the real-time data streams to the appropriate computational nodes and link ports.
- Spike discrimination software to identify action potentials in the real-time neural signals.
- Correlation measure software to identify correlations among the multi-channel inputs.
- Data visualization software to assess the quality of the recorded neural signals and to visualize the simulations taking place.
- Simulation and modeling software to implement models of neural processing.
- Stimulus synthesis software to provide real-time stimuli for interactive simulations.
- FPGA configuration software to change the functionality of the FPGA in response to the needs of the computational algorithms.

The real-time operating system (RTOS) to be used will be Eonic's Virtuoso v4.1 RTOS and development environment. This operating system has been developed for the SHARC DSP and provides routing support using the SHARC link ports. This will substantially reduce software development time for the real-time system. Software functions will be implemented as tasks running on the real-time operating system and the link ports will be used to pass messages and data between the tasks. The operating system is capable of finding and using the shortest path through the 3-dimensional mesh to pass messages between communicating computational nodes. By utilizing the SHARC's built in DMA controllers, passing messages and data will not interfere with numerical processing that will be executing on the DSPs.

Spike discrimination software will be implemented on the distributed real-time system. Software will be implemented so that the user will have the choice of various spike-sorting methods available, such as adaptive linear filtering (Gozani and Miller, 1994), templates (Worgotter et al., 1986), principle components (Gerstein et al., 1983), Bayesian spike sorting, (Lewicki, 1994), and cluster-based approaches (Fee et al., 1996; Snider and Bonds, 1998).

Correlation algorithms will be developed to enable use of the massive parallelism intrinsic to this specialized hardware. In particular, the extremely high rates of data will pose a number of challenges. The investigation of large or high-dimensional data sets is a problem of recent interest in computer science (Indyk et al., 1999). Because of the enormous data rates, data processing must be done online. Online problems are well-studied in traditional computer science domains (e.g., virtual memory page replacement policies, Karlin et al., 1996), so we expect to adapt some existing methods to multi-channel data analysis. Specific problems of interest include clustering similar signals, recognizing common patterns in signals, inferring causality between signal patterns, and developing and refining models of observed activity.
Figure 4. Functional blocks to be implemented on the distributed real-time system.
Various different inter-channel correlation measures will be implemented, including measures based on mutual information (Cover and Thomas, 1991), cross-correlation (Perkel et al., 1967), joint post-stimulus time histograms (Aertsen et al., 1989), typical sequence analysis (Johnson et al., 2000). New algorithms will also be implemented, including one based on data structures representing probabilistic suffix trees (Ron et al., 1994) and one based on jointly-typical sequences (Dimitrov and Miller, 2000).

Visualization of the data at various stages of analysis will be essential to comprehending the encoding of information in neural spike trains, and will be essential for real-time interactive control of the experiments. Real-time analysis and visualization experimental modes require linking multiple hardware and software components into a robust, flexible system. Thus the appropriate visualization software will be developed to monitor the real-time computations taking place.

The simulation and modeling software will be specific to the system under study and will be developed in cooperation with Dr. Miller at the Center for Computational Biology at Montana State University. Stimulus software will also be developed that will take output from the modeling software and will generate real-time stimuli that will be used to interact with the neural system under study.

The FPGAs will be used to accelerate software algorithms. This will be done by profiling software algorithms and determining the computationally intensive segments of code that may not meet real-time constraints. These code segments will then be mapped to hardware and implemented on the FPGAs. It is likely that multiple FPGAs will be necessary for complex algorithms and will be configured as systolic arrays.

**Software Design Environment**

The software design environment will be based on 4 commercial software packages. The real-time operating system environment will be based on Eonic's Virtuoso development environment for embedded DSPs. Using the virtual single processor model will allow code to be written for 1000 processors the same way it is written for one. It also provides debugging support at the task level for distributed real-time systems. This will be used in conjunction with the Analog Devices VisualDSP software environment, which provides compiler, linker, assembler, and low level debugging support for SHARC DSPs.

To implement signal-processing functions on the SHARC DSP, the Real-Time Workshop from Mathworks will be used in conjunction with Simulink and Matlab. This will allow DSP algorithms to be easily prototyped in the Matlab/Simulink programming environment and then converted to C code using the Real-Time Workshop. The code will then be compiled in the VisualDSP environment to be implemented on the SHARC DSPs.

Implementing signal processing functions for the FPGAs will be similar to how they are implemented on the SHARC DSPs. DSP algorithms will be prototyped in the Matlab/Simulink programming environment. The algorithms will then be converted to their fixed point representations using the fixed-point block sets found in Simulink. Once the algorithms have been verified using a fixed-point representation, they will be converted to VHDL code using the Xilinx System Generator for Mathworks Simulink. The VHDL code will then be compiled for the FPGAs using Xilinx's Foundation series software.
Neuroscience Research to be Performed with the Proposed System

The research test-bed application will be the analysis of signal processing in a simple nervous system. Specifically, the system will be used to discover the cooperative neural encoding schemes through which sensory information is represented and transmitted within a nervous system. The system will enable real-time decoding of the neural information streams, and will enable experimental perturbation of the encoded information while the neural signals are in transit between peripheral and central processing stages. This will provide an unprecedented degree of interactive control in the analysis of neural function, and could lead to major insights into the biological basis of neural computation.

The testing and refinement of the devices proposed here will be achieved through neurophysiological research carried out by Dr. John Miller, along with his grad students, post doctoral researchers and technical staff. Dr. Miller is a senior experimental neurophysiologist at Montana State University, who moved to MSU three years ago from U.C. Berkeley to become the founding director of the Center for Computational Biology (CCB). No specific funding is requested for the direct support of Dr. Miller’s research, which is well funded through NIH and NSF.

Summary of Specific Neuroscience Objectives

The ultimate scientific goal of Dr. Miller’s studies is to understand the neural basis of information encoding and processing in sensory systems. Examples of specific questions to be addressed include the following: a) What are the stimulus features encoded by neural ensemble activity patterns? b) What are the time scales over which meaningful representations of stimuli become established in neuronal ensembles? c) What are the correlates of a single information channel? d) What are the encoding schemes operating on those channels? In particular, what are the functional correlates of ensemble synchronization? The solution to these and many other fundamental unsolved questions related to nervous system function will require a level of dynamic on-line data stream analysis and interactive control of experimental parameters that is currently unattainable, and which appear unlikely to be attainable through a scale-up of existing technologies. Development of the hardware and associated software technology proposed here will enable fundamental breakthroughs in understanding the general problem of neural computation.

The neurophysiological preparation to be used as the test bed is the cercal sensory system of the cricket. The cercal system mediates the detection, localization, and identification of behaviorally-relevant air currents in the animal’s immediate environment, and can be thought of as a low-frequency, near-field extension of the animal’s auditory system. The sense organs for this modality are two antenna-like cerci at the rear of the abdomen. Each cricket cercus is covered by approximately 1000 mechanosensory hairs, which are deflected by air currents in the animal’s immediate environment. The entire sensory epithelium for the cricket filiform system consists of the 2000 receptors, which innervate these hairs. Afferent axons from these receptors project into the terminal abdominal ganglion, where they make synaptic connections onto approximately 50 sensory interneurons. The entire output layer of this system consists of only 20 of these interneurons. These 20 cells are the only neurons associated with the filiform air-current-detection system, which send axons out of the ganglion to higher centers. All of these projecting interneurons are identified, and the activity of all of these interneuron axons can be monitored and individually identified simultaneously using multi-channel extracellular recordings from nerves.

Previous neurophysiological studies in Dr. Miller’s lab have shown that a significant quantity of information about the direction and velocity of mechanosensory stimuli is represented by cercal neurons with a linear rate code (Landolfa & Miller, 1995; Levin & Miller, 1996; Miller et al., 1991; Roddey et al., 2000; Theunissen & Miller, 1991; Theunissen & Miller, 1995; Theunissen et al., 1996). However, his recent studies demonstrate that there is also a substantial amount of information in the spike trains that
cannot be accounted for by a simple linear encoding scheme. Evidence suggests the implementation of a nonlinear ensemble encoding scheme in this system, which allows for a substantial increase in information encoding, transmission and processing capabilities over those which are achievable with linear schemes. A central goal of the studies of Dr. Miller’s ongoing research program is to discover the biological basis for these extremely efficient information extraction, encoding, transmission and processing schemes.

To achieve the specific objectives listed above, new approaches are being developed by personnel in Dr. Miller’s and Dr. Snider’s labs for analyzing, visualizing and interpreting the massive data streams that are generated during Dr. Miller’s experiments. New paradigms are being used to validate and refine large-scale system models. Those paradigms center on the use of information theory to estimate the amount of mutual information between the ensemble neural activity patterns and dynamic parameters of the sensory stimuli. In all studies to date, all aspects of the data analysis are carried on off-line, after the data has been recorded. That is, there is no capability for on-line interactive control or analysis of the experiment. Through the work proposed here, hardware will be developed to enable a high level of on-line analysis and control capabilities.

**Experimental methods**

The **general experimental strategy** will be to present air current or tactile stimuli to the animals and record the ensemble responses of sensory afferents and interneurons. Multunit recording and spike separation techniques (Gozani & Miller, 1994) will allow the simultaneous monitoring of the activity patterns of all 20 projecting interneurons in the sensory system’s output stage, up to 20 receptors in the input stage, and up to 32 motor neurons projecting out of the thoracic ganglia. The sensory stimuli used in these studies include stochastic and deterministic air current waveforms, many of which are constructed to mimic behaviorally-relevant signals that the animal would be expected to encounter in its natural environment.

The **general strategy for data analysis** will be to examine the statistical relationship between a) the stimuli and ensemble neural responses, and b) the stimuli and mechanical responses, using the principles of information theory to derive objective measures of significance.

The **general framework for the interpretation** of our results will be a model of the functional organization of the system, formulated in terms of the *multiple parallel information channels*. All of the basic analytic approaches are already used in Dr. Miller’s lab. A new paradigm will be used to validate and refine the large-scale model to be developed in the later stages of the proposed work. That paradigm will center on a) the use of information theory to estimate the amount of mutual information between the ensemble neural activity patterns, motor responses and dynamic parameters of the sensory stimuli and b) the use of advanced multivariate statistical techniques and dynamical systems theory to identify and understand the functional significance of coherent activity across ensembles of neurons (Dimitrov & Miller, 2000).

While development of the hardware aspect of the system is proceeding, we will implement simple but efficient forms of the same filtering and correlational methods on the Origin 2000 computer in the Center for Computational Biology. Thus we will create rapidly, in software, the basis for experiments with higher-order algorithms, while the hardware is in development. The higher-order algorithms will include, for example, linear-estimation and clustering techniques, as well as more sophisticated behavior- and task-related statistical models. Visualization tools will be developed in a common framework, but tailored for specific modalities and test preparations. We will study of the feasibility of various topologies for implementing real-time analysis of multi-channel data. Linking the components into a flexible, robust, efficient pipeline will require development of appropriate data transmission protocols and component
synchronization methods. The Origin 2000 will be used for development and implementation of the advanced multi-dimensional graphical user interface to the proposed system.

**Facilities and site of performance of the research**

The proposed research (and associated training) activities will be distributed between two locations in close proximity to one another on the MSU campus. One is in the Snider laboratory where hardware and software will be developed for the distributed real-time system. The lab will house 3 graduate students who will develop the system hardware and software under the supervision of Dr. Snider.

The second location is the newly-renovated ground floor of Lewis Hall, which houses the Center for Computational Biology (CCB). Three neuroscientists have their labs and offices in the CCB: John Miller, Gwen Jacobs and Charles Gray, who just moved to MSU from U.C. Davis. The Silicon Graphics Origin 2000 computer, a large array of common-use SGI and Linux workstations, and 7 advanced electrophysiological workstations are also located in the CCB. (NOTE: The Origin 2000 supercomputer was purchased in part by a grant from NSF, and in part from a grant from Silicon Graphics.) After construction in the Snider lab, the prototype system will be moved to the Lewis Hall site for interfacing with the neurophysiology experimental stations.

**Scope of Potential Use**

While there are commercial boards that contain multiprocessor DSPs or FPGAs, there are no commercial off the shelf (COTS) boards that combine the capabilities of both, which are packaged in a form that will allow them to be connected in a 3-dimensional mesh. This is important for the system to be scalable to any number of I/O channels and computational nodes. Further there are no circuit boards that implement parallel reconfigurable computing by implementing FPGAs as signal processing coprocessors that will dramatically speed up fixed-point algorithms by implementing them in hardware.

Understanding the nature of neural computations is of great importance since it is one of the fundamental goals of neuroscience research. The proposed system will enable hypotheses of neural computation to be directly tested in living neural systems. The system is essential for this type of research since it is necessary to perform real-time decoding of neural information streams, and it will enable experimental perturbation of the encoded information while the neural signals are in transit between peripheral and central processing stages. This will provide an unprecedented degree of interactive control in the analysis of neural function, and could lead to major insights into the biological basis of neural computation.

This system will provide an enabling platform for experimental and computational neuroscientists to collaborate for potentially breakthrough neuroscience research. The potential number of users will be proportional to the number of neurophysiologists and computational neuroscientists studying neural systems. The potential research areas where this could be applicable will also be proportional to the number of neural systems being studied. Thus there is significant potential for a large number of users and research areas in neuroscience where the proposed system could be used.

**Education and Human Resources**

The hardware development program will have a significant impact on the training of engineering and science students at the undergraduate, graduate, and postdoctoral levels. At the engineering undergraduate level it will provide opportunities for students to participate in the development of an advanced computational platform where they can develop both hardware and software skills. This project will also provide illustrative examples to bring to life relevant course material for a number of undergraduate and graduate engineering courses such as Logic Circuits, Logic Design, Microprocessor

For engineering graduate students it would provide a platform for them to study distributed real-time systems and reconfigurable computing and the issues associated with such systems, such as real-time operating systems, monitoring and debugging, and parallel processing. For neuroscience graduates and postdoctoral associates it would provide an enabling platform to start exploring issues related to interactive computational neuroscience. This will allow questions of neural processing to be directly tested within in vivo systems.

Graduate students working on components of the intended research will be drawn from (and will integrate with) the established graduate training programs in the Departments of Electrical and Computer Engineering, Computer Science, Mathematics, and Cell Biology & Neuroscience. Of even more significance will be the excellent match between the proposed research program and the new interdisciplinary graduate training program in the *Structural and Functional Analysis of Complex Biological Systems*, funded through the most recent round of NSF IGERT funding (see the *Complex Biological Systems* website: www.cbs.montana.edu). Snider and Miller will integrate the proposed research into the CBS Graduate Training program, including integration of experimental, theoretical and computational approaches, through essential involvement of mathematicians and computational scientists with experimentalists, in research and academic training.

This project will also allow the opportunity to work with a current NSF funded program within the college of engineering to increase the number of computer scientist, engineers, and mathematicians for low-income students. The program provides scholarships to high-achieving but financially needy juniors, seniors, and graduate students. The program also has a special emphasis to increase the retention of women and Native American students (Montana's largest minority group) in engineering programs. One of the requirements for these students is to participate in an active research laboratory. It is the intent of this proposal to work with this program by providing engineering and neuroscience research opportunities for such identified students.
References Cited


